PY32F040-EP Reference Manual

32-bit ARM® Cortex®-M0+ Microcontrollers

Puya Semiconductor (Shanghai) Co., Ltd.

Contents

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

Puya Confidential

1. Documentation conventions

1.1. List of abbreviations for registers

Abbreviation	Description
Read/Write (RW)	Software can read and write to this bit.
Read-only (R)	Software can only read this bit.
Write-only (W)	Software can only write to this bit.
Read/Clear Write 0 (RC_W0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on this bit.
Read/Clear Write 1 (RC_W1)	The software can read this bit or clear this bit by writing 1, writing 0 has no effect on this bit.
Read/Clear Write (RC_W)	Software can read as well as clear this bit by writing to the register. The value written to this bit is not important.
Read/Clear by Read (RC_R)	Software can read this bit. Reading this bit automatically clears it to 0.Writing this bit has no effect on the bit value.
Read/Set by read(RS_R)	Software can read this bit. Reading this bit automatically sets it to 1, writing this bit has no effect on its value.
Read/set (RS)	The software can read this bit or set it to 1. Writing 0 has no effect on this bit.
Toggle (T)	The software can toggle this bit by writing 1. Writing 0 has no effect.
Reserved (Res.)	Reserved bit, must be kept at reset value.

1.2. Availability of peripherals

For availability of peripherals and their number across all sales types, refer to the particular device datasheet.

Puya Confidential

2. System architecture block

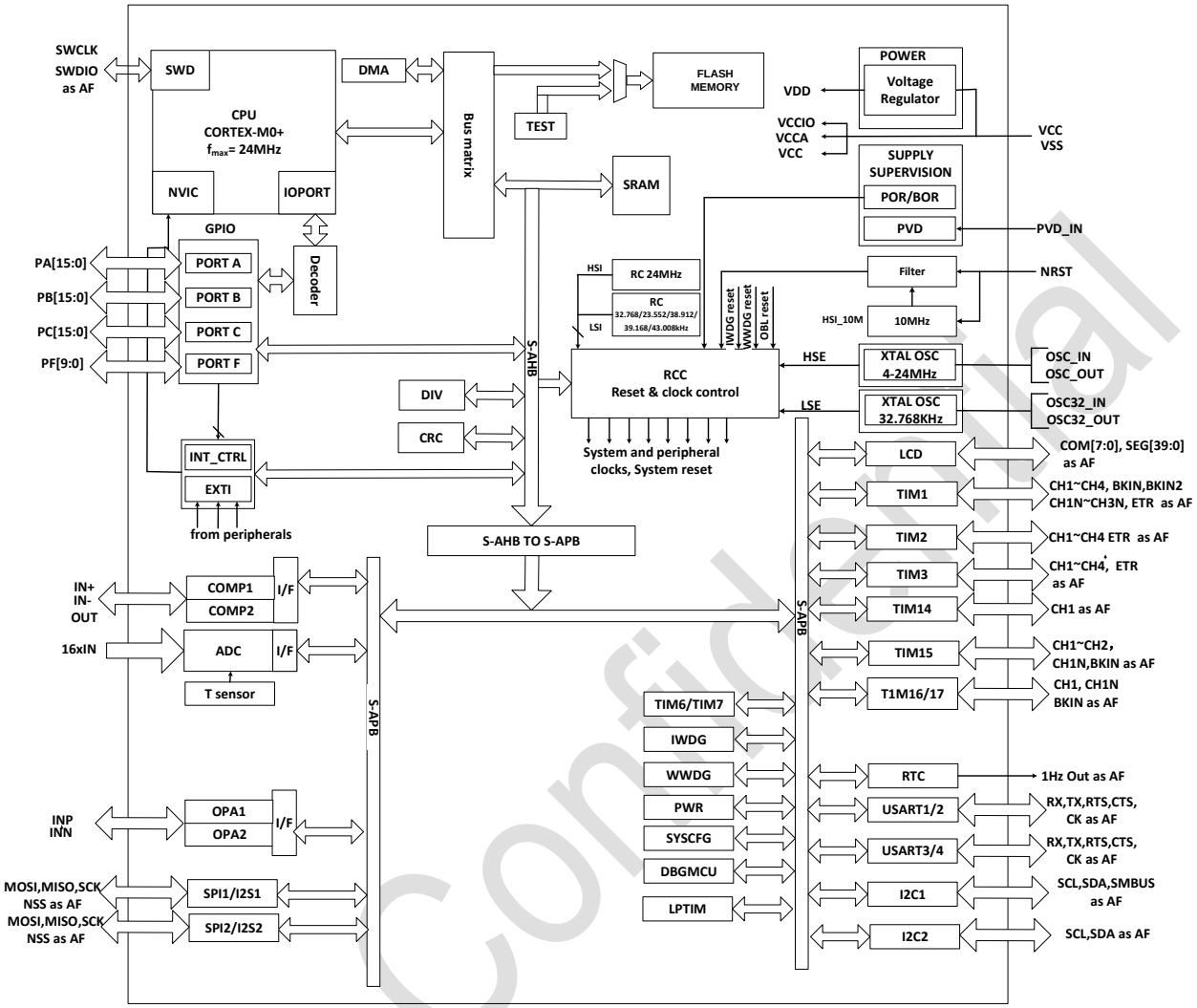


Figure 2-1 System architecture block

Puya Confidential

3. Memory and bus architecture

3.1. System architecture

The system consists of:

- Two Masters
  - Cortex-M0 + core
  - General-purpose DMA
- Three Slaves
  - Internal SRAM
  - Internal Flash memory
  - AHB with AHB-to-APB bridge that connects all the APB peripherals

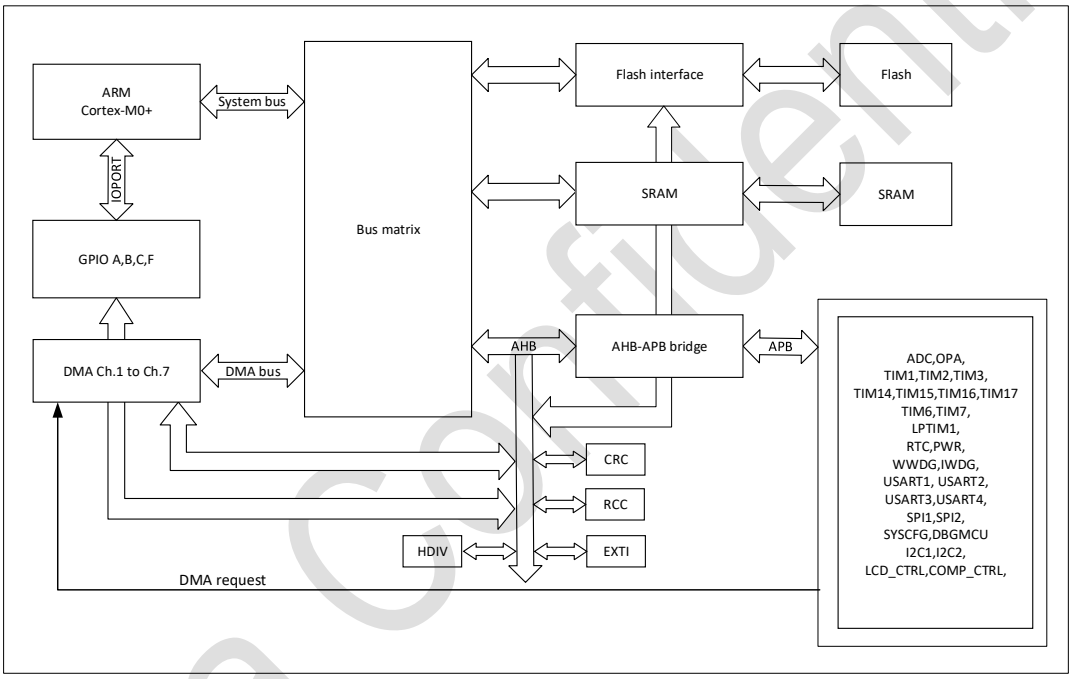


Figure 3-1 System architecture

- System bus

This bus connects the system bus of the Cortex®-M0+ core (peripheral bus) to a bus matrix that manages the arbitration between the core and the DMA.

- DMA bus

This bus connects the AHB master interface of the DMA to the bus matrix that manages the access of CPU and DMA to SRAM, Flash memory and AHB/APB peripherals.

- Bus matrix

The bus matrix manages the access arbitration between the core system bus and the DMA master bus. The arbitration uses a Round Robin algorithm. The bus matrix is composed of masters (CPU, DMA) and slaves (Flash memory interface, SRAM and AHB-to-APB bridge).

- AHB-to-APB Bridge (APB)

Puya Confidential

The AHB-to-APB bridge provides full synchronous connections between the AHB and the APB bus. Refer to Section 3.2:Memory organization for the address mapping of the peripherals connected to this bridge.

3.2. Memory organization

3.2.1. Memory organization

Program memory, data memory, registers and I/O ports are organized within the same linear 4 GB address space. The bytes are coded in memory in Little Endian format (the lowest numbered byte in a word is considered the word's least significant byte).

The addressable memory space is divided into eight main blocks, of 512 MB each.

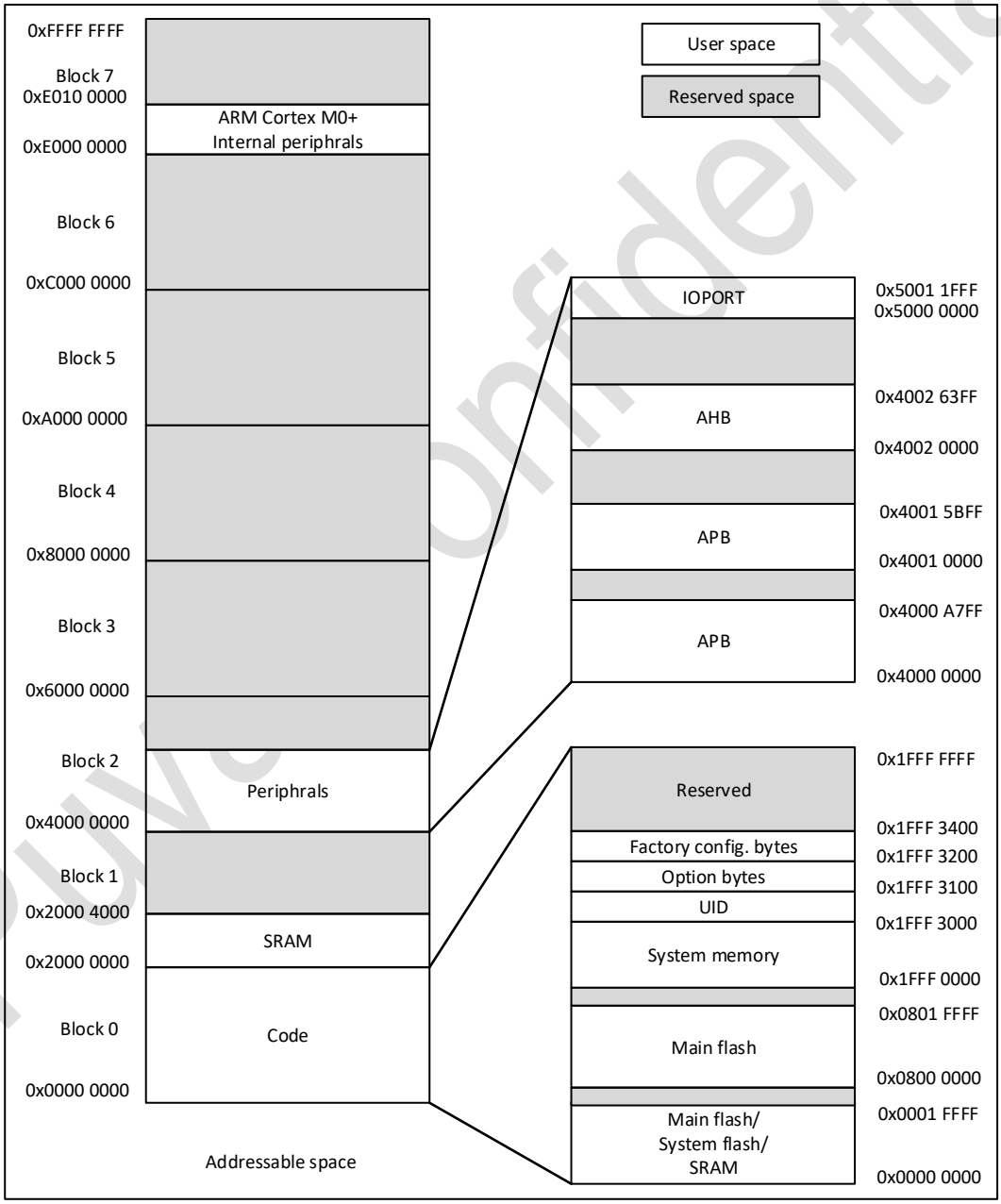


Figure 3-2 Memory map

Puya Confidential

Table 3-1 Memory boundary address

Type	Boundary address	Size	Memory area	Descriptions
SRAM	0x2000 4000-0x3FFF FFFF	-	Reserved(1)	-
	0x2000 0000-0x2000 3FFF	16 KB	SRAM	SRAM up to 16 KB
Code	0x1FFF 3400-0x1FFF FFFF	-	Reserved	-
	0x1FFF 3200-0x1FFF 33FF	256 Bytes	Factory configuration bytes	-
	0x1FFF 3100-0x1FFF 31FF	256 Bytes	Option bytes	See section 4.4 Flash option bytes
	0x1FFF 3000-0x1FFF 30FF	256 Bytes	UID	Refer to section 4.3 Unique device ID (UID)
	0x1FFF 0000-0x1FFF 2FFF	12 KB	System memory	-
	0x0802 0000-0x1FFE FFFF	-	Reserved	-
	0x0800 0000-0x0801 FFFF	128 KB	Main flash	-
	0x0002 0000-0x07FF FFFF	-	Reserved	-
	0x0000 0000-0x0001 FFFF	128 KB	Depending on the Boot configuration: 1) Main flash 2) System memory 3) SRAM	-

Table 3-2 Peripheral register address

Bus	Boundary address	Size	Peripheral
	0xE000 0000-0xE00F FFFF	-	M0+
IOPORT	0x5000 1800-0x5FFF FFFF	-	Reserved
	0x5000 1400-0x5000 17FF	1 KB	GPIOF
	0x5000 1000-0x5000 13FF	-	Reserved
	0x5000 0C00-0x5000 0FFF	-	Reserved
	0x5000 0800-0x5000 0BFF	1 KB	GPIOC
	0x5000 0400-0x5000 07FF	1 KB	GPIOB
	0x5000 0000-0x5000 03FF	1 KB	GPIOA
AHB	0x4002 4000-0x4FFF FFFF	-	Reserved
	0x4002 3C00-0x4002 3FFF	-	Reserved
	0x4002 3800-0x4002 3BFF	1 KB	DIV
	0x4002 3400-0x4002 37FF	-	Reserved
	0x4002 3000-0x4002 33FF	1 KB	CRC

Private Confidential

Bus	Boundary address	Size	Peripheral
	0x4002 2400-0x4002 2FFF	-	Reserved
	0x4002 2000-0x4002 23FF	1 KB	Flash
	0x4002 1C00-0x4002 1FFF	-	Reserved
	0x4002 1800-0x4002 1BFF	1 KB	EXTI
	0x4002 1400-0x4002 17FF	1 KB	Reserved
	0x4002 1000-0x4002 13FF	1 KB	RCC
	0x4002 0400-0x4002 0FFF	-	Reserved
	0x4002 0000-0x4002 03FF	1 KB	DMA
APB	0x4001 5C00-0x4001 FFFF	-	Reserved
	0x4001 5800 - 0x4001 5BFF	1 KB	DBG
	0x4001 4C00 - 0x4001 57FF	-	Reserved
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16
	0x4001 4000 - 0x4001 43FF	1 KB	TIM15
	0x4001 3C00 - 0x4001 3FFF	-	Reserved
	0x4001 3800 - 0x4001 3BFF	1 KB	USART1
	0x4001 3400 - 0x4001 37FF	1 KB	Reserved
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1/I2S1
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1
	0x4001 2800 - 0x4001 2BFF	-	Reserved
	0x4001 2400 - 0x4001 27FF	1 KB	ADC
	0x4001 0400 - 0x4001 23FF	-	Reserved
	0x4001 0300 - 0x4001 03FF	1 KB	OPA
	0x4001 0200 - 0x4001 02FF		COMP
	0x4001 0000 - 0x4001 01FF		SYSCFG
	0x4000 8000-0x4000 FFFF	-	Reserved
	0x4000 7C00 - 0x4000 7FFF	1 KB	LPTIM1
	0x4000 7400 - 0x4000 7BFF	-	Reserved
	0x4000 7000 - 0x4000 73FF	1 KB	PWR
	0x4000 5C00 - 0x4000 6FFF	-	Reserved
	0x4000 5800 - 0x4000 5BFF	1 KB	I^2C2
	0x4000 5400 - 0x4000 57FF	1 KB	I^2C1
	0x4000 5000 - 0x4000 53FF	-	Reserved

Bus	Boundary address	Size	Peripheral
	0x4000 4C00 - 0x4000 4FFF	1 KB	USART4
	0x4000 4800 - 0x4000 4BFF	1 KB	USART3
	0x4000 4400 - 0x4000 47FF	1 KB	USART2
	0x4000 3C00 - 0x4000 43FF	2 KB	Reserved
	0x4000 3800 - 0x4000 3BFF	1 KB	SPI2/I2S2
	0x4000 3400 - 0x4000 37FF	-	Reserved
	0x4000 3000 - 0x4000 33FF	1 KB	IWDG
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC
	0x4000 2400 - 0x4000 27FF	1 KB	LCD
	0x4000 2000 - 0x4000 23FF	1 KB	TIM14
	0x4000 1800 - 0x4000 1FFF	-	Reserved
	0x4000 1400 - 0x4000 17FF	1 KB	TIM7
	0x4000 1000 - 0x4000 13FF	1 KB	TIM6
	0x4000 0800 - 0x4000 0FFF	-	Reserved
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3
	0x4000 0000 - 0x4000 03FF	1 KB	TIM2

Puya Co

3.3. Embedded SRAM

Embedded up to 16 KB SRAM is accessible by byte, half-word (16 bits), or word (32 bits). Accessing memory outside the defined address range generates a HardFault interrupt.

3.4. Flash memory

The Flash memory is composed of two distinct physical areas:

- The Main Flash area:128 KB, consisting of application and user data, accessing memory outside the defined address range generates a HardFault interrupt.
- 13 KB of Information area:
  - Factory configuration:256*2 Bytes, used to store TS DATA, HSI Clock Re-trimming and other data
  - Option bytes:256 Bytes, used to store the configuration information
  - UID:256 Bytes, used to store the UID
  - System memory:12 KB, used to store the Boot loader.

Flash controller realizes instruction reading and data access based on AHB protocol, and it also realizes basic program/erase operations of Flash through registers.

3.5. Boot modes

At startup, the BOOT0 pin and the boot configuration bit nBOOT1 (stored in option bytes) are used to select one of the three boot options in the following table:

Table 3-3 Boot configuration

Boot mode configuration		Mode
nBOOT1 bit	BOOT0 pin	
X	0	Boot from Main flash
1	1	Boot from System memory
0	1	Boot from SRAM

After this startup delay has elapsed, the CPU fetches the top-of-stack value from address 0x0000 0000, then starts code execution from the boot memory at 0x0000 0004. Depending on the selected boot mode, Main flash, system memory or SRAM is accessible as follows:

- Boot from Main flash:The Main flash memory is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x0800 0000). In other words, the Flash memory contents can be accessed starting from address 0x0000 0000 or 0x0800 0000.
- Boot from System memory:The system memory is aliased in the boot memory space (0x0000 0000), but it is still accessible from its original memory space (0x1FFF 0000).
- Boot from SRAM:the SRAM is aliased in the boot memory space (0x0000 0000), but it is still accessible from its original memory space (0x2000 0000).

Puya Confidential

3.5.1. Memory physical mapping

Once the boot mode is selected, the application software can modify the memory accessible in the code area. This modification is performed by programming the MEM_MODE bits in the SYS-CFG_CFGR1 (see the SYSCFG chapter for details).

3.5.2. Embedded boot loader

The embedded boot loader is located in the System memory, programmed during production. It is used to reprogram the Flash memory with one of the available serial interfaces:

- USART1, PA9/PA10
- PA11/PA12

Puya Confidential

4. Embedded Flash memory

4.1. Flash main features

- Main flash block:up to 128 KB
- Information block:13 KB
- Page size:256 Bytes
- Sector size:8 KB

Flash memory interface features:

- Flash program and erase, and the smallest unit is page
- Only supports writing 0 but not writing 1. The erased value is 1
- Programming operations of option bytes
- Read protection
- Write protection

4.2. Flash memory functional description

4.2.1. Flash memory organization

The Flash memory is organized as 64-bit wide memory cells that can be used for storing both code and data constants. The Page size is 256 Bytes and the Sector size is 8 KB.

Functionally, Flash memory is divided into Main flash and Information flash. The former has a maximum capacity of 128 KB, and the latter has a capacity of 13 KB.

Table 4-1 Flash structure and boundary address

Block	Sector	Page	Base address	Size
Main flash	Sector 0	Page 0-31	0x0800 0000-0x0800 1FFF	8 KB
	Sector 1	Page 32-63	0x0800 2000-0x0800 3FFF	8 KB
	Sector 2	Page 64-95	0x0800 4000-0x0800 5FFF	8 KB
	…	…	…	…
	Sector 14	Page 448-479	0x0801 C000-0x0801 DFFF	8 KB
	Sector 15	Page 480-511	0x0801 E000-0x0801 FFFF	8 KB
System flash	Info	Page 0-46	0x1FFF 0000-0x1FFF 2EFF	11.75 KB
Reserved		Page 47	0x1FFF 2F00-0x1FFF 2FFF	256 Bytes
UID		Page 48	0x1FFF 3000-0x1FFF 30FF	256 Bytes
Option bytes		Page 49	0x1FFF 3100-0x1FFF 31FF	256 Bytes
Factory configuration 0		Page 50	0x1FFF 3200-0x1FFF 32FF	256 Bytes
Factory configuration 1		Page 51	0x1FFF 3300-0x1FFF 33FF	256 Bytes

Puya Confidential

4.2.2. Flash read operation and access latency

The embedded Flash module can be addressed directly, as a common memory space. Through the special read control timing, the contents of the Flash memory can be read. The instruction fetch and data access are both done through the AHB bus.

4.2.3. Flash program and erase operations

The Flash memory can be programmed using in-circuit programming (ICP) or in-application programming (IAP).

ICP:used to update the entire contents of the Flash memory, using the SWD protocol or the boot loader to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

IAP:using any communication interface supported by the microcontroller to download programming data into Flash memory. IAP allows the user to re-program the Flash memory while the application is running. Nevertheless, part of the application has to have been previously programmed in the Flash memory using ICP.

If a reset occurs during Flash program and erase operations, the contents of the Flash memory are not protected.

During a program/erase operation to the Flash memory, any attempt to read the Flash memory will stall the bus. The read operation will proceed correctly once the program/erase operation has completed. This means that code or data fetches cannot be made while a program/erase operation is ongoing.

For program and erase operations on the Flash memory (write/erase), the HSI must be ON.

4.2.3.1. Unlocking the Flash memory

After reset, the Flash memory is protected against unwanted (caused by electrical interference) write or erase operations. The FLASH_CR register is not accessible in write mode, except for the OBL_LAUNCH bit, used to reload the option bits. An unlocking sequence should be written to the FLASH_KEYR register to open the access to the FLASH_CR register.

This is done in the following steps:

Step 1:write KEY1=0x4567 0123 to FLASH_KEYR register

Step 2:write KEY2=0xCDEF 89AB to FLASH_KEYR register

Any wrong sequence locks up the FLASH_CR register until the next reset. In the case of a wrong key sequence, a bus error is detected, and a HardFault interrupt is generated. This is done after the first write cycle if KEY1 does not match, or during the second write cycle if KEY1 has been correctly written but KEY2 does not match.

The FLASH_CR register can be locked again by user software by writing the LOCK bit in the FLASH_CR register to 1.

Puya Confidential

In addition, the FLASH_CR register cannot be written when the BSY bit of the FLASH_SR register is set. Meanwhile, any attempt to write FLASH_CR register will cause the AHB bus to stall until the BSY bit is cleared.

4.2.3.2. Flash memory programming

Flash memory writes the entire page in word (32-bit) units (HardFault will be generated if writes half-word or byte) at a time. The program operation is started when the CPU writes a half-word into a main Flash memory address with the PG bit of the FLASH_CR register set. Any attempt to write data that are not full word long will result in a HardFault interrupt.

If the address main Flash memory location is write-protected by the FLASH_WRPR register, the program operation is skipped and the WRPERR bit in the FLASH_SR register will be set. The end of the program operation is indicated by the EOP bit in the FLASH_SR register.

The Flash memory programming sequence is as follows:

1. Check that no main Flash memory operation is ongoing by checking the BSY bit in the FLASH_SR register.
2. If there is no flash erase or program operation in progress, the software reads out 64 words of the Page (this step is performed if the Page already has data stored, otherwise this step is skipped)
3. Write KEY1 and KEY2 to the FLASH_KEYR register to unlock the protection of the FLASH_CR register
4. Set the PG bit and EOPIE bit in the FLASH_CR register
5. Program the 1st to 63rd words to the target address (only 32-bits programs are accepted)
6. Set the PGSTRT bit in the FLASH_CR register
7. Write the 64th Word
8. Wait until the BSY bit is reset in the FLASH_SR register
9. Check that EOP flag is set in the FLASH_SR register (meaning that the programing operation has succeed), and then clear it by software
10. Clear the PG bit if there is no more programming request anymore

When the step 7 is carried out, the program operation is automatically started and the BSY bit is set simultaneously.

4.2.3.3. Flash memory erase

The Flash memory erase operation can be performed at page level, sector level (sector erase) or on the whole Flash memory (mass erase).

Page erase

When a page is write-protected, it will not be erased and the WRPERR bit is set. To erase a page, follow the procedure below:

1. Check that no main Flash memory operation is ongoing by checking the BSY bit in the FLASH_SR register.
2. Write KEY1 and KEY2 to the FLASH_KEYR register to unlock the protection of the FLASH_CR register

Puya Confidential

3. Set the PER bit and EOPIE bit in the FLASH_CR register
4. Write arbitrary data to this Page (must be 32-bit data)
5. Wait for the BSY bit to be cleared.
6. Check that EOP flag bit is set
7. Clear EOP flag

Mass erase

Mass erase is used to erase the whole Main flash memory. Further, when WRP is enabled, the mass erase function is invalidated, no mass erase operation is generated, and the WRPERR bit is set.

To perform a Mass erase, follow the procedure below:

1. Check the BSY bit to confirm if there are ongoing Flash operations
2. Write KEY1 and KEY2 to the FLASH_KEYR register to unlock the protection of the FLASH_CR register
3. Set the MER bit and EOPIE bit in the FLASH_CR register
4. Write any data (32-bit data) to any Main flash space
5. Wait for the BSY bit to be cleared.
6. Check that EOP flag bit is set
7. Clear EOP flag

Sector erase

Sector erase is used to erase 8 KB of Main flash. In addition, when a sector is protected by WRP, it will not be erased, and the WRPERR bit is set at this time.

To perform a Sector erase, follow the procedure below:

- Check the BSY bit to confirm if there are ongoing Flash operations
- Write KEY1 and KEY2 to the FLASH_KEYR register to unlock the protection of the FLASH_CR register
- Set the SER bit and EOPIE bit in the FLASH_CR register
- Write arbitrary data to this Sector
- Wait for the BSY bit to be cleared.
- Check that EOP flag bit is set
- Clear EOP flag

4.2.3.4. Program and erase time configuration

The time of Flash program and erase needs to be strictly controlled, otherwise the operation will fail. Power-on by default, the hardware design sets the time parameters of program and erase operations to the parameters with HSI of 24 MHz. When the HSI output frequency is changed, the Flash program and erase time control registers need to be configured according to the following table.

Table 4-2 Program and erase time configuration

Register	8 MHz	24 MHz
TS0	0x3C	0xB4
TS1	0x90	0x1B0
TS2P	0x3C	0xB4

Puya Confidential

Register	8 MHz	24 MHz
TPS3	0x240	0x6C0
TS3	0x3C	0xB4
PERTPE	0x6D60	0x14820
SMERTPE	0x6D60	0x14820
PRGTPE	0x1F40	0x5DC0
PRETPE	0x640	0x12C0

4.3. Unique device ID (UID)

Typical application scenarios of unique identification code:

■ Used as a serial number

■ When programming internal flash memory, use it as a key or encryption primitive to improve code security

■ To activate secure boot processes, etc.

The UID provides a reference number that is unique to any device.

The user can never change these bits. Unique identifiers can also be read in different ways, such as byte/half word/word, and then concatenated using custom algorithms.

The UID of this product is shown in the following table:

Base address:0x1FFF 3000

Table 4-3 Unique device ID (UID)

Address offset	Descriptions	UID Bits							
		7	6	5	4	3	2	1	0
0	Lot Numer	Lot Number ASCII encoded							
1	Lot Numer	Lot Number ASCII encoded							
2	Lot Numer	Lot Number ASCII encoded							
3	Lot Numer	Lot Number ASCII encoded							
4	Wafer Number	Wafer Number							
5	Lot Numer	Lot Number ASCII encoded							
6	Lot Numer	Lot Number ASCII encoded							
7	Lot Numer	Lot Number ASCII encoded							
8	Internal encoded	Internal encoded							
9	Low Y-coordinate	Low Y-coordinate							
10	Low X-coordinate	Low X-coordinate							
11	High X,Y-coordinate	High Y-coordinate				High X-coordinate			
12	CP pass ID	CP pass ID							
13	CRC8	CRC8 calibration value							
14	Internal encoded	Internal encoded							
15	Internal encoded	Internal encoded							

Puya Confidential

4.4. Flash option bytes

4.4.1. Flash option bytes

Part of the information area of flash is used as option bytes, which are configured by the end user depending on the application requirements. As a configuration example, the watchdog may be selected in hardware or software mode.

For data security, option bytes are stored separately in positive code and negative code form.

Table 4-4 Option byte format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Complemented option byte 1								Complemented option byte 0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Option byte 1								Option byte 0							

The software can read the option bytes from these flash memory locations or from their corresponding option registers referenced in the table.

- Flash option register (FLASH_OPTR)
- Flash BORCR address register (FLASH_BORCR)
- Flash WRP area address register (FLASH_WRPR)

Table 4-5 Option byte organization

Word address	Description
0x1FFF 3100	Flash user option bytes and its complement
0x1FFF 3108	Option bytes for BOR control and its complement
0x1FFF 3110	Reserved
0x1FFF 3118	Option bytes for Flash WRP address and its complement
0x1FFF 3120	Reserved
0x1FFF 3128	Reserved
…	Reserved
…	Reserved
…	Reserved
0x1FFF 31F8	Reserved

Puya Confidential

4.4.1.1. Option bytes for Flash user options and its complement

Flash memory address:0x1FFF 3100

Production value:0x2755 D8AA

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option byte area of the Flash information memory and written to the corresponding option bit of the register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
~IWDG_STOP	~ nBOOT1	~ NRST_MODE	~ WWDG_SW	~ IWDG_SW	Res	Res	Res	~RDP[7:0]							
R	R	R	R	R	-	-	-	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG_STOP	nBOOT1	NRST_MODE	WWDG_SW	IWDG_SW	Res	Res	Res	RDP[7:0]							
R	R	R	R	R	-	-	-	R	R	R	R	R	R	R	R

Bit	Name	R/W	Function
31	~IWDG_STOP	R	Complemented code of IWDG_STOP
30	~ nBOOT1	R	Complemented code of nBOOT1
29	~ NRST_MODE	R	Complemented code of NRST_MODE
28	~ WWDG_SW	R	Complemented code of WWDG_SW
27	~ IWDG_SW	R	Complemented code of IWDG_SW
26:24	Reserved	-	Reserved
23:16	~ RDP	R	Complemented code of RDP
15	IWDG_STOP	R	Set the running state of IWDG timer in Stop mode 0:Freeze timer 1:Normal operation
14	nBOOT1	R	Together with the BOOT PIN, select the boot mode
13	NRST_MODE	R	0:Reset input only 1:GPIO function
12	WWDG_SW	R	0:Hardware window watchdog 1:Software window watchdog
11	IWDG_SW	R	0:Hardware window watchdog 1:Software window watchdog
10:8	Reserved	-	Reserved
7:0	RDP	R	0xAA:level 0, read protection is invalid Non-0xAA:level 1, read protection valid

4.4.1.2. Option bytes for BOR control and its complement

Flash memory address:0x1FFF 3108

Production value:0x1FFF E000

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option byte area of the Flash information memory and written to the corresponding option bit of the register.

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res			Res	Res	Res	Res	Res	Res	Res	~BOR_EN	Res	Res	Res	Res	Res
-			-							R	-				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			Res	Res	Res	Res	Res	Res	Res	BOR_EN	Res	Res	Res	Res	Res
-			-							R	-				

Bit	Name	R/W	Function
31:22	Reserved	-	Reserved
21	~BOR_EN	R	Complemented code of BOR_EN
20:6	Reserved	-	Reserved
5	BOR_EN	R	BOR enabled 0:BOR disabled 1:BOR enabled
4:0	Reserved	-	Reserved

4.4.1.3. Option bytes for Flash WRP address and its complement

Flash memory address:0x1FFF 3118

Production value:0x0000 FFFF

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option byte area of the Flash information memory and written to the corresponding option bit of the register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
~WRP[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Function
31:16	~ WRP	R	Complemented code of WRP
15:0	WRP	R	0:sector [y] is protected 1:sector [y] unprotected y=0 to15

4.4.2. Flash option byte programming

After reset, the bits associated with the option bytes in the FLASH_CR register are write-protected. The OPTLOCK bit in the FLASH_CR register must be cleared before relevant operations can be performed on the option byte.

The following steps are used to unlock the register:

1. Unlock the FLASH_CR with the LOCK clearing sequence

Puya Confidential

2. Write OPTKEY1 = 0x0819 2A3B to the FLASH_OPTKEYR register
3. Write OPTKEY2 = 0x4C5D 6E7F to the FLASH_OPTKEYR register

Any wrong sequence will lock up the FLASH_CR register until the next system reset. In the case of a wrong key sequence, a bus error is detected, and a HardFault interrupt is generated.

The user options (the option byte of information flash) can be protected against unwanted erase/program operations by setting the OPTLOCK bit in the FLASH_CR register by software.

If the software sets the Lock bit, the OPTLOCK bit is also automatically set.

4.4.2.1. Modifying user options

The option bytes are programmed differently from a Main memory user address. To modify the option bytes, the following steps are needed:

- Clear the OPTLOCK bit using the previously described steps
- Check the BSY bit to confirm that there are no ongoing Flash operations
- Write the desired value (1 to 3 words) to the option byte register FLASH_OPTR / FLASH_BORCR / FLASH_WRPR
- Set OPTSTRT bit
- Write any 32-bit data to the Main flash 0x4002 2080 address (trigger a formal write operation)
- Wait for the BSY bit to be cleared.
- Waiting for EOP to pull higher
- EOP is cleared by software

For any changes to the option byte, the hardware will first erase the entire page corresponding to the option byte, and then write it to the option byte with the values of the FLASH_OPTR, FLASH_BORCR or FLASH_WRPR registers. The hardware automatically calculates the corresponding complement code and writes the calculated value to the corresponding area of the option byte.

4.4.2.2. Reload option bytes

After the BSY bit is cleared, all new option bytes are written to the Flash information memory, but are not applied to the system. A read operation on the option byte register still returns the value in the last loaded option byte. Only when they (new values) are loaded do they work on the system.

The loading of option bytes occurs in the following two cases:

1. When the OBL_LAUNCH bit in the FLASH_CR register is set
2. After a power reset (POR, BOR)

The operation performed by load option byte is to read the option byte in the Information memory area, and then store the read data in the internal option register (FLASH_OPTR, FLASH_BORCR, and FLASH_WRPR). These internal registers configure the system and can be read by software. Setting the OBL_LAUNCH bit generates a reset, so that the loading of option bytes can be performed under the system reset.

Each option bit has also its complement in the same double word. During option loading, a verification of the option bit and its complement allows to check the loading has correctly taken place.

If the word and its complement are matching, the option byte is copied into the option register.

Puya Confidential

If the word and its complement are not matching, the OPTVERR status bit of the FLASH_SR register is set. Mismatch value are forced into the option registers:

- For User option
- The BOR_EN bit is written as 0 (BOR disabled)
- The NRST_MODE bit is written as 0 (reset input only)
- The RDP bit is written as 0xff (i.e. level 1)
- The rest of the mismatched values are written as 1
- For WRP option, the value of mismatch is because the default value is unprotected

Upon system reset, the option bytes are copied into the following option registers that can be read and written by software:

- FLASH_OPTR
- FLASH_BORCR
- FLASH_WRPR

These registers are also used to modify option bytes. If these registers are not modified by user, they reflect the options states of the system.

4.5. Flash configuration bytes

A partial section (one page in total) of the Flash Information area in the device is used as the factory configuration 0.

Page 0 stores information for software to read (only the code, not its complement is stored):

- HSI frequency selection value and corresponding Trimming value
- Configuration parameter values of erase and write time corresponding to different frequencies of HSI

Table 4-6 Factory config.bytes0 configuration

Page	Word	Address	Contents
0	0	0x1FFF 3200	Reserved
	1	0x1FFF 3208	Stores HSI 8 MHz frequency selection control and corresponding Trimming values
	2	0x1FFF 3210	Reserved
	3	0x1FFF 3218	Reserved
	4	0x1FFF 3220	Stores HSI 24 MHz frequency selection control and corresponding Trimming values
	5	0x1FFF 3228	TS_CAL1, calibration value of 30 ℃ temperature sensor
	6	0x1FFF 3230	TS_CAL2, calibration value of 105 ℃ temperature sensor
	7	0x1FFF 3238	Reserved
	8	0x1FFF 3240	Reserved
	9	0x1FFF 3248	Reserved
	10	0x1FFF 3250	Reserved
	11	0x1FFF 3258	Reserved
	12	0x1FFF 3260	Store the configuration values of the corresponding FLASH_TS0 and FLASH_TS1 registers at HSI 8 MHz frequency

Puya Confidential

	13	0x1FFF 3268	Store the configuration values of the corresponding FLASH_TS2P and FLASH_TPS3 registers at HSI 8 MHz frequency
	14	0x1FFF 3270	Stores the configuration value of the corresponding FLASH_PERTPE register at the frequency of HSI 8 MHz
	15	0x1FFF 3278	Stores the configuration value of the corresponding FLASH_SMERTPE register at the frequency of HSI 8 MHz
	16	0x1FFF 3280	Store the configuration values of the corresponding FLASH_PRGTPE and FLASH_PRETPE registers at HSI 8 MHz frequency
	17	0x1FFF 3288	Reserved
	18	0x1FFF 3290	Reserved
	19	0x1FFF 3298	Reserved
	20	0x1FFF 32A0	Reserved
	21	0x1FFF 32A8	Reserved
	22	0x1FFF 32B0	Reserved
	23	0x1FFF 32B8	Reserved
	24	0x1FFF 32C0	Reserved
	25	0x1FFF 32C8	Reserved
	26	0x1FFF 32D0	Reserved
	27	0x1FFF 32D8	Store the configuration values of the corresponding FLASH_TS0 and FLASH_TS1 registers at HSI 24 MHz frequency
	28	0x1FFF 32E0	Store the configuration values of corresponding FLASH_TS2P and FLASH_TPS3 registers at HSI 24 MHz frequency
	29	0x1FFF 32E8	Stores the configuration value of the corresponding FLASH_PERTPE register at the frequency of HSI 24 MHz
	30	0x1FFF 32F0	Stores the configuration value of the corresponding FLASH_SMERTPE register at the frequency of HSI 24 MHz
	31	0x1FFF 32F8	Store the configuration values of the corresponding FLASH_PRGTPE and FLASH_PRETPE registers at HSI 24 MHz frequency

4.5.1. HSI_TRIMMING_FOR_USER

Address:0x1FFF 3200 to 0x1FFF 3220

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HSI_FS[2:0]			HSI_TRIM[12:0]												
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to read the data from this address and then write it to HSI_FS [2:0] and HSI_TRIM [12:0] corresponding to the RCC_ICSCR register to change the HSI frequency.

Puya Confidential

4.5.2. Calibration value of temperature sensor

Address:0x1FFF 3228 (30 ℃), 0x1FFF 3230 (105 ℃)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	TSCAL[11:0]											
-	-	-	-	R											

The software needs to read data from this address.

4.5.3. HSI_8M/24M_EPPARA0

Address:0x1FFF 3260 (8 MHz),0x1FFF 32D8 (24 MHz)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	TS1[8:0]								
-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TS3[7:0]								TS0 [7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to read data from the corresponding address according to the required HSI clock frequency, and then write it to the FLASH_S0, FLASH_TS1, and FLASH_TS3 registers to realize the configuration of the erasing and writing time required for the corresponding HSI frequency.

4.5.4. HSI_8M/24M_EPPARA1

Address:0x1FFF 3268 (8 MHz), 0x1FFF 32E0 (24 MHz)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	TPS3 [10:0]										
-	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	TS2P [7:0]							
-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R

The software needs to read data from the corresponding address according to the required HSI clock frequency, and then write it to the FLASH_TS2P and FLASH_TPS3 registers to realize the configuration of the erasing and writing time required for the corresponding HSI frequency.

4.5.5. HSI_8M/24M_EPPARA2

Address: 0x1FFF 3270 (8 MHz),0x1FFF 32E8 (24 MHz)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PERTPE [16]
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERTPE [15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to choose to read data from the corresponding address according to the HSI clock frequency that needs to be set, and then write it into the FLASH_PERTPE register to realize the configuration of the erasing and writing time required for the corresponding HSI frequency.

4.5.6. HSI_8M/24M_EPPARA3

Address:0x1FFF 3278 (8 MHz),0x1FFF 32F0 (24 MHz)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SMERTPE[16]
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMERTPE [15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to choose to read data from the corresponding address according to the HSI clock frequency that needs to be set, and then write it into the FLASH_SMERTPE register to realize the configuration of the erasing and writing time required for the corresponding HSI frequency.

4.5.7. HSI_8M/24M_EPPARA4

Address:0x1FFF 3280 (8 MHz), 0x1FFF 32F8 (24 MHz)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	PRETPE [11:0]										
-	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRGTPE [15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to choose to read data from the corresponding address according to the required HSI clock frequency, and then write it to the FLASH_PRGTPE and FLASH_PRETPE registers to realize the configuration of the erasing and writing time required for the corresponding HSI frequency.

4.6. Flash protection

The protection of Main flash area includes the following mechanisms:

- Read protection (RDP) blocks external access.
- Write protection (WRP) prevents unwanted write operations (caused by confusion of program). The size of write protection is designed to be 8 KB.
- Option byte write protection is a special design for unlock.

Puya Confidential

4.6.1. Flash read protection

The read protection is activated by setting the RDP option byte and then, by applying a system reset (POR/BOR or OBL reset) to reload the new RDP option byte. RDP protects Main flash memory, option bytes, SRAM.

If the read protection is set while the debugger is still connected through SWD, apply power reset instead of system reset.

The Flash memory is protected when the RDP option byte and its complement contain the pair of values.

Table 4-7 Flash read protection status

RDP byte value	RDP completed byte value	Read protection level
0xAA	0x55	Level 0
Any value of the combination except (0xAA and 0x55)		Level 1

Regardless of any protection level, System memory can only be read, and cannot Program and Erase.

- **Level 0:Unprotected**

Read, program and erase operations within the Main flash memory area are possible. The option bytes are also accessible by all operations.

- **Level 1:Read protection**

Level 1 read protection is set when the RDP byte and the RDP complemented byte contain any value combinations other than (0xAA, 0x55). Level 1 is the default protection level.

- User mode:Code executing in user mode (boot from Main flash) can access Main flash memory and option bytes with all operations.
- Debug, boot from SRAM and boot from System memory modes:In debug mode or when code boots from SRAM or System memory, the Main flash memory is inaccessible. In these modes, a read or write access to the Flash memory generates a bus error and a HardFault interrupt.

When it is already at Level 1 (any number other than 0xAA), if you want to modify it to Level 0 (write 0xAA), the hardware will perform a mass erase operation on the Main flash.

Table 4-8 Relationship of access status to protection level and execution mode

Area	Read protection level	Boot from Main flash (CPU)			Debug/Boot from SRAM/ Boot from System memory			DMA		
		Users execution								
		Read	Write	Erase	Read	Write	Erase	Read	Write	Erase
System memory	0/1	Yes	No	No	Yes	No	No	No	No	No
		Yes	No	No	Yes	No	No	No	No	No
Option byte area	0/1	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
		Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
	0/1	Yes	No	No	Yes	No	No	No	No	No

Puya Confidential

Area	Read protection level	Boot from Main flash (CPU)			Debug/Boot from SRAM/ Boot from System memory			DMA		
		Users execution								
		Read	Write	Erase	Read	Write	Erase	Read	Write	Erase
Factory configura-tion 0		Yes	No	No	Yes	No	No	No	No	No
UID	0/1	Yes	No	No	Yes	No	No	No	No	No
		Yes	No	No	Yes	No	No	No	No	No

Notes:

1. Any modification from Level 1 to Level 0 will trigger a mass erase of the Main flash by the hardware.
2. There are two cases for executing programs from SRAM or System memory:one is Boot from SRAM or System memory; The other is to Boot from another memory and then the program jumps to SRAM or System memory.

4.6.2. Flash write protection

Flash can be set to write-protected in response to unwanted write operations. Define that each WRP register controls a write protection (WRP) area with a size of 8 KB, i.e. 1 sector size. See the description of the WRP register for details.

When the WRP area is activated, no erase or write operation is allowed. Accordingly, even if only one area is set to write protection, the mass erase function does not function.

In addition, if an erase or program operation is attempted on a write-protected area, the write-protected error (WRPERR) of the FLASH_SR register is set.

Note:Write protection only works for Main flash, not for System memory.

4.6.3. Option byte write protection

By default, option bytes are readable and write-protected. In order to obtain a program or erase access to the option byte, the correct sequence needs to be written to the OPTKEYR register.

4.7. Flash interrupt

Table 4-9 Flash interrupt request

Interrupt event	Event flag	Time flag/interrupt clearing	Control bit enable
End of operation	EOP	Write EOP=1	EOPIE
Write protection	WRPERR	Write WRPERR=1	ERRIE

Note:The following events do not have separate interrupt flags but generate a HardFault:

1. Sequence error in unlocking FLASH_CR register of Flash memory
2. Write sequence wrong to unlock Flash option byte
3. Flash program fails to align 32-bit data
4. Flash erase (including page erase, sector erase, and mass erase) operations does not perform 32-bit data alignment
5. Write operation to option byte register fails to align 32-bit data

Puya Confidential

4.8. Flash registers

4.8.1. Flash key register (Flash_KEYR)

Address offset:0x08

Reset value:0x0000 0000

All register bits are Write-Only and read out returns 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY [31:16]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEY [15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:0	KEY [31:0]	W	0	The following values must be written consecutively to unlock the FLASH_CR register and allow Flash's program/erase operation KEY1:0x4567 0123 KEY2:0xCDEF 89AB

4.8.2. Flash option key register (Flash_OPTKEYR)

Address offset:0x0C

Reset value:0x0000 0000

All register bits are Write-Only and read out returns 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY [31:16]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPTKEY [15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:0	OPTKEY [31:0]	W	0	The following values must be written consecutively to unlock the flash option register and enable the program/erase operation of the option byte KEY1:0x0819 2A3B KEY2:0x4C5D 6E7F

4.8.3. Flash status register (Flash_SR)

Address offset:0x10

Puya Confidential

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BSY
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPTV ERR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WRP ERR	Res	Res	Res	EOP
RC_W1	-	-	-	-	-	-	-	-	-	-	RC_W1	-	-	-	RC_W1

Bit	Name	R/W	Reset Value	Function
31:17	Reserved	-	-	Reserved
16	BSY	R	0	Busy bit This bit indicates that the operation of the flash is in progress. This bit is set by the hardware at the beginning of the flash operation, and is cleared by the hardware when the operation is completed or an error occurs.
15	OPTVERR	RC_W1	0	Option and trimming bit loading error When the option and trimming bits and their complement do not match, the hardware sets this bit. If loading mismatched option bytes, it will be forced to safe values. This bit is cleared by writing 1.
14:5	Reserved	-	-	Reserved
4	WRPERR	RC_W1	0	Write protection error When the address to be programmed/erase is in the write-protected Flash area (WRP), the hardware sets this bit. This bit is cleared by writing 1.
3:1	Reserved	-	-	Reserved
0	EOP	RC_W1	0	When the program/erase operation of Flash is successfully completed, the hardware is set. This bit is set only if the EOPIE bit of the FLASH_CR register is enabled. This bit is cleared by writing 1.

4.8.4. Flash control register (FLASH_CR)

Address offset:0x14

Reset value:0xC000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPT LOCK	Res		OBL_LAUNCH	Res	ERR IE	EOP IE	Res				PGSTRT	Res	OPT STRT	Res
RS	RS	-		RC_W1		RW	RW	-				RW	-	RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Puya Confidential

Res	Res	Res	SER	Res	Res	Res	MER	PER	PG
-	-	-	RW	-	-	-	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31	Lock	RS	1	FLASH_CR Lock bit. The software can only set this bit. When set, the FLASH_CR register is locked. When the unlocking timing is successfully given, this bit is cleared by hardware, unlocking the FLASH_CR register. [This bit is set by software after the program/erase operation is completed] When an unsuccessful unlock timing is given, the bit remains set until the next system reset.
30	OPTLOCK	RS	1	Option byte Lock bit. The software can only set this bit. After reset, the bits associated with the option bytes in the FLASH_CR register are locked. When the unlocking timing is successfully given, this bit is cleared by hardware, unlocking the FLASH_CR register. [This bit is set by software after the program/erase operation is completed] When an unsuccessful unlock timing is given, the bit remains set until the next system reset.
29:28	Reserved	-	-	Reserved
27	OBL_LAUNCH	RC_W1	0	Force option byte reloading. When set, this bit forces the system to reload the option byte. This bit is cleared by hardware only when the option byte load is completed. If the OPTLOCK bit is set, the bit cannot be written. 0:Option byte reload complete 1:Generate an option byte reload request, and the system generates a reset to reload the option byte.
26	Reserved	-	-	Reserved
25	ERRIE	RW	0	When the WRPERR bit of the FLASH_SR register is set, if the bit is enabled, an interrupt request is generated. 0:No interrupt occurrence 1:An interrupt occurs
24	EOPIE	RW	0	End of operation interrupt enable When the EOP bit of the FLASH_SR register is set, if the bit is enabled, an interrupt request is generated.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:EOP interrupt disabled 1:EOP interrupt enabled
23:20	Reserved	-	-	Reserved
19	PGSTRT	RW	0	The start bit of the program operation for the Main flash memory. This bit starts the program operation of the Main flash memory when set. This bit is cleared by hardware when the BSY bit is cleared in FLASH_SR.
18	Reserved	-	-	Reserved
17	OPTSTRT	RW	0	Flash option byte modification start bit This bit triggers the modification of the option byte. This bit is set by software and is cleared when the BSY bit is cleared in FLASH_SR. Note:When the flash option bytes are modified, the hardware automatically performs an erase operation on the entire 256 Bytes page, and then performs a program operation, which also includes automatic write of complemented code.
16:12	Reserved	-	-	Reserved
11	SER	RW	0	8 KB Sector erase operation 0:Sector erase operation for Flash is not selected 1:Sector erase operation for Flash selected Notes: Sector erase does not work on Flash information memory. Sector erase does not work for WRP areas.
10:3	Reserved	-	-	Reserved
2	MER	RW	0	Mass erase operation 0:Mass erase operation for Flash is not selected 1:Mass erase operation for Flash selected Notes: Mass erase will not work on Flash information memory. Mass erase does not work when there is a WRP setting
1	PER	RW	0	Page erase operation 0:Page erase operation for Flash is not selected 1:Page erase operation for Flash selected
0	PG	RW	0	Program operation 0:program operation of Flash is not selected 1:program operation of Flash selected

Puya Confidential

4.8.5. Flash option register (Flash_OPTR)

Address offset:0x20

Reset value:0x0000D8AA. After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option byte area of the flash information memory and written to the corresponding option bit of the register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG_STOP	nBOOT1	NRST_MODE	WWDG_SW	IWDG_SW	Res			RDP[7:0]							
RW	RW	RW	RW	RW	-			RW							

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	IWDG_STOP	RW	1	Set the running state of IWDG timer in Stop mode 0:Freeze timer 1:Normal operation
14	nBOOT1	RW	1	Together with the BOOT PIN, select the boot mode
13	NRST_MODE	RW	0	0:Reset input only 1:GPIO:GPIO function
12	WWDG_SW	RW	1	0:Hardware window watchdog 1:Software window watchdog
11	IWDG_SW	RW	1	0:Hardware window watchdog 1:Software window watchdog
10:8	Reserved	-	-	Reserved
7:0	RDP	RW	0xAA	0xAA:level 0, read protection is invalid Non-0xAA:level 1, read protection valid

4.8.6. Flash BORCR address register (FLASH_BORCR)

Address offset:0x24

Reset value:0x0000 E000. After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option byte area of the flash information memory and written to the corresponding option bit of the register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			Res							BOR_EN	Res				
-			-							RW	-				

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	BOR_EN	RW	0	BOR enabled 0:BOR disabled 1:BOR enabled
4:0	Reserved	-	-	Reserved

Attention：The reset value for 13 ~ 15 bit is fixed to 3'b111 and cannot be modified.

4.8.7. Flash WRP area address register (FLASH_WRPR)

Address offset:0x2C

Reset value:0x0000 FFFF

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option byte area of the Flash information memory and written to the corresponding option bit of the register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WRP[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	WRP	RW	1	0:sector 15 with write protection, program and erase are not allowed 1:sector 15 without write protection
14	WRP	RW	1	0:sector 14 with write protection, program and erase are not allowed 1:sector 14 without write protection
13	WRP	RW	1	0:sector 13 with write protection, program and erase are not allowed 1:sector 13 without write protection
12	WRP	RW	1	0:sector 12 with write protection, program and erase are not allowed 1:sector 12 without write protection
11	WRP	RW	1	0:sector 11 with write protection, program and erase are not allowed 1:sector 11 without write protection
10	WRP	RW	1	0:sector 10 with write protection, program and erase are not allowed 1:sector 10 without write protection
9	WRP	RW	1	0:sector 9 with write protection, program and erase are not allowed 1:sector 9 without write protection
8	WRP	RW	1	0:sector 8 with write protection, program and erase are not allowed 1:sector 8 without write protection
7	WRP	RW	1	0:sector 7 with write protection, program and erase are not allowed 1:sector 7 without write protection

Puya Confidential

Bit	Name	R/W	Reset Value	Function
6	WRP	RW	1	0:sector 6 with write protection, program and erase are not allowed 1:sector 6 without write protection
5	WRP	RW	1	0:sector 5 with write protection, program and erase are not allowed 1:sector 5 with write protection
4	WRP	RW	1	0:sector 4 with write protection, program and erase are not allowed 1:sector 4 without write protection
3	WRP	RW	1	0:sector 3 with write protection, program and erase are not allowed 1:sector 3 without write protection
2	WRP	RW	1	0:sector 2 with write protection, program and erase are not allowed 1:sector 2 without write protection
1	WRP	RW	1	0:sector 1 with write protection, program and erase are not allowed 1:sector 1 without write protection
0	WRP	RW	1	0:sector 0 with write protection, program and erase are not allowed 1:sector 0 without write protection

4.8.8. Flash sleep time configuration register (Flash_STCR)

Address offset:0x90

Reset value:0x0000 6400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SLEEP_TIME [7:0]								Res	Res	Res	Res	Res	Res	Res	SLEEP_EN
RW	RW	RW	RW	RW	RW	RW	RW	-	-	-	-	-	-	-	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:8	SLEEP_TIME	RW	0x64	Flash sleep time count (counter based on HSI_10M clock) When the system clock selects LSI or LSE, in order to obtain a more optimized operating mode power consumption, the function of using this register can be selected (this function is only recommended when LSI or LSE is the system clock). When this feature is enabled, the time width for which Flash is in Sleep mode for every half of the system clock low cycle is: t_{HSI_10M} * SLEEP_TIME Notes:

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				t_{HSI_10M} is the period of HSI_10M.
7:1	Reserved	-	-	Reserved
0	SLEEP_EN	RW	0	Flash sleep enable 1:Flash sleep enable 0:Flash sleep disabled

4.8.9. Flash TS0 register (FLASH_TS0)

Address offset:0x100

Reset value:0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	TS0							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7:0	TS0	RW	0xB4	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8MHz:0x3C HSI at 24MHz:0xB4

4.8.10. Flash TS1 register (FLASH_TS1)

Address offset:0x104

Reset value:0x0000 01B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	TS1								
-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:9	Reserved	-	-	Reserved
8:0	TS1	RW	0x1B0	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x90 HSI at 24 MHz:0x1B0

Puya Confidential

4.8.11. Flash TS2P register (FLASH_TS2P)

Address offset:0x108

Reset value:0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	TS2P							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7:0	TS2P	RW	0xB4	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x3C HSI at 24 MHz:0xB4

4.8.12. Flash TPS3 register (FLASH_TPS3)

Address offset:0x10C

Reset value:0x0000 06C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	TPS3										
-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:11	Reserved	-	-	Reserved
10:0	TPS3	RW	0x6C0	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x240 HSI at 24 MHz:0x6C0

4.8.13. Flash TS3 register (FLASH_TS3)

Address offset:0x110

Reset value:0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	TS3							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7:0	TS3	RW	0xB4	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x3C HSI at 24 MHz:0xB4

4.8.14. Flash PAGE ERASE TPE register (Flash_PERTPE)

Address offset:0x114

Reset value:0x0001 1940

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PERTPE[16]
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERTPE [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:17	Reserved	-	-	Reserved
16:0	PERTPE	RW	0x11940	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x6D60 HSI at 24 MHz:0x14820

4.8.15. Flash SECTOR/MASS ERASE TPE register (FLASH_SMERTPE)

Address offset:0x118

Reset value:0x0001 1940

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SMERTPE [16]
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMERTPE [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

puya Confidential

Bit	Name	R/W	Reset Value	Function
31:17	Reserved	-	-	Reserved
16:0	SMERTPE	RW	0x11940	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x6D60 HSI at 24 MHz:0x14820

4.8.16. Flash PROGRAM TPE register (FLASH_PRGTPE)

Address offset:0x11C

Reset value:0x0000 A8C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRGTPE															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PRGTPE	RW	0xA8C0	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x1F40 HSI at 24 MHz:0x5DC0

4.8.17. Flash PRE-PROGRAM TPE register (FLASH_PRETPE)

Address offset:0x120

Reset value:0x0000 12C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	PRETPE [13:0]													
-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:14	Reserved	-	-	Reserved
13:0	PRETPE	RW	0x12C0	If the HSI output frequency is different, the following corresponding values need to be set HSI at 8 MHz:0x640 HSI at 24 MHz:0x12C0

Puya Confidential

5. Power control

5.1. Power supply

5.1.1. Power supply overview

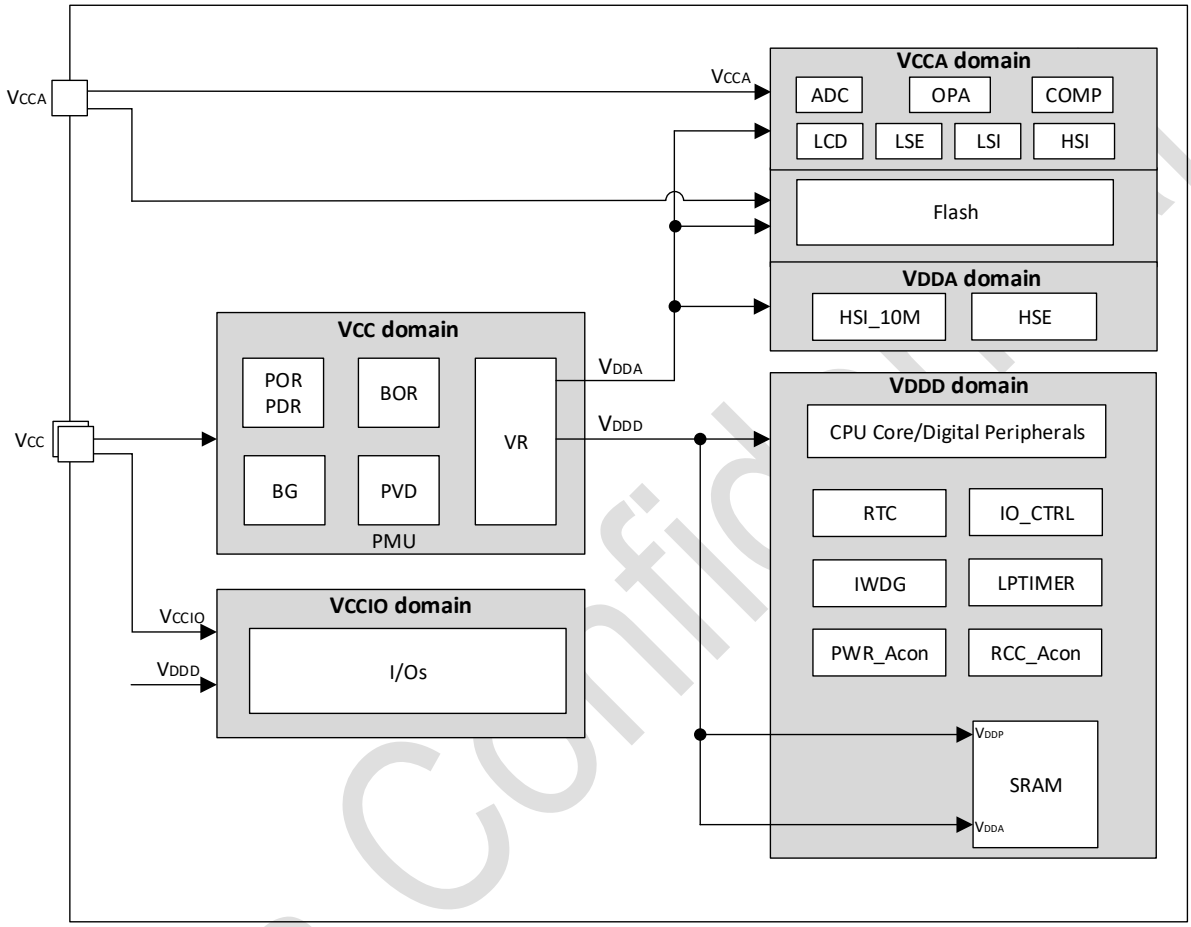


Figure 5-1 Power supply overview

Table 5-1 Power supply overview

No.	Power supply	Power value	Description
1	V_{CC}	3.0 to 5.5 V	The power is supplied to the device through the power pins.
2	V_{CCA}	3.0 to 5.5 V	The power is supplied to the analog modules through the power pins.
3	V_{DDx} （V_{DDD}/V_{DDA}）	1.2 V/1. 0 V/0. 9 V/0. 8 V	VR supplies power to the main logic circuits and SRAM inside the device. When the MR is powered, it outputs 1.2 V. When entering the Stop mode, according to the software configuration, power can be supplied by MR or LPR, and the LPR output is determined to be 1.2 V/1. 0 V/0. 9 V/0. 8 V.

Puya Confidential

5.2. Voltage regulator

The device has two voltage regulators:

- Main regulator (MR) is used in Run mode.
- Low power regulator (LPR) provides lower power consumption options in Stop mode.

The power supply of V_{DDX} comes from MR or LPR depending on the operating mode of the device.

In Run mode, the MR keeps working and outputs 1.2 V voltage.

In Stop mode, power can be supplied from MR or LPR by software. Similarly, it is determined by the software that after entering Stop, the V_{DDX} with LPR power supply is 1.2 V/1.0 V/0.9 V/0.8 V.

5.3. Power monitoring

5.3.1. Power-on reset (POR) / power-down reset (PDR) / brown-out reset (BOR)

The device has an integrated power-on reset (POR) / power-down reset (PDR). The module keeps working in all modes.

In addition to POR/PDR, BOR (Brown-out reset) is implemented. The BOR can only be enabled and disabled by option bytes.

When the BOR is enabled, the threshold of the BOR can be selected by the option byte.

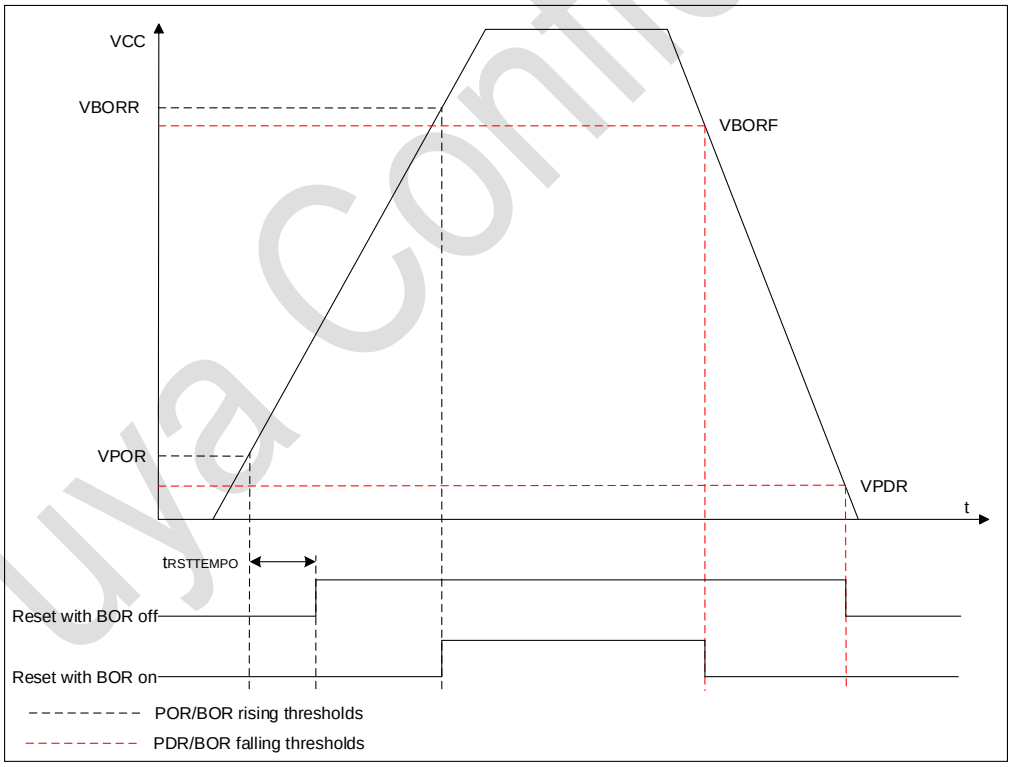


Figure 5-2 POR/PDR/BOR threshold

Puya Confidential

5.3.2. Programmable voltage detector (PVD)

This module can be used to detect the V_{CC} power supply (it can also detect the voltage of the PB7 pin), and the detection point can be configured through registers. When the V_{CC} is higher or lower than the detection point of the PVD, a corresponding mark is generated.

This event is internally connected to line 16 of EXTI. Depending on the rising/falling edge configuration of EXTI line 16, when V_{CC} rises beyond the detection point of PVD, or V_{CC} falls below the detection point of PVD, an interrupt is generated, and the user can perform an emergency Shutdown task in the interrupt service routine.

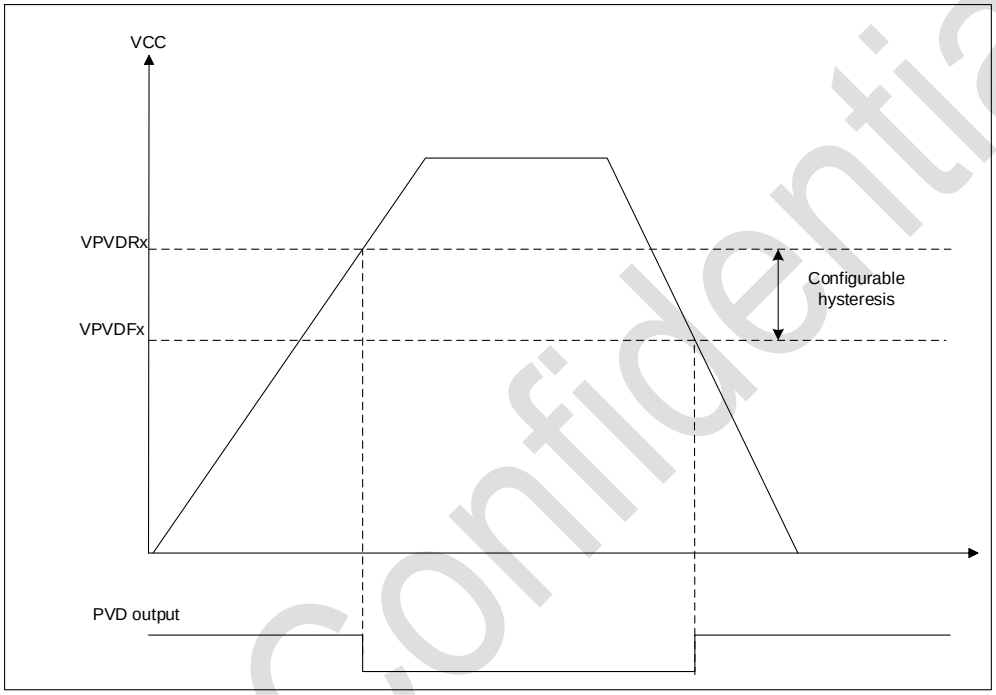


Figure 5-3 PVD threshold

Puya Confidential

6. Low power control

By default, the microcontroller is in Run mode after a system or a power reset. Low-power modes are available to save power when the CPU does not need to be kept running. For example, when waiting for an external event, software can choose a compromise between power consumption, wake-up time and wake-up source.

6.1. Low-power modes

6.1.1. Introduction

In addition to the Run mode, the device has 2 low-power modes:

- Sleep mode:Peripherals can be configured to keep working when the CPU clock is off (NVIC, SysTick, etc.). It is recommended only to enable the modules that must work, and close the module after the module works.
- Stop mode:In this mode, the contents of SRAM and registers are maintained, HSI and HSE are disabled, and the clocks of most modules in the V_{DDD} domain are stopped.

In Stop mode, LSI and LSE can keep working, and RTC, LPTIMER, IWDG, etc. can keep working. For details of the working conditions of each module in this mode, refer to table below.

In the Stop mode, the corresponding VR state can be controlled by software and set to be powered by MR or LPR. When powered by LPR, the device consumption is reduced with longer wake-up time, When powered by MR, it has larger consumption but shorter wake-up time.

In addition, power consumption can be reduced in Run mode by:

1. Reduce the system clock frequency
2. For unused peripherals, turn off peripheral clocks (system clock and module clock)

To sum up, the low-power mode transitions diagram of this project are as follows.

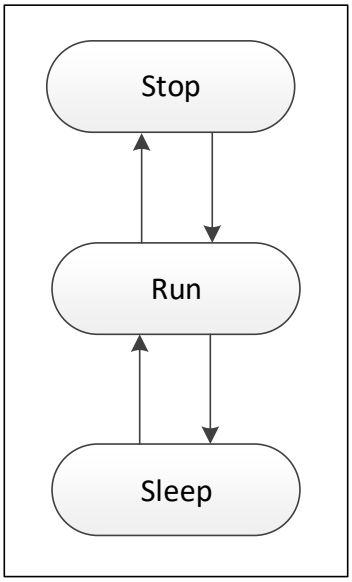


Figure 6-1 Power mode

Puya Confidential

6.1.2. Low power mode

Table 6-1 Low power modes

Mode	Entry	Wake-up source	Wake-up clock	Effect	Voltage regulators	
					MR	LPR
Sleep (sleep-now or sleep-on-exit)	WFI or Return from ISR	Any interrupt	Same as before entering Sleep mode	CPU clock OFF, no effect on other clocks or analog clock sources	ON(1)	ON
	WFE	Wake up event				
Stop	SLEEPDEEP bit+ WFI or Return from ISR or WFE Note:LSI cannot be selected for system clock	Any wake up EXTI Line (configured in EXTI register), IWDG and NRST	HSI. HSI maintains the frequency configuration before entering Stop mode, without frequency division.	HSI OFF, HSE OFF, LSI can be selected as ON or OFF, LPTIM, RTC and IWDG:configured by the software Low power wake-up and some modules such as RCC keep working, The clock is off for the remaining modules.	Software configuration working mode	Software configuration working mode, output voltage 1.2 V/1. 0 V/0. 9 V/0. 8 V configurable

Note 1:The software must configure the state of VR to MR mode before entering the Sleep mode.

6.1.3. Functionalities depending on the working mode

Table 6-2 Functionalities depending on the working mode(1)

Peripheral	Run	Sleep	Stop	
			VR@LPR or VR@MR	Wake up capability
CPU	Y	-	-	-
Flash memory	Y	Y	-(2)	-
SRAM	Y	O(3)	-(4)	-
Brown-out reset (BOR)	Y	Y	O	O
PVD	O	O	O	O
DMA	O	O	-	-
HSI	O	O	-	-
HSE	O	O	-	-
LSI	O	O	O	-
HDIV	O	O	-	-
HSE clock security system (CSS)	O	O	-	-
RTC	O	O	O	O

Puya Confidential

Peripheral	Run	Sleep	Stop	
			VR@LPR or VR@MR	Wake up capability
USART1/USART2	O	O	-	-
I2C1/I2C2	O	O	-	-
SPI1/SPI2	O	O	-	-
ADC	O	O	-	-
COMP1/COMP2	O	O	O	O
Temperature sensor	O	O	-	-
OPA	O	O	-	-
LCD	O (5)	O (5)	O (5)	-
Timers (TIM1/TIM2/TIM3/TIM6/TIM7/ /TIM14/TIM15/TIM16/TIM17)	O	O	-	-
LPTIM	O	O	O	O
IWDG	O	O	O	O
WWDG	O	O	-	-
SysTick timer	O	O	-	-
CRC	O	O	-	-
GPIOs	O	O	O	O

1. Legend:Y = Yes (Enable). O = Optional (Disable by default. Can be enabled by software). - = Not available
2. Flash does not power down, but no clock is provided, and enters the lowest power consumption state.
3. The SRAM clock can be gated on or off.
4. The SRAM is not powered down, but no clock is provided and enters the lowest power consumption state.
5. LCD controller can display in Run, Sleep and Stop modes.

6.2. Sleep mode

6.2.1. Enter Sleep mode

The Sleep mode is entered by executing the WFI (Wait for Interrupt) or WFE (Wait for Event) instructions. Two options are available to select the Sleep mode entry mechanism, depending on the SLEEPONEXIT bit in the Cortex®-M0+ System Control register.

- Sleep-now:if the SLEEPONEXIT bit is cleared, the MCU enters Sleep mode as soon as WFI or WFE instruction is executed.
- Sleep-on-exit:if the SLEEPONEXIT bit is set, the MCU enters Sleep mode as soon as it exits the lowest priority ISR.

In the Stop mode, all I/O pins keep the same state as in the Run mode.

Puya Confidential

6.2.2. Exit Sleep mode

If the WFI instruction is used to enter Sleep mode, any peripheral interrupt acknowledged by the nested vectored interrupt controller (NVIC) can wake up the device from Sleep mode.

If the WFE instruction is used to enter Sleep mode, the MCU exits Sleep mode as soon as an event occurs. Wakeup events can be generated by:

- Enable the interrupt in the peripheral control register, not in the NVIC, and enable the SEVON-PEND bit of Cortex M0 +. When the device continues execution after waking up from the WFE, the peripheral interrupt Pending bit and the peripheral NVIC IRQ channel Pending bit (in the NVIC's interrupt clear Pending register) must be cleared.
- Configuring an external or internal EXTI line in event mode. When the CPU continues execution after waking up from the WFE, it is not necessary to clear the peripheral interrupt Pending bit, or the NVIC IRQ channel Pending bit corresponding to the event Line is not set.

This mode offers the lowest wake up time as no time is wasted in interrupt entry/exit.

Table 6-3 Sleep-now

Sleep-now mode	Description
Mode entry	WFI (wait for interrupt) or WFE (wait for event) while: - SLEEPDEEP = 0 and - SLEEPONEXIT = 0
Mode exit	If WFI was used for entry, the exit mode is:interrupt. If WFE was used for entry, the exit mode is:wake-up event.
Wakeup latency	None

Table 6-4 Sleep-on-exit

Sleep-on-exit	Description
Mode entry	WFI (wait for interrupt) while: - SLEEPDEEP = 0 and - SLEEPONEXIT = 1
Mode exit	Interrupts and events
Wakeup latency	None

6.3. Stop mode

The Stop mode is based on the Cortex®-M0+ deep sleep mode combined with peripheral clock gating. The voltage regulator can be Supplied by either MR or LPR. In this mode, HSI and HSE are turned off, SRAM and register contents are preserved, LSI, LPTIMER, RTC, IWDG can be configured by software, low power wake-up and some RCC logic are kept working, and the clock inputs of digital modules in the remaining V_{DDx} domain are turned off.

In Stop mode, all IO pins remain in the same state as in Run mode.

Puya Confidential

6.3.1. Enter Stop mode

In order to further reduce the power consumption of Stop mode, when PWR_CR1. LPR = 1 is configured, VR can enter LPR power supply.

If Flash memory programming is ongoing, the Stop mode entry is delayed until the memory access is finished (the software reads the BSY bit of the Flash_SR register to determine whether the erase and write operation has finished).

If an access to the APB domain is ongoing, The Stop mode entry is delayed until the APB access is finished.

If the system clock source is a high-speed clock source (HSE) before entering the low-power mode, to ensure the successful switching, software needs to switch the system clock source to HSI.

6.3.2. Exit Stop mode

When exiting Stop mode by issuing an interrupt or a wakeup event, the HSI oscillator is selected as system clock.

When the voltage regulator operates in low-power mode, an additional startup delay is incurred when waking up from Stop mode.

By keeping the internal regulator ON during Stop mode, the consumption is higher, although the startup time is reduced.

Table 6-5 Stop mode

Stop mode	Description
Mode entry	WFI (wait for interrupt) or WFE (wait for event), while: 1. Settings: - Select the voltage regulator mode by configuring LPR bit in PWR_CR1 - The LPR mode is selected to provide 1.2 V, 1.0 V, 0.9 V, 0.8 V by the VOS bit of PWR_CR1 - Set the wake-up time by configuring FLS_SLPTIME bit in PWR_CR1 2. Set SLEEPDEEP bit in Cortex®-M0+ System Control register Notes: 1. To enter Stop mode, all EXTI line pending bits (EXTI_PR register), all peripheral interrupt pending bits and RTC alarm flag bits must be reset. Otherwise, the Stop mode entry procedure is ignored and program execution continues. 2. If the application needs to turn off HSE before entering Stop mode, the system clock source must first switch to HSI and then clear the HSEON bit. 3. In order to make the change of power consumption as balanced as possible, the software needs to follow the principle of step-by-step shutdown:step-by-step shutdown the clock of each module, select HSI as the system clock, and turn off HSE. 4. In order to shorten the wake-up time, before entering Stop mode, the system clock should be configured to select the HSI high-frequency clock, and the HPRE of the RCC _CFGR register should be set to 0, otherwise the hardware switching clock will consume additional clock after wake-up.

Puya Confidential

Stop mode	Description
Mode exit	If WFI was used for entry: - Any EXTI line configured in interrupt mode (the corresponding EXTI interrupt vector must be enabled in NVIC) If WFE was used for entry: - Any EXTI Line configured in Interrupt mode - Interrupt pending flag bit in case of CPU SEVONPEND is set
Wakeup latency	Wake-up time from LPR to MR + HSI wake-up time + Flash wake-up time

6.4. Reduce the system clock frequency

In Run mode, the frequency of the system clock (SYSCLK, HCLK, PCLK) can be reduced by configuring frequency division through the prescaler register. These prescalers can also be used to reduce the frequency of peripherals before entering sleep mode.

6.5. Peripheral clock gating

In Run mode, the AHB clock (HCLK) and APB clock (PCLK) of individual peripherals and memories can be stopped at any time to reduce power consumption.

To further reduce power consumption in sleep mode, the clock of the peripheral may be stopped prior to execution of WFI or WFE instructions.

6.6. Power control registers

The peripheral's registers can be accessed by half-word or word.

6.6.1. Power control register 1 (PWR_CR1)

Address offset:0x00

Reset value:0x00000000 (reset by POR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res												HSION_CTRL	Res		
-												RW	-		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	LPR	FLS_SLPTIME[1:0]		Res	VOS [1:0]		DBP	Res				Res			
-	RW	RW	RW		RW		RW	-				-			

Bit	Name	R/W	Reset Value	Function
31:20	Reserved	-	-	Reserved
19	HSION_CTRL	RW	0	When waking up from Stop mode, the HSI turns on time control. 0:After waiting for MR to stabilize, enable HSI; 1:Enable HSI immediately when waking up.
18:15	Reserved	-	-	Reserved
14	LPR	RW	0	Low power regulator

Puya Confidential

				0:Main regulator works in Stop mode 1:Low power regulator works in Stop mode
13:12	FLS_SLPTIME[1:0]	RW	2’b00	In the Stop mode wake-up timing, after the HSI is stabilized, a waiting time is required before the Flash operation. 2 'b00:1 μs 2 'b01:2 μs 2 'b10:3 μs 2 'b11:0 μs Note:When this register is set to 2’b11, it indicates that the program is executed from SRAM after wakeup, not Flash. And the program guarantees that Flash will not be accessed within 3us after waking up the execution program.
11	Reserved	-	-	Reserved
10:9	VOS [1:0]	RW	0	Voltage regulation range selection 00:After entering Stop mode, V_{DD} = 1.2 V 01:After entering Stop mode, V_{DD} = 1.0 V 10:After entering Stop mode, V_{DD} = 0.9 V 11:After entering Stop mode, V_{DD} = 0.8 V
8	DBP	RW	0	RTC write protection disabled After reset, the RTC is write-protected to prevent unwanted writes. To access RTC this bit must be set to 1. 0:Access to RTC disabled 1:Access to RTC abled
7:0	Reserved	-	-	Reserved

6.6.2. Power control register 2 (PWR_CR2)

Address offset:0x04

Reset value:0x0000 0500(reset by POR)

Note:This register is a PVD function related register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res				FLT_TIME[2:0]			FLTEN	Res	PVDT [2:0]			Res	SRCSEL	Res	PVDE
-				RW			RW	-	RW			-	RW	-	RW

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved

Puya Confidential

Bit	Name	R/W	Reset Value	Function
11:9	FLT_TIME[2:0]	RW	3'b010	Digital filtering time configuration 110:Filtering time is approximately 30.7 ms (1024 LSI/LSE clocks) 101:Filtering time is approximately 3.8 ms (128 LSI/LSE clocks) 100:Filtering time is approximately 1.92 ms (64 LSI/LSE clocks) 011:Filtering time is approximately 480 μs (16 LSI/LSE clocks) 010:Filter time is approximately 120 μs (4 LSI/LSE clocks) 001:Filtering time is approximately 60 μs (2 LSI/LSE clocks) 000:Filtering time is approximately 30 μs (1 LSI/LSE clock)
8	FLTEN	RW	1	Digital filter function 0:Disabled 1:Enabled
7	Reserved	-	-	Reserved
6:4	PVDT [2:0]	RW	000	Voltage rising edge detection threshold (falling edge detection threshold is reduced by 0.1 V accordingly) and PVDIN detection control. 000:Reserved(default) 001:Reserved 010:Reserved 011:Reserved 100:Reserved 101:Reserved 110:Reserved 111:VPVD7 (around 3.2 V)
3	Reserved	-	-	Reserved
2	SRCSEL	RW	0	PVD detects power selection. 0:V_{CC} 1:Detect PB7 pin If this bit is set, the voltage on PB7 is internally compared to VREF1P2 (including rise and fall thresholds). In this case, the setting of the PVDT register is invalid.
1	Reserved	-	-	Reserved
0	PVDE	RW	0	Voltage detection enable bit 0:Voltage detection disabled 1:Voltage detection enabled PVDE write protection if SYSCFG_C, G2.PVD_LOCK = 1. Write protection is reset only when the system is reset.

Puya Confidential

6.6.3. PWR status register (PWR_SR)

Address offset:0x14

Reset value:0x00000000 (reset by POR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	PVDO	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	**-**	**-**	**-**	R	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11	PVDO	R	0	PVD detection result output. 0:The detected V_{CC} or PB7 exceeds the comparison threshold of PVD selection 1:The detected V_{CC} or PB7 is lower than the comparison threshold of PVD selection
10:0	Reserved	-	-	Reserved

Puya Confidential

7. Reset

Two resets are designed:power reset and system reset.

7.1. Reset source

7.1.1. Power reset

A power reset is generated when one of the following events occurs:

The POR/BOR generated by the analog circuit detects V_{CC}.

- Releasing the reset when the V_{CC} voltage rises to the trigger value;
- A reset occurs when the V_{CC} voltage drops to a certain trigger value.

7.1.2. System reset

A system reset sets all registers to their reset values except the reset flags in the clock control/status register and the registers in the RTC domain.

System reset is generated when one of the following events occurs:

- NRST pin (external reset)
- Window watchdog reset (WWDG)
- Independent watchdog reset (IWDG)
- Cortex-M0+ SYSRESETREQ software reset
- Option byte load reset (OBL)

7.1.3. NRST pin (external reset)

Through specific option bits (NRST_MODE), the NRST pin is configurable for operating as (see the option byte description for specific configuration):

- Reset input

  In this mode, any valid reset signal on the NRST pin is propagated to device internal logic, but resets generated internally by the device are not visible on the pin.

  In this mode, the GPIO functionality (PF2) is not available.

  After the NRST pin is input, an external reset of the device is generated after passing through the deburring circuit (deburring can be configured to disable).

- GPIO

  In this mode, the pin can be used as PF2 standard GPIO. The reset function of the pin is not available. Reset is only possible from device internal reset sources, and it is not propagated to the pin.

Note:After power-on reset, the NRST pin is configured to reset input mode by default.

7.1.4. Watchdog reset

See independent watchdog (IWDG) and window watchdog (WWDG).

Puya Confidential

7.1.5. Software reset

Software reset can be achieved by setting the SYSRESETREQ bit of the ARM M0+ interrupt and reset control register.

7.1.6. Option byte loader reset

The option byte loader reset is generated when the OBL_LAUNCH bit is set in the FLASH_CR register.

Puya Confidential

8. Clocks

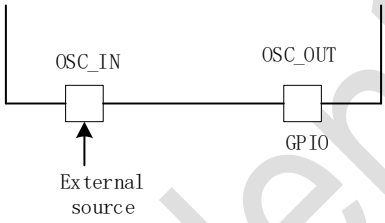
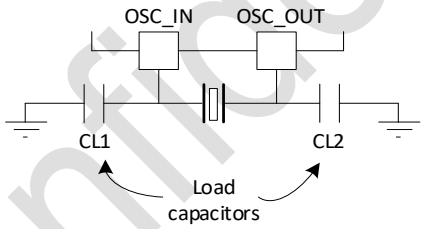
8.1. Clock sources

8.1.1. External high speed clock HSE

The external high speed clock (HSE) comes from two sources:

- External XTAL OSC + internal oscillator circuit
- External clock input via OSC_IN (HSEBYP = 1)

Table 8-1 HSE clock source

Clock source	Hardware configuration
External clock	
Circumscribed crystal	

External high frequency OSC, frequency range 4 to 24 MHz.

The stable time of the HSE clock is determined by RCC_ECSCR. HSE_STARTUP register configuration. When HSE goes from OFF to ON, it needs to wait for stabilization time. After stabilization, the hardware sets the RCC_CR. HSERDY register. When HSEBYP = 1, the settling time is halved compared to the non-bypass mode.

The HSE clock related register refers to RCC_ECSCR.

8.1.2. Internal high-speed clock HSI

Internal RC oscillator, reference frequencies can be 8 MHz or 24 MHz Compared with XTAL OSC, RC OSC has low power consumption, shorter stabilization time, but low accuracy.

After the power-on reset, the HSI calibration value needs to be loaded by software into the RCC_ICSCR.HSI_TRIM register. When the system is reset, this register is reset with it.

After waking up from Stop mode, only HSI can be used as the system clock source.

8.1.3. Internal low speed clock LSI

Internal low frequency 32.768/23.552/38.912/39.168/43.008 kHz clock.

Puya Confidential

8.1.4. HSI10M clock

This clock serves as a low-precision clock, used as a filter count for NRST pins, and as low-power processing when Flash runs at low speeds.

8.1.5. LSE clock

External 32.768 kHz OSC, used as a low power clock.

A balance between settling time and power consumption can be achieved by configuring LSE_DRV. The LSE stabilization time is determined by RCC_ECSCR. LSE_STARTUP register configuration.

Similar to the HSE source, the LSE also has two sources:

- 32.768 kHz XTAL+ Internal vibration circuit
- External clock input via OSC_IN (LSEBYP = 1)

In the LSE bypass case, the settling time is halved compared to the non-bypass mode.

Puya Confidential

8.2. Clock tree

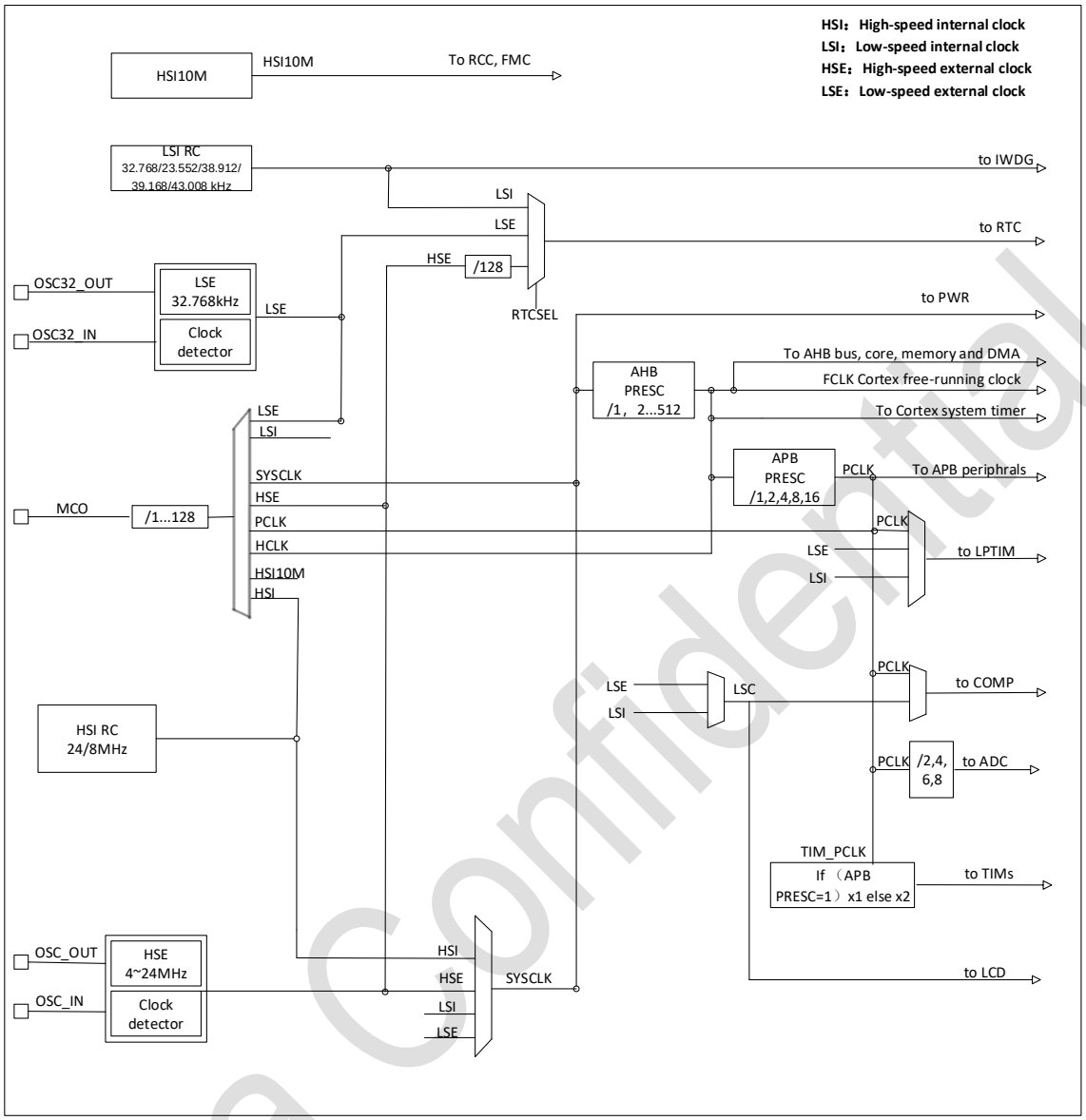


Figure 8-1 System clock structure diagram

8.3. Clock security system (CSS)

Clock security mainly includes the following aspects:

- Clock configuration and state security
- Clock source HSE security
- Clock source LSE security
- Clock security based on IWDG
- Timer-based clock security

8.3.1. Clock configuration and state security

The software periodically reads back the clock configuration and status registers to obtain the current clock information of the system and judge whether it is consistent with the expectation.

Puya Confidential

8.3.1.1. Clock source HSE monitoring

The HSE clock safety system can be activated by software by configuring RCC_CR. CSSON. In this case, after the HSE starts, the clock detection function is turned on. When the HSE is turned off, the clock detection function is turned off.

If a clock error is found on the HSE, the HSE will be automatically shut down, and the clock error event is sent to the brake inputs of TIM1 (advanced timer) and TIM15/TIM16/TIM17 (general timer), and an interrupt is generated to notify the software of the error (clock security system interrupt CSSI), which in turn allows the MCU to perform rescue operations. The CSSI is linked to the NMI (Non-maskable interrupt) Exception vector of Cortex-M0 +.

Note: Once CSS is enabled, and if the HSE clock failure, a CSS interrupt is generated and an NMI is automatically generated. The NMI will be executed indefinitely unless the CSS interrupt pending bit is cleared. As a consequence, in the NMI ISR user must clear the CSS interrupt by setting the CSSC bit in the Clock interrupt register (RCC_CIR).

8.3.1.2. Clock source LSE monitoring

The LSE clock security system can be activated by software by configuring RCC_BDCR.LSECS-SON. In this case, after the LSE starts, the clock detection function is turned on. When the LSE is turned off, the clock detection function is turned off.

If a clock error is found on the LSE, the LSE will be automatically shut down, and the clock error event is sent to the brake inputs of TIM1 (advanced timer) and TIM15/TIM16/TIM17 (general timer), and an interrupt is generated to notify the software of the error (CSSI), which in turn allows the MCU to perform rescue operations. The CSSI is linked to the NMI (Non-maskable interrupt) of Cortex-M0+.

Note: Once the LSECSS is enabled and if the HSE clock fails, the CSS interrupt occurs and an NMI is automatically generated. The NMI will be executed indefinitely unless the CSS interrupt pending bit is cleared. As a consequence, in the NMI ISR user must clear the CSS interrupt by setting the CSSC bit in the Clock interrupt register (RCC_CIR).

If the LSE oscillator is used as the system clock, a detected failure causes a switch of the system clock to the LSI oscillator and the disabling of the LSE oscillator. At the same time, if the LPTIM and RTC counting clocks select LSE, they will also automatically switch to LSI.

8.4. Output clock capability

In order to facilitate board-level applications, save BOM costs, and meet debugging requirements, the device needs to provide clock output functions. That is, the MCO signal (parallel frequency division) in the table below is used to realize the clock output function through the multiplexing function of GPIO.

Table 8-2 Output clock selection

Clock sources	MCO output clock source
HSI	√
SYSCLK	√

Puya Confidential

Clock sources	MCO output clock source
HSE	√
LSI	√
LSE	√

Note:When the MCO clock source is switched and the GPIO AF function is selected as the initial stage of the MCO, the MCO may generate glitches and need to be avoided for this period of time.

8.5. Reset/clock register

The registers can be accessed in words (32 bits), half words (16 bits), and bytes (8 bits).

8.5.1. Clock control register (RCC_CR)

Address offset:0x00

Reset value:0x0000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	ADC_DIV		Res	CSS ON	HSE BYP	HSE RDY	HSE ON
-	-	-	-	-	-	R	RW	-	RW		-	RS	RW	R	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	HSI RDY	Res	HSION	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	R	-	RW	-	-	-	-	-	-	-	-

Bit	Name	R/W	Reset Value	Function
31:23	Reserved	-	-	Reserved
22:21	ADC_DIV	RW	0	ADC frequency division coefficient 00:2 division 01:4 division 10:6 division 11:8 division
20	Reserved	-	-	Reserved
19	CSSON	RS	0	HSE clock security system enable. When the bit is 1, the hardware enables the clock detection module if the HSE OSC is ready; If the HSE detection fails, the clock detection module is turned off. 0:Clock safety system off (clock detection off) 1:Clock safety system is on (if HSE clock is stable, clock detection is on, otherwise clock detection is off)
18	HSEBYP	RW	0	HSE shielded crystal oscillator, select pin input clock. This bit can only be written if HSEON = 0.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:HSE crystal oscillator is not shielded, external high-speed clock selects external crystal oscillator 1:HSE crystal oscillator shielding, external high-speed clock select external pin input clock source
17	HSERDY	R	0	HSE crystal oscillator clock ready flag. This bit is set by hardware to indicate that the HSE crystal oscillator is stable. 0:HSE crystal oscillator is not ready 1:HSE crystal oscillator is ready Note:Once the HSEON bit is cleared, HSERDY clears after 6 HSE clock cycles.
16	HSEON	RW	0	HSE crystal oscillator enabled. When entering Stop mode, the hardware will clear this bit and turn off the HSE crystal oscillator. When HSE is the system clock source, this bit cannot be set to 0. 0:HSE crystal oscillator off 1:HSE crystal oscillator on
15:11	Reserved	-	-	Reserved
10	HSIRDY	R	0	HSI clock ready flag. Set by hardware to indicate that the HSI oscillator is stable This bit is only valid if HSION = 1. 0:HSI OSC not ready 1:HSI OSC ready
9	Reserved	-	-	Reserved
8	HSION	RW	1	HSI clock enable. When entering Stop mode, this bit will be cleared to stop HSI if needed. 0:HSI OSC disabled 1:HSI OSC enabled
7:0	Reserved	-	-	Reserved

8.5.2. Internal clock sources calibration register (RCC_ICSCR)

Address offset:0x043080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res		Res	LSI_TRIM[8:0]								
-	-	-	-	-		-	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HSI_FS[2:0]			HSI_TRIM[12:0]												
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:25	Reserved	-	-	Reserved
24:16	LSI_TRIM	RW	9'h0FF	Internal low-speed clock frequency adjusts through calibration and its frequency is 32.768 kHz. The LSI frequency can be changed by the Trim value from the following address: 0x1FFF 3080:32.768 kHz 0x1FFF 3088:23.552 kHz 0x1FFF 3090:39.168 kHz 0x1FFF 3098:38.912 kHz 0x1FFF 30A0:43.008 kHz
15:13	HSI_FS	RW	3'h001	HSI frequency: 00:Reserved 001:8 MHz 010:Reserved 011:Reserved 100:24 MHz ≥101:Reserved
12:0	HSI_TRIM	RW	13'h1080	Clock frequency adjustment, changing the value of this register can adjust the output frequency of HSI. Every increase in the register value increases the output frequency of the HSI by approximately 0.2%, and the total adjustment range is 8 to 24 MHz. The calibration values corresponding to 24 MHz/8 MHz are stored in Flash at the following address: 24 MHz calibration address:0x1FFF 3220 8 MHz calibration address:0x1FFF 3208

8.5.3. Clock configuration register (RCC_CFGR)

Address offset:0x08

Reset value:0x0000 0000

When the clock source is switched, there is a waiting period of 1 or 2 clocks to access this register.

When the APH or AHB frequency division value is updated, there may be a waiting period of 0 to 15 clocks to access this register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	MCOPRE [2:0]			MCOSEL [3:0]				Res	Res	Res	Res	Res	Res	Res	Res
-	RW	RW	RW	RW	RW	RW	RW	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	PPRE [2:0]			HPRE [3:0]				Res	Res	SWS [2:0]			SW [2:0]		

Puya Confidential

-	RW	RW	RW	RW	RW	RW	RW	-	-	R	R	R	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31	Reserved	-	-	Reserved
30:28	MCOPRE [2:0]	RW	0	MCO (Microcontroller clock output) frequency division factor. The software controls these bits and sets the frequency division factor of the MCO output: 000:1 001:2 010:4 011:8 100:16 101:32 110:64 111:128 The division coefficient is configured before the MCO output is enabled.
27:24	MCOSEL [3:0]	RW	0	MCO Selection 0000:no clock，MCO output disabled 0001:SYSCLK 0010:HSI10M 0011:HSI 0100:HSE 0101:Reserved 0110:LSI 0111:LSE 1000:HCLK 1001:PCLK Others:no clock Note:The output clock may be incomplete during the clock startup or switching phase.
23:15	Reserved	-	-	Reserved
14:12	PPRE [2:0]	RW	0	This bit is controlled by software. To generate the PCLK clock, it sets the division coefficients of HCLK as follows: 0xx:1 100:2 101:4 110:8 111:16
11:8	HPRE [3:0]	RW	0	AHB clock division factor.

Bit	Name	R/W	Reset Value	Function
				The software controls this bit. To generate the HCLK clock, it sets the division coefficients of SYSCLK as follows: 0xxx:1 1000:2 1001:4 1010:8 1011:16 1100:64 1101:128 1110:256 1111:512 To ensure the normal operation of the system, the appropriate frequency needs to be configured according to the VR power supply situation. Note:It is recommended to switch the frequency division coefficients gradually.
7:6	Reserved	-	-	Reserved
5:3	SWS [2:0]	R	0	System clock switching status bit These bits are controlled by hardware and indicate which clock source is currently being used as the system clock: 000:HSI 001:HSE 010:Reserved 011:LSI 100:LSE Others:Reserved
2:0	SW [2:0]	RW	0	System clock source select bit. These bits are controlled by software and hardware and are used to select the system clock: 000:HSI 001:HSE 010:Reserved 011:LSI 100:LSE Others:Reserved Scenarios where the hardware is configured as HSI include: 1) The system exits from Stop mode

Bit	Name	R/W	Reset Value	Function
				2) Software configuration 001 (HSE), and HSE failure occurs (HSE is the system clock source)

8.5.4. External clock sources control register (RCC_ECSCR)

Address offset:0x10

Reset value:0x0003 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LSE_STARTUP		Res	Res	LSE_DRV	
-	-	-	-	-	-	-	-	-	-	RW		-	-	RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	HSE_STARTUP		Res	HSE_DRV	
-	-	-	-	-	-	-	-	-	-	-	RW			RW	

Bit	Name	R/W	Reset Value	Function
31:22	Reserved	-	-	Reserved
21:20	LSE_STARTUP	RW	0	LSE crystal oscillator stabilization time selection. LSEBYP=0: 00:4096 LSE clock cycles 01:2048 LSE clock cycles 10:8192 LSE clock cycles 11:Direct output regardless of stabilization time LSEBYP=1: 00:2048 LSE clock cycles 01:1024 LSE clock cycles 10:4096 LSE clock cycles 11:Direct output regardless of stabilization time
19:18	Reserved	-	-	Reserved
17:16	LSE_DRV	RW	0x3	Low-speed crystal oscillator driving capability selection. 00:Reserved 01:Weak driving ability 10:Strong driving ability (recommended) 11:Strongest driving capability Note:It is necessary to select the appropriate driving capability according to the crystal oscillator characteristics, load capacitance and parasitic parameters of the circuit board. The greater the driving capability, the greater the power consumption, and vice versa.
15:5	Reserved	-	-	Reserved
4:3	HSE_STARTUP	RW	0	HSE crystal oscillator stabilization time selection.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				HSEBYP=0: 00:4096 HSE clocks 01:2048 HSE clocks 10:8192 HSE clocks 11:Direct output regardless of stabilization time HSEBYP=1: 00:2048 HSE clocks 01:1024 HSE clocks 10:4096 HSE clocks 11:Direct output regardless of stabilization time
2	Reserved	-	-	Reserved
1:0	HSE_DRV	RW	0x3	High-speed crystal oscillator driving capability selection. 00:Reserved; 01:Weak driving ability 10:Strong driving ability (recommended) 11:Strongest driving capability Note:It is necessary to select the appropriate driving capability according to the crystal oscillator characteristics, load capacitance and parasitic parameters of the circuit board. The greater the driving capability, the greater the power consumption, and vice versa.

8.5.5. Clock interrupt enable register (RCC_CIER)

Address offset:0x18

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res											HSE RDYIE	HSI RDYIE	Res	LSE RDYIE	LSI RDYIE
-											RW	RW	-	RW	RW

Bit	Name	R/W	Reset Value	Function
31:5	Reserved	-	-	Reserved
4	HSERDYIE	RW	0	HSE clock ready interrupt enable 0:Disabled 1:Enabled
3	HSIRDYIE	RW	0	HSI clock ready interrupt enable

Confidential

Bit	Name	R/W	Reset Value	Function
				0:Disabled 1:Enabled
2	Reserved	-	-	Reserved
1	LSERDYIE	RW	0	LSE clock ready interrupt enable 0:Disabled 1:Enabled
0	LSIRDYIE	RW	0	LSI clock ready interrupt enable 0:Disabled 1:Enabled

8.5.6. Clock interrupt flag register (RCC_CIFR)

Address offset:0x1C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res						LSE CSSF	CSSF	Res	Res	Res	HSE RDYF	HSI RDYF	Res	LSE RDYF	LSI RDYF
-						R	R	-	-	-	R	R	-	R	R

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9	LSECSSF	R	0	LSE Clock security system interrupt flag. Set by hardware when a failure is detected in the LSE oscillator. 0:No clock security interrupt caused by LSE clock failure 1:Clock security interrupt caused by LSE clock failure Write LSECSSC register 1 to clear this bit.
8	CSSF	R	0	HSE clock safe system interrupt flag bit. Set by hardware when a failure is detected in the HSE oscillator. 0:No clock security interrupt caused by HSE clock failure 1:Clock security interrupt caused by HSE clock failure Write CSSC register 1 to clear this bit.
7:5	Reserved	-	-	Reserved
4	HSERDYF	R	0	HSE ready interrupt flag bit

Bit	Name	R/W	Reset Value	Function
				Set by hardware when the HSE clock becomes stable and HSERDYIE is enabled. Cleared by software setting the HSERDYC bit. 0:No clock ready interrupt caused by the HSE oscillator 1:Clock ready interrupt caused by the HSE oscillator Write HSERDYC register 1 to clear this bit.
3	HSIRDYF	R	0	HSI clock ready interrupt flag. Set by hardware when the HSI clock becomes stable and HSIRDYDIE = 1. 0:HSI clock ready interrupt not generated; 1:HSI clock ready interrupt generated; Write HSIRDYC register 1 to clear this bit.
2	Reserved	-	-	Reserved
1	LSERDYF	R	0	LSERDY clock ready interrupt flag. Set by hardware when the LSE clock becomes stable and LSERDYDIE = 1. 0:LSERDY clock ready interrupt not generated; 1:LSERDY clock ready interrupt generated; Write LSERDYC register 1 to clear this bit.
0	LSIRDYF	R	0	LSI ready interrupt flag Set by hardware when the LSE clock becomes stable and LSERDYIE is enabled. Cleared by software setting the LSERDYC bit. 0:No clock ready interrupt caused by the LSI oscillator 1:Clock ready interrupt caused by the LSI oscillator Write LSIRDYC register 1 to clear this bit.

8.5.7. Clock interrupt clear register (RCC_CICR)

Address offset:0x20

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res						LSE CSSC	CSSC	Res		Res	HSE RDYC	HSI RDYC	Res	LSE RDYC	LSI RDYC
-						W	W	-		-	W	W	-	W	W

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9	LSECSSC	W	0	LSE clock security system interrupt flag. 0:No effect 1:Clear LSECSSF flag
8	CSSC	W	0	Clock safe interrupt clear bit. 0:No effect 1:Clear CSSF flag bit
7:5	Reserved	-	-	Reserved
4	HSERDYC	W	0	HSE ready flag cleared. 0:No effect 1:Clear HSERDYF bit
3	HSIRDYC	W	0	HSI ready flag cleared. 0:No effect 1:Clear HSIRDYF bit
2	Reserved	-	-	Reserved
1	LSERDYC	W	0	LSE ready interrupt flag cleared. 0:No effect 1:Clear LSERDYF flag
0	LSIRDYC	W	0	LSI ready flag cleared. 0:No effect 1:Clear LSIRDYF bit

8.5.8. I/O port reset register (RCC_IOPRSTR)

Address offset:0x24

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GPIOFRST	Res	Res	GPIOCRST	GPIOBRST	GPIOARST
-	-	-	-	-	-	-	-	-	-	RW	-	-	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	GPIOFRST	RW	0	I/O Port F reset. 0:No effect 1:Port F I/O reset
4:3	Reserved	-	-	Reserved

Bit	Name	R/W	Reset Value	Function
2	GPIOCRST	RW	0	I/O Port C reset. 0:No effect 1:Port C I/O reset
1	GPIOBRST	RW	0	I/O Port B reset. 0:No effect 1:Port B I/O reset
0	GPIOARST	RW	0	I/O Port A reset. 0:No effect 1:Port A I/O reset

8.5.9. AHB peripheral reset register (RCC_AHBRSTR)

Address offset:0x28

Reset value:0x0000 0000

The register is set and cleared by the software. After the software is set, the module maintains reset until the software clears the reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	DIV RST	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	CRC RST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMA RST
-	-	-	RW	-	-	-	-	-	-	-	-	-	-	-	RW

Bit	Name	R/W	Reset Value	Function
31:25	Reserved	-	-	Reserved
24	DIVRST	RW	0	Divider module reset. 0:No effect 1:Divider module reset
23:13	Reserved			Reserved
12	CRCRST	RW	0	CRC reset 0:No effect 1:CRC reset
11:9	Reserved	-	-	Reserved
8:1	Reserved	-	-	Reserved
0	DMARST	RW	0	DMA reset. 0:No effect 1:DMA reset

8.5.10. APB peripheral reset register 1 (RCC_APBRSTR1)

Address offset:0x2C

Reset value:0x0000 0000

The register is set and cleared by the software. After the software is set, the module maintains reset until the software clears the reset.

31	30	29	28	27	26	25	24	23	22	21	20
LPTIM RST	OPA RST	Res	PWR RST	Res	Res	Res	Res	Res	I2C2 RST	I2C1 RST	Res
RW	RW	-	RW	-	-	-	-	-	RW	RW	-
19	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
USART4 RST	USART3 RST	USART2 RST	Res	Res	SPI2 RST	Res	Res	WWDG RST	RTC APB RST	Res	Res
RW	RW	RW	-	-	RW	-	-	RW	RW	-	-
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**				
Res	Res	TIM7 RST	TIM6 RST	Res	Res	TIM3RST	TIM2RST				
-	-	RW	RW	-	-	RW	RW				

Bit	Name	R/W	Reset Value	Function
31	LPTIMRST	RW	0	LPTIM reset 0:No effect 1:The module is reset
30	OPARST	RW	0	OPA reset 0:No effect 1:The module is reset
29	Reserved	-	-	Reserved
28	PWRRST	RW	0	Power interface reset 0:No effect 1:The module is reset
27:23	Reserved	-	-	Reserved
22	I2C2RST	RW	0	I2C2 reset 0:No effect 1:The module is reset
21	I2C1RST	RW	0	I2C1 reset. 0:No effect 1:The module is reset
20	Reserved	-	-	Reserved
19	USART4RST	RW	0	USART4 reset

Bit	Name	R/W	Reset Value	Function
				0:No effect; 1:The module is reset
18	USART3RST	RW	0	USART3 reset 0:No effect 1:The module is reset
17	USART2RST	RW	0	USART2 reset 0:No effect 1:The module is reset
16:15	Reserved	-	-	Reserved
14	SPI2RST	RW	0	SPI2 reset 0:No effect 1:The module is reset
13:12	Reserved	-	-	Reserved
11	WWDGRST	RW	0	WWDG reset 0:No effect 1:The module is reset
10	RTCAPBRST	RW	0	RTC module APB reset. 0:No effect 1:The module is reset
9:6	Reserved	-	-	Reserved
5	TIM7RST	RW	0	TIM7 reset 0:No effect 1:The module is reset
4	TIM6RST	RW	0	TIM6 reset 0:No effect; 1:The module is reset
3:2	Reserved	-	-	Reserved
1	TIM3RST	RW	0	TIM3 reset 0:No effect 1:The module is reset
0	TIM2RST	RW	0	TIM2 reset 0:No effect 1:The module is reset

8.5.11. APB peripheral reset register 2 (RCC_APBRSTR2)

Address offset:0x30

Reset value:0x0000 0000

The register is set and cleared by the software. After the software is set, the module maintains reset until the software clears the reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res								LCD RST	Res	COMP2 RST	COMP1 RST	Res	TIM17 RST	TIM16 RST	TIM15 RST
-								RW	-	RW	RW	-	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIM14 RST	USART1 RST	Res	SPI1 RST	TIM1 RST	MCUDBG RST	ADC RST	Res	Res							SYS CFG RST
RW	RW	-	RW	RW	RW	RW	-	-							RW

Bit	Name	R/W	Reset Value	Function
31:24	Reserved	-	-	Reserved
23	LCDRST	RW	0	LCD reset 0:No effect 1:The module is reset
22	Reserved	-	-	Reserved
21	COMP2RST	RW	0	COMP2 reset 0:No effect 1:The module is reset
20	COMP1RST	RW	0	COMP1 reset 0:No effect 1:The module is reset
19	Reserved	-	-	Reserved
18	TIM17RST	RW	0	TIM17 reset 0:No effect 1:The module is reset
17	TIM16RST	RW	0	TIM16 reset 0:No effect 1:The module is reset
16	TIM15RST	RW	0	TIM15 reset 0:No effect 1:The module is reset
15	TIM14RST	RW	0	TIM14 reset. 0:No effect 1:The module is reset
14	USART1RST	RW	0	USART1 reset 0:No effect 1:The module is reset
13	Reserved	-	-	Reserved
12	SPI1RST	RW	0	SPI1 reset

Bit	Name	R/W	Reset Value	Function
				0:No effect 1:The module is reset
11	TIM1RST	RW	0	TIM1 reset 0:No effect 1:The module is reset
10	MCUDBGRST	RW	0	DBG reset 0:No effect 1:The module is reset
9	ADCRST	RW	0	ADC reset. 0:No effect 1:The module is reset
8:1	Reserved	-	-	Reserved
0	SYSCFGRST	RW	0	SYSCFG reset 0:No effect 1:The module is reset

8.5.12. I/O port clock enable register (RCC_IOPENR)

Address offset:0x34

Reset value:0x0000 0000

This bit is set and cleared by software.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GPIOFEN	Res	Res	GPIOCEN	GPIOBEN	GPIOAEN
-	-	-	-	-	-	-	-	-	-	RW	-	-	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	GPIOFEN	RW	0	I/O Port F clock enable 0:clock disabled 1:clock enabled
4:3	Reserved	-	-	Reserved
2	GPIOCEN	RW	0	I/O Port C clock enable 0:clock disabled 1:clock enabled
1	GPIOBEN	RW	0	I/O Port B clock enable 0:clock disabled

				1:clock enabled
0	GPIOAEN	RW	0	I/O Port A clock enable 0:clock disabled 1:clock enabled

8.5.13. AHB peripheral clock enable register (RCC_AHBENR)

Address offset:0x38

Reset value:0x0000 0300

This bit is set and cleared by software.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	DIVEN	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	CRC EN	Res	Res	SRAM EN	FLASH EN	Res	Res	Res	Res	Res	Res	Res	DMA EN
-	-	-	RW	-	-	RW	RW	-	-	-	-	-	-	-	RW

Bit	Name	R/W	Reset Value	Function
31:25	Reserved	-	-	Reserved
24	DIVEN	RW	0	Divider module clock enable 0:Disabled 1:Enabled
23:13	Reserved	-	-	Reserved
12	CRCEN	RW	0	Enable CRC clock 0:Disabled 1:Enabled
11:10	Reserved	-	-	Reserved
9	SRAMEN	RW	1	In Sleep mode, clock enable control of SRAM 0:The clock is disabled in Sleep mode 1:The clock is enabled in Sleep mode Note:This bit only affects the clock enable of this module in Sleep mode. In Run mode, the clock of this module will not be turned off
8	FLASHEN	RW	1	In Sleep mode, Flash's clock enables control 0:The clock is disabled in Sleep mode 1:The clock is enabled in Sleep mode Note:This bit only affects the clock enable of this module in Sleep mode. In Run mode, the clock of this module will not be turned off
7:1	Reserved	-	-	Reserved

Bit	Name	R/W	Reset Value	Function
0	DMAEN	RW	0	DMA clock enable 0:Disabled 1:Enabled

8.5.14. APB peripheral clock enable register 1 (RCC_APBENR1)

Address offset:0x3C

Reset value:0x0000 0000

This bit is set and cleared by software.

31	30	29	28	27	26	25	24	23	22	21
LPTIMEN	OPAEN	Res	PWREN	Res	Res	Res	Res	Res	I2C2EN	I2C1EN
RW	RW	-	RW	-	-	-	-	-	RW	RW
20	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**
Res	USART4EN	USART3EN	USART2EN	Res	Res	SPI2EN	Res	Res	WWDGEN	TIM-DIV_EN
-	RW	RW	RW	-	-	-	RW	-	-	RW
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Res	Res	Res	Res	TIM7EN	TIM6EN	Res	Res	TIM3EN	TIM2EN	
-	RW	-	-	RW	RW	-	-	RW	RW	

Bit	Name	R/W	Reset Value	Function
31	LPTIMEN	RW	0	Low power timer 1 clock enable 0:Disabled 1:Enabled
30	OPAEN	RW	0	OPA clock enable 0:Disabled 1:Enabled
29	Reserved	-	-	Reserved
28	PWREN	RW	0	Power interface clock enable 0:Disabled 1:Enabled
27:23	Reserved	-	-	Reserved
22	I2C2EN	RW	0	I2C2 enable 0:Disabled 1:Enabled
21	I2C1EN	RW	0	I2C1 enable 0:Disabled 1:Enabled

Bit	Name	R/W	Reset Value	Function
20	Reserved	-	-	Reserved
19	USART4EN	RW	0	USART4 enable 0:Disabled 1:Enabled
18	USART3EN	RW	0	USART3 enable 0:Disabled 1:Enabled
17	USART2EN	RW	0	USART2 enable 0:Disabled 1:Enabled
16:15	Reserved	-	-	Reserved
14	SPI2EN	RW	0	SPI2 enable 0:Disabled 1:Enabled This register is reset and cleared by the hardware.
13:12	Reserved	-	-	Reserved
11	WWDGEN	RW	0	Window Watchdog clock enabled. 0:Disabled 1:Enabled This register is reset and cleared by the hardware.
10	TIMDIV_EN	RW	0	TIMER PCLK frequency control. 0:TIMER PCLK is system PCLK * 2, but the frequency will not exceed HCLK 1:TIMER PCLK is system PCLK * 1
9:6	Reserved	-	-	Reserved
5	TIM7EN	RW	0	TIM7 enable 0:Disabled 1:Enabled
4	TIM6EN	RW	0	TIM6 enable 0:Disabled 1:Enabled
3:2	Reserved	-	-	Reserved
1	TIM3EN	RW	0	TIM3 enable 0:Disabled 1:Enabled
0	TIM2EN	RW	0	TIM2 enable 0:Disabled 1:Enabled

8.5.15. APB peripheral clock enable register 2 (RCC_APBENR2)

Address offset:0x40

Reset value:0x0000 0001

This bit is set and cleared by software.

31	30	29	28	27	26	25	24	23	22	21
Res	Res	Res	Res	Res	Res	Res	Res	LCDEN	Res	COMP2EN
-	-	-	-	-	-	-	-	RW	-	RW
20	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**
COMP1EN	Res	TIM17 EN	TIM16 EN	TIM15 EN	TIM14 EN	USART1E N	Res	SPI1EN	TIM1EN	MCUD-BGEN
RW	-	RW	RW	RW	RW	RW	-	RW	RW	RW
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
ADCEN	Res	Res	Res	Res	Res	Res	Res	Res	SYSCFGEN	
RW	-	-	-	-	-	-	-		RW	

Bit	Name	R/W	Reset Value	Function
31:24	Reserved	-	-	Reserved
23	LCDEN	RW	0	LCD clock enable 0:Disabled 1:Enabled
22	Reserved	-	-	Reserved
21	COMP2EN	RW	0	COMP2 enable 0:Disabled 1:Enabled
20	COMP1EN	RW	0	COMP1 enable 0:Disabled 1:Enabled
19	Reserved	-	-	Reserved
18	TIM17EN	RW	0	TIM17 enable 0:Disabled 1:Enabled
17	TIM16EN	RW	0	TIM16 enable 0:Disabled 1:Enabled
16	TIM15EN	RW	0	TIM15 enable 0:Disabled 1:Enabled
15	TIM14EN	RW	0	TIM14 enable 0:Disabled

Bit	Name	R/W	Reset Value	Function
				1:Enabled
14	USART1EN	RW	0	USART1 enable 0:Disabled 1:Enabled
13	Reserved	-	-	Reserved
12	SPIEN	RW	0	SPI1 enable 0:Disabled 1:Enabled
11	TIM1EN	RW	0	TIM1 enable 0:Disabled 1:Enabled
10	MCUDBGEN	RW	0	MCUDBG clock enable 0:Disabled 1:Enabled
9	ADCEN	RW	0	ADC enable 0:Disabled 1:Enabled
8:1	Reserved	-	-	Reserved
0	SYSCFGEN	RW	1	SYSCFG clock enable 0:Disabled 1:Enabled

8.5.16. Peripherals independent clock configuration register (RCC_CCIPR)

Address offset:0x54

Reset value:0x0000 0000

This bit is set and cleared by software.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LPTIM1SEL [1:0]		Res	Res
-		-	-	-	-	-	-	-	-	-	-	RW	RW	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res		Res		Res	Res	COMP2 SEL	COMP1 SEL	PVD SEL	Res	Res	Res	Res	Res	Res	Res
-		-		-	-	RW	RW	RW	-	-	-	-	-	-	-

Bit	Name	R/W	Reset Value	Function
31:20	Reserved	-	-	Reserved
19:18	LPTIMSEL[1:0]	RW	2'b00	LPTIM internal clock source selection. 00:PCLK

Bit	Name	R/W	Reset Value	Function
				01:LSI 10:No clock 11:LSE Note:When PCLK is selected for the LPTIM internal clock source, the LPTIM_CFGR.PRESC register needs to be configured to divide 2 and above.
17:10	Reserved	-	-	Reserved
9	COMP2SEL	RW	0	COMP2 clock source selection. 0:PCLK 1:LSC (clock selected by RCC_BDCR.LSCOSEL) Note:Configure and select LSC clock before enabling COMP2_FR2. FLTEN.
8	COMP1SEL	RW	0	COMP1 clock source selection. 0:PCLK 1:LSC (clock selected by RCC_BDCR.LSCOSEL) Note:Configure selection clock before enabling COMP1_FR.FLTEN.
7	PVDSEL	RW	0	PVD detect clock source selection. 0:PCLK 1:LSC (clock selected by RCC_BDCR.LSCOSEL) Note:When the clock source is PCLK, PWR_CR1. FLTEN needs to be configured to 0. If FLTEN is enabled, the LSC clock must be selected and configured to select the LSC clock before FLTEN is enabled.
6:0	Reserved	-	-	Reserved

8.5.17. RTC domain control register (RCC_BDCR)

Address offset:0x5C

Reset value:0x0000 0000, reset by POR/BOR

When this register is accessed continuously, 0 ≤ wait state ≤ 3.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res						LSCOSEL	LSCOEN	Res	Res	Res	Res	Res	Res	Res	BDRST
-						RW	RW	RW	-	-	-	-	-	-	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTCEN	Res					RTCSEL [1:0]		Res	LSECSSD	LSECSSON	Res	Res	LSE-BYP	LSERDY	LSEON
RW	-					RW	RW	-	R	RW	-	-	RW	R	RW

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25	LSCOSEL	RW	0	Low speed clock output selection 0:LSI 1:LSE
24	LSCOEN	RW	0	Low speed clock enabled. 0:Disabled 1:Enabled
23:17	Reserved	-	-	Reserved
16	BDRST	RW	0	RTC domain software reset. 0:No effect 1:Reset
15	RTCEN	RW	0	RTC clock enabled. Set and cleared by software. 0:Disabled 1:Enabled
14:10	Reserved	-	-	Reserved
9:8	RTCSEL [1:0]	RW	0	RTC clock source selection. 00:No clock 01:LSE 10:LSI 11:HSE 128 divided frequency Once the RTC clock source is selected, it cannot be changed unless: 1. RTC domain reset to 00 2. Select LSE (LSECSSD = 1) but no LSE
7	Reserved	-	-	Reserved
6	LSECSSD	R	0	CSS failed to detect LSE. Set by hardware to indicate when a failure has been detected by the clock security system (CSS) on the external 32.768 kHz oscillator (LSE). 0:No failure detected on LSE 1:Failure detected on LSE
5	LSECSSON	RW	0	CSS enables LSE clock. 0:Interrupt is disabled 1:Enabled Note:LSECSSON must be enabled after the LSE oscillator is enabled (LSEON=1) and ready (LSERDY=1). Once enabled this bit cannot be disabled, except after an LSE failure detection (LSECSSD=1).

Bit	Name	R/W	Reset Value	Function
4:3	Reserved	-	-	Reserved
2	LSEBYP	RW	0	LSE OSC bypass 0:No bypass, low-speed external clock selects crystal oscillator; 1:Bypass, low-speed external clock select external interface input clock; This bit can be written only when the external 32.768 kHz oscillator is disabled (LSEON=0 and LSERDY=0).
1	LSERDY	R	0	LSE OSC ready. Hardware configuration:A value of 1 in this bit indicates the LSE clock is ready.
0	LSEON	RW	0	LSE oscillator enable 0:Disabled 1:Enabled

8.5.18. RCC control/status register (RCC_CSR)

Address offset:0x60

Reset value:0x0800 0000

The reset flag bit in this register can only be reset by power reset, and the rest is reset by the system.

When this register is accessed continuously, 0 ≤ wait state ≤ 3.

31	30	29	28	27	26	25	24	23	22	21
Res	WWDG RSTF	IWDG RSTF	SFT RSTF	PWR RSTF	PIN RSTF	OBL RSTF	Res	RMVF	Res	Res
-	R	R	R	R	R	R	-	RW	-	-
20	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
-	-	-	-	-	-	-	-	-	-	
10	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	PINRST _FLTDIS	Res	Res	Res	Res	Res	Res	LSI RDY	LSION
-	-	RW	-	-	-	-	-	-	R	RW

Bit	Name	R/W	Reset Value	Function
31	Reserved	-	-	Reserved
30	WWDGRSTF	R	0	Window watchdog reset flag. Cleared by writing 1 to the RMVF bit.
29	IWDGRSTF	R	0	Independent watchdog reset flag Cleared by writing 1 to the RMVF bit.
28	SFTRSTF	R	0	Software reset flag

Bit	Name	R/W	Reset Value	Function
				Cleared by writing 1 to the RMVF bit.
27	PWRRSTF	R	1	BOR/POR/PDR reset flag Cleared by writing to the RMVF bit.
26	PINRSTF	R	0	PIN reset flag Cleared by writing 1 to the RMVF bit.
25	OBLRSTF	R	0	Option byte loader reset flag Cleared by writing 1 to the RMVF bit.
24	Reserved	-	-	Reserved
23	RMVF	RW	0	Set by software to clear the reset flags of [30:25].
22:9	Reserved	-	-	Reserved
8	PINRST_FLTDIS	RW	0	NRST filter width 40 as disabled 0:HSI_10M enabled, and filtering 40 μs width function enabled 1:The filtering function is disabled, and the system reset is generated by synchronization to the system clock
7:2	Reserved	-	-	Reserved
1	LSIRDY	R	0	LSI oscillator stable flag 0:LSI oscillator unstable 1:LSI oscillator stable
0	LSION	RW	0	LSI oscillator enable 0:Disabled 1:Enabled When the hardware turns on the analog LSI: 1. Hardware IWDG enabled 2. LSECSS enabled

Puya

9. General-purpose I/Os (GPIO)

9.1. Introduction

The GPIO contains PA [15:0], PB [15:0], PC [15:0] and PF [9:0], and each GPIO port has:

- Four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR)
- Two 32-bit data registers (GPIOx_IDR and GPIOx_ODR)
- A 32-bit set/reset register (GPIOx_BSRR)
- A 32-bit locking register (GPIOx_LCKR)
- Two multiplexing function selection registers (GPIOx_AFRH and GPIOx_AFRL).
- A 32-bit reset register (GPIOx_BRR)

9.2. GPIO functional description

- Register supports Fast IO/AHB bus read and write
- Output status:push-pull or open-drain + pull-up/pull-down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Speed selection for each I/O
- Input status:floating, pull-up/pull-down, analog
- Input data to input data register (GPIOx_IDR) or peripheral (alternate function input)
- Set/Reset register (GPIOx_BSRR) to allow bit write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O port configurations
- Analog function
- Alternate function selection registers (at most 16 AFs possible per I/O)
- Ability to flip quickly in a single cycle
- Highly flexible I/O multiplexing function, allowing the I/O port to function as GPIO or as various peripheral interface functions

9.3. GPIO functional description

Each port bit of the general-purpose I/O (GPIO) ports can be individually configured by software in several modes:

- Input floating
- Input pull-up
- Input-pull-down
- Analog
- Output open-drain with pull-up or pull-down capability
- Output push-pull with pull-up or pull-down capability
- Push-pull with pull-up or pull-down multiplexing function
- Alternate function open-drain with pull-up or pull-down capability

Puya Confidential

Each I/O port bit is freely programmable, however the I/O port registers have to be accessed as words, half-words or bytes. The purpose of the GPIOx_BSRR register is to allow atomic read/modify accesses to any of the GPIOx_ODR registers. In this way, there is no risk of an IRQ occurring between the read and the modify access.

Figure below gives the possible port bit configurations.

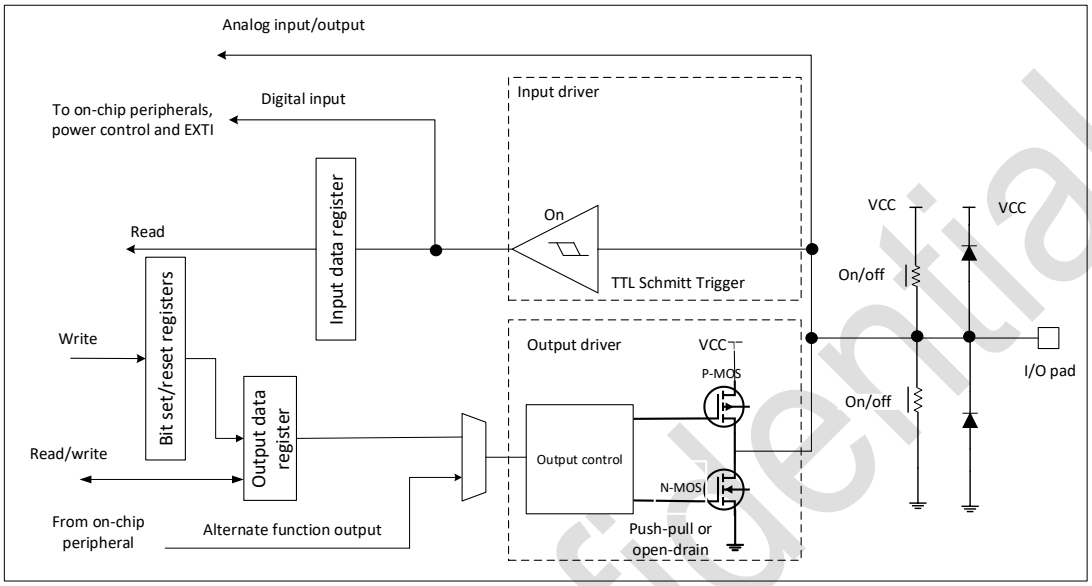


Figure 9-1 Basic structure of an I/O port bit

9.3.1. General-purpose I/Os (GPIO)

During and just after reset, the alternate functions are not active and most of the I/O ports are configured in analog mode.

The debug pins are in AF pull-up/pull-down after reset:

- PA14-SWCLK:put in pull-down mode
- PA13-SWDIO:put in pull-up mode

The Boot pin is put into input mode by default, pull-down mode

- PF8-Boot:put into pull-down mode

When the pin is configured as output, the value written to the output data register (GPIOx_ODR) is output on the I/O pin. It is possible to use the output driver in push-pull mode or open-drain mode (only the low level is driven, high level is HI-Z).

The input data register (GPIOx_IDR) captures the data present on the I/O pin at every AHB clock cycle.

All GPIO pins have weak internal pull-up and pull-down resistors, which can be activated ornot depending on the value in the GPIOx_PUPDR register.

9.3.2. I/O pin alternate function multiplexer and mapping

The device I/O pins are connected to on-board peripherals/modules through a multiplexer that allows only one peripheral alternate function (AF) connected to an I/O pin at a time. In this way, there can be no conflict between peripherals available on the same I/O pin.

Puya Confidential

The multiplexer on each I/O port has up to 16 multiplexing function inputs (AF0 to AF15), which can be configured through the registers GPIOx_AFRL (for pin 0 to 7) and GPIOx_AFRH (for pin 8 to 15).

- After reset the multiplexer selection is alternate function 0 (AF0). The I/Os are configured in alternate function mode through GPIOx_MODER register
- Please refer to the datasheet for the distribution of multiplexing functions of each pin

In addition to this flexible I/O multiplexing architecture, each peripheral has alternate functions mapped onto different I/O pins to optimize the number of peripherals available in smaller packages.

To use an I/O in a given configuration, the user has to proceed as follows:

- Debug function:after each device reset these pins are assigned as alternate function pins immediately usable by the debugger host
- GPIO:configure the desired I/O as output, input or analog in the GPIOx_MODER register.
- Peripheral alternate function:
  - The I/O corresponding to the register GPIOx_AFRL or GPIOx_AFRH configuration is multiplexing function x (x = 0...15)
  - Select the type, pull-up/pull-down and output speed via the GPIOx_OTYPER, GPIOx_PUPDR and GPIOx_OSPEEDER registers, respectively.
  - Configure the desired I/O as an alternate function in the GPIOx_MODER register.
  - Bus read and write function:In addition to supporting Fast IO read and write GPIOx registers, the GPIO module also supports read and write registers through the AHB bus. The register access method is selected by the GPIO_AHB_SEL bit of the SYSCFG module. When the GPIO_AHB_SEL bit is 0, the GPIOx register can only be accessed through Fast IO; When GPIO_AHB_SEL is 1, only the GPIOx register is accessed through the AHB bus.
  - The way the AHB bus accesses the GPIO register, in addition to supporting the CPU, also supports DMA, that is, DMA can directly access the GPIO register through the AHB bus.
- Additional functions:
  - ADC connection could be enabled in ADC registers regardless the configured GPIO mode. It is recommended to configure GPIO in analog mode in the GPIOx_MODER register when ADC or COMP is used.
  - For the additional functions like oscillators, configure the required function in the related PWR and RCC registers. These functions have priority over the configuration in the standard GPIO registers.

9.3.3. I/O port control registers

Each of the GPIO ports has four 32-bit memory-mapped control registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR) to configure up to 16 I/Os. The GPIOx_MODER register is used to select the I/O mode (input, output, AF, analog). The GPIOx_OTYPER and GPIOx_OSPEEDR registers are used to select the output type (push- pull or open-drain) and speed. The GPIOx_PUPDR register is used to select the pull- up/pull-down whatever the I/O direction.

Puya Confidential

9.3.4. I/O port data registers

Each GPIO has two 16-bit memory-mapped data registers:input and output data registers (GPIOx_IDR and GPIOx_ODR). GPIOx_ODR stores the data to be output, it is read/write accessible. The data input through the I/O are stored into the input data register (GPIOx_IDR), a read-only register.

9.3.5. I/O data bitwise handling

The bit set reset register (GPIOx_BSRR) is a 32-bit register which allows the application to set and reset each individual bit in the output data register (GPIOx_ODR). The bit set reset register has twice the size of GPIOx_ODR.

To each bit in GPIOx_ODR, correspond two control bits in GPIOx_BSRR:BS(i) and BR(i). When written to 1, bit BS(i) sets the corresponding ODR(i) bit. When written to 1, bit BR(i) resets the ODR(i) corresponding bit.

Writing any bit to 0 in GPIOx_BSRR does not have any effect on the corresponding bit in GPIOx_ODR. If there is an attempt to both set and reset a bit in GPIOx_BSRR, the set action takes priority.

Using the GPIOx_BSRR register to change the values of individual bits in GPIOx_ODR is a one-shot effect that does not lock the GPIOx_ODR bits. The GPIOx_ODR bits can always be accessed directly. The GPIOx_BSRR register provides a way of performing atomic bitwise handling.

There is no need for the software to disable interrupts when programming the GPIOx_ODR at bit level:it is possible to modify one or more bits in a single atomic AHB write access.

It is possible to freeze the GPIO control registers by applying a specific write sequence to the GPIOx_LCKR register. The frozen registers are GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL and GPIOx_AFRH.

To write the GPIOx_LCKR register, a specific write / read sequence has to be applied. When the right LOCK sequence is applied to bit 16 in this register, the value of LCKR [15:0] is used to lock the configuration of the I/Os (during the write sequence the LCKR [15:0] value must be the same). When the LOCK sequence has been applied to a port bit, the value of the port bit can no longer be modified until the next MCU reset or peripheral reset. Each GPIOx_LCKR bit freezes the corresponding bit in the control registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL and GPIOx_AFRH.

The LOCK sequence can only be performed using a word (32-bit long) access to the GPIOx_LCKR register due to the fact that GPIOx_LCKR bit 16 has to be set at the same time as the [15:0] bits.

9.3.6. I/O alternate function input/output

Two registers are provided to select one of the alternate function inputs/outputs available for each I/O. With these registers, the user can connect an alternate function to some other pin as required by the application.

This means that a number of possible peripheral functions are multiplexed on each GPIO using the GPIOx_AFRL and GPIOx_AFRH alternate function registers. The application can thus select any one of the possible functions for each I/O. The AF selection signal being common to the alternate function

Puya Confidential

input and alternate function output, a single channel is selected for the alternate function input/output of a given I/O.

9.3.7. External interrupt/wakeup lines

All ports have external interrupt capability. To use external interrupt lines, the given pin must not be configured in analog mode or being used as oscillator pin, so the input trigger is kept enabled.

9.3.8. Input configuration

When the I/O port is programmed as input:

- The output buffer is disabled
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx_PUPDR register
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- A read access to the input data register provides the I/O state

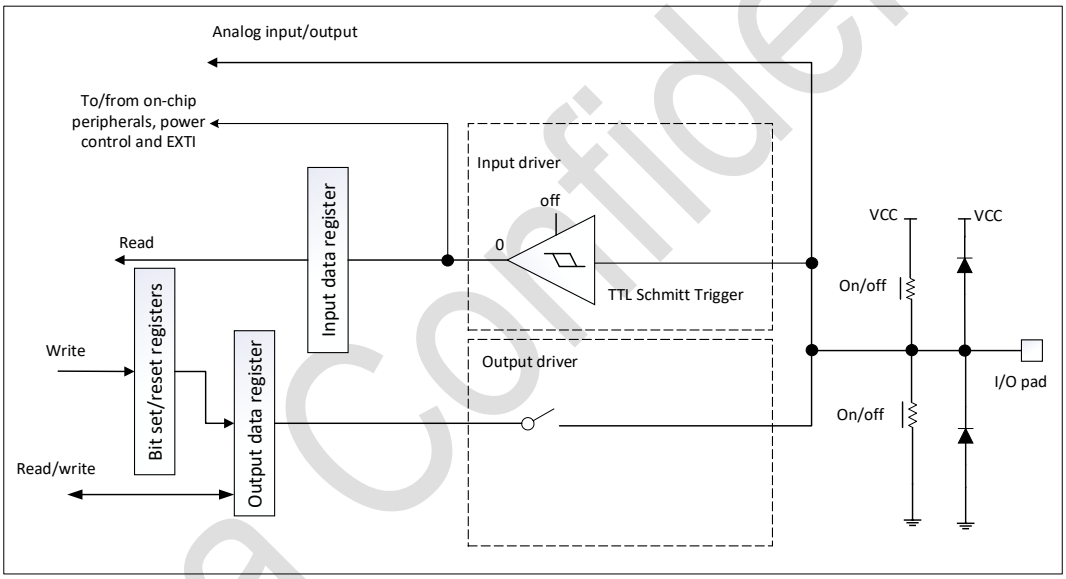


Figure 9-2 Input floating/pull-up/pull-down configuration

9.3.9. Output configuration

When the I/O port is programmed as output:

- The output buffer is disabled
  - Open drain mode:A “0” in the Output register activates the N-MOS whereas a “1” in the output register leaves the port in Hi-Z (the P-MOS is never activated).
  - Push-pull mode:A “0” in the Output register activates the N-MOS whereas a “1” in the Output register activates the P-MOS.
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx_PUPDR register
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- A read access to the input data register provides the I/O state

Puya Confidential

- A read access to the output data register gets the last written value

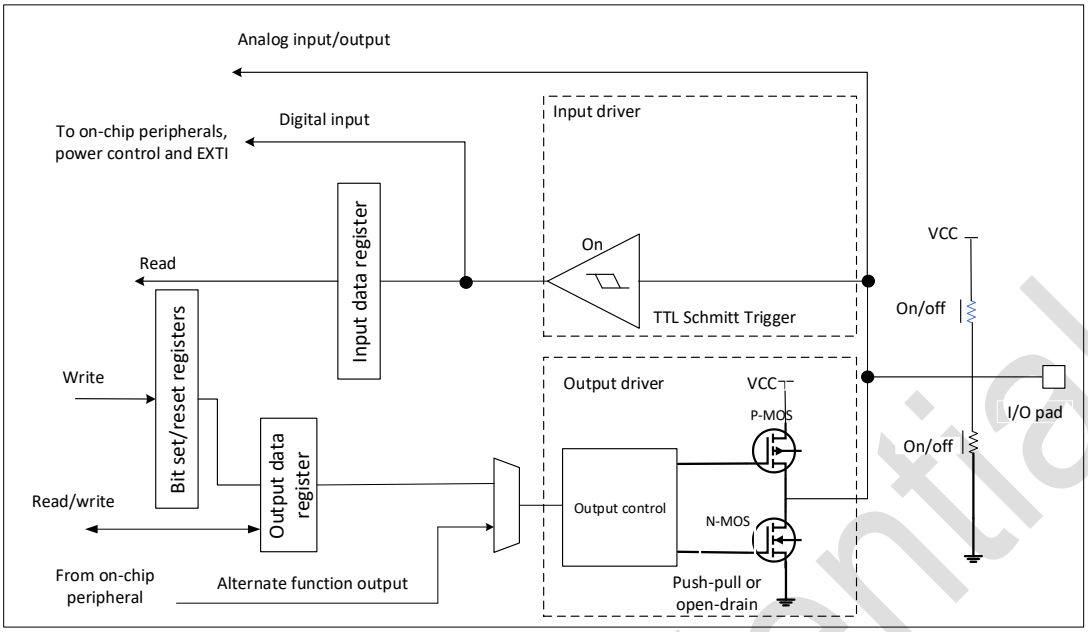


Figure 9-3 Output configuration

9.3.10. Alternate function configuration

When the I/O port is programmed as alternate function:

- The output buffer can be configured in open-drain or push-pull mode
- The output buffer is driven by the signals coming from the internal peripheral (alternate function output)
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx_PUPDR register
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- A read access to the input data register provides the I/O state

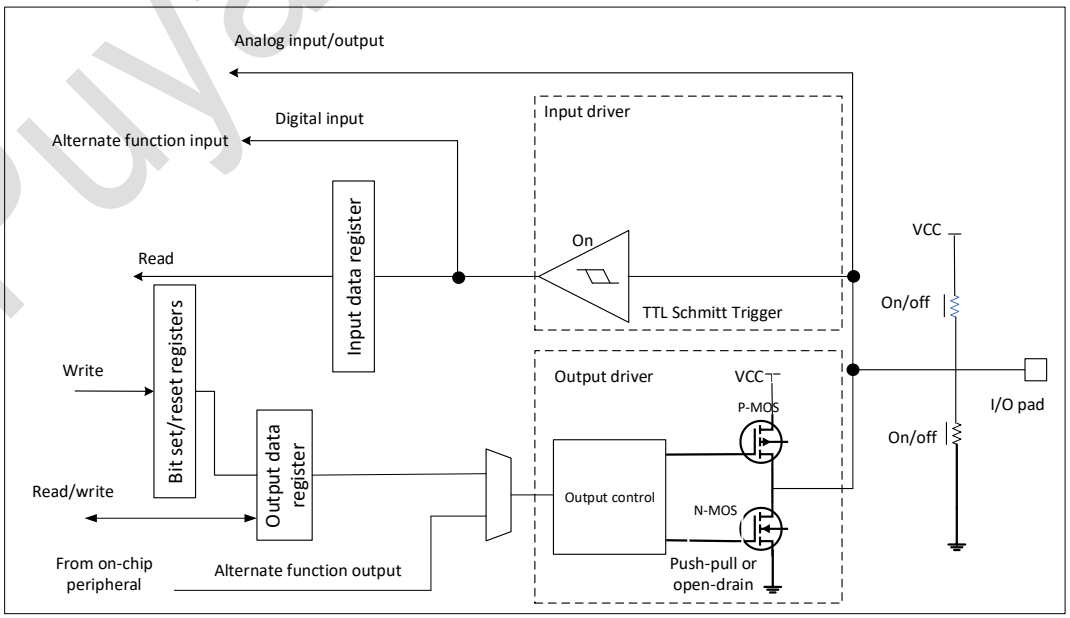


Figure 9-4 Alternate function configuration

Puya Confidential

9.3.11. Analog configuration

When the I/O port is programmed as analog configuration:

- The output buffer is disabled;
- The Schmitt trigger input is deactivated, providing zero consumption for every analog value of the I/O pin. The output of the Schmitt trigger is forced to a constant value (0);
- Weak pull-up and pull-down resistors are disabled (software setting is required);
- Read access to the input data register gets the value “0”.

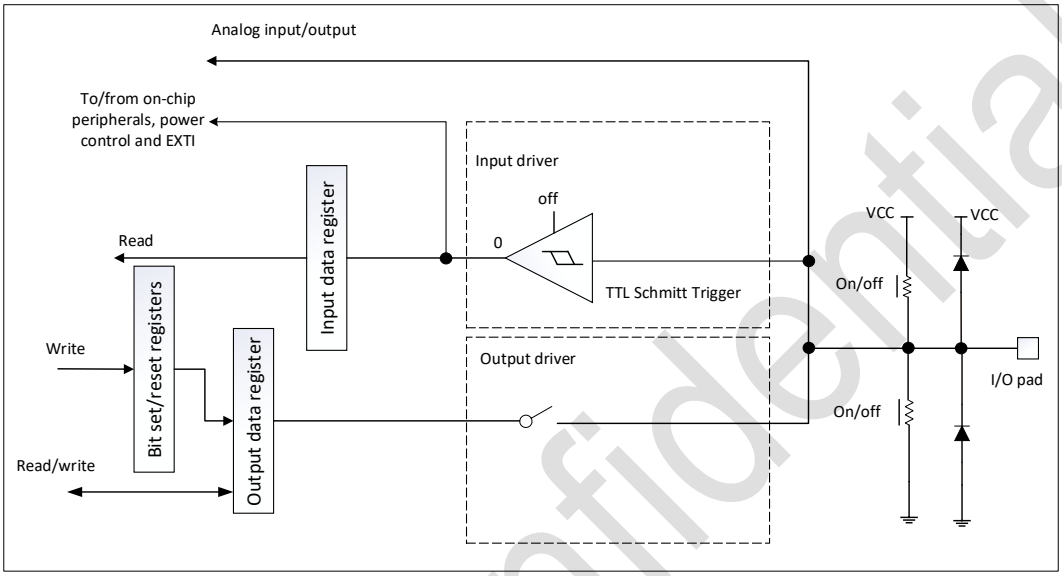


Figure 9-5 High impedance-analog configuration

9.3.12. Using HSE/LSE pins as GPIO

When the HSE or LSE oscillator is turned off (the default after reset), the corresponding pin can be used as a normal GPIO.

When the HSE or LSE oscillator is turned on (HSEON or LSEON in the RCC_CSR register is set), the software needs to configure the corresponding port as an analog port.

When the oscillator is configured in user external clock mode, only OSC_IN or OSC32_IN is reserved for the clock input, while the OSC_OUT or OSC32_OUT pin can still be used as the normal GPIO.

9.4. GPIO registers

All GPIO-related registers are available for word, half-word, and byte writes.

9.4.1. GPIO port mode register (GPIOx_MODER) (x = A, B, C, F)

Address offset:0x00

Reset value:

- 0xEBFF FFFF (Port A)
- 0xFFFF FFFF (port B)
- 0xFFFF FFFF (port C)
- 0x000C FFFF (Port F)

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15 [1:0]		MODE14 [1:0]		MODE13 [1:0]		MODE12 [1:0]		MODE11 [1:0]		MODE10 [1:0]		MODE9 [1:0]		MODE8 [1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MODE7 [1:0]		MODE6 [1:0]		MODE5 [1:0]		MODE4 [1:0]		MODE3 [1:0]		MODE2 [1:0]		MODE1 [1:0]		MODE0 [1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MODEy [1:0]	RW	0xEBFFFFFF (PA) 0xFFFFFFF (PB) 0xFFFFFFF (PC) 0x000CFFFF (PF)	y = 15..0 These bits are written by software to configure the I/O mode. 00:Input mode 01:General purpose output mode 10:Alternate function mode 11:Reset state

9.4.2. GPIO port output type register (GPIOx_OTYPER) (x = A, B, C, F)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	OT [15:0]	RW	0	These bits are written by software to configure the I/O output type. 0:push-pull output (reset state) 1:Output open-drain

9.4.3. GPIO port output speed register (GPIOx_OSPEEDR) (x = A, B, C, F)

Address offset:0x08

Reset value:0x0C00 0000 (port A)

Reset value:0x0000 0000 (other ports)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15		OSPEED14		OSPEED13		OSPEED12		OSPEED11		OSPEED10		OSPEED9		OSPEED8	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEED7		OSPEED6		OSPEED5		OSPEED4		OSPEED3		OSPEED2		OSPEED1		OSPEED0	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	OSPEEDy [1:0]	RW	0x0C000000 (PA) 0x0 (Others)	Y = 15.. 0 These bits are written by software to configure the I/O output speed. 00:Low speed 01:Medium speed 10:high speed 11:Very high speed

9.4.4. GPIO port pull-down register (GPIOx_PUPDR) (x = A, B, C, F)

Address offset:0x0C

Reset value:

- 0x2400 0000 (Port A)
- 0x0000 0000 (Port B, C)
- 0x0002 0000 (Port F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15 [1:0]		PUPD14 [1:0]		PUPD13 [1:0]		PUPD12 [1:0]		PUPD11 [1:0]		PUPD10 [1:0]		PUPD9 [1:0]		PUPD8 [1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD7 [1:0]		PUPD6 [1:0]		PUPD5 [1:0]		PUPD4 [1:0]		PUPD3 [1:0]		PUPD2 [1:0]		PUPD1 [1:0]		PUPD0 [1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PUPDy [1:0]	RW	0x24000000 (PA) 0x0 (PB) 0x0 (PC) 0x20000 (PF)	Y = 15.. 0 These bits are written by software to configure the I/O pull-up or pull-down 00:No pull-up, pull-down 01:pull-up 10:pull down 11:Reserved

9.4.5. GPIO port input data register (GPIOx_IDR) (x = A, B, C, F)

Address offset:0x10

Reset value:0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

Puya Confidential

-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	IDy	R	-	y = 15..0 These bits are read-only. They contain the input value of the corresponding I/O port.

9.4.6. GPIO port output data register (GPIOx_ODR) (x = A, B, C, F)

Address offset:0x14

Reset value:0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	ODy	RW	0	y = 15..0 These bits can be read and written by software. Note:For GPIOx_BSRR or GPIOx_BRR (x = A, B, C, F), each ODR bit can be set/cleared independently.

9.4.7. GPIO port bit set/reset register (GPIOx_BSRR) (x = A, B, C, F)

Address offset:0x18

Reset value:0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:16	BRy	W	0	y = 15..0

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				These bits are write-only. A read to these bits returns the value 0. 0:No effect on the corresponding ODRy bit 1:Clear the corresponding ODRy bit Note:If both BSy and BRy are set, BSy has priority.
15:0	BSy	W	0	y = 15..0 These bits are write-only. A read to these bits returns the value 0. 0:No effect on the corresponding ODRy bit 1:Set the corresponding ODRy bit

9.4.8. GPIO port configuration lock register (GPIOx_LCKR) (x = A, B, C, F)

This register is used to lock the configuration of the port bits when a correct write sequence is applied to bit 16 (LCKK). LCKy [15:0] is used to lock the configuration of the GPIO port and cannot be changed during a specified write operation. When the LOCK sequence has been applied on a port bit, the value of this port bit can no longer be modified until the next system reset.

Note:A specific write sequence is used to write to the GPIOx_LCKR register. Only word access (32-bit long) is allowed during this locking sequence.

Each lock bit freezes a specific configuration register (control and alternate function registers).

Address offset:0x1C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LCKK
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:17	Reserved	-	-	Reserved
16	LCKK	RW	0	This bit can be read any time. It can only be modified using the lock key write sequence. 0:Port configuration lock key not active 1:Port configuration lock key active. The GPIOx_LCKR register is locked until the next system reset. Write timing of lock key: WR LCKR [16] = '1' + LCKR [15:0] WR LCKR [16] = '0' + LCKR [15:0] WR LCKR [16] = '1' + LCKR [15:0]

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				RD LCKR RD LCKR [16] = ‘1’ (this read operation is optional but it confirms that the lock is active) Note:During the LOCK key write sequence, the value of LCK [15:0] must not change. Any error in the lock sequence aborts the lock. After the first lock sequence on any bit of the port, any read access on the LCKK bit returns ‘1’ until the next MCU reset or peripheral reset.
15:0	LCKy	RW	0	y = 15..0 These bits are read/write but can only be written when the LCKK bit is '0'. 0:Port configuration not locked 1:Port configuration locked

9.4.9. GPIO alternate function register (low) (GPIOx_AFRL) (x = A, B, C, F)

Address offset:0x20

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7 [3:0]				AFSEL6 [3:0]				AFSEL5 [3:0]				AFSEL4 [3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AFSEL3 [3:0]				AFSEL2 [3:0]				AFSEL1 [3:0]				AFSEL0 [3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	AFSELy [3:0] (y = 7 to 0)	RW	0	These bits are written by software to configure alternate function I/Os. AFSELy selection: 0000:AF0 1000:AF8 0001:AF1 1001:AF9 0010:AF2 1010:AF10 0011:AF3 1011:AF11 0100:AF4 1100:AF12 0101:AF5 1101:AF13 0110:AF6 1110:AF14 0111:AF7 1111:AF15

9.4.10. GPIO alternate function register (high) (GPIOx_AFRH) (x = A, B, C, F)

Address offset:0x24

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15 [3:0]				AFSEL14 [3:0]				AFSEL13 [3:0]				AFSEL12 [3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AFSEL11 [3:0]				AFSEL10 [3:0]				AFSEL9 [3:0]				AFSEL8 [3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	AFSELy [3:0] (y = 8 to 15)	RW	0	These bits are written by software to configure alternate function I/Os. AFSELy selection: 0000:AF0 1000:AF8 0001:AF1 1001:AF9 0010:AF2 1010:AF10 0011:AF3 1011:AF11 0100:AF4 1100:AF12 0101:AF5 1101:AF13 0110:AF6 1110:AF14 0111:AF7 1111:AF15

9.4.11. GPIO port reset register (GPIOx_BRR) (x = A, B, C, F)

Address offset:0x28

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**2**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	BRy	W	0	y = 15..0 These bits are write-only. A read to these bits returns the value 0. 0:No effect on the corresponding ODRy bit 1:Clear the corresponding ODRy bit

Puya Confidential

10. System configuration controller (SYSCFG)

The devices feature a set of configuration registers. The main purposes of the system configuration controller are the following:

- Enable and disable I²C type IO filter
- Mapping the initial program area according to different boot modes
- DMA peripheral channel selection control
- TIMx cascade control.
- Enable and disable PVD Lock
- Enable and disable Cortex-M0+ LOCKUP
- Enable and disable Noise filter for all GPIOs

10.1. SYSCFG registers

10.1.1. SYSCFG configuration register 1 (SYSCFG_CFGR1)

The MEM_MODE bit of this register is used to configure the type of memory address 0x0000 0000 access. These bits are used to select the physical remap by software and so, bypass the hardware Boot selection. After power-on or PIN reset, these bits take the value selected by the actual boot mode configuration.

Address offset:0x00

Reset value:0x0000 000x (X is the memory mode selected by the actual boot mode configuration)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res							GPIO_AHB_SEL	Res					ETR_SRC_TIM3[2:0]		
-							RW	-					RW		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	ETR_SRC_TIM2[2:0]			Res	ETR_SRC_TIM1[2:0]			TIM3_IC1_SRC		TIM2_IC4_SRC		TIM1_IC1_SRC		MEM_MODE	
-	RW			-	RW			RW		RW		RW		RW	

Bit	Name	R/W	Reset Value	Function
31:25	Reserved	-	-	Reserved
24	GPIO_AHB_SEL	RW	0	CPU FASTIO or AHB bus access GPIO register control. 0:FASTIO bus access; 1:AHB bus access
23:19	Reserved	-	-	Reserved
18:16	ETR_SRC_TIM3[2:0]	RW	0	TIMER3 ETR input source selection. 3 'b000:ETR derived from GPIO 3 'b001:ETR derived from COMP1 3 'b010:ETR derived from COMP2

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				3 'b011:ETR derived from ADC 3 'b100:Reserved
15	Reserved	-	-	Reserved
14:12	ETR_SRC_TIM2[2:0]	RW	0	TIMER2 ETR input source selection. 3 'b000:ETR derived from GPIO 3 'b001:ETR derived from COMP1 3 'b010:ETR derived from COMP2 3 'b011:ETR derived from ADC 3 'b100:Reserved
11	Reserved	-	-	Reserved
10:8	ETR_SRC_TIM1[2:0]	RW	0	TIMER1 ETR input source selection. 3 'b000:ETR derived from GPIO 3 'b001:ETR derived from COMP1 3 'b010:ETR derived from COMP2 3 'b011:ETR derived from ADC 3 'b100:Reserved
7:6	TIM3_IC1_SRC	RW	0	TIM13 CH1 input source 00:from TIM3_CH1 IO 01:from COMP1 10:from COMP2 11:Reserved
5:4	TIM2_IC4_SRC	RW	0	TIM2 CH4 Source of input 00:from TIM2_CH4 IO; 01:from COMP1 10:from COMP2 11:Reserved
3:2	TIM1_IC1_SRC	RW	0	TIM1 CH1 input source 00:from TIM1_CH1 IO; 01:from COMP1 10:from COMP2 11:Reserved
1:0	MEM_MODE	RW	x	Memory mapping selection bits x0:Main flash memory mapped at 0x0000 0000 01:System flash memory mapped at 0x0000 0000 11:Embedded SRAM mapped at 0x0000 0000

Puya Confidential

10.1.2. SYSCFG configuration register 2 (SYSCFG_CFGR2)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-
20	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**
COMP2_Ocref_CLR_TIM3	COMP2_Ocref_CLR_TIM2	COMP2_Ocref_CLR_TIM1	COMP1_Ocref_CLR_TIM3	COMP1_Ocref_CLR_TIM2	COMP1_Ocref_CLR_TIM1	Res	COMP2_BRK_TIM17	COMP1_BRK_TIM17	Res	COMP2_BRK_TIM16
RW	RW	RW	RW	RW	RW	-	RW	RW	-	RW
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
COMP1_BRK_TIM16	Res	COMP2_BRK_TIM15	COMP1_BRK_TIM15	Res	COMP2_BRK_TIM1	COMP1_BRK_TIM1	PVD_LOCK	Res	LOCKUP_LOCK	
RW	-	RW	RW	-	RW	RW	RW	-	RW	

Bit	Name	R/W	Reset Value	Function
31:21	Reserved	-	-	Reserved
20	COMP2_Ocref_CLR_TIM3	RW	0	1:COMP2 output as TIM3 ocref_clr input 0:COMP2 output is not input as TIM3 ocref_clr
19	COMP2_Ocref_CLR_TIM2	RW	0	1:COMP2 output as TIM2 ocref_clr input 0:COMP2 output is not input as TIM2 ocref_clr
18	COMP2_Ocref_CLR_TIM1	RW	0	1:COMP2 output as TIM1 ocref_clr input 0:COMP2 output is not input as TIM1 ocref_clr
17	COMP1_Ocref_CLR_TIM3	RW	0	1:COMP1 output as TIM3 ocref_clr input 0:COMP1 output is not input as TIM3 ocref_clr
16	COMP1_Ocref_CLR_TIM2	RW	0	1:COMP1 output as TIM2 ocref_clr input 0:COMP1 output is not input as TIM2 ocref_clr
15	COMP1_Ocref_CLR_TIM1	RW	0	1:COMP1 output as TIM1 ocref_clr input 0:COMP1 output is not input as TIM1 ocref_clr
14	Reserved	-	-	Reserved
13	COMP2_ BRK_TIM17	RW	0	COMP2 is enabled as a TIM17 break input 0:COMP2 output is not connected to TIM17 break input 1:COMP2 output as TIM17 break input
12	COMP1_BRK_TIM17	RW	0	COMP1 is enabled as a TIM17 break input

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:COMP1 output is not connected to TIM17 break input 1:COMP1 output as TIM17 break input
11	Reserved	-	-	Reserved
10	COMP2_ BRK_TIM16	RW	0	COMP2 is enabled as a TIM16 break input. 0:COMP2 output is not connected to TIM16 break input 1:COMP2 output as TIM16 break input
9	COMP1_BRK_TIM16	RW	0	COMP1 is enabled as a TIM16 break input. 0:COMP1 output is not connected to TIM16 break input; 1:COMP1 output as TIM16 break input;
8	Reserved	-	-	Reserved
7	COMP2_ BRK_TIM15	RW	0	COMP2 is enabled as a TIM15 break input. 0:COMP2 output is not connected to TIM15 break input 1:COMP2 output as TIM15 break input
6	COMP1_BRK_TIM15	RW	0	COMP1 is enabled as a TIM15 break input. 0:COMP1 output is not connected to TIM15 break input 1:COMP1 output as TIM15 break input
5	Reserved	-	-	Reserved
4	COMP2_ BRK_TIM1	RW	0	COMP2 is enabled as a TIM1 break input. 0:The COMP2 output is not connected to the TIM1 break input 1:COMP2 output as TIM1 break input
3	COMP1_BRK_TIM1	RW	0	COMP1 is enabled as a TIM1 break input. 0:The COMP1 output is not connected to the TIM1 break input 1:COMP1 output as TIM1 break input
2	PVD_LOCK	RW	0	PVD lock enable bit. 0:PVD interrupt disconnected from TIM1/15/16/17 break input. PVDE and PLS [2:0] bits can be programmed by the application 1:PVD interrupt connected to TIM1/15/16/17 break input, PVDE and PLS are read-only
1	Reserved	-	-	Reserved
0	LOCKUP_LOCK	RW	0	Cortex-M0+ LOCKUP bit enable bit

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Cortex-M0 + LOCKUP bit disconnected from TIM1/15/16/17 break input 1:Cortex-M0 + LOCKUP bit connected to TIM1/15/16/17 break input

10.1.3. SYSCFG configuration register 3 (SYSCFG_CFGR3)

Address offset:0x08

Reset value:0x3F3F 3F3F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res		DMA4_MAP						Res		DMA3_MAP					
-		RW						-		RW					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res		DMA2_MAP						Res		DMA1_MAP					
-		RW						-		RW					

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29:24	DMA4_MAP	RW	0x3F	DMA1 channel 4 mapping. See DMA1_MAP description.
23:22	Reserved	Reserved	-	Reserved
21:16	DMA3_MAP	RW	0x3F	DMA1 channel 3 mapping. See DMA1_MAP description.
15:14	Reserved	Reserved	-	Reserved
13:8	DMA2_MAP	RW	0x3F	DMA1 channel 2 mapping. See DMA1_MAP description.
7:6	Reserved	Reserved	-	Reserved
5:0	DMA1_MAP	RW	0x3F	DMA1 channel 1 mapping. 000000:ADC1 000001:Reserved 000010:Reserved 000011:SPI1_RD 000100:SPI1_WR 000101:SPI2_RD 000110:SPI2_WR 000111:USART1_RD 001000:USART1_WR 001001:USART2_RD 001010:USART2_WR 001011:USART3_RD

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				001100:USART3_WR 001101:USART4_RD 001110:USART4_WR 001111:I2C1_RD 010000:I2C1_WR 010001:I2C2_RD 010010:I2C2_WR 010011:TIM1_CH1 010100:TIM1_CH2 010101:TIM1_CH3 010110:TIM1_CH4 010111:TIM1_COM 011000:TIM1_TRG 011001:TIM1_UP 011010:TIM2_CH1 011011:TIM2_CH2 011100:TIM2_CH3 011101:TIM2_CH4 011110:TIM2_UP 011111:TIM2_TRG 100000:TIM3_CH1 100001:TIM3 _CH2 100010:TIM3_CH3 100011:TIM3_CH4 100100:TIM3_UP 100101:TIM3_TRG 100110:TIM6_UP 100111:TIM7_UP 101000:TIM15_CH1 101001:TIM15_CH2 101010:TIM15_UP 101011:TIM15_TRG 101100:TIM15_COM 101101:TIM16_CH1 101110:TIM16_UP 101111:TIM17_CH1 110000:TIM17_UP 110001:Reserved 110010:LCD

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Others:Reserved

10.1.4. SYSCFG configuration register 4 (SYSCFG_CFGR4)

Address offset:0x0C

Reset value:0x003F 3F3F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res										DMA7_MAP					
-										RW					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res		DMA6_MAP						Res		DMA5_MAP					
-		RW						-		RW					

Bit	Name	R/W	Reset Value	Function
31:22	Reserved	-	-	Reserved
21:16	DMA7_MAP	RW	0x3F	DMA1 channel 7 mapping. See DMA1_MAP description.
15:14	Reserved	-	-	Reserved
13:8	DMA6_MAP	RW	0x3F	DMA1 channel 6 mapping. See DMA1_MAP description.
7:6	Reserved	-	-	Reserved
5:0	DMA5_MAP	RW	0x3F	DMA1 channel 5 mapping. See DMA1_MAP description.

10.1.5. GPIOA filter enable (PA_ENS)

Address offset:0x10

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA_ENS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PA_ENS[x]	RW	0x0000	PORTA IO filtering enabled. 0:Filter disabled 1:Filter enabled

Puya Confidential

10.1.6. GPIOB filter enable (PB_ENS)

Address offset:0x14

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PB_ENS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PB_ENS[x]	RW	0x0000	PORTB IO filtering enabled. 0:Filter disabled 1:Filter enabled

10.1.7. GPIOC filter enable (PC_ENS)

Address offset:0x18

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC_ENS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PC_ENS[x]	RW	0x0000	PORTC IO filtering enabled. 0:Filter disabled 1:Filter enabled

10.1.8. GPIOF filter enable (PF_ENS)

Address offset:0x1C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						PF_ENS[9:0]									
-						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9:0	PF_ENS[x]	RW	0x000	PORTF IO filtering enabled. 0:Filter disabled 1:Filter enabled

10.1.9. I²C configuration register (SYSCFG_EIIC)

Address offset:0x20

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res						PF_EIIC[1:0]		Res							
-						RW		-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PB_EIIC[10:0]								Res						PA_EIIC[1:0]	
RW								-						RW	

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25:24	PF_EIIC[1:0]	RW	0	EIIC control signal of PF. Used as an I²C type IO Bit0:control PF3 Bit1:control PF4
23:16	Reserved	-	-	Reserved
15:8	PB_EIIC[7:0]	RW	0	EIIC control signal for PB. Used as an I²C type IO Bit0:control PB6 Bit1:control PB7 Bit2:control PB8 Bit3:control PB9 Bit4:control PB10 Bit5:control PB11 Bit6:control PB13 Bit7:control PB14
7:2	Reserved	-	-	Reserved
1:0	PA_EIIC[1:0]	RW	0	EIIC control signal of PA. Used as an I²C type IO Bit0:control PA9 Bit1:control PA10

Puya Confidential

11. Direct memory access (DMA)

11.1. Introduction

Direct memory access (DMA) is used to provide a high-speed data transfer between peripherals and memory as well as from memory to memory. Data can be quickly moved by DMA without any CPU actions. This keeps the CPU resources free for other operations. The DMA controller have 7 channels in total, each one dedicated to manage memory access requests from one or more peripherals. Each controller has an arbiter for handling the priority between DMA requests.

11.2. DMA main features

- Single AHB Master
- Supports peripheral-to-memory, memory-to-peripheral, memory-to-memory and peripheral-to-peripheral data transfers
- On-chip memory devices, such as Flash, an SRAM, AHB and APB peripherals, as the source and target
- All DMA channels can be configured independently:
  - Each channel is associated either with a DMA request signal coming from a peripheral, or with a software trigger in memory-to-memory transfers. This configuration is done by software.
  - The priority between requests is programmable by software (4 levels per channel:very high, high, medium and low) and, in equal cases, by hardware (such as a request for channel 1 taking precedence over a request for channel 2).
  - The transfer sizes of the source and destination are independent (byte, half word and word), simulating packing and unpacking. The source and destination addresses must be aligned by data size.
  - Programmable transmission data number:0 to 65535
- Each channel generates an interrupt request. Each interrupt request is caused from any of the three DMA events:transfer complete, half transfer, or transfer error.

Puya Confidential

11.3. DMA functional description

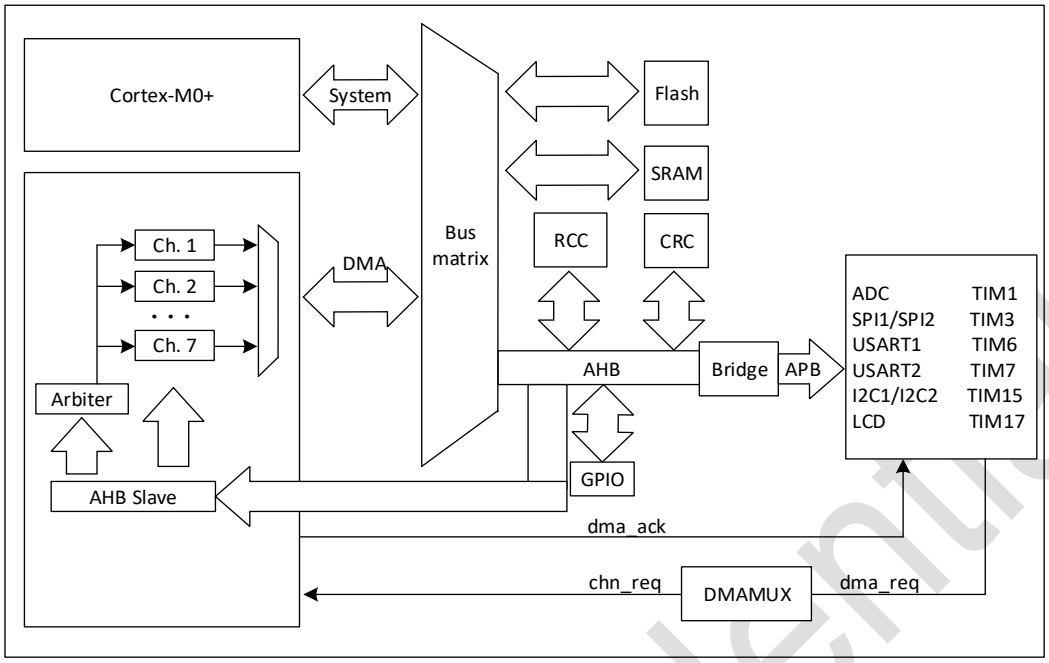


Figure 11-1 DMA block diagram

11.3.1. DMA processing

After completing an event, the peripheral sends a request signal to the DMA controller. The DMA controller serves the request, depending on the priority of the channel associated to this peripheral request. As soon as the DMA controller grants the peripheral, an acknowledge is sent to the peripheral by the DMA controller. The peripheral releases its request as soon as it gets the acknowledge from the DMA controller. Once the request is de-asserted by the peripheral, the DMA controller releases the acknowledge. The peripheral may order a further single request and initiate another single DMA transfer.

In summary, each DMA transfer consists of three operations:

- From the peripheral data register or from the current peripheral/memory address register, the start address used for the first single transfer is the base address of the peripheral or memory, and is programmed in the DMA_CPARx or DMA_CMARx register.
- The data is stored to the peripheral register or the memory address indicated by the current peripheral/memory address register, and the start address used for the first single transfer is the base address of the peripheral or memory, and is programmed in the DMA_CPARx or DMA_CMARx register.
- post-decrementing of the programmed DMA_CNDTRx register, which contains the remaining number of data items to transfer.

11.3.2. Arbiter

The arbiter initiates access to the peripheral/memory according to the priority of the channel request.

The priorities are managed in two stages:

- Software:The priority of each channel can be set in the DMA_CCRx register, with 4 levels

Puya Confidential

- Highest priority
- High priority
- Medium priority
- Low priority

- Hardware:if two requests have the same software priority level, the channel with the lowest index gets priority. For example, channel 2 gets priority over channel 4.

11.3.3. DMA channel

Each channel may handle a DMA transfer between a peripheral register located at a fixed address, and a memory address. The amount of data items to transfer is programmable, up to 65,535 bytes. The register that contains the amount of data items to transfer is decremented after each transfer.

Programmable data sizes

The transfer sizes of a single data to the peripheral and memory are programmable through, respectively, the PSIZE and MSIZE fields of the DMA_CCRx register.

Pointer incrementation

The peripheral and memory pointers may be automatically incremented after each transfer, depending on the PINC and MINC bits of the DMA_CCRx register.

If the incremented mode is enabled, the address of the next transfer is the address of the previous one incremented by 1, 2 or 4. The first transfer address is the one programmed in the DMA_CPARx or DMA_CMARx register. During transfers, these registers keep the initially programmed value. The current transfer addresses (in the current internal peripheral /memory address register) are not accessible by software.

If the channel x is configured in non-circular mode, no DMA request is served after the last data transfer. The DMA channel must be disabled in order to reload a new number of data items into the DMA_CNDTRx register.

In circular mode, after the last data transfer, the DMA_CNDTRx register is automatically reloaded with the initially programmed value. The current internal address registers are reloaded with the base address values from the DMA_CPARx and DMA_CMARx registers.

Channel configuration procedure

The following sequence is needed to configure a DMA channel x:

- Set the peripheral register address in the DMA_CPARx register. When a peripheral data transfer request occurs, this address will be the source or destination of the data transfer.
- Set the memory address in the DMA_CMARx register. When a peripheral data transfer request occurs, the transferred data will be read from or written to this address.
- Configure the total number of data to transfer in the DMA_CNDTRx register. After each data transfer, this value is decremented.
- Configure the parameters listed below in the DMA_CCRx register:
  - the channel priority
  - the data transfer direction

Puya Confidential

- the circular mode
- the peripheral and memory incremented mode
- the peripheral and memory data size
- the interrupt enable

■ Activate the channel by setting the EN bit in the DMA_CCRx register.

A channel, as soon as enabled, may serve any DMA request from the peripheral connected to this channel.

When half of the data is transmitted, the half transfer interrupt flag (HTIF) is set to 1. When the half transfer interrupt enable bit (HTIE) is set, an interrupt request will be generated. After the data transmission is completed, the transfer complete interrupt flag (TCIF) is set to 1. When the transfer complete interrupt enable (TCIE) is set, an interrupt request will be generated.

Channel state and disabling a channel

A channel x in active state is an enabled channel (read DMA_CCRx.EN = 1). An active channel x is a channel that must have been enabled by the software (DMA_CCRx.EN set to 1) and afterwards with no occurred transfer error (DMA_ISR.TEIFx = 0). In case there is a transfer error, the channel is automatically disabled by hardware (DMA_CCRx.EN = 0).

The three following use cases may happen:

1. Suspend and resume a channel

   This corresponds to the two following actions:

   1) An active channel is disabled by software (writing DMA_CCRx.EN = 0).
   2) The software enables the channel again (DMA_CCRx.EN set to 1) without reconfiguring the other channel registers (such as DMA_CNDTRx, DMA_CPARx and DMA_CMARx); Or a transfer that was not completed while the software was disabled pended the bus.

This case is not supported by the DMA hardware, that does not guarantee that the remaining data transfers are performed correctly.

2. Stop and abort a channel

   If the application does not need any more the channel, this active channel can be disabled by software. The channel is stopped and aborted but the DMA_CNDTRx register content may not correctly reflect the remaining data transfers.

3. Abort and restart a channel

This corresponds to the software sequence:disable an active channel, then reconfigure the channel and enable it again.

This is supported by the hardware if the following conditions are met:

1. The application guarantees that, when the software is disabling the channel, a DMA data transfer is not occurring at the same time over its master port. For example, the application can first disable the peripheral in DMA mode, in order to ensure that there is no pending hardware DMA request from this peripheral.

Puya Confidential

2. The software must operate separated write accesses to the same DMA_CCRx register:First disable the channel. Second reconfigure the channel for a next block transfer including the DMA_CCRx if a configuration change is needed. Finally enable again the channel.

When a channel transfer error occurs, the EN bit of the DMA_CCRx register is cleared by hardware. This EN bit cannot be set again by software to re-activate the channel x, until the TEIFx bit of the DMA_ISR register is set.

DMA circular mode

The circular mode is available to handle circular buffers and continuous data flows (such as ADC scan mode). This feature is enabled using the CIRC bit in the DMA_CCRx register. When the circular mode is started and the number of data transmissions becomes 0, the initial value set when the channel is configured will be automatically restored, and the DMA operation will continue.

Note:The circular mode must not be used in memory-to-memory mode. Before enabling a channel in circular mode (CIRC = 1), the software must clear the MEM2MEM bit of the DMA_CCRx register. When circular mode is activated, the amount of data to be transferred will be automatically reloaded during the channel configuration phase using the programmed initial values, and DMA requests will continue to be responded to.

In order to stop a circular transfer, the software needs to stop the peripheral from generating DMA requests (such as quit the ADC scan mode), before disabling the DMA channel.

The software must explicitly program the DMA_CNDTRx value before starting/enabling atransfer, and after having stopped a circular transfer.

Memory-to-memory mode

The DMA channels may operate without being triggered by a request from a peripheral. This mode is called memory-to-memory mode.

If the MEM2MEM bit in the DMA_CCRx register is set and the channel is enabled by the EN bit in the DMA_CCRx register, transfer initiates. The transfer stops once the DMA_CNDTRx register reaches zero.

The memory-to-memory mode must not be used in circular mode.

Peripheral-to-peripheral mode

Any DMA channel can operate in peripheral-to-peripheral mode:

- when the hardware request from a peripheral is selected to trigger the DMA channel

This peripheral is the DMA initiator and paces the data transfer from/to this peripheral to/from a register belonging to another memory-mapped peripheral (this one being not configured in DMA mode).

- When no peripheral request is selected and connected to the DMA channel

The software configures a register-to-register transfer by setting the MEM2MEM bit of the DMA_CCRx register.

Puya Confidential

Programming transfer direction, assigning source/destination

The value of the DIR bit of the DMA_CCRx register sets the direction of the transfer, and consequently, it identifies the source and the destination, regardless the source/destination type (peripheral or memory):

- DIR = 1 defines typically a memory-to-peripheral transfer. More generally, if DIR = 1:
- The source attributes are defined by the DMA_MARx register, the MSIZE [1:0] field and MINC bit of the DMA_CCRx register.

Regardless of their usual naming, these memory register, field and bit are used to define the source peripheral in peripheral-to-peripheral mode.

- The destination attributes are defined by the DMA_PARx register, the PSIZE [1:0] field and PINC bit of the DMA_CCRx register.

Regardless of their usual naming, these peripheral register, field and bit are used to define the destination memory in memory-to-memory mode.

- DIR = 0 typically defines a peripheral-to-memory transfer. More generally, if DIR = 0:
- The source attributes are defined by the DMA_PARx register, the PSIZE [1:0] field and PINC bit of the DMA_CCRx register. Regardless of their usual naming, these peripheral register, field and bit are used to define the source memory in memory-to-memory mode.
- The destination attributes are defined by the DMA_MARx register, the MSIZE [1:0] field and MINC bit of the DMA_CCRx register.

Regardless of their usual naming, these memory register, field and bit are used to define the destination peripheral in peripheral-to-peripheral mode.

11.3.4. DMA data width, alignment and endianness

When PSIZE and MSIZE are not equal, the DMA controller performs some data alignments as described in the table below.

Table 11-1 Data width and endian (PINC = MINC = 1)

Source Width	Destination width	Transmission Number	Source: address/data	DMA transfers	Destination: address/data
8	8	4	0x0/B0	1:Read B0 [7:0] at 0x0, write B0 [7:0] at 0x0	0x0/B0
			0x1/B1	2:Read B1 [7:0] at 0x1, write B1 [7:0] at 0x1	0x1/B1
			0x2/B2	3:Read B2 [7:0] at 0x2, write B2 [7:0] at 0x2	0x2/B2
			0x3/B3	4:Read B3 [7:0] at 0x3, write B3 [7:0] at 0x3	0x3/B3
8	16	4	0x0/B0	1:Read B0 [7:0] at 0x0, write 00B0 [7:0] at 0x0	0x0/00B0
			0x1/B1	2:Read B1 [7:0] at 0x1, write 00B1 [7:0] at 0x2	0x2/00B1
			0x2/B2	3:Read B2 [7:0] at 0x2, write 00B2 [7:0] at 0x4	0x4/00B2
			0x3/B3	4:Read B3 [7:0] at 0x3, write 00B3 [7:0] at 0x6	0x6/00B3
8	32	4	0x0/B0	1:Read B0 [7:0] at 0x0, write 000000B0 [31:0] at 0x0	0x0/000000B0
			0x1/B1	2:Read B1 [7:0] at 0x1, write 000000B1 [31:0] at 0x4	0x4/000000B1

Puya Confidential

			0x2/B2	3:Read B2 [7:0] at 0x2, write 000000B2 [31:0] at 0x8	0x8/000000B2
			0x3/B3	4:Read B3 [7:0] at 0x3, write 000000B3 [31:0] at 0xC	0xC/000000B3
16	8	4	0x0/B1B0	1:Read B1B0 [15:0] at 0x0, write B0 [7:0] at 0x0	0x0/B0
			0x2/B3B2	2:Read B3B2 [15:0] at 0x2, write B2 [7:0] at 0x1	0x1/B2
			0x4/B5B4	3:Read B5B4 [15:0] at 0x4, write B4 [7:0] at 0x2	0x2/B4
			0x6/B7B6	4:Read B7B6 [15:0] at 0x6, write B6 [7:0] at 0x3	0x3/B6
16	16	4	0x0/B1B0	1:Read B1B0 [15:0] at 0x0, write B1B0 [15:0] at 0x0	0x0/B1B0
			0x2/B3B2	2:Read B3B2 [15:0] at 0x2, write B3B2 [15:0] at 0x2	0x2/B3B2
			0x4/B5B4	3:Read B5B4 [15:0] at 0x4, write B5B4 [15:0] at 0x4	0x4/B5B4
			0x6/B7B6	4:Read B7B6 [15:0] at 0x6, write B7B6 [15:0] at 0x6	0x6/B7B6
16	32	4	0x0/B1B0	1:Read B1B0 [7:0] at 0x0, write 0000B1B0 [31:0] at 0x0	0x0/0000B1B0
			0x2/B3B2	2:Read B3B2 [7:0] at 0x2, write 0000B3B2 [31:0] at 0x4	0x4/000B3B2
			0x4/B5B4	3:Read B5B4 [7:0] at 0x4, write 0000B5B4 [31:0] at 0x8	0x8/0000B5B4
			0x6/B7B6	4:Read B7B6 [7:0] at 0x6, write 0000B7B6 [31:0] at 0xC	0xC/0000B7B6
32	8	4	0x0/B3B2B1B0	1:Read B3B2B1B0 [31:0] at 0x0, write B0 [7:0] at 0x0	0x0/B0
			0x4/B7B6B5B4	2:Read B7B6B5B4 [31:0] at 0x4, write B4 [7:0] at 0x1	0x1/B4
			0x8/BBBAB9B8	3:Read BBBAB9B8 [31:0] at 0x8, write B8 [7:0] at 0x2	0x2/B8
			0xC/BFBEBDBC	4:Read BFBEBDBC [31:0] at 0xc, write BC [7:0] at 0x3	0x3/BC
32	16	4	0x0/B3B2B1B0	1:Read B3B2B1B0 [31:0] at 0x0, write B1B0 [7:0] at 0x0	0x0/B1B0
			0x4/B7B6B5B4	2:Read B7B6B5B4 [31:0] at 0x4, write B5B4 [7:0] at 0x2	0x2/B5B4
			0x8/BBBAB9B8	3:Read BBBAB9B8 [31:0] at 0x8, write B9B8 [7:0] at 0x4	0x4/B9B8
			0xC/BFBEBDBC	4:Read BFBEBDBC at 0xc [31:0], write BDBC at 0x6 [7:0]	0x6/BDBC
32	32	4	0x0/B3B2B1B0	1:Read B3B2B1B0 [31:0] at 0x0, write B3B2B1B0 [7:0] at 0x0	0x0/B3B2B1B0
			0x4/B7B6B5B4	2:Read B7B6B5B4 [31:0] at 0x4, write B7B6B5B4 [7:0] at 0x2	0x4/B7B6B5B4
			0x8/BBBAB9B8	3:Read BBBAB9B8 [31:0] at 0x8, write BBBAB9B8 [7:0] at 0x4	0x8/BBBAB9B8
			0xC/BFBEBDBC	4:Read BFBEBDBC [31:0] at 0xc, write BFBEBDBC [7:0] at 0x6	0xC/BFBEBDBC

Addressing AHB peripherals not supporting byte/half-word write transfers

When the AHB peripheral does not support byte or half-word write transfers and does not generate any error, the DMA controller writes the 32 HWDATA bits as shown in the two examples below:

- To write the half-word 0xABCD, the DMA controller sets the HWDATA bus to 0xABCDABCD with a half-word data size (HSIZE = Half-word).
- To write the byte 0xAB, the DMA controller sets the HWDATA bus to 0xABABABAB with a byte data size (HSIZE = Byte).

Puya Confidential

Assuming the AHB/APB bridge is an AHB 32-bit slave peripheral that does not take into account the HSIZE data, any AHB byte or half-word transfer is changed into a 32-bit APB transfer as described below:

- An AHB byte write transfer of 0xB0 to one of the 0x0, 0x1, 0x2 or 0x3 addresses, is converted to an APB word write transfer of 0xB0B0B0B0 to the 0x0 address.
- An AHB half-word write transfer of 0xB1B0 to the 0x0 or 0x2 addresses, is converted to an APB word write transfer of 0xB1B0B1B0 to the 0x0 address.

11.3.5. DMA error management

A DMA transfer error is generated when reading from or writing to a reserved address space. When a DMA transfer error occurs during a DMA read or write access, the faulty channel x is automatically disabled through a hardware clear of its EN bit in the corresponding DMA_CCRx register. The TEIF bit of the DMA_ISR register is set. An interrupt is then generated if the TEIE bit of the DMA_CCRx register is set.

The EN bit of the DMA_CCRx register cannot be set again by software (channel x reactivated) until the TEIFx bit of the DMA_ISR register is cleared (by setting the Citify bit of the DMA_IFCR register).

11.3.6. DMA interrupts

An interrupt can be generated on a half transfer, transfer complete or transfer error for each DMA channel x. Separate interrupt enable bits are available for flexibility.

Table 11-2 DMA interrupt request

Interrupt event	Event flag	Enable control bit
Half transfer	Hotfix	HTIEx
Transfer complete	TCIFx	TCIEx
Transfer error	TEIFx	TEIEx
Half transfer/Transfer complete/Transfer error	GIFx	-

Notes:

1. When the DMA_CNDTRx register is 1, the HTIFx bit will not be set, and the TCIFx bit will be set when the transmission is completed.

2. When the transmission length NDT is odd (greater than 1), both HTIF and TCIF flags are generated. The internal signal TCIF will be generated when NDT = 1; The HTIF is generated at (NDT-(NDT/2 (rounded)-1)). For example, when NDT = 5, TCIF is generated when NDT is reduced to 1; HTIF is generated when NDT is reduced to 4.

3. When the transmission length NDT is even (greater than 1), both HTIF and TCIF flags are generated. The internal signal TCIF will be generated when NDT = 1; The HTIF is generated at (NDT-(NDT/2 (rounded)-1)). For example, when NDT = 10, TCIF is generated when NDT is reduced to 1; HTIF is generated when NDT is reduced to 6.

Puya Confidential

11.3.7. DMAMUX mapping

DMA peripheral requests are mapped to each channel of DMA, which is controlled by the DMAx_MAP register bits of two SYSCFG registers (SYSCFG_CFGR3 and SYSCFGR_CFGR4). Each peripheral request can be mapped to any of the 7 channels through configuration.

Table 11-3 DMA requests per channel

DMA request MUX input	Source	DMA request MUX input	Source	DMA request MUX input	Source
0	ADC1	17	I2C2_RD	34	TIM3_CH3
1	-	18	I2C2_WR	35	TIM3_CH4
2	-	19	TIM1_CH1	36	TIM3_UP
3	SPI1_RD	20	TIM1_CH2	37	TIM3_TRG
4	SPI1_WR	21	TIM1_CH3	38	TIM6_UP
5	SPI2_RD	22	TIM1_CH4	39	TIM7_UP
6	SPI2_WR	23	TIM1_COM	40	TIM15_CH1
7	USART1_RD	24	TIM1_TRG	41	TIM15_CH2
8	USART1_WR	25	TIM1_UP	42	TIM15_UP
9	USART2_RD	26	TIM2_CH1	43	TIM15_TRG
10	USART2_WR	27	TIM2_CH2	44	TIM15_COM
11	USART3_RD	28	TIM2_CH3	45	TIM16_CH1
12	USART3_WR	29	TIM2_CH4	46	TIM16_UP
13	USART4_RD	30	TIM2_UP	47	TIM17_CH1
14	USART4_WR	31	TIM2_TRG	48	TIM17_UP
15	I2C1_RD	32	TIM3_CH1	49	-
16	I2C1_WR	33	TIM3_CH2	50	-

11.4. DMA registers

11.4.1. DMA interrupt status register (DMA_ISR)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:28	Reserved	-	-	Reserved
27	TEIF7	R	0	Transfer error (TE) flag for channel 7 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TE event 1:a TE event occurred
26	HTIF7	R	0	Half transfer (HT) flag for channel 7 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no HT event 1:a HT event occurred
25	TCIF7	R	0	Transfer complete (TC) flag for channel 7 0:no TC event 1:a TC event occurred
24	GIF7	R	0	Global interrupt flag for channel 7 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:No TE/HT/TC event 1:a TE/HT/TC event occurred
23	TEIF6	R	0	Transfer error (TE) flag for channel 6 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TE event 1:a TE event occurred
22	HTIF6	R	0	Half transfer (HT) flag for channel 6 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no HT event 1:a HT event occurred
21	TCIF6	R	0	Transfer complete (TC) flag for channel 6 0:no TC event 1:a TC event occurred
20	GIF6	R	0	Global interrupt flag for channel 6 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:No TE/HT/TC event 1:a TE/HT/TC event occurred
19	TEIF5	R	0	Transfer error (TE) flag for channel 5 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TE event 1:a TE event occurred
18	HTIF5	R	0	Half transfer (HT) flag for channel 5 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no HT event 1:a HT event occurred

Puya Confidential

Bit	Name	R/W	Reset Value	Function
17	TCIF5	R	0	Transfer complete (TC) flag for channel 5 0:no TC event 1:a TC event occurred
16	GIF5	R	0	Global interrupt flag for channel 5 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:No TE/HT/TC event 1:a TE/HT/TC event occurred
15	TEIF4	R	0	Transfer error (TE) flag for channel 4 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TE event 1:a TE event occurred
14	HTIF4	R	0	Half transfer (HT) flag for channel 4 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no HT event 1:a HT event occurred
13	TCIF4	R	0	Transfer complete (TC) flag for channel 4 0:no TC event 1:a TC event occurred
12	GIF4	R	0	Global interrupt flag for channel 4 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:No TE/HT/TC event 1:a TE/HT/TC event occurred
11	TEIF3	R	0	Transfer error (TE) flag for channel 3 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TE event 1:a TE event occurred
10	HTIF3	R	0	Half transfer (HT) flag for channel 3 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no HT event 1:a HT event occurred
9	TCIF3	R	0	Transfer complete (TC) flag for channel 3 0:no TC event 1:a TC event occurred
8	GIF3	R	0	Global interrupt flag for channel 3 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:No TE/HT/TC event 1:a TE/HT/TC event occurred
7	TEIF2	R	0	Transfer error (TE) flag for channel 2 The hardware is set, and the software writes DMA_IFCR = 1 to clear.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:no TE event 1:a TE event occurred
6	HTIF2	R	0	Half transfer (HT) flag for channel 2 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no HT event 1:a HT event occurred
5	TCIF2	R	0	Transfer complete (TC) flag for channel 2 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TC event 1:a TC event occurred
4	GIF2	R	0	Global interrupt flag for channel 2 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:No TE/HT/TC event 1:a TE/HT/TC event occurred
3	TEIF1	R	0	Transfer error (TE) flag for channel 1 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TE event 1:a TE event occurred
2	HTIF1	R	0	Half transfer (HT) flag for channel 1 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no HT event 1:a HT event occurred
1	TCIF1	R	0	Transfer complete (TC) flag for channel 1 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:no TC event 1:a TC event occurred
0	GIF1	R	0	Global interrupt flag for channel 1 The hardware is set, and the software writes DMA_IFCR = 1 to clear. 0:No TE/HT/TC event 1:a TE/HT/TC event occurred

11.4.2. DMA interrupt flag clear register (DMA_IFCR)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
-	-	-	-	W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

Puya Confidential

w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit	Name	R/W	Reset Value	Function
31:28	Reserved	-	-	Reserved
27	CTEIF7	W	0	Transfer error flag clear for channel 7 0:No effect 1:Clear TEIF7
26	CHTIF7	W	0	Half transfer flag clear for channel 7 0:No effect 1:Clear HTIF7
25	CTCIF7	W	0	Transfer complete flag clear for channel 7 0:No effect 1:Clear TCIF7
24	CGIF7	W	0	Global interrupt flag clear for channel 7 0:No effect 1:GIF/TEIF/HTIF/TCIF of channel 7 is cleared
23	CTEIF6	W	0	Transfer error flag clear for channel 6 0:No effect 1:Clear TEIF6
22	CHTIF6	W	0	Half transfer flag clear for channel 6 0:No effect 1:Clear HTIF6
21	CTCIF6	W	0	Transfer complete flag clear for channel 6 0:No effect 1:Clear TCIF6
20	CGIF6	W	0	Global interrupt flag clear for channel 6 0:No effect 1:GIF/TEIF/HTIF/TCIF of channel 6 is cleared
19	CTEIF5	W	0	Transfer error flag clear for channel 5 0:No effect 1:Clear TEIF5
18	CHTIF5	W	0	Half transfer flag clear for channel 5 0:No effect 1:Clear HTIF5
17	CTCIF5	W	0	Transfer complete flag clear for channel 5 0:No effect 1:Clear TCIF5
16	CGIF5	W	0	Global interrupt flag clear for channel 5 0:No effect 1:GIF/TEIF/HTIF/TCIF of channel 5 is cleared

Puya Confidential

Bit	Name	R/W	Reset Value	Function
15	CTEIF4	W	0	Transfer error flag clear for channel 4 0:No effect 1:Clear TEIF4
14	CHTIF4	W	0	Half transfer flag clear for channel 4 0:No effect 1:Clear HTIF4
13	CTCIF4	W	0	Transfer complete flag clear for channel 4 0:No effect; 1:Clear TCIF4;
12	CGIF4	W	0	Global interrupt flag clear for channel 4 0:No effect 1:GIF/TEIF/HTIF/TCIF of channel 4 is cleared
11	CTEIF3	W	0	Transfer error flag clear for channel 3 0:No effect 1:Clear TEIF3
10	CHTIF3	W	0	Half transfer flag clear for channel 3 0:No effect 1:Clear HTIF3
9	CTCIF3	W	0	Transfer complete flag clear for channel 3 0:No effect 1:Clear TCIF3
8	CGIF3	W	0	Global interrupt flag clear for channel 3 0:No effect 1:GIF/TEIF/HTIF/TCIF of channel 3 is cleared
7	CTEIF2	W	0	Transfer error flag clear for channel 2 0:No effect 1:Clear TEIF2
6	CHTIF2	W	0	Half transfer flag clear for channel 2 0:No effect 1:HTIF2 cleared
5	CTCIF2	W	0	Transfer complete flag clear for channel 2 0:No effect 1:TCIF2 cleared
4	CGIF2	W	0	Global interrupt flag clear for channel 2 0:No effect 1:GIF/TEIF/HTIF/TCIF of channel 2 is cleared
3	CTEIF1	W	0	Transfer error flag clear for channel 1 0:No effect 1:Clear TEIF1

Puya Confidential

Bit	Name	R/W	Reset Value	Function
2	CHTIF1	W	0	Half transfer flag clear for channel 1 0:No effect 1:Clear HTIF1
1	CTCIF1	W	0	Transfer complete flag clear for channel 1 0:No effect 1:Clear TCIF1
0	CGIF1	W	0	Global interrupt flag clear for channel 1 0:No effect 1:GIF/TEIF/HTIF/TCIF of channel 1 is cleared

11.4.3. DMA channel 1 configuration register (DMA_CCR1)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel 1 memory to memory mode. 0:Disabled 1:Memory-to-memory mode enabled
13:12	PL [1:0]	RW	0	Channel 1 priority configuration. 00:low 01:Medium 10:high 11:very high
11:10	MSIZE [1:0]	RW	0	Channel 1 memory data width. 00:8-bit 01:16-bit 10:32-bit 11:Reserved
9:8	PSIZE [1:0]	RW	0	Channel 1 peripheral data width. 00:8-bit 01:16-bit 10:32-bit

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:Reserved
7	MINC	RW	0	Channel 1 memory increment mode. 0:Disabled 1:Enabled
6	PINC	RW	0	Channel 1 peripheral increment mode. 0:Disabled 1:Enabled
5	CIRC	RW	0	Channel 1 circular mode. 0:Disabled 1:Circular mode enabled
4	DIR	RW	0	Channel 1 data transfer direction. 0:read from peripheral 1:read from memory
3	TEIE	RW	0	Channel 1 transfer error interrupt (TE) enable 0:Disabled 1:TE interrupt enabled
2	HTIE	RW	0	Channel 1 half transfer interrupt (HT) enable 0:Disabled 1:HT interrupt enabled
1	TCIE	RW	0	Channel 1 transfer complete interrupt (TC) enable 0:Disabled 1:TC interrupt enabled
0	EN	RW	0	Channel 1 enable 0:Disabled 1:Channel 1 enabled

11.4.4. DMA channel 1 number of data to transfer register (DMA_CNDTR1)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT [15:0]	RW	0	Channel 1 number of data to transfer

Puya Confidential

				The number of data transfer is 0 to 65535. This register is written only when the channel is not working (DMA_CCR1.EN = 0). This register is read-only when the channel is enabled, indicating the number of bytes remaining to be transferred. This register value is decremented after each DMA transfer. After the data transfer is completed, the contents of the register either become 0, or when the channel is configured in circular mode, the contents of the register will be automatically reloaded to the previously configured value. If this field is zero, no transfer can be served whatever the channel status.

11.4.5. DMA channel 1 peripheral address register (DMA_CPAR1)

Address offset:0x10

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA [31:0]	RW	0	Channel 1 peripheral address The base address of the channel 1 peripheral data register as the source or destination of data transmission. When PSIZE = 2 'b01, the PA [0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE = 2 'b10, the PA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.6. DMA channel 1 memory address register (DMA_CMAR1)

Address offset:0x14

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MA [15:0]															

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA [31:0]	RW	0	Channel 1 memory address. Channel 1 memory address, as the source or destination of data transfer. When MSIZE = 2 'b01, the MA [0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE = 2 'b10, the MA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.7. DMA channel 2 configuration register (DMA_CCR2)

Address offset:0x1C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel 2 memory to memory mode. 0:Disabled 1:Memory-to-memory mode enabled
13:12	PL [1:0]	RW	0	Channel 2 priority configuration. 00:low 01:Medium 10:high 11:very high
11:10	MSIZE [1:0]	RW	0	Channel 2 memory data width. 00:8-bit 01:16-bit 10:32-bit 11:Reserved
9:8	PSIZE [1:0]	RW	0	Channel 2 peripheral data width. 00:8-bit 01:16-bit

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				10:32-bit 11:Reserved
7	MINC	RW	0	Channel 2 memory increment mode. 0:Disabled 1:Enabled
6	PINC	RW	0	Channel 2 peripheral increment mode. 0:Disabled 1:Enabled
5	CIRC	RW	0	Channel 2 circular mode. 0:Disabled 1:Circular mode enabled
4	DIR	RW	0	Channel 2 data transfer direction. 0:read from peripheral 1:read from memory
3	TEIE	RW	0	Channel 2 transfer error interrupt (TE) enable 0:Disabled 1:TE interrupt enabled
2	HTIE	RW	0	Channel 2 half transfer interrupt (HT) enable 0:Disabled 1:HT interrupt enabled
1	TCIE	RW	0	Channel 2 transfer complete interrupt (TC) enable 0:Disabled 1:TC interrupt enabled
0	EN	RW	0	Channel 2 enable 0:Disabled 1:Channel 1 enabled

11.4.8. DMA channel 2 number of data to transfer register (DMA_CNDTR2)

Address offset:0x20

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved

puya Confidential

Bit	Name	R/W	Reset Value	Function
15:0	NDT [15:0]	RW	0	Channel 2 number of data to transfer The number of data transmissions is 0 to 65535. This register is only written when the channel is not working (DMA_CCR2. EN = 0). This register is read-only when the channel is enabled, indicating the number of bytes remaining to be transferred. This register value is decremented after each DMA transfer. After the data transfer is completed, the contents of the register either become 0, or when the channel is configured in circular mode, the contents of the register will be automatically reloaded to the previously configured value. If this field is zero, no transfer can be served whatever the channel status.

11.4.9. DMA channel 2 peripheral address register (DMA_CPAR2)

Address offset:0x24

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA [31:0]	RW	0	Channel 2 peripheral address It contains the base address of the peripheral data register from/to which the data will be read/written. When PSIZE = 2 'b01, the PA [0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE = 2 'b10, the PA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.10. DMA channel 2 memory address register (DMA_CMAR2)

Address offset:0x28

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA [31:0]	RW	0	Channel 2 memory address. It contains the base address of the memory from/to which the data will be read/written. When MSIZE = 2 'b01, the MA [0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE = 2 'b10, the MA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.11. DMA channel 3 configuration register (DMA_CCR3)

Address offset:0x30

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel 3 memory to memory mode. 0:Disabled 1:Memory-to-memory mode enabled
13:12	PL [1:0]	RW	0	Channel 3 priority configuration. 00:low 01:Medium 10:high 11:very high
11:10	MSIZE [1:0]	RW	0	Channel 3 memory data width. 00:8-bit 01:16-bit 10:32-bit 11:Reserved
9:8	PSIZE [1:0]	RW	0	Channel 3 peripheral data width.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				00:8-bit 01:16-bit 10:32-bit 11:Reserved
7	MINC	RW	0	Channel 3 memory increment mode. 0:Disabled 1:Enabled
6	PINC	RW	0	Channel 3 peripheral increment mode. 0:Disabled 1:Enabled
5	CIRC	RW	0	Channel 3 circular mode. 0:Disabled 1:Circular mode enabled
4	DIR	RW	0	Channel 3 data transfer direction. 0:read from peripheral 1:read from memory
3	TEIE	RW	0	Channel 3 transfer error interrupt (TE) enable 0:Disabled 1:TE interrupt enabled
2	HTIE	RW	0	Channel 3 half transfer interrupt (HT) enable 0:Disabled 1:HT interrupt enabled
1	TCIE	RW	0	Channel 3 transfer complete interrupt (TC) enable 0:Disabled 1:TC interrupt enabled
0	EN	RW	0	Channel 3 enable 0:Disabled 1:Channel 1 enabled

11.4.12. DMA channel 3 number of data to transfer register (DMA_CNDTR3)

Address offset:0x34

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT [15:0]	RW	0	Channel 3 number of data to transfer The number of data transmissions is 0 to 65535. This register is only written when the channel is not working (DMA_CCR3. EN = 0). This register is read-only when the channel is enabled, indicating the number of bytes remaining to be transferred. This register value is decremented after each DMA transfer. After the data transfer is completed, the contents of the register either become 0, or when the channel is configured in circular mode, the contents of the register will be automatically reloaded to the previously configured value. If this field is zero, no transfer can be served whatever the channel status.

11.4.13. DMA channel 3 peripheral address register (DMA_CPAR3)

Address offset:0x38

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA [31:0]	RW	0	Channel 3 peripheral address. It contains the base address of the peripheral data register from/to which the data will be read/written. When PSIZE = 2 'b01, the PA [0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE = 2 'b10, the PA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.14. DMA channel 3 memory address register (DMA_CMAR3)

Address offset:0x3C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Puya Confidential

MA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA [31:0]	RW	0	Channel 3 memory address. It contains the base address of the memory from/to which the data will be read/written. When MSIZE = 2 'b01, the MA [0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE = 2 'b10, the MA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.15. DMA channel 4 configuration register (DMA_CCR4)

Address offset:0x44

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel memory-to-memory mode. 0:Disabled 1:Memory-to-memory mode enabled
13:12	PL [1:0]	RW	0	Channel priority configuration. 00:low 01:medium 10:high 11:very high
11:10	MSIZE [1:0]	RW	0	Channel memory data width. 00:8-bit 01:16-bit 10:32-bit

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:Reserved
9:8	PSIZE [1:0]	RW	0	Channel peripheral data width. 00:8-bit 01:16-bit 10:32-bit 11:Reserved
7	MINC	RW	0	Channel memory increment mode. 0:Disabled 1:Enabled
6	PINC	RW	0	Channel peripheral increment mode. 0:Disabled 1:Enabled
5	CIRC	RW	0	Channel circular mode. 0:Disabled 1:Circular mode enabled
4	DIR	RW	0	Channel data transfer direction. 0:read from peripheral 1:read from memory
3	TEIE	RW	0	Channel transfer error interrupt (TE) enable 0:Disabled 1:TE interrupt enabled
2	HTIE	RW	0	Channel half transfer interrupt (HT) enable 0:Disabled 1:HT interrupt enabled
1	TCIE	RW	0	Channel transfer complete interrupt (TC) enable 0:Disabled 1:TC interrupt enabled
0	EN	RW	0	Channel enable 0:Disabled 1:Channel enable

11.4.16. DMA channel 4 number of data to transfer register (DMA_CNDTR4)

Address offset:0x48

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT [15:0]															

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT [15:0]	RW	0	Channel number of data to transfer The number of data transmissions is 0 to 65535. This register is only written when the channel is not working (DMA_CCR4. EN = 0). This register is read-only when the channel is enabled, indicating the number of bytes remaining to be transferred. This register value is decremented after each DMA transfer. After the data transfer is completed, the contents of the register either become 0, or when the channel is configured in circular mode, the contents of the register will be automatically reloaded to the previously configured value. If this field is zero, no transfer can be served whatever the channel status.

11.4.17. DMA channel 4 peripheral address register (DMA_CPAR4)

Address offset:0x4C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA [31:0]	RW	0	Channel peripheral address. It contains the base address of the peripheral data register from/to which the data will be read/written. When PSIZE = 2 'b01, the PA [0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE = 2 'b10, the PA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.18. DMA channel 4 memory address register (DMA_CMAR4)

Address offset:0x50

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA [31:0]	RW	0	Channel memory address. It contains the base address of the memory from/to which the data will be read/written. When MSIZE = 2 'b01, the MA [0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE = 2 'b10, the MA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.19. DMA channel 5 configuration register (DMA_CCR5)

Address offset:0x58

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel memory-to-memory mode. 0:Disabled 1:Memory-to-memory mode enabled
13:12	PL [1:0]	RW	0	Channel priority configuration. 00:low 01:Medium 10:high 11:very high
11:10	MSIZE [1:0]	RW	0	Channel memory data width. 00:8-bit 01:16-bit 10:32-bit

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:Reserved
9:8	PSIZE [1:0]	RW	0	Channel peripheral data width. 00:8 bits 01:16-bit 10:32-bit 11:Reserved
7	MINC	RW	0	Channel memory increment mode. 0:Disabled 1:Enabled
6	PINC	RW	0	Channel peripheral increment mode. 0:Disabled 1:Enabled
5	CIRC	RW	0	Channel circular mode. 0:Disabled 1:Circular mode enabled
4	DIR	RW	0	Channel data transfer direction. 0:read from peripheral 1:read from memory
3	TEIE	RW	0	Channel transfer error interrupt (TE) enable 0:Disabled 1:TE interrupt enabled
2	HTIE	RW	0	Channel half transfer interrupt (HT) enable 0:Disabled 1:HT interrupt enabled
1	TCIE	RW	0	Channel transfer complete interrupt (TC) enable 0:Disabled 1:TC interrupt enabled
0	EN	RW	0	Channel enable. 0:Disabled 1:Channel enabled

11.4.20. DMA channel 5 number of data to transfer register (DMA_CNDTR5)

Address offset:0x5C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT [15:0]															

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT [15:0]	RW	0	Channel number of data to transfer The number of data transmissions is 0 to 65535. This register is only written when the channel is not working (DMA_CCR5. EN = 0). This register is read-only when the channel is enabled, indicating the number of bytes remaining to be transferred. This register value is decremented after each DMA transfer. After the data transfer is completed, the contents of the register either become 0, or when the channel is configured in ciucular mode, the contents of the register will be automatically reloaded to the previously configured value. If this field is zero, no transfer can be served whatever the channel status.

11.4.21. DMA channel 5 peripheral address register (DMA_CPAR5)

Address offset:0x60

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA [31:0]	RW	0	Channel peripheral address. It contains the base address of the peripheral data register from/to which the data will be read/written. When PSIZE = 2 'b01, the PA [0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE = 2 'b10, the PA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.22. DMA channel 5 memory address register (DMA_CMAR5)

Address offset:0x64

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA [31:0]	RW	0	Channel memory address. It contains the base address of the memory from/to which the data will be read/written. When MSIZE = 2 'b01, the MA [0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE = 2 'b10, the MA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.23. DMA channel 6 configuration register (DMA_CCR6)

Address offset:0x6C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel memory-to-memory mode. 0:Disabled 1:Memory-to-memory mode enabled
13:12	PL [1:0]	RW	0	Channel priority configuration. 00:low 01:medium 10:high 11:very high
11:10	MSIZE [1:0]	RW	0	Channel memory data width. 00:8 bits 01:16-bit 10:32-bit

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:Reserved
9:8	PSIZE [1:0]	RW	0	Channel peripheral data width. 00:8 bits 01:16-bit 10:32-bit 11:Reserved
7	MINC	RW	0	Channel memory increment mode. 0:Disabled 1:Enabled
6	PINC	RW	0	Channel peripheral increment mode. 0:Disabled 1:Enabled
5	CIRC	RW	0	Channel circular mode. 0:Disabled 1:Circular mode enabled
4	DIR	RW	0	Channel data transfer direction. 0:read from peripheral 1:read from memory
3	TEIE	RW	0	Channel transfer error interrupt (TE) enable 0:Disabled 1:TE interrupt enabled
2	HTIE	RW	0	Channel half transfer interrupt) enable 0:Disabled 1:HT interrupt enabled
1	TCIE	RW	0	Channel transfer complete interrupt (TC) enable 0:Disabled 1:TC interrupt enabled
0	EN	RW	0	Channel enable. 0:Disabled 1:Channel enable

11.4.24. DMA channel 6 number of data to transfer register (DMA_CNDTR6)

Address offset:0x70

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT [15:0]															

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT [15:0]	RW	0	Channel number of data to transfer The number of data transmissions is 0 to 65535. This register is only written when the channel is not working (DMA_CCR6. EN = 0). This register is read-only when the channel is enabled, indicating the number of bytes remaining to be transferred. This register value is decremented after each DMA transfer. After the data transfer is completed, the contents of the register either become 0, or when the channel is configured in circular mode, the contents of the register will be automatically reloaded to the previously configured value. If this field is zero, no transfer can be served whatever the channel status.

11.4.25. DMA channel 6 peripheral address register (DMA_CPAR6)

Address offset:0x74

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA [31:0]	RW	0	Channel peripheral address. It contains the base address of the peripheral data register from/to which the data will be read/written. When PSIZE = 2 'b01, the PA [0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE = 2 'b10, the PA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.26. DMA channel 6 memory address register (DMA_CMAR6)

Address offset:0x78

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA [31:0]	RW	0	Channel memory address. It contains the base address of the memory from/to which the data will be read/written. When MSIZE = 2 'b01, the MA [0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE = 2 'b10, the MA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.27. DMA channel 7 configuration register (DMA_CCR7)

Address offset:0x80

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel memory-to-memory mode. 0:Disabled 1:Memory-to-memory mode enabled
13:12	PL [1:0]	RW	0	Channel priority configuration. 00:low 01:medium 10:high 11:very high
11:10	MSIZE [1:0]	RW	0	Channel memory data width. 00:8-bit 01:16-bit 10:32-bit

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:Reserved
9:8	PSIZE [1:0]	RW	0	Channel peripheral data width. 00:8-bit 01:16-bit 10:32-bit 11:Reserved
7	MINC	RW	0	Channel memory increment mode. 0:Disabled 1:Enabled
6	PINC	RW	0	Channel peripheral increment mode. 0:Disabled 1:Enabled
5	CIRC	RW	0	Channel circular mode. 0:Disabled 1:Circular mode enabled
4	DIR	RW	0	Channel data transfer direction. 0:read from peripheral 1:read from memory
3	TEIE	RW	0	Channel transfer error interrupt (TE) enable 0:Disabled 1:TE interrupt enabled
2	HTIE	RW	0	Channel half transfer interrupt (HT) enable 0:Disabled 1:HT interrupt enabled
1	TCIE	RW	0	Channel transfer complete interrupt (TC) enable 0:Disabled 1:TC interrupt enabled
0	EN	RW	0	Channel enable. 0:Disabled 1:Channel enable

11.4.28. DMA channel 7 number of data to transfer register (DMA_CNDTR7)

Address offset:0x84

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT [15:0]															

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT [15:0]	RW	0	Channel number of data to transfer The number of data transmissions is 0 to 65535. This register is only written when the channel is not working (DMA_CCR7. EN = 0). This register is read-only when the channel is enabled, indicating the number of bytes remaining to be transferred. This register value is decremented after each DMA transfer. After the data transfer is completed, the contents of the register either become 0, or when the channel is configured in circular mode, the contents of the register will be automatically reloaded to the previously configured value. If this field is zero, no transfer can be served whatever the channel status.

11.4.29. DMA channel 7 peripheral address register (DMA_CPAR7)

Address offset:0x88

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA [31:0]	RW	0	Channel peripheral address. It contains the base address of the peripheral data register from/to which the data will be read/written. When PSIZE = 2 'b01, the PA [0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE = 2 'b10, the PA [1:0] bit is ignored. Access is automatically aligned with the word address.

11.4.30. DMA channel 7 memory address register (DMA_CMAR7)

Address offset:0x8C

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MA [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA [31:0]	RW	0	Channel memory address. It contains the base address of the memory from/to which the data will be read/written. When MSIZE = 2 'b01, the MA [0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE = 2 'b10, the MA [1:0] bit is ignored. Access is automatically aligned with the word address.

Puya Confidential

12. Interrupts and events

12.1. Nested vectored interrupt controller (NVIC)

12.1.1. Main features

- Support 30 maskable external interrupts (not including the sixteen CPU interrupt lines)
- 4 programmable priority levels (2 bits of interrupt priority are used)
- Low-latency exception and interrupt handling
- Power management control

The NVIC and the processor core interface are closely coupled, which enables low latency interrupt processing and efficient processing of late arriving interrupts. All interrupts inclusing the core exceptions are managed by the NVIC.

12.1.2. SysTick calibration value register

The SysTick calibration value is set to 9000, which gives a reference time base of 1 ms with the SysTick clock set to 9 MHz (max f_{HCLK}/8).

12.1.3. Interrupt and exception vectors

Table 12-1 Interrupt and exception vectors

Position	Priority	Type of priority	Pin name	Description	Address
-	-	-	-	Reserved	0x0000_0000
-	-3	fixed	Reset	Reset	0x0000_0004
-	-2	fixed	NMI_Handler	The RCC clock security system (CSS) is coupled to NMI vector	0x0000_0008
-	-1	fixed	HardFualt_Handler	All classes of fault	0x0000_000C
-	3	settable	SVCall	System service call via SWI instruction	0x0000_002C
-	5	settable	PendSV	Pendable request for system service	0x0000_0038
	6	settable	SysTick	System tick timer	0x0000_003C
0	7	settable	WWDG	Window watchdog interrupt	0x0000_0040
1	8	settable	PVD	PVD through EXTI line 16 interrupt	0x0000_0044
2	9	settable	RTC	RTC interrupt (combined EXTI lines 19)	0x0000_0048
3	10	settable	Flash	Flash global interrupt	0x0000_004C
4	11	settable	RCC	RCC global interrupt	0x0000_0050
5	12	settable	EXTI0_1	EXTI line [1:0] interrupt	0x0000_0054
6	13	settable	EXTI2_3	EXTI line [3:2] interrupt	0x0000_0058
7	14	settable	EXTI4_15	EXTI line [15:4] interrupt	0x0000_005C
8	15	settable	LCD	LCD global interrupt	0x0000_0060
9	16	settable	DMA_Channel1	DMA Channel 1 Interrupt	0x0000_0064
10	17	settable	DMA_Channel 2_3	DMA channel 2 & 3 interrupt	0x0000_0068
11	18	settable	DMA_Channel4_5_6_7	DMA channel 4 & 5 & 6 & 7 interrupt	0x0000_006C

PUYA Confidential

Position	Priority	Type of priority	Pin name	Description	Address
12	19	settable	ADC_COMP	ADC and COMP interrupts (COMP combined with EXTI 17 & 18 & 20)	0x0000_0070
13	20	settable	TIM1_BRK_UP_TRG_COM	TIM1 break, update, trigger and commutation interrupt	0x0000_0074
14	21	settable	TIM1_CC	TIM1 capture compare interrupt	0x0000_0078
15	22	settable	TIM2	TIM2 global interrupt	0x0000_007C
16	23	settable	TIM3	TIM3 global interrupt	0x0000_0080
17	24	settable	TIM6/LPTIM1	TIM6/LPTIM global interrupt	0x0000_0084
18	25	settable	TIM7	TIM7 global interrupt	0x0000_0088
19	26	settable	TIM14	TIM14 global interrupt	0x0000_008C
20	27	settable	TIM15	TIM15 global interrupt	0x0000_0090
21	28	settable	TIM16	TIM16 global interrupt	0x0000_0094
22	29	settable	TIM17	TIM17 global interrupt	0x0000_0098
23	30	settable	I2C1	I2C1 global interrupt	0x0000_009C
24	31	settable	I2C2	I2C2 global interrupt	0x0000_00A0
25	32	settable	SPI1	SPI1 global interrupt	0x0000_00A4
26	33	settable	SPI2	SPI2 global interrupt	0x0000_00A8
27	34	settable	USART1	USART1 global interrupt	0x0000_00AC
28	35	settable	USART2	USART2 global interrupt	0x0000_00B0
29	36	settable	USART3_4	USART3_4 global interrupt	0x0000_00B4
30	37	-	Reserved	-	0x0000_00B8
31	38	-	Reserved	-	0x0000_00BC

12.2. Extended interrupts and events controller (EXTI)

The extended interrupt and event controller (EXTI) manages the CPU and system wake-up through configurable and direct event inputs (Lines). It outputs following signals:

- Interrupt request, generating IRQ of CPU
- Event request, generating CPU event input (RXEV)
- The wake-up request is sent to the power consumption management control module

The EXTI wake-up requests allow the system to be woken up from Stop modes. The interrupt request and event request generation can also be used in Run mode.

EXTI allows the management of up to 21 configurable/direct event lines (19 configurable event lines and 2 direct event lines).

12.2.1. EXTI main features

- The system can wake up via GPIO and specified module (PVD/COMP/RTC/LPTIM) input events
- Configurable events (from I/Os, peripherals not having an associated interrupt pending status bit, or peripherals generating a pulse)
  - Selectable active trigger edge
  - Interrupt pending status bits
  - Individual interrupt and event generation mask

Puya Confidential

- SW trigger possibility

- Direct events (from peripherals having an associated flag and interrupt pending status bit)
  - Fixed rising edge active trigger
  - No interrupt pending bit in the EXTI
  - Individual interrupt and event generation mask
  - No SW trigger possibility
- Configurable IO port selector

12.2.2. EXTI block diagram

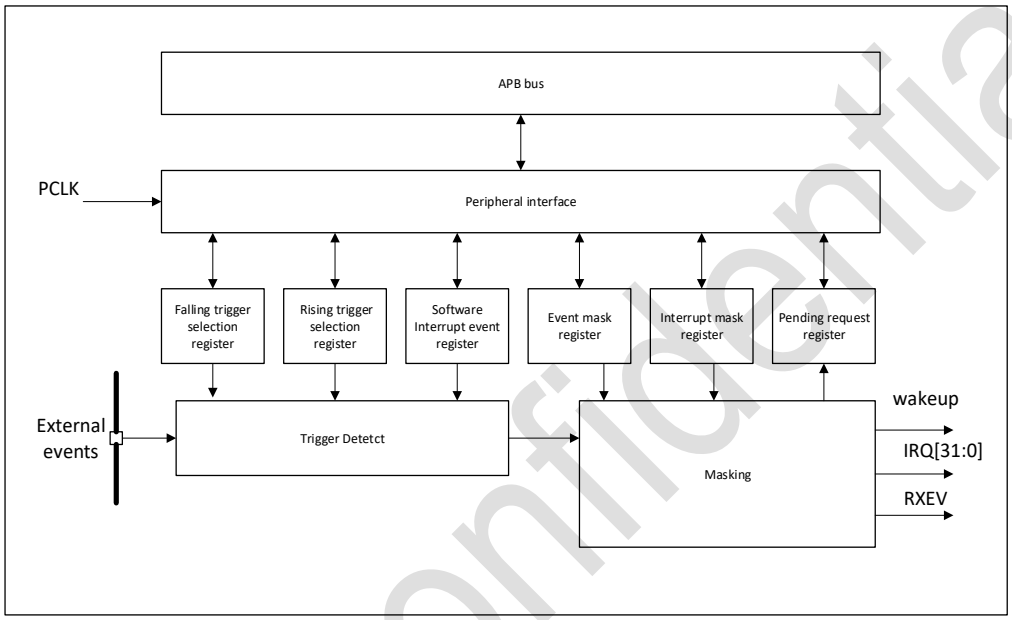


Figure 12-1 EXTI block diagram

12.2.3. Interrupt management

The wake up event can be generated either by:

- Interrupt mode

Enable interrupts in the peripheral control register, but not in the NVIC (nested vector interrupt controller), and enable the SEVONPEND bit in the Cortex-M0 system control register.

When the MCU resumes from the WFE state, the EXTI peripheral interrupt pending bit and the pending bit of the peripheral NVIC IRQ channel (located in the NVIC interrupt clear pending register) must be cleared.

- Event mode

Configure the external or internal EXTI line to event mode. When the CPU resumes from the WFE state, there is no need to clear the peripheral interrupt pending bit or the pending bit of the NVIC IRQ channel because the pending bit corresponding to the event line will not be set.

12.2.4. Functional description

For external interrupt lines, to generate interrupts, interrupt lines must be configured and enabled. This is done by programming the two trigger registers to set the required edge detection and by writing '1' to the corresponding bit in the interrupt mask register to enable the interrupt request. An interrupt

Puya Confidential

request is generated when the selected edge appears on the outer interrupt line. At the same time, the pending bit corresponding to the interrupt line is also set. This request may be cleared by writing '1' to the corresponding bit in the pending register.

For internal interrupt lines, the active edge is always a rising edge, the interrupt is enabled by default in the interrupt mask register, and there is no corresponding pending bit in the pending register.

In order to generate events, the event line needs to be configured and enabled. This is done by programming the two trigger registers to set the required edge detection, and by writing '1' to the corresponding bit in the event mask register to enable the event request. When the selected edge appears on the event line, an event pulse is generated. The pending bit corresponding to this event line will not be set.

For external lines, an interrupt or event request can also be generated by software writing '1' in the software interrupt/event register.

Note:Interrupts or events related to internal lines can only be triggered when the system is in Stop mode. If the system is still running, no interrupts or events are generated.

12.2.5. Hardware interrupt selection

Direct events will generate interrupts in the EXTI module and will generate event signals that wake up the system and CPU subsystem. When the CPU handles the interrupt generated by this type of trigger event, it needs to clear the interrupt status bit of the peripheral module.

To configure a line as an interrupt source, follow these steps:

- Configure the corresponding mask bit in the EXTI_IMR register:Enable this interrupt line by setting the corresponding bit in the EXTI_IMR (interrupt mask register).
- Configure trigger select bits for interrupt lines:Set the required edge detection (rising edge, falling edge, or both) by programming EXTI_RTSR (rising edge trigger select register) and EXTI_FTSR (falling edge trigger select register).
- Configure enable and mask bits for NVIC interrupt channels:The configuration controls the enable and mask bits of the NVIC IRQ channel mapped to EXTI so that interrupts from an EXTI line are correctly responded to.

12.2.6. Hardware event selection

To configure a line as an event source, follow these steps:

- Configure the corresponding mask bit in the EXTI_EMR register:This event line is enabled by setting the corresponding bit in the EXTI_EMR (event mask register).
- Configure the trigger select bit of the event line:Set the required edge detection (rising edge, falling edge, or both) by programming EXTI_RTSR and EXTI_FTSR.

12.2.7. Software interrupt/Event selection

Any external line can be configured as a software interrupt/event line. Here are the steps to generate a software interrupt:

Puya Confidential

- Configure corresponding mask bits:Configure the corresponding bits in EXTI_IMR (interrupt mask register) or EXTI_EMR (event mask register) as needed.
- Set the corresponding bit in the software interrupt register:Generate the software interrupt by setting the required bit in the EXTI_SWIER (software interrupt/event register

12.2.8. EXTI multiplexer

The GPIOs are connected to 16 external interrupt/event lines:

Puya Confidential

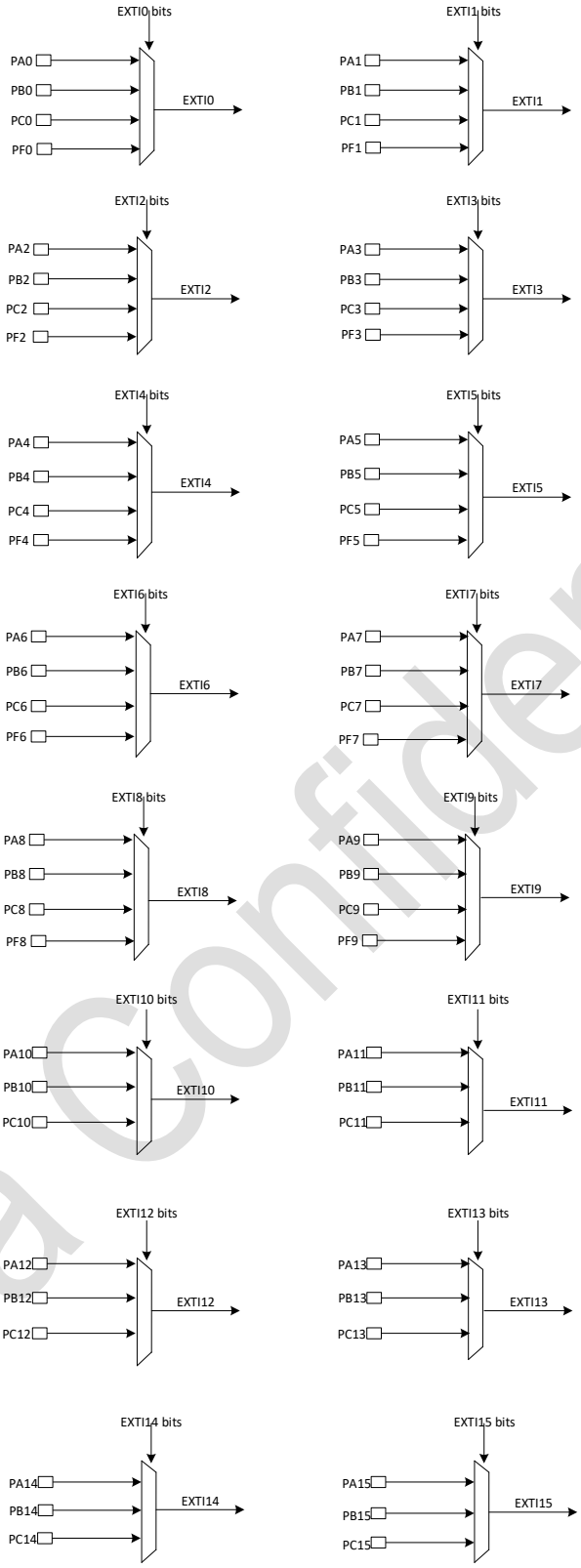


Figure 12-2 External interrupt/event GPIO mapping

The EXTI lines are connected as shown as follows:

Table 12-2 EXTI lines connections

EXTI line	Line source	Line type
Line 0-15	GPIO	Configurable
Line 16	PVD output	Configurable

Puya Confidential

EXTI line	Line source	Line type
Line 0-7	COMP 1 output	Configurable
Line 18	COMP2 output	Configurable
Line 19	RTC	Direct
Line 20	Reserved	-
Line 21 to 28	Reserved	-
Line 29	LPTIM	Direct
Line 30 to 31	Reserved	-

12.3. EXTI register

The peripheral registers have to be accessed by word (32-bit), half-word (16 bits), and byte (8 bits).

12.3.1. EXTI rising trigger selection register (EXTI_RTSR)

Address offset:0x00

Reset value:0x0000 0000

Contains only register bits for configurable events.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res											Res	Res	RT18	RT17	RT16
-											-	-	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:19	Reserved	-	-	Reserved
18	RT18	RW	0	Rising trigger event configuration bit of line18 0:Disabled 1:Enabled
17	RT17	RW	0	Rising trigger event configuration bit of line17 0:Disabled 1:Enabled
16	RT16	RW	0	Rising trigger event configuration bit of line16 0:Disabled 1:Enabled
15	RT15	RW	0	Rising trigger event configuration bit of line15 0:Disabled 1:Enabled
14	RT14	RW	0	Rising trigger event configuration bit of line14 0:Disabled 1:Enabled

Puya Confidential

Bit	Name	R/W	Reset Value	Function
13	RT13	RW	0	Rising trigger event configuration bit of line13 0:Disabled 1:Enabled
12	RT12	RW	0	Rising trigger event configuration bit of line12 0:Disabled 1:Enabled
11	RT11	RW	0	Rising trigger event configuration bit of line11 0:Disabled 1:Enabled
10	RT10	RW	0	Rising trigger event configuration bit of line10 0:Disabled 1:Enabled
9	RT9	RW	0	Rising trigger event configuration bit of line9 0:Disabled 1:Enabled
8	RT8	RW	0	Rising trigger event configuration bit of line8 0:Disabled 1:Enabled
7	RT7	RW	0	Rising trigger event configuration bit of line7 0:Disabled 1:Enabled
6	RT6	RW	0	Rising trigger event configuration bit of line6 0:Disabled 1:Enabled
5	RT5	RW	0	Rising trigger event configuration bit of line5 0:Disabled 1:Enabled
4	RT4	RW	0	Rising trigger event configuration bit of line4 0:Disabled 1:Enabled
3	RT3	RW	0	Rising trigger event configuration bit of line3 0:Disabled 1:Enabled
2	RT2	RW	0	Rising trigger event configuration bit of line2 0:Disabled 1:Enabled
1	RT1	RW	0	Rising trigger event configuration bit of line1 0:Disabled 1:Enabled

Puya Confidential

Bit	Name	R/W	Reset Value	Function
0	RT0	RW	0	Rising trigger event configuration bit of line0 0:Disabled 1:Enabled

The configurable wakeup lines are edge-triggered. No glitches must be generated on these lines. If a rising edge on a configurable interrupt line occurs during a write operation to the EXTI_RTSR register, the pending bit is not set.

Rising and falling edge triggers can be set for the same interrupt line. In this case, both generate a trigger condition.

12.3.2. EXTI falling trigger selection register (EXTI_FTSR)

Address offset:0x04

Reset value:0x0000 0000

Contains only register bits for configurable events.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res											Res	Res	FT18	FT17	FT16
-											-	-	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:19	Reserved	-	-	Reserved
18	FT18	RW	0	Falling trigger event configuration bit of line18 0:Disabled 1:Enabled
17	FT17	RW	0	Falling trigger event configuration bit of line17 0:Disabled 1:Enabled
16	FT16	RW	0	Falling trigger event configuration bit of line16 0:Disabled 1:Enabled
15	FT15	RW	0	Falling trigger event configuration bit of line15 0:Disabled 1:Enabled
14	FT14	RW	0	Falling trigger event configuration bit of line14 0:Disabled 1:Enabled
13	FT13	RW	0	Falling trigger event configuration bit of line13 0:Disabled

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:Enabled
12	FT12	RW	0	Falling trigger event configuration bit of line12 0:Disabled 1:Enabled
11	FT11	RW	0	Falling trigger event configuration bit of line11 0:Disabled 1:Enabled
10	FT10	RW	0	Falling trigger event configuration bit of line10 0:Disabled 1:Enabled
9	FT9	RW	0	Falling trigger event configuration bit of line9 0:Disabled 1:Enabled
8	FT8	RW	0	Falling trigger event configuration bit of line8 0:Disabled 1:Enabled
7	FT7	RW	0	Falling trigger event configuration bit of line7 0:Disabled 1:Enabled
6	FT6	RW	0	Falling trigger event configuration bit of line6 0:Disabled 1:Enabled
5	FT5	RW	0	Falling trigger event configuration bit of line5 0:Disabled 1:Enabled
4	FT4	RW	0	Falling trigger event configuration bit of line4 0:Disabled 1:Enabled
3	FT3	RW	0	Falling trigger event configuration bit of line3 0:Disabled 1:Enabled
2	FT2	RW	0	Falling trigger event configuration bit of line2 0:Disabled 1:Enabled
1	FT1	RW	0	Falling trigger event configuration bit of line1 0:Disabled 1:Enabled
0	FT0	RW	0	Falling trigger event configuration bit of line0 0:Disabled

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:Enabled

The configurable wakeup lines are edge-triggered. No glitches must be generated on these lines. If a falling edge on a configurable interrupt line occurs during a write operation to the EXTI_FTSR register, the pending bit is not set.

Rising and falling edge triggers can be set for the same interrupt line. In this case, both generate a trigger condition.

12.3.3. Software interrupt event register (EXTI_SWIER)

Address offset:0x08

Reset value:0x0000 0000

Contains only register bits for configurable events.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SW18	SW17	SW16
-	-	-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SW15	SW14	SW13	SW12	SW11	SW10	SW9	SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:19	Reserved	-	-	Reserved
18	SW18	RW	0	Rising trigger event configuration bit of line18 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
17	SW17	RW	0	Rising trigger event configuration bit of line17 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by the hardware. Read returns 0.
16	SW16	RW	0	Rising trigger event configuration bit of line16 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
15	SW15	RW	0	Rising trigger event configuration bit of line15 0:No effect 1:Generate rising trigger event, which in turn generates interrupt

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
14	SW14	RW	0	Rising trigger event configuration bit of line14 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by the hardware. Read returns 0.
13	SW13	RW	0	Rising trigger event configuration bit of line13 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by the hardware. Read returns 0.
12	SW12	RW	0	Rising trigger event configuration bit of line12 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
11	SW11	RW	0	Rising trigger event configuration bit of line11 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
10	SW10	RW	0	Rising trigger event configuration bit of line10 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
9	SW9	RW	0	Rising trigger event configuration bit of line9 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by the hardware. Read returns 0.
8	SW8	RW	0	Rising trigger event configuration bit of line8 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
7	SW7	RW	0	Rising trigger event configuration bit of line7 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)

Puya Confidential

Bit	Name	R/W	Reset Value	Function
6	SW6	RW	0	Rising trigger event configuration bit of line6 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
5	SW5	RW	0	Rising trigger event configuration bit of line5 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
4	SW4	RW	0	Rising trigger event configuration bit of line4 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
3	SW3	RW	0	Rising trigger event configuration bit of line3 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
2	SW2	RW	0	Rising trigger event configuration bit of line2 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
1	SW1	RW	0	Rising trigger event configuration bit of line1 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
0	SW0	RW	0	Rising trigger event configuration bit of line0 0:No effect 1:Generate rising trigger event, which in turn generates interrupt This bit is cleared by hardware, and read returns 0 (after hardware clearing) or configuration value (before hardware clearing)

12.3.4. EXTI pending register (EXTI_PR)

Address offset:0x0C

Reset value:0x0000 0000

Puya Confidential

Contains only register bits for configurable events.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PR18	PR17	PR16
-											-	-	RC_W1		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
RC_W1															

Bit	Name	R/W	Reset Value	Function
31:19	Reserved	-	-	-
18	PR18	RC_W1	0	Configurable EXTI line 18 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
17	PR17	RC_W1	0	Configurable EXTI line 17 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
16	PR16	RC_W1	0	Configurable EXTI line 16 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
15	PR15	RC_W1	0	Configurable EXTI line 15 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
14	PR14	RC_W1	0	Configurable EXTI line 14 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
13	PR13	RC_W1	0	Configurable EXTI line 13 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
12	PR12	RC_W1	0	Configurable EXTI line 12 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
11	PR11	RC_W1	0	Configurable EXTI line 11 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
10	PR10	RC_W1	0	Configurable EXTI line 10 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
9	PR9	RC_W1	0	Configurable EXTI line 9 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
8	PR8	RC_W1	0	Configurable EXTI line 8 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
7	PR7	RC_W1	0	Configurable EXTI line 7 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
6	PR6	RC_W1	0	Configurable EXTI line 6 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
5	PR5	RC_W1	0	Configurable EXTI line 5 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
4	PR4	RC_W1	0	Configurable EXTI line 4 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
3	PR3	RC_W1	0	Configurable EXTI line 3 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
2	PR2	RC_W1	0	Configurable EXTI line 2 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
1	PR1	RC_W1	0	Configurable EXTI line 1 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;
0	PR0	RC_W1	0	Configurable EXTI line 0 interrupt request pending flag When software or hardware generates rising/falling edge trigger events, this bit is set. This bit is cleared by writing 1 0:No trigger request occurred 1:Generate rising edge/falling edge/software trigger event request;

12.3.5. EXTI external interrupt selection register 1 (EXTI_EXTICR1)

Address offset:0x60

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	EXTI3 [1:0]		Res	Res	Res	Res	Res	Res	EXTI2 [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	EXTI1 [1:0]		Res	Res	Res	Res	Res	Res	EXTIO [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25:24	EXTI3 [1:0]	RW	0	EXIT3 GPIO port selection 2 'b00:PA [3] pin 2 'b01:PB [3] pin 2 'b10:PC [3] pin 2 'b11:PF [3] pin
23:18	Reserved	-	-	Reserved
17:16	EXTI2 [1:0]	RW	0	EXTI2 corresponds to GPIO port selection. 2 'b00:PA [2] pin 2 'b01:PB [2] pin 2 'b10:PC [2] pin 2 'b11:PF [2] pin
15:10	Reserved	-	-	Reserved
9:8	EXTI1 [1:0]	RW	0	EXTI1 corresponds to GPIO port selection. 2 'b00:PA [1] pin 2 'b01:PB [1] pin 2 'b10:PC [1] pin 2 'b11:PF [1] pin
7:2	Reserved	-	-	Reserved
1:0	EXTIO [1:0]	RW	0	EXTI0 corresponds to GPIO port selection. 2 'b00:PA [0] pin 2 'b01:PB [0] pin 2 'b10:PC [0] pin 2 'b11:PF [0] pin

12.3.6. EXTI external interrupt selection register 2 (EXTI_EXTICR2)

Address offset:0x64

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	EXTI7 [1:0]		Res	Res	Res	Res	Res	Res	EXTI6 [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	EXTI5 [1:0]		Res	Res	Res	Res	Res	Res	EXTI4 [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25:24	EXTI7 [1:0]	RW	0	EXTI7 corresponds to GPIO port selection. 2 'b00:PA [7] pin

Piya Confidential

Bit	Name	R/W	Reset Value	Function
				2 'b01:PB [7] pin 2 'b10:PC [7] pin 2 'b11:PF [7] pin
23:18	Reserved	-	-	Reserved
17:16	EXTI6 [1:0]	RW	0	EXTI6 corresponds to GPIO port selection. 2 'b00:PA [6] pin 2 'b01:PB [6] pin 2 'b10:PC [6] pin 2 'b11:PF [6] pin
15:10	Reserved	-	-	Reserved
9:8	EXTI5 [1:0]	RW	0	EXTI5 corresponds to GPIO port selection. 2 'b00:PA [5] pin 2 'b01:PB [5] pin 2 'b10:PC [5] pin 2 'b11:PF [5] pin
7:2	Reserved	-	-	Reserved
1:0	EXTI4 [1:0]	RW	0	EXTI4 corresponds to GPIO port selection. 2 'b00:PA [4] pin 2 'b01:PB [4] pin 2 'b10:PC [4] pin 2 'b11:PF [4] pin

12.3.7. EXTI external interrupt selection register 3 (EXTI_EXTICR3)

Address offset:0x68

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	EXTI11 [1:0]		Res	Res	Res	Res	Res	Res	EXTI10 [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	EXTI9 [1:0]		Res	Res	Res	Res	Res	Res	EXTI8 [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25:24	EXTI11 [1:0]	RW	0	EXTI11 corresponds to GPIO port selection. 2 'b00:PA [11] pin 2 'b01:PB [11] pin 2 'b10:PC [11] pin 2'b11:Reserved

PWC Confidential

23:18	Reserved	-	-	Reserved
17:16	EXTI10 [1:0]	RW	0	EXTI11 corresponds to GPIO port selection. 2 'b00:PA [10] pin 2 'b01:PB [10] pin 2 'b10:PC [10] pin 2'b11:Reserved
15:10	Reserved	-	-	Reserved
9:8	EXTI9 [1:0]	RW	0	EXTI11 corresponds to GPIO port selection. 2 'b00:PA [9] pin 2 'b01:PB [9] pin 2 'b10:PC [9] pin 2 'b11:PF [9] pin
7:2	Reserved	-	-	Reserved
1:0	EXTI8 [1:0]	RW	0	EXTI8 corresponds to GPIO port selection. 2 'b00:PA [8] pin 2 'b01:PB [8] pin 2 'b10:PC [8] pin 2 'b11:PF [8] pin

12.3.8. EXTI external interrupt selection register 4 (EXTI_EXTICR4)

Address offset:0x6C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	EXTI15 [1:0]		Res	Res	Res	Res	Res	Res	EXTI14 [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	EXTI13 [1:0]		Res	Res	Res	Res	Res	Res	EXTI12 [1:0]	
-	-	-	-	-	-	RW	RW	-	-	-	-	-	-	RW	RW

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25:24	EXTI15 [1:0]	RW	0	EXTI15 corresponds to GPIO port selection. 2 'b00:PA [15] pin 2 'b01:PB [15] pin 2 'b10:PC [15] pin 2'b11:Reserved
23:18	Reserved	-	-	Reserved
17:16	EXTI14 [1:0]	RW	0	EXTI14 corresponds to GPIO port selection. 2 'b00:PA [14] pin 2 'b01:PB [14] pin

PUYA Confidential

Bit	Name	R/W	Reset Value	Function
				2 'b10:PC [14] pin 2'b11:Reserved
15:10	Reserved	-	-	Reserved
9:8	EXTI13 [1:0]	RW	0	EXTI13 corresponds to GPIO port selection. 2 'b00:PA [13] pin 2 'b01:PB [13] pin 2 'b10:PC [13] pin 2'b11:Reserved
7:2	Reserved	-	-	Reserved
1:0	EXTI12 [1:0]	RW	0	EXTI12 corresponds to GPIO port selection. 2 'b00:PA [12] pin 2 'b01:PB [12] pin 2 'b10:PC [12] pin 2'b11:Reserved

12.3.9. Interrupt mask register (EXTI_IMR)

Address offset:0x80

Reset value:0x2008 0000

Note:The reset value is set such as to, by default, enable interrupt from direct lines, and disable interrupt from configurable lines.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	IM29	Res	Res	Res	Res	Res	Res	Res	Res	Res	IM19	IM18	IM17	IM16
-	-	RW	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29	IM29	RW	1	Interrupt mask on line 29. 0:Interrupt request is masked 1:Interrupt request is not masked
28:20	Reserved	-	-	Reserved
19	IM19	RW	1	EXTI line19 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
18	IM18	RW	0	Interrupt mask on line 18. 0:Interrupt request is masked 1:Interrupt request is not masked

Puya Confidential

Bit	Name	R/W	Reset Value	Function
17	IM17	RW	0	Interrupt mask on line 17. 0:Interrupt request is masked 1:Interrupt request is not masked
16	IM16	RW	0	EXTI line16 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
15	IM15	RW	0	EXTI line15 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
14	IM14	RW	0	EXTI line14 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
13	IM13	RW	0	EXTI line13 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
12	IM12	RW	0	EXTI line12 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
11	IM11	RW	0	EXTI line11 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
10	IM10	RW	0	EXTI line10 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
9	IM9	RW	0	EXTI line9 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
8	IM8	RW	0	EXTI line8 wakes up CPU mask control as an interrupt. 0:Interrupt request is masked 1:Interrupt request is not masked
7	IM7	RW	0	Interrupt mask on line 7. 0:Interrupt request is masked 1:Interrupt request is not masked
6	IM6	RW	0	Interrupt mask on line 6. 0:Interrupt request is masked 1:Interrupt request is not masked
5	IM5	RW	0	Interrupt mask on line 5. 0:Interrupt request is masked 1:Interrupt request is not masked

Puya Confidential

Bit	Name	R/W	Reset Value	Function
4	IM4	RW	0	Interrupt mask on line 4. 0:Interrupt request is masked 1:Interrupt request is not masked
3	IM3	RW	0	Interrupt mask on line 3. 0:Interrupt request is masked 1:Interrupt request is not masked
2	IM2	RW	0	Interrupt mask on line 2. 0:Interrupt request is masked 1:Interrupt request is not masked
1	IM1	RW	0	Interrupt mask on line 1. 0:Interrupt request is masked 1:Interrupt request is not masked
0	IM0	RW	0	Interrupt mask on line 0. 0:Interrupt request is masked 1:Interrupt request is not masked

12.3.10. EXTI event mask register (EXTI_EMR)

Address offset:0x84

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	EM29	Res	Res	Res	Res	Res	Res	Res	Res	Res	EM19	EM18	EM17	EM16
-	-	RW	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29	EM29	RW	0	Event mask on line 29. 0:Event request from Line 29 is masked 1:Event request from Line 29 is not masked
28:20	Reserved	-	-	Reserved
19	EM19	RW	0	Event mask on line 19. 0:Event request from Line 19 is masked 1:Event request from Line 19 is not masked
18	EM18	RW	0	Event mask on line 18. 0:Event request from Line 18 is masked

PY Confidential

Bit	Name	R/W	Reset Value	Function
				1:Event request from Line 18 is not masked
17	EM17	RW	0	Event mask on line 17. 0:Event request from Line 17 is masked 1:Event request from Line 17 is not masked
16	EM16	RW	0	Event mask on line 16. 0:Event request from Line 16 is masked 1:Event request from Line 16 is not masked
15	EM15	RW	0	Event mask on line 15. 0:Event request from Line 15 is masked 1:Event request from Line 15 is not masked
14	EM14	RW	0	Event mask on line 14. 0:Event request from Line 14 is masked 1:Event request from Line 14 is not masked
13	EM13	RW	0	Event mask on line 13. 0:Event request from Line 13 is masked 1:Event request from Line 13 is not masked
12	EM12	RW	0	Event mask on line 12. 0:Event request from Line 12 is masked 1:Event request from Line 12 is not masked
11	EM11	RW	0	Event mask on line 11. 0:Event request from Line 11 is masked 1:Event request from Line 11 is not masked
10	EM10	RW	0	Event mask on line 10. 0:Event request from Line 10 is masked 1:Event request from Line 10 is not masked
9	EM9	RW	0	Event mask on line 9. 0:Event request from Line 9 is masked 1:Event request from Line 9 is not masked
8	EM8	RW	0	Event mask on line 8. 0:Event request from Line 8 is masked 1:Event request from Line 8 is not masked
7	EM7	RW	0	Event mask on line 7. 0:Event request from Line 7 is masked 1:Event request from Line 7 is not masked
6	EM6	RW	0	Event mask on line 6. 0:Event request from Line 6 is masked 1:Event request from Line 6 is not masked
5	EM5	RW	0	Event mask on line 5. 0:Event request from Line 5 is masked

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:Event request from Line 5 is not masked
4	EM4	RW	0	Event mask on line 4. 0:Event request from Line 4 is masked 1:Event request from Line 4 is not masked
3	EM3	RW	0	Event mask on line 3. 0:Event request from Line 3 is masked 1:Event request from Line 3 is not masked
2	EM2	RW	0	Event mask on line 2. 0:Event request from Line 2 is masked 1:Event request from Line 2 is not masked
1	EM1	RW	0	Event mask on line 1. 0:Event request from Line 1 is masked 1:Event request from Line 1 is not masked
0	EM0	RW	0	Event mask on line 0. 0:Event request from Line 0 is masked 1:Event request from Line 0 is not masked

Puya Confidential

13. Cyclic redundancy check calculation unit (CRC)

13.1. Introduction

The CRC (cyclic redundancy check) calculation unit is used to get a CRC code from 32-bit data word and a generator polynomial.

13.2. CRC main features

- The default polynomial value is the CRC-32 (Ethernet) polynomial:0x4C11DB7.
  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Supports 32-bit data input
- Single input/output 32-bit data register
- General-purpose 8-bit register (can be used for temporary storage)
- CRC computation done in 4 AHB clock cycles (HCLK) for the 32-bit data size

Note:Since the calculation time of CRC is 4 AHB clocks, for the correctness of the calculation, it is necessary to write data at an interval of 4 AHB clocks. Continuous writing of data using DMA is not recommended.

13.3. CRC functional description

13.3.1. CRC block diagram

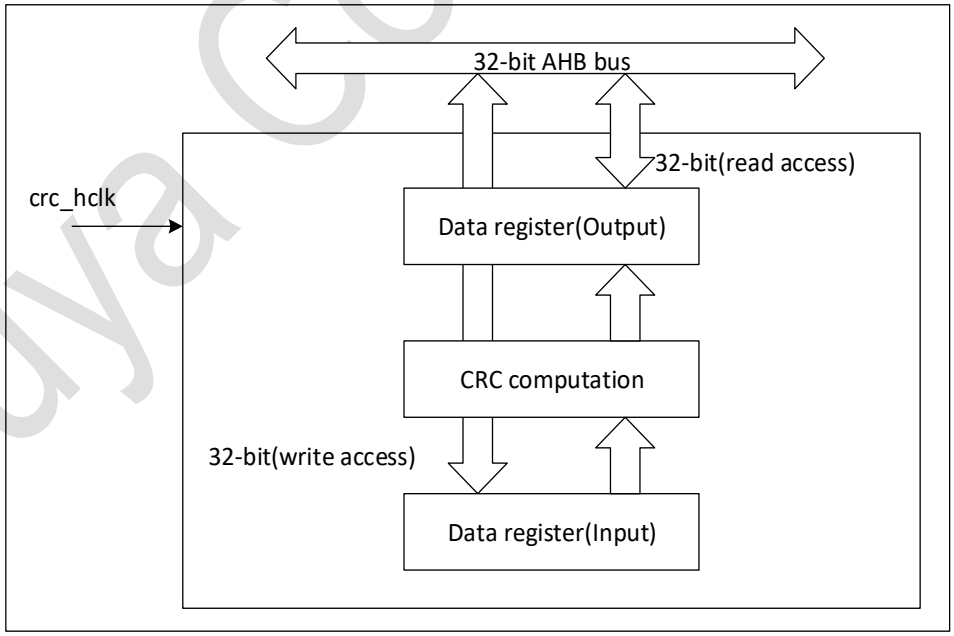


Figure 13-1 CRC calculation unit block diagram

The CRC calculation unit has a single 32-bit data register (CRC_DR).

- When writing to this register, it serves as an input register, allowing you to input new data for CRC calculation.

Puya Confidential

- When reading from this register, it returns the result of the previous CRC calculation.

Each time data is written to the register, the calculation result is a combination of the previous CRC calculation result and the new one (CRC calculation is performed on the entire 32-bit word rather than byte by byte).

Support configuration of the initial CRC value

You can reset the register CRC_DR to 0xFFFFFFFF by setting the RESET bit in the register CRC_CR. This operation does not affect the data in the register CRC_IDR

13.4. CRC registers

13.4.1. CRC data register (CRC_DR)

Address offset:0x00

Reset value:0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:0	DR	RW	0xFFFF FFFF	Data register. This register is used to write new data to the CRC calculator It holds the previous CRC calculation result when it is read.

13.4.2. CRC independent data register (CRC_IDR)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	IDR [7:0]							
-	-	-	-	-	-	-	-	RW							

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7:0	IDR [7:0]	RW	0	General purpose 8-bit data register bits. These bits can be used as a temporary storage location for four bytes.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This register is not affected by CRC resets generated by the RESET bit in the CRC_CR register. Note:This register does not participate in CRC calculation and can store any data.

13.4.3. CRC control register (CRC_CR)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RESET
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	W

Bit	Name	R/W	Reset Value	Function
31:1	Reserved	-	-	Reserved
0	RESET	W	0	This bit is set by software to reset the CRC calculation unit The software can only write 1, which is cleared by the hardware.

Puya Confidential

14. Analog-to-digital converters (ADC)

14.1. Introduction

The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 21 multiplexed channels allowing it to measure signals from 16 external and 5 internal sources. A/D conversion of the various channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored in a left-aligned or right-aligned 16-bit data register.

The analog watchdog feature allows the application to detect if the input voltage goes outside the user-defined higher or lower thresholds.

14.2. ADC main features

- High performance
  - 12-bit, 10-bit, 8-bit or 6-bit configurable resolution
  - ADC conversion time:1 μs @ 12-bit (1 Msps)
  - Self-calibration
  - Programmable sampling time
  - Programmable data alignment mode
  - DMA support
- Analog input channels
  - 16 external analog inputs
  - 1 channel for internal temperature sensor (T_{SENSOR})
  - 1 channel for internal reference voltage (V_{REFINT})
  - 1 internal reference voltage input channel (V_{REFBUF})
  - 2 internal OPA input voltage channels
- Start-of-conversion can be initiated
  - By software
  - By hardware triggers (TIM1, TIM2, TIM3, TIM15, or GPIO)
- Conversion modes
  - Single shot mode:a single channel can be converted
  - Scan mode:can scan a sequence of channels
  - Continuous mode:converts selected inputs continuously
  - Discontinuous mode:each trigger to convert subsequence channels, multiple triggers until the complete sequence is converted
- Interrupt generation
  - End of conversion
  - Analog watchdog
- Analog watchdog

Puya Confidential

14.3. ADC functional description

14.3.1. ADC block diagram

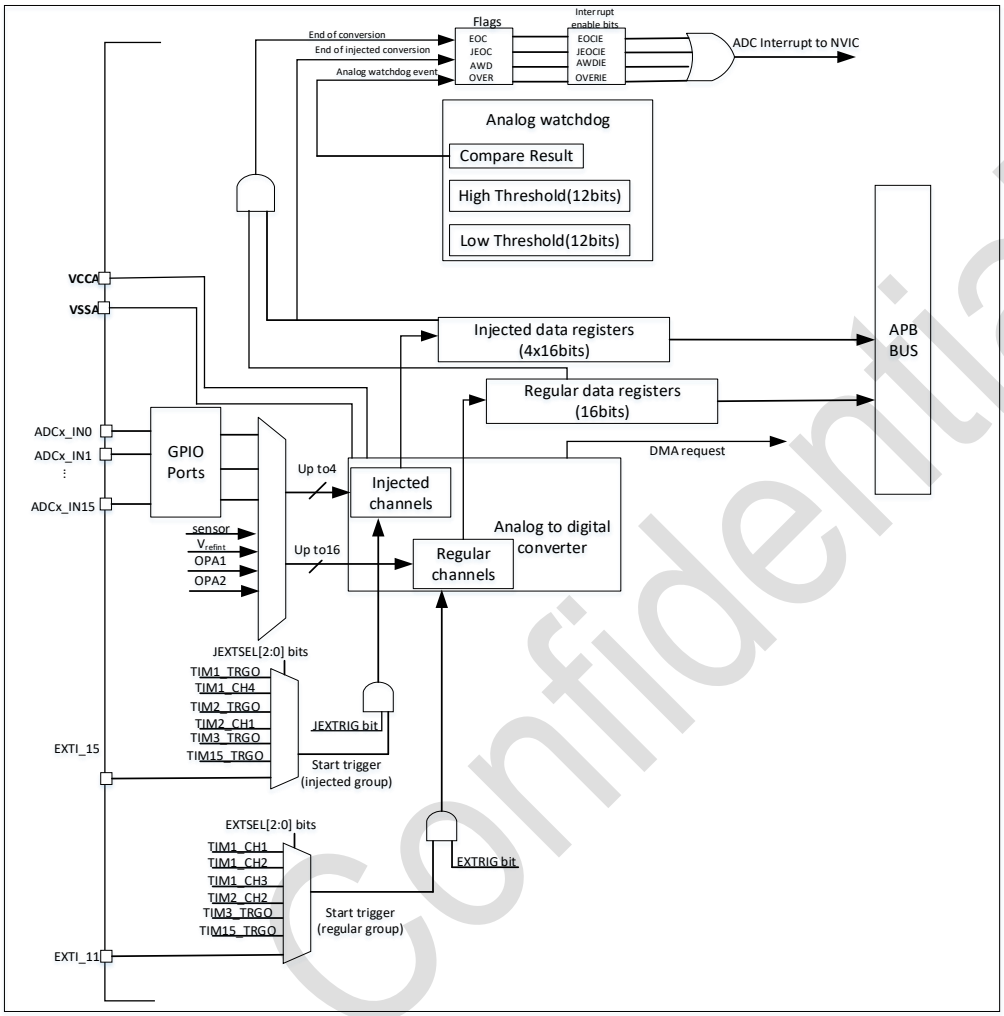


Figure 14-1 ADC block diagram

14.3.2. Calibration

The ADC has a calibration feature. During the procedure, the ADC calculates a calibration factor which is internally applied to the ADC until the next ADC power-off (lost after the ADC is powered down). The application cannot use the ADC module during the ADC calibration and until the calibration is complete.

Calibration is preliminary to any ADC operation. It removes the offset error which may vary from chip to chip due to process.

The software sets ADC_CR2.CAL = 1 to start the calibration. The calibration can only be started when the ADC is not enabled (ADC_CR2.ADON = 0), and only the system clock is supported as the clock of the ADC. When the calibration is complete, the CAL is cleared to 0 by the hardware.

When the operating conditions of the ADC change (V_{CCA} change is the main factor of ADC offset shift, followed by temperature change), it is recommended to perform a recalibration operation.

Software procedure to calibrate the ADC:

Puya Confidential

- Confirm ADON = 0
- Set CAL=1
- Wait until CAL = 0

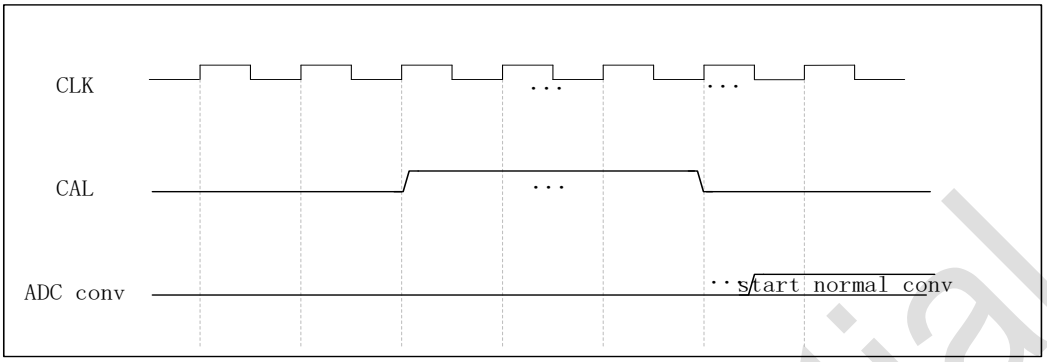


Figure 14-2 ADC calibration timing diagram

14.3.3. ADC on-off control

The ADC can be powered-on by setting the ADON bit in the ADC_CR2 register. When the ADON bit is set for the first time, it wakes up the ADC from power down mode.

After the ADC power-on delay for a period of time (t_{STAB}, not less than 1μs), the conversion begins.

The conversion can be stopped, and the ADC put in power down mode by resetting the ADON bit.

The conversion can be stopped by clearing the ADON bit and the ADC is put in power-down mode.

14.3.4. ADC clock

The ADCCLK clock provided by the clock controller is synchronous with the PCLK (APB clock). The RCC controller (CLK controller) provides a dedicated programmable prescaler for the ADC clock, and the ADCCLK clock division is detailed in RCC_CR.ADC_DIV [22:21].

14.3.5. Configuring the ADC

The ADC_CRx (x = 1, 2), ADC_SMPRx (x = 1, 2, 3), ADC_SQRx (x = 1, 2, 3), ADC_JQR, ADC_HTR, and ADC_LTR registers must be rewritten by the software during the ADC no transition period.

14.3.6. Channel selection

There are 16 external channels and 5 internal channels, of which the internal channels are:

- Temperature sensor/V_{REFINT} internal channel

The temperature sensor is connected to the channel ADC_IN23, and the internal reference voltage V_{REFINT} is connected to ADC_IN17.

- V_{CC} /3

V_{CCA}/3 is connected to channel ADC_IN18.

- OPA

OPA1_VIN is connected to channel ADC_IN21, OPA2_VIN is connected to channel ADC_IN22.

Conversions can be organized into two groups:regular groups and injected groups. A group consists of a sequence of conversions which can be done on any channel and in any order. For instance, it is possible to do the conversion in the following order:Ch3, Ch8, Ch2, Ch2, Ch0, Ch2, Ch2, Ch15.

Puya Confidential

The regular group is composed of up to 16 conversions. The regular channels and their order in the conversion sequence must be selected in the ADC_SQRx registers. The total number of conversions in the regular group must be written in the L [3:0] bits in the ADC_SQR1 register.

The injected group is composed of up to 4 conversions. The injected channels and their order in the conversion sequence must be selected in the ADC_JSQR register. The total number of conversions in the injected group must be written in the JL [1:0] bits in the ADC_JSQR register.

If the ADC_SQRx or ADC_JSQR registers are modified during a conversion, the current conversion is reset and a new start pulse is sent to the ADC to convert the new chosen group.

14.3.7. Programmable sampling time

The ADC samples the input voltage using several ADC_CLK cycles, and the number of sampling cycles can be changed by SMP [2:0] bits in the ADC_SMPR1, ADC_SMPR2, and ADC_SMPR3 registers. Each channel may be sampled separately at a different time.

The total conversion time is calculated as follows:

$$t_{CONV}$$ = sampling time + 12.5 cycles (RESSEL = 00B)

For example:

When f_{ADC} = 16 MHz, sampling time is 3.5 cycles

t_{CONV} = 3.5 + 12.5 = 16 Cycle = 1 μs

14.3.8. Configurable resolution

Fast conversion can be performed by reducing the ADC resolution. The RESSEL bit of the ADC_CR1 register is used to select the number of bits available in the data register. The minimum conversion time for each resolution is as follows:

- 12 bits:3.5 + 12.5 = 16 ADCCLK cycles
- 10 bits:3.5 + 10.5 = 14 ADCCLK cycles
- 8-bit:3.5 + 8.5 = 12 ADCCLK cycles
- 6 bits:3.5 + 6.5 = 10 ADCCLK cycles

14.3.9. Single conversion mode

In Single conversion mode, the ADC performs once all the conversions of the channels. When EXTTRIG/JEXTTRIG is enabled, an external event (e.g., timer capture, EXTI interrupt, software trigger) triggers a start conversion (for regular or injected channels), at which time the CONT bit is 0.

Once the conversion of the selected channel is complete:

- If a regular channel was converted:
  - The converted data is stored in the 16-bit ADC_DR register
  - The EOC (end of conversion) flag is set
  - An interrupt is generated if EOCIE is set.
- If an injected channel is converted:
  - The converted data is stored in the 16-bit ADC_JDRx register
  - The JEOC (end of conversion injected) flag is set

Puya Confidential

— An interrupt is generated if the JEOCIE bit is set.

Then the ADC stops.

14.3.10. Continuous conversion mode

In continuous conversion mode ADC starts another conversion as soon as it finishes one. When EXTTRIG/JEXTTRIG is enabled, an external event (e.g. timer catch, EXTI interrupt, software trigger) triggers a start transition (for regular channels; injected channels behave as a single transition mode), at which time the CONT bit is 1.

After each conversion:

- If a regular channel was converted:
  - The converted data is stored in the 16-bit ADC_DR register
  - The EOC (end of conversion) flag is set
  - An interrupt is generated if EOCIE is set.
- If an injected channel is converted:
  - The EOC (end of conversion) flag is set
  - The converted data is stored in the 16-bit ADC_JDRx register
  - The JEOC (end of conversion injected) flag is set
  - An interrupt is generated if the JEOCIE bit is set.

14.3.11. Scan mode

This mode is used to scan a group of analog channels

Scan mode can be selected by setting the SCAN bit in the ADC_CR1 register. Once this bit is set, ADC scans all the channels selected in the ADC_SQRx registers (for regular channels) or in the ADC_JSQR (for injected channels). A single conversion is performed for each channel of the group. After each end of conversion, the next channel of the group is converted automatically. If the CONT bit is set, conversion does not stop at the last selected group channel but continues again from the first selected group channel.

If the DMA bit is set, the DMA controller transfers the translation data of the rule group channel to the SRAM. The injected channel converted data is always stored in the ADC_JDRx registers.

14.3.12. Discontinuous mode

14.3.12.1. Regular group

This mode is enabled by setting the DISCEN bit in the ADC_CR1 register. It can be used to convert a short sequence of n conversions (n≤8) which is a part of the sequence of conversions selected in the ADC_SQRx registers. The value of n is specified by writing to the DISCNUM [2:0] bits in the ADC_CR1 register.

When an external trigger occurs, it starts the next n conversions selected in the ADC_SQRx registers until all the conversions in the sequence are done. The total sequence length is defined by the L [3:0] bits in the ADC_SQR1 register.

Examples:

Puya Confidential

n = 3, channels to be converted = 0, 1, 2, 3, 6, 7, 9, 10

First trigger:the sequence converted 0, 1, 2

Second trigger:the sequence converted 3, 6, 7

Third trigger:the sequence converted 9, 10. An EOC event is generated.

Fourth trigger:the sequence converted 0, 1, 2

Note:When a regular group is converted in discontinuous mode, no rollover will occur.

When all sub groups are converted, the next trigger starts conversion of the first sub-group. In the example above, the fourth trigger reconverts the first sub-group channels 0, 1 and 2.

14.3.12.2. Injected group

This mode is enabled by setting the JDISCEN bit in the ADC_CR1 register. It can be used to convert the sequence selected in the ADC_JSQR register, channel by channel, after an external trigger event.

When an external trigger occurs, it starts the next channel conversions selected in the ADC_JSQR registers until all the conversions in the sequence are done.

The total sequence length is defined by the JL [1:0] bits in the ADC_JSQR register.

Example:

n = 1, channels to be converted = 1, 2, 3

First trigger:Channel 1 converted

Second trigger:Channel 2 converted

Third trigger:Channel 3 converted and EOC and JEOC events generated

Fourth trigger:Channel 1 converted

Note:When all injected channels are converted, the next trigger starts the conversion of the first injected channel. In the example above, the fourth trigger reconverts the first injected channel 1.

It is not possible to use both auto-injected and discontinuous modes simultaneously.

14.3.13. Injected channel management

The external trigger of the injected channel has a higher priority than the external trigger of the regular channel, that is, the external trigger of the injected channel can interrupt the ongoing regular channel transition. There are two ways to inject the channel:triggered injection and auto-injection.

14.3.13.1. Triggered injection

To use triggered injection, the JAUTO bit must be cleared and SCAN bit must be set in the ADC_CR1 register.

- Start conversion of a group of regular channels either by external trigger or by setting the ADON bit in the ADC_CR2 register.
- If an external injected trigger occurs during the regular group channel conversion, the current conversion is reset and the injected channel sequence is converted in single scan mode.
- Then, the regular group channel conversion is resumed from the last interrupted regular conversion. If a regular event occurs during an injected conversion, it doesn’t interrupt it but the regular sequence is executed at the end of the injected sequence.

Puya Confidential

- Discontinuous mode does not support triggered injection. Regular triggers during trigger injection are non-responsive.

Note:When using triggered injection, the interval between trigger events must be longer than the injection sequence. For instance, if the sequence length is 28 ADC clock cycles (that is two conversions with a 1.5 clock-period sampling time), the minimum interval between triggers must be 29 ADC clock cycles.

14.3.13.2. Auto-injection

If the JAUTO bit is set, then the injected group channels are automatically converted after the regular group channels. This can be used to convert a sequence of up to 20 conversions programmed in the ADC_SQRx and ADC_JSQR registers.

In this mode, external trigger on injected channels must be disabled.

If the CONT bit is also set in addition to the JAUTO bit, regular channels followed by injected channels are continuously converted.

Note:It is not possible to use both auto-injected and discontinuous modes simultaneously.

14.3.14. Stopping an ongoing conversion (ADSTP)

The software can decide to stop any ongoing conversions by setting ADSTP=1 in the ADC_CR1 register. This will reset the ADC operation and the ADC will be idle, ready for a new operation.

When the ADSTP bit is set by software, any ongoing conversion is aborted and the result is discarded (ADC_DR register is not updated with the current conversion).

The scan sequence is also aborted and reset (meaning that restarting the ADC would restart a new sequence).

Once this procedure is complete, the ADSTP and SWSTART bits are both cleared by hardware.

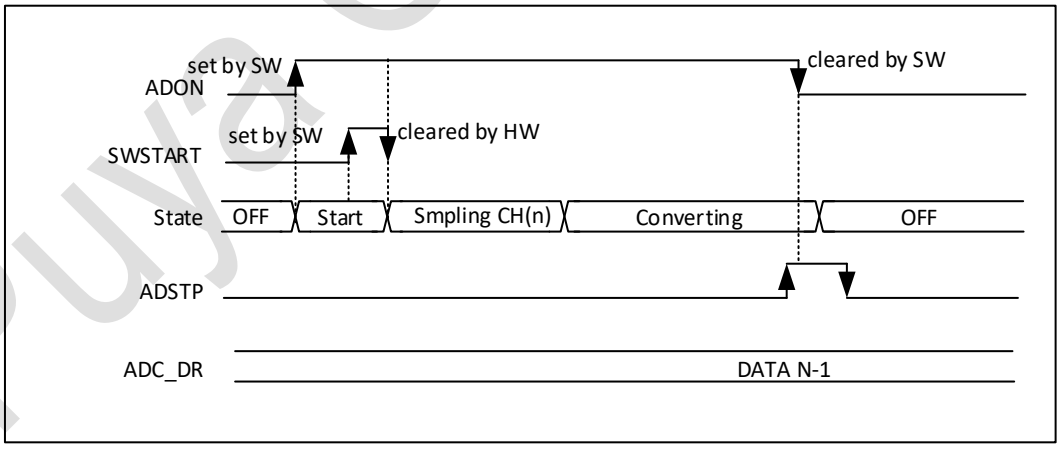


Figure 14-3 Stop timing

14.3.15. Analog watchdog

The AWD analog watchdog status bit is set if the analog voltage converted by the ADC is below a lower threshold or above a higher threshold. These thresholds are programmed in the 12 least significant bits of the ADC_HTR and ADC_LTR registers. An interrupt can be enabled by using the AWDIE bit in the ADC_CR1 register.

Puya Confidential

The threshold value is independent of the alignment selected by the ALIGN bit in the ADC_CR2 register. The threshold comparison is done before the alignment (before the injection channel is subtracted from the offset value).

The analog watchdog can be enabled on one or more channels by configuring the ADC_CR1 register as shown in the table below:

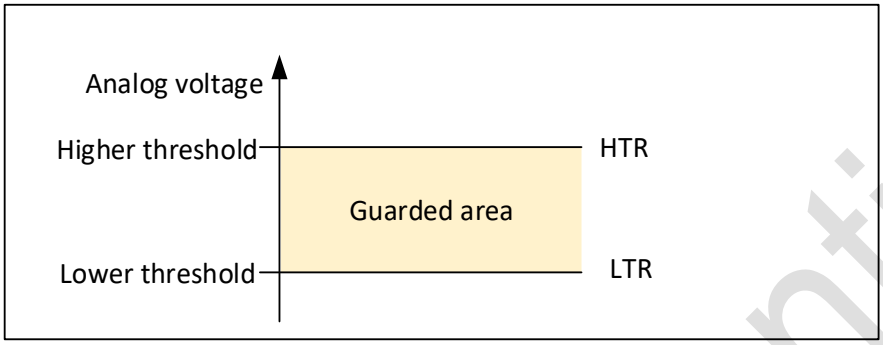


Figure 14-4 Analog watchdog guarded area

Table 14-1 Analog watchdog channel selection

Channels to be guarded by analog watchdog	ADC_CR1 register control bits		
	AWDSGL	AWDEN	JAWDEN
None	X	0	0
All injected channels	0	0	1
All regular channels	0	1	0
All injected and regular channels	0	1	1
Single injected channel	1	0	1
Single regular channel	1	1	0
Single injected or regular channel	1	1	1

14.3.16. Conversion on external trigger

Conversion can be triggered by an external event (e.g. timer capture, EXTI line). If the EXTTRIG or JEXTTRIG control bit is set, then external events are able to trigger a conversion. The EXTSEL [2:0] and JEXTSEL [2:0] control bits allow the application to select which out of 8 possible events can trigger conversion for the regular and injected groups.

Note:When an external trigger is selected for ADC regular or injected conversion, only the rising edge of the signal can start the conversion.

The table below gives possible external triggers for the conversion. Software source trigger events can be generated by setting the ADSTART bit in the ADC_CR register.

Table 14-2 ADC for external triggering of regular channels

Source	Type	EXTSEL [2:0]
TIM1_CC1 event	Internal signal from on-chip timers	000
TIM1_CC2 event		001

Puya Confidential

Source	Type	EXTSEL [2:0]
TIM1_CC3 event		010
TIM2_TRGO event		011
TIM3_TRGO event		100
TIM15_TRGO event		101
EXTI line 11	External pin	110
SWSTART	Software control bit	111

Table 14-3 External trigger for injected channels for ADC

Source	Type	JEXTSEL [2:0]
TIM1_TRGO event	Internal signal from on-chip timers	000
TIM1_CC4 event		001
TIM2_TRGO event		010
TIM2_CC1 event		011
TIM3_CC4 event		100
TIM15_TRGO event		101
EXTI line 15 output	External pin	110
JSWSTART	Software control bit	111

14.3.17. Data alignment

At the end of each conversion, the result of the converted data is stored in the ADC_DR data register which is 16-bit wide. ALIGN bit in the ADC_CR2 register selects the alignment of data stored after conversion. Data can be left or right aligned. The injected group channels converted data value is decreased by the user-defined offset written in the ADC_JOFRx registers so the result can be a negative value. The SEXT bit is the extended sign value.

For regular group channels no offset is subtracted so only twelve bits are significant.

Table 14-4 Right alignment of data

Injected group

SEXT	SEXT	SEXT	SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Regular group

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Table 14-5 Left alignment of data

Injected group

SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0

Regular group

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

14.3.18. Data overload

The overrun flag (OVER) indicates a data overrun event, when the converted data was not read in time by the CPU or the DMA, before the data from a new conversion is available.

Puya Confidential

14.3.19. DMA request

Since converted regular channels value are stored in a unique data register, it is necessary to use DMA for conversion of more than one regular channel. This avoids the loss of data already stored in the ADC_DR register.

Only the end of conversion of a regular channel generates a DMA request, which allows the transfer of its converted data from the ADC_DR register to the destination location selected by the user.

14.3.20. Temperature sensor and internal reference voltage

The temperature sensor can be used to measure the ambient temperature (T_A) of the device. The temperature sensor is internally connected to the ADCx_IN23 input channel which is used to convert the sensor output voltage into a digital value. The recommended sampling time for the temperature sensor is 17.1 μs. When not in use, the sensor can be put in power down mode.

The temperature sensor output voltage changes linearly with temperature. The offset of this line varies from chip to chip due to process variations (up to 45°C from one chip to another). The internal temperature sensor is more suited to applications that detect temperature variations instead of absolute temperatures. If accurate temperature readings are needed, an external temperature sensor part should be used.

Note:The TSVREFE bit must be set to enable both internal channels:ADC_IN23 (temperature sensor) and ADC_IN17 (V_{REFINT}) conversion.

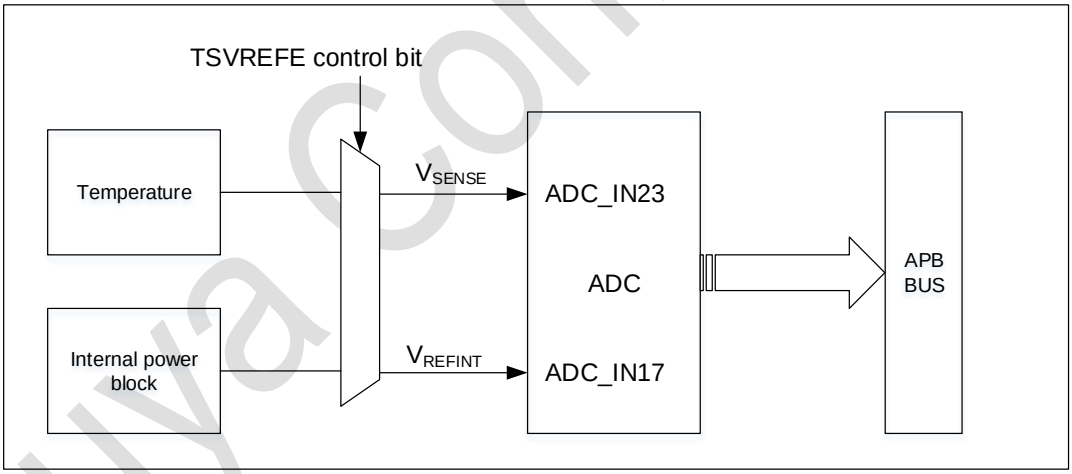


Figure 14-5 Temperature sensor channel block diagram

Reading the temperature

To use the sensor:

1. Select the ADC_IN23 input channel
2. Select a sampling time that is greater than the minimum sampling time (SMP23) specified in the datasheet.
3. Set the TSVREFE bit in the ADC_CR2 register to wake up the temperture sensor from power down mode.
4. Start the ADC conversion by setting the ADON bit (or by external trigger).
5. Read the resulting VSENSE data in the ADC data register

Puya Confidential

6. Calculate the temperature using the following formula:

Temperature (°C) = {(V_{SENSE}-V30)/Avg_Slope} + 30 °C

with:

V_{30} = V_{SENSE} value at 30 °C

Avg_Slope = Average Slope for curve between Temperature vs. V_{SENSE} (given in mV/° C or µV/ °C).

(For information on actual values of V30 and Avg_Slope, see the Electrical Characteristics section in the datasheet.)

Note:The sensor has a startup time after waking from power down mode before it can output V_{SENSE} at the correct level. The ADC also has a startup time after power-on, so to minimize the delay, the ADON and TSVREFE bits should be set at the same time.

14.4. ADC interrupts

An interrupt can be produced on end of conversion for regular and injected groups and when the analog watchdog status bit is set. An interrupt can also be generated when the regular group transition data is not read in time. Separate interrupt enable bits are available for flexibility.

Two other flags are present in the ADC_SR register, but there is no interrupt associated with them:

- JSTRT (Start of conversion for injected group channels)
- STRT (Start of conversion for regular group channels)

Table 14-6 ADC interrupts

Interrupt event	Event flag	Enable control bit
End of conversion regular group	EOC	EOCIE
End of conversion injected group	JEOC	JEOCIE
Analog watchdog status bit is set	AWD	AWDIE
Overrun flag	OVER	OVERIE

14.5. ADC registers

14.5.1. ADC status register (ADC_SR)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res										OVER	STRT	JSTRT	JEOC	EOC	AWD
-										RC	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	OVER	RC	0	ADC overrun

Priya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is set by hardware when an overrun occurs. When the EOC flag is set, a new conversion has been completed. 0:No overrun occurred (DMA or CPU has read the last conversion result) 1:Overrun has occurred
4	STRT	RC_W0	0	Regular channel Start flag This bit is set by hardware when regular channel conversion starts. It is cleared by software. 0:No regular channel conversion started 1:Regular channel conversion has started
3	JSTRT	RC_W0	0	Injected channel Start flag This bit is set by hardware when injected channel group conversion starts. It is cleared by software. 0:No injected group conversion started 1:injected group conversion has started
2	JEOC	RC_W0	0	Injected channel end of conversion This bit is set by hardware at the end of all injected group channel conversion. It is cleared by software. 0:Conversion is not complete 1:Conversion complete
1	EOC	RC_W0	0	End of conversion This bit is set by hardware at the end of a group channel conversion (regular or injected). It is cleared by software or by reading the ADC_DR. 0:Conversion is not complete 1:Conversion complete
0	AWD	RC_W0	0	Analog watchdog flag This bit is set by hardware when the converted voltage crosses the values programmed in the ADC_LTR and ADC_HTR registers. It is cleared by software. 0:No Analog watchdog event occurred 1:Analog watchdog event occurred

14.5.2. ADC control register 1 (ADC_CR1)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res		OVRIE	Res	AD-STP	Res	RESSEL [1:0]		AWD EN	JAWD EN	Res					

Puya Confidential

-		RW	-	RC_W1	-	RW		RW	RW	-					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DISCNUM [2:0]			JDISCEN	DIS-CEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH [4:0]				
RW			RW	RW	RW	RW	RW	RW	RW	RW	RW				

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29	OVRIE	RW	0	OVER FLAG interrupt enable 0:overload interrupt disabled 1:overload interrupt enabled
28	Reserved	-	-	Reserved
27	ADSTP	RC_W1	0	ADC stop conversion enabled. This bit is set by software. When the ADC stop signal is received, the hardware sets 0. 1:Stop ADC conversion 0:ADC conversion is not stopped
26	Reserved	-	-	Reserved
25:24	RESSEL [1:0]	RW	0	Resolution control bit. These bits are written by software to select the resolution of the conversion. 00:12 bits (16 ADCCLK cycles) 01:10 bit (14 ADCCLK cycles) 10:8 bits (12 ADCCLK cycles) 11:6 bits (10 ADCCLK cycles)
23	AWDEN	RW	0	Analog watchdog enable on regular channels Turn on the simulated watchdog on the regular channel. This bit is set/reset by software. 0:Analog watchdog disabled on regular channels 1:Analog watchdog enabled on injected channels
22	JAWDEN	RW	0	Analog watchdog enable on injected channels Turn on the analog watchdog on the injection channel. This bit is set/reset by software. 0:Analog watchdog disabled on injected channels 1:Analog watchdog enabled on injected channels
21:16	Reserved	-	-	Reserved
15:13	DISCNUM [2:0]	RW	0	Discontinuous mode channel count These bits are written by software to define the number of regular channels to be converted in discontinuous mode, after receiving an external trigger. These bits are set by software.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				000:1 channel 001:2 channels …… 111:8 channels
12	JDISCEN	RW	0	Discontinuous mode on injected channels. This bit set and cleared by software to enable/disable discontinuous mode on injected group channels 0:Discontinuous mode on injected channels disabled 1:Discontinuous mode on injected channels enabled
11	DISCEN	RW	0	Discontinuous mode on regular channels This bit set and cleared by software to enable/disable discontinuous mode on regular channels. 0:Discontinuous mode on regular channels disabled 1:Discontinuous mode on regular channels enabled
10	JAUTO	RW	0	Automatic injected group conversion This bit set and cleared by software to enable/disable automatic injected group conversion after regular group conversion. 0:Automatic injected group conversion disabled 1:Automatic injected group conversion enabled
9	AWDSGL	RW	0	Enable the watchdog on a single channel This bit set and cleared by software to enable/disable the analog watchdog on the channel identified by the AWDCH [4:0] bits. 0:Analog watchdog enabled on all channels 1:Analog watchdog enabled on a single channel
8	SCAN	RW	0	Scan mode This bit is set and cleared by software to enable/disable Scan mode. In Scan mode, the inputs selected through the ADC_SQRx or ADC_JSQRx registers are converted. 0:scan mode disabled 1:scan mode enabled
7	JEOCIE	RW	0	Interrupt enable for injected channels This bit is set/reset by software. After this bit is enabled, an interrupt request is generated when JEOC is valid. 0:JEOC interrupt disabled 1:JEOC interrupt enabled
6	AWDIE	RW	0	Analog watchdog interrupt enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is set/reset by software. This bit is set and cleared by software to enable/disable the analog watchdog interrupt. 0:Analog watchdog interrupt disabled 1:Analog watchdog interrupt enabled
5	EOCIE	RW	0	Interrupt enable for EOC This bit is set/reset by software. After this bit is enabled, an interrupt request is generated when the EOC is valid. 0:EOC interrupt disabled 1:EOC interrupt enabled.
4:0	AWDCH [4:0]	RW	0	Analog watchdog channel select bits These bits are set and cleared by software. They select the input channel to be guarded by the Analog watchdog. 00000:ADC analog input Channel 0 00001:ADC analog input Channel 1 …… 01111:ADC analog input Channel 15 10000: Reserved 10001:V_{REFINT} input 10010:V_{CCA}/3 10011:Reserved 10100 Reserved 10101:OPA1_VIN 10110:OPA2_VIN 10111:TS_VIN input All other numerical values are reserved.

14.5.3. ADC control register 2 (ADC_CR2)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res				VREFBUF_SEL		VREF BUFFERE	Res	TSVREFE	SWSTART	JSWSTART	EXTTRIG	EXTSEL [2:0]			Res
-				RW		RW	-	RW	RC_W1	RC_W1	RW	RW			-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JEXTTRIG	JEXTSEL [2:0]			ALIGN	Res	Res	DMA	Res	Res	Res	Res	RSTCAL	CAL	CONT	ADON
RW					-	-	RW	-				RC_W1		RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:28	Reserved	-	-	Reserved
27:26	VREFBUF_SEL	RW	0	V_{REFBUF} output voltage selection 00:1.5 V 01:2.048 V 10:2.5 V 11:Reserved Note:The Trim values corresponding to each voltage are as follows at the Flash address: 1.5V calibration value storage address:0x1FFF 302C 2.048V calibration value storage address:0x1FFF 3030 2.5V calibration value storage address:0x1FFF 3034
25	VREF BUFFERE	RW	0	VerfBuffer enable This bit is set/reset by software 0:VerfBuffer disabled 1:VerfBuffer enabled
24	Reserved	-	-	Reserved
23	TSVREFE	RW	0	Temperature sensor and V_{REFINT} enable This bit is set and cleared by software to enable/disable the temperature sensor and V_{REFINT} channel. Can be enabled in ADC. 0:Temperature sensor and V_{REFINT} channel disabled 1:Temperature sensor and V_{REFINT} channel enabled
22	SWSTART	RC_W1	0	Start conversion of regular channels This bit is set by software to start conversion and cleared by hardware as soon as conversion starts. 0:Reset state 1:Start converting regular channels
21	JSWSTART	RC_W1	0	Start conversion of injected channels This bit is set by software to start conversion and cleared by hardware as soon as conversion starts. 0:Reset state 1:Starts conversion of injected channels
20	EXTTRIG	RW	0	External trigger conversion mode for regular channels This bit is set and cleared by software to enable/disable the external trigger used to start conversion of a regular channel group. 0:Conversion on external event disabled 1:Conversion on external event enabled

Puya Confidential

Bit	Name	R/W	Reset Value	Function
19:17	EXTSEL [2:0]	RW	0	External event select for regular group These bits select the external event used to trigger the start of conversion of a regular group: ADC external trigger events are as follows: 000:Timer 1 CC1 event 001:Timer 1 CC2 event 010:Timer 1 CC3 event 011:Timer 2 CC2 event 100:Timer 3 TRGO event 101:trigger timer15_TRGO event 110:EXTI11 111 SWSTART
16	Reserved	-	-	Reserved
15	JEXTTRIG	RW	0	External trigger conversion mode for injected channels 0:Conversion on external event disabled 1:Conversion on external event enabled
14:12	JEXTSEL [2:0]	RW	0	External event select for injected group ADC external trigger events are as follows: 000:Timer 1 TRGO event 001:Timer 1 CC4 event 010:Timer 2 TRGO event 011:Timer 2 CC1 event 100:Timer 3 TRGO event 101:Timer 15 TRGO event 110:EXTI15 111:JSWSTART
11	ALIGN	RW	0	Data alignment control bit This bit is set and cleared by software. 0:Right alignment 1:Left alignment
10:9	Reserved	-	-	Reserved
8	DMA	RW	0	DMA enable bit This bit is set and cleared by software. 0:DMA mode disabled 1:DMA mode enabled
7:4	Reserved	-	-	Reserved
3	RSTCAL	Res	0	Reset calibration

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is set by software and cleared by hardware. It is cleared after the calibration registersare initialized (RSTCAL is set to 1). 0:Calibration register initialized 1:Initialize calibration register Note:If RSTCAL is set when conversion is ongoing, additional cycles are required to clear the calibration registers.
2	CAL	RC_W1	0	Calibration enable This bit is set by software to start the calibration. It is reset by hardware after calibration is complete When ADON is invalid, and SWSTART, JSWSTART are invalid; The software initiates the calibration. 0:Calibration completed 1:Enable calibration
1	CONT	RW	0	Continuous conversion enable This bit is set and cleared by software. If set conversion takes place continuously till this bit is reset. 0:Single conversion mode 1:Continuous conversion mode
0	ADON	RW	0	A/D converter ON / OFF The ADC converter is enabled. When it is set to 1, the ADC converter wakes up. There is a delay t_{STAB} from the time the converter is powered on to the start of conversion. When set to 0, the ADC converter is in a power-down state. 0:Disable ADC conversion/calibration and go to power down mode. 1:Enable ADC and to start conversion. Note:If any other bit in this register apart from ADON is changed at the same time, then conversion is not triggered. This is to prevent triggering an erroneous conversion.

14.5.4. ADC sampling time register (ADC_SMPR)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

Puya Confidential

-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res				SMP23 [2:0]			SMP22 [2:0]			SMP21 [2:0]			SMP20 [2:0]		
-				RW			RW			RW			RW		

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11/0	SMPx [2:0]	RW	0	Channel x sample time selection These bits are written by software to select the sample time individually for each channel. During sample cycles channel selection bits must remain unchanged. 000:3.5 cycles 100:28.5 cycles 001:5.5 cycles 101:41.5 cycles 010:7.5 cycles 110:134.5 cycles 011:13.5 cycles 111:239.5 cycles

14.5.5. ADC sampling time register 2 (ADC_SMPR2)

Address offset:0x10

Reset value:0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res		SMP19 [2:0]			SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:1]	
-		RW			RW			RW			RW			RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMP15[0]	SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [2:0]		
RW	RW			RW			RW			RW			RW		

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29:0	SMPx [2:0]	RW	0	Channel x sample time selection These bits are written by software to select the sample time individually for each channel. During sample cycles channel selection bits must remain unchanged. 000:3.5 cycles 100:28.5 cycles 001:5.5 cycles 101:41.5 cycles 010:7.5 cycles 110:134.5 cycles 011:13.5 cycles 111:239.5 cycles

14.5.6. ADC sampling time register 3 (ADC_SMPR3)

Address offset:0x14

Puya Confidential

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res		SMP9 [2:0]			SMP8 [2:0]			SMP7 [2:0]			SMP6 [2:0]			SMP5 [2:1]	
-		RW			RW			RW			RW			RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMP5[0]	SMP4 [2:0]			SMP3 [2:0]			SMP2 [2:0]			SMP1 [2:0]			SMP0 [2:0]		
RW	RW			RW			RW			RW			RW		

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29:0	SMPx [2:0]	RW	0	Channel x sample time selection These bits are written by software to select the sample time individually for each channel. During sample cycles channel selection bits must remain unchanged. 000:3.5 cycles 100:28.5 cycles 001:5.5 cycles 101:41.5 cycles 010:7.5 cycles 110:134.5 cycles 011:13.5 cycles 111:239.5 cycles

14.5.7. ADC injected channel data offset register x (ADC_JOFRx) (x=1..4)

Address offset:0x18-0X24

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res				JOFFSETx [11:0]											
-				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11/0	JOFFSETx [11:0]	RW	0	Data offset for injected channel x These bits define the numerical values used to subtract from the original converted data when the conversion is injected into the channel. The conversion result can be read from in the ADC_JDRx registers.

14.5.8. ADC watchdog high threshold register (ADC_HTR)

Address offset:0x28

Reset value:0x0000 0FFF

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				HTR [11:0]											
-				RW											

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11/0	HTR [11:0]	RW	0xFFF	Analog watchdog high threshold These bits are written by software to define the high threshold for the analog watchdog.

14.5.9. ADC watchdog low threshold register (ADC_LTR)

Address offset:0x2C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				LTR [11:0]											
-				RW											

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11/0	LTR [11:0]	RW	0x000	Analog watchdog low threshold These bits are written by software to define the low threshold for the analog watchdog.

14.5.10. ADC regular sequence register 1 (ADC_SQR1)

Address offset:0x30

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res								L [3:0]				SQ16 [4:1]			
-								RW				RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16[0]	SQ15 [4:0]					SQ14 [4:0]					SQ13 [4:0]				
RW	RW					RW					RW				

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:24	Reserved	-	-	Reserved
23:20	L [3:0]	RW	0	Regular channel sequence length These bits are written by software to define the total number of conversions in the regular channel conversion sequence. 0000:1 conversion 0001:2 conversions …… 1111:16 conversions
19:15	SQ16 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 16th in the conversion sequence.
14:10	SQ15 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 15th in the conversion sequence.
9:5	SQ14 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 14th in the conversion sequence.
4:0	SQ13 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 13th in the conversion sequence.

14.5.11. ADC regular sequence register 2 (ADC_SQR2)

Address offset:0x34

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res		SQ12 [4:0]					SQ11 [4:0]					SQ10 [4:1]			
-		RW					RW					RW			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SQ10[0]		SQ9 [4:0]				SQ8 [4:0]					SQ7 [4:0]				
RW		RW				RW					RW				

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29:25	SQ12 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 12th in the conversion sequence.

Puya Confidential

24:20	SQ11 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 11th in the conversion sequence.
19:15	SQ10 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 10th in the conversion sequence.
14:10	SQ9 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 9th in the conversion sequence.
9:5	SQ8 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 8th in the conversion sequence.
4:0	SQ7 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 7th in the conversion sequence.

14.5.12. ADC regular sequence register 3 (ADC_SQR3)

Address offset:0x38

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res		SQ6 [4:0]					SQ5 [4:0]					SQ4 [4:1]			
-		RW					RW					RW			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SQ4[0]	SQ3 [4:0]					SQ2 [4:0]					SQ1 [4:0]				
RW	RW					RW					RW				

Bit	Name	R/W	Reset Value	Function
31:30	Reserved	-	-	Reserved
29:25	SQ6 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 6th in the conversion sequence.
24:20	SQ5 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 5th in the conversion sequence.
19:15	SQ4 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 4th in the conversion sequence.
14:10	SQ3 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 3rd in the conversion sequence.

Puya Confidential

9:5	SQ2 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 2nd in the conversion sequence.
4:0	SQ1 [4:0]	RW	0	These bits are written by software with the channel number (0 to 23) assigned as the 1st in the conversion sequence.

14.5.13. ADC injected sequence register (ADC_JSQR)

Address offset:0x3C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res										JL [1:0]		JSQ4 [4:1]			
-										RW		RW			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**		**4**	**3**	**2**	**1**	**0**
JSQ4[0]	JSQ3 [4:0]					JSQ2 [4:0]						JSQ1 [4:0]				
RW	RW					RW						RW				

Bit	Name	R/W	Reset Value	Function
31:22	Reserved	-	-	Reserved
21:20	JL [1:0]	RW	0	Injected sequence length These bits are written by software to define the total number of conversions in the injected channel conversion sequence. 00:1 conversion 01:2 conversions 10:3 conversions 11:4 conversions
19:15	JSQ4 [4:0]	RW	0	fourth conversion in injected sequence These bits are written by software with the channel number (0 to 23) assigned as the fourth in the sequence to be converted. Note:Unlike a regular conversion sequence, if JL [1:0] length is less than four, the channels are converted in a sequence starting from (4-JL). Example:ADC_JSQR[21:0] = 1000011 00011 00111 00010 means that a scan conversion will convert the following channel sequence:7, 3, 3. (not 2, 7, 3)
14:10	JSQ3 [4:0]	RW	0	third conversion in injected sequence
9:5	JSQ2 [4:0]	RW	0	second conversion in injected sequence
4:0	JSQ1 [4:0]	RW	0	first conversion in injected sequence

Puya Confidential

14.5.14. ADC injected data register x (ADC_JDRx) (x= 1..4)

Address offset:0x40-4C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA [15:0]															
R															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	JDATA [15:0]	R	0	Injected data The software can read the values of these bits. These bits are read only. They contain the conversion result from injected channel x. The data is left or right-aligned.

14.5.15. ADC regular data register (ADC_DR)

Address offset:0x50

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA [15:0]															
R															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	DATA [15:0]	R	0	Regular data The software can read the values of these bits. These bits are read only. They contain the conversion result from regular channel x. The data is left or right aligned

14.5.16. ADC calibration configuration and status register (ADC_CCSR)

Address offset:0x54

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALON.	CAPSUC	OFFSUC	Res												
R	RC_W1	RC_W1	-												

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALSET	CALBYP	CALSMP [1:0]		CALSEL	Res										
RC_W1	RC_W1	RW		RW	-										

Bit	Name	R/W	Reset Value	Function
31	CALON	R	0	Calibration flag, indicating that ADC calibration is in progress. 1:ADC calibration in progress 0:ADC calibration has ended or ADC calibration has not started
30	CAPSUC	RC_W1	0	Capacitance calibration status bit. Indicates whether the ADC capacitance calibration is successful. Hardware set 1; The software writes 1 and sets 0; CALON = 0, CALSEL = 0, CALSUC = 1:Invalid status CALON = 0, CALSEL = 0, CALSUC = 0:CAPs calibration is not performed CALON = 0, CALSEL = 1, CALSUC = 1:ADC CAPs calibration successful CALON = 0, CALSEL = 1, CALSUC = 0:ADC CAPs calibration failed
29	OFFSUC	RC_W1	0	Offset calibration status bit. Indicates whether the ADC offset calibration was successful. Hardware set 1; The software writes 1 and sets 0; CALON = 0, CALSEL = 0, OFFSUC = 0:ADC OFFSET calibration failed CALON = 0, CALSEL = 0, OFFSUC = 1:ADC OFFSET calibration successful CALON = 0, CALSEL = 1, OFFSUC = 1:ADC OFFSET calibration successful CALON = 0, CALSEL = 1, OFFSUC = 0:ADC OFFSET calibration failed
28:16	Reserved	-	-	Reserved
15	CALSET	RC_W1	0	Calibration factor selection. When CAL is 0, the software writes 1 to 1. When CAL is valid or injection/regular channel SWSTART, JWSTART is valid, the hardware sets 0. 1:Set CAL_CXIN data as final calibration data 0:Close the path from CAL_CXIN to CAL_CXOUT, and select the result generated internally by the calibration circuit.
14	CALBYP	RC_W1	0	Calibration factor bypass. When CAL is 0, the software writes 1 to 1. When CAL is valid or injection/regular channel SWSTART, JWSTART is valid, the hardware sets 0.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:The calibration result is the reset value 0:The calibration result is a self-calibration result or a calibration factor input value
13:12	CALSMP [1:0]	RW	0	Calibration sample time selection Configure the number of clock cycles in the calibration sampling phase based on the following information: 00:1 ADC clock cycle 01:2 ADC clock cycles 10:4 ADC clock cycles 11:8 ADC clock cycles The longer the SMP is configured during calibration, the more accurate the calibration result is, but this configuration will bring the problem of extended calibration period
11	CALSEL	RW	0	Calibration content selection bit, used to select the content to be calibrated 1:Calibration OFFSET and Linearity 0:Calibration OFFSET only
10:0	Reserved	-	-	Reserved

Puya Confidential

15. Liquid crystal display (LCD) controller

15.1. Introduction

The LCD controller is a digital controller/driver for monochrome passive liquid crystal displays (LCD), with up to 8 common terminals (COM) and 40 segment terminals (SEG) to drive 160 (4*40) or 288 (8*32) LCD pixels. The exact number of terminals depends on the device pins described in the data manual. The LCD is composed of several segments (pixels or symbols), which can all be lit or off. Each segment contains a layer of liquid crystal molecules aligned between the two electrodes. When a voltage above the threshold voltage is applied to the liquid crystal, the corresponding segment is visible. The segment voltage must be AC to avoid electrophoretic effects in the liquid crystal (which will affect the display effect). After that, waveforms must be generated across the section to avoid DC.

Glossary

- Liquid crystal display (LCD):A passive display panel with terminals leading directly to the section.
- Common (COM):A common electrical connection terminal connected to multiple sections.
- BIAS:The level used when driving the LCD, defined as 1/ (the voltage level of driving the LCD display-1).
- Segment (SEG):The smallest visual unit (the smallest constituent element, line or point on an LCD display).
- DUTY cycle (DUTY):1/The number of common terminals on the LCD display.
- Frame:A period of the waveform of the written segment.
- Frame rate:frames per second, i.e., the number of times the LCD segment is excited per second.

15.2. LCD main features

- Highly flexible frame rate control
- Supports static,1/4, 1/6, and 1/8 duty cycles.
- Supports 1/2, 1/3 bias voltage.
- Up to 16 registers LCD data RAM
- By software configuration of LCD contrast
- 2 kinds of waveform generation
  - Internal resistance divider, external resistance divider
  - By way of internal resistance of the software configuration partial pressure power consumption, so as to match the capacitance charge needed for the LCD panel
- Support low power consumption mode:LCD controller can display in Run, Sleep and Stop modes.
- Configurable frame interrupts.
- Support LCD flashing function and configuration of multiple flicker frequency configuration
- Unused LCD segments and public pin can be configured to digital or analog functions

Puya Confidential

15.3. LCD block diagram

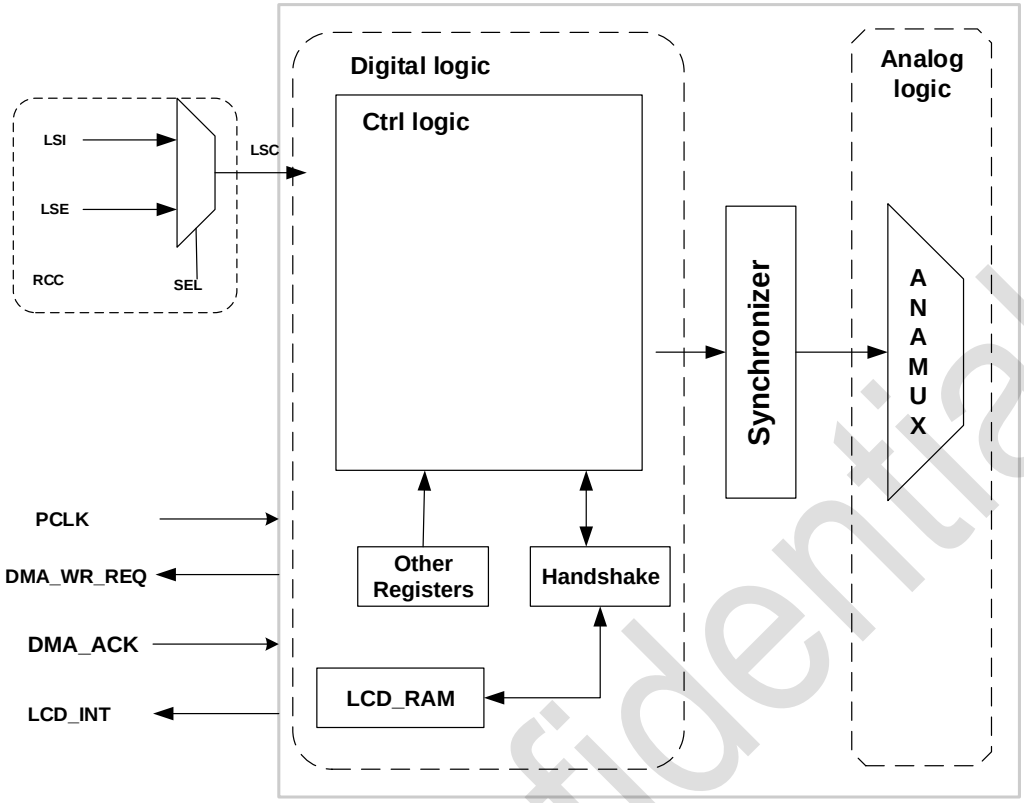


Figure 15-1 LCD block diagram

15.4. LCD clock

The LCD clock source can be selected as LSI or LSE, and the input clock can be determined by LSCOSEL and LSCOEN of the RCC_BDCR register.

15.5. LCD driving waveform

The LCD supports 5 Duty cycles driving waveforms:static,1/4, 1/6 and 1/8, which are set by LCD_CR0. Duty.

The LCD supports two BIAS driving waveforms:1/2 and 1/3, which are set by LCD_CR0. BIAS.

The suggested combination is shown in the following table:

Table 15-1 LCD supported drive waveforms

-	1/4 DUTY	1/6 DUTY	1/8 DUTY
1/2 BIAS	Not recommended	Not recommended	Not recommended
1/3 BIAS	✓	✓	✓

Puya Confidential

15.5.1. Static drive waveform

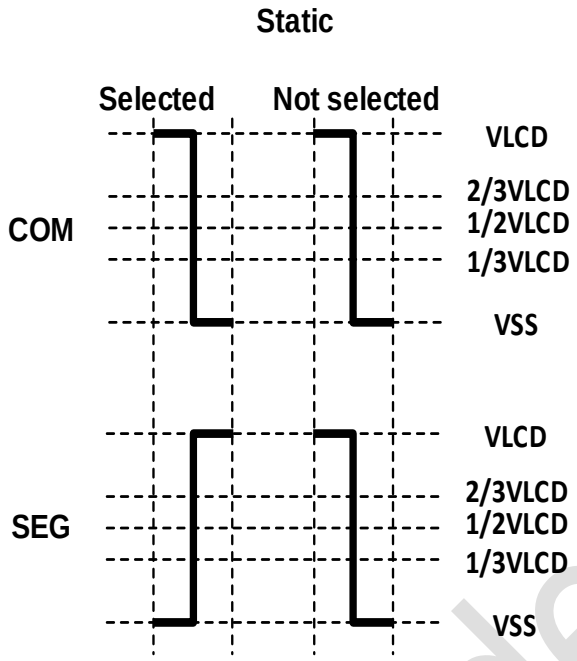


Figure 15-2 Static drive waveform

15.5.2. 1/8 DUTY 1/3 BIAS driving waveform

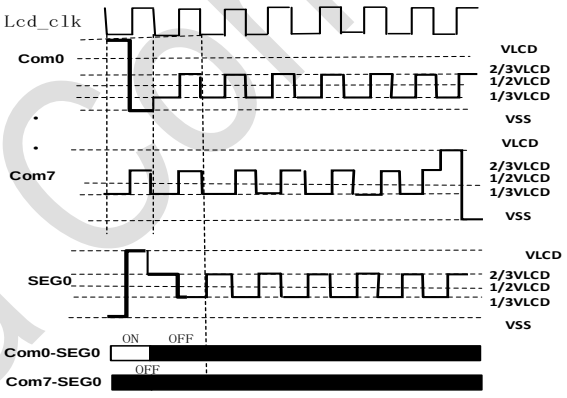


Figure 15-3 1/8 DUTY 1/3 BIAS drive waveform

15.6. LCD BIAS generating circuit

The BIAS voltage of the LCD comes from two sources:internal resistor voltage division and external resistor voltage division. When the internal resistor voltage divider is selected, the device automatically switches the internal circuit to generate a voltage that complies with BIAS and DUTY. When selecting an external resistor to divide voltage, users need to build related circuits on the peripheral pins of the device.

15.6.1. Internal resistance mode

The internal resistance modes VLCDH, VLCD1 to VLCD3 can be used as an LCD SEG output or an IO port.

Puya Confidential

In internal resistance mode, the driving voltage of the LCD is controlled by CR0. CONTRAST, as shown in the following table:

Table 15-2 Internal resistance mode

CR0.CONTRAST	VLCD (1/3 BIAS)	VLCD (1/2 BIAS)
0	1.00 * V_{CCA}	1.00 * V_{CCA}
1	0.95* V_{CCA}	0.92* V_{CCA}
2	0.9 * V_{CCA}	0.86 * V_{CCA}
3	0.86* V_{CCA}	0.8* V_{CCA}
4	0.82 * V_{CCA}	0.75 * V_{CCA}
5	0.78 * V_{CCA}	0.71 * V_{CCA}
6	0.75 * V_{CCA}	0.67 * V_{CCA}
7	0.72 * V_{CCA}	0.63 * V_{CCA}
8	0.69 * V_{CCA}	0.60 * V_{CCA}
9	0.67 * V_{CCA}	0.57 * V_{CCA}
10	0.64 * V_{CCA}	0.55 * V_{CCA}
11	0.62 * V_{CCA}	0.52 * V_{CCA}
12	0.60 * V_{CCA}	0.50 * V_{CCA}
13	0.58* V_{CCA}	0.48 * V_{CCA}
14	0.56 * V_{CCA}	0.46* V_{CCA}
15	0.55 * V_{CCA}	0.44 * V_{CCA}

Puya Confidential

15.6.2. External resistance mode

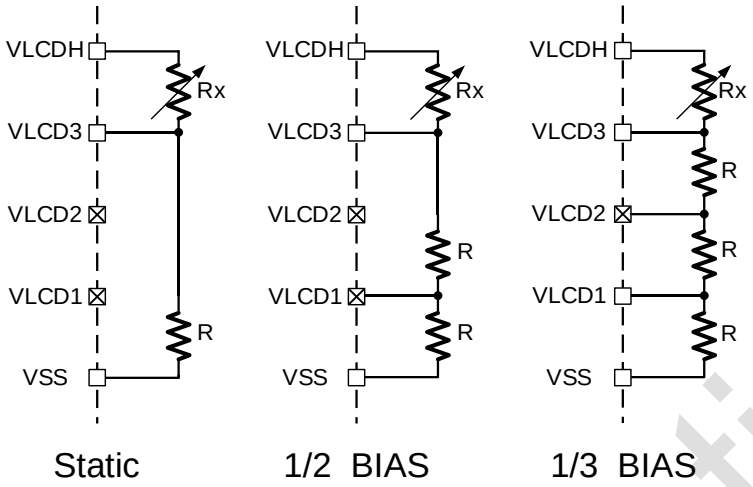


Figure 15-4 External resistance mode

Note:

1. Rx is an adjustable resistor used to adjust the LCD display contrast.
2. Select the appropriate resistor R according to the use of the LCD screen.
3. Unused VLCD2/VLCD1 can be configured for SEG or IO use as required.

15.7. DMA

LCD supports software and hardware to trigger DMA data transmission, which can automatically move the content to be displayed from the stored content to the LCD display RAM. The hardware trigger uses a frame interrupt signal.

DMA data transfer configuration process:

1. Enable DMA
2. Select LCD DMA
3. Set transmission type, transmission length, and transmission method
4. Set the source start address and the destination start address
5. Set the increment method of source address and destination address
6. Enable DMA interrupts as required
7. Enable LCD DMA Trigger

15.8. Interrupts and events

When the LCD setting is active, the LCD interrupt may be configured to generate a frame number interrupt.

15.9. LCD display mode

The LCD supports two display modes. One with COM as the display unit, and all COM segments of the same SEG are in the same byte (mode 0). The other is that different SEGs of the same COM are in the same byte (mode 1).

Choosing the appropriate display mode according to the LCD panel can simplify the program operation.

Puya Confidential

15.9.1. LCD display mode 1 (MODE = 1)

■ 1/8 DUTY

	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
COM4																																	LCDRAM4
COM5																																	LCDRAM5
COM6																																	LCDRAM6
COM7																																	LCDRAM7

	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32	
COM0									LCDRAM8
COM1									LCDRAM9
COM2									LCDRAMA
COM3									LCDRAMB
COM4									LCDRAMC
COM5									LCDRAMD
COM6									LCDRAME
COM7									LCDRAMF

Figure 15-5 1/8 DUTY LCD display mode 1

■ 1/6 DUTY

	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
COM4																																	LCDRAM4
COM5																																	LCDRAM5
COM6																																	LCDRAM6
COM7																																	LCDRAM7

	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32	
COM0									LCDRAM8
COM1									LCDRAM9
COM2									LCDRAMA
COM3									LCDRAMB
COM4									LCDRAMC
COM5									LCDRAMD
COM6									LCDRAME
COM7									LCDRAMF

Figure 15-6 1/6 DUTY LCD display mode 1

■ 1/4 DUTY

	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
COM4																																	LCDRAM4
COM5																																	LCDRAM5
COM6																																	LCDRAM6
COM7																																	LCDRAM7

	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32	
COM0									LCDRAM8
COM1									LCDRAM9
COM2									LCDRAMA
COM3									LCDRAMB
COM4									LCDRAMC
COM5									LCDRAMD
COM6									LCDRAME
COM7									LCDRAMF

Figure 15-7 1/4 DUTY LCD display mode 1

Puya Confidential

15.9.2. LCD Display MODE 0 (MODE = 0)

■ 1/8 DUTY

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	
SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	LCDRAM0
SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	LCDRAM1
SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	LCDRAM2
SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	LCDRAM3
SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	LCDRAM4
SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	LCDRAM5
SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	LCDRAM6
SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	LCDRAM7
																								SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	LCDRAM8
																								SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	LCDRAM9
																								SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	LCDRAMA
																								SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	LCDRAMB
																								SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	LCDRAMC
																								SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	LCDRAMD
																								SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	LCDRAME
																								SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	LCDRAMF

Figure 15-8 1/8 DUTY LCD display mode 0

■ 1/6 DUTY

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	
SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	LCDRAM0
SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	LCDRAM1
SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	LCDRAM2
SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	LCDRAM3
SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	LCDRAM4
SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	LCDRAM5
SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	LCDRAM6
SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	LCDRAM7
																								SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	LCDRAM8
																								SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	LCDRAM9
																								SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	LCDRAMA
																								SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	LCDRAMB
																								SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	LCDRAMC
																								SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	LCDRAMD
																								SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	LCDRAME
																								SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	LCDRAMF

Figure 15-9 1/6 DUTY LCD display mode 0

■ 1/4 DUTY

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	
SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	LCDRAM0
SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	LCDRAM1
SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	LCDRAM2
SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	LCDRAM3
SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	LCDRAM4
SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	LCDRAM5
SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	LCDRAM6
SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	LCDRAM7
																								SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	LCDRAM8
																								SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	LCDRAM9
																								SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	LCDRAMA
																								SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	LCDRAMB
																								SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36	LCDRAMC
																								SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	LCDRAMD
																								SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	LCDRAME
																								SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	LCDRAMF

Figure 15-10 1/4 DUTY LCD display mode 0

Puya Confidential

15.10. LCD registers

15.10.1. LCD configuration register 0 (LCD_CR0)

Address offset:0x00

Reset value:0x0000 00C2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONTRAST [3:0]				BSEL [2:0]			DUTY [2:0]			BIAS	Res	Res	LCDCLK [1:0]		EN
RW				RW			RW			RW	-	-	RW		RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:12	CONTRAST [3:0]	RW	0	LCD contrast adjustment Note:Only valid if internal resistor divider is selected for the BIAS voltage source. The larger the value, the smaller the amplitude of the LCD waveform. At 0x0, the LCD waveform has the largest amplitude and the largest contrast ratio; …… At 0xF, the LCD waveform has the smallest amplitude and the smallest contrast ratio;
11:9	BSEL [2:0]	RW	0	BIAS voltage source selection 111:Reserved 110:internal resistor voltage division, large power consumption mode 101:Reserved 100:internal resistor voltage division, low power consumption mode 011:Reserved 010:Internal resistor voltage division, medium power consumption mode 001:Reserved 000:External resistance mode, requires external circuit cooperation Note:Only external resistor mode can be used at 1/6 or 1/8 duty (SEG 32 to SEG35 disabled).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
8:6	DUTY [2:0]	RW	3'b011	LCD DUTY configuration 000:Static 001:Reserved 010:Reserved 011:1/4 DUTY 100:Reserved 101:1/6 DUTY 110 Reserved 111 1/8 DUTY
5	BIAS	RW	0	0:1/3 bias (initial value) 1:1/2 bias
4:3	Reserved	-	-	Reserved
2:1	LCDCLK [1:0]	RW	2'b01	LCD scanning frequency selection 00:64 Hz 01:128 Hz 10:256 Hz 11:512 Hz Note:LCD frame frequency = LCD scanning frequency × DUTY
0	EN	RW	0	LCD enable control 1:Enabled 0:Disabled

15.10.2. LCD configuration register 1 (LCD_CR1)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	INTF	DMAEN	IE	MODE	Res	BLINKEN	BLINKCNT [5:0]					
-	-	-	-	R	RW	RW	RW	-	RW	RW					

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11	INTF	R	0	LCD interrupt flag 1:Interrupt 0:No interrupt
10	DMAEN	RW	0	DMA hardware trigger enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:LCD interrupt trigger DMA enabled 0:LCD interrupt trigger DMA disabled
9	IE	RW	0	the interrupt enable 1:Enabled 0:Disabled
8	MODE	RW	0	LCD RAM display mode deflection 0:Mode 0 1:Mode 1
7	Reserved	-	-	Reserved
6	BLINKEN	RW	0	LCD splash screen configuration 1:Enabled 0:Disabled
5:0	BLINKCNT [5:0]	RW	0	Splash screen frequency and LCD interruption interval setting Note:LCD blinking frequency is = LCD frame frequency/(BLINKCNT+1) LCD interrupt interval = (BLINKCNT+1) * (1/LCD frame frequency)

15.10.3. LCD interrupt clear register (LCD_INTCLR)

Address offset:0x08

Reset value:0x0000 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	INTF_CLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-					R1W0	-									

Bit	Name	R/W	Reset Value	Function
31:11	Reserved	-	-	Reserved
10	INTF_CLR	RC_W0	1	Interrupt flag clear, write 0 clear, write 1 invalid
9:0	Reserved	-	-	Reserved

15.10.4. LCD output configuration register (LCD_POEN0)

Address offset:0x0C

Reset value:0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	Sx	RW	0xFFFFFFF	SEGx output control bit 0:SEG output enabled 1:SEG output is off, other functions can be used, such as IO, analog input and output

15.10.5. LCD output configuration register 1 (LCD_POEN1)

Address offset:0x10

Reset value:0x1FFF

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res			MUX	C3	C2	C1	C0	S36_C7	S37_C6	S38_C5	S39_C4	S35	S34	S33	S32
-			RW												

Bit	Name	R/W	Reset Value	Function
31:13	Reserved	-	-	Reserved
12	MUX	RW	1	SEG32 to SEG35 port function selection, see the following table for details
11:8	Cx	RW	0xF	COMx output control bits (COM0 to COM3) 0:COM output enabled 1:COM output disabled, other functions can be used, such as IO, analog input and output
7:4	SxCy [3:0]	RW	0xF	SEGx/COMy output control bit 0:SEG/COM output enabled 1:SEG/COM output is turned off, other functions can be used, such as IO, analog input and output SEG COM pin function selection is determined by CR0. DUTY
3:0	S[3:0]	RW	0xF	SEGx output control bit 0:SEG output enabled 1:SEG output is off, other functions can be used, such as IO, analog input and output

Puya Confidential

Table 15-3 COM/SEG multiplexing selection configuration

VLCDxSEGxPAD			How to configure registers				
			MUX	S < 35:32 >	BSEL		
VLCDXSEGXPAD select GPIO, LCD disable (LCD_ON = 0)		VLCDHSEG35=IO	1	1111	X	X	X
		VLCDHSEG34=IO					
		VLCDHSEG33=IO					
		VLCDHSEG32=IO					
When VLCDXSEGXPAD selects IO and LCD enable (LCD_ON = 1), only the internal resistor operation mode can be selected	Low resistance (high current) mode	VLCDHSEG35=IO	1	1111	1	1	0
		VLCDHSEG34=IO					
		VLCDHSEG33=IO					
		VLCDHSEG32=IO					
	Medium resistance (medium current) mode	VLCDHSEG35=IO	1	1111	0	1	0
		VLCDHSEG34=IO					
		VLCDHSEG33=IO					
		VLCDHSEG32=IO					
	Large resistance (low current) mode	VLCDHSEG35=IO	1	1111	1	0	0
		VLCDHSEG34=IO					
		VLCDHSEG33=IO					
		VLCDHSEG32=IO					
Select Internal Resistor Operating Mode	Low resistance (high current) mode	VLCDHSEG35=SEG35/IO	0	XXXX	1	1	0
		VLCDHSEG34=SEG34/IO					
		VLCDHSEG33=SEG33/IO					
		VLCDHSEG32=SEG32/IO					
	Medium resistance (medium current) mode	VLCDHSEG35=SEG35/IO	0	XXXX	0	1	0
		VLCDHSEG34=SEG34/IO					
		VLCDHSEG33=SEG33/IO					
		VLCDHSEG32=SEG32/IO					
	Large resistance (low current) mode	VLCDHSEG35=SEG35/IO	0	XXXX	1	0	0
		VLCDHSEG34=SEG34/IO					
		VLCDHSEG33=SEG33/IO					
		VLCDHSEG32=SEG32/IO					
Select the external resistor working mode		VLCDHSEG35=VOUT	0	1111	0	0	0
		VLCDHSEG34=VLCD3					
		VLCDHSEG33=VLCD2					
		VLCDHSEG32=VLCD1					

15.10.6. LCD_RAM 0 to 7

Address offset:0x14 to 0x30

Reset value:0x00000000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	Dx	RW	0	LCD dot output, displaying reference LCD display mode 0:The corresponding SEG COM intersection is not lit 1:Corresponding SEG COM cross-lit

Note:When configuring LCD_RAMx, ensure that it is completed within 2 LCD clocks (LSI or LSE). And the interval between the two writes satisfies 2 LCD clocks.

15.10.7. LCD_RAM 8 to F

Address offset:0x34 to 0x50

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7:0	Dx	RW	0	LCD dot output, displaying reference LCD display mode 0:The corresponding SEG COM intersection point is not lit; 1:Corresponding SEG COM cross-lit;

Note:When configuring LCD_RAMx, ensure that it is completed within 2 LCD clocks (LSI or LSE). And the interval between the two writes satisfies 2 LCD clocks.

Puya Confidential

16. Comparator (COMP)

16.1. Introduction

The device integrates two general-purpose comparators (COMP). The COMP1/2 module can be used as a separate module or in combination with timer.

The comparator may be used as follows:

- Triggered by analog signal to wake-up function from low-power mode
- Analog signal conditioning
- Cycle by cycle current control loop when comparators are connected with PWM output from timer.

16.2. COMP main features

- Each comparator has configurable positive or negative input for flexible voltage selection:
  - Multiple I/O pins
  - Output of the temperature sensor
  - Internal reference voltages V_{REFBUF}, V_{CCA}, V_{REF1P2} 64 fractional values provided by voltage division (1/64, 2/64 … 64/64)(That's V_{REFCMP})
  - V_{REFINT}
  - OPA output as INP input
- Configurable hysteresis function
- Programmable speed and consumption
- The output can be triggered by a connection to the I/O or Timer input
  - OCREF_CLR event (cycle-by-cycle current control)
  - Break events for fast PWM shutdowns
- COMP1 and COMP2 may be combined into a window comparator
- Each COMP has interrupt generation capability and is used to wake up the device from low power mode (Sleep/Stop) (via EXTI)

Puya Confidential

16.3. COMP functional description

16.3.1. COMP block diagram

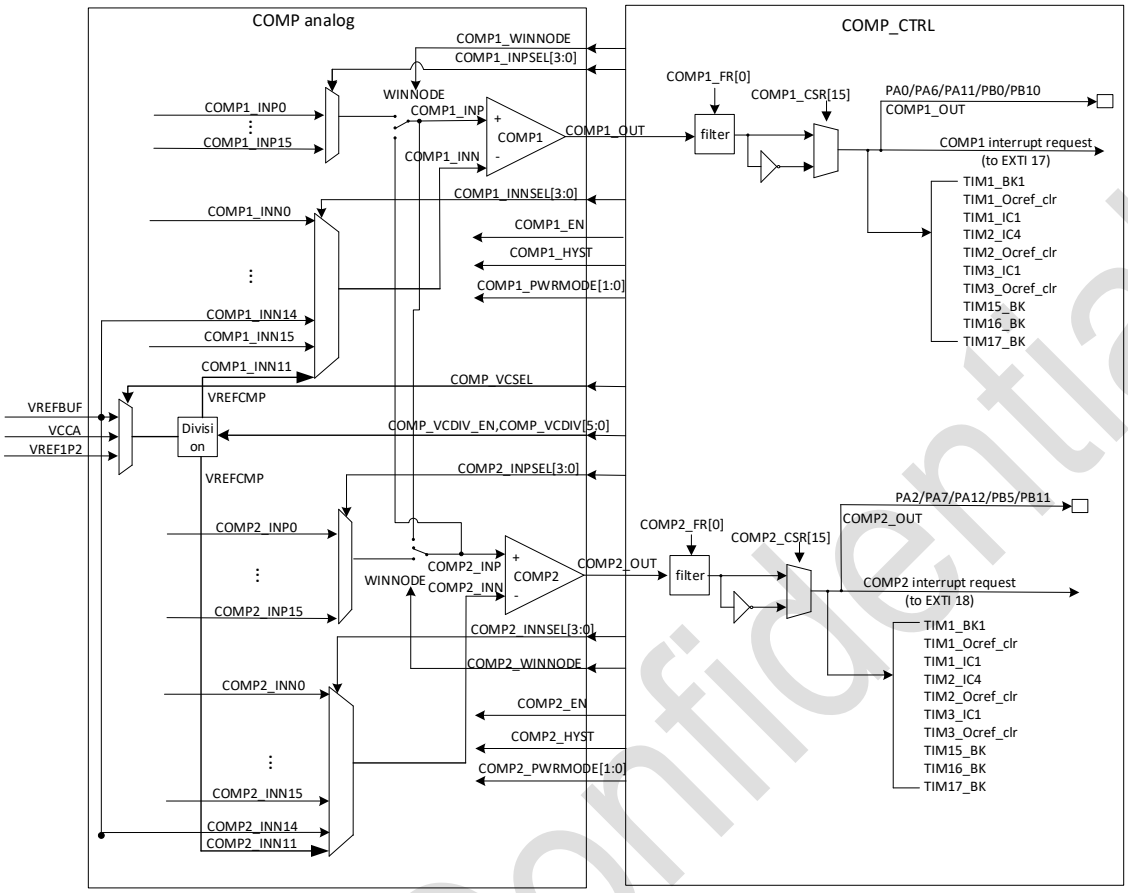


Figure 16-1 Comparator block diagram

Note:V_{REF1P2} is the reference voltage generated by the on-chip bandgap reference circuit, which is always enabled after the chip is powered on.

Puya Confidential

16.3.2. COMP pins and internal signals

The I/Os used as comparators inputs must be configured in analog mode in the GPIOs registers.

The comparator output can be connected to the I/Os using the alternate function channel.

The output can also be internally redirected to a variety of timer input for the following purposes:

- When connecting the brake input, the PWM signal can be turned off in an emergency
- Cycle-by-cycle current control, using OCREF_CLR inputs
- Input capture for timing measures

16.3.3. COMP reset and clocks

The COMP module has two clock sources:

- PCLK (APB clock), used to provide clock to the configuration register
- COMP clock is used to simulate the clock of the circuit after the output of the comparator (analog output latch circuit, glitch filter circuit, etc.). It can be selected as PCLK, LSI or LSE. When you need to work in Stop mode, select LSI or LSE.

Note:

1. PCLK and COMP CLK are simultaneously enabled by the RCC_APBENR2 control bit. If enabled, PCLK and COMP CLK are turned off, and if enabled, PCLK and COMP CLK are turned on simultaneously.
2. Before entering Stop mode, it is recommended to configure COMP CLK to LSI or LSE, and then enter Stop mode.
3. The reset signal of the COMP module includes APB reset source and COMP module software reset source:
    1) PCLK clock domain reset for COMP register reset
    2) COMP CLK clock domain reset, used for resetting circuits after analog comparator output (analog output latch circuit, glitch filter circuit, etc.).

16.3.4. Window comparator

The purpose of window comparator is to monitor the analog voltage if it is within specified voltage range defined by lower and upper threshold.

Two embedded comparators can be utilized to create window comparator. The monitored analog voltage is connected to the non-inverting (plus) inputs of comparators connected together and the upper and lower threshold voltages are connected to the inverting (minus) inputs of the comparators.

Two non-inverting inputs can be connected internally together by enabling WINMODE bit to save one IO for other purposes.

Note:The WINMODE mode of both COMPs cannot be enabled simultaneously.

Puya Confidential

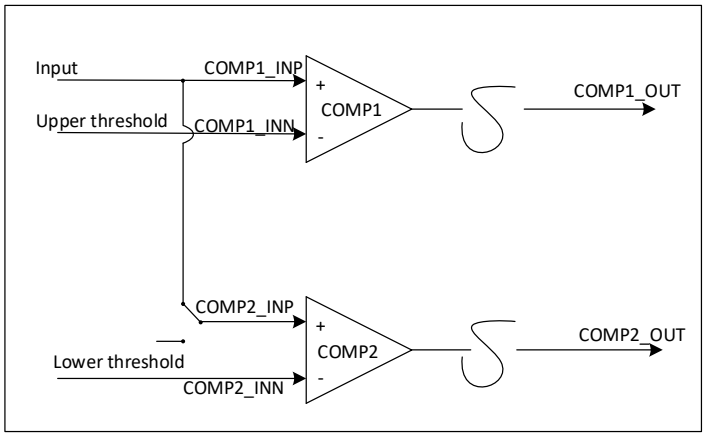


Figure 16-2 Window comparator

16.3.5. Hysteresis

To avoid false output transitions in the case of noisy signals, the comparator can enable functions with hysteresis (COMP1,and COMP2 have independent hysteresis function enable signals COMP1_HYST and COMP2_HYST).

16.3.6. Power mode

The power consumption and propagation delay of the comparator can be selected in different modes by the PWRMODE [1:0] bit of the COMPx_CSR register to achieve the most suitable trade-off for a given application. The optional modes include high-speed mode and medium-speed mode. Relatively speaking, the high-speed mode consumes more power and has smaller propagation delay. Note that before entering Stop, if you select the PWR_CR2 register LPR = 1 (i.e. select to power with Low power regulator), you need to first set COMP in medium speed mode (PWRMODE = 01).

16.3.7. Comparator filtering

The output filtering function of COMP and the corresponding filter width can be enabled by setting the COMP_FR register. Note that this setting should be completed before COMP_EN is enabled.

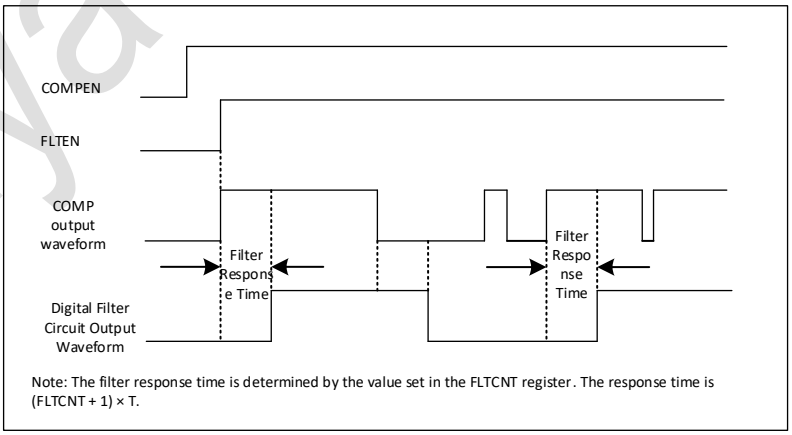


Figure 16-3 Comparator filtering

Puya Confidential

16.3.8. COMP interrupts

The comparator outputs are internally connected to the extended interrupts and events controller. Each comparator has its own EXTI line (17, 18 and 20) and can generate either interrupts or events. The same mechanism is used to exit from low-power modes.

16.4. COMP registers

16.4.1. Comparator 1 control and status register (COMP1_CSR)

Address offset:0x0200_0000

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	COMP_OUT	Res		COMP_VCSEL [1:0]		COMP_VCDIV [5:0]						PWRMODE [1:0]		Res	COMP1_HYST
-	R	-		RW		RW						RW		-	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PO-LARITY	Res			WINMODE	Res	INPSEL [3:0]				INNSEL [3:0]				Res	COMP1_EN
RW	-			RW	-	RW				RW				-	RW

Bit	Name	R/W	Reset Value	Function
31	Reserved	-	-	Reserved
30	COMP_OUT	R	0	COMP1 output This read-only flag reflects the level of the comparator 1 output before the polarity selector.
29:28	Reserved	-	-	Reserved
27:26	COMP_VCSEL	RW	0	V_{REFCMP} reference voltage source selection. 00:Voltage division is not enabled 01:V_{CCA} 10:V_{REFBUF} 11:V_{REF1P2}
25:20	COMP_VCDIV [5:0]	RW	100000	V_{REFCMP} partial voltage selection 00_0000:1/64 V_{REFBUF} or V_{CCA} or V_{REF1P2} 00_0001:2/64 V_{REFBUF} or V_{CCA} or V_{REF1P2} 00_0010:3/64 V_{REFBUF} or V_{CCA} or V_{REF1P2} 11_1110:63/64 V_{REFBUF} or V_{CCA} or V_{REF1P2} 11_1111:V_{REFBUF} or V_{CCA} or V_{REF1P2}
19:18	PWRMODE [1:0]	RW	0	Comparator 1 power mode selector It selects the power consumption and as a consequence the speed of the comparator 1 00:High speed mode

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				01:Medium speed mode 10:Reserved 11:Reserved
17	Reserved	-	-	Reserved
16	COMP1_HYST	RW	0	Comparator 1 hysteresis selector 0:No hysteresis 1:Hysteresis voltage about 20 mV
15	POLARITY	RW	0	Comparator 1 polarity selector These bits can be read and written by software. 0:Non-inverted 1:Inverted
14:12	Reserved	-	-	Reserved
11	WINMODE	RW	0	Comparator 1 non-inverting input selector for window mode These bits can be read and written by software. 0:Signal selected by INPSEL [3:0] 1:COMP2_INP signal of COMP2 Note:The WINMODE mode of both COMPs cannot be enabled simultaneously.
10	Reserved	-	-	Reserved
9:6	INPSEL [3:0]	RW	0	Comparator 1 signal selector for non-inverting input 0000:COMP1_INP0 from OPA1 output 0001:COMP1_INP1 from PC1 0010:COMP1_INP2 from PC2 0011:COMP1_INP3 from PC3 0100:COMP1_INP4 from PA0 0101 COMP1_INP5 from PA1 0110 COMP1_INP6 from PA2 0111 COMP1_INP7 from PA3 1000 COMP1_INP8 from PA4 1001:COMP1_INP9 from PA5 1010:COMP1_INP10 from PA6 1011:COMP1_INP11 from PA7 1100:COMP1_INP12 from PB4 1101 COMP1_INP13 from PB5 1110:COMP1_INP14 from PB6 1111:Reserved
5:2	INNSEL [3:0]	RW	0	Comparator 1 signal selector for inverting input 0000:COMP1_INN0 from PA0

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0001:COMP1_INN1 from PA1 0010:COMP1_INN2 from PA2 0011:COMP1_INN3 from PA3 0100:COMP1_INN4 from PA4 0101:COMP1_INN5 from PA5 0110:COMP1_INN6 from PA6 0111:COMP1_INN7 from PA7 1000:COMP1_INN8 from PC4 1001:COMP1_INN9 from PC5 1010:Reserved 1011:COMP1_INN11 from V_{REFCMP} 1100:COMP1_INN12 from TS_VIN temperature sensor voltage) 1101:COMP1_INN13 from V_{REFINT} (ADC_CR2. TSVREFE register required to enable the ADC module) 1110:COMP1_INN14 from V_{REFBUF} 1111:COMP1_INN15 from PC0
1	Reserved	-	-	Reserved
0	COMP1_EN	RW	0	Comparator 1 enable Software can be read and written (if not locked) 0:Disabled 1:Enabled

16.4.2. Comparator 1 filtering register (COMP1_FR)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT1 [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FLTEN1
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW

Bit	Name	R/W	Reset Value	Function
31:16	FLTCNT1 [15:0]	RW	0	Comparator 1 sampling filter counter The sampling clock is APB or LSI or LSE. The filter count value is configurable. When the number of samples reaches the filter count value, the results are output consistently.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Sample count period = FLTCNT [15:0]
15:1	Reserved	-	-	Reserved
0	FLTEN1	RW	0	Comparator 1 digital filter function configuration 0:Digital filtering function disabled 1:Digital filtering function enabled Note:This bit must be set when COMP1_EN is 0

16.4.3. Comparator 2 control and status register (COMP2_CSR)

Address offset:0x10

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	COMP_OUT	Res										PWRMODE [1:0]		Res	Res
-	R	-										RW		-	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY	Res	Res		WINMODE	Res	INPSEL [3:0]				INNSEL [3:0]				COMP2_HYST	COMP2_EN
RW	-	-		RW	-	RW									

Bit	Name	R/W	Reset Value	Function
31	Reserved	-	-	Reserved
30	COMP_OUT	R	0	COMP2 output This read-only flag reflects the level of the comparator 2 output before the polarity selector.
29:20	Reserved	-	-	Reserved
19:18	PWRMODE [1:0]	RW	0	Comparator 2 power mode selector It selects the power consumption and as a consequence the speed of the comparator 2. 00:High speed mode 01:Medium speed mode 10:Reserved 11:Reserved
17:16	Reserved	-	-	Reserved
15	POLARITY	RW	0	Comparator 2 polarity selector These bits can be read and written by software. 0:Non-inverted 1:Inverted
14:12	Reserved	-	-	Reserved

Puya Confidential

Bit	Name	R/W	Reset Value	Function
11	WINMODE	RW	0	COMP2 non-inverted output selector These bits can be read and written by software. 0:Signal selected by INPSEL [1:0] 1:COMP1_INP signal of COMP1 Note:The WINMODE mode of both COMPs cannot be enabled simultaneously.
10	Reserved	-	-	Reserved
9:6	INPSEL [3:0]	RW	0	COMP2 signal selection without inverting input, software readable and writeable 0000:COMP2_INP0 from PA0 0001:COMP2_INP1 from PA1 0010:COMP2_INP2 from PA2 0011:COMP2_INP3 from PA3 0100:COMP2_INP4 from PA4 0101 COMP2_INP5 from PA5 0110 COMP2_INP6 from PB1 0111:COMP2_INP7 from PB2 1000:COMP2_INP8 from PB10 1001:COMP2_INP9 from OPA2 Output 1010:COMP2_INP10 from PB13 1011:COMP2_INP11 from PB14 1100:COMP2_INP12 from PB4 1101：COMP2_INP14 from PB6 1110：COMP2_INP15 from PB7 1111：Reserved
5:2	INNSEL [3:0]	RW	0	Comparator 2 signal selector for non-inverting input 0000:COMP2_INN0 from PC0 0001:COMP2_INN1 from PC1 0010:COMP2_INN2 from PC2 0011:COMP2_INN3 from PC3 0100:COMP2_INN4 from PA0 0101 COMP2_INN5 from PA1 0110 COMP1_INN6 from PB0 0111:COMP1_INN7 from PB1 1000 COMP2_INN8 from PB2 1001:COMP2_INN9 from PB3 1010:Reserved 1011:COMP2_INN11 from V_{REFCMP}

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1100:COMP2_INN12 from TS_VIN Temperature sensor voltage) 1101:COMP2_INN13 from V_{REFINT} (ADC_CR2. TSVREFE register required to enable the ADC module) 1110:COMP2_INN14 from V_{REFBUF} 1111:COMP2_INN15 from PB12
1	COMP2_HYST	RW	0	Comparator 2 hysteresis selector 0:No hysteresis 1:Hysteresis voltage about 20 mV
0	COMP2_EN	RW	0	Comparator 2 enable These bits can be read and written by software. 0:Disabled 1:Enabled

16.4.4. Comparator 2 filtering register (COMP2_FR)

Address offset:0x14

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT2 [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FLTEN2
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW

Bit	Name	R/W	Reset Value	Function
31:16	FLTCNT2 [15:0]	RW	0	Comparator 2 sampling filter counter The sampling clock is APB or LSI or LSE. The filter count value is configurable. When the number of samples reaches the filter count value, the results are output consistently. Sample count period = FLTCNT [15:0]
15:1	Reserved	-	-	Reserved
0	FLTEN2	RW	0	Comparator 2 digital filter function configuration 0:Digital filtering function disabled 1:Digital filtering function enabled Note:This bit must be set when COMP2_EN is 0

Puya Confidential

17. Operational amplifier (OPA)

17.1. Introduction

The OPA module is suitable for simple amplifiers. The two internal OPA can be cascaded using external resistors. The input range of the OPA is from 0 to V_{CCA}, and the output range is from 0.2 V to V_{CCA}-0.2 V.

17.2. OPA main features

- Three independently configured operational amps
- The input range is from 0 to V_{CCA}, and the output range is from 0.2 V to V_{CCA}-0.2 V.
- Can be configured for the following models
  - General operational amplifier mode

17.3. OPA functional description

Three OPAs can amplify small-signal analog input signals by using external components to form amplifiers, and the output is an amplified signal.

17.3.1. OPA block diagram

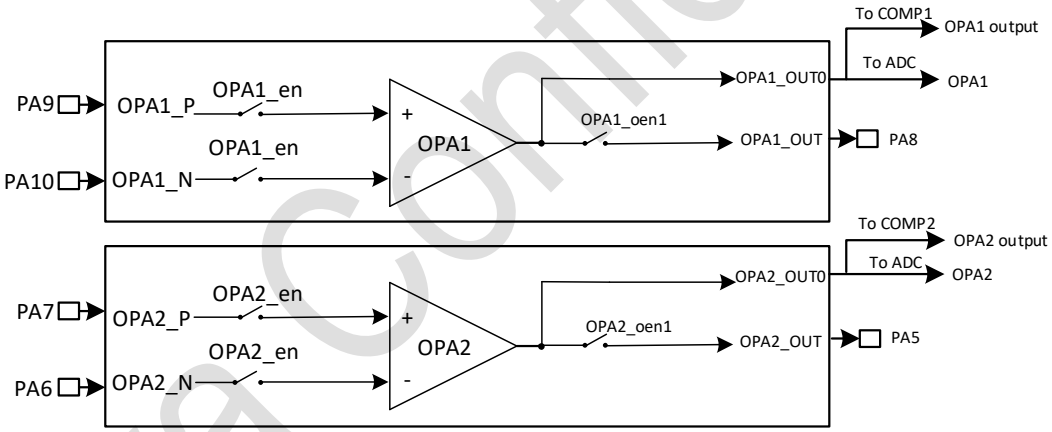


Figure 17-1 OPA block diagram

17.4. OPA registers

17.4.1. OPA output enable register (OPA_CR0)

Address offset:0x30

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res									OP2OEN1	Res				OP1OEN1	Res

Puya Confidential

-	RW	-	RW	-

Bit	Name	R/W	Reset Value	Function
31:7	Reserved	-	-	Reserved
6	OPA2OEN1	RW	0	OPA2 output 1 enable
5:2	Reserved	-	-	Reserved
1	OPA1OEN1	RW	0	OPA1 output 1 enable
0	Reserved	-	-	Reserved

17.4.2. OPA control register 1 (OPA_CR1)

Address offset:0x34

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res									EN2	EN1	Res				
-									RW	RW	-				

Bit	Name	R/W	Reset Value	Function
31:7	Reserved	-	-	Reserved
6	EN2	RW	0	OPA2 enabled
5	EN1	RW	0	OPA1 enabled
4:0	Reserved	-	-	Reserved

Puya Confidential

18. Hardware divider (DIV)

18.1. Introduction

Hardware divider is a 32-bit signed/unsigned integer hardware divider.

18.2. DIV main features

- Support 32-bit division
- The data in the register cannot be changed while the current division is not finished
- Configurable signed/unsigned division calculation
- 32-bit dividend, 32-bit divisor
- Outputs 32-bit quotient and 32-bit remainder
- Divide-by-zero warning flag bit, end-of-division flag bit
- 8 clock cycles to complete a division operation
- Write the divisor register to trigger the start of the division circuit
- After writing the divisor, when reading the quotient and remainder registers, you need to wait for the completion flag DIV_END
- When the divisor is 0, the result of quotient and remainder is 0

18.3. DIV functional description

18.3.1. DIV operation process

1. Turn on the module clock enable bit of the hardware divider in the RCC system clock controller.
2. The configuration register DIV_SIGN sets signed/unsigned division operation.
3. Configure the register DIV_DEND to set the dividend.
4. Configure the register DIV_SOR to set the divisor, and the division operation starts.
5. The operation end flag bit DIV_END in the query register DIV_STAT. DIV_END is 1 to mark the operation end. Read register DIV_QUOT to get the quotient and read register HDIV_REMD to get the remainder.
6. When the divisor is zero, the division operation ends immediately, the quotient and the remainder are set to 0 at the same time, and the divisor is zero warning flag DIV_ZERO is set.
7. Before the end of the division operation, when reading the register DIV_QUOT/DIV_REMD, the CPU will read the last calculated value

Example:Calculate an unsigned division with a dividend of 1917887483 (0x7250A3FB) and a divisor of 9597 (0x257D)

Step 1, the configuration register DIV_SIGN is 0, that is, unsigned division operation

Step 2, configure the register DIV_DEND to 0x7250A3FB, that is, set the dividend

Step 3:configure the register DIV_SOR as 0x257D, that is, set the divisor, and start the calculation

Step 4, search DIV_END bit in the DIV_STAT register, when DIV_END is set, the operation is over.

Read register DIV_QUOT to get quotient 199842 (0x30CA2)

Read register DIV_REMD to get remainder 3809 (0xEE1)

Puya Confidential

18.4. DIV registers

18.4.1. DIV dividend register (DIV_DEND)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_DEND [31:16]															
RW															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV_DEND [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:0	DIV_DEND	RW	0	Dividend

18.4.2. DIV divisor register (DIV_SOR)

Address offset:0x04

Reset value:0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_SOR [31:16]															
RW															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV_SOR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:0	DIV_SOR	RW	1	Divisor (writing this register automatically triggers the division operation)

18.4.3. DIV quotient register (DIV_QUOT)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_QUOT [31:16]															
RW															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV_QUOT [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:0	DIV_QUOT	RW	0	Store the quotient calculated by the divider

Puya Confidential

18.4.4. DIV remainder register (DIV_REMD)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_REMD [31:16]															
RW															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV_REMD [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:0	DIV_REMD	RW	0	Store the remainder calculated by the divider

18.4.5. DIV sign register (DIV_SIGN)

Address offset:0x10

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res															SIGN
-															RW

Bit	Name	R/W	Reset Value	Function
31:1	Reserved	-	-	Reserved
0	SIGN	RW	0	Sign selection 0:unsigned division operation 1:signed division operation

18.4.6. DIV status register (DIV_STAT)

Address offset:0x14

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res														DIV_ZERO	DIV_END
-														RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:2	Reserved	-	-	Reserved
1	DIV_ZERO	R	0	Divisor is zero warning flag 0:The divisor is not zero 1:The divisor is zero
0	DIV_END	R	0	End of division flag 0:Operation in progress 1:End of operation

Puya Confidential

19. Advanced-control timer (TIM1)

19.1. TIM1 introduction

The advanced-control timer (TIM1) is consist of a 16-bit auto-reload counter driven by a programmable prescaler. It can be used in various scenarios, including pulse length measurement of input signals (input capture) or generating output waveforms (output compare, output PWM, complementary PWM with dead-time insertion).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers. The advanced-control (TIM1) and general-purpose (TIMx) timers are completely independent, and do not share any resources. They can be synchronized together.

19.2. TIM1 main features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing dividing (also on the fly) the counter clock frequency either by any factor between 1 and 65536.
- Up to 4 independent channels
  - Input capture
  - Output compare
  - PWM generation (edge or center-aligned mode)
  - One-pulse mode output
- Complementary outputs with programmable dead-time
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Repetition counter to update the timer registers only after a given number of cycles of the counter.
- Break input to put the timer's output signals in reset state or in a known state.
- Interrupt/DMA generation on the following events:
  - Update:counter overflow/underflow, counter initialization (by software or internal/external trigger)
  - Trigger event
  - Input capture
  - Output compare
  - Break input
- Supports incremental (quadrature) encoder and hall-sensor circuitry for positioning purposes
- Trigger input for external clock or cycle-by-cycle current management

Puya Confidential

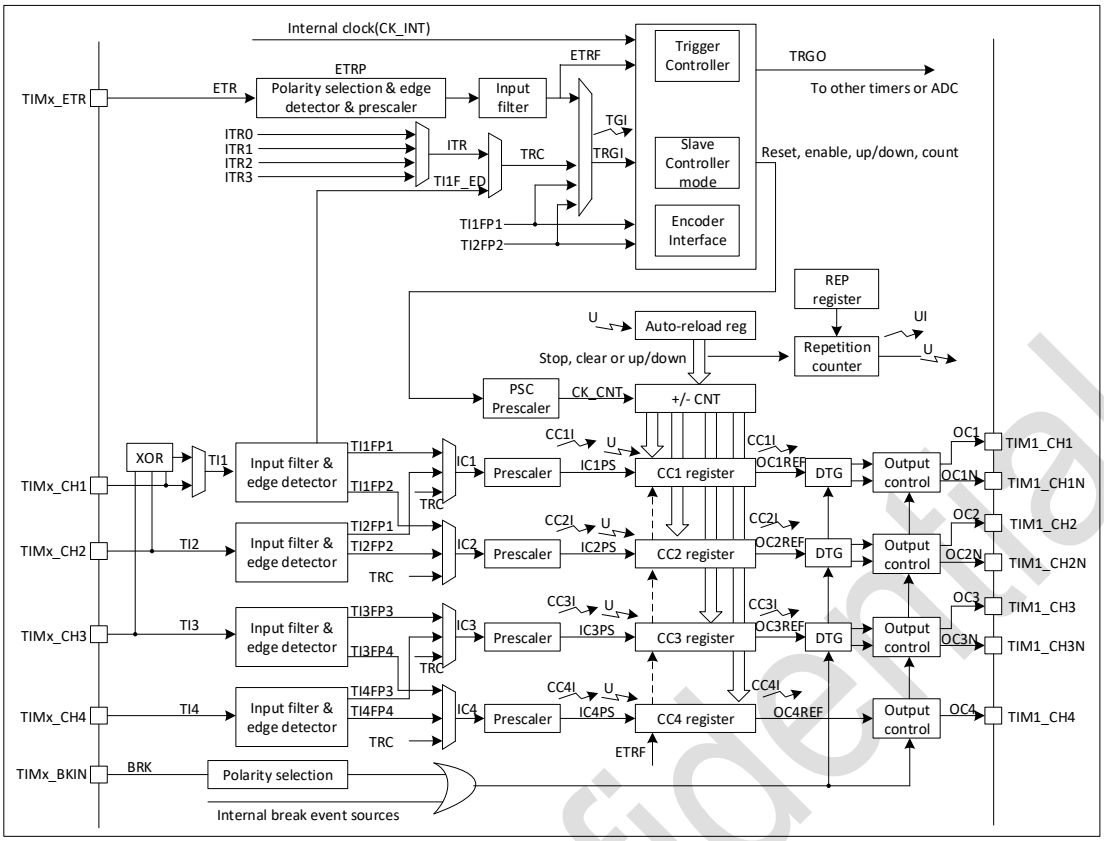


Figure 19-1 Advanced-control timer block diagram

19.3. TIM1 functional description

19.3.1. Time-base unit

The main block of the programmable advanced-control timer is a 16-bit counter with its related auto-reload register. The counter can count up, down or both up and down. The counter clock can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.

The time-base unit includes:

- Counter register (TIMx_CNT)
- Prescaler register (TIMx_PSC)
- Auto-reload register (TIMx_ARR)
- Repetition counter register (TIMx_RCR)

The auto-load register is preloaded. Writing to or reading from the auto-load register accesses the preload register. The content of the preload register is transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload preload enable bit (ARPE) in TIMx_CR1 register. The update event is sent when the counter reaches the overflow (or underflow when downcounting) and if the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software.

Puya Confidential

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in TIMx_CR1 register is set.

Note that the counter starts counting 1 clock cycle after setting the CEN bit in the TIMx_CR register.

Prescaler description

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIMx_PSC register). It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

Figures below give some examples of the counter behavior when the prescaler ratio is changed on the fly:

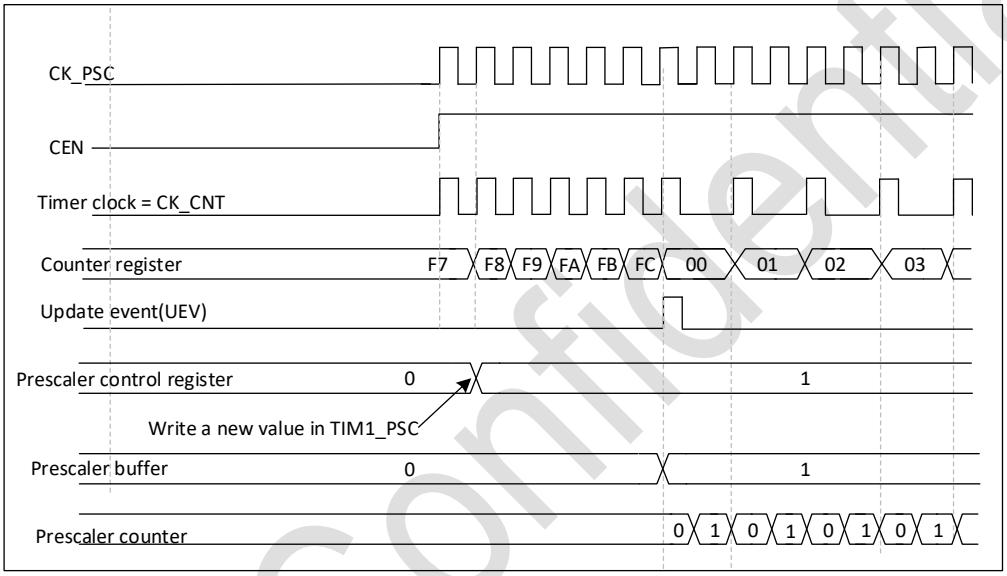


Figure 19-2 Counter timing diagram with prescaler division change from 1 to 2

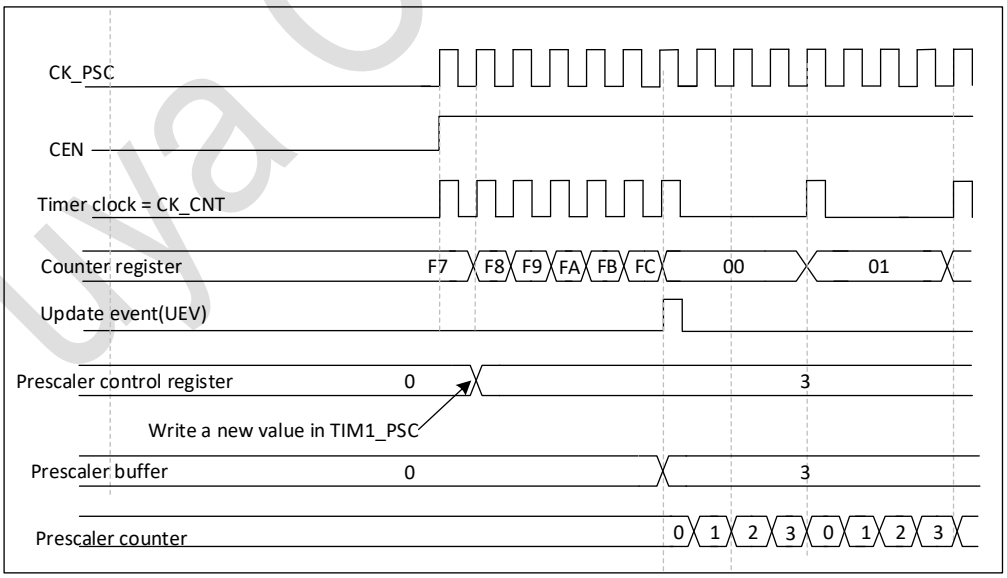


Figure 19-3 Counter timing diagram with prescaler division change from 1 to 4

Puya Confidential

19.3.2. Counter mode

Upcounting mode

In upcounting mode, the counter counts from 0 to the auto-reload value, then restarts from 0 and generates a counter overflow event.

If the repetition counter is used, the update event (UEV) is generated after downcounting is repeated for the number of times programmed in the repetition counter register (TIMx_RCR). Else the update event is generated at each counter underflow.

Setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller) also generates an update event.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts from 0, as well as the counter of the prescaler (but the prescale rate does not change). In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag (thus no interrupt or DMA request is sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The repetition counter is reloaded with the content of TIMx_RCR register.
- The auto-reload shadow register is updated with the preload value (TIMx_ARR).
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR=0x36.

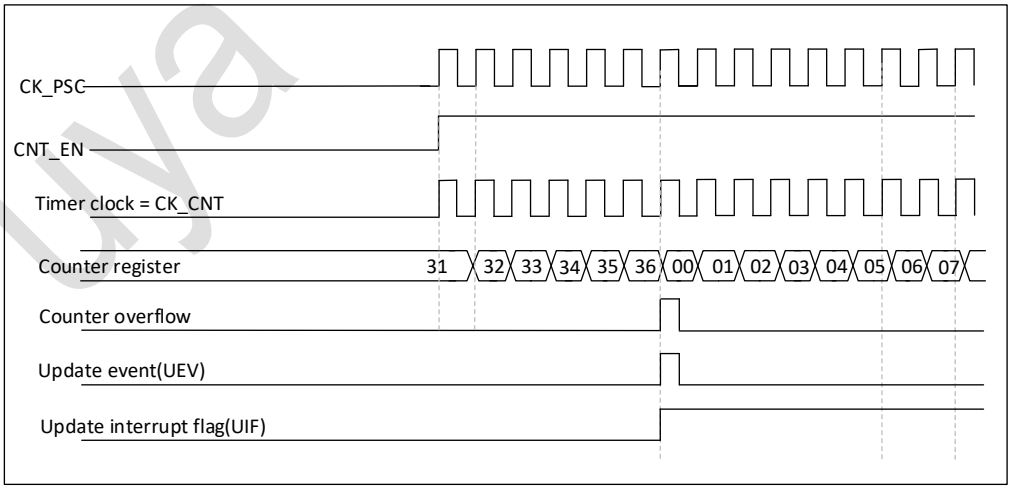


Figure 19-4 Counter timing diagram, internal clock divided by 1

Puya Confidential

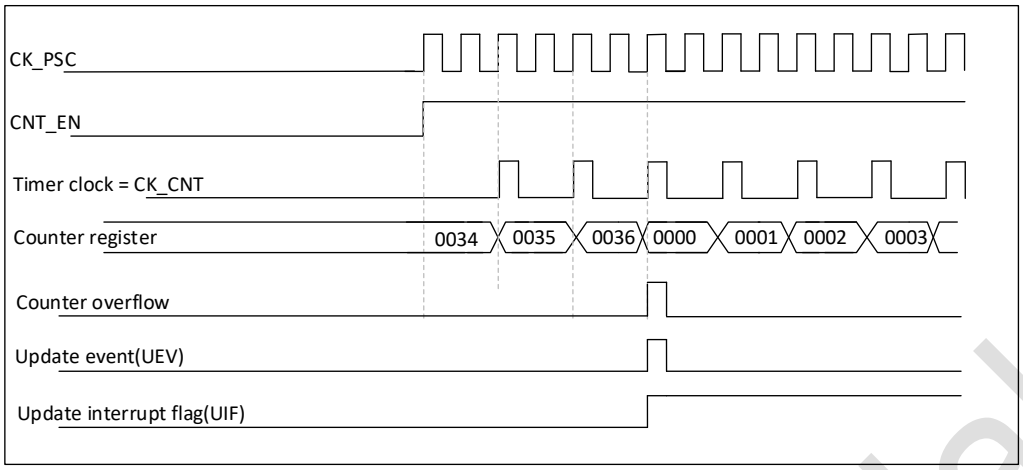


Figure 19-5 Counter timing diagram, internal clock divided by 2

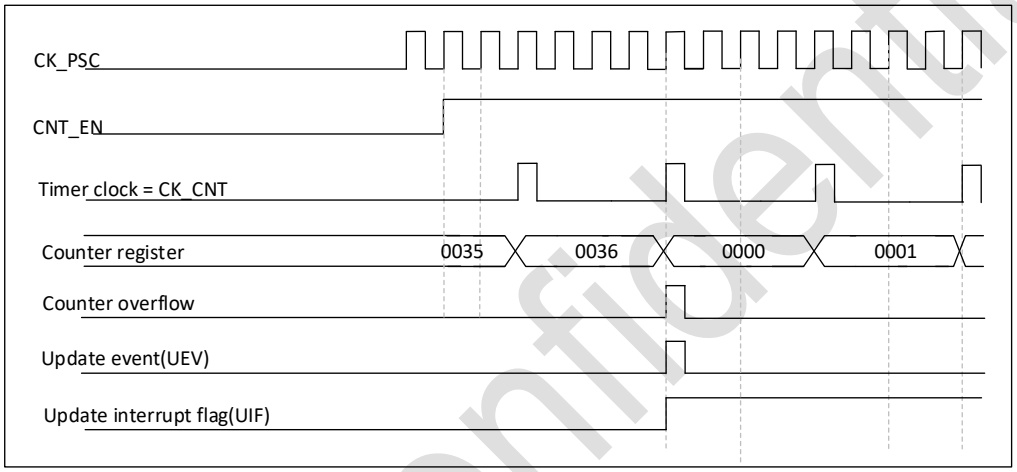


Figure 19-6 Counter timing diagram, internal clock divided by 4

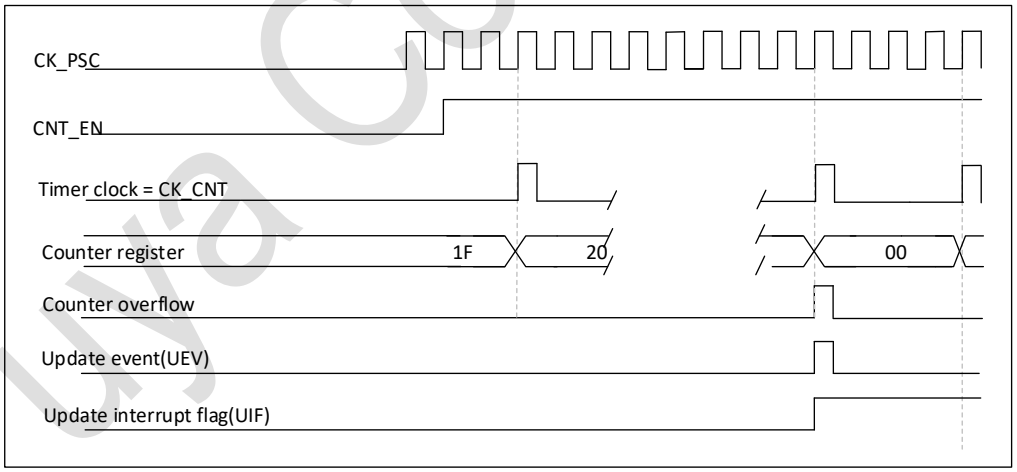


Figure 19-7 Counter timing diagram, internal clock divided by N

Puya Confidential

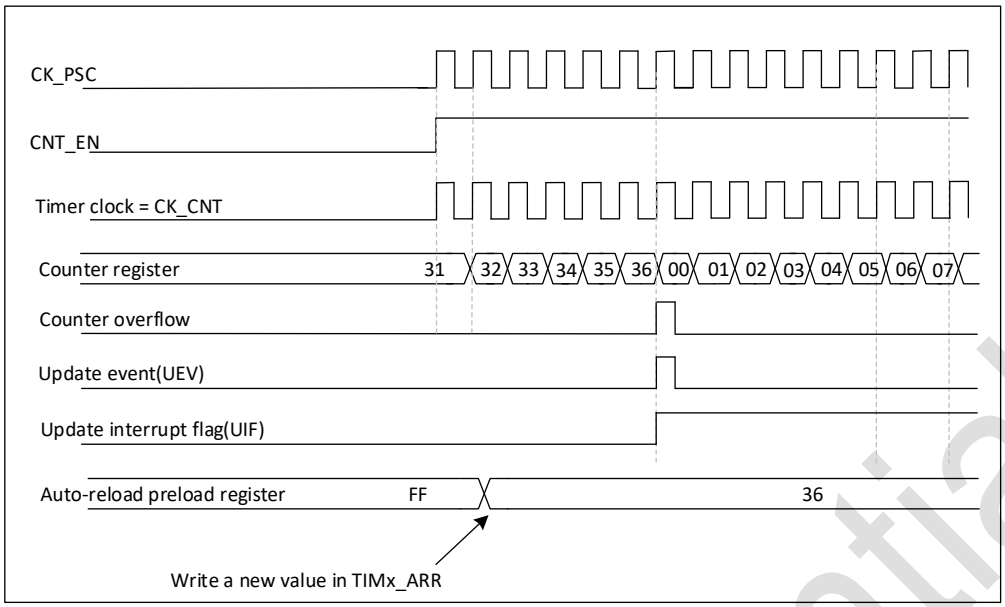


Figure 19-8 Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded)

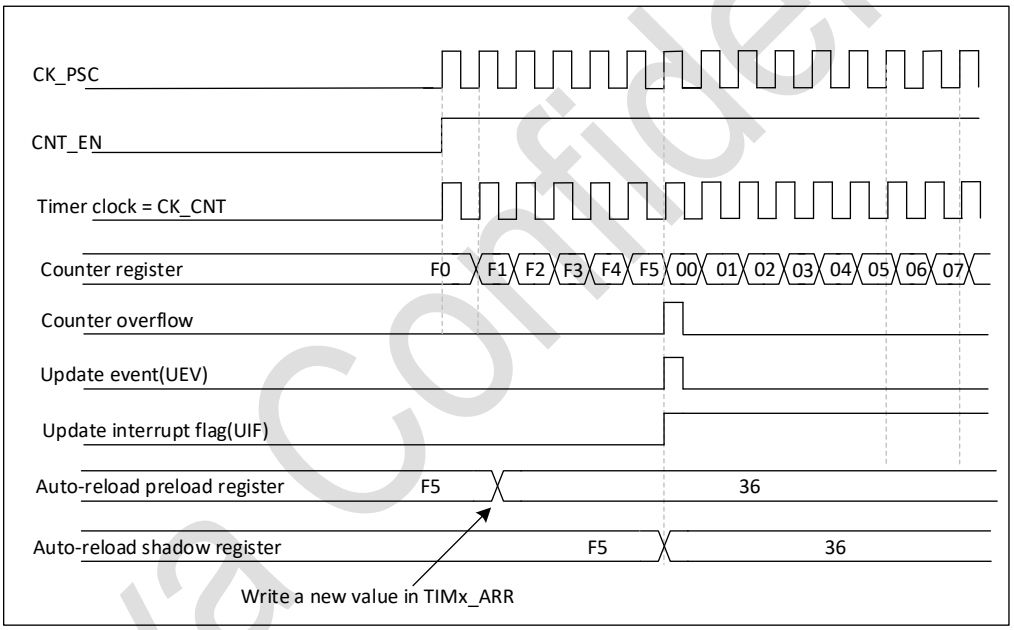


Figure 19-9 Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded)

Downcounting mode

In downcounting mode, the counter counts from the auto-reload value down to 0, then restarts from the auto-reload value and generates a counter underflow event.

If the repetition counter is used, the update event (UEV) is generated after downcounting is repeated for the number of times programmed in the repetition counter register (TIMx_RCR). Else the update event is generated at each counter underflow.

Setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller) also generates an update event.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until UDIS bit has been written to 0. However, the counter restarts from the current

Puya Confidential

auto-reload value, whereas the counter of the prescaler restarts from 0 (but the prescaler rate doesn't change).

In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an UEV update event but without setting the UIF flag (thus no interrupt and DMA requests are sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The repetition counter is reloaded with the content of TIMx_RCR register
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).
- The auto-reload active register is updated with the preload value (content of the TIMx_ARR register).

Note that the auto-reload is updated before the counter is reloaded, so that the next period is the expected one.

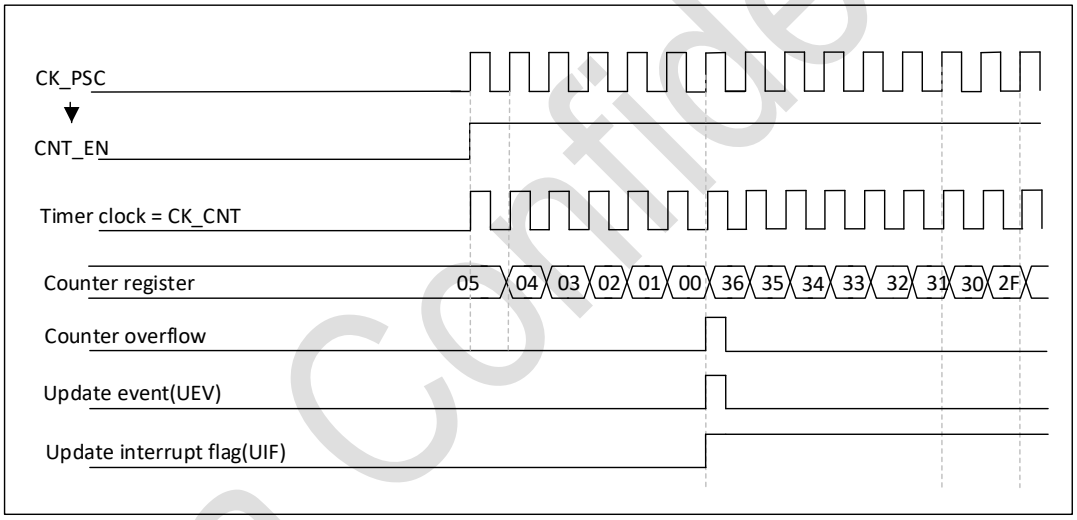


Figure 19-10 Counter timing diagram, internal clock divided by 1

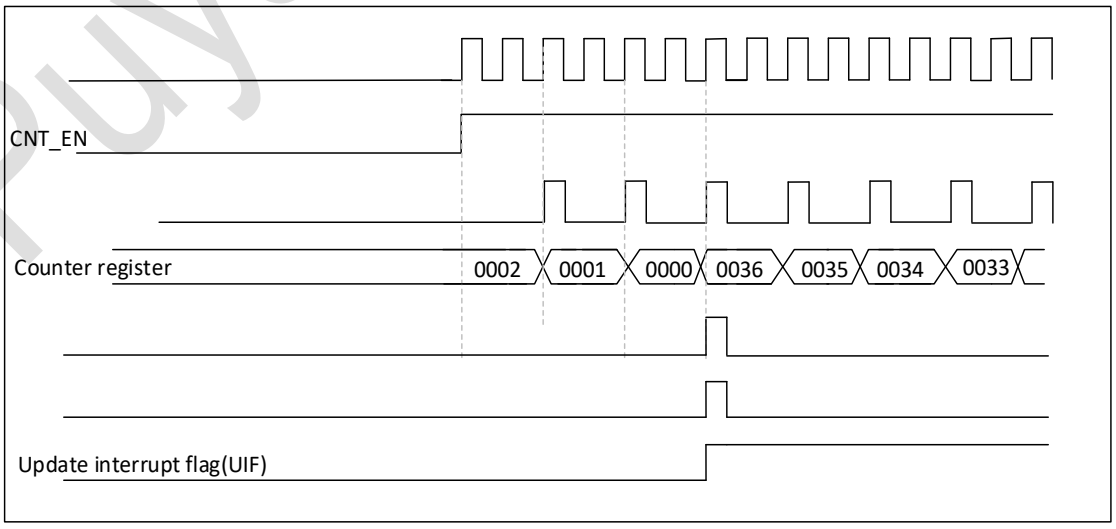


Figure 19-11 Counter timing diagram, internal clock divided by 2

Puya Confidential

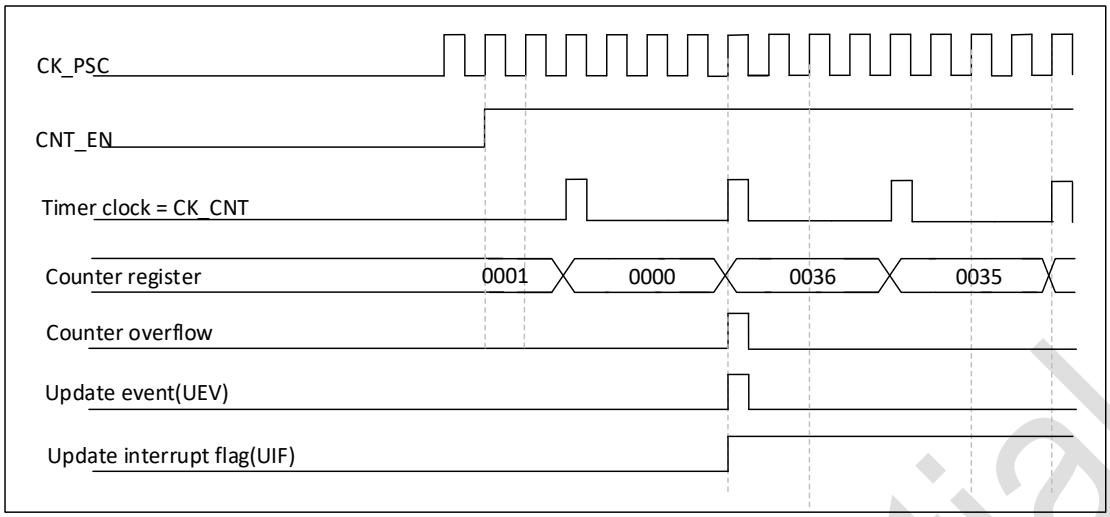


Figure 19-12 Counter timing diagram, internal clock divided by 4

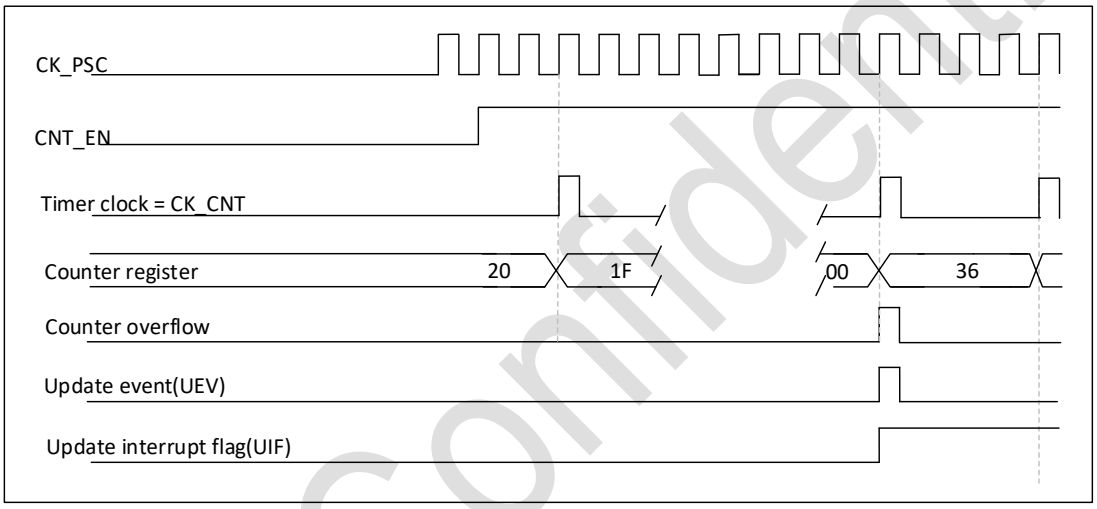


Figure 19-13 Counter timing diagram, internal clock divided by N

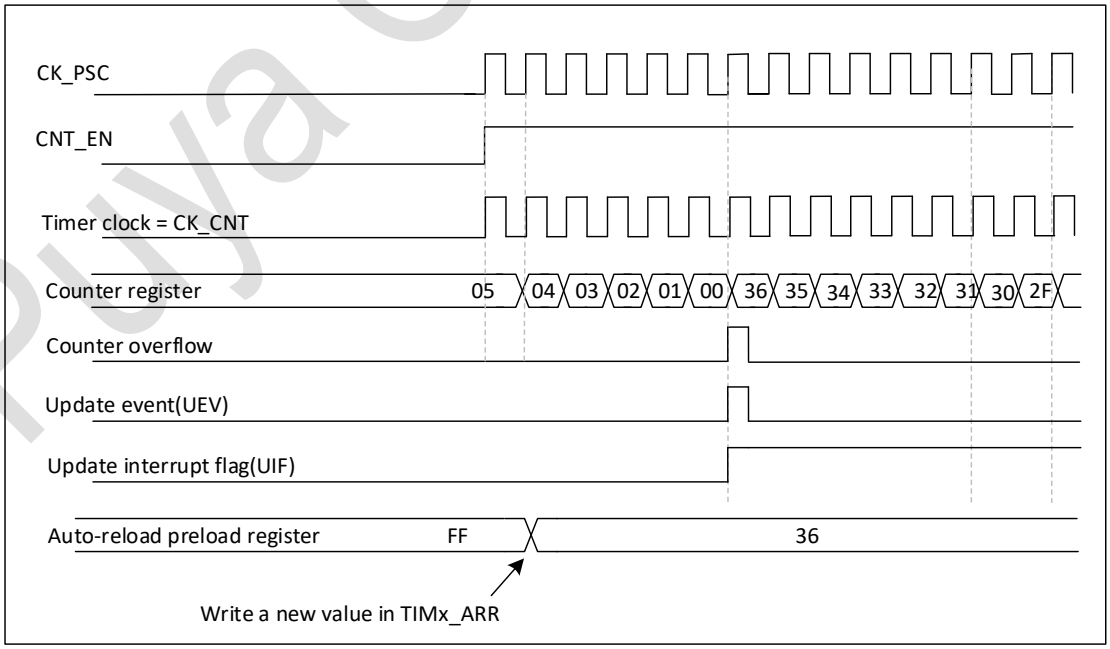


Figure 19-14 Counter timing diagram, update event when repetition counter is not used

Puya Confidential

Center-aligned mode (up/down counting)

In center-aligned mode, the counter counts from 0 to the auto-reload value (content of the TIMx_ARR register) – 1, generates a counter overflow event, then counts from the auto-reload value down to 1 and generates a counter underflow event. Then it restarts counting from 0.

Center-aligned mode is active when the CMS bits in TIMx_CR1 register are not equal to '0'. The Output compare interrupt flag of channels configured in output is set when:the counter counts down (Center aligned mode 1, CMS = “01”), the counter counts up (Center aligned mode 2, CMS = “10”) the counter counts up and down (Center aligned mode 3, CMS = “11”).

In this mode, the DIR direction bit in the TIMx_CR1 register cannot be written. It is updated by hardware and gives the current direction of the counter.

The update event can be generated at each counter overflow and at each counter underflow or by setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller) also generates an update event. In this case, the counter restarts counting from0, as well as the counter of the prescaler.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until UDIS bit has been written to 0. However, the counter continues counting up and down, based on the current auto-reload value.

In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an UEV update event but without setting the UIF flag (thus no interrupt and DMA requests are sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The repetition counter is reloaded with the content of TIMx_RCR register
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).
- The auto-reload active register is updated with the preload value (content of the TIMx_ARR register).

Note that if the update source is a counter overflow, the auto-reload is updated before the counter is reloaded, so that the next period is the expected one (the counter is loaded with the new value).

Puya Confidential

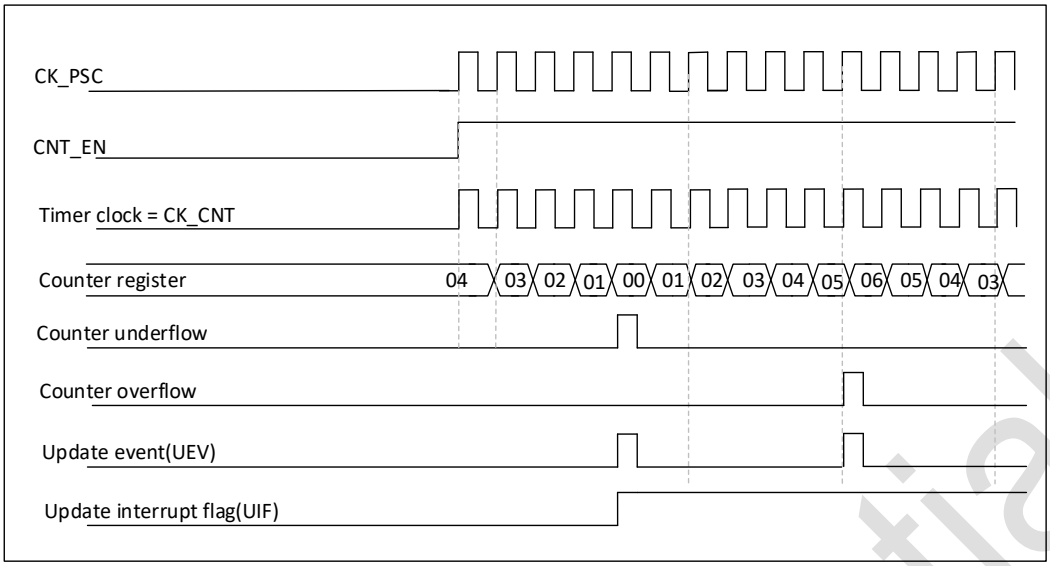


Figure 19-15 Counter timing diagram, internal clock divided by 1, TIMx_ARR = 0x6

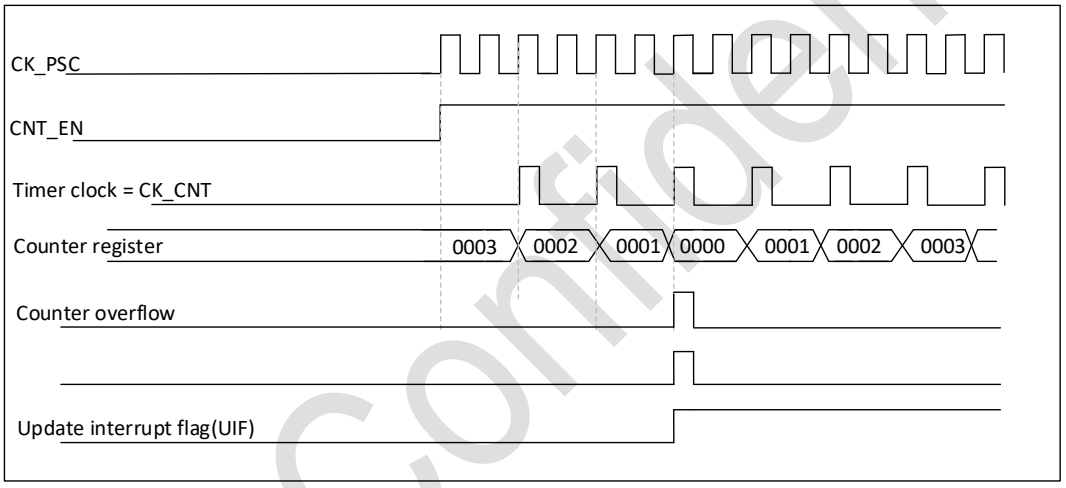


Figure 19-16 Counter timing diagram, internal clock divided by 2, TIMx_ARR=0x36

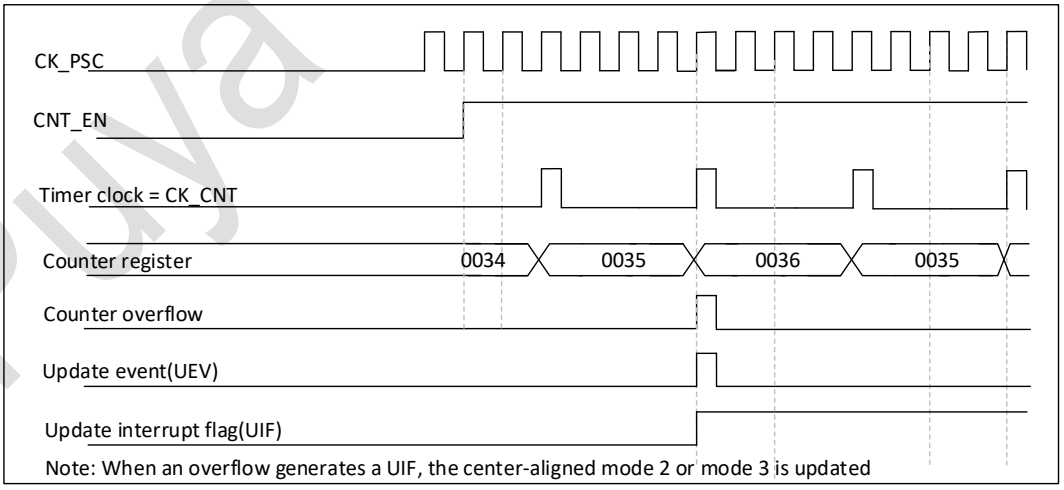


Figure 19-17 Counter timing diagram, internal clock divided by 4, TIMx_ARR=0x36

Puya Confidential

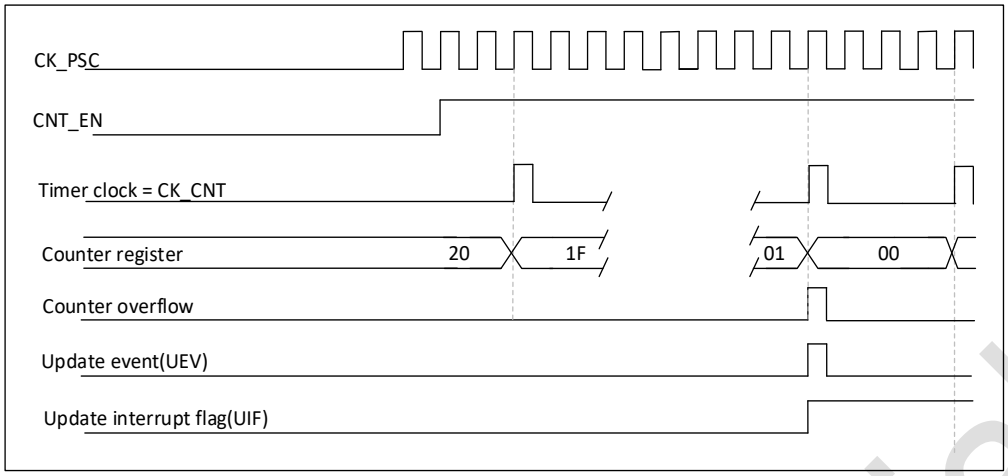


Figure 19-18 Counter timing diagram, internal clock divided by N

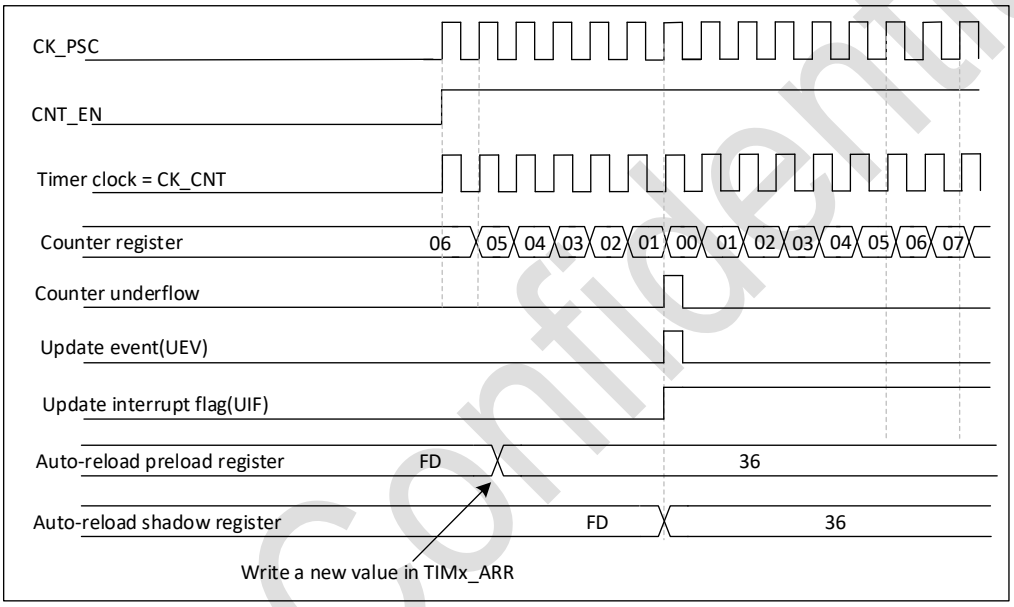


Figure 19-19 Counter timing diagram, update event with ARPE=1 (counter underflow)

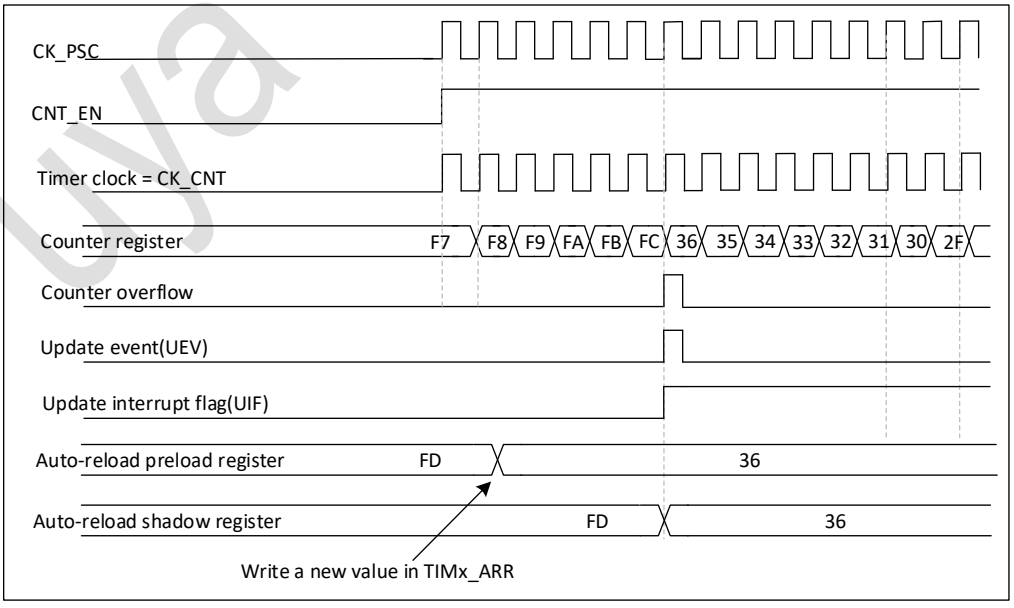


Figure 19-20 Counter timing diagram, update event with ARPE=1 (counter overflow)

Puya Confidential

19.3.3. Repetition counter

Time-base unit describes how the update event (UEV) is generated with respect to the counter overflows/underflows. It is actually generated only when the repetition counter has reached zero. This can be useful when generating PWM signals.

This means that data are transferred from the preload registers to the shadow registers (TIMx_ARR auto-reload register, TIMx_PSC prescaler register, but also TIMx_CCRx capture/compare registers in compare mode) every N + 1 counter overflows or underflows, where N is the value in the TIMx_RCR repetition counter register.

The repetition counter is decremented:

- At each counter overflow in upcounting mode
- At each counter underflow in downcounting mode
- At each counter overflow and at each counter underflow in center-aligned mode.

Although this limits the maximum number of repetition to 128 PWM cycles, it makes it possible to update the duty cycle twice per PWM period. When refreshing compare registers only once per PWM period in center-aligned mode, maximum resolution is 2xTck, due to the symmetry of the pattern.

The repetition counter is an auto-reload type; the repetition rate is maintained as defined by the TIMx_RCR register value. When the update event is generated by software (by setting the UG bit in TIMx_EGR register) or by hardware through the slave mode controller, it occurs immediately whatever the value of the repetition counter is, and the repetition counter is reloaded with the content of the TIMx_RCR register.

In center-aligned mode, for odd values of RCR, the update event occurs either on the overflow or on the underflow depending on when the RCR register was written and when the counter was started. If the RCR was written before starting the counter, the UEV occurs on the overflow. For example, for RCR = 3, the UEV is generated on each 4th overflow or underflow event depending on when RCR was written.

Puya Confidential

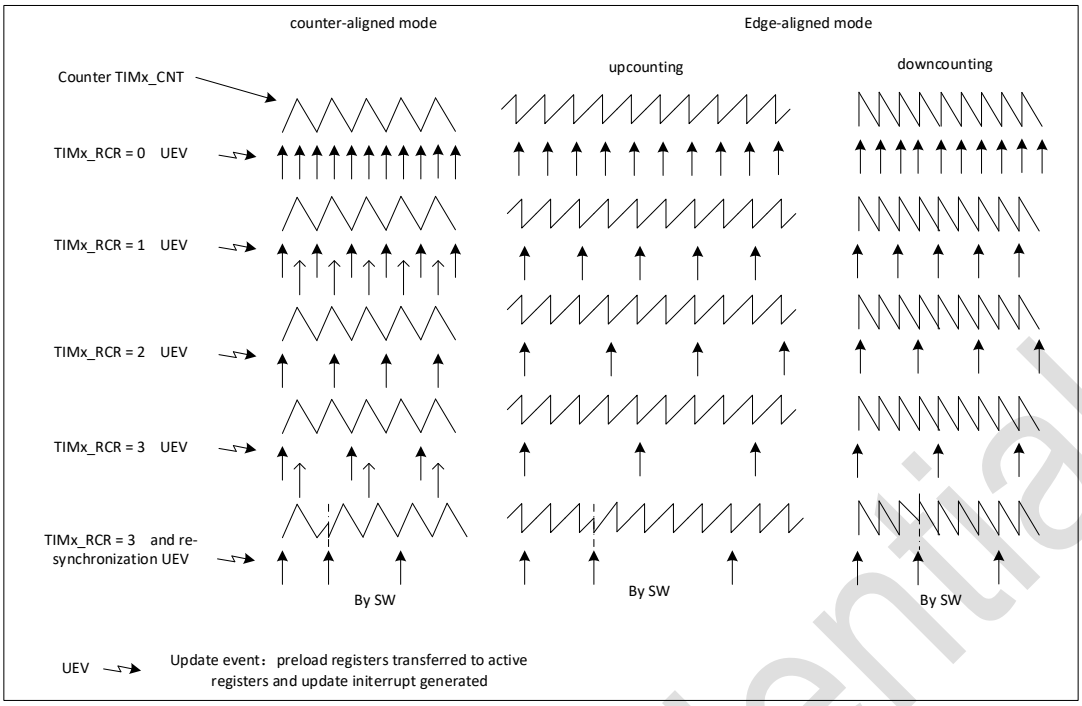


Figure 19-21 Update rate examples depending on mode and TIMx_RCR register settings

19.3.4. Clock sources

The counter clock can be provided by the following clock sources:

- Internal clock (CK_INT)
- External clock mode1:external input pin
- External clock mode2:external trigger input ETR
- Internal trigger inputs (ITRx):using one timer as prescaler for another timer. For example, the user can configure Timer 1 to act as a prescaler for Timer 2.

Internal clock source (CK_INT)

If the slave mode controller is disabled, then the CEN, DIR (in the TIMx_CR1 register) and UG bits (in the TIMx_EGR register) are actual control bits and can be changed only by software. As soon as the CEN bit is written to 1, the prescaler is clocked by the internal clock CK_INT.

The following figure shows the behavior of the control circuit and the upcounter in normal mode, without prescaler.

Puya Confidential

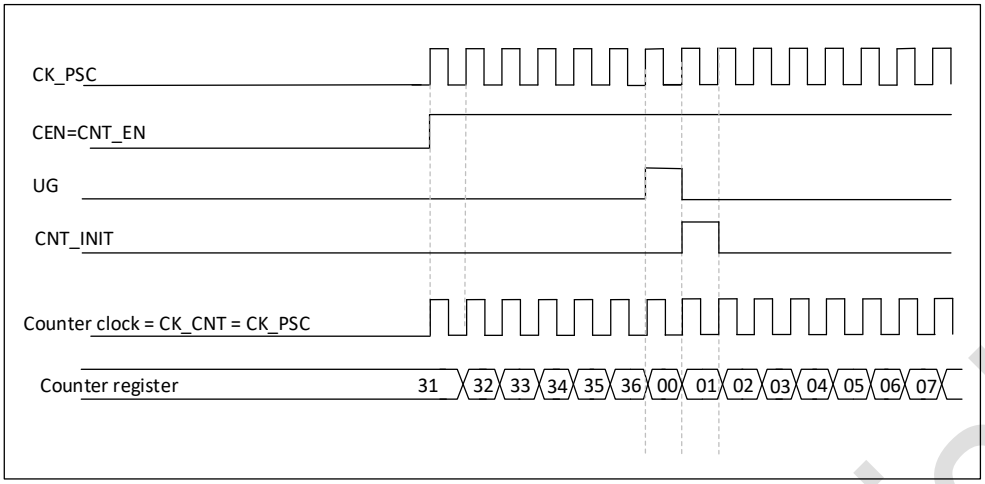


Figure 19-22 Control circuit in normal mode, internal clock divided by 1

External clock source mode 1

This mode is selected when SMS=111 in the TIMx_SMCR register. The counter can count at each rising or falling edge on a selected input.

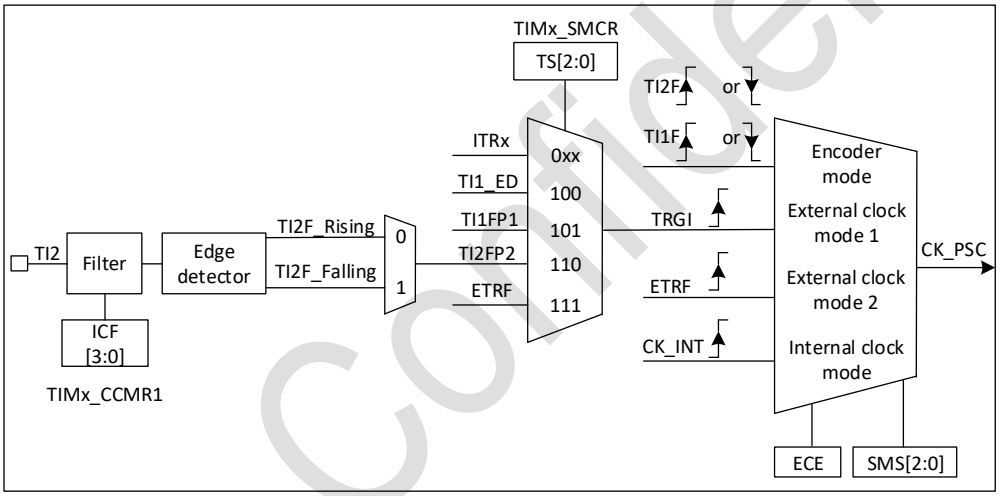


Figure 19-23 TI2 external clock connection example

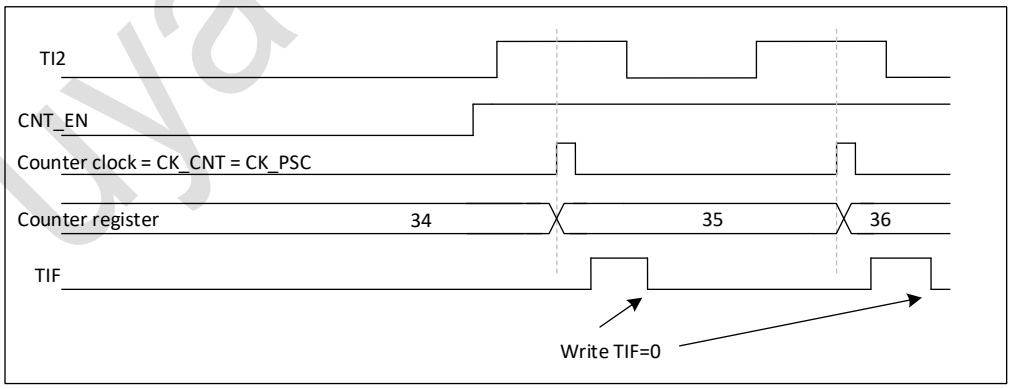


Figure 19-24 Control circuit in external clock mode 1

External clock source mode 2

This mode is selected by writing ECE=1 in the TIMx_SMCR register. The counter can count at each rising or falling edge on the external trigger input ETR.

Puya Confidential

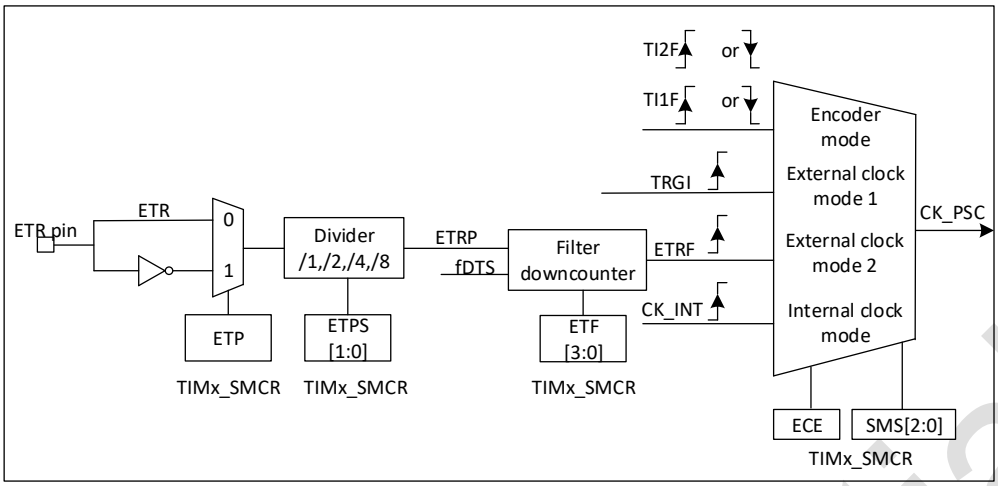


Figure 19-25 TI2 external trigger input block

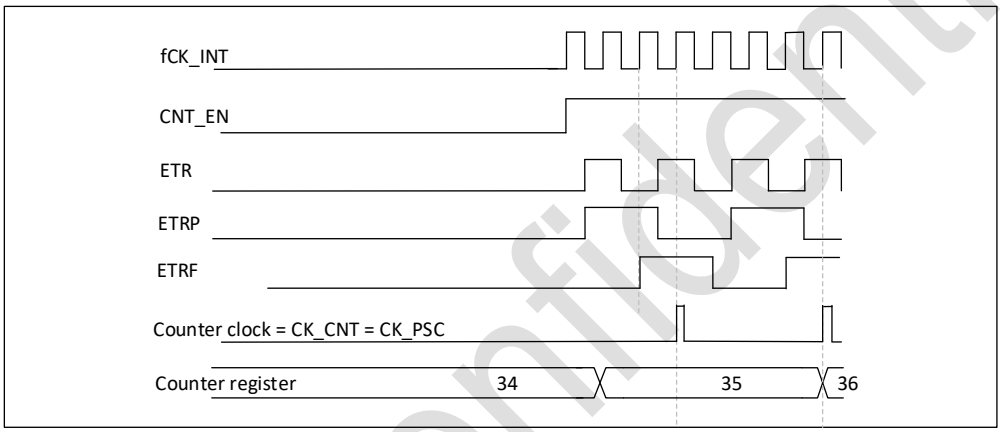


Figure 19-26 Control circuit in external clock mode 2

19.3.5. Capture/Compare channels

Each Capture/Compare channel is built around a capture/compare register (including a shadow register), an input stage for capture (with digital filter, multiplexing and prescaler) and an output stage (with comparator and output control).

The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detector with polarity selection generates a signal (TIxFPx) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (ICxPS).

Puya Confidential

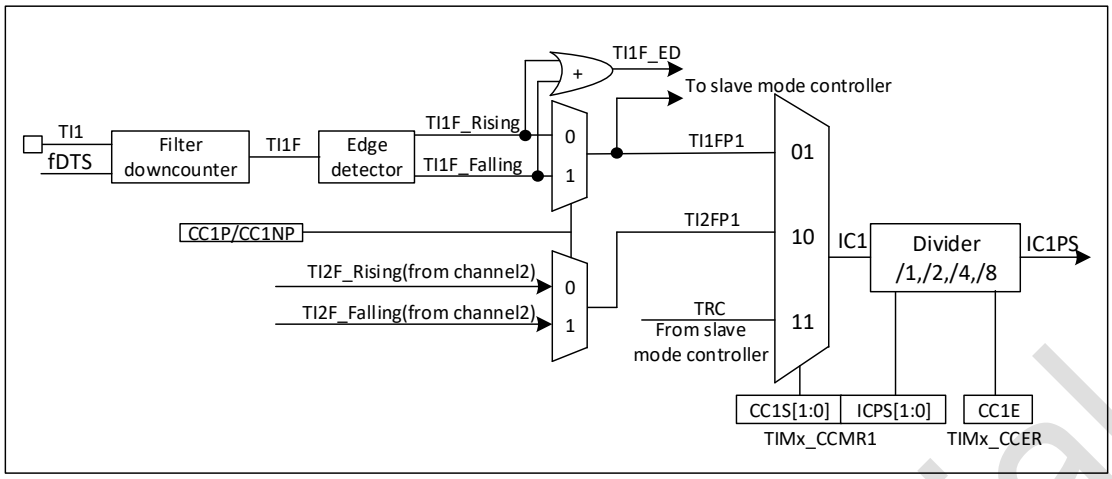


Figure 19-27 Capture/compare channel (example:channel 1 input stage)

The output stage generates an intermediate waveform which is then used for reference:OCxREF (active high). The polarity acts at the end of the chain.

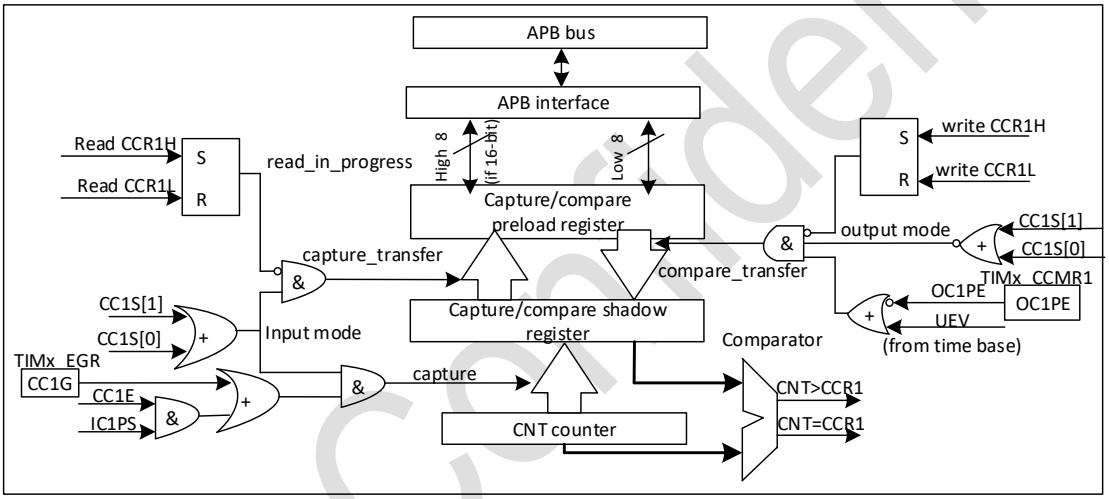


Figure 19-28 Capture/compare channel 1 main circuit

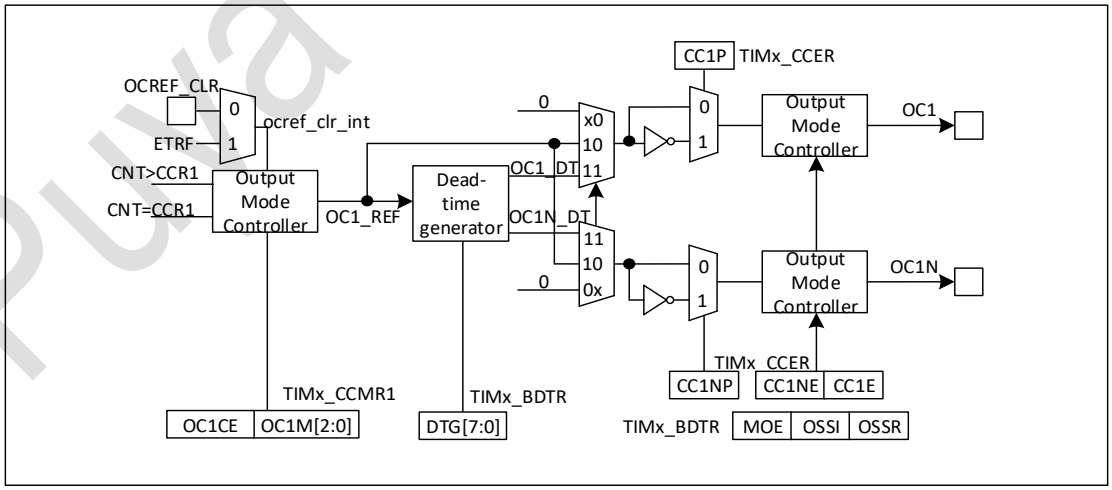


Figure 19-29 Output stage of capture/compare channel (channel 1 to 3)

Puya Confidential

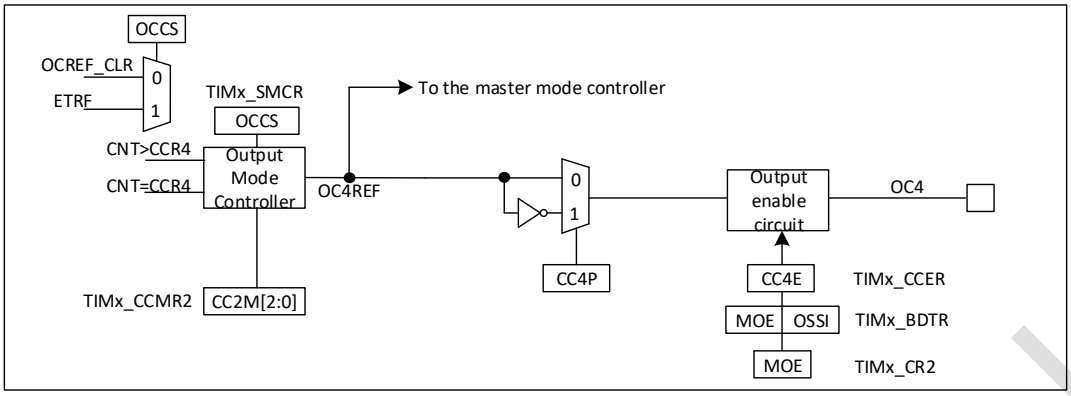


Figure 19-30 Output stage of capture/compare channel (channel 4)

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register.

In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

19.3.6. Input capture mode:

In Input capture mode, the capture/compare registers are used to latch the value of the counter after a transition detected by the corresponding ICx signal. When a capture occurs, the corresponding CCXIF flag (TIMx_SR register) is set and an interrupt or a DMA request can be sent if they are enabled. If a capture occurs while the CCxIF flag was already high, then the over-capture flag CCxOF (TIMx_SR register) is set. CCxIF can be cleared by software by writing it to ‘0’ or by reading the captured data stored in the TIMx_CCRx register. CCxOF is cleared when you write it to ‘0’.

The following example shows how to capture the counter value in TIMx_CCR1 when TI1 input rises. To do this, use the following procedure:

1. Select the active input:TIMx_CCR1 must be linked to the TI1 input, so write the CC1S bits to 01 in the TIMx_CCMR1 register. As soon as CC1S becomes different from 00, the channel is configured in input and the TIMx_CCR1 register becomes read-only.
2. Program the needed input filter duration with respect to the signal connected to the timer (by programming ICxF bits in the TIMx_CCMRx register if the input is a TIx input). Let’s imagine that, when toggling, the input signal is not stable during at most 5 internal clock cycles. We must program a filter duration longer than these 5 clock cycles. We can validate a transition on TI1 when 8 consecutive samples with the new level have been detected (sampled at f_{DTS} frequency). Then write IC1F bits to 0011 in the TIMx_CCMR1 register.
3. Select the edge of the active transition on the TI1 channel by writing CC1P bit to 0 in the TIMx_CCER register (rising edge in this case).
4. Program the input prescaler. In our example, we wish the capture to be performed at each valid transition, so the prescaler is disabled (write IC1PS bits to ‘00’ in the TIMx_CCMR1 register).

Puya Confidential

5. Enable capture from the counter into the capture register by setting the CC1E bit in the TIMx_CCER register.
6. If needed, enable the related interrupt request by setting the CC1IE bit in the TIMx_DIER register, and/or the DMA request by setting the CC1DE bit in the TIMx_DIER register.

When an input capture occurs:

- The TIMx_CCR1 register gets the value of the counter on the active transition.
- CC1IF flag is set (interrupt flag). CC1OF is also set if at least two consecutive captures occurred whereas the flag was not cleared.
- An interrupt is generated depending on the CC1IE bit.
- A DMA request is generated depending on the CC1DE bit.

In order to handle the overcapture, it is recommended to read the data before the overcapture flag. This is to avoid missing an overcapture which could happen after reading the flag and before reading the data.

Note:IC interrupt and/or DMA requests can be generated by software by setting the corresponding CCxG bit in the TIMx_EGR register.

19.3.7. PWM input mode

This mode is a particular case of input capture mode. The procedure is the same except:

- Two ICx signals are mapped on the same TIx input.
- These 2 ICx signals are active on edges with opposite polarity.
- One of the two TIxFP signals is selected as trigger input and the slave mode controller is configured in reset mode.

For example, user can measure the period (in TIMx_CCR1 register) and the duty cycle (in TIMx_CCR2 register) of the PWM applied on TI1 using the following procedure (depending on CK_INT frequency and prescaler value):

- Select the active input for TIMx_CCR1:write the CC1S bits to 01 in the TIMx_CCMR1 register (TI1 selected).
- Select the active polarity for TI1FP1 (used both for capture in TIMx_CCR1 and counter clear):write the CC1P bit to ‘0’ (active on rising edge).
- Select the active input for TIMx_CCR2:write the CC2S bits to 10 in the TIMx_CCMR1 register (TI1 selected).
- Select the active polarity for TI1FP2 (used for capture in TIMx_CCR2):write the CC2Pbit to ‘1’ (active on falling edge).
- Select the valid trigger input:write the TS bits to 101 in the TIMx_SMCR register (TI1FP1 selected).
- Configure the slave mode controller in reset mode:write the SMS bits to 100 in the TIMx_SMCR register.
- Enable the captures:write the CC1E and CC2E bits to ‘1’ in the TIMx_CCER register.

Puya Confidential

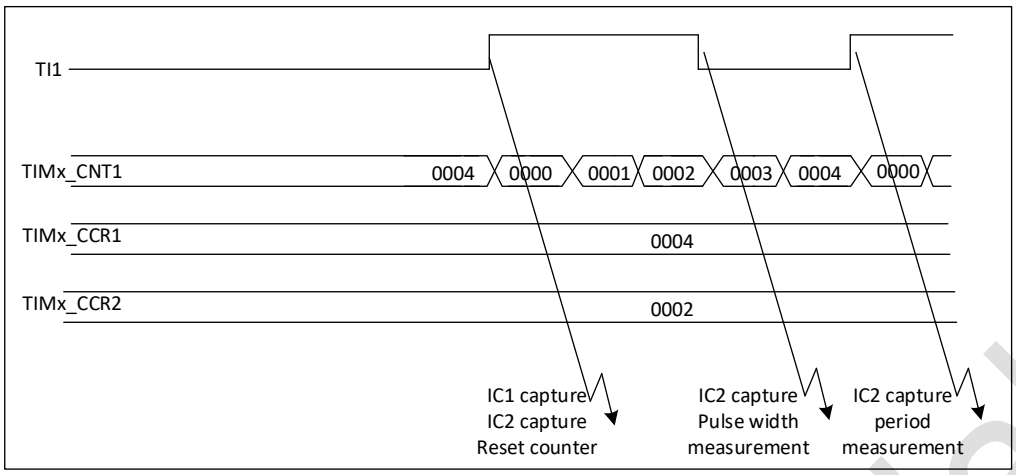


Figure 19-31 PWM input mode timing

19.3.8. Forced output mode

In output mode (CCxS bits = 00 in the TIMx_CCMRx register), each output compares signal (OCxREF and then OCx/OCxN) can be forced to active or inactive level directly by software, independently of any comparison between the output compare register and the counter. To force an output compare signal (OCXREF/OCx) to its active level, you just need to write 101 in the OCxM bits in the corresponding TIMx_CCMRx register. Thus OCxREF is forced high (OCxREF is always active high) and OCx get opposite value to CCxP polarity bit.

For example:CCxP=0 (OCx active high) => OCx is forced to high level. The OCxREF signal can be forced low by writing the OCxM bits to 100 in the TIMx_CCMRx register.

Anyway, the comparison between the TIMx_CCRx shadow register and the counter is still performed and allows the flag to be set. Interrupt and DMA requests can be sent accordingly. This is described in the output compare mode section below.

19.3.9. Output compare mode

This function is used to control an output waveform or indicating when a period of time has elapsed. When a match is found between the capture/compare register and the counter, the output compare function:

- Assigns the corresponding output pin to a programmable value defined by the output compare mode (OCxM bits in the TIMx_CCMRx register) and the output polarity (CCxP bit in the TIMx_CCER register). The output pin can keep its level (OCXM=000), be set active (OCxM=001), be set inactive (OCxM=010) or can toggle (OCxM=011) on match.
- Sets a flag in the interrupt status register (CCxIF bit in the TIMx_SR register).
- Generates an interrupt if the corresponding interrupt mask is set (CCXIE bit in the TIMx_DIER register).
- Sends a DMA request if the corresponding enable bit is set (CCxDE bit in the TIMx_DIER register, CCDS bit in the TIMx_CR2 register for the DMA request selection).

Puya Confidential

The TIMx_CCRx registers can be programmed with or without preload registers using the OCxPE bit in the TIMx_CCMRx register. In output compare mode, the update event UEV has no effect on OCxREF and OCx output.

The timing resolution is one count of the counter. Output compare mode can also be used to output a single pulse (in one pulse mode).

Procedure:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data in the TIMx_ARR and TIMx_CCRx registers.
3. Set the CCxIE bit if an interrupt request is to be generated.
4. Select the output mode. For example:
   - Write OCxM = 011 to toggle OCx output pin when CNT matches CCRx
   - Write OCxPE = 0 to disable preload register
   - Write CCxP = 0 to select active high polarity
   - Write CCxE = 1 to enable the output
5. Enable the counter by setting the CEN bit in the TIMx_CR1 register.

The TIMx_CCRx register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled (OCxPE= ’0’, else TIMx_CCRx shadow register is updated only at the next update event UEV). An example is given in the figure below.

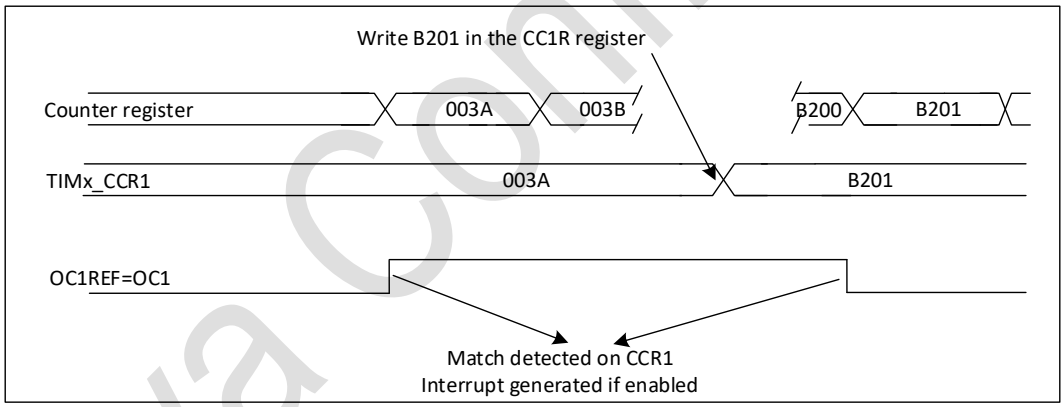


Figure 19-32 Output compare mode, toggle on OC1

19.3.10. PWM mode

Pulse Width Modulation mode allows you to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register.

The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing ‘110’ (PWM mode 1) or ‘111’ (PWM mode 2) in the OCxM bits in the TIMx_CCMRx register. You must enable the corresponding preload register by setting the OCxPE bit in the TIMx_CCMRx register, and eventually the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIMx_CR1 register.

As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, all registers must be initialized by setting the UG bit in the TIMx_EGR register.

Puya Confidential

OCx polarity is software programmable using the CCxP bit in the TIMx_CCER register. It can be programmed as active high or active low. OCx output is enabled by a combination of the CCxE, CCxNE, MOE, OSSI and OSSR bits (TIMx_CCER and TIMx_BDTR registers). Refer to the TIMx_CCER register description for more details.

In PWM mode (1 or 2), TIMx_CNT and TIMx_CCRx are always compared to determine whether TIMx_CCRx ≤ TIMx_CNT or TIMx_CNT ≤ TIMx_CCRx (depending on the direction of the counter).

The timer is able to generate PWM in edge-aligned mode or center-aligned mode depending on the CMS bits in the TIMx_CR1 register.

PWM edge-aligned mode

- Upcounting configuration

Upcounting is active when the DIR bit in the TIMx_CR1 register is low. In the following example, we consider PWM mode 1. The reference PWM signal OCxREF is high as long as TIMx_CNT < TIMx_CCRx else it becomes low. If the compare value in TIMx_CCRx is greater than the auto-reload value (in TIMx_ARR) then OCxREF is held at '1'. If the compare value is 0 then OCxRef is held at '0'. Figure below shows some edge-aligned PWM waveforms in an example where TIMx_ARR=8.

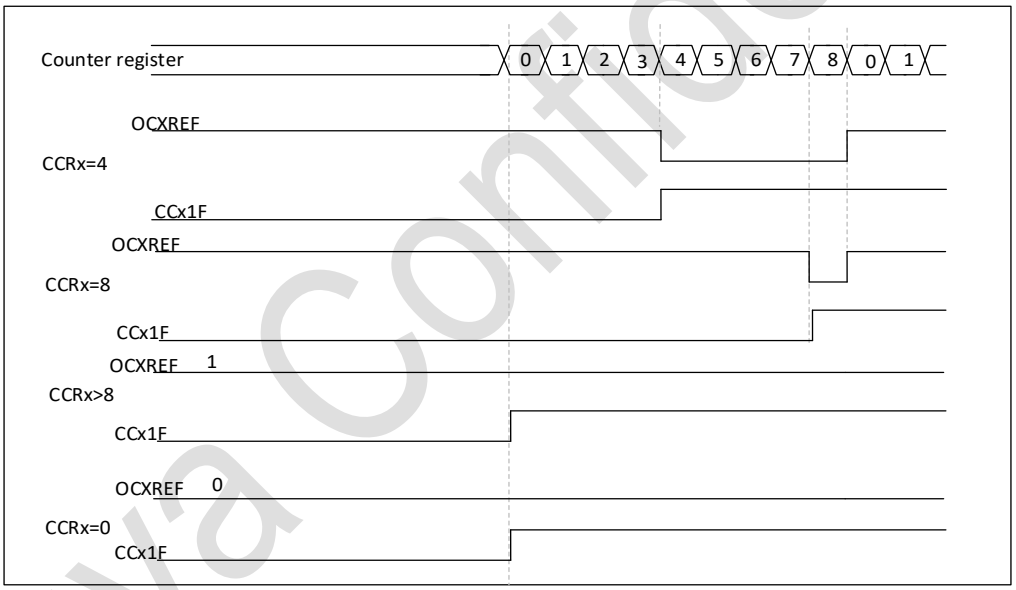


Figure 19-33 Edge-aligned PWM waveforms (ARR=8)

- Downcounting configuration

Downcounting is active when DIR bit in TIMx_CR1 register is high.

In PWM mode 1, the reference signal OCxRef is low as long asTIMx_CNT > TIMx_CCRx else it becomes high. If the compare value in TIMx_CCRx is greater than the auto-reload value in TIMx_ARR, then OCxREF is held at '1'. 0% PWM is not possible in this mode.

PWM center-aligned mode

Center-aligned mode is active when the CMS bits in TIMx_CR1 register are different from '00' (all the remaining configurations having the same effect on the OCxRef/OCx signals). The compare flag is set when the counter counts up, when it counts down or both when it counts up and down depending

Puya Confidential

on the CMS bits configuration. The direction bit (DIR) in the TIMx_CR1 register is updated by hardware and must not be changed by software.

Figure below shows some center-aligned PWM waveforms in an example where:

- TIMx_ARR = 8
- PWM mode is the PWM mode 1
- The flag is set when the counter counts down corresponding to the center-aligned mode 1 selected for CMS=01 in TIMx_CR1 register.

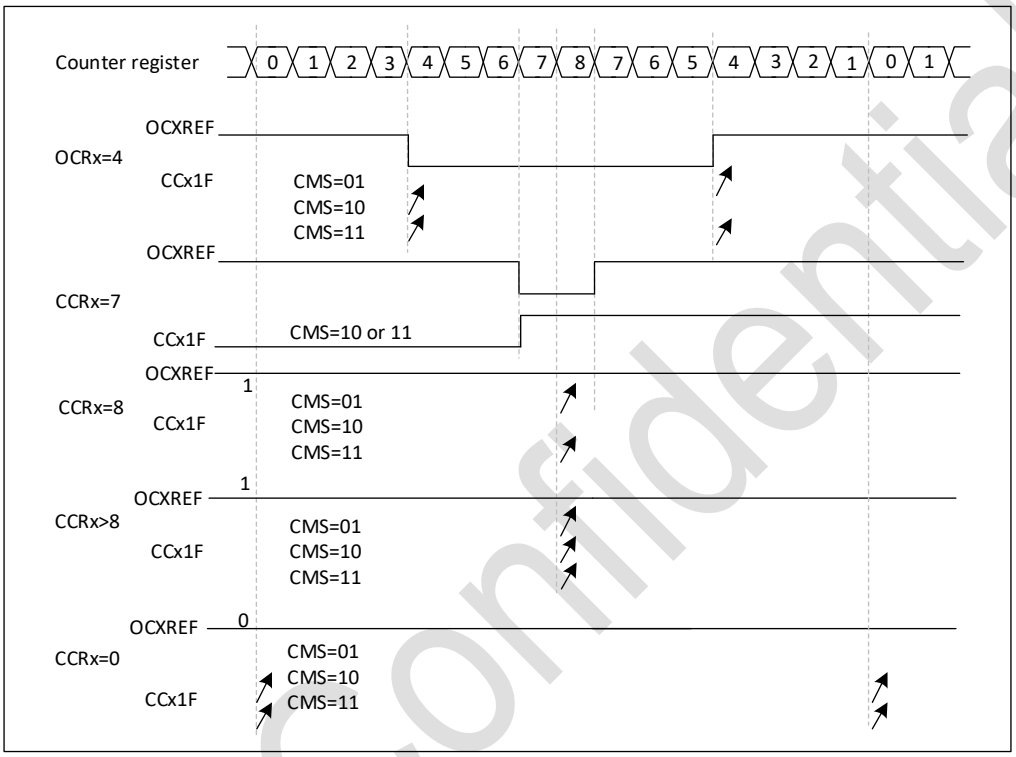


Figure 19-34 Center-aligned PWM waveforms (ARR=8)

Hints on using center-aligned mode:

- When starting in center-aligned mode, the current up-down configuration is used. It means that the counter counts up or down depending on the value written in the DIR bit in the TIMx_CR1 register. Moreover, the DIR and CMS bits must not be changed at the same time by the software.
- Writing to the counter while running in center-aligned mode is not recommended as it can lead to unexpected results. In particular:The direction is not updated if you write a value in the counter that is greater than the auto-reload value (TIMx_CNT>TIMx_ARR). For example, if the counter was counting up, it continues to count up. The direction is updated if you write 0 or write the TIMx_ARR value in the counter but no Update Event UEV is generated.
- The safest way to use center-aligned mode is to generate an update by software (setting the UG bit in the TIMx_EGR register) just before starting the counter and not to write the counter while it is running.

19.3.11. Complementary outputs and dead-time insertion

The advanced-control timers (TIM1) can output two complementary signals and manage the switching-off and the switching-on instants of the outputs. This time is generally known as dead-time, and you

Puya Confidential

have to adjust it depending on the devices you have connected to the outputs and their characteristics (intrinsic delays of level-shifters, delays due to power switches...)

You can select the polarity of the outputs (main output OCx or complementary OCxN) independently for each output. This is done by writing to the CCxP and CCxNP bits in the TIMx_CCER register.

The complementary signals OCx and OCxN are activated by a combination of several control bits:the CCxE and CCxNE bits in the TIMx_CCER register and the MOE, OISx, OISxN, OSSI and OSSR bits in the TIMx_BDTR and TIMx_CR2 registers. In particular, the dead-time is activated when switching to the IDLE state (MOE falling down to 0).

Dead-time insertion is enabled by setting both CCxE and CCxNE bits, and the MOE bit if the break circuit is present. There is one 8-bit dead-time generator for each channel. From a reference waveform OCxREF, it generates 2 outputs OCx and OCxN. If OCx and OCxN are active high:

- The OCx output signal is the same as the reference signal except for the rising edge, which is delayed relative to the reference rising edge.
- The OCxN output signal is the opposite of the reference signal except for the rising edge, which is delayed relative to the reference falling edge.

If the delay is greater than the width of the active output (OCx or OCxN) then the corresponding pulse is not generated.

The following figures show the relationships between the output signals of the dead-time generator and the reference signal OCxREF. (we suppose CCxP=0, CCxNP=0, MOE=1, CCxE=1 and CCxNE=1 in these examples)

Figure 19-35 Complementary output with dead-time insertion

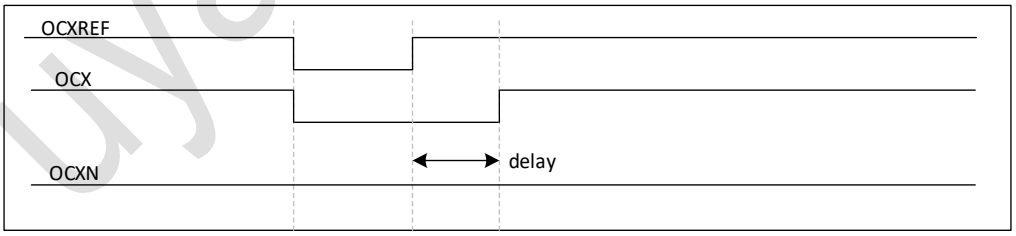


Figure 19-36 Dead-time waveforms with delay greater than the negative pulse

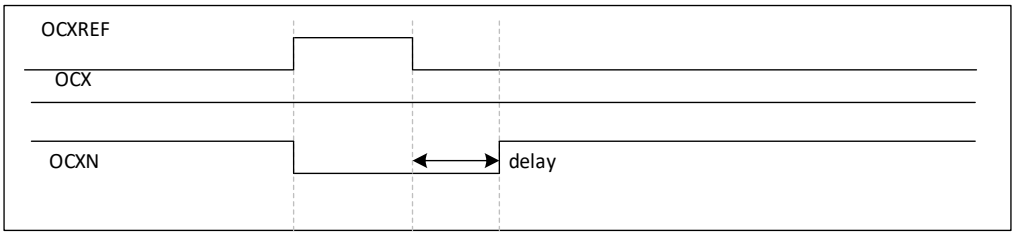


Figure 19-37 Dead-time waveforms with delay greater than the positive pulse.

Puya Confidential

The dead-time delay is the same for each of the channels and is programmable with the DTG bits in the TIMx_BDTR register.

Re-directing OCxREF to OCx or OCxN

In output mode (forced, output compare or PWM), OCxREF can be re-directed to the OCx output or to OCxN output by configuring the CCxE and CCxNE bits in the TIMx_CCER register. This allows you to send a specific waveform (such as PWM or static active level) on one output while the complementary remains at its inactive level. Other alternative possibilities are to have both outputs at inactive level or both outputs active and complementary with dead-time.

Note:When only OCxN is enabled (CCxE=0, CCxNE=1), it is not complemented and becomes active as soon as OCxREF is high. For example, if CCxNP=0 then OCxN=OCxREF. On the other hand, when both OCx and OCxN are enabled (CCxE=CCxNE=1) OCx becomes active when OCxREF is high whereas OCxN is complemented and becomes active when OCxREF is low.

19.3.12. Using the break function

When using the break function, the output enable signals and inactive levels are modified according to additional control bits. In any case, the OCx and OCxN outputs cannot be set both to active level at a given time.

The source for break (BRK) channel can be an external source connected to the BKIN pin or one of the following internal sources:

- the CPU LOCKUP output
- the PVD output
- a clock failure event generated by the CSS detector
- the output from a comparator

When exiting from reset, the break circuit is disabled and the MOE bit is low. You can enable the break function by setting the BKE bit in the TIMx_BDTR register. The break input polarity can be selected by configuring the BKP bit in the same register. BKE and BKP can be modified at the same time. When the BKE and BKP bits are written, a delay of 1 APB clock cycle is applied before the writing is effective. Consequently, it is necessary to wait 1 APB clock period to correctly read back the bit after the write operation.

MOE falling edge can be asynchronous, thus a resynchronization circuit has been inserted between the actual signal (acting on the outputs) and the synchronous control bit (accessed in the TIMx_BDTR register). It results in some delays between the asynchronous and the synchronous signals. In particular, if you write MOE to 1 whereas it was low, you must insert a delay (dummy instruction) before reading it correctly. This is because the write acts on the asynchronous signal whereas the read reflects the synchronous signal.

When a break occurs (selected level on the break input):

- The MOE bit is cleared asynchronously, putting the outputs in inactive state, idle state or in reset state (selected by the OSSI bit). This feature functions even if the MCU oscillator is off.

Puya Confidential

- Each output channel is driven with the level programmed in the OISx bit in the TIMx_CR2 register as soon as MOE=0. If OSSI=0 then the timer releases the enable output else the enable output remains high.
- When complementary outputs are used:
  - The outputs are first put in reset state inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
  - If the timer clock is still present, then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the OISx and OISxN bits after a dead-time. Even in this case, OCx and OCxN cannot be driven to their active level together. Note that because of the resynchronization on MOE, the dead-time duration is a bit longer than usual (around 2 ck_tim clock cycles).
  - If OSSI=0 then the timer releases the enable outputs else the enable outputs remain or become high as soon as one of the CCxE or CCxNE bits is high.
- The break status flag (BIF bit in the TIMx_SR register) is set. An interrupt can be generated if the BIE bit in the TIMx_DIER register is set.
- If the AOE bit in the TIMx_BDTR register is set, the MOE bit is automatically set again at the next update event UEV. This can be used to perform a regulation, for instance. Else, MOE remains low until it is written to '1' again. In this case, it can be used for security and the break input can be connected to an alarm from power drivers, thermal sensors or any security components.

Note:The break inputs are active on level. Thus, the MOE cannot be set while the break input is active (neither automatically nor by software). In the meantime, the status flag BIF cannot be cleared.

The break can be generated by the BRK input which has a programmable polarity and an enable bit BG in the TIMx_BDTR Register.

In addition to the break input and the output management, a write protection has been implemented inside the break circuit to safeguard the application. It allows to freeze the configuration of several parameters (dead-time duration, OCx/OCxN polarities and state when disabled, OCxM configurations, break enable and polarity). The application can choose from 3 levels of protection selected by the LOCK bits in the TIMx_BDTR register. The LOCK bits can be written only once after an MCU reset.

The figure below shows an example of behavior of the outputs in response to a break.

Puya Confidential

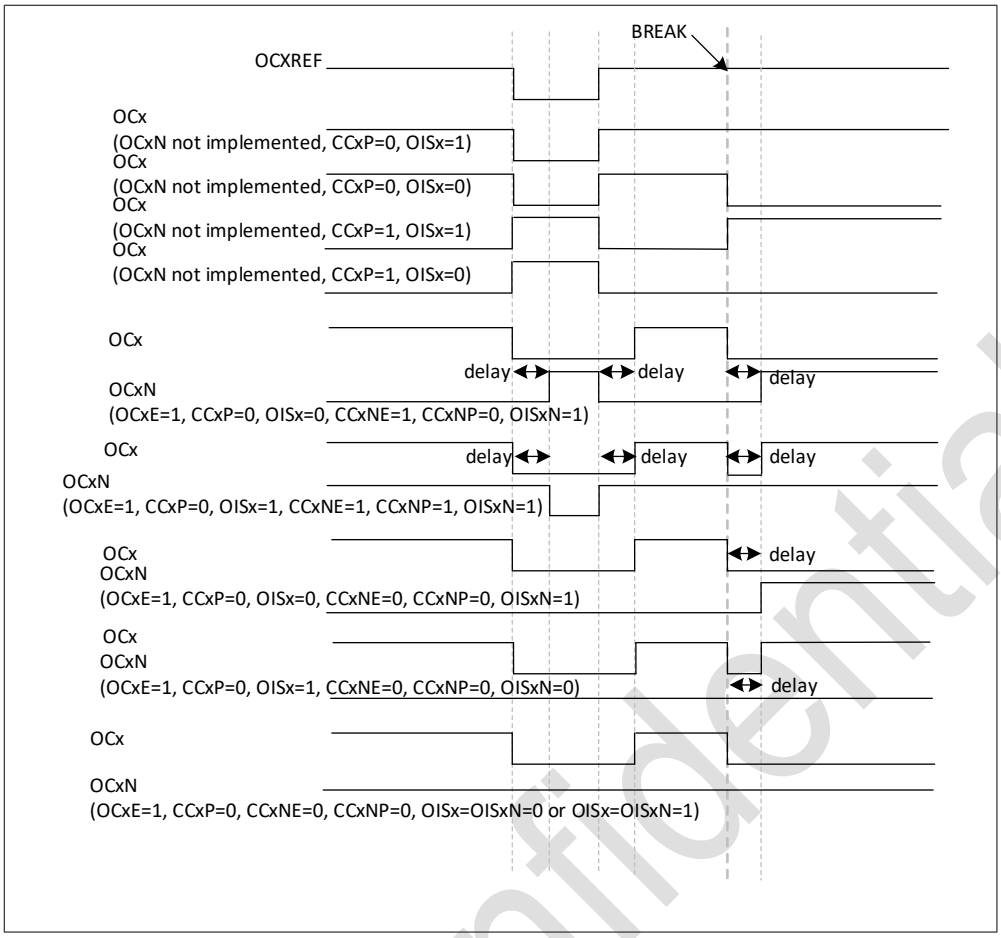


Figure 19-38 Output behavior in response to a break

19.3.13. Clearing the OCxREF signal on an external event

The OCxREF signal for a given channel can be driven Low by applying a High level to the ETRF input (OCxCE enable bit of the corresponding TIMx_CCMRx register set to ‘1’). The OCxREF signal remains Low until the next update event, UEV, occurs.

This function can only be used in output compare and PWM modes. It does not work in forced mode.

While OCREF_CLR_INPUT can be selected between OCREF_CLR and ETRF (after ETR filtering) by configuring the OCCS bit in the TIMx_SMCR register.

For example, the OCxREF signal can be connected to the output of a comparator to be used for current handling. In this case, ETR must be configured as follows:

1. The external trigger prescaler should be kept off:bits ETPS [1:0] in the TIMx_SMCR register are cleared to 00.

2. The external clock mode 2 must be disabled:bit ECE of the TIMx_SMCR register set to ‘0’.

3. The external trigger polarity (ETP) and the external trigger filter (ETF) can be configured according to the user needs.

The figure below shows the behavior of the OCxREF signal when the ETRF Input becomes High, for both values of the enable bit OCxCE. In this example, the timer TIMx is programmed in PWM mode.

Puya Confidential

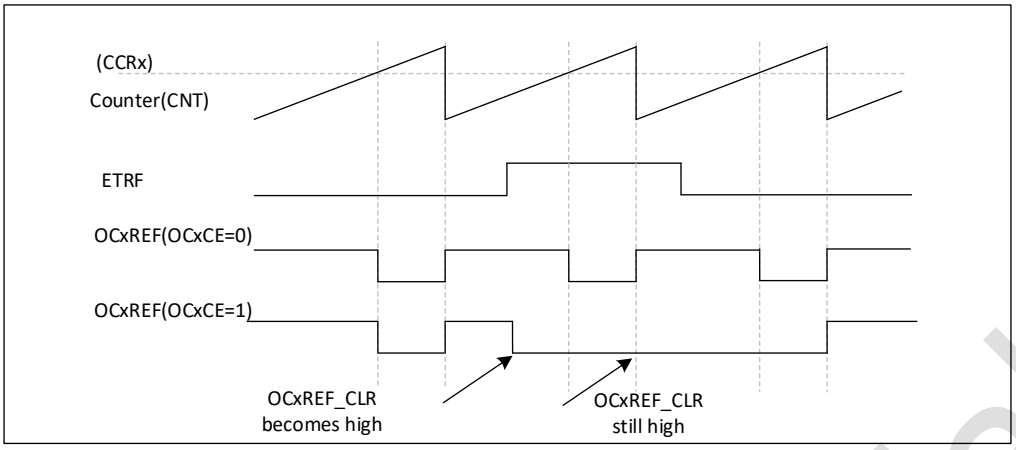


Figure 19-39 OCxREF for clearing TIM1

19.3.14. 6-step PWM generation

When complementary outputs are used on a channel, preload bits are available on the OCxM, CCxE and CCxNE bits. The preload bits are transferred to the shadow bits at the COM commutation event. Thus you can program in advance the configuration for the next step and change the configuration of all the channels at the same time. COM can be generated by software by setting the COM bit in the TIMx_EGR register or by hardware (on TRGI rising edge).

A flag is set when the COM event occurs (COMIF bit in the TIMx_SR register), which can generate an interrupt (if the COMIE bit is set in the TIMx_DIER register) or a DMA request (if the COMDE bit is set in the TIMx_DIER register).

The figure below describes the behavior of the OCx and OCxN outputs when a COM event occurs, in 3 different examples of programmed configurations.

Puya Confidential

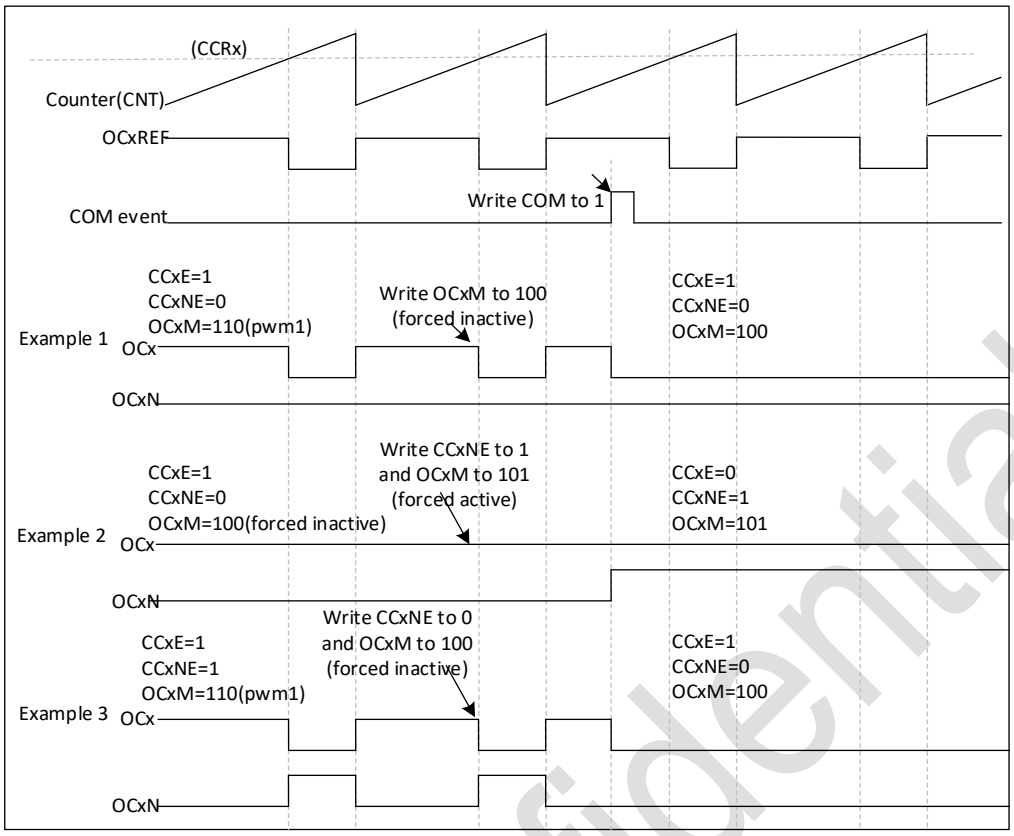


Figure 19-40 6-step generation, COM example (OSSR=1)

19.3.15. One-pulse mode

One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. One-pulse mode is selected by setting the OPM bit in the TIMx_CR1 register. This makes the counter stop automatically at the next update event UEV.

A pulse can be correctly generated only if the compare value is different from the counter initial value. Before starting (when the timer is waiting for the trigger), the configuration must be:

- In upcounting:CNT < CCRx ≤ ARR (in particular, 0 < CCRx)
- In downcounting:CNT > CCRx

Puya Confidential

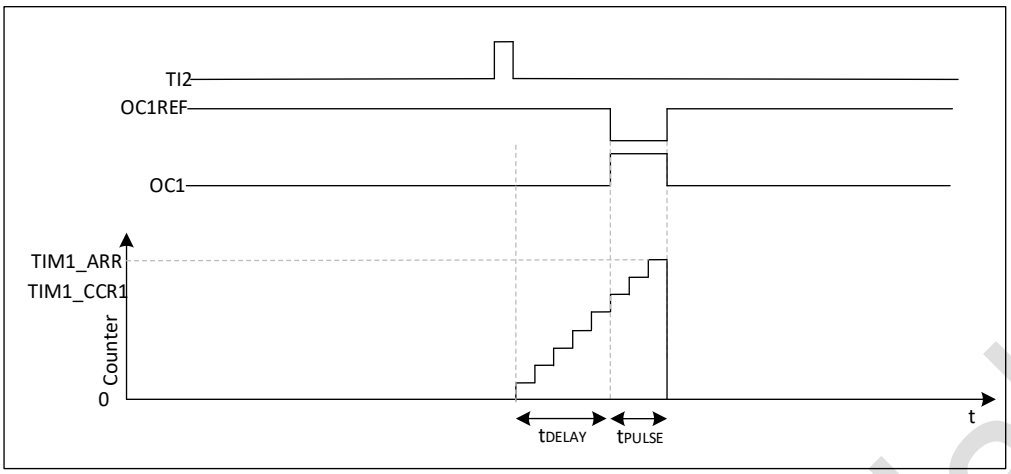


Figure 19-41 Example of one pulse mode

For example, one may want to generate a positive pulse on OC1 with a length of t_{PULSE} and after a delay of t_{DELAY} as soon as a positive edge is detected on the TI2 input pin.

Using TI2FP2 as trigger:

- Map TI2FP2 on TI2 by writing CC2S=01 in the TIMx_CCMR1 register.
- TI2FP2 must detect a rising edge, write CC2P=0 in the TIMx_CCER register.
- Configure TI2FP2 as trigger for the slave mode controller (TRGI) by writing TS=110 in the TIMx_SMCR register.
- TI2FP2 is used to start the counter by writing SMS to '110' in the TIMx_SMCR register (trigger mode).

The OPM waveform is defined by writing the compare registers (taking into account the clock frequency and the counter prescaler).

- The t_{DELAY} is defined by the value written in the TIMx_CCR1 register.
- The t_{PULSE} is defined by the difference between the auto-reload value and the compare value (TIMx_ARR - TIMx_CCR1).
- Let’s say one want to build a waveform with a transition from ‘0’ to ‘1’ when a compare match occurs and a transition from ‘1’ to ‘0’ when the counter reaches the auto-reload value. To do this PWM mode 2 must be enabled by writing OC1M=111 in the TIMx_CCMR1 register. Optionally the preload registers can be enabled by writing OC1PE='1' in the TIMx_CCMR1 register and ARPE in the TIMx_CR1 register. In this case one has to write the compare value in the TIMx_CCR1 register, the auto-reload value in the TIMx_ARR register, generate an update by setting the UG bit and wait for external trigger event on TI2. CC1P is written to ‘0’ in this example.

In our example, the DIR and CMS bits in the TIMx_CR1 register should be low.

The user only wants one pulse (Single mode), so '1’ must be written in the OPM bit in theTIMx_CR1 register to stop the counter at the next update event (when the counter rolls over from the auto-reload value back to 0).

Puya Confidential

Particular case:OCx fast enable:

In One-pulse mode, the edge detection on TIx input set the CEN bit which enables the counter. Then the comparison between the counter and the compare value makes the output toggle. But several clock cycles are needed for these operations, and it limits the minimum delay t_{DELAY} min we can get.

If one wants to output a waveform with the minimum delay, the OCxFE bit can be set in the TIMx_CCMRx register. Then OCxREF (and OCx) is forced in response to the stimulus, without taking in account the comparison. Its new level is the same as if a compare match had occurred. OCxFE acts only if the channel is configured in PWM1 or PWM2 mode.

19.3.16. Encoder interface mode

To select Encoder Interface mode:write SMS='001' in the TIMx_SMCR register if the counter is counting on TI2 edges only, SMS=010 if it is counting on TI1 edges only and SMS=011 if it is counting on both TI1 and TI2 edges.

Select the TI1 and TI2 polarity by programming the CC1P and CC2P bits in the TIMx_CCER register. When needed, you can program the input filter as well.

The two inputs TI1 and TI2 are used to interface to an incremental encoder. The counter is clocked by each valid transition on TI1FP1 or TI2FP2 (TI1 and TI2 after input filter and polarity selection, TI1FP1=TI1 if not filtered and not inverted, TI2FP2=TI2 if not filtered and not inverted) assuming that it is enabled (CEN bit inTIMx_CR1 register written to ‘1’). The sequence of transitions of the two inputs is evaluated and generates count pulses as well as the direction signal. Depending on the sequence the counter counts up or down, the DIR bit in the TIMx_CR1 register is modified by hardware accordingly. The DIR bit is calculated at each transition on any input (TI1 or TI2), whatever the counter is counting on TI1 only, TI2 only or both TI1 and TI2.

Encoder interface mode acts simply as an external clock with direction selection. This means that the counter just counts continuously between 0 and the auto-reload value in the TIMx_ARR register (0 to ARR or ARR down to 0 depending on the direction). So the TIMx_ARR must be configured before starting. In the same way, the capture, compare, prescaler, repetition counter, trigger output features continue to work as normal. Encoder mode and External clock mode 2 are not compatible and must not be selected together. In this mode, the counter is modified automatically following the speed and the direction of the quadrature encoder and its content, therefore, always represents the encoder’s position. The count direction corresponds to the rotation direction of the connected sensor. The table summarizes the possible combinations, assuming TI1 and TI2 do not switch at the same time.

Table 19-1 Counting direction versus encoder signals

Active edge	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No count	No count
	Low	Up	Down	No count	No count
Counting on TI2 only	High	No count	No count	Up	Down
	Low	No count	No count	Down	Up

Puya Confidential

Active edge	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

An external incremental encoder can be connected directly to the MCU without external interface logic. However, comparators are normally used to convert the encoder's differential outputs to digital signals. This greatly increases noise immunity. The third encoder output which indicate the mechanical zero position, may be connected to an external interrupt input and trigger a counter reset.

The figure below gives an example of counter operation, showing count signal generation and direction control. It also shows how input jitter is compensated where both edges are selected. This might occur if the sensor is positioned near to one of the switching points. For this example, we assume that the configuration is the following:

- CC1S='01' (TIMx_CCMR1 register, TI1FP1 mapped on TI1).
- CC2S='01' (TIMx_CCMR2 register, TI2FP2 mapped on TI2).
- CC1P = '0' (TIMx_CCER register, TI1FP1 is not inverted, TI1FP1 = TI1)
- CC2P = '0' (TIMx_CCER register, TI2FP2 is not inverted, TI2FP2 = TI2)
- SMS= 011 (TIMx_SMCR register, both inputs are active on both rising and falling edges)
- CEN='1' (TIMx_CR1 register, counter enabled)

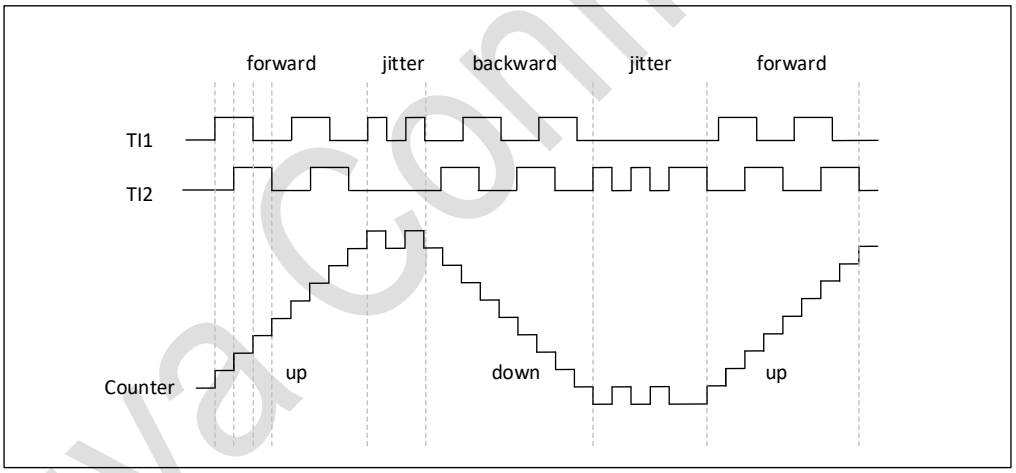


Figure 19-42 Example of counter operation in encoder interface mode

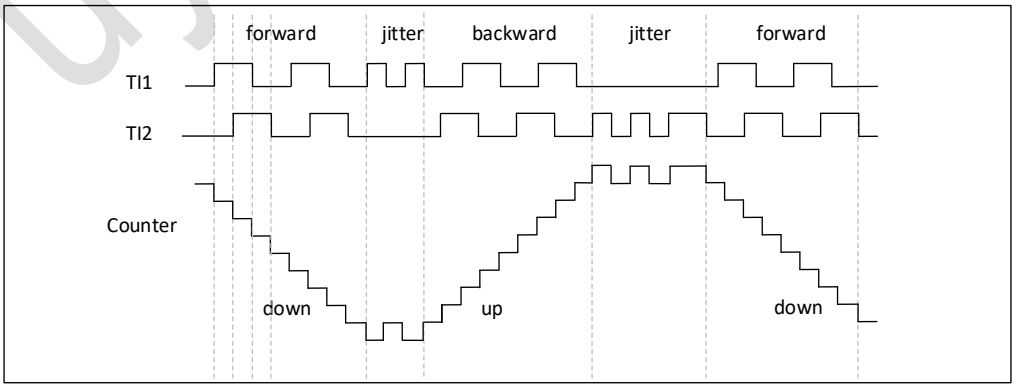


Figure 19-43 Example of encoder interface mode with TI1FP1 polarity inverted

The timer, when configured in Encoder Interface mode provides information on the sensor's current position. Dynamic information can be obtained (speed, acceleration, deceleration) by measuring the

Puya Confidential

period between two encoder events using a second timer configured in capture mode. The output of the encoder which indicates the mechanical zero can be used for this purpose. Depending on the time between two events, the counter can also be read at regular times. This can be done by latching the counter value into a third input capture register if available (then the capture signal must be periodic and can be generated by another timer). When available, it is also possible to read its value through a DMA request generated by a real-time clock.

19.3.17. Timer input XOR function

The TI1S bit in the TIM_CR2 register, allows the input filter of channel 1 to be connected to the output of a XOR gate, combining the three input pins TIMx_CH1, TIMx_CH2 and TIMx_CH3.

The XOR output can be used with all the timer input functions such as trigger or input capture.

19.3.18. Interfacing with Hall sensors

This is done using the advanced-control timers (TIM1) to generate PWM signals to drive the motor and another timer TIM3 referred to as interfacing timer. The interfacing timer captures the 3 timer input pins (CC1, CC2, CC3) connected through an XOR to the TI1 input channel (selected by setting the TI1S bit in the TIMx_CR2 register).

The slave mode controller is configured in reset mode, the slave input is TI1F_ED. Thus, each time one of the 3 inputs toggles, the counter restarts counting from 0. This creates a time base triggered by any change on the Hall inputs.

On the interfacing timer, capture/compare channel 1 is configured in capture mode, capture signal is TRC. The captured value, which corresponds to the time elapsed between 2 changes on the inputs, gives information about motor speed.

The interfacing timer can be used in output mode to generate a pulse which changes the configuration of the channels of the advanced-control timer (TIM1) (by triggering a COM event). The TIM1 timer is used to generate PWM signals to drive the motor. To do this, the interfacing timer channel must be programmed so that a positive pulse is generated after a programmed delay (in output compare or PWM mode). This pulse is sent to the advanced-control timer (TIM1) through the TRGO output.

Example:one wants to change the PWM configuration of the advanced-control timer after a programmed delay each time a change occurs on the Hall inputs connected to one of the TIMx timers.

- Configure 3 timer inputs ORed to the TI1 input channel by writing the TI1S bit in the TIMx_CR2 register to ‘1’.
- Program the time base:write the TIMx_ARR to the max value (the counter must be cleared by the TI1 change. Set the prescaler to get a maximum counter period longer than the time between 2 changes on the sensors.
- Program the channel 1 in capture mode (TRC selected):write the CC1S bits in the TIMx_CCMR1 register to ‘01’. The digital filter can also be programmed if needed.
- Program channel 2 in PWM 2 mode with the desired delay:write the OC2M bits to ‘111’ and the CC2S bits to ‘00’ in the TIMx_CCMR1 register.
- Select OC2REF as trigger output on TRGO:write the MMS bits in the TIMx_CR2 register to ‘101’.

Puya Confidential

In the advanced-control timer TIM1, the right ITR input must be selected as trigger input, the timer is programmed to generate PWM signals, the capture/compare control signals are preloaded (CCPC=1 in the TIMx_CR2 register) and the COM event is controlled by the trigger input (CCUS=1 in the TIMx_CR2 register). The PWM control bits (CCxE, OCxM) are written after a COM event for the next step, which can be done in an interrupt subroutine generated by the rising edge of OC2REF.

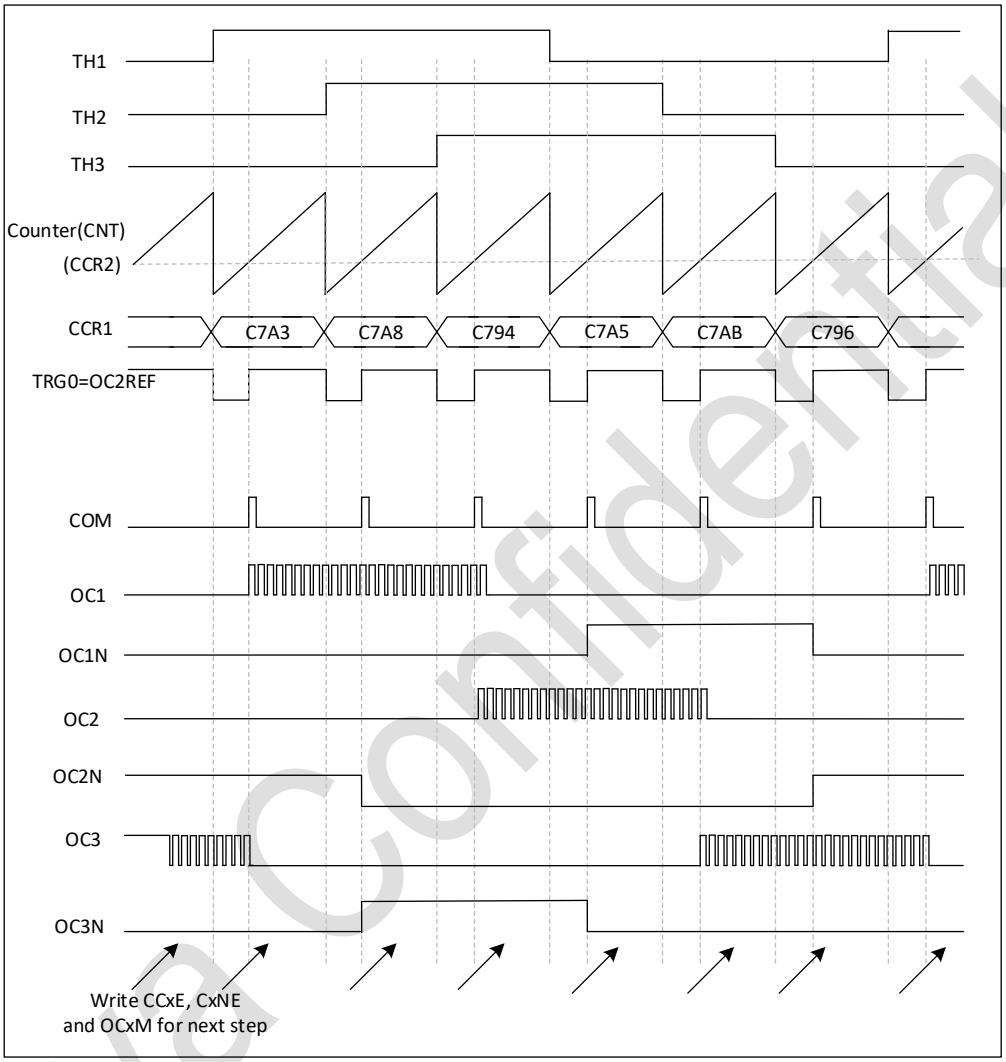


Figure 19-44 Example of Hall sensor interface

19.3.19. TIMx and external trigger synchronization

The TIMx timer can be synchronized with an external trigger in several modes:Reset mode, Gated mode and Trigger mode.

Slave mode:Reset mode

The counter and its prescaler can be reinitialized in response to an event on a trigger input. Moreover, if the URS bit from the TIMx_CR1 register is low, an update event UEV is generated. Then all the preloaded registers (TIMx_ARR, TIMx_CCRx) are updated.

In the following example, the upcounter is cleared in response to a rising edge on TI1 input:

- Configure the channel 1 to detect rising edges on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used

Puya Confidential

for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=0 in TIMx_CCER register to validate the polarity (and detect rising edges only).

- Configure the timer in reset mode by writing SMS=100 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.
- Enable the counter by writing CEN=1 in the TIMx_CR1 register.

The counter starts counting on the internal clock, then behaves normally until TI1 rising edge. When TI1 rises, the counter is cleared and restarts from 0. In the meantime, the trigger flag is set (TIF bit in the TIMx_SR register) and an interrupt request, or a DMA request can be sent if enabled (depending on the TIE and TDE bits in TIMx_DIER register).

The following figure shows this behavior when the auto-reload register TIMx_ARR=0x36. The delay between the rising edge on TI1 and the actual reset of the counter is due to the resynchronization circuit on TI1 input.

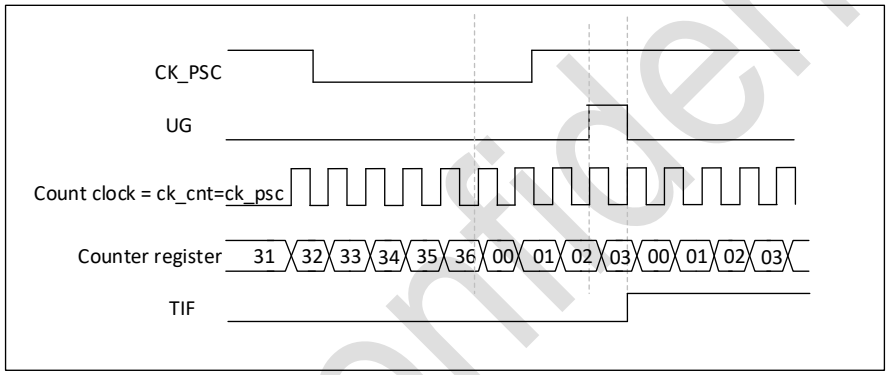


Figure 19-45 Control circuit in reset mode

Slave mode:Gated mode

The counter can be enabled depending on the level of a selected input.

In the following example, the upcounter counts only when TI1 input is low:

- Configure the channel 1 to detect low levels on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=1 in TIMx_CCER register to validate the polarity (and detect low level only).
- Configure the timer in gated mode by writing SMS=101 in TIMx_SMCR register. SelectTI1 as the input source by writing TS=101 in TIMx_SMCR register.
- Enable the counter by writing CEN=1 in the TIMx_CR1 register. In gated mode, the counter doesn't start if CEN=0, whatever is the trigger input level.

The counter starts counting on the internal clock as long as TI1 is low and stops as soon as TI1 becomes high. The TIF flag in the TIMx_SR register is set both when the counter starts or stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

Puya Confidential

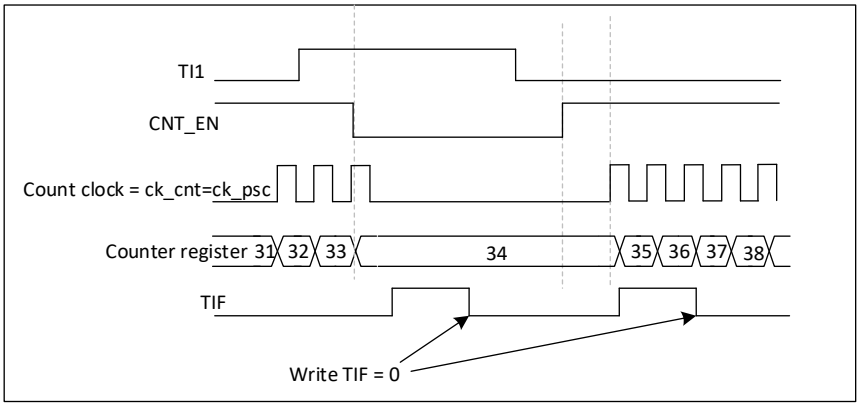


Figure 19-46 Control circuit in Gated mode

The selected event on the input enables the counter.

In the following example, the upcounter starts in response to a rising edge on TI2 input:

1. Configure the channel 2 to detect rising edges on TI2. Configure the input filter duration (in this example, we do not need any filter, so we keep IC2F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC2S bits are configured to select the input capture source only, CC2S=01 in TIMx_CCMR1 register. Write CC2P=1 in TIMx_CCER register to validate the polarity (and detect low level only).
2. Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI2 as the input source by writing TS=110 in TIMx_SMCR register.

When a rising edge occurs on TI2, the counter starts counting on the internal clock and the TIF flag is set. The delay between the rising edge on TI2 and the actual start of the counter is due to the resynchronization circuit on TI2 input.

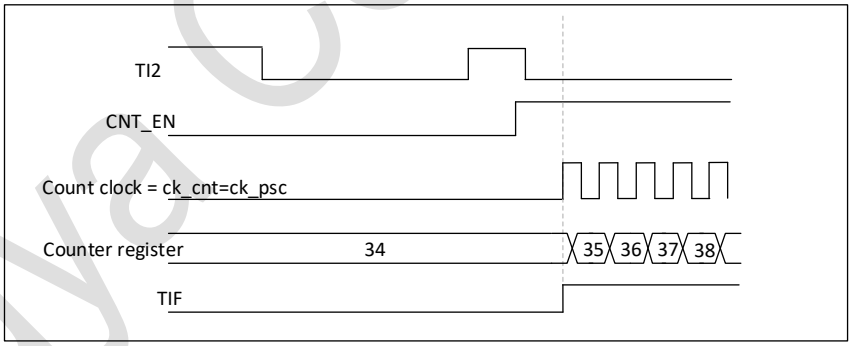


Figure 19-47 Control circuit in Gated mode

Slave mode:external clock mode 2 + trigger mode

The external clock mode 2 can be used in addition to another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as external clock input, and another input can be selected as trigger input when operating in reset mode, gated mode or trigger mode. It is recommended not to select ETR as TRGI through the TS bits of TIMx_SMCR register.

In the following example, the upcounter is incremented at each rising edge of the ETR signal as soon as a rising edge of TI1 occurs:

- Configure the external trigger input circuit by programming the TIMx_SMCR register as follows:

Puya Confidential

— ETF = 0000:no filter

— ETPS = 00:prescaler disabled

— ETP = 0:detection of rising edges on ETR and ECE=1 to enable the external clock mode 2.

- Configure the channel 1 as follows, to detect rising edges on TI:

  — IC1F = 0000:no filter

  — The capture prescaler is not used for triggering, so it does not need to be configured.

  — CC1S=01 in TIMx_CCMR1 register to select only the input capture source

Write CC1P=0 in TIMx_CCER register to validate the polarity (and detect rising edges only).

- Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.

A rising edge on TI1 enables the counter and sets the TIF flag. The counter then counts on ETR rising edges. The delay between the rising edge of the ETR signal and the actual reset of the counter is due to the resynchronization circuit on ETRP input.

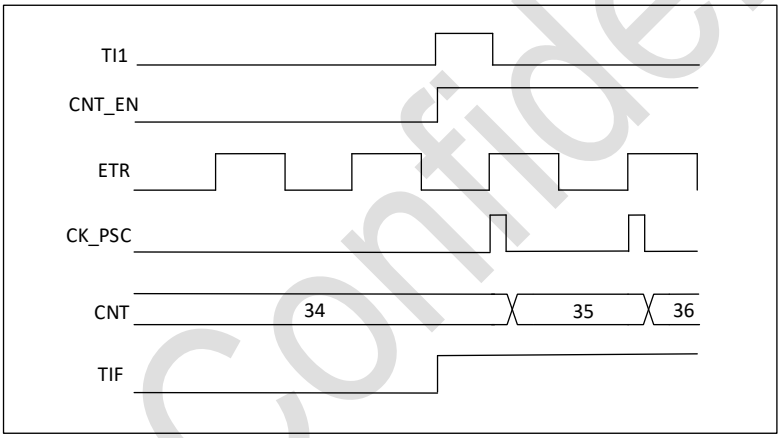


Figure 19-48 Control circuit in external clock mode 2 + trigger mode

19.3.20. Timer synchronization

The TIM timers are linked together internally for timer synchronization or chaining. When one timer is in master mode, it can reset, start, stop, etc. the counter of another timer in slave mode. Refer to Section 22.3.14, Timer synchronization for details.

19.3.21. Debug mode

When the microcontroller enters debug mode, the TIMx counter either continues to work normally or stops, depending on DBG_TIMx_STOP configuration bit in DBG module.

Puya Confidential

19.4. TIM1 registers

19.4.1. TIM1 control register 1 (TIM1_CR1)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
-	-	-	-	-	-	RW		RW	RW		RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9:8	CKD [1:0]	RW	00	Clock division This bit-field indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock (t_{DTS}) used by the dead-time generators and the digital filters (ETR, TIx) 00:$t_{DTS} = t_{CK\_INT}$ 01:$t_{DTS} = 2 \times t_{CK\_INT}$ 10:$t_{DTS} = 4 \times t_{CK\_INT}$ 11:Reserved, do not program this value
7	ARPE	RW	0	Auto-reload preload enable 0:TIMx_ARR register is not buffered 1:TIMx_ARR register is buffered
6:5	CMS [1:0]	RW	00	Center-aligned mode selection 00:Edge-aligned mode. The counter counts up or down depending on the direction bit (DIR). 01:Center-aligned mode 1. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set only when the counter is counting down. 10:Center-aligned mode 2. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set only when the counter is counting up.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:Center-aligned mode 3. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set both when the counter is counting up or down. Note:It is not allowed to switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN=1).
4	DIR	RW	0	Counting direction 0:Counter used as upcounter 1:Counter used as downcounter Note:This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.
3	OPM	RW	0	One-pulse mode 0:Counter is not stopped at update event 1:Counter stops counting at the next update event (clearing the bit CEN)
2	URS	RW	0	Update request source This bit is set and cleared by software to select the UEV event sources. 0:Any of the following events generate an update interrupt or DMA request if enabled. These events can be: – Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller 1:Only counter overflow/underflow generates an update interrupt or DMA request if enabled
1	UDIS	RW	0	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0:UEV enabled. The Update (UEV) event is generated by one of the following events: – Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1:UEV disabled. The update event is not generated, shadow registers keep their value (ARR, PSC, CCRx).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				However, the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	RW	0	Counter enabled 0:Counter disabled 1:Counter enabled Note:external clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However, trigger mode can set the CEN bit automatically by hardware.

19.4.2. TIM1 control register 2 (TIM1_CR2)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res	CCPC
-	RW	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	-	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	OIS4	RW	0	Output idle state 4 (OC4 output) Refer to OIS1 bit.
13	OIS3N	RW	0	Output idle state 3 (OC3N output) Refer to OIS1N bit
12	OIS3	RW	0	Output idle state 3 (OC3 output) Refer to OIS1 bit.
11	OIS2N	RW	0	Output idle state 2 (OC2N output) Refer to OIS1N bit
10	OIS2	RW	0	Output idle state 2 (OC2 output) Refer to OIS1 bit
9	OIS1N	RW	0	Output idle state 1 (OC1N output) 0:OC1N=0 after a dead-time when MOE=0 1:OC1N=1 after a dead-time when MOE=0 Note:this bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BKR register).
8	OIS1	RW	0	Output idle state 1 (OC1 output) 0:OC1=0 (after a dead-time if OC1N is implemented) when MOE=0

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:OC1=1 (after a dead-time if OC1N is implemented) when MOE=0 Note:this bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BKR register).
7	TI1S	RW	0	TI1 selection 0:The TIMx_CH1 pin is connected to TI1 input 1:The TIMx_CH1, CH2 and CH3 pins are connected to the TI1 input.
6:4	MMS [2:0]	RW	000	Master mode selection These two bits are used to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000:Reset - the UG bit from the TIMx_EGR register is used as trigger output (TRGO). If the reset is generated by the trigger input (slave mode controller configured in reset mode) then the signal on TRGO is delayed compared to the actual reset. 001:Enable - the counter enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enabled. The counter enable signal is generated by a logic AND between CEN control bit and the trigger input when configured in gated mode. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected (see the MSM bit description in TIMx_SMCR register). 010:Update - The update event is selected as trigger output (TRGO). For instance, a master timer can then be used as a prescaler for a slave timer. 011:Compare pulse - The trigger output sends a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or a compare match occurred (TRGO). 100:Compare - OC1REF signal is used as trigger output (TRGO) 101:Compare - OC2REF signal is used as trigger output (TRGO)

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				110:Compare - OC3REF signal is used as trigger output (TRGO) 111:Compare - OC4REF signal is used as trigger output (TRGO) Note: 1. The clock of the slave timer or ADC must be enabled prior to receive events from the master timer, and must not be changed. 2. If the master and slave timers are not on the same bus, the master mode should be configured to the width that can be picked by the slave timer.
3	CCDS	RW	0	Capture/compare DMA selection 0:CCx DMA request sent when CCx event occurs 1:CCx DMA requests sent when update event occurs
2	CCUS	RW	0	Capture/compare control update selection 0:When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COM bit only. 1:When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COM bit or when a rising edge occurs on TRGI. Note:This bit acts only on channels that have a complementary output.
1	Res	-	0	Reserved, must be kept at reset value.
0	CCPC	RW	0	Capture/compare preloaded control 0:CCxE, CCxNE and OCxM bits are not preloaded. 1:CCxE, CCxNE, and OCxM bits are preloaded, Once this bit is set, they are only set when COM Note:This bit acts only on channels that have a complementary output.

19.4.3. TIM1 slave mode control register (TIM1_SMCR)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETP	ECE	ETPS [1:0]		ETF [3:0]				MSM	TS [2:0]			OCCS	SMS [2:0]		
RW	RW	RW		RW				RW	RW			RW	RW		

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	ETP	RW	0	External trigger polarity. This bit selects whether ETR or inverted ETR is used for trigger operations. 0:ETR is non-inverted, active at high level or rising edge. 1:ETR is inverted, active at low level or falling edge.
14	ECE	RW	0	External clock enable bit. This bit enables external clock mode 2 0:External clock mode 2 disabled 1:External clock mode 2 enabled. The counter is driven by any active edge on the ETRF signal. Note:1:Setting the ECE bit has the same effect as selecting external clock mode 1 with TRGI connected to ETRF (SMS=111 and TS=111). Note 2:It is possible to simultaneously use external clock mode 2 with the following slave modes:reset mode, gated mode and trigger mode. Nevertheless, TRGI must not be connected to ETRF in this case (TS bits must not be 111). Note 3:If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.
13:12	ETPS [1:0]	RW	00	External trigger prescaler. External trigger signal ETRP frequency must be at most 1/4 of TIM1 CLK frequency. A prescaler can be enabled to reduce ETRP frequency. 00:Prescaler OFF 01:ETRP frequency divided by 2 10:ETRP frequency divided by 4 11:ETRP frequency divided by 8
11:8	ETF [3:0]	RW	0000	External trigger filter. This bit-field then defines the frequency used to sample ETRP signal and the length of the digital filter applied to ETRP. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output. 0000:No filter, sampling is done at f_{DTS} 0001:$f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010:$f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011:$f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100:$f_{SAMPLING} = f_{DTS}/2$, N = 6 0101:$f_{SAMPLING} = f_{DTS}/2$, N = 8 0110:$f_{SAMPLING} = f_{DTS}/4$, N = 6 0111:$f_{SAMPLING} = f_{DTS}/4$, N = 8

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1000:$f_{SAMPLING}$ = f_{DTS}/8, N = 6 1001:$f_{SAMPLING}$ = f_{DTS}/8, N = 8 1010:$f_{SAMPLING}$ = f_{DTS}/16, N = 5 1011:$f_{SAMPLING}$ = f_{DTS}/16, N = 6 1100:$f_{SAMPLING}$ = f_{DTS}/16, N = 8 1101:$f_{SAMPLING}$ = f_{DTS}/32, N = 5 1110:$f_{SAMPLING}$ = f_{DTS}/32, N = 6 1111:$f_{SAMPLING}$ = f_{DTS}/32, N = 8
7	MSM	RW	0	Master/slave mode 0:No action 1:The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.
6:4	TS [2:0]	RW	000	Trigger selection. This bit-field selects the trigger input to be used to synchronize the counter. 000:TIM15 (ITR0) 001:TIM2 (ITR1) 010:TIM3 (ITR2) 011:TIM17 (ITR3) 100:edge detector of TI1 (TI1F_ED) 101:Filtered timer input 1 (TI1FP1) 110:Filtered timer input 2 (TI2FP2) 111 external trigger input (ETRF) Note:These bits must be changed only when they are not used (e.g. when SMS=000) to avoid wrong edge detections at the transition.
3	OCCS	RW	0	OCREF clear selection This bit is used to select the OCREF clear source. 0:OCREF_CLR_INT clear source is connected to OCREF_CLR input 1:OCREF_CLR_INT is connected to ETRF
2:0	SMS [2:0]	RW	000	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input (see Input Control register and Control Register description). 000:Slave mode disabled If CEN = ‘1’ then the prescaler is clocked directly by the internal clock.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				001:Encoder mode 1 Counter counts up/down on TI1FP1 edge depending on TI2FP2 level. 010:Encoder mode 2 Counter counts up/down on TI2FP2 edge depending on TI1FP1 level. 011:Encoder mode 3 Counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of the other input. 100:Reset mode Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101:Gated mode The counter clock is enabled when the trigger input (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop of the counter are controlled. 110:Trigger mode The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only the start of the counter is controlled. 111:External clock mode 1 Rising edges of the selected trigger (TRGI) clock the counter. Note:The gated mode must not be used if TI1F_ED is selected as the trigger input (TS=100). Indeed, TI1F_ED outputs 1 pulse for each transition on TI1F, whereas the gated mode checks the level of the trigger signal. Note:Do not use UEV as TRGO output signal in encoder mode (i.e. MMS cannot be configured to 010)

Table 19-2 TIM1 internal trigger connection

Slave TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM15_TRGO	TIM2_TRGO	TIM3_TRGO	TIM17_OC

19.4.4. TIM1 DMA/interrupt enable register (TIM1_DIER)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Puya Confidential

Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
-	RW														

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	TDE	RW	0	TDE:Trigger DMA request enable 0:Trigger DMA request disabled 1:Trigger DMA request enabled
13	COMDE	RW	0	COMDE:COM DMA request enable 0:COM DMA request disabled 1:COM DMA request enabled
12	CC4DE	RW	0	CC4DE:Capture/Compare 4 DMA request enable 0:CC4 DMA request disabled 1:CC4 DMA request enabled
11	CC3DE	RW	0	CC3DE:Capture/Compare 3 DMA request enable 0:CC3 DMA request disabled 1:CC3 DMA request enabled
10	CC2DE	RW	0	CC2DE:Capture/Compare 2 DMA request enable 0:CC2 DMA request disabled 1:CC2 DMA request enabled
9	CC1DE	RW	0	CC1DE:Capture/Compare 1 DMA request enable 0:CC1 DMA request disabled 1:CC1 DMA request enabled
8	UDE	RW	0	UDE:Update DMA request enable 0:Update DMA request disabled 1:Update DMA request enabled
7	BIE	RW	0	BIE:Break interrupt enable 0:Break interrupt disabled 1:Break interrupt enabled
6	TIE	RW	0	TIE:Trigger interrupt enable 0:Trigger interrupt disabled. 1:Trigger interrupt enabled
5	COMIE	RW	0	COMIE:COM interrupt enable 0:COM interrupt disabled 1:COM interrupt enabled
4	CC4IE	RW	0	CC4IE:Capture/Compare 4 interrupt enable 0:Capture/Compare 4 interrupt disabled 1:Capture/Compare 4 interrupt enabled
3	CC3IE	RW	0	CC3IE:Capture/Compare 3 interrupt enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Capture/Compare 3 interrupt disabled 1:Capture/Compare 3 interrupt enabled
2	CC2IE	RW	0	CC2IE:Capture/Compare 2 interrupt enable 0:Capture/Compare 2 interrupt disabled 1:Capture/Compare 2 interrupt enabled
1	CC1IE	RW	0	CC1IE:Capture/Compare 1 interrupt enable 0:Capture/Compare 1 interrupt disabled 1:Capture/Compare 1 interrupt enabled
0	UIE	RW	0	UIE:Update interrupt enable 0:Update interrupt disabled. 1:Update interrupt enabled

19.4.5. TIM1 status register (TIM1_SR)

Address offset:0x010

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res			Res					IC4IF	IC3IF	IC2IF	IC1IF	IC4IR	IC3IR	IC2IR	IC1IR
-								RC_W0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res			CC4OF	CC3OF	CC2OF	CC1OF	Res	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
-			RC_W0				-	RC_W0							

Bit	Name	R/W	Reset Value	Function
31:13	Reserved	-	-	Reserved
23	IC4IF	RC_W0	0	Falling edge capture 4 flag Refer to IC1IF description
22	IC3IF	RC_W0	0	Falling edge capture 3 flag Refer to IC1IF description
21	IC2IF	RC_W0	0	Falling edge capture 2 flag Refer to IC1IF description
20	IC1IF	RC_W0	0	Falling edge capture 1 flag This flag is set by hardware only when the corresponding channel is configured in input capture mode and triggered by falling edge. It is cleared by software or reading TIMx_CCR1. 0:No falling edge capture occurs

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:A falling edge capture event occurs.
19	IC4IR	RC_W0	0	Rising edge capture 4 flag Refer to IC1IR description
18	IC3IR	RC_W0	0	Rising edge capture 3 flag Refer to IC1IR description
17	IC2IR	RC_W0	0	Rising edge capture 2 flag Refer to IC1IR description
16	IC1IR	RC_W0	0	Rising edge capture 1 flag This flag is set by hardware only when the corresponding channel is configured in input capture mode and triggered by rising edge. It is cleared by software or reading TIMx_CCR1. 0:No rising edge capture occurs; 1:A rising edge capture event occurs.
15:13	Reserved	-	-	Reserved
12	CC4OF	RC_W0	0	Capture/Compare 4 overcapture flag Refer to CC1OF description
11	CC3OF	RC_W0	0	Capture/Compare 3 overcapture flag Refer to CC1OF description
10	CC2OF	RC_W0	0	Capture/Compare 2 overcapture flag Refer to CC1OF description
9	CC1OF	RC_W0	0	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to ‘0’. 0:No overcapture has been detected. 1:The counter value has been captured in TIMx_CCR1 register while CC1IF flag was already set
8	Reserved	-	-	Reserved
7	BIF	RC_W0	0	Break interrupt flag This flag is set by hardware as soon as the break input goes active. It can be cleared by software if the break input is not active. 0:No break event occurred. 1:An active level has been detected on the break input.
6	TIF	RC_W0	0	Trigger interrupt flag This flag is set by hardware on trigger event (active edge detected on TRGI input when the slave mode controller is enabled in all modes but gated mode). It is set when the

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				counter starts or stops when gated mode is selected. It is cleared by software. 0:No trigger event occurred. 1:Trigger interrupt pending.
5	COMIF	RC_W0	0	COM interrupt flag This flag is set by hardware on COM event (when Capture/compare Control bits - CCxE, CCxNE, OCxM - have been updated). It is cleared by software. 0:No update occurred. 1:COM interrupt pending.
4	CC4IF	RC_W0	0	Capture/Compare 4 interrupt flag Refer to CC1IF description
3	CC3IF	RC_W0	0	Capture/Compare 3 interrupt flag Refer to CC1IF description
2	CC2IF	RC_W0	0	Capture/Compare 2 interrupt flag Refer to CC1IF description
1	CC1IF	RC_W0	0	Capture/Compare 1 interrupt flag If channel CC1 is configured as output: This flag is set by hardware when the counter matches the compare value. However, CMS needs to be judged in centrosymmetric mode; When the comparison value is greater than the overload value, it is set to 1 in overflow (up count mode, center alignment count mode) or underflow (down count mode). It is cleared by software. 0:No match; 1:The content of the counter TIMx_CNT matches the content of the TIMx_CCR1 register. If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading the TIMx_CCR1 register. 0:No input capture occurred 1:The counter value has been captured in TIMx_CCR1 register (An edge has been detected on IC1 which matches the selected polarity) Note:When CEN is on, this bit will also be set.
0	UIF	RC_W0	0	Update interrupt flag This bit is set by hardware on an update event. It is cleared by software.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:No update occurred. 1:Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow or underflow regarding the repetition counter value (update if REP_CNT = 0) and if the UDIS=0 in the TIMx_CR1 register. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0 in the TIMx_CR1 register. − If UDIS = 0 and URS = 0 of the TIMx_CR1 register, an update event occurs when the CNT is reinitialized by the trigger event (Refer to TIM1 slave mode control register (TIM1_SMCR))

19.4.6. TIM1 event generation register (TIM1_EGR)

Address offset:0x14

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
-	-	-	-	-	-	-	-	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7	BG	W	0	Break generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A break event is generated. MOE bit is cleared and BIF flag is set. Related interrupt or DMA transfer can occur if enabled.
6	TG	W	0	Trigger generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:The TIF flag is set in TIMx_SR register. Related interrupt or DMA transfer can occur if enabled.
5	COMG	W	0	Capture/Compare control update generation This bit can be set by software, it is automatically cleared by hardware. 0:No action 1:When CCPC bit is set, it allows to update CCxE, CCxNE and OCxM bits Note:This bit acts only on channels that have a complementary output.
4	CC4G	W	0	Capture/Compare 4 generation Refer to CC1G description
3	CC3G	W	0	Capture/Compare 3 generation Refer to CC1G description
2	CC2G	W	0	Capture/Compare 2 generation Refer to CC1G description
1	CC1G	W	0	Capture/Compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A capture/compare event is generated on channel CC1. If channel CC1 is configured as output: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input: The current value of the counter is captured in TIMx_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if theCC1IF flag was already set.
0	UG	W	0	Update generation This bit can be set by software; it is automatically cleared by hardware. 0:No action 1:Reinitialize the counter and generate an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected). The counter is cleared if the center-aligned mode is selected or if DIR=0 (upcounting), else it takes the auto-reload value (TIMx_ARR) if DIR=1 (downcounting).

Puya Confidential

19.4.7. TIM1 capture/compare mode register 1 (TIM1_CCMR1)

Address offset:0x18

Reset value:0x0000 0000

Output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M [2:0]			OC2PE	CO2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	OC2CE	RW	0	Output Compare 2 clear enable
14:12	OC2M [2:0]	RW	000	Output Compare 2 mode
11	OC2PE	RW	0	Output Compare 2 preload enable
10	OC2FE	RW	0	Output Compare 2 fast enable
9:8	CC2S [1:0]	RW	00	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC2 channel is configured as output; 01:CC2 channel is configured as input, IC2 is mapped on TI2; 10:CC2 channel is configured as input, IC2 is mapped on TI1; 11:CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC2S bits are writable only when the channel is OFF (CC2E = '0' in TIMx_CCER).
7	OC1CE	RW	0	Output Compare 1 clear enable 0:OC1REF is not affected by the ETRF signal; 1:OC1REF is cleared as soon as a high level is detected on ETRF signal.
6:4	OC1M [2:0]	RW	00	Output compare 1 mode These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000:Frozen. The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on the outputs. 001:Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 010:Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 011:Toggle OC1REF toggles when TIMx_CNT=TIMx_CCR1. 100:Force inactive level OC1REF is forced low. 101:Force active level OC1REF is forced high. 110:PWM mode 1 - In upcounting, channel 1 is active as long as TIMx_CNT<TIMx_CCR1 else inactive. In downcounting, channel 1 is inactive (OC1REF= ‘0’) as long as TIMx_CNT>TIMx_CCR1 else active (OC1REF=’1’). 111:PWM mode 2 - In upcounting, channel 1 is inactive as long as TIMx_CNT<TIMx_CCR1 else active. In downcounting, channel 1 is active as long as TIMx_CNT>TIMx_CCR1 else inactive. Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output). Note:In PWM mode 1 or 2, the OCREF level changes when the result of the comparison changes or when the output compare mode switches from frozen to PWM mode.
3	OC1PE	RW	0	Output Compare 1 preload enable 0:Preload register on TIMx_CCR1 disabled. TIMx_CCR1 can be written at any time, the new value is taken in account immediately. 1:Preload register on TIMx_CCR1 enabled. Read/Write operations access the preload register. TIMx_CCR1 preload value is loaded in the active register at each update event.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output).
2	OC1FE	RW	0	Output compare 1 fast enable This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0:CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1:An active edge on the trigger input acts like a compare match on OC1 output. Then, OC1 is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OCFE acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S [1:0]	RW	00	Capture/Compare 1 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:CC1 channel is configured as input, IC1 is mapped on TI2; 11:CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC1S bits are writable only when the channel is OFF (CC1E = ‘0’ in TIMx_CCER).

Input capture mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F [3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:12	IC2F	RW	0000	Input capture 2 filter
11:10	IC2PSC [1:0]	RW	00	Input capture 2 prescaler
9:8	CC2S [1:0]	RW	0	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC2 channel is configured as output; 01:CC2 channel is configured as input, IC2 is mapped on TI2; 10:CC2 channel is configured as input, IC2 is mapped on TI1; 11:CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC2S bits are writable only when the channel is OFF (CC2E = '0' in TIMx_CCER).
7:4	IC1F [3:0]	RW	0000	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output: 0000:No filter, sampling is done at f_{DTS} 0001:$f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010:$f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011:$f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100:$f_{SAMPLING} = f_{DTS}/2$, N = 6 0101:$f_{SAMPLING} = f_{DTS}/2$, N = 8 0110:$f_{SAMPLING} = f_{DTS}/4$, N = 6 0111:$f_{SAMPLING} = f_{DTS}/4$, N = 8 1000:$f_{SAMPLING} = f_{DTS}/8$, N = 6 1001:$f_{SAMPLING} = f_{DTS}/8$, N = 8 1010:$f_{SAMPLING} = f_{DTS}/16$, N = 5 1011:$f_{SAMPLING} = f_{DTS}/16$, N = 6 1100:$f_{SAMPLING} = f_{DTS}/16$, N = 8 1101:$f_{SAMPLING} = f_{DTS}/32$, N = 5 1110:$f_{SAMPLING} = f_{DTS}/32$, N = 6 1111:$f_{SAMPLING} = f_{DTS}/32$, N = 8
3:2	IC1PSC [1:0]	RW	00	Input capture 1 prescaler 00:no prescaler, capture is done each time an edge is detected on the capture input;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				01:capture is done once every 2 events 10:capture is done once every 4 events 11:capture is done once every 8 events
1:0	CC1S [1:0]	RW	00	CC1S [1:0]:capture/compare 1 selection. This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output 01:CC1 channel is configured as input, IC1 is mapped on TI1 10:CC1 channel is configured as input, IC1 is mapped on TI2 11:CC1 channel is configured as input, IC1 is mapped on TRC.This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC1S bits are writable only when the channel is OFF (CC1E = '0' in TIMx_CCER).

19.4.8. TIM1 capture/compare mode register 2 (TIM1_CCMR2)

Address offset:0x1C

Reset value:0x0000 0000

Output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OC4CE	OC4M [2:0]			OC4PE	CO4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
IC4F [3:0]				IC4PSC [1:0]				IC3F [3:0]				IC3PSC [1:0]			
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	OC4CE	RW	0	Output Compare 4 clear enable
14:12	OC4M [2:0]	RW	000	Output compare 4 mode
11	OC4PE	RW	0	Output Compare 4 preload enable
10	OC4FE	RW	0	Output Compare 4 fast enable
9:8	CC4S [1:0]	RW	00	Capture/Compare 4 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC4 channel is configured as output 01:CC4 channel is configured as input, IC4 is mapped on TI4

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				10:CC4 channel is configured as input, IC4 is mapped on TI3 11:CC4 channel is configured as input, IC4 is mapped on TRC.This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register). Note:CC4S bits are writable only when the channel is OFF (CC4E = '0' in TIMx_CCER).
7	OC3CE	RW	0	Output Compare 3 clear enable
6:4	OC3M [2:0]	RW	00	Output compare 3 mode
3	OC3PE	RW	0	Output Compare 3 preload enable
2	OC3FE	RW	0	Output Compare 3 fast enable
1:0	CC3S [1:0]	RW	00	Capture/Compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC3 channel is configured as output 01:CC3 channel is configured as input, IC3 is mapped on TI3 10:CC3 channel is configured as input, IC3 is mapped on TI4 11:CC3 channel is configured as input, IC3 is mapped on TRC This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC3S bits are writable only when the channel is OFF (CC3E = '0' in TIMx_CCER).

Input capture mode:

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:12	IC4F [3:0]	RW	0000	Input capture 4 filter
11:10	IC4PSC [1:0]	RW	00	Capture/Compare 4 prescaler
9:8	CC4S [1:0]	RW	00	Capture/Compare 4 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC4 channel is configured as output 01:CC4 channel is configured as input, IC4 is mapped on TI4. 10:CC4 channel is configured as input, IC4 is mapped on TI3. 11:CC4 channel is configured as input, IC4 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register). Note:CC4S bits are writable only when the channel is OFF (CC4E = '0' in TIMx_CCER).

Puya Confidential

7:4	IC3F [3:0]	RW	0000	Capture/Compare 3 filter
3:2	IC3PSC [1:0]	RW	00	Input capture 3 prescaler
1:0	CC3S [1:0]	RW	00	Capture/Compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC3 channel is configured as output 01:CC3 channel is configured as input, IC3 is mapped on TI3. 10:CC3 channel is configured as input, IC3 is mapped on TI4. 11:CC3 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC3S bits are writable only when the channel is OFF (CC3E = '0' in TIMx_CCER).

19.4.9. TIM1 capture/compare enable register (TIM1_CCER)

Address offset:0x20

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:14	Reserved	-	0	Reserved, must be kept at reset value.
13	CC4P	RW	0	Capture/compare 4 output polarity Refer to CC1P description.
12	CC4E	RW	0	Capture/compare 4 output enable Refer to CC1E description.
11	CC3NP	RW	0	Capture/compare 3 complementary output polarity. Refer to CC1NP description.
10	CC3NE	RW	0	Capture/compare 3 complementary output enable Refer to CC1NE description.
9	CC3P	RW	0	Capture/compare 3 output polarity Refer to CC1P description.
8	CC3E	RW	0	Capture/compare 3 output enable Refer to CC1E description.
7	CC2NP	RW	0	Capture/compare 2 complementary output polarity. Refer to CC1NP description.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
6	CC2NE	RW	0	Capture/compare 2 complementary output enable Refer to CC1NE description.
5	CC2P	RW	0	Capture/compare 2 output polarity Refer to CC1P description.
4	CC2E	RW	0	Capture/compare 2 output enable Refer to CC1E description.
3	CC1NP	RW	0	Capture/compare 1 complementary output polarity. 0:OC1N active high. 1:OC1N active low. Note:Once the LOCK level (LOCK bit in the TIMx_BDTR register) is set to 3 or 2 and CC1S = 00 (channel configuration as output), this bit cannot be modified.
2	CC1NE	RW	0	Capture/compare 1 complementary output enable 0:OC1N output disabled 1:OC1N signal output to corresponding output pin When the CC1 channel is configured as an output, the OC1N output level is determined by the MOE, OSSI, OSSR, OIS1, OIS1N, CC1E, and CC1NE bits, as shown in the table below For complementary output channels, this bit is preloaded. If the CCPC bit is set in the TIMx_CR2 register then the CC1E active bit takes the new value from the preloaded bit only when a com event is generated.
1	CC1P	RW	0	Capture/compare 1 output polarity CC1 channel configured as output: 0:OC1 active high. 1:OC1 active low. CC1 channel configured as input: CC1NP/CC1P bits select the active polarity of TI1FP1 and TI2FP1 for trigger or capture operations. 00:Non-inverted/rising edge: The circuit is sensitive to TIxFP1 rising edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode or encoder mode). 01:Inverted/falling edge: The circuit is sensitive to TIxFP1 falling edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is inverted (trigger operation in gated mode or encoder mode). 10:reserved, do not use this configuration. 11:non-inverted/double edge

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				The circuit is sensitive to both TIxFP1 rising and falling edges (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode). This configuration must not be used in encoder mode. Notes: 1. For complementary output channels, this bit is preloaded. If the CCPC bit is set in the TIMx_CR2 register, then the CC1P active bit takes the new value from the preloaded bit only when a Commutation event is generated. 2. Once the LOCK level (LOCK bit in the TIMx_BDTR register) is set to 3 or 2, this bit cannot be modified
0	CC1E	RW	0	Capture/Compare 1 output enable 0:Off - OC1 is not active. 1:On - OC1 signal is output on the corresponding output pin depending on MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits. Notes: For complementary output channels, this bit is preloaded. If the CCPC bit is set in the TIMx_CR2 register then the CC1E active bit takes the new value from the preloaded bit only when a Commutation event is generated.

Table 19-3 Output control bits for complementary OCx and OCxN channels with break feature

Control bits					Output states	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx output state	OCxN output state
1	x	0	0	0	Output disabled (not driven by the timer anymore). OCx = 0, OCx_EN = 0	Output disabled (not driven by the timer anymore). OCxN = 0, OCxN_EN = 0
		0	0	1	Output disabled (not driven by the timer anymore). OCx = 0, OCx_EN = 0	OCxREF + polarity, OCxN = OCREF xor CCxNP, OCxN_EN = 1
		0	1	0	OCxREF + polarity, OCx = OCREF xor CCxP, OCx_EN = 1	Output disabled (not driven by the timer anymore). OCxN = 0, OCxN_EN = 0
		0	1	1	OCxREF + Polarity + dead-time, OCx_EN = 1	OCxREF + polarity + dead-time OCxN_EN = 1

Puya Confidential

Control bits					Output states	
MOE	**OSSI**	**OSSR**	**CCxE**	**CCxNE**	**OCx output state**	**OCxN output state**
		1	0	0	Output disabled (not driven by the timer anymore). OCx = CCxP, OCx_EN = 0	Output disabled (not driven by the timer anymore). OCxN = CCxNP, OCxN_EN = 0
		1	0	1	Off-State (output enabled with inactive state) OCx = CCxP, OCx_EN = 1	OCxREF + polarity, OCxN = OCREF xor CCxNP, OCxN_EN = 1
		1	1	0	OCxREF + polarity, OCx = OCREF xor CCxP, OCx_EN = 1	Off-State (output enabled with inactive state) OCxN = CCxNP, OCxN_EN = 1
		1	1	1	OCxREF + Polarity + dead-time, OCx_EN = 1	OCxREF + polarity + dead-time OCxN_EN = 1
0	0	x	0	0	Output disabled (not driven by the timer anymore).	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1	Off-State (output enabled with inactive state) Asynchronously:OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCx_EN=1 Then if the clock is present:OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state	
	1		1	0		
	1		1	1		

19.4.10. TIM1 counter (TIM1_CNT)

Address offset:0x24

Reset value:0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CNT [15:0]	RW	0	Counter value

Puya Confidential

19.4.11. TIM1 prescaler (TIM1_PSC)

Address offset:0x28

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSC [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PSC [15:0]	RW	0	Prescaler value The counter clock frequency (CK_CNT) is equal to fCK_PSC / (PSC [15:0] +1). PSC contains the value to be loaded in the active pre-scaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in reset mode).

19.4.12. TIM1 auto-reload register (TIM1_ARR)

Address offset:0x2C

Reset value:0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ARR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	ARR [15:0]	RW	0xFFFF	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. The counter is blocked while the auto-reload value is null.

19.4.13. TIM1 Repetition counter register (TIM1_RCR)

Address offset:0x30

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Puya Confidential

Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	REP [7:0]							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7:0	REP [7:0]	RW	0	Repetition counter value These bits allow the user to set-up the update rate of the compare registers (i.e. periodic transfers from preload to active registers) when preload registers are enabled, as well as the update interrupt generation rate, if this interrupt is enabled. Each time the REP_CNT related downcounter reaches zero, an update event is generated, and it restarts counting from REP value. As REP_CNT is reloaded with REP value only at the repetition update event U_RC, any write to the TIMx_RCR register is not taken in account until the next repetition update event. It means in PWM mode (REP+1) corresponds to: - the number of PWM periods in edge-aligned mode - the number of half PWM period in center-aligned mode.

19.4.14. TIM1 capture/compare register 1 (TIM1_CCR1)

Address offset:0x34

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR1 [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CCR1 [15:0]	RW	0	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).

19.4.15. TIM1 capture/compare register 2 (TIM1_CCR2)

Address offset:0x38

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR2 [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CCR2 [15:0]	RW	0	Capture/Compare 2 value If channel CC2 is configured as output: CCR2 is the value to be loaded in the actual capture/compare 2 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC2PE). Else the preload value is copied in the active capture/compare 2 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC2 output. If channel CC2 is configured as input: CCR2 is the counter value transferred by the last input capture 2 event (IC2).

19.4.16. TIM1 capture/compare register 3 (TIM1_CCR3)

Address offset:0x3C

Puya Confidential

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR3 [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CCR3 [15:0]	RW	0	Capture/Compare 3 value If channel CC3 is configured as output: CCR3 is the value to be loaded in the actual capture/compare 3 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR2 register (bit OC3PE). Else the preload value is copied in the active capture/compare 3 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC output. If channel CC3 is configured as input: CCR3 is the counter value transferred by the last input capture 3 event (IC3).

19.4.17. TIM1 capture/compare register 4 (TIM1_CCR4)

Address offset:0x40

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR4 [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CCR4 [15:0]	RW	0	Capture/Compare 4 value If channel CC4 is configured as output:

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				CCR4 is the value to be loaded in the actual capture/compare 4 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR2 register (bit OC4PE). Else the preload value is copied in the active capture/compare 4 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC output. If channel CC4 is configured as input: CCR4 is the counter value transferred by the last input capture 4 event (IC4).

19.4.18. TIM1 break and dead-time register (TIM1_BDTR)

Address offset:0x44

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DTG [7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	MOE	RW	0	Main output enable This bit is cleared asynchronously by hardware as soon as one of the break inputs is active. It is cleared by software or automatically setting depending on the AOE bit. It is acting only on the channels which are configured in output. 0:OC and OCN outputs are disabled or forced to idle state. 1:OC and OCN outputs are enabled if their respective enable bits are set (CCxE, CCxNE in TIMx_CCER register).
14	AOE	RW	0	Automatic output enable 0:MOE can be set only by software; 1:MOE can be set by software or automatically at the next update event (if none of the break inputs is active). Note:This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIM1_BDTR register).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
13	BKP	RW	0	Break polarity 0:Break input BRK is active low; 1:Break input BRK is active high Notes: 1. This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIMx_BDTR register). 2. Any write operation to this bit takes a delay of 1 APB clock cycle to become effective.
12	BKE	RW	0	Break enable 0:Break function disabled; 1:Break function enabled Notes: 1. This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIMx_BDTR register). 2. Any write operation to this bit takes a delay of 1 APB clock cycle to become effective.
11	OSSR	RW	0	Off-state selection for Run mode This bit is used when MOE=1 on channels having a complementary output which are configured as outputs. OSSR bit is not implemented if no complementary output is implemented in the timer. See OC/OCN enable description for more details (TIMx capture/compare enable register (TIMx_CCER)). 0:OC/OCN outputs are disabled 1:OC/OCN outputs are enabled with their inactive level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1 Note:This bit cannot be modified as long as LOCK level 2 has been set (LOCK bits in TIMx_BDTR register).
10	OSSI	RW	0	Off-state selection for Idle mode This bit is used when MOE=0 and on channels configured as outputs. See OC/OCN enable description for more details (TIMx capture/compare enable register (TIMx_CCER)). 0:OC/OCN outputs are disabled 1:OC/OCN outputs are forced first with their idle level as soon as CCxE=1 or CCxNE=1. Note:This bit cannot be modified as long as LOCK level 2 has been set (LOCK bits in TIMx_BDTR register).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
9:8	LOCK [1:0]	RW	00	Lock configuration These bits offer a write protection against software errors. 00:LOCK OFF, no bit is write protected. 01:LOCK Level 1, DTG, BKE, BKP, AOE bits in TIMx_BDTR register and OISx and OISxN bits in TIMx_CR2 register can no longer be written. 10:LOCK Level 2 = LOCK Level 1 + CC Polarity bits (CCxP/CCxNP bits in TIMx_CCER register, as long as the related channel is configured in output through the CCxS bits) as well as OSSR and OSSI bits can no longer be written. 11:LOCK level 3 = LOCK level 2 + CC control bits (OCxM and OCxPE bits in TIMx_CCMRx registers, as long as the related channel is configured in output through the CCxS bits) can no longer be written. Note:The LOCK bits can be written only once after the reset. Once the TIMx_BDTR register has been written, their content is frozen until the next reset.
7:0	DTG [7:0]	RW	0000 0000	Dead-time generator setup This bit-field defines the duration of the dead-time inserted between the complementary outputs. DT correspond to this duration. DTG [7:5] = 0xx = > DT = DTG [7:0] × Tdtg, Tdtg = TDTS; DTG [7:5] = 10x = > DT = (64 + DTG [5:0]) × Tdtg, Tdtg = 2 × TDTS; DTG [7:5] = 110 = > DT = (32 + DTG [4:0]) × Tdtg, Tdtg = 8 × TDTS; DTG [7:5] = 111 = > DT = (32 + DTG [4:0]) × Tdtg, Tdtg = 16 × TDTS; Example:If TDTS = 125ns (8MHz), the dead time possible value are: 0 to 15875 ns by 125 ns steps; 16 µs to 31750 ns by 250 ns steps; 32 µs to 63us by 1 µs steps; 64 µs to 126 us by 2 µs steps Note:This bit-field cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).

19.4.19. TIM1 DMA control register (TIM1_DCR)

Address offset:0x48

Puya Confidential

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	DBL [4:0]					Res			DBA [4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:13	Reserved	-	-	Reserved
12:8	DBL [4:0]	RW	0 0000	DMA burst length These bits define the transfer length of the DMA in continuous mode (when reading or writing to the address of the TIMx_DMAR register address is performed). 00000:1 transfer 00001:2 transfers 00010:3 transfer .... 10001:18 transfers
7:5	Reserved	RW	0	Reserved, must be kept at reset value.
4:0	DBA [4:0]	RW	0 0000	DBA [4:0]:DMA base address These bits define the base address of the DMA in continuous mode (when read/write access are done through the TIMx_DMAR address). DBA is defined as an offset starting from the address of the TIMx_CR1 register. 00000:TIMx_CR1 00001:TIMx_CR2 00010:TIMx_SMCR .....

19.4.20. TIM1 DMA address for full transfer (TIM1_DMAR)

Address offset:0x4C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAB [15:0]															
RW															

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	DMAB [15:0]	RW	0	DMA register for burst accesses A read or write operation to the DMAR register accesses the register located at the address: (TIMx_CR1 address) + (DBA + DMA index) x 4 where: TIMx_CR1 address is the address of the control register 1; DBA is the DMA base address configured in TIMx_DCR register; DMA index is automatically controlled by the DMA transfer, and ranges from 0 to DBL (DBL configured in TIMx_DCR).

Puya Confidential

20. General-purpose timers (TIM2/3)

20.1. TIM2/TIM3 introduction

The TIM2 general-purpose timer consists of a 32-bit auto-reload counter driven by a programmable prescaler.

The TIM3 general-purpose timer consists of a 16-bit auto-reload counter driven by a programmable prescaler.

It may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers.

The timers are completely independent, and do not share any resources. They can be synchronized together.

20.2. TIM2/3 main features

General purpose TIM2/3 timer features include:

- 16-bit (TIM3), 32-bit (TIM2) up, down, up/down auto-reload counter
- 16-bit programmable prescaler used to divide the counter clock frequency by any factor between 1 and 65536
- 4 independent channels for
  - Input capture
  - Output compare
  - PWM generation (edge- and center-aligned modes)
  - One-pulse mode output
- Synchronization circuit to control the timer with external signals and to interconnect several timers.
- Interrupt/DMA generation on the following events:
  - Update:counter overflow/underflow, counter initialization (by software or internal/external trigger)
  - Trigger event (counter start, stop, initialization or count by internal/external trigger)
  - Input capture
  - Output compare
- Support incremental (quadrature) encoder and Hall sensor circuits for positioning.
- Trigger input for external clock or cycle-by-cycle current management

Puya Confidential

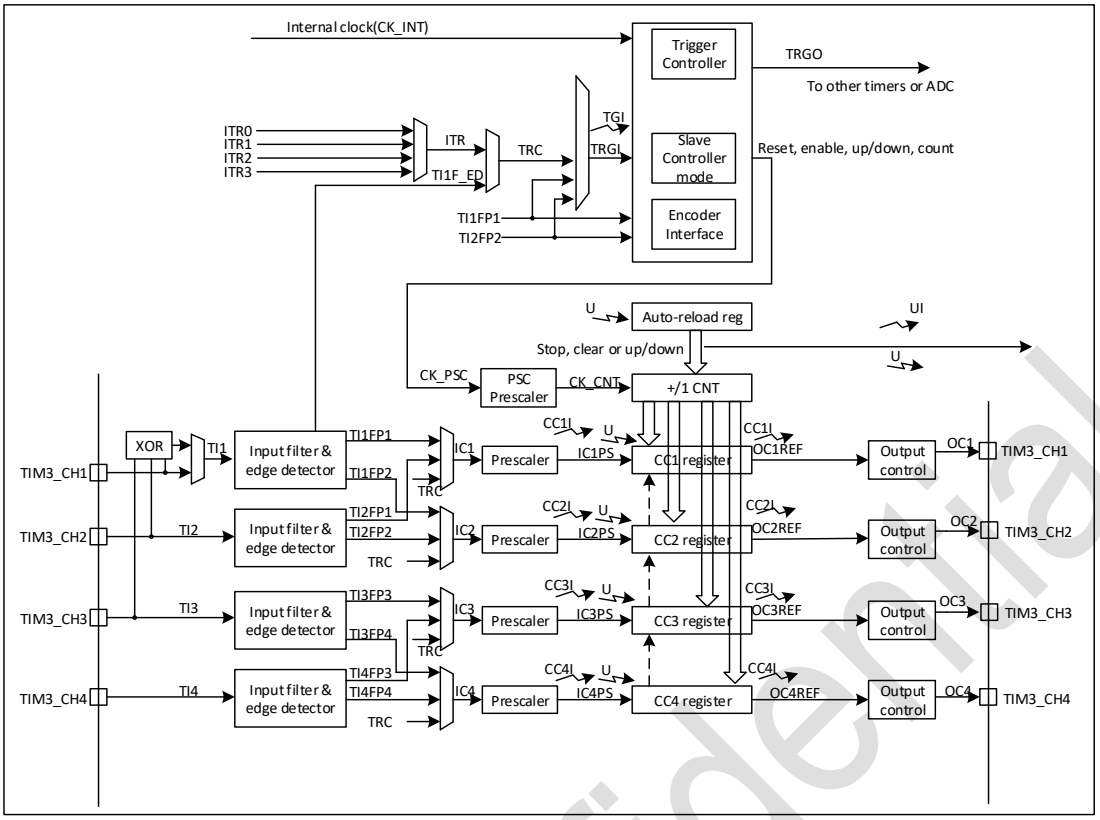


Figure 20-1 General-purpose timer block diagram

20.3. TIM2/3 functional description

20.3.1. Time-base unit

The main block of the TIM2/3 timer is a 16-bit (TIM3) or 32-bit (TIM2) counter with its related auto-reload register. The counter can count up, down or both up and down. The counter clock can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.

The time-base unit includes:

- Counter register (TIMx_CNT)
- Prescaler register (TIMx_PSC)
- Auto-reload register (TIMx_ARR)

The auto-load register is preloaded. Writing to or reading from the auto-load register accesses the preload register. The content of the preload register is transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload preload enable bit (ARPE) in TIMx_CR1 register. The update event is sent when the counter reaches the overflow (or underflow when downcounting) and if the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software.

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in TIMx_CR1 register is set.

Note that the counter starts counting 1 clock cycle after setting the CEN bit in the TIMx_CR register.

Puya Confidential

Prescaler description

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIMx_PSC register). It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

Figures below give some examples of the counter behavior when the prescaler ratio is changed on the fly:

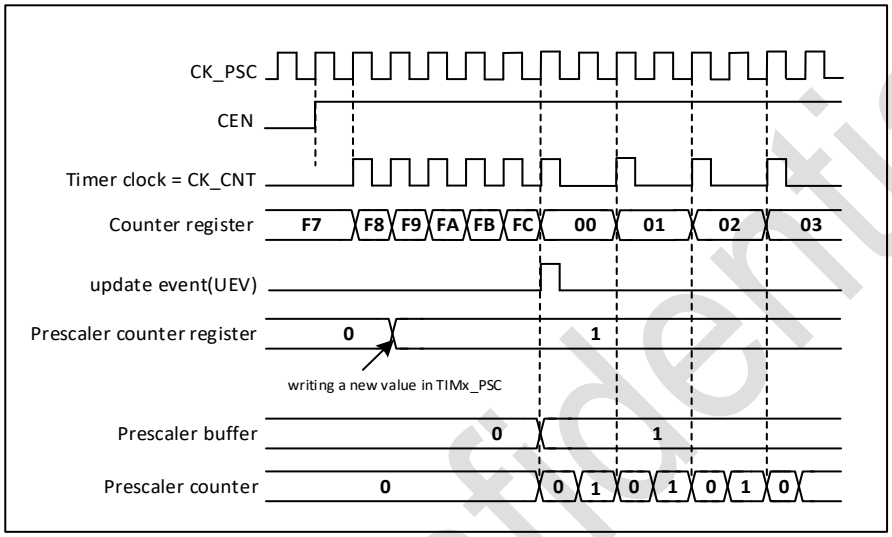


Figure 20-2 Counter timing diagram with prescaler division change from 1 to 2

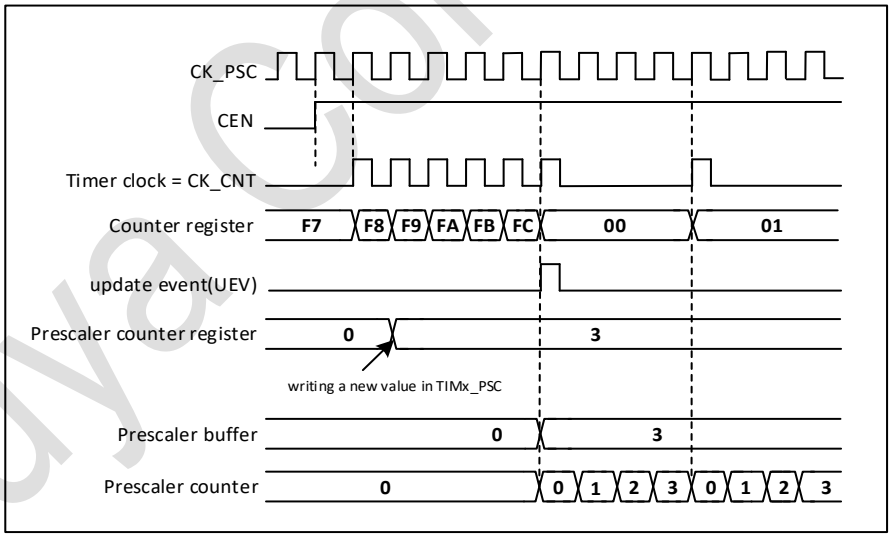


Figure 20-3 Counter timing diagram with prescaler division change from 1 to 4

20.3.2. Counter mode

20.3.2.1. Upcounting mode

In upcounting mode, the counter counts from 0 to the auto-reload value, then restarts from 0 and generates a counter overflow event.

An Update event can be generated at each counter overflow or by setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller).

Puya Confidential

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts from 0, as well as the counter of the prescaler (but the prescale rate does not change). In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag (thus no interrupt or DMA request is sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The auto-reload shadow register is updated with the preload value (TIMx_ARR).
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR=0x36.

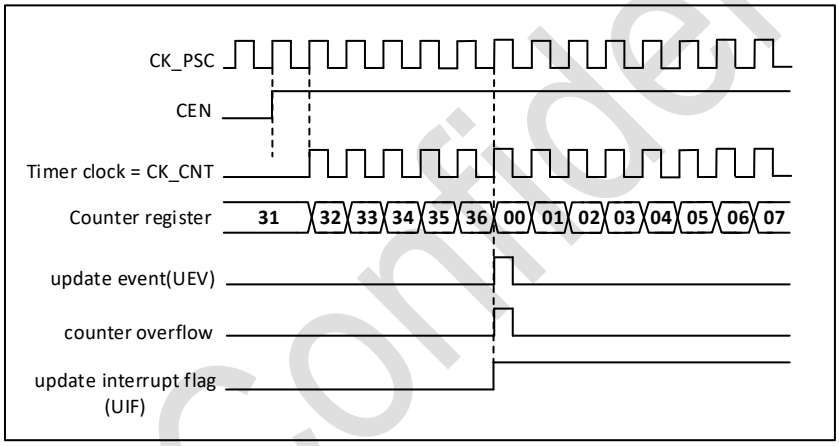


Figure 20-4 Counter timing diagram, internal clock divided by 1

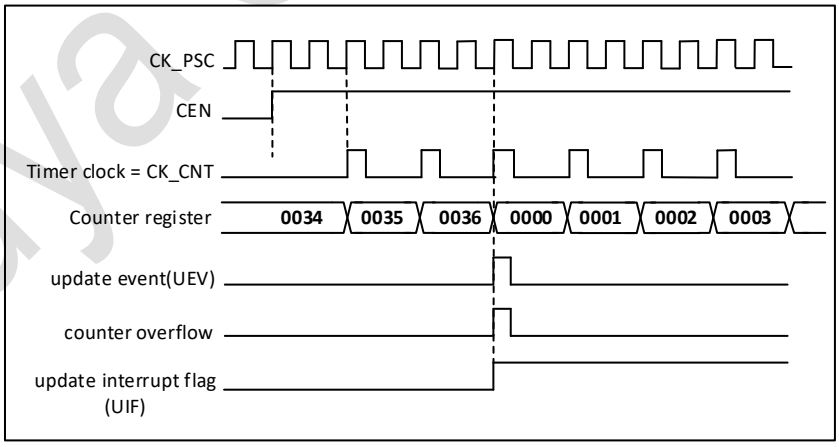


Figure 20-5 Counter timing diagram, internal clock divided by 2

Puya Confidential

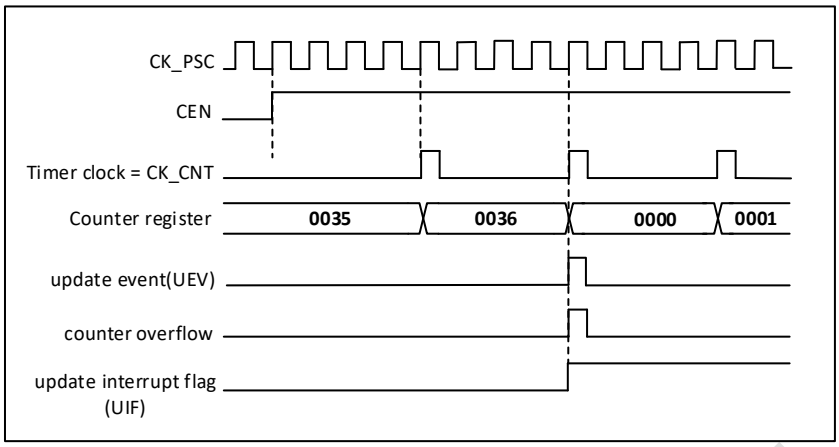


Figure 20-6 Counter timing diagram, internal clock divided by 4

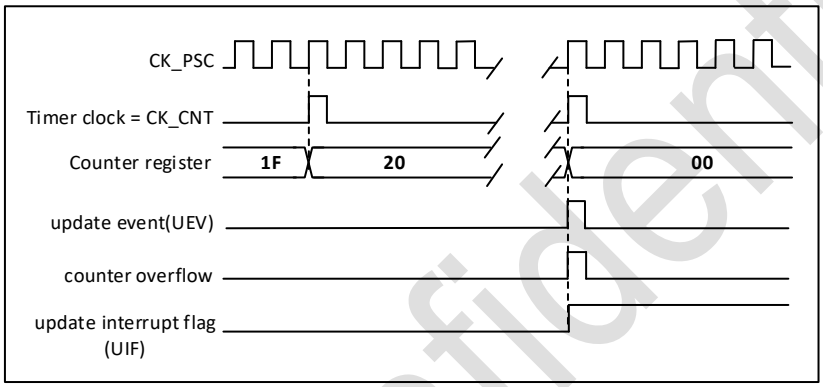


Figure 20-7 Counter timing diagram, internal clock divided by N

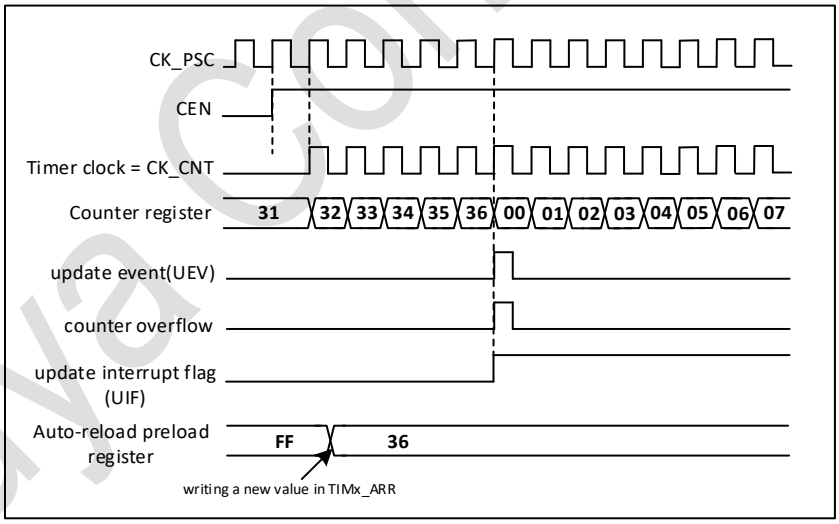


Figure 20-8 Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded)

Puya Confidential

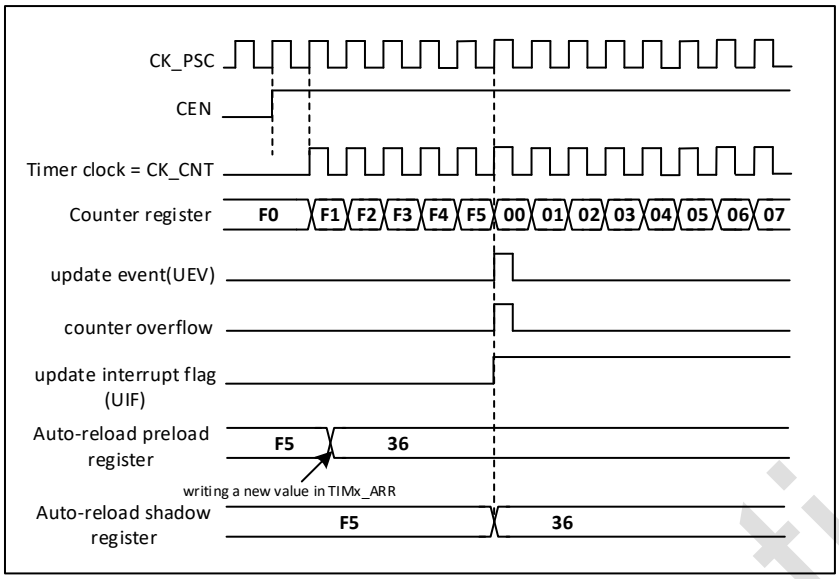


Figure 20-9 Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded)

20.3.2.2. Downcounting mode

In downcounting mode, the counter counts from the auto-reload value down to 0, then restarts from the auto-reload value and generates a counter underflow event.

An Update event can be generating at each counter underflow or by setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller).

The UEV update event can be disabled by software through setting the UDIS bit in TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until UDIS bit has been written to 0. However, the counter restarts from the current auto-reload value, whereas the counter of the prescaler restarts from 0 (but the prescale rate doesn't change).

In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an UEV update event but without setting the UIF flag (thus no interrupt and DMA requests are sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).
- The auto-reload active register is updated with the preload value (content of the TIMx_ARR register).

Note that the auto-reload is updated before the counter is reloaded, so that the next period is the expected one.

Puya Confidential

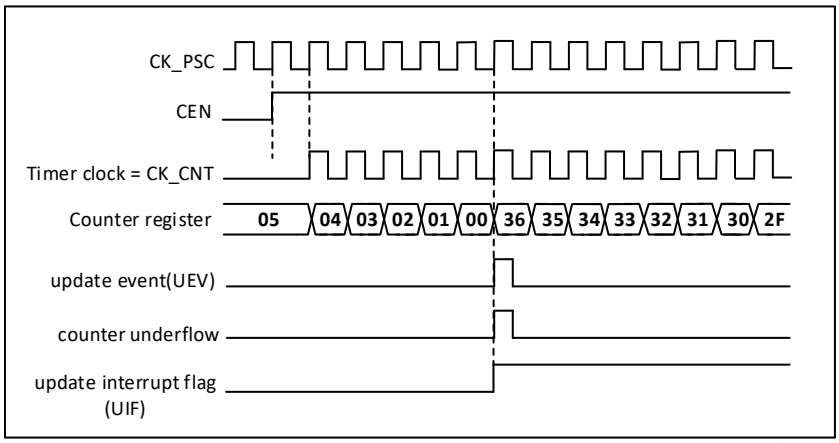


Figure 20-10 Counter timing diagram, internal clock divided by 1

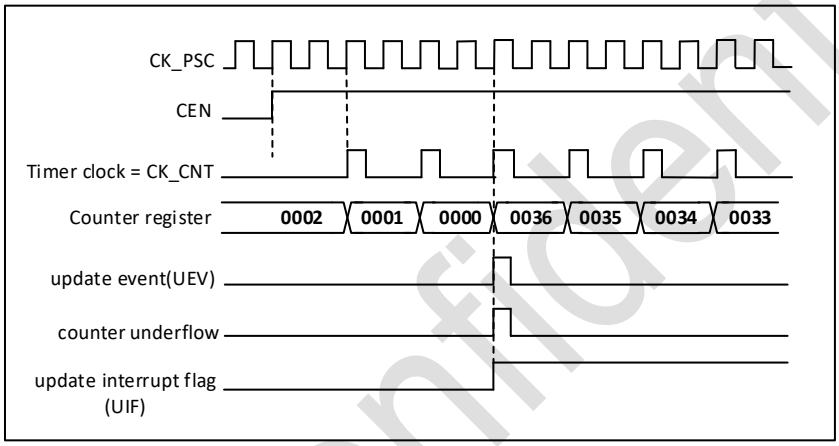


Figure 20-11 Counter timing diagram, internal clock divided by 2

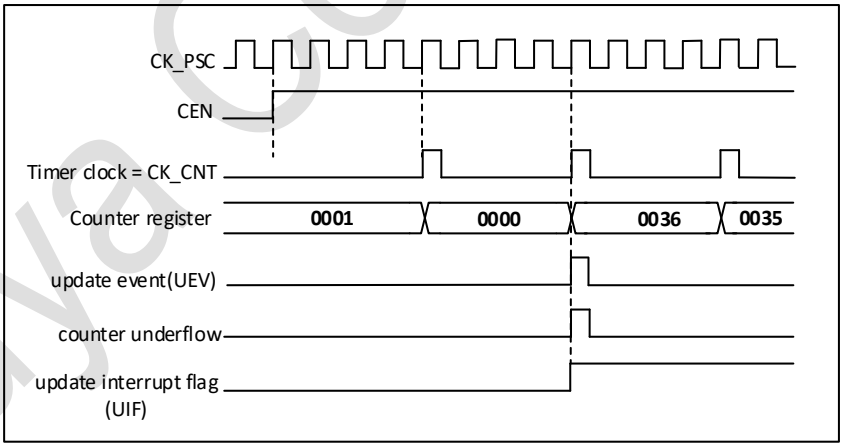


Figure 20-12 Counter timing diagram, internal clock divided by 4

Puya Confidential

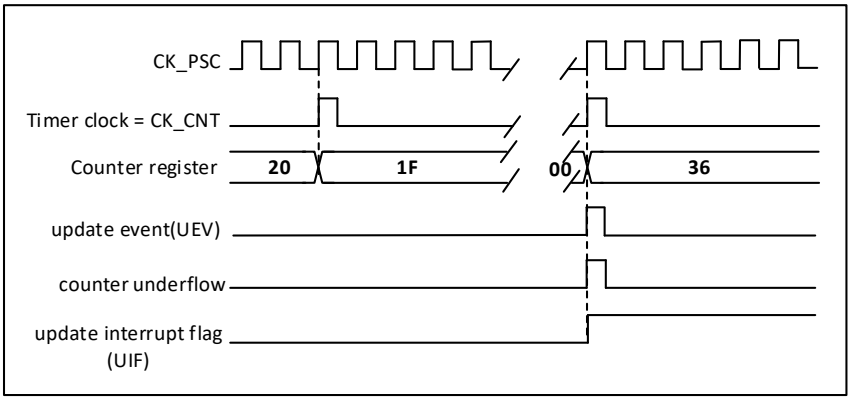


Figure 20-13 Counter timing diagram, internal clock divided by N

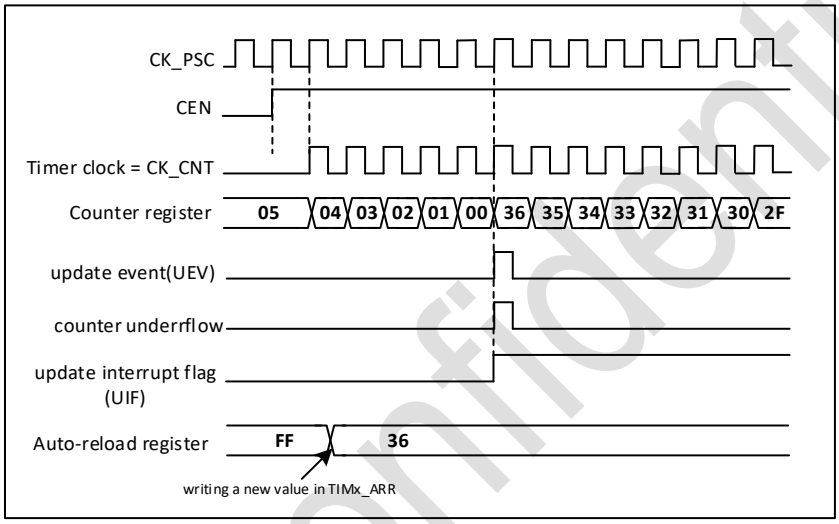


Figure 20-14 Counter timing diagram, update event when repetition counter is not used

Center-aligned mode (up/down counting)

In center-aligned mode, the counter counts from 0 to the auto-reload value (content of the TIMx_ARR register) – 1, generates a counter overflow event, then counts from the auto-reload value down to 1 and generates a counter underflow event. Then it restarts counting from 0.

Center-aligned mode is active when the CMS bits in TIMx_CR1 register are not equal to '0'. The Output compare interrupt flag of channels configured in output is set when:the counter counts down (Center aligned mode 1, CMS = “01”), the counter counts up (Center aligned mode 2, CMS = “10”) the counter counts up and down (Center aligned mode 3, CMS = “11”).

In this mode, the DIR direction bit in the TIMx_CR1 register cannot be written. It is updated by hardware and gives the current direction of the counter.

The update event can be generated at each counter overflow and at each counter underflow or by setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller) also generates an update event. In this case, the counter restarts counting from0, as well as the counter of the prescaler.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no

Puya Confidential

update event occurs until UDIS bit has been written to 0. However, the counter continues counting up and down, based on the current auto-reload value.

In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an UEV update event but without setting the UIF flag (thus no interrupt and DMA requests are sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).
- The auto-reload active register is updated with the preload value (content of the TIMx_ARR register).

Note that if the update source is a counter overflow, the auto-reload is updated before the counter is reloaded, so that the next period is the expected one (the counter is loaded with the new value).

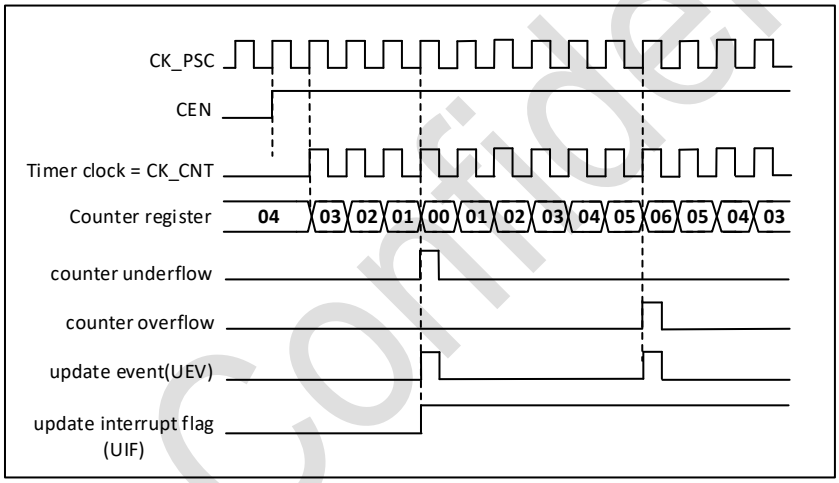


Figure 20-15 Counter timing diagram, internal clock divided by 1, TIMx_ARR = 0x6

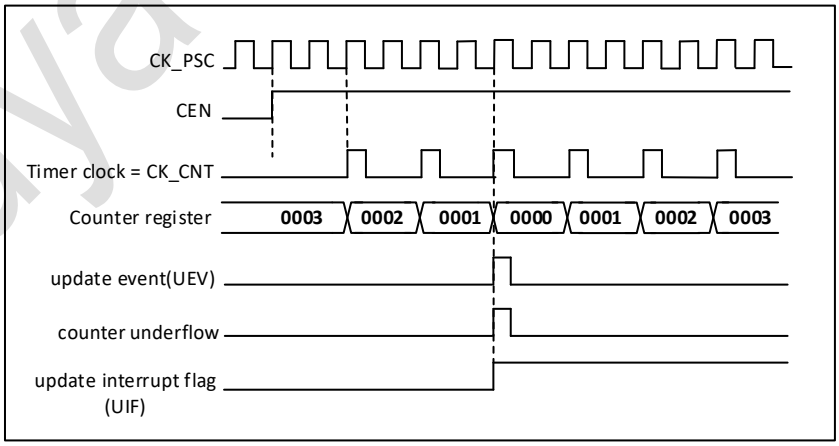


Figure 20-16 Counter timing diagram, internal clock divided by 2

Puya Confidential

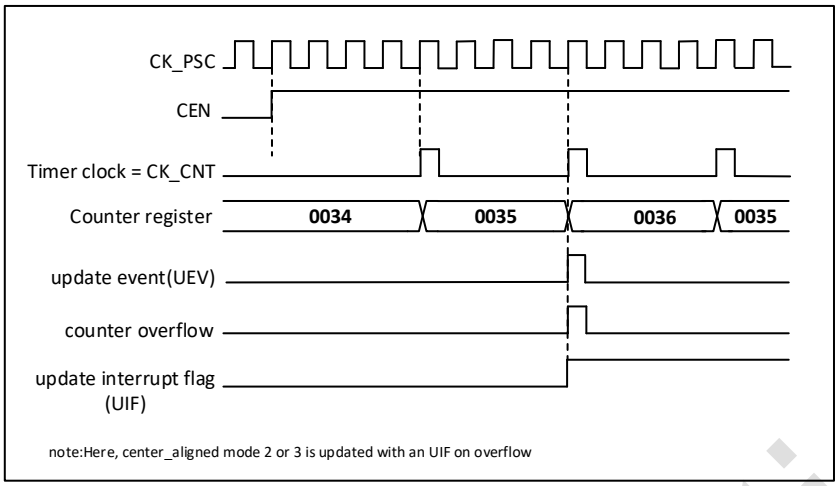


Figure 20-17 Counter timing diagram, internal clock division factor is 4, TIMx_ARR = 0x36

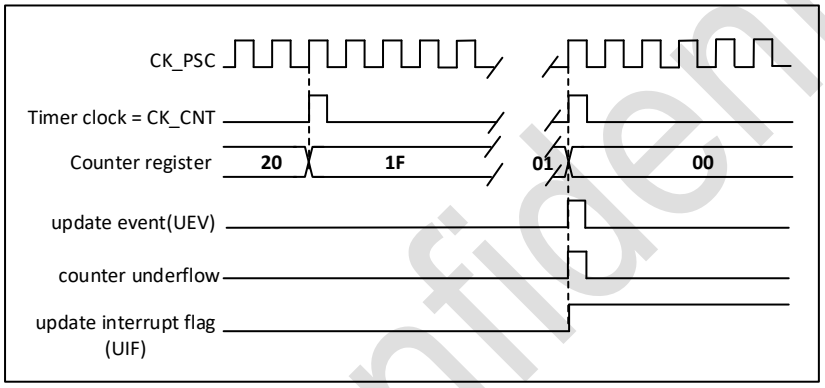


Figure 20-18 Counter timing diagram, internal clock divided by N

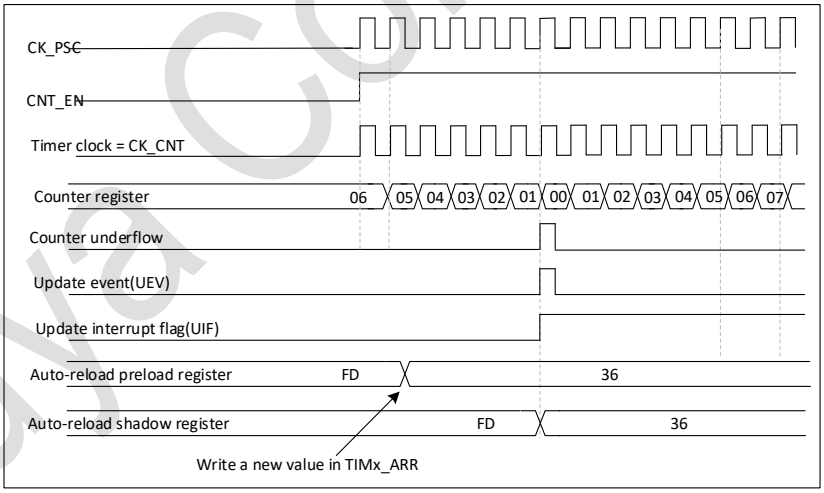


Figure 20-19 Counter timing diagram, update event with ARPE=1 (counter underflow)

Puya Confidential

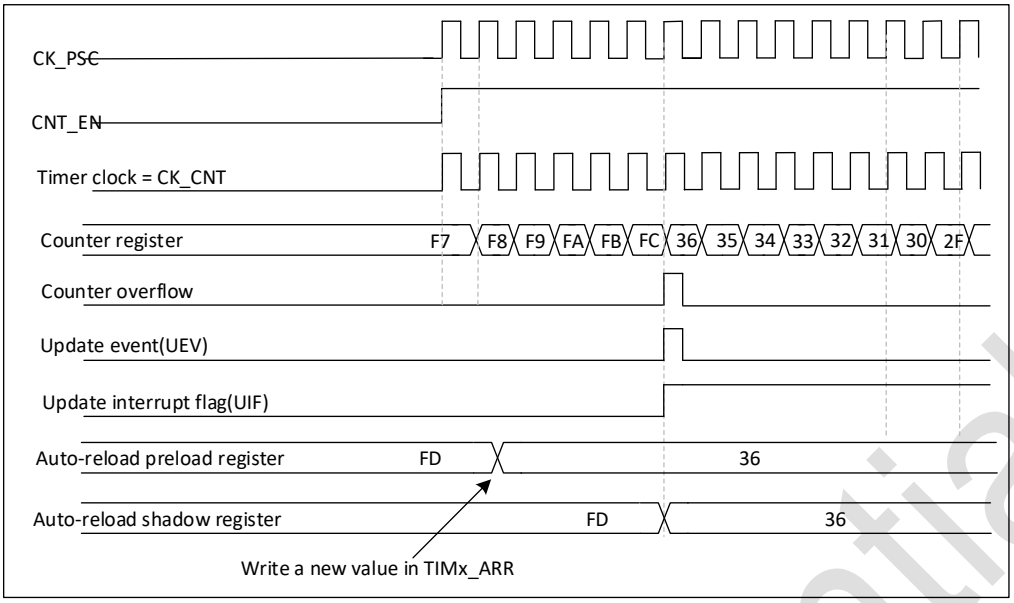


Figure 20-20 Counter timing diagram, update event with ARPE=1 (counter overflow)

20.3.3. Clock sources

The counter clock can be provided by the following clock sources:

- Internal clock (CK_INT)
- External clock mode1:external input pin
- Internal trigger inputs (ITRx):using one timer as prescaler for another timer. For example, the user can configure Timer 1 to act as a prescaler for Timer 3.

Internal clock source (CK_INT)

If the slave mode controller is disabled, then the CEN, DIR (in the TIMx_CR1 register) and UG bits (in the TIMx_EGR register) are actual control bits and can be changed only by software. As soon as the CEN bit is written to 1, the prescaler is clocked by the internal clock CK_INT.

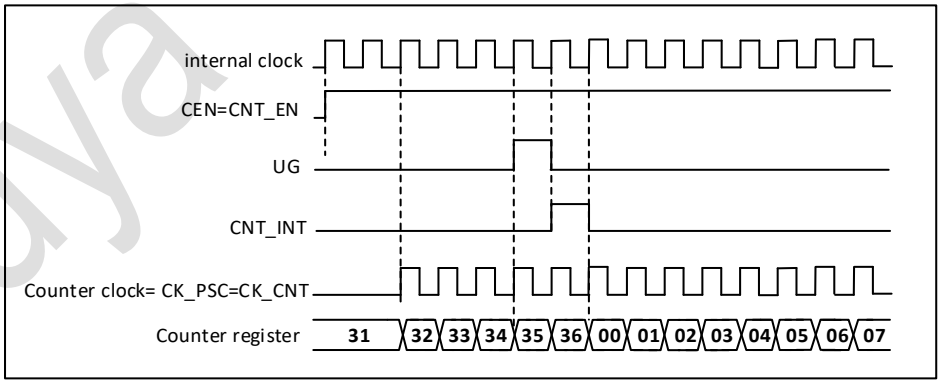


Figure 20-21 Control circuit in normal mode, internal clock divided by 1

External clock source mode 1

This mode is selected when SMS=111 in the TIMx_SMCR register. The counter can count at each rising or falling edge on a selected input.

Puya Confidential

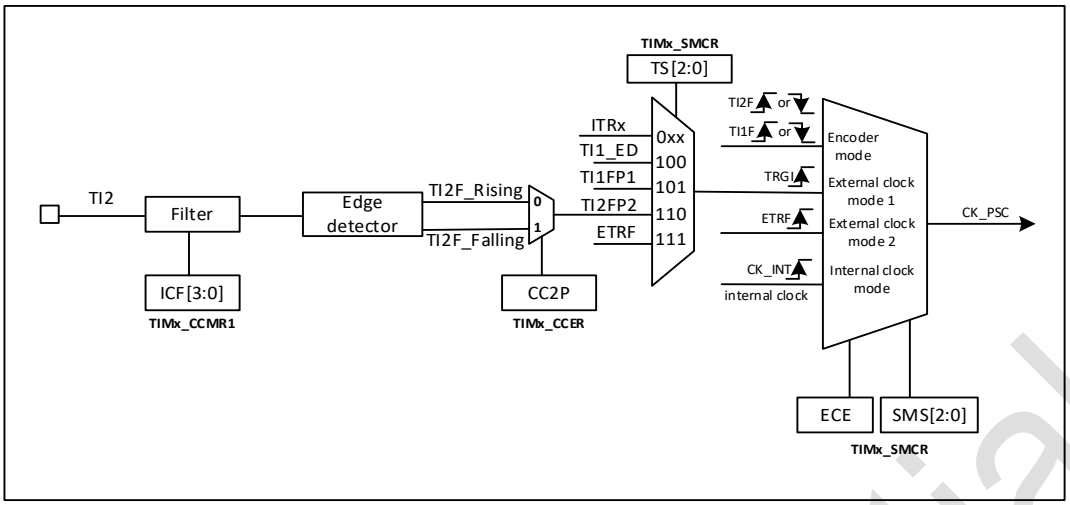


Figure 20-22 TI2 external clock connection example

For example, to configure the upcounter to count in response to a rising edge on the TI2 input, use the following procedure:

1.Configure channel 2 to detect rising edges on the TI2 input by writing CC2S= 01 in the TIMx_CCMR1 register.

2.Configure the input filter duration by writing the IC2F [3:0] bits in the TIMx_CCMR1 register (if no filter is needed, keep IC2F=0000).

3.Select rising edge polarity by writing CC2P=0 in the TIMx_CCER register.

4.Configure the timer in external clock mode 1 by writing SMS=111 in the TIMx_SMCR register.

5. Select TI2 as the input source by writing TS=110 in the TIMx_SMCR register.

6. Enable the counter by writing CEN=1 in the TIMx_CR1 register.

Note:The capture prescaler is not used for triggering, so there's no need to configure it. When a rising edge occurs on TI2, the counter counts once and the TIF flag is set.

The delay between the rising edge on TI2 and the actual clock of the counter is due to the resynchronization circuit on TI2 input.

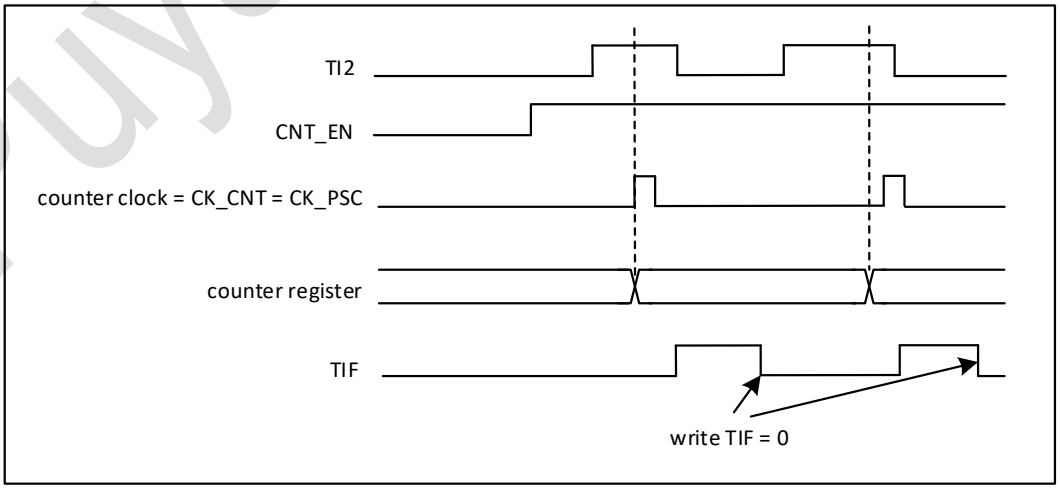


Figure 20-23 Control circuit in external clock mode 1

Puya Confidential

20.3.4. Capture/Compare channels

Each Capture/Compare channel is built around a capture/compare register (including a shadow register), an input stage for capture (with digital filter, multiplexing and prescaler) and an output stage (with comparator and output control).

The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detector with polarity selection generates a signal (TIxFPx) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (ICxPS).

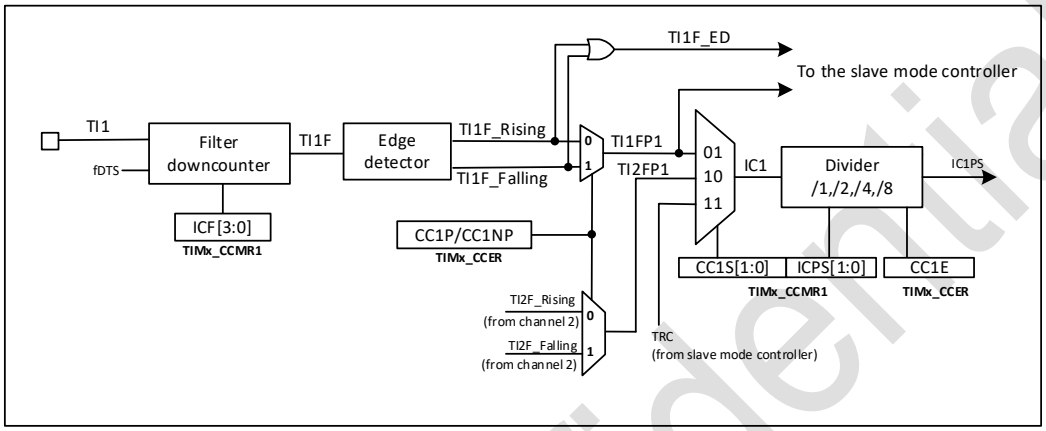


Figure 20-24 Capture/compare channel (example:channel 1 input stage)

The output stage generates an intermediate waveform which is then used for reference:OCxREF (active high). The polarity acts at the end of the chain.

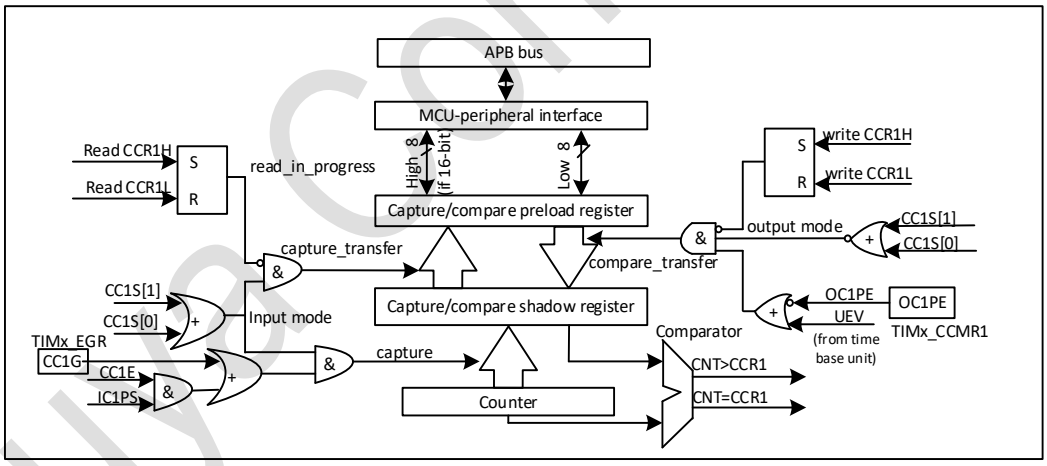


Figure 20-25 Capture/compare channel 1 main circuit

Puya Confidential

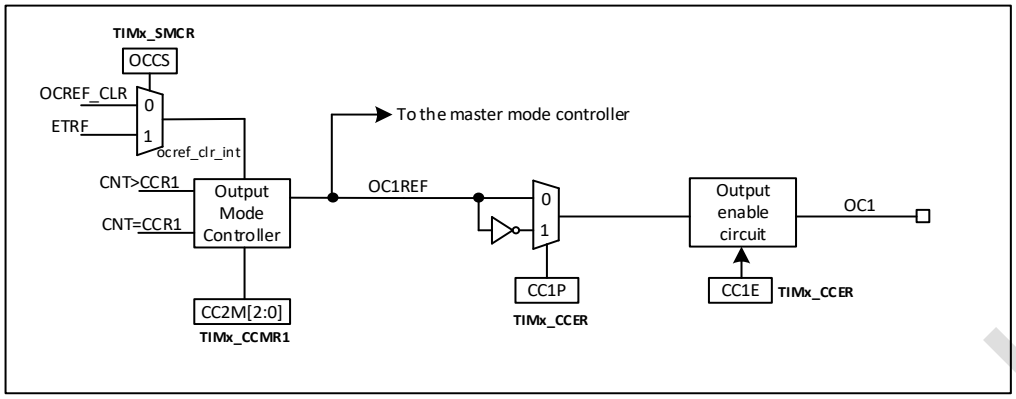


Figure 20-26 Output stage of capture/compare channel (channel 1)

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register.

In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

20.3.5. Input capture mode

In Input capture mode, the capture/compare registers are used to latch the value of the counter after a transition detected by the corresponding ICx signal. When a capture occurs, the corresponding CCXIF flag (TIMx_SR register) is set and an interrupt or a DMA request can be sent if they are enabled. If a capture occurs while the CCxIF flag was already high, then the over-capture flag CCxOF (TIMx_SR register) is set. CCxIF can be cleared by software by writing it to ‘0’ or by reading the captured data stored in the TIMx_CCRx register. CCxOF is cleared when you write it to ‘0’.

The following example shows how to capture the counter value in TIMx_CCR1 when TI1input rises. To do this, use the following procedure:

- Select the active input:TIMx_CCR1 must be linked to the TI1 input, so write the CC1S bits to 01 in the TIMx_CCR1 register. As soon as CC1S becomes different from 00, the channel is configured in input and the TIMx_CCR1 register becomes read-only.
- Program the needed input filter duration with respect to the signal connected to the timer (by programming ICxF bits in the TIMx_CCMRx register if the input is a TIx input). Let’s imagine that, when toggling, the input signal is not stable during at most 5 internal clock cycles. We must program a filter duration longer than these 5 clock cycles. We can validate a transition on TI1 when 8 consecutive samples with the new level have been detected (sampled at f_{DTS} frequency). Then write IC1F bits to 0011 in the TIMx_CCMR1 register.
- Select the edge of the active transition on the TI1 channel by programming CC1P and CC1NP bits to ‘00’ in the TIMx_CCER register (rising edge in this case).
- Program the input prescaler. In our example, we wish the capture to be performed at each valid transition, so the prescaler is disabled (write IC1PS bits to ‘00’ in the TIMx_CCMR1 register).

Puya Confidential

- Enable capture from the counter into the capture register by setting the CC1E bit in the TIMx_CCER register.
- If needed, enable the related interrupt request by setting the CC1IE bit in the TIMx_DIER register, and/or the DMA request by setting the CC1DE bit in theTIMx_DIER register.

When an input capture occurs:

- The TIMx_CCR1 register gets the value of the counter on the active transition.
- CC1IF flag is set (interrupt flag). CC1OF is also set if at least two consecutive captures occurred whereas the flag was not cleared.
- An interrupt is generated depending on the CC1IE bit.
- A DMA request is generated depending on the CC1DE bit.

In order to handle the overcapture, it is recommended to read the data before the overcapture flag. This is to avoid missing an overcapture which could happen after reading the flag and before reading the data.

Note:IC interrupt and/or DMA requests can be generated by software by setting the corresponding CCxG bit in the TIMx_EGR register.

20.3.6. PWM input mode

This mode is a particular case of input capture mode. The procedure is the same except:

- Two ICx signals are mapped on the same TIx input.
- These 2 ICx signals are active on edges with opposite polarity.
- One of the two TIxFP signals is selected as trigger input and the slave mode controller is configured in reset mode.

For example, user can measure the period (in TIMx_CCR1 register) and the duty cycle (in TIMx_CCR2 register) of the PWM applied on TI1 using the following procedure (depending on CK_INT frequency and prescaler value):

- Select the active input for TIMx_CCR1:write the CC1S bits to 01 in the TIMx_CCMR1 register (TI1 selected).
- Select the active polarity for TI1FP1 (used both for capture in TIMx_CCR1 and counter clear):write the CC1P bit to ‘0’ (active on rising edge).
- Select the active input for TIMx_CCR2:write the CC2S bits to 10 in the TIMx_CCMR1 register (TI1 selected).
- Select the active polarity for TI1FP2 (used for capture in TIMx_CCR2):write the CC2Pbit to ‘1’ (active on falling edge).
- Select the valid trigger input:write the TS bits to 101 in the TIMx_SMCR register (TI1FP1 selected).
- Configure the slave mode controller in reset mode:write the SMS bits to 100 in the TIMx_SMCR register.
- Enable the captures:write the CC1E and CC2E bits to ‘1’ in the TIMx_CCER register.

Puya Confidential

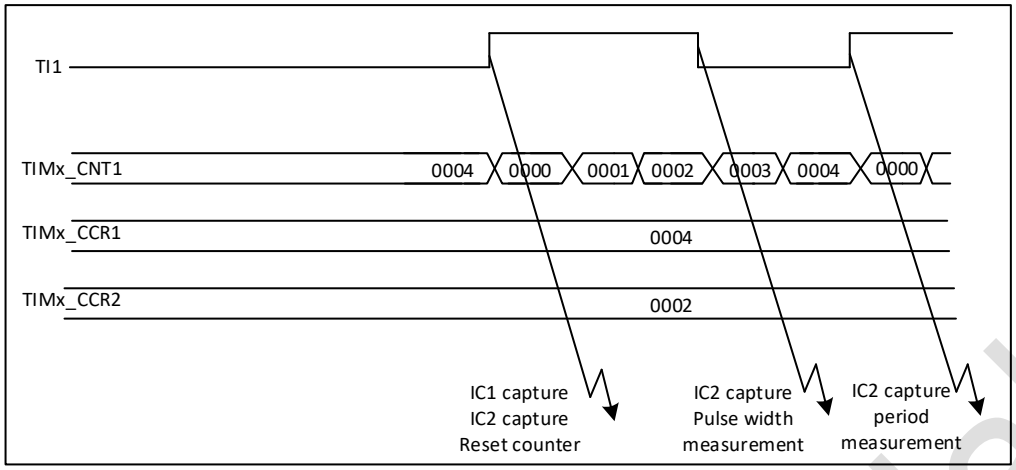


Figure 20-27 PWM input mode timing

20.3.7. Forced output mode

In output mode (CCxS bits = 00 in the TIMx_CCMRx register), each output compares signal (OCxREF and then OCx) can be forced to active or inactive level directly by software, independently of any comparison between the output compare register and the counter. To force an output compare signal (OCXREF/OCx) to its active level, you just need to write 101 in the OCxM bits in the corresponding TIMx_CCMRx register. Thus OCxREF is forced high (OCxREF is always active high) and OCx get opposite value to CCxP polarity bit.

For example:CCxP=0 (OCx active high) => OCx is forced to high level. The OCxREF signal can be forced low by writing the OCxM bits to 100 in the TIMx_CCMRx register.

Anyway, the comparison between the TIMx_CCRx shadow register and the counter is still performed and allows the flag to be set. Interrupt and DMA requests can be sent accordingly. This is described in the output compare mode section below.

20.3.8. Output compare mode

This function is used to control an output waveform or indicating when a period of time has elapsed. When a match is found between the capture/compare register and the counter, the output compare function:

- Assigns the corresponding output pin to a programmable value defined by the output compare mode (OCxM bits in the TIMx_CCMRx register) and the output polarity (CCxP bit in the TIMx_CCER register). The output pin can keep its level (OCXM=000), be set active (OCxM=001), be set inactive (OCxM=010) or can toggle (OCxM=011) on match.
- Sets a flag in the interrupt status register (CCxIF bit in the TIMx_SR register).
- Generates an interrupt if the corresponding interrupt mask is set (CCXIE bit in the TIMx_DIER register).
- Sends a DMA request if the corresponding enable bit is set (CCxDE bit in the TIMx_DIER register, CCDS bit in the TIMx_CR2 register for the DMA request selection).

The TIMx_CCRx registers can be programmed with or without preload registers using the OCxPE bit in the TIMx_CCMRx register.

Puya Confidential

In output compare mode, the update event UEV has no effect on OCxREF and OCx output. The timing resolution is one count of the counter. Output compare mode can also be used to output a single pulse (in one pulse mode).

Procedure:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data in the TIMx_ARR and TIMx_CCRx registers.
3. Set the CCxIE bit if an interrupt request is to be generated.
4. Select the output mode. For example:
   - Write OCxM = 011 to toggle OCx output pin when CNT matches CCRx
   - Write OCxPE = 0 to disable preload register
   - Write CCxP = 0 to select active high polarity
   - Write CCxE = 1 to enable the output
5. Enable the counter by setting the CEN bit in the TIMx_CR1 register.

The TIMx_CCRx register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled (OCxPE= ’0’, else TIMx_CCRx shadow register is updated only at the next update event UEV). An example is given in the figure below.

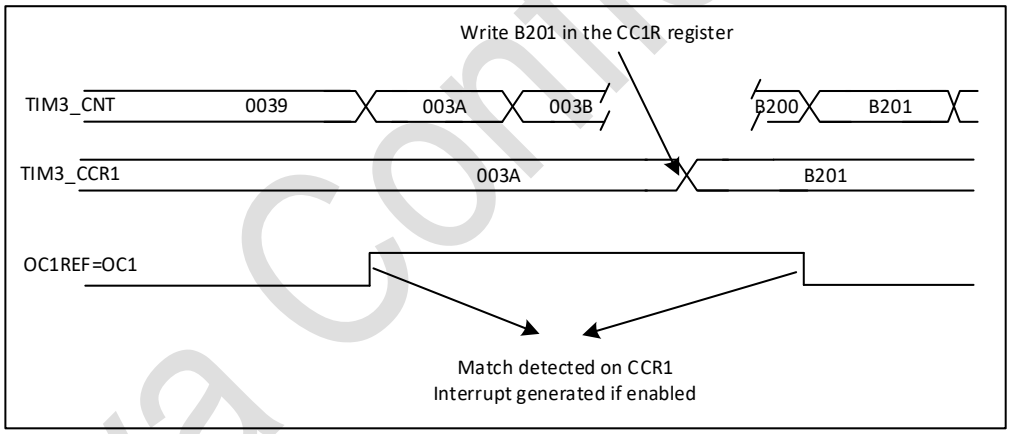


Figure 20-28 Output compare mode, toggle on OC1

20.3.9. PWM mode

Pulse width modulation (PWM) mode allows you to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register.

The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing ‘110’ (PWM mode 1) or ‘111’ (PWM mode 2) in the OCxM bits in the TIMx_CCMRx register. You must enable the corresponding preload register by setting the OCxPE bit in the TIMx_CCMRx register, and eventually the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIMx_CR1 register.

Puya Confidential

As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, all registers must be initialized by setting the UG bit in the TIMx_EGR register.

OCx polarity is software programmable using the CCxP bit in the TIMx_CCER register. It can be programmed as active high or active low. OCx output is enabled by a combination of the CCxE, CCxNE, MOE, OSSI and OSSR bits (TIMx_CCER and TIMx_BDTR registers). Refer to the TIMx_CCER register description for more details.

In PWM mode (1 or 2), TIMx_CNT and TIMx_CCRx are always compared to determine whether TIMx_CCRx ≤ TIMx_CNT or TIMx_CNT ≤ TIMx_CCRx (depending on the direction of the counter).

The timer is able to generate PWM in edge-aligned mode or center-aligned mode depending on the CMS bits in the TIMx_CR1 register.

PWM edge-aligned mode

- Upcounting configuration

Upcounting is active when the DIR bit in the TIMx_CR1 register is low. In the following example, we consider PWM mode 1. The reference PWM signal OCxREF is high as long as TIMx_CNT < TIMx_CCRx else it becomes low. If the compare value in TIMx_CCRx is greater than the auto-reload value (in TIMx_ARR) then OCxREF is held at ‘1’. If the compare value is 0 then OCxRef is held at ‘0’. Figure below shows some edge-aligned PWM waveforms in an example where TIMx_ARR=8.

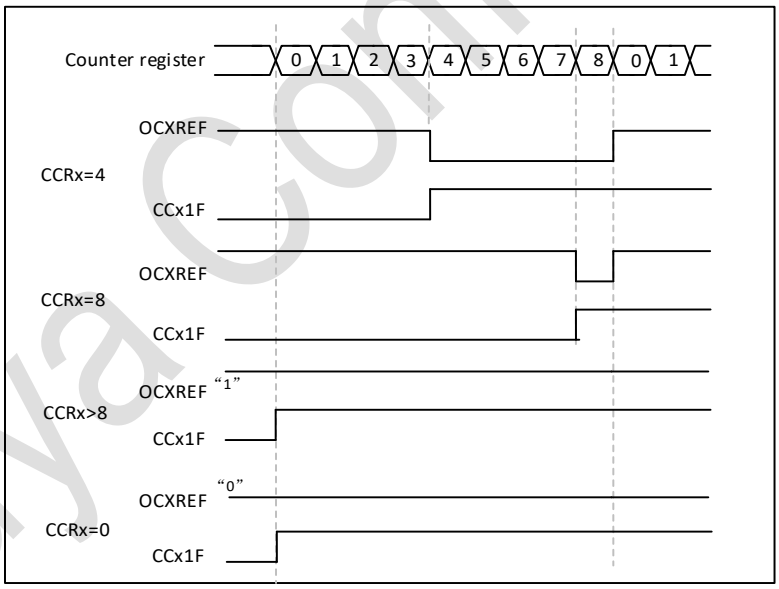


Figure 20-29 Edge-aligned PWM waveforms (ARR = 8)

Downcounting configuration

Downcounting is active when DIR bit in TIMx_CR1 register is high.

In PWM mode 1, the reference signal OCxRef is low as long asTIMx_CNT > TIMx_CCRx else it becomes high. If the compare value in TIMx_CCRx is greater than the auto-reload value in TIMx_ARR, then OCxREF is held at ‘1’. 0% PWM is not possible in this mode.

Puya Confidential

PWM center-aligned mode

Center-aligned mode is active when the CMS bits in TIMx_CR1 register are different from '00' (all the remaining configurations having the same effect on the OCxRef/OCx signals). The compare flag is set when the counter counts up, when it counts down or both when it counts up and down depending on the CMS bits configuration. The direction bit (DIR) in the TIMx_CR1 register is updated by hardware and must not be changed by software.

Figure below shows some center-aligned PWM waveforms in an example where:

- TIMx_ARR = 8
- PWM mode is the PWM mode 1
- The flag is set when the counter counts down corresponding to the center-aligned mode 1 selected for CMS=01 in TIMx_CR1 register.

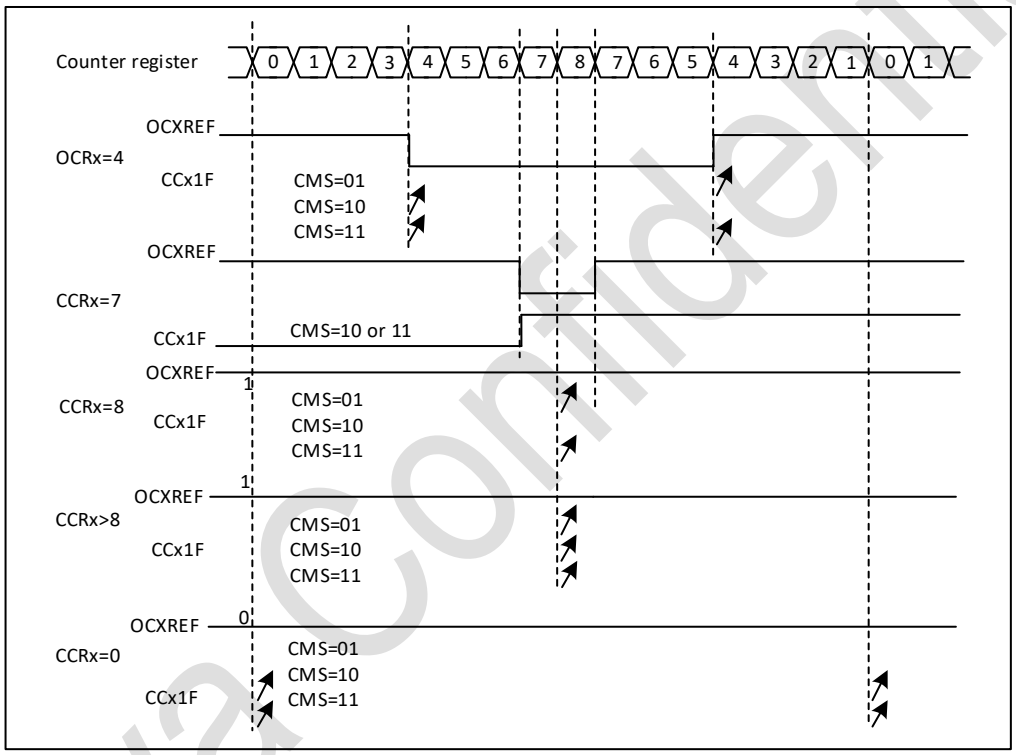


Figure 20-30 Center-aligned PWM waveforms (ARR=8)

Hints on using center-aligned mode:

- When starting in center-aligned mode, the current up-down configuration is used. It means that the counter counts up or down depending on the value written in the DIR bit in the TIMx_CR1 register. Moreover, the DIR and CMS bits must not be changed at the same time by the software.
- Writing to the counter while running in center-aligned mode is not recommended as it can lead to unexpected results. In particular:

The direction is not updated if you write a value in the counter that is greater than the auto-reload value (TIMx_CNT>TIMx_ARR). For example, if the counter was counting up, it continues to count up.

The direction is updated if you write 0 or write the TIMx_ARR value in the counter but no update event (UEV) is generated.

Puya Confidential

- The safest way to use center-aligned mode is to generate an update by software (setting the UG bit in the TIMx_EGR register) just before starting the counter and not to write the counter while it is running.

20.3.10. One-pulse mode

One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. One-pulse mode is selected by setting the OPM bit in the TIMx_CR1 register. This makes the counter stop automatically at the next update event UEV.

A pulse can be correctly generated only if the compare value is different from the counter initial value. Before starting (when the timer is waiting for the trigger), the configuration must be:

- In upcounting:CNT < CCRx ≤ ARR (in particular, 0 < CCRx)
- In downcounting:CNT > CCRx

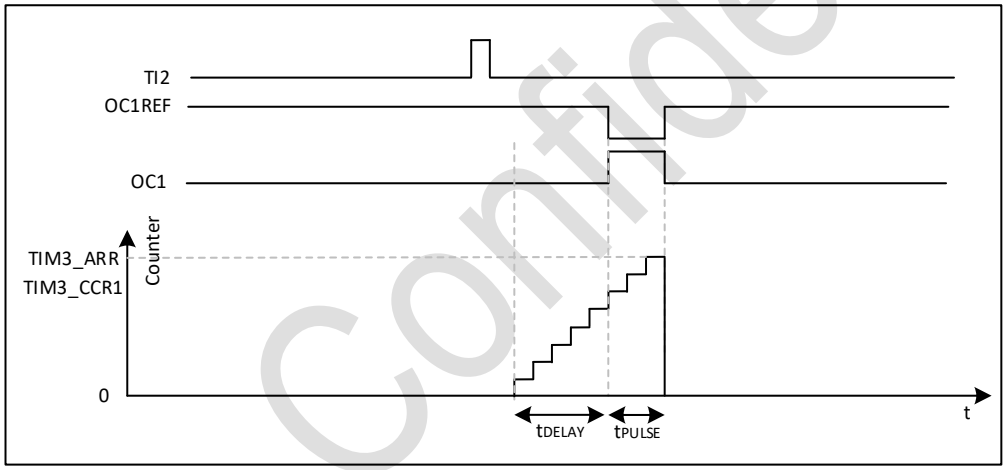


Figure 20-31 Example of one pulse mode

For example, one may want to generate a positive pulse on OC1 with a length of t_{PULSE} and after a delay of t_{DELAY} as soon as a positive edge is detected on the TI2 input pin.

Let's use TI2FP2 as trigger 1:

- Map TI2FP2 on TI2 by writing CC2S=01 in the TIMx_CCMR1 register.
- TI2FP2 must detect a rising edge, write CC2P=0 in the TIMx_CCER register.
- Configure TI2FP2 as trigger for the slave mode controller (TRGI) by writing TS=110 in the TIMx_SMCR register.
- TI2FP2 is used to start the counter by writing SMS to '110' in the TIMx_SMCR register (trigger mode).

The OPM waveform is defined by writing the compare registers (taking into account the clock frequency and the counter prescaler).

- The t_{DELAY} is defined by the value written in the TIMx_CCR1 register.
- The t_{PULSE} is defined by the difference between the auto-reload value and the compare value (TIMx_ARR - TIMx_CCR1).

Puya Confidential

- Let's say one want to build a waveform with a transition from '0' to '1' when a compare match occurs and a transition from '1' to '0' when the counter reaches the auto-reload value. To do this PWM mode 2 must be enabled by writing OC1M=111 in the TIMx_CCMR1 register. Optionally the preload registers can be enabled by writing OC1PE='1' in the TIMx_CCMR1 register and ARPE in the TIMx_CR1 register. In this case one has to write the compare value in the TIMx_CCR1 register, the auto-reload value in the TIMx_ARR register, generate an update by setting the UG bit and wait for external trigger event on TI2. CC1P is written to '0' in this example.

In our example, the DIR and CMS bits in the TIMx_CR1 register should be low.

The user only wants one pulse, so '1' must be written in the OPM bit in theTIMx_CR1 register to stop the counter at the next update event (when the counter rolls over from the auto-reload value back to 0).

Particular case:OCx fast enable:

In One-pulse mode, the edge detection on TIx input set the CEN bit which enables the counter. Then the comparison between the counter and the compare value makes the output toggle. But several clock cycles are needed for these operations, and it limits the minimum delay tDELAY min we can get.

If one wants to output a waveform with the minimum delay, the OCxFE bit can be set in the TIMx_CCMRx register. Then OCxREF (and OCx) is forced in response to the stimulus, without taking in account the comparison. Its new level is the same as if a compare match had occurred. OCxFE acts only if the channel is configured in PWM1 or PWM2 mode.

20.3.11. Encoder interface mode

To select encoder interface mode:write SMS='001' in the TIMx_SMCR register if the counter is counting on TI2 edges only, SMS=010 if it is counting on TI1 edges only and SMS=011 if it is counting on both TI1 and TI2 edges.

Select the TI1 and TI2 polarity by programming the CC1P and CC2P bits in the TIMx_CCER register. When needed, you can program the input filter as well.

The two inputs TI1 and TI2 are used to interface to an incremental encoder. The counter is clocked by each valid transition on TI1FP1 or TI2FP2 (TI1 and TI2 after input filter and polarity selection, TI1FP1=TI1 if not filtered and not inverted, TI2FP2=TI2 if not filtered and not inverted) assuming that it is enabled (CEN bit inTIMx_CR1 register written to '1'). The sequence of transitions of the two inputs is evaluated and generates count pulses as well as the direction signal. Depending on the sequence the counter counts up or down, the DIR bit in the TIMx_CR1 register is modified by hardware accordingly. The DIR bit is calculated at each transition on any input (TI1 or TI2), whatever the counter is counting on TI1 only, TI2 only or both TI1 and TI2.

Encoder interface mode acts simply as an external clock with direction selection. This means that the counter just counts continuously between 0 and the auto-reload value in the TIMx_ARR register (0 to ARR or ARR down to 0 depending on the direction). So the TIMx_ARR must be configured before starting. In the same way, the capture, compare, prescaler, trigger output features continue to work as normal. Encoder mode and External clock mode 2 are not compatible and must not be selected together. In this mode, the counter is modified automatically following the speed and the direction of the

Puya Confidential

quadrature encoder and its content, therefore, always represents the encoder's position. The count direction corresponds to the rotation direction of the connected sensor. The table summarizes the possible combinations, assuming TI1 and TI2 do not switch at the same time.

Table 20-1 Counting direction versus encoder signals

Active edge	Level on opposite signal (TI1FP1 forTI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	Not count	Not count
	Low	Up	Down	Not count	Not count
Counting on TI2 only	High	Not count	Not count	Up	Down
	Low	Not count	Not count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

An external incremental encoder can be connected directly to the MCU without external interface logic. However, comparators are normally used to convert the encoder's differential outputs to digital signals. This greatly increases noise immunity. The third encoder output which indicate the mechanical zero position, may be connected to an external interrupt input and trigger a counter reset.

The figure below gives an example of counter operation, showing count signal generation and direction control. It also shows how input jitter is compensated where both edges are selected. This might occur if the sensor is positioned near to one of the switching points. For this example, we assume that the configuration is the following:

- CC1S='01' (TIMx_CCMR1 register, TI1FP1 mapped on IC1).
- CC2S='01' (TIMx_CCMR2 register, TI2FP2 mapped on IC2).
- CC1P = '0' (TIMx_CCER register, TI1FP1 is not inverted, TI1FP1 = TI1)
- CC2P = '0' (TIMx_CCER register, TI2FP2 is not inverted, TI2FP2 = TI2)
- SMS= 011 (TIMx_SMCR register, both inputs are active on both rising and falling edges)
- CEN='1' (TIMx_CR1 register, counter enabled)

Puya Confidential

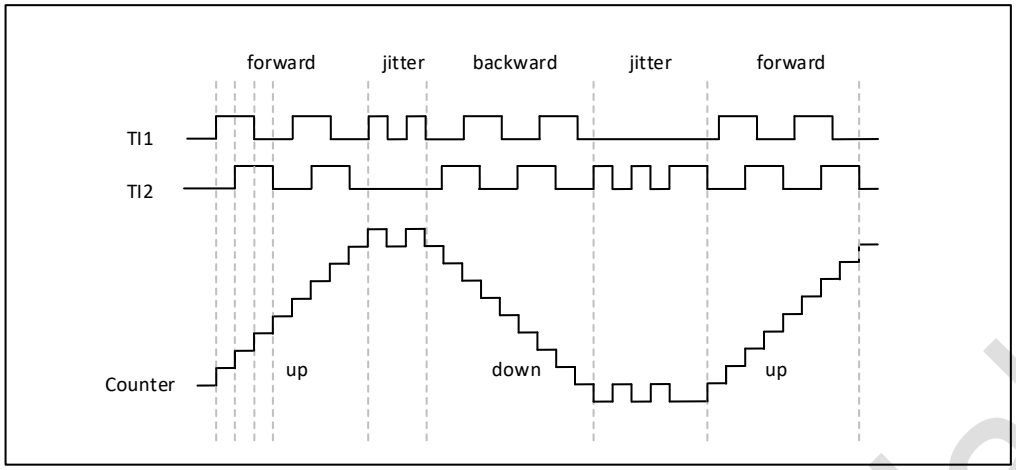


Figure 20-32 Example of counter operation in encoder interface mode

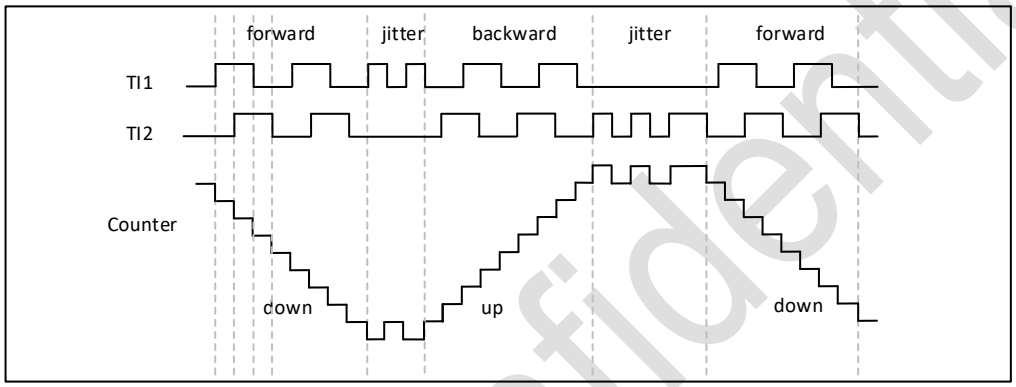


Figure 20-33 Example of encoder interface mode with IC1FP1 polarity inverted

The timer, when configured in encoder interface mode provides information on the sensor's current position. Dynamic information can be obtained (speed, acceleration, deceleration) by measuring the period between two encoder events using a second timer configured in capture mode. The output of the encoder which indicates the mechanical zero can be used for this purpose. Depending on the time between two events, the counter can also be read at regular times. This can be done by latching the counter value into a third input capture register if available (then the capture signal must be periodic and can be generated by another timer). when available, it is also possible to read its value through a DMA request generated by a real-time clock.

20.3.12. Timer input XOR function

The TI1S bit in the TIMx_CR2 register, allows the input filter of channel 1 to be connected to the output of a XOR gate, combining the three input pins TIMx_CH1, TIMx_CH2 and TIMx_CH3.

The XOR output can be used with all the timer input functions such as trigger or input capture.

An example of this feature is used to interface Hall sensors.

20.3.13. Timers and external trigger synchronization

The TIMx timer can be synchronized with an external trigger in several modes:Reset mode, Gated mode and Trigger mode.

Puya Confidential

Slave mode:Reset mode

The counter and its prescaler can be reinitialized in response to an event on a trigger input. Moreover, if the URS bit from the TIMx_CR1 register is low, an update event UEV is generated. Then all the preloaded registers (TIMx_ARR, TIMx_CCRx) are updated.

In the following example, the upcounter is cleared in response to a rising edge on TI1 input:

- Configure the channel 1 to detect rising edges on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=0 (and CC1NP = 0) in TIMx_CCER register to validate the polarity (and detect rising edges only).
- Configure the timer in reset mode by writing SMS=100 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.
- Enable the counter by writing CEN=1 in the TIMx_CR1 register.

The counter starts counting on the internal clock, then behaves normally until TI1 rising edge. When TI1 rises, the counter is cleared and restarts from 0. In the meantime, the trigger flag is set (TIF bit in the TIMx_SR register) and an interrupt request, or a DMA request can be sent if enabled (depending on the TIE and TDE bits in TIMx_DIER register).

The following figure shows this behavior when the auto-reload register TIMx_ARR=0x36. The delay between the rising edge on TI1 and the actual reset of the counter is due to the resynchronization circuit on TI1 input.

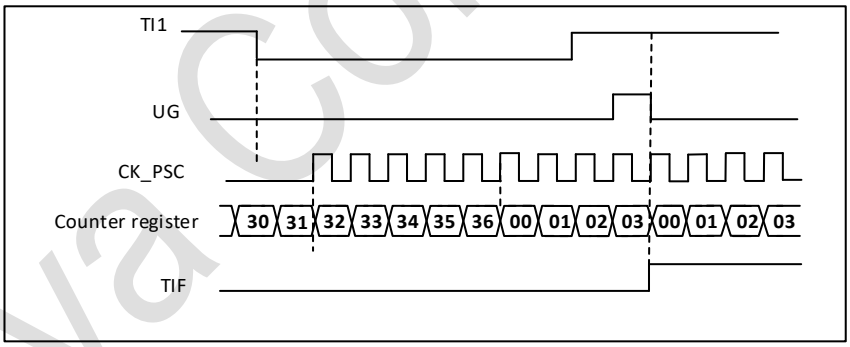


Figure 20-34 Control circuit in reset mode

Slave mode:Gated mode

The counter can be enabled depending on the level of a selected input.

In the following example, the upcounter counts only when TI1 input is low:

- Configure the channel 1 to detect low levels on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=1 in TIMx_CCER register to validate the polarity (and detect low level only).
- Configure the timer in gated mode by writing SMS=101 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.

Puya Confidential

- Enable the counter by writing CEN=1 in the TIMx_CR1 register. In gated mode, the counter doesn't start if CEN=0, whatever is the trigger input level.

The counter starts counting on the internal clock as long as TI1 is low and stops as soon as TI1 becomes high. The TIF flag in the TIMx_SR register is set both when the counter starts or stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

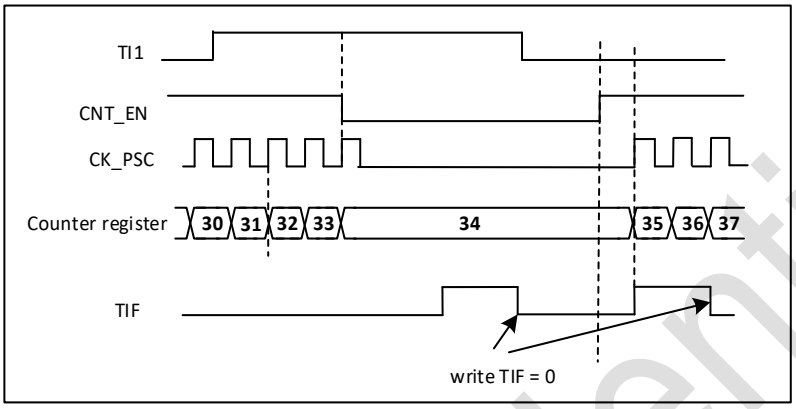


Figure 20-35 Control circuit in Gated mode

Slave mode:Trigger mode

The selected event on the input enables the counter.

In the following example, the upcounter starts in response to a rising edge on TI2 input:

a) Configure the channel 2 to detect rising edges on TI2. Configure the input filter duration (in this example, we do not need any filter, so we keep IC2F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC2S bits are configured to select the input capture source only, CC2S=01 in TIMx_CCMR1 register. Write CC2P=1 and CC2NP = 0 in TIMx_CCER register to validate the polarity (and detect low level only).

b) Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI2 as the input source by writing TS=110 in TIMx_SMCR register.

When a rising edge occurs on TI2, the counter starts counting on the internal clock and the TIF flag is set. The delay between the rising edge on TI2 and the actual start of the counter is due to the resynchronization circuit on TI2 input.

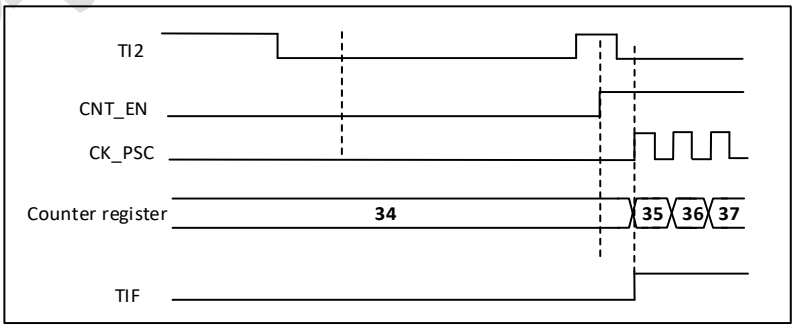


Figure 20-36 Control circuit in trigger mode

Puya Confidential

Slave mode:external clock mode 2 + trigger mode

The external clock mode 2 can be used in addition to another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as external clock input, and another input can be selected as trigger input when operating in reset mode, gated mode or trigger mode. It is recommended not to select ETR as TRGI through the TS bits of TIMx_SMCR register.

In the following example, the upcounter is incremented at each rising edge of the ETR signal as soon as a rising edge of TI1 occurs:

- Configure the external trigger input circuit by programming the TIMx_SMCR register as follows:
  - ETF = 0000:no filter
  - ETPS = 00:prescaler disabled
  - ETP = 0:detection of rising edges on ETR and ECE=1 to enable the external clock mode 2.
- Configure the channel 1 as follows, to detect rising edges on TI:
  - IC1F = 0000:no filter
  - The capture prescaler is not used for triggering, so it does not need to be configured.
  - CC1S=01 in TIMx_CCMR1 register to select only the input capture source

Write CC1P=0 in TIMx_CCER register to validate the polarity (and detect rising edges only).

- Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.

A rising edge on TI1 enables the counter and sets the TIF flag. The counter then counts on ETR rising edges. The delay between the rising edge of the ETR signal and the actual reset of the counter is due to the resynchronization circuit on ETRP input.

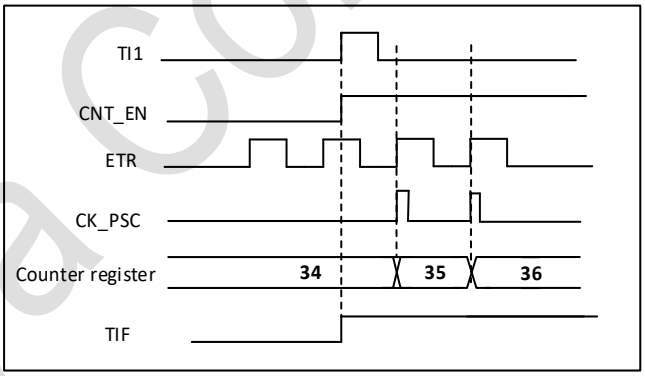


Figure 20-37 Control circuit in external clock mode 2 + trigger mode

20.3.14. Timer synchronization

The TIMx timers are linked together internally for timer synchronization or chaining. When one Timer is configured in Master mode, it can reset, start, stop or clock the counter of another Timer configured in Slave mode.

The following presents an overview of the trigger selection and the master mode selection blocks.

Puya Confidential

Using one timer as prescaler for another timer

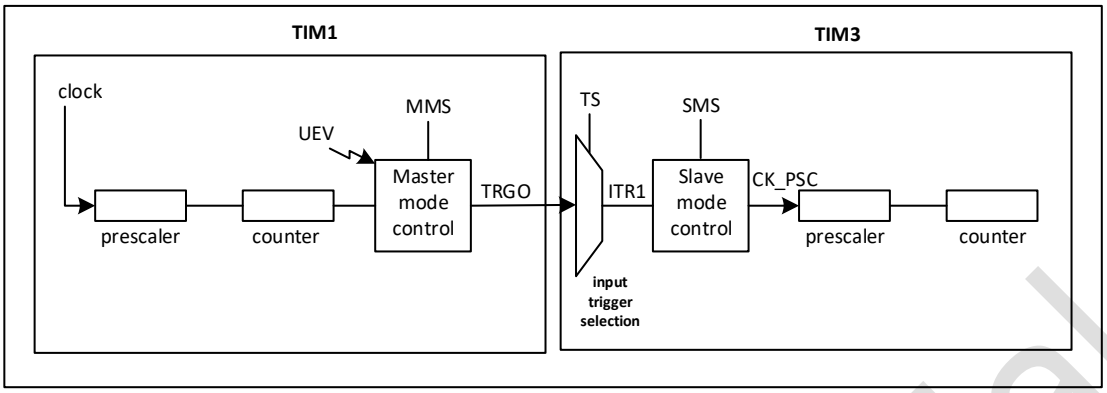


Figure 20-38 Master/Slave timer example

For example, the user can configure Timer 1 to act as a prescaler for Timer 3. To do this:

- Configure Timer 1 in master mode so that it outputs a periodic trigger signal on each update event UEV. If you write MMS=010 in the TIM1_CR2 register, a rising edge is output on TRGO each time an update event is generated.
- To connect the TRGO output of Timer 1 to Timer 3, Timer 3 must be configured in slave mode using ITR0 as internal trigger. You select this through the TS bits in the TIM3_SMCR register (writing TS=000).
- Then you put the slave mode controller in external clock mode 1 (write SMS=111 in the TIM3_SMCR register). This causes Timer 3 to be clocked by the rising edge of the periodic Timer 1 trigger signal (which correspond to the timer 1 counter overflow).
- Finally, both timers must be enabled by setting their respective CEN bits (TIMx_CR1 register) to ensure starting Timer 3 first and then Timer 1.

Note:If OCx is selected on Timer 1 as trigger output (MMS=1xx), its rising edge is used to clock the counter of timer 3.

Using one timer to enable another timer

In this example, we control the enable of Timer 3 with the output compare 1 of Timer 1. Refer to the diagram above for connections. Timer 3 counts on the divided internal clock only when OC1REF of Timer 1 is high. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_INT (fCK_CNT = fCK_INT/3).

- Configure Timer 1 master mode to send its Output compare 1 reference (OC1REF) signal as trigger output (MMS=100 in the TIM1_CR2 register).
- Configure the Timer 1 OC1REF waveform (TIM1_CCMR1 register).
- Configure Timer 3 to get the input trigger from Timer 1 (TS=000 in the TIM3_SMCR register).
- Configure Timer 3 in gated mode (SMS=101 in TIM3_SMCR register).
- Enable Timer 3 by writing ‘1 in the CEN bit (TIM3_CR1 register).
- Start Timer 1 by writing ‘1 in the CEN bit (TIM1_CR1 register).

Note:The counter 3 clock is not synchronized with counter 1, this mode only affects the Timer 3 counter enable signal.

Puya Confidential

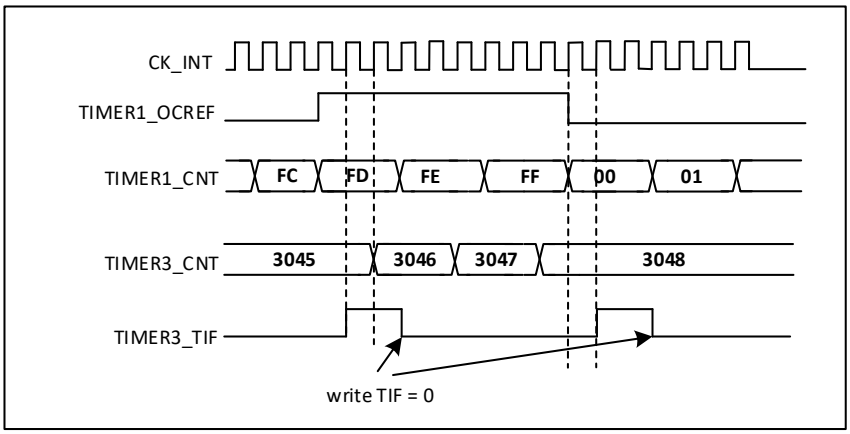


Figure 20-39 Gating timer 3 with OC1REF of timer 1

In the example of the figure above, the Timer 3 counter and prescaler are not initialized before being started. So they start counting from their current value. It is possible to start from a given value by resetting both timers before starting Timer 1. You can then write any value you want in the timer counters. The timers can easily be reset by software using the UG bit in the TIMx_EGR registers.

In the next example, we synchronize Timer 1 and Timer 2. Timer 1 is the master and starts from 0. Timer 2 is the slave and starts from 0xE7. The prescaler ratio is the same for both timers. Timer 2 stops when Timer 1 is disabled by writing 0 to the CEN bit in the TIM1_CR1 register.

- Configure Timer 1 master mode to send its enable (CNT_EN) signal as trigger output (MMS=100 in the TIM1_CR2 register).
- Configure the Timer 1 OC1REF waveform (TIM1_CCMR1 register).
- Configure Timer 2 to get the input trigger from Timer 1 (TS=000 in the TIM2_SMCR register).
- Configure Timer 2 in gated mode (SMS=101 in TIM2_SMCR register).
- Reset Timer 1 by writing 1 in UG bit (TIM1_EGR register).
- Reset Timer 2 by writing 1 in UG bit (TIM2_EGR register).
- Initialize Timer 2 to 0xE7 by writing ‘0xE7’ in the timer 2 counter (TIM2_CNT).
- Enable Timer 2 by writing 1 in the CEN bit (TIM2_CR1 register).
- Start Timer 1 by writing 1 in the CEN bit (TIM1_CR1 register).
- Stop Timer 1 by writing 0 in the CEN bit (TIM1_CR1 register).

Puya Confidential

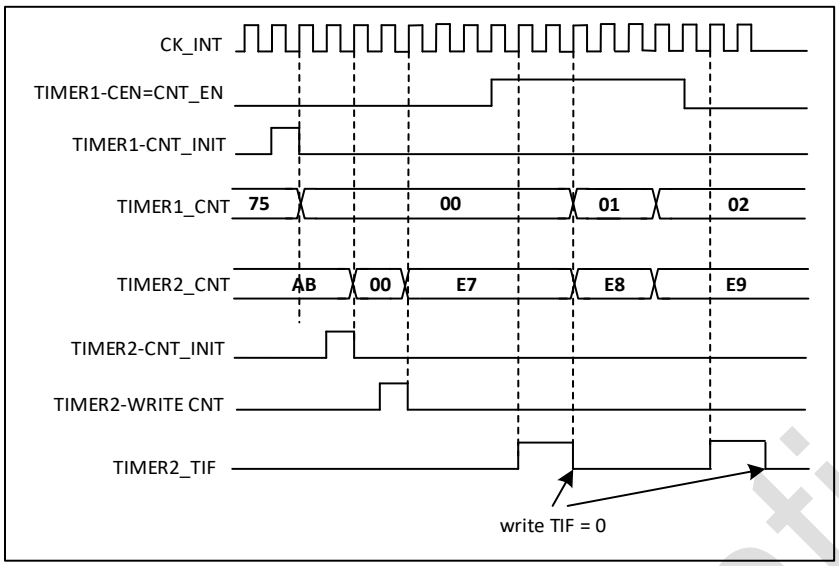


Figure 20-40 Gating timer 2 with enable of timer 1

Using one timer to start another timer

In this example, we set the enable of Timer 3 with the update event of Timer 1. Timer 3 starts counting from its current value (which can be nonzero) on the divided internal clock as soon as the update event is generated by Timer 1. When Timer 3 receives the trigger signal its CEN bit is automatically set and the counter counts until we write 0 to the CEN bit in the TIM3_CR1 register. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_INT ($f_{CK\_CNT} = f_{CK\_INT}/3$).

- Configure Timer 1 master mode to send its update event (UEV) as trigger output (MMS=010 in the TIM1_CR2 register).
- Configure the Timer 1 period (TIM1_ARR registers)
- Configure Timer 3 to get the input trigger from Timer 1 (TS=000 in the TIM3_SMCR register).
- Configure Timer 3 in trigger mode (SMS=110 in TIM3_SMCR register).
- Start Timer 1 by writing ‘1 in the CEN bit (TIM1_CR1 register).

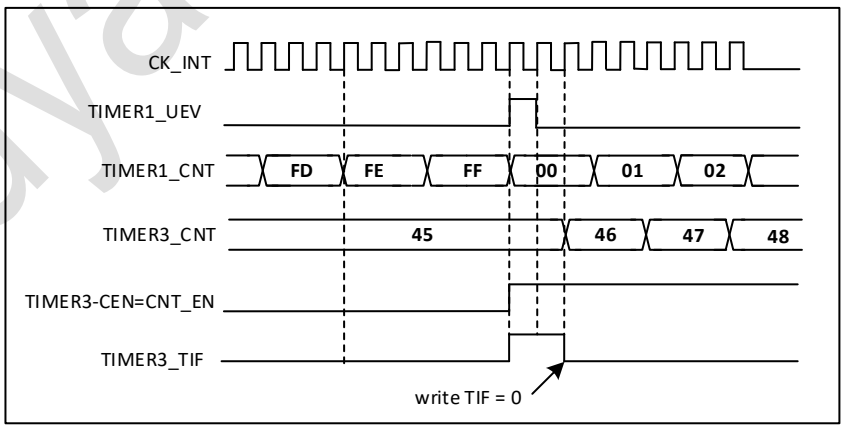


Figure 20-41 Triggering timer 3 with update of timer 1

As in the previous example, the user can initialize both counters before starting counting. The following figure shows the behavior with the same configuration as in Figure above but in trigger mode instead of gated mode (SMS=110 in the TIM3_SMCR register).

Puya Confidential

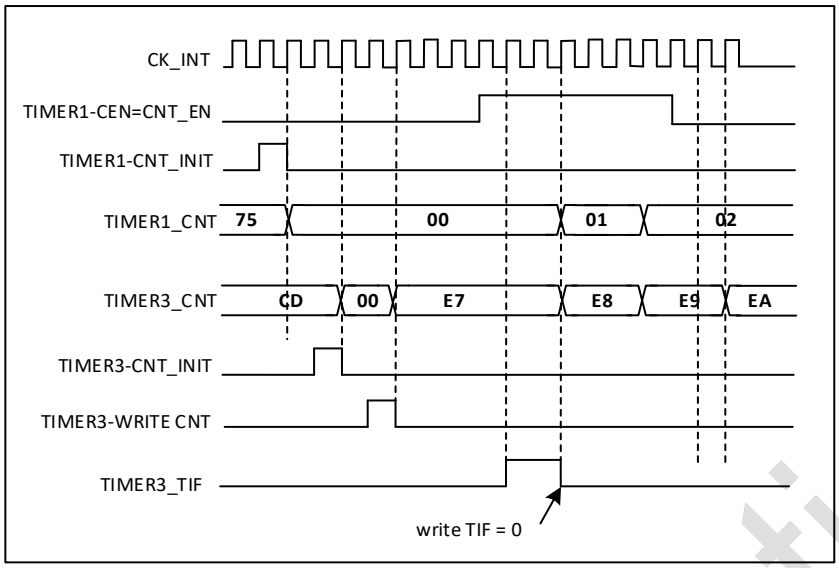


Figure 20-42 Triggering timer 3 with enable of timer 1

Starting 2 timers synchronously in response to an external trigger

In this example, we set the enable of timer 1 when its TI1 input rises, and the enable of Timer 3 with the enable of Timer 1. To ensure the counters are aligned, Timer 1 must be configured in Master/Slave mode (slave with respect to TI1, master with respect to Timer 3):

- Configure Timer 1 master mode to send its Enable as trigger output (MMS=001 in the TIM1_CR2 register).
- Configure Timer 1 slave mode to get the input trigger from TI1 (TS=100 in the TIM1_SMCR register).
- Configure Timer 1 in trigger mode (SMS=110 in the TIM1_SMCR register).
- Configure the Timer 1 in Master/Slave mode by writing MSM=1 (TIM1_SMCR register).
- Configure Timer 3 to get the input trigger from Timer 1 (TS=000 in the TIM3_SMCR register).
- Configure Timer 3 in trigger mode (SMS=110 in the TIM3_SMCR register).

When a rising edge occurs on TI1 (Timer 1), both counters starts counting synchronously on the internal clock and both TIF flags are set.

Note:In this example both timers are initialized before starting (by setting their respective UG bits). Both counters start from 0, but you can easily insert an offset between them by writing any of the counter registers (TIMx_CNT). You can see that the master/slave mode insert a delay between CNT_EN and CK_PSC on timer 1.

20.3.15. Debug mode

When the microcontroller enters debug mode, the TIMx counter either continues to work normally or stops, depending on DBG_TIMx_STOP configuration bit in DBG module.

20.4. TIMx registers

TIM2 register base address:0x4000 0000

TIM3 register base address:0x4000 0400

Puya Confidential

20.4.1. TIM2/3 control register 1 (TIMx_CR1)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
-	-	-	-	-	-	RW		RW	RW		RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9:8	CKD [1:0]	RW	00	Clock division This bit-field indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock (t_{DTS}) used by the dead-time generators and the digital filters (ETR, TIx) 00:t_{DTS} = t_{CK_INT} 01:t_{DTS} = 2 x t_{CK_INT} 10:t_{DTS} = 4 x t_{CK_INT} 11:Reserved, do not program this value
7	ARPE	RW	0	Auto-reload preload enable 0:TIMx_ARR register is not buffered 1:TIMx_ARR register is buffered
6:5	CMS [1:0]	RW	00	Center-aligned mode selection 00:Edge-aligned mode. The counter counts up or down depending on the direction bit (DIR). 01:Center-aligned mode 1. The counter counts up and down alternatively. Configured as output channel Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set only when the counter is counting down. 10:Center-aligned mode 2. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set only when the counter is counting up.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:Center-aligned mode 3. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set both when the counter is counting up or down. Note:It is not allowed to switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN=1).
4	DIR	RW	0	Direction 0:Counter used as upcounter 1:Counter used as downcounter Note:This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.
3	OPM	RW	0	One-pulse mode 0:Counter is not stopped at update event 1:Counter stops counting at the next update event (clearing the bit CEN)
2	URS	RW	0	Update request source This bit is set and cleared by software to select the UEV event sources. 0:Any of the following events generate an update interrupt or DMA request if enabled. These events can be: – Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller 1:Only counter overflow/underflow generates an update interrupt or DMA request if enabled
1	UDIS	RW	0	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0:UEV enabled. The update event (UEV) is generated by one of the following events: – Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1:UEV disabled. The update event is not generated, shadow registers keep their value (ARR, PSC, CCRx).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				However, the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	RW	0	Counter enable 0:Counter disabled 1:Counter enabled Note:external clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However, trigger mode can set the CEN bit automatically by hardware.

20.4.2. TIM2/3 control register 2 (TIMx_CR2)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	TI1S	MMS [2:0]			CCDS	Res	Res	Res
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	-	-	-

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7	TI1S	RW	0	TI1 selection 0:The TIMx_CH1 pin is connected to TI1 input 1:The TIMx_CH1, CH2 and CH3 pins are connected to the TI1 input.
6:4	MMS [2:0]	RW	000	Master mode selection These two bits are used to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000:Reset - the UG bit from the TIMx_EGR register is used as trigger output (TRGO). If the reset is generated by the trigger input (slave mode controller configured in reset mode) then the signal on TRGO is delayed compared to the actual reset. 001:Enable - the counter enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				is enabled. The counter enable signal is generated by a logic AND between CEN control bit and the trigger input when configured in gated mode. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected (see the MSM bit description in TIMx_SMCR register). 010:Update - The update event is selected as trigger output (TRGO). For instance, a master timer can then be used as a prescaler for a slave timer. 011:Compare pulse - The trigger output sends a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or a compare match occurred (TRGO). 100:Compare - OC1REF signal is used as trigger output (TRGO) 101:Compare - OC2REF signal is used as trigger output (TRGO) 110:Compare - OC3REF signal is used as trigger output (TRGO) 111:Compare - OC4REF signal is used as trigger output (TRGO)
3	CCDS	RW	0	Capture/compare DMA selection 0:CCx DMA request sent when CCx event occurs 1:CCx DMA requests sent when update event occurs
2:0	Reserved	-	-	Reserved

20.4.3. TIM2/3 slave mode control register (TIMx_SMCR)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETP	ECE	ETPS [1:0]		ETF [3:0]				MSM	TS [2:0]			OCCS	SMS [2:0]		
RW	RW	RW		RW				RW	RW			RW	RW		

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	ETP	RW	0	External trigger polarity

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit selects whether ETR or the inversion of the ETR is used for trigger operations 0:ETR is non-inverted, active at high level or rising edge; 1:ETR is inverted, active at low level or falling edge
14	ECE	RW	0	External clock enable This bit enables external clock mode 2 0:External clock mode 2 disabled; 1:External clock mode 2 enabled. The counter is driven by any active edge on the ETRF signal. Note:1:Setting the ECE bit has the same effect as selecting external clock mode 1 with TRGI connected to ETRF (SMS=111 and TS=111). Note 2:It is possible to simultaneously use external clock mode 2 with the following slave modes:reset mode, gated mode and trigger mode. Nevertheless, TRGI must not be connected to ETRF in this case (TS bits must not be 111). Note 3:If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.
13:12	ETPS [1:0]	RW	0	External trigger prescaler External trigger signal ETRP frequency must be at most 1/4 of CK_INT frequency. A prescaler can be enabled to reduce ETRP frequency. 00:Prescaler OFF 01:ETRP frequency divided by 2 10:ETRP frequency divided by 4 11:ETRP frequency divided by 8
11:8	ETF [3:0]	RW	0	External trigger filter This bit-field then defines the frequency used to sample ETRP signal and the length of the digital filter applied to ETRP. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output: 0000:No filter, sampling is done at f_{DTS} 0001:$f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010:$f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011:$f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100:$f_{SAMPLING} = f_{DTS}/2$, N = 6 0101:$f_{SAMPLING} = f_{DTS}/2$, N = 8 0110:$f_{SAMPLING} = f_{DTS}/4$, N = 6 0111:$f_{SAMPLING} = f_{DTS}/4$, N = 8 1000:$f_{SAMPLING} = f_{DTS}/8$, N = 6

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1001:$f_{SAMPLING}$ = f_{DTS}/8, N = 8 1010:$f_{SAMPLING}$ = f_{DTS}/16, N = 5 1011:$f_{SAMPLING}$ = f_{DTS}/16, N = 6 1100:$f_{SAMPLING}$ = f_{DTS}/16, N = 8 1101:$f_{SAMPLING}$ = f_{DTS}/32, N = 5 1110:$f_{SAMPLING}$ = f_{DTS}/32, N = 6 1111:$f_{SAMPLING}$ = f_{DTS}/32, N = 8
7	MSM	RW	0	Master/slave mode 0:No action; 1:The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.
6:4	TS	RW	0	Trigger selection These bit-fields select the trigger input to be used to synchronize the counter. 000:Internal Trigger 0 (ITR0) 100:TI1 Edge detector (TI1F_ED) 001:Internal trigger 1 (ITR1) 101:Filtered timer input 1 (TI1FP1) 010:Internal trigger 2 (ITR2) 110:Filtered timer input 2 (TI2FP2) 011:Internal trigger 3 (ITR3) 111:External trigger input (ETRF) Note:These bits must be changed only when they are not used (e.g. when SMS=000) to avoid wrong edge detections at the transition.
3	OCCS	RW	0	OCREF clear selection This bit is used to select the OCREF clear source. 0:OCREF_CLR_INPUT is connected to OCREF_CLR input 1:OCREF_CLR_INPUT is connected to ETRF
2:0	SMS	RW	0	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input (see input control register and control register description). 000:Slave mode disabled - if CEN = 1 then the prescaler is clocked directly by the internal clock. 001:Encoder mode 1 - Counter counts up/down on TI2FP2 edge depending on TI1FP1 level. 010:Encoder mode 2 - Counter counts up/down on TI1FP1 edge depending on TI2FP2 level. 011:Encoder mode 3 - Counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of the other input.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				100:Reset mode - Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101:Gated mode - The counter clock is enabled when the trigger input (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop of the counter are controlled. 110:Trigger mode - The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only the start of the counter is controlled. 111:External clock mode 1 - Rising edges of the selected trigger (TRGI) clock the counter. Note:The gated mode must not be used if TI1F_EN is selected as the trigger input (TS=100). Indeed, TI1F_ED outputs 1 pulse for each transition on TI1F, whereas the gated mode checks the level of the trigger signal.

Table 20-2 TIMx internal trigger connection

Slave timer	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM15	TIM3	TIM14_OC
TIM3	TIM1	TIM2	TIM15	TIM14_OC

20.4.4. TIM2/3 DMA/interrupt enable register (TIMx_DIER)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	TDE	Res	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res	TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE
-	RW	-	RW					-	RW	-	RW				

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	TDE	RW	0	TDE:Trigger DMA request enable 0:Trigger DMA request disabled 1:Trigger DMA request enabled
13	Reserved	-	-	Reserved
12	CC4DE	RW	0	CC4DE:Capture/Compare 4 DMA request enable 0:CC4 DMA request disabled 1:CC4 DMA request enabled

Puya Confidential

Bit	Name	R/W	Reset Value	Function
11	CC3DE	RW	0	CC3DE:Capture/Compare 3 DMA request enable 0:CC3 DMA request disabled 1:CC3 DMA request enabled
10	CC2DE	RW	0	CC2DE:Capture/Compare 2 DMA request enable 0:CC2 DMA request disabled 1:CC2 DMA request enabled
9	CC1DE	RW	0	CC1DE:Capture/Compare 1 DMA request enable 0:CC1 DMA request disabled 1:CC1 DMA request enabled
8	UDE	RW	0	UDE:Update DMA request enable 0:Update DMA request disabled 1:Update DMA request enabled
7	Reserved	-	-	Reserved
6	TIE	RW	0	TIE:Trigger interrupt enable 0:Trigger interrupt disabled. 1:Trigger interrupt enabled
5	Reserved	-	-	Reserved
4	CC4IE	RW	0	CC4IE:Capture/Compare 4 interrupt enable 0:Capture/Compare 4 interrupt disabled 1:Capture/Compare 4 interrupt enabled
3	CC3IE	RW	0	CC3IE:Capture/Compare 3 interrupt enable 0:Capture/Compare 3 interrupt disabled 1:Capture/Compare 3 interrupt enabled
2	CC2IE	RW	0	CC2IE:Capture/Compare 2 interrupt enable 0:Capture/Compare 2 interrupt disabled 1:Capture/Compare 2 interrupt enabled
1	CC1IE	RW	0	CC1IE:Capture/Compare 1 interrupt enable 0:Capture/Compare 1 interrupt disabled 1:Capture/Compare 1 interrupt enabled
0	UIE	RW	0	UIE:Update interrupt enable 0:Update interrupt disabled. 1:Update interrupt enabled

20.4.5. TIM2/3 status register (TIMx_SR)

Address offset:0x10

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res			Res	Res	Res	Res	Res	IC4IF	IC3IF	IC2IF	IC1IF	IC4IR	IC3IR	IC2IR	IC1IR

Puya Confidential

-			-	-	-	-	-	RC_W0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res			CC4OF	CC3OF	CC2OF	CC1OF	Res		TIF	Res	CC4IF	CC3IF	CC2IF	CC1F	UIF
-			RC_W0				-		RC_W0	-	RC_W0				

Bit	Name	R/W	Reset Value	Function
31:24	Reserved	-	-	Reserved
23	IC4IF	RC_W0	0	Falling edge capture 4 flag Refer to IC1IF description
22	IC3IF	RC_W0	0	Falling edge capture 3 flag Refer to IC1IF description
21	IC2IF	RC_W0	0	Falling edge capture 2 flag Refer to IC1IF description
20	IC1IF	RC_W0	0	Falling edge capture 1 flag This flag is set by hardware only when the corresponding channel is configured in input capture mode and triggered by falling edge. It is cleared by software or reading TIMx_CCR1. 0:No falling edge capture occurs; 1:A falling edge capture event occurs.
19	IC4IR	RC_W0	0	Rising edge capture 4 flag Refer to IC1IR description
18	IC3IR	RC_W0	0	Rising edge capture 3 flag Refer to IC1IR description
17	IC2IR	RC_W0	0	Rising edge capture 2 flag Refer to IC1IR description
16	IC1IR	RC_W0	0	Rising edge capture 1 flag This flag is set by hardware only when the corresponding channel is configured in input capture mode and triggered by rising edge. It is cleared by software or reading TIMx_CCR1. 0:No rising edge capture occurs; 1:A rising edge capture event occurs.
15:13	Reserved	-	-	Reserved
12	CC4OF	RC_W0	0	Capture/Compare 4 overcapture flag Refer to CC1OF description
11	CC3OF	RC_W0	0	Capture/Compare 3 overcapture flag Refer to CC1OF description
10	CC2OF	RC_W0	0	Capture/Compare 2 overcapture flag Refer to CC1OF description

Puya Confidential

Bit	Name	R/W	Reset Value	Function
9	CC1OF	RC_W0	0	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'. 0:No overcapture has been generated. 1:The counter value has been captured in TIMx_CCR1 register while CC1IF flag was already set.
8:7	Reserved	-	-	Reserved
6	TIF	RC_W0	0	Trigger interrupt flag This flag is set by hardware on trigger event (active edge detected on TRGI input when the slave mode controller is enabled in all modes but gated mode). It is set when the counter starts or stops when gated mode is selected. It is cleared by software. 0:No trigger event occurred. 1:Trigger interrupt pending.
5	Reserved	-	-	Reserved
4	CC4IF	RC_W0	0	Capture/Compare 4 interrupt flag Refer to CC1IF description
3	CC3IF	RC_W0	0	Capture/Compare 3 interrupt flag Refer to CC1IF description
2	CC2IF	RC_W0	0	Capture/Compare 2 interrupt flag Refer to CC1IF description
1	CC1IF	RC_W0	0	Capture/Compare 1 interrupt flag If channel CC1 is configured as output: This flag is set by hardware when the counter matches the compare value, with some exception in center-aligned mode (refer to the CMS bits in the TIMx_CR1 register description). It is cleared by software. 0:No match; 1:The content of the counter TIMx_CNT matches the content of the TIMx_CCR1 register. When the contents of TIMx_CCR1 are larger than the contents of TIMx_APR, the CC1IF bit becomes high when the counter overflow condition in up or up/down count mode, or when the counter overflow condition in down count mode If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading the TIMx_CCR1 register.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:No input capture occurred 1:The counter value has been captured in TIMx_CCR1 register (An edge has been detected on IC1 which matches the selected polarity).
0	UIF	RC_W0	0	Update interrupt flag This bit is set by hardware on an update event. It is cleared by software. 0:No update occurred. 1:Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow or underflow regarding the repetition counter value (update if repetition counter = 0) and if the UDIS=0 in the TIMx_CR1 register. − If URS = 0 and UDIS = 0 of the TIMx_CR1 register, an update event is generated when UG = 1 of the TIMx_EGR register is set, and the counter CNT is re-initialized by software. − If UDIS = 0 and URS = 0 of the TIMx_CR1 register, an update event occurs when the CNT is reinitialized by the trigger event (Refer to TIM2/3 slave mode control register (TIMx_SMCR)).

20.4.6. TIM2/3 event generation register (TIMx_EGR)

Address offset:0x14

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	TG	Res	CC4G	CC3G	CC2G	CC1G	UG
-	-	-	-	-	-	-	-	-	W	-	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:7	Reserved	-	-	Reserved
6	TG	W	0	Trigger generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:The TIF flag is set in TIMx_SR register. Related interrupt or DMA transfer can occur if enabled.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
5	Reserved	-	-	Reserved
4	CC4G	W	0	Capture/Compare 4 generation Refer to CC1G description
3	CC3G	W	0	Capture/Compare 3 generation Refer to CC1G description
2	CC2G	W	0	Capture/Compare 2 generation Refer to CC1G description
1	CC1G	W	0	Capture/Compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A capture/compare event is generated on channel CC1: If channel CC1 is configured as input: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input: The current value of the counter is captured in TIMx_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if theCC1IF flag was already set.
0	UG	W	0	Update generation This bit can be set by software, it is automatically cleared by hardware. 0:No action 1:Reinitialize the counter and generate an update of the registers. Note that the counter of the prescaler is also cleared to 0 (but the prescaler coefficient is unchanged). The counter is cleared if the center-aligned mode is selected or if DIR=0 (upcounting), else it takes the auto-reload value (TIMx_ARR) if DIR=1 (downcounting).

20.4.7. TIM2/3 capture/compare mode register 1 (TIMx_CCMR1)

Address offset:0x18

Reset value:0x0000 0000

Output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Puya Confidential

OC2CE	OC2M [2:0]	OC2PE	CO2FE	CC2S	OC1CE	OC1M [2:0]	OC1PE	OC1FE	CC1S [1:0]
IC2F [3:0]		IC2PSC [1:0]		[1:0]	IC1F [3:0]		IC1PSC [1:0]		
RW									

Output compare mode

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	OC2CE	RW	0	Output compare 2 clear enable
14:12	OC2M [2:0]	RW	000	Output compare 2 mode
11	OC2PE	RW	0	Output compare 2 preload enable
10	OC2FE	RW	0	Output compare 2 fast enable
9:8	CC2S [1:0]	RW	00	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC2 channel is configured as output; 01:CC2 channel is configured as input, IC2 is mapped on TI2; 10:CC2 channel is configured as input, IC2 is mapped on TI1; 11:CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC2S bits are writable only when the channel is OFF (CC2E = '0' in TIMx_CCER).
7	OC1CE	RW	0	Output compare 1 clear enable 0:OC1REF is not affected by the ETRF signal; 1:OC1REF is cleared as soon as a High level is detected on ETRF signal.
6:4	OC1M [2:0]	RW	00	Output compare 1 mode These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000:Frozen. The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on the outputs. 001:Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				010:Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 011:Toggle OC1REF toggles when TIMx_CNT=TIMx_CCR1. 100:Forced inactive level OC1REF is forced low. 101:Forced active level OC1REF is forced high. 110:PWM mode 1 - In upcounting, channel 1 is active as long as TIMx_CNT<TIMx_CCR1 else inactive. In downcounting, channel 1 is inactive (OC1REF= ‘0’) as long as TIMx_CNT>TIMx_CCR1 else active (OC1REF=’1’). 111:PWM mode 2 - In upcounting, channel 1 is inactive as long as TIMx_CNT<TIMx_CCR1 else active. In downcounting, channel 1 is active as long as TIMx_CNT>TIMx_CCR1 else inactive. Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output). Note:In PWM mode 1 or 2, the OCREF level changes when the result of the comparison changes or when the output compare mode switches from frozen to PWM mode.
3	OC1PE	RW	0	Output Compare 1 preload enable 0:Preload register on TIMx_CCR1 disabled. TIMx_CCR1 can be written at any time, the new value is taken in account immediately. 1:Preload register on TIMx_CCR1 enabled. Read/Write operations access the preload register. TIMx_CCR1 preload value is loaded in the active register at each update event. Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output). Note:The PWM mode can be used without validating the preload register only in one pulse mode. Else the behavior is not guaranteed.
2	OC1FE	RW	0	Output compare 1 fast enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0:CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1:An active edge on the trigger input acts like a compare match on OC1 output. Then, OC1 is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OCFE acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S [1:0]	RW	00	Capture/Compare 1 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:CC1 channel is configured as input, IC1 is mapped on TI2; 11:CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC1S bits are writable only when the channel is OFF (CC1E = ‘0’ in TIMx_CCER).

Input capture mode:

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:12	IC2F [3:0]	RW	0000	Input capture 2 filter
11:10	IC2PSC [1:0]	RW	00	Capture/Compare 2 prescaler
9:8	CC2S [1:0]	RW	0	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC2 channel is configured as output; 01:CC2 channel is configured as input, IC2 is mapped on TI2; 10:CC2 channel is configured as input, IC2 is mapped on TI1;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC2S bits are writable only when the channel is OFF (CC2E = '0' in TIMx_CCER).
7:4	IC1F [3:0]	RW	0000	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output: 0000:No filter, sampling is done at fDTS 0001:$f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010:$f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011:$f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100:$f_{SAMPLING} = f_{DTS}/2$, N = 6 0101:$f_{SAMPLING} = f_{DTS}/2$, N = 8 0110:$f_{SAMPLING} = f_{DTS}/4$, N = 6 0111:$f_{SAMPLING} = f_{DTS}/4$, N = 8 1000:$f_{SAMPLING} = f_{DTS}/8$, N = 6 1001:$f_{SAMPLING} = f_{DTS}/8$, N = 8 1010:$f_{SAMPLING} = f_{DTS}/16$, N = 5 1011:$f_{SAMPLING} = f_{DTS}/16$, N = 6 1100:$f_{SAMPLING} = f_{DTS}/16$, N = 8 1101:$f_{SAMPLING} = f_{DTS}/32$, N = 5 1110:$f_{SAMPLING} = f_{DTS}/32$, N = 6 1111:$f_{SAMPLING} = f_{DTS}/32$, N = 8
3:2	IC1PSC [1:0]	RW	00	Input capture 1 prescaler 00:no prescaler, capture is done each time an edge is detected on the capture input; 01:capture is done once every 2 events 10:capture is done once every 4 events 11:capture is done once every 8 events
1:0	CC1S [1:0]	RW	00	CC1S [1:0]:capture/compare 1 selection. This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:CC1 channel is configured as input, IC1 is mapped on TI2;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC1S bits are writable only when the channel is OFF (CC1E = '0' in TIMx_CCER).

20.4.8. TIM2/3 capture/compare mode register 2 (TIMx_CCMR2)

Address offset:0x1C

Reset value:0x0000 0000

Output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M [2:0]			OC4PE	CO4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
IC4F [3:0]				IC4PSC [1:0]				IC3F [3:0]				IC3PSC [1:0]			
RW															

Output compare mode

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	OC4CE	RW	0	Output Compare 4 clear enable
14:12	OC4M [2:0]	RW	000	Output Compare 4 mode
11	OC4PE	RW	0	Output Compare 4 preload enable
10	OC4FE	RW	0	Output Compare 4 fast enable
9:8	CC4S [1:0]	RW	00	Capture/Compare 4 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC4 channel is configured as output 01:CC4 channel is configured as input, IC4 is mapped on TI4. 10:CC4 channel is configured as input, IC4 is mapped on TI3. 11:CC4 channel is configured as input, IC4 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC4S bits are writable only when the channel is OFF (CC4E = '0' in TIMx_CCER).
7	OC3CE	RW	0	Output Compare 3 clear enable
6:4	OC3M [2:0]	RW	00	Output Compare 3 mode
3	OC3PE	RW	0	Output Compare 3 preload enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
2	OC3FE	RW	0	Output Compare 3 fast enable
1:0	CC3S [1:0]	RW	00	Capture/Compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC3 channel is configured as output 01:CC3 channel is configured as input, IC3 is mapped on TI3. 10:CC3 channel is configured as input, IC3 is mapped on TI4. 11:CC3 channel is configured as input, IC3 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC3S bits are writable only when the channel is OFF (CC3E = '0' in TIMx_CCER).

Input capture mode:

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:12	IC4F [3:0]	RW	0000	Input capture 4 filter
11:10	IC4PSC [1:0]	RW	00	Capture/Compare 4 prescaler
9:8	CC4S [1:0]	RW	0	Capture/Compare 4 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC4 channel is configured as output 01:CC4 channel is configured as input, IC4 is mapped on TI4. 10:CC4 channel is configured as input, IC4 is mapped on TI3. 11:CC4 channel is configured as input, IC4 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC4S bits are writable only when the channel is OFF (CC4E = '0' in TIMx_CCER).
7:4	IC3F [3:0]	RW	0000	Input capture 3 filter
3:2	IC3PSC [1:0]	RW	00	Input capture 3 prescaler
1:0	CC3S [1:0]	RW	00	Capture/Compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC3 channel is configured as output 01:CC3 channel is configured as input, IC3 is mapped on TI3. 10:CC3 channel is configured as input, IC3 is mapped on TI3;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				11:CC3 channel is configured as input, IC3 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC3S bits are writable only when the channel is OFF (CC3E = '0' in TIMx_CCER).

20.4.9. TIM2/3 capture/compare enable register (TIMx_CCER)

Address offset:0x20

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res	CC4P	CC4E	CC3NP	Res	CC3P	CC3E	CC2NP	Res	CC2P	CC2E	CC1NP	Res	CC1P	CC1E
RW	-	RW	RW	RW	-	RW	RW	RW	-	RW	RW	RW	-	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15	CC4NP	RW	0	Capture/compare 4 complementary output polarity. Refer to CC1NP description.
14	Reserved	-	-	Reserved
13	CC4P	RW	0	Capture/compare 4 output polarity Refer to CC1P description.
12	CC4E	-	0	Capture/Compare 4 output enable Refer to CC1E description.
11	CC3NP	RW	0	Capture/compare 3 complementary output polarity. Refer to CC1NP description.
10	Reserved	-	-	Reserved
9	CC3P	RW	0	Capture/compare 3 output polarity Refer to CC1P description.
8	CC3E	RW	0	Capture/Compare 3 output enable Refer to CC1E description.
7	CC2NP	RW	0	Capture/compare 2 complementary output polarity. Refer to CC1NP description.
6	Reserved	-	-	Reserved
5	CC2P	RW	0	Capture/compare 2 output polarity Refer to CC1P description.
4	CC2E	RW	0	Capture/Compare 2 output enable Refer to CC1E description.
3	CC1NP	RW	0	Capture/compare 1 complementary output polarity. 0:OC1N active high. 1:OC1N active low. This bit is used in conjunction with CC1P to define that polarity of TI1FP1/TI2FP1, refer to CC1P description
2	Reserved	-	-	Reserved

Puya Confidential

Bit	Name	R/W	Reset Value	Function
1	CC1P	RW	0	Capture/compare 1 output polarity CC1 channel configured as output: 0:OC1 active high. 1:OC1 active low. CC1 channel configured as input: CC1NP/CC1P bits select the active polarity of TI1FP1and TI2FP1 for trigger or capture operations. 00:Non-inverted/rising edge: The circuit is sensitive to TIxFP1 rising edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode or encoder mode). 01:Inverted/falling edge: The circuit is sensitive to TIxFP1 falling edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is inverted (trigger operation in gated mode or encoder mode). 10:reserved, do not use this configuration. 11:non-inverted/double edge The circuit is sensitive to both TIxFP1 rising and falling edges (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode). This configuration must not be used in encoder mode.
0	CC1E	RW	0	Capture/Compare 1 output enable CC1 channel configured as output: 0:Off - OC1 is not active. 1:On - OC1 signal is output on the corresponding output pin. CC1 channel configured as input: This bit determines if a capture of the counter value can actually be done into the input capture/compare register 1 (TIMx_CCR1) or not. 0:Capture disabled. 0:Capture enabled.

Output control bit for standard OCx channels

CcxE bit	OCx output state
0	Output Disabled (OCx=0, OCx_EN=0)
1	OCx= OCxREF+Polarity, OCx_EN=1

Puya Confidential

20.4.10. TIM2/3 counter (TIMx_CNT)

Address offset:0x24

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT [31:16] (TIM2 only)															
RW															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	CNT [31:16]	RW	0	Counter value (TIM2 only)
15:0	CNT [15:0]	RW	0	Counter value

20.4.11. TIM2/3 prescaler (TIMx_PSC)

Address offset:0x28

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSC [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PSC [15:0]	RW	0	Prescaler value The counter clock frequency (CK_CNT) is equal to f_{CK_PSC} / (PSC [15:0] +1). PSC contains the value to be loaded in the active pre-scaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in reset mode).

20.4.12. TIM2/3 auto-reload register (TIMx_ARR)

Address offset:0x2C

Reset value:0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR [31:16] (TIM2 only)															
RW															

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	ARR [31:16]	RW	FFFF	Auto-reload value (TIM2 only)
15:0	ARR [15:0]	RW	FFFF	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. Refer to 22.3.1:Time-base unit for more details about ARR update and behavior. The counter is blocked while the auto-reload value is null.

20.4.13. TIM2/3 capture/compare register 1 (TIMx_CCR1)

Address offset:0x34

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1 [31:16] (TIM2 only)															
RW/R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
RW/R															

Bit	Name	R/W	Reset Value	Function
31:16	CCR1 [31:16]	RW	0	Capture/Compare values for 1 (TIM2 only)
15:0	CCR1 [15:0]	RW	0	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).

Puya Confidential

20.4.14. TIM2/3 capture/compare register 2 (TIMx_CCR2)

Address offset:0x38

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2 [31:16] (TIM2 only)															
RW/R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
RW/R															

Bit	Name	R/W	Reset Value	Function
31:16	CCR2 [31:16]	RW	0	Capture/Compare values for 2 (TIM2 only)
15:0	CCR2 [15:0]	RW	0	Capture/Compare 2 value If channel CC2 is configured as output: CCR2 is the value to be loaded in the actual capture/compare 2 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC2PE). Else the preload value is copied in the active capture/compare 2 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC output. If channel CC2 is configured as input: CCR2 is the counter value transferred by the last input capture 2 event (IC2).

20.4.15. TIM2/3 capture/compare register 3 (TIMx_CCR3)

Address offset:0x3C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3 [31:16] (TIM2 only)															
RW/R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3 [15:0]															
RW/R															

Bit	Name	R/W	Reset Value	Function
31:16	CCR3 [31:16]	RW	0	Capture/Compare 3 value (TIM2 only)

Puya Confidential

Bit	Name	R/W	Reset Value	Function
15:0	CCR3 [15:0]	RW	0	Capture/Compare 3 value **If channel CC3 is configured as output:** CCR3 is the value to be loaded in the actual capture/compare 3 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR2 register (bit OC3PE). Else the preload value is copied in the active capture/compare 3 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC output. **If channel CC3 is configured as input:** CCR3 is the counter value transferred by the last input capture 3 event (IC3).

20.4.16. TIM2/3 capture/compare register 4 (TIMx_CCR4)

Address offset:0x40

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4 [31:16] (TIM2 only)															
RW/R															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR4 [15:0]															
RW/R															

Bit	Name	R/W	Reset Value	Function
31:16	CCR4 [31:16]	RW	0	Capture/Compare 4 value (TIM2 only)
15:0	CCR4 [15:0]	RW	0	Capture/Compare 4 value **If channel CC4 is configured as output:** CCR4 is the value to be loaded in the actual capture/compare 4 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR2 register (bit OC4PE). Else the preload value is copied in the active capture/compare 4 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC output. **If channel CC4 is configured as input:**

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				CCR4 is the counter value transferred by the last input capture 4 event (IC4).

20.4.17. TIM2/3 DMA control register (TIMx_DCR)

Address offset:0x48

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	DBL [4:0]					Res			DBA [4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:13	Reserved	-	-	Reserved
12:8	DBL [4:0]	RW	0 0000	DMA burst length These bits define the transfer length of the DMA in continuous mode (when reading or writing to the address of the TIMx_DMAR register address is performed). 00000:1 transfer 00001:2 transfers 00010:3 transfers .... 10001:18 transfers Example:We consider such a transfer:DBL = 7, DBA = TIMx_CR1 If DBL = 7, DBA = TIMx_CR1 represents the address of the data to be transferred, then the transferred address is given by: (address of TIMx_CR1) + DBA + (DMA index), where DMA index = DBL Where (address of TIMx_CR1) + DBA plus 7 gives the address where data will be written or read out, so that data transfer will occur in 7 registers starting from address (address of TIMx_CR1) + DBA. Depending on the setting of the DMA data length, the following can occur: -If the data is set to half word (16 bits), then the data will be transferred to all 7 registers.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				-If the data is set to bytes, the data will still be transferred to all 7 registers:the first register contains the first MSB byte, the second register contains the first LSB byte, and so on. Therefore, for the timer, the user must specify the data width to be transmitted by the DMA.
7:5	Reserved	-	-	Reserved
4:0	DBA [4:0]	RW	0 0000	DBA [4:0]:DMA base address These bits define the base address of the DMA in continuous mode (when read/write access are done through the TIMx_DMAR address). DBA is defined as an offset starting from the address of the TIMx_CR1 register. 00000:TIMx_CR1 00001:TIMx_CR2 00010:TIMx_SMCR .....

20.4.18. TIM2/3 DMA address for full transfer (TIMx_DMAR)

Address offset:0x4C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB [31:16]															
RW															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAB [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:0	DMAB [31:0]	RW	0	DMA register for burst accesses A read or write operation to the DMAR register accesses the register located at the address: (TIMx_CR1 address) + (DBA + DMA index) x 4 where: TIMx_CR1 address is the address of the control register 1; DBA is the DMA base address configured in TIMx_DCR register; DMA index is automatically controlled by the DMA transfer, and ranges from 0 to DBL (DBL configured in TIMx_DCR).

Puya Confidential

21. Basic timers (TIM6/TIM7)

21.1. TIM6 and TIM7 introduction

The basic timers TIM6/TIM7 consist of a 16-bit auto-reload counter driven by their programmable prescaler respectively.

The timers are completely independent, and do not share any resources.

21.2. TIM6 and TIM7 main features

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide the counter clock frequency by any factor between 1 and 65536 (can be changed on the fly).
- Interrupt/DMA generation on the following events:

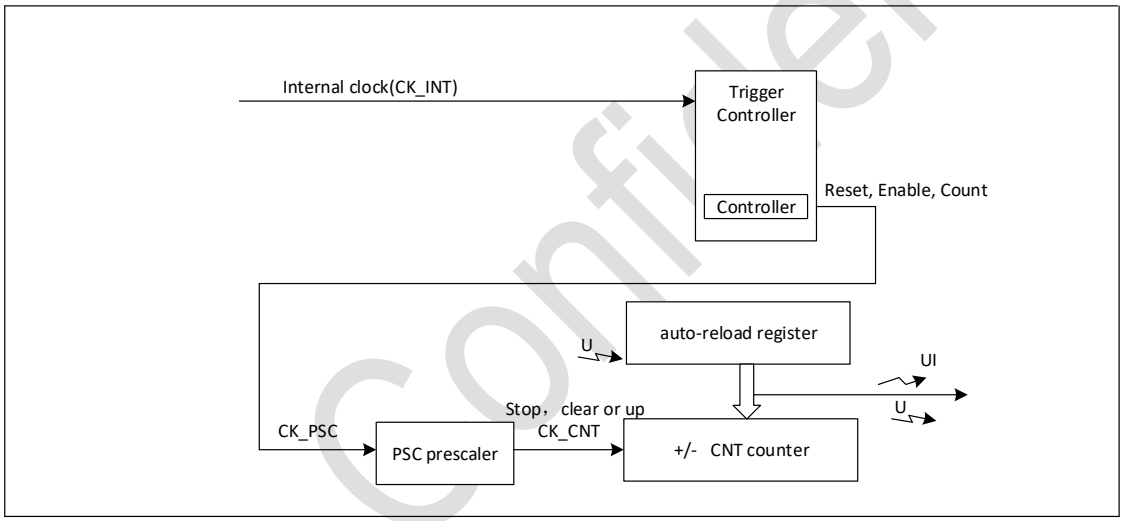


Figure 21-1 Basic timer block diagram

21.3. TIM6/TIM7 functional description

21.3.1. Time-base unit

The main block of the timer is a 16-bit up-counter with its related auto-reload register. The counter clock can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.

The time-base unit includes:

- Counter register (TIMx_CNT)
- Prescaler register (TIMx_PSC)
- Auto-reload register (TIMx_ARR)

Puya Confidential

The auto-reload register is preloaded. Writing to or reading from the auto-reload register accesses the preload register. The content of the preload register is transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload preload enable bit (ARPE) in TIMx_CR1 register. The update event is sent when the counter reaches the overflow value and if the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software.

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in TIM14_CR1 register is set.

Note that the counter starts counting 1 clock cycle after setting the CEN bit in the TIMx_CR register.

Prescaler description

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIMx_PSC register). It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

Tables below give some examples of the counter behavior when the prescaler ratio is changed on the fly.

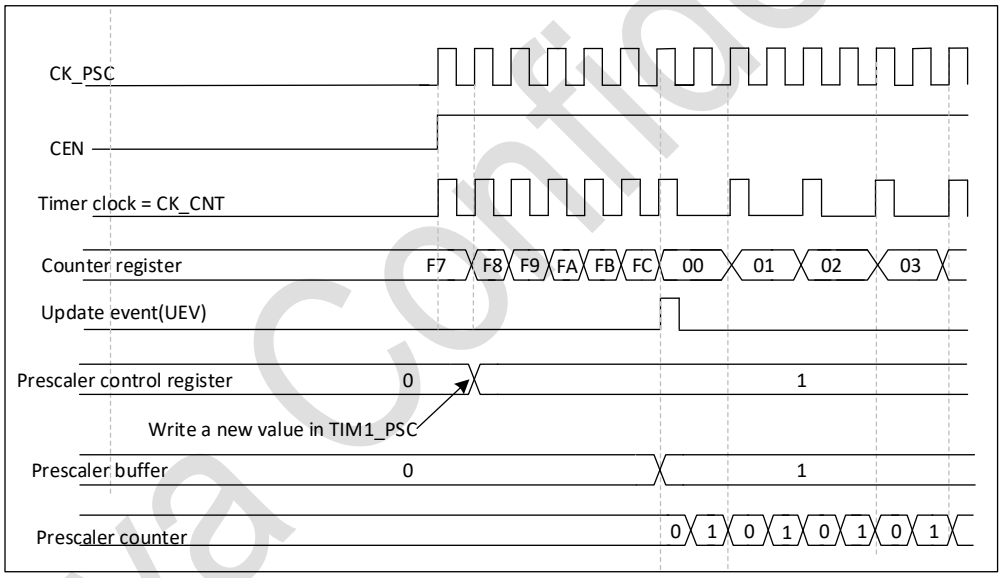


Figure 21-2 Counter timing diagram with prescaler division change from 1 to 2

Puya Confidential

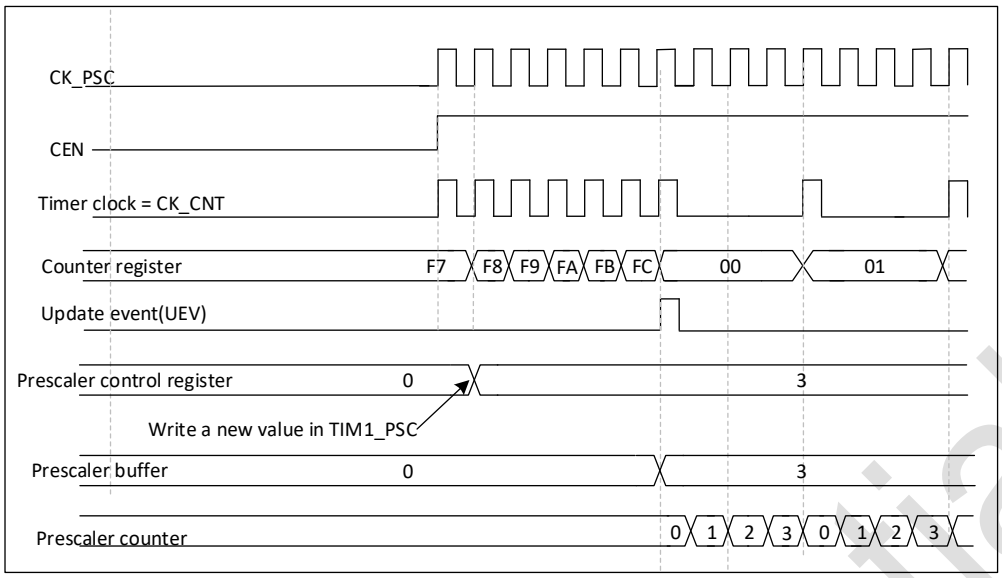


Figure 21-3 Counter timing diagram with prescaler division change from 1 to 4

Upcounting mode

The counter counts from 0 to the auto-reload value (contents of the TIMx_ARR register), then restarts from 0 and generates a counter overflow event.

Setting the UG bit in the TIMx_EGR register (by software) also generates an update event.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts from 0, as well as the counter of the prescaler (but the prescale rate does not change). In addition, if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag (thus no interrupt or DMA request is sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The auto-reload shadow register is updated with the preload value (TIMx_ARR).
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register). The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR=0x36.

Puya Confidential

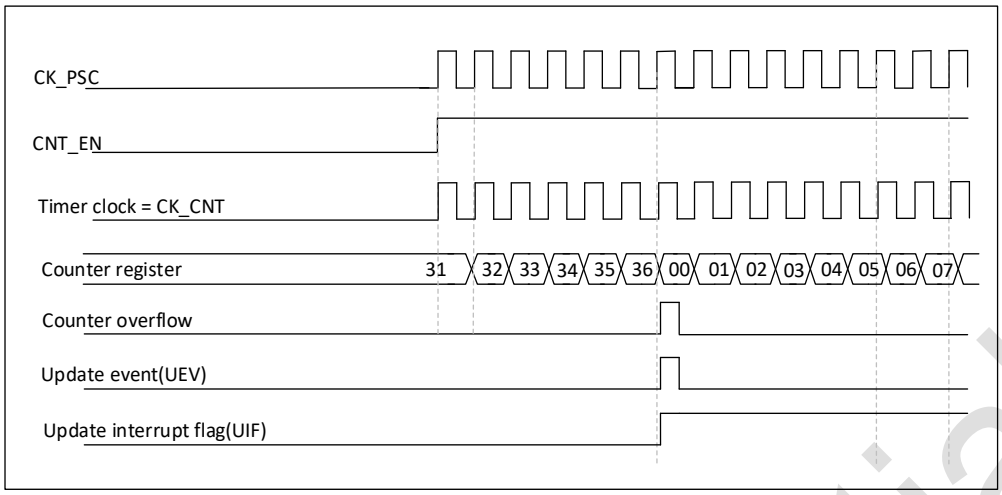


Figure 21-4 Counter timing diagram, internal clock divided by 1

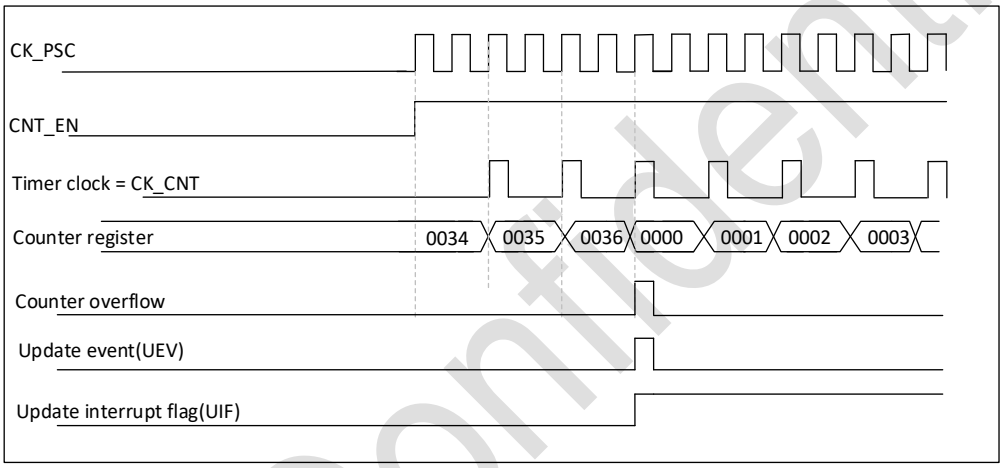


Figure 21-5 Counter timing diagram, internal clock divided by 2

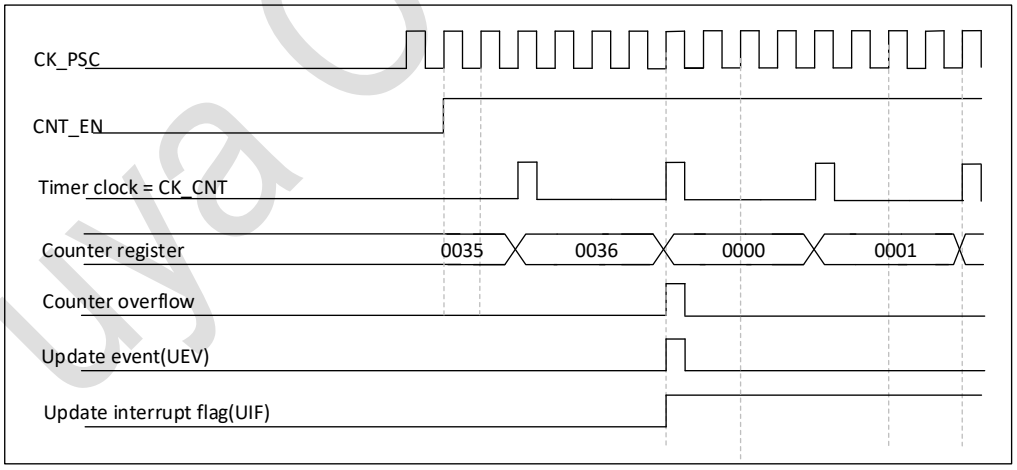


Figure 21-6 Counter timing diagram, internal clock divided by 4

Puya Confidential

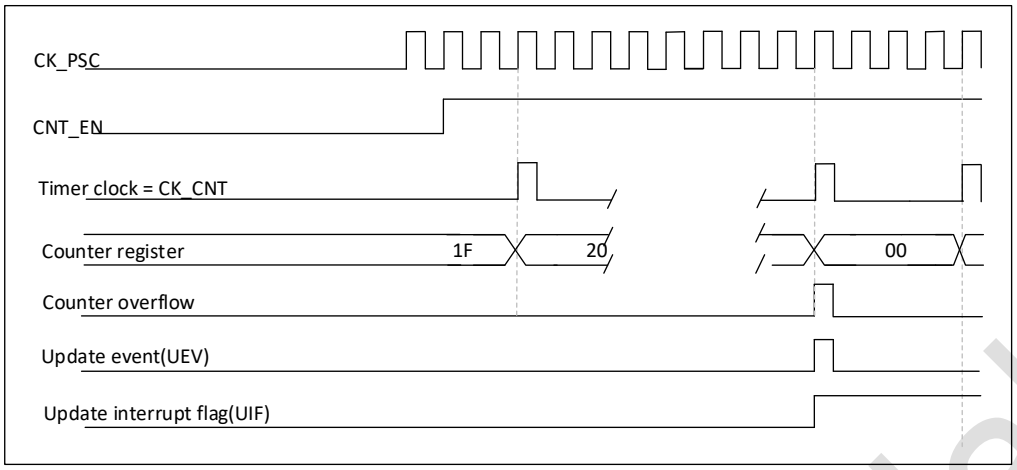


Figure 21-7 Counter timing diagram, internal clock divided by N

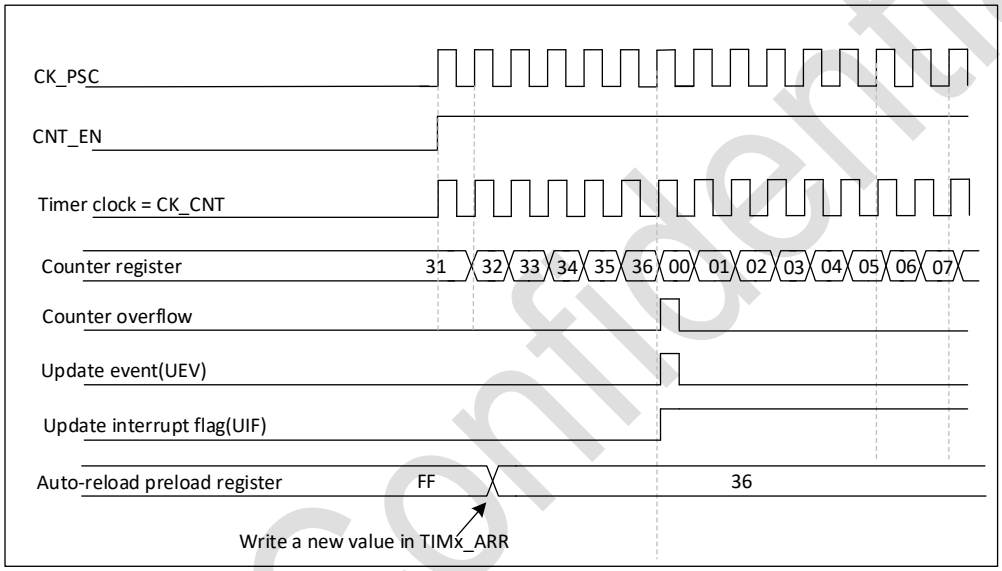


Figure 21-8 Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded)

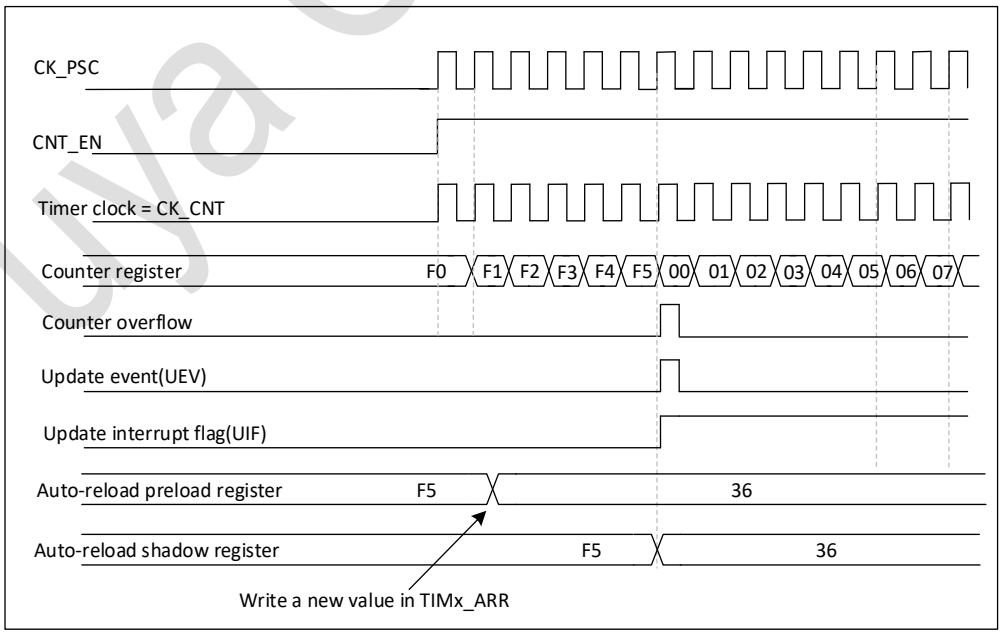


Figure 21-9 Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded)

Puya Confidential

21.3.2. Clock sources

The counter clock is provided by the Internal clock (CK_INT) source. The CEN (in the TIMx_CR1 register) and UG bits (in the TIMx_EGR register) are actual control bits and can be changed only by software (except for UG that remains cleared automatically). As soon as the CEN bit is written to 1, the prescaler is clocked by the internal clock CK_INT.

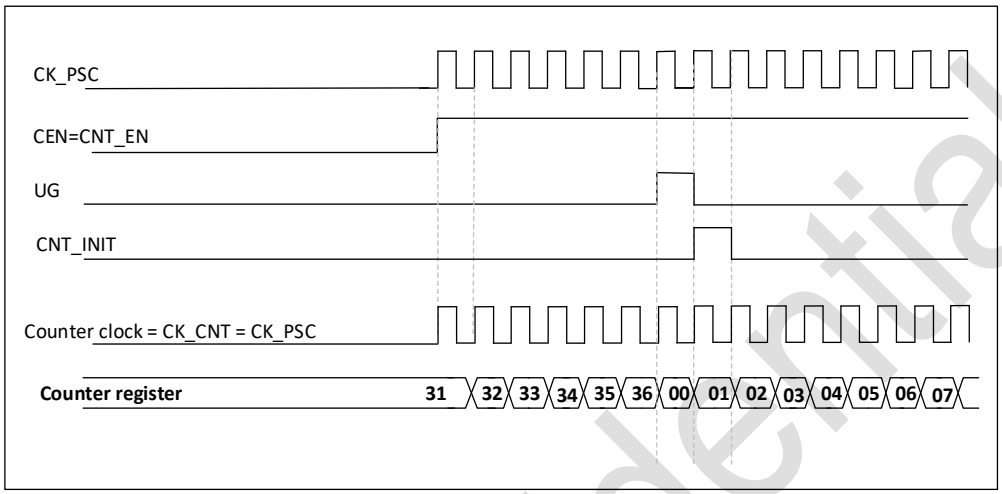


Figure 21-10 Control circuit in normal mode, internal clock divided by 1

21.3.3. Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TIMx counter either continues to work normally or stops, depending on DBG_TIMx_STOP configuration bit in DBG module.

21.4. TIM6 and TIM7 registers

21.4.1. TIM6 and TIM7 control register 1 (TIMx_CR1)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res		ARPE	Res	Res	Res	OPM	URS	UDIS	CEN
-	-	-	-	-	-	-		RW	-	-	-	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7	ARPE	RW	0	Auto-reload preload enable 0:TIMx_ARR register is not buffered 1:TIMx_ARR register is buffered
6:4	Reserved	-	-	Reserved

Puya Confidential

Bit	Name	R/W	Reset Value	Function
3	OPM	RW	0	One pulse mode 0:Counter is not stopped at update event 1:Counter stops counting at the next update event (clearing the bit CEN)
2	URS	RW	0	Update request source This bit is set and cleared by software to select the UEV event sources. 0:Any of the following events generate an update interrupt or DMA request if enabled. These events can be: – Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller 1:Only counter overflow/underflow generates an update interrupt or DMA request if enabled
1	UDIS	RW	0	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0:UEV enabled. The Update (UEV) event is generated by one of the following events: – Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1:UEV disabled. The update event is not generated, shadow registers keep their value (ARR, PSC, CCRx). However, the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	RW	0	Counter enable 0:Counter disabled 1:Counter enabled Note:external clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However, trigger mode can set the CEN bit automatically by hardware.

21.4.2. TIM6 and TIM7 control register 2 (TIMx_CR2)

Address offset:0x04

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res		Res	MMS [2:0]			Res	Res	Res	Res
-	-	-	-	-	-	-		-	RW			-	-	-	-

Bit	Name	R/W	Reset Value	Function
31:7	Reserved	-	-	Reserved
6:4	MMS [2:0]	RW	0	Master mode selection These three bits are used to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000:Reset - the UG bit from the TIMx_EGR register is used as trigger output (TRGO). If reset is generated by the trigger input (slave mode controller configured in reset mode) then the signal on TRGO is delayed compared to the actual reset. 001:Enable - the counter enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enabled. The counter enable signal is generated by a logic AND between CEN control bit and the trigger input when configured in gated mode. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected (see the MSM bit description in TIMx_SMCR register). 010:Update - The update event is selected as trigger output (TRGO). For instance, a master timer can then be used as a prescaler for a slave timer. Note:The clock of the slave timer or ADC must be enabled prior to receive events from the master timer, and must not be changed.
3:0	Reserved	-	-	Reserved

21.4.3. TIM6 and TIM7 DMA/Interrupt enable registers (TIMx_DIER)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	UDE	Res	Res	Res	Res	Res	Res	Res	UIE
-	-	-	-	-	-	-	RW	-	-	-	-	-	-	-	RW

Bit	Name	R/W	Reset Value	Function
31:9	Reserved	-	-	Reserved
8	UDE	RW	0	UDE:Update DMA request enable 0:Update DMA request disabled 1:Update DMA request enabled
7:1	Reserved	-	-	Reserved
0	UIE	RW	0	UIE:Update interrupt enable 0:Update interrupt disabled. 1:Update interrupt enabled

21.4.4. TIM6 and TIM7 status registers (TIMx_SR)

Address offset:0x010

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIF
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RC_W0

Bit	Name	R/W	Reset Value	Function
31:1	Reserved	-	-	Reserved
0	UIF	RC_W0	0	Update interrupt flag. This bit is set by hardware on an update event. It is cleared by software. 0:No update occurred. 1:Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow and if UDIS '0' in the TIMx_CR1 register. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0 in the TIMx_CR1 register.

21.4.5. TIM6 and TIM7 event generation register (TIMx_EGR)

Address offset:0x14

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res															UG
-															W

Bit	Name	R/W	Reset Value	Function
31:1	Reserved	-	-	Reserved
0	UG	W	0	Update generation. This bit can be set by software, it is automatically cleared by hardware. 0:No action 1:Reinitialize the counter and generates an update of the registers. Note that the counter of the prescaler is also cleared to 0 (but the prescaler coefficient is unchanged).

21.4.6. TIM6 and TIM7 counters (TIMx_CNT)

Address offset:0x24

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CNT [15:0]	RW	0	Counter value

21.4.7. TIM6 and TIM7 prescalers (TIMx_PSC)

Address offset:0x28

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSC [15:0]															
RW															

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PSC [15:0]	RW	0	Prescaler value The counter clock frequency (CK_CNT) is equal to fCK_PSC / (PSC [15:0] +1). PSC contains the value to be loaded in the active pre-scaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in reset mode).

21.4.8. TIM6 and TIM7 auto-reload register (TIMx_ARR)

Address offset:0x2C

Reset value:0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ARR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	ARR [15:0]	RW	FFFF	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. Refer to 23.3.1:Time-base unit for more details about ARR update and behavior. The counter is blocked while the auto-reload value is null.

Puya Confidential

22. General-purpose timers (TIM14)

22.1. TIM14 introduction

The universal timer TIM14 consists of a 16-bit auto-load counter driven by a programmable prescaler.

It may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers.

The timers are completely independent, and do not share any resources. They can be synchronized together.

22.2. TIM14 main features

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide the counter clock frequency by any factor between 1 and 65536 (can be changed on the fly).
- One independent channel for:
  - Input capture
  - Output compare
  - PWM generation (edge-aligned mode)

Interrupt generation on the following events:

- Update:counter overflow, counter initialization (by software)
- Input capture
- Output compare

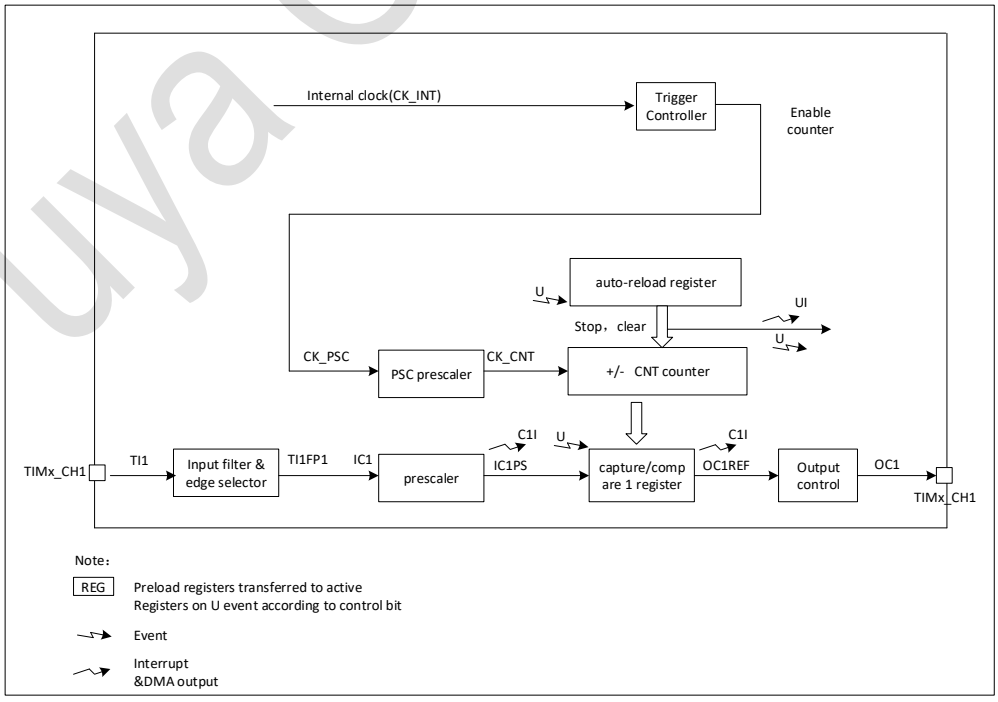


Figure 22-1 General-purpose timer block diagram (TIM14)

Puya Confidential

22.3. TIM14 functional description

22.3.1. Time-base unit

The main block of the timer is a 16-bit up-counter with its related auto-reload register. The counter clock can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.

The time-base unit includes:

- Counter register (TIM14_CNT)
- Prescaler register (TIM14_PSC)
- Auto-reload register (TIM14_ARR)

The auto-reload register is preloaded. Writing to or reading from the auto-reload register accesses the preload register. The content of the preload register is transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload preload enable bit (ARPE) in TIM14_CR1 register. The update event is sent when the counter reaches the overflow value and if the UDIS bit equals 0 in the TIM14_CR1 register. It can also be generated by software.

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in TIM14_CR1 register is set.

Note that the counter starts counting 1 clock cycle after setting the CEN bit in the TIM14_CR1 register.

Prescaler description

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIM14_PSC register). It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

Tables below give some examples of the counter behavior when the prescaler ratio is changed on the fly.

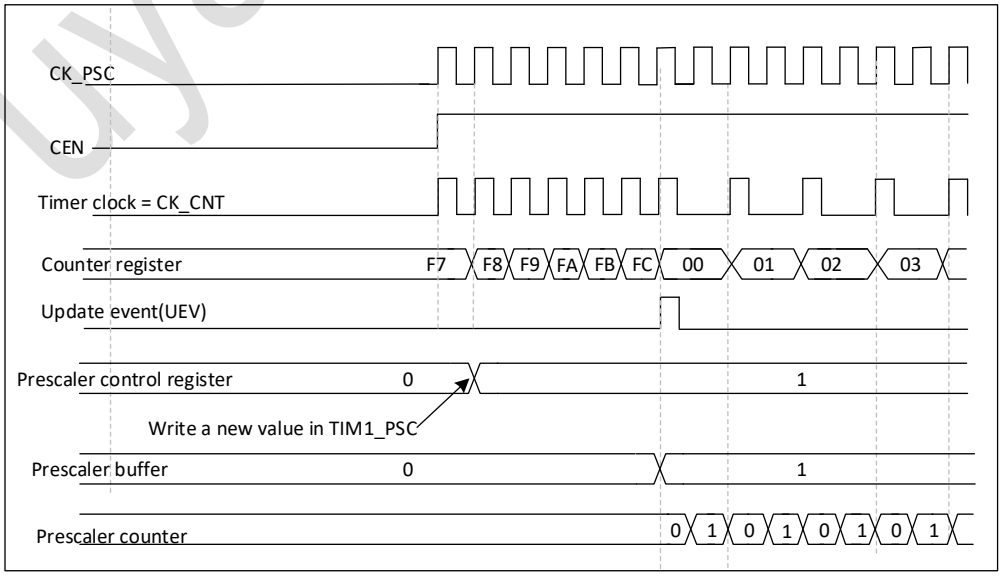


Figure 22-2 Counter timing diagram with prescaler division change from 1 to 2

Puya Confidential

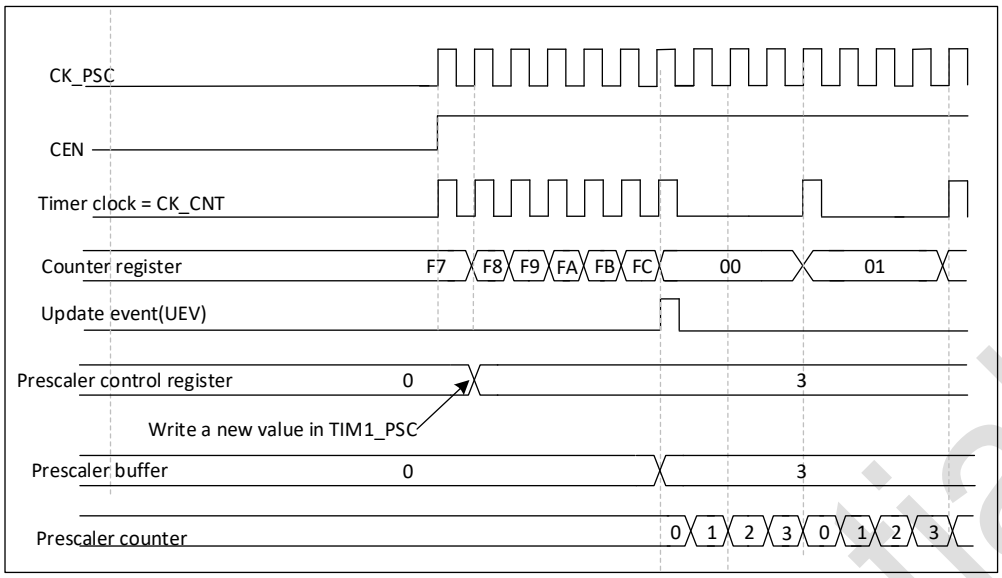


Figure 22-3 Counter timing diagram with prescaler division change from 1 to 4

Upcounting mode

The counter counts from 0 to the auto-reload value (contents of the TIM14_ARR register), then restarts from 0 and generates a counter overflow event.

Setting the UG bit in the TIM14_EGR register (by software) also generates an update event.

The UEV event can be disabled by software by setting the UDIS bit in the TIM14_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts from 0, as well as the counter of the prescaler (but the prescale rate does not change). In addition, if the URS bit (update request selection) in TIM14_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag (thus no interrupt or DMA request is sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIM14_SR register) is set (depending on the URS bit).

- The auto-reload shadow register is updated with the preload value (TIM14_ARR).
- The buffer of the prescaler is reloaded with the preload value (content of the TIM14_PSC register).

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR=0x36.

Puya Confidential

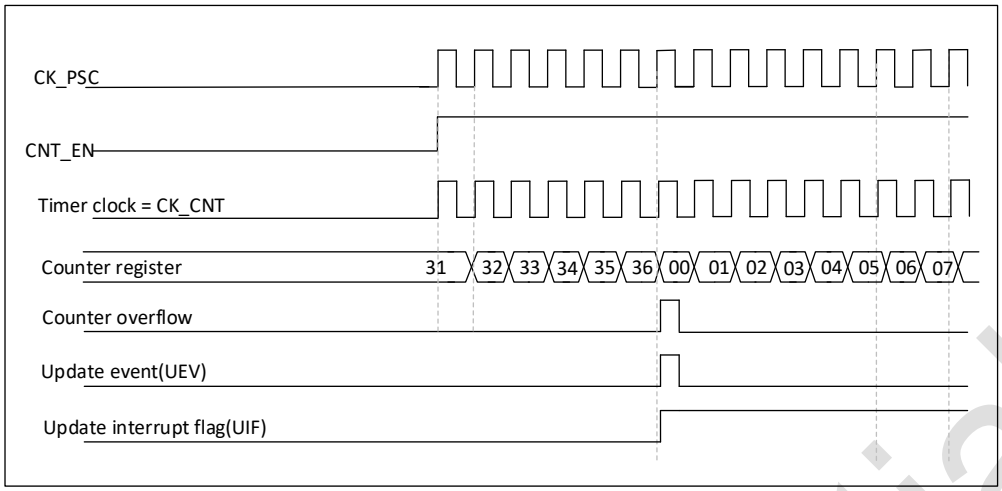


Figure 22-4 Counter timing diagram, internal clock divided by 1

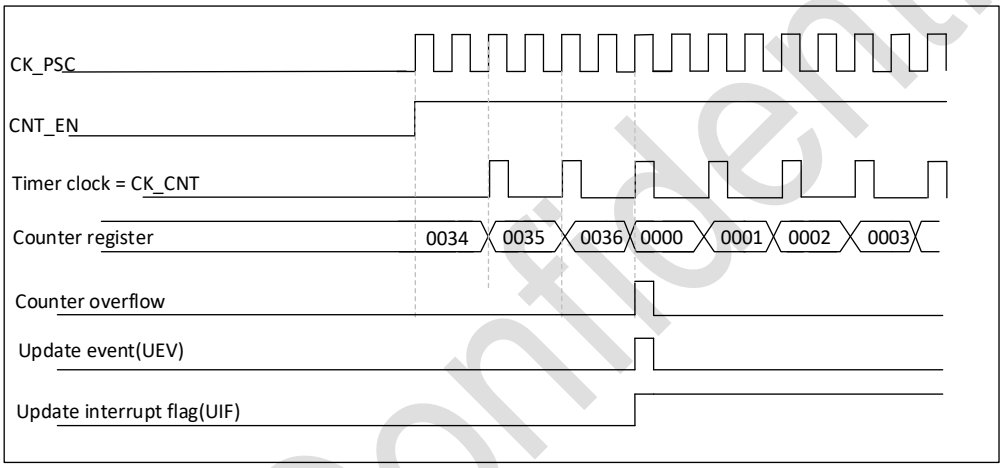


Figure 22-5 Counter timing diagram, internal clock divided by 2

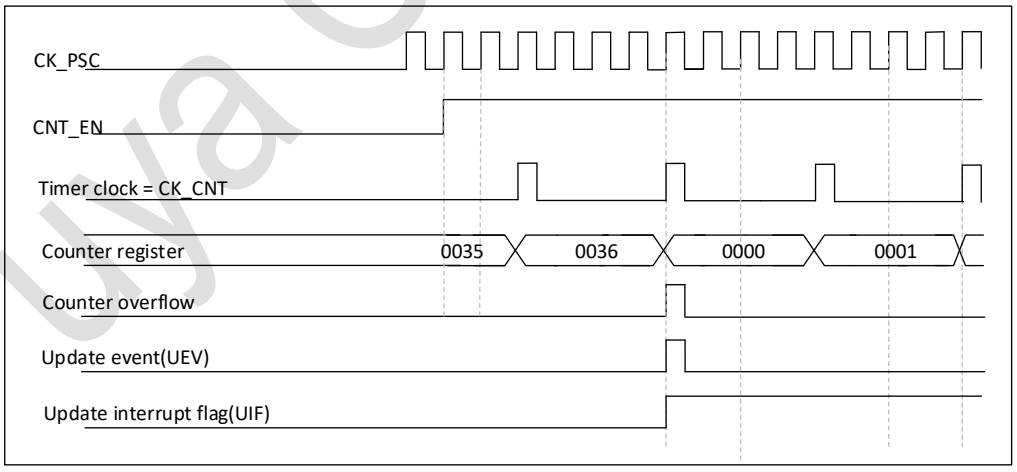


Figure 22-6 Counter timing diagram, internal clock divided by 4

Puya Confidential

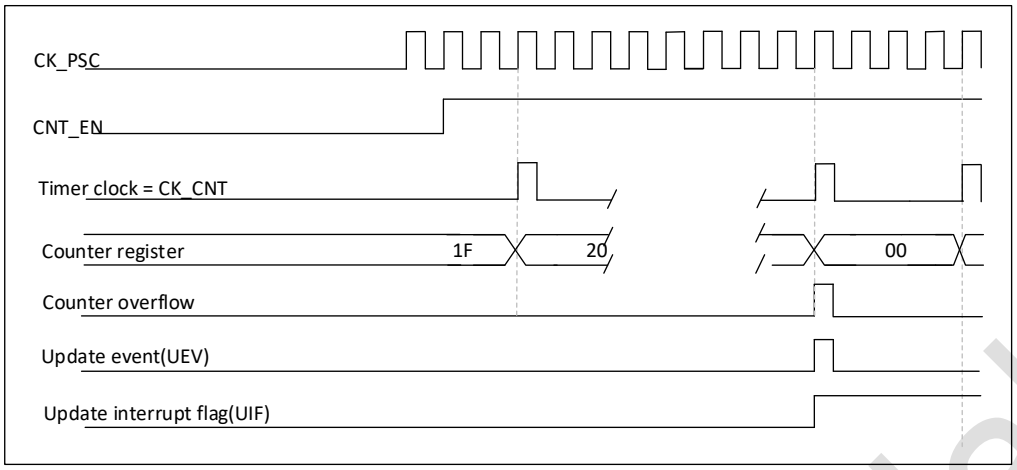


Figure 22-7 Counter timing diagram, internal clock divided by N

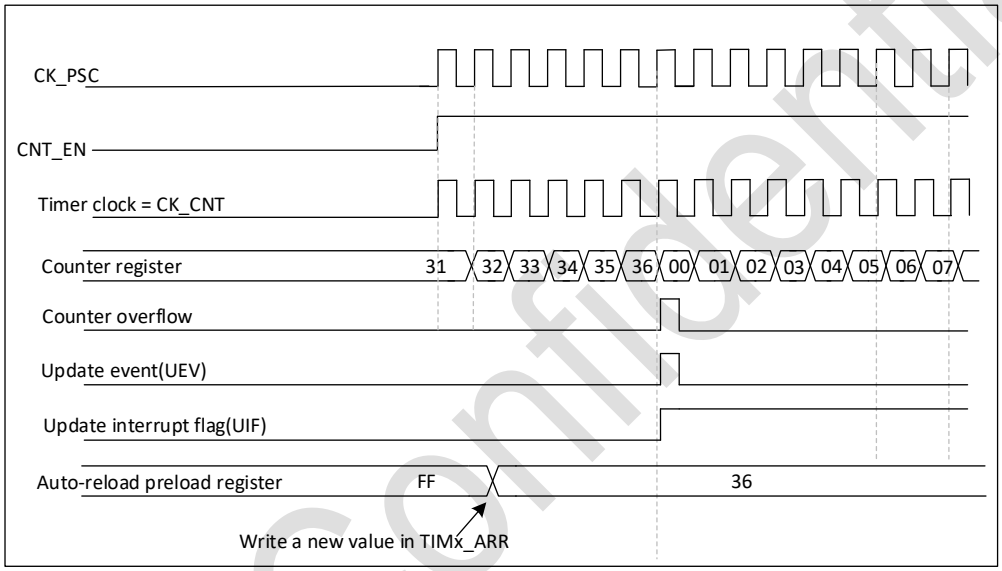


Figure 22-8 Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded)

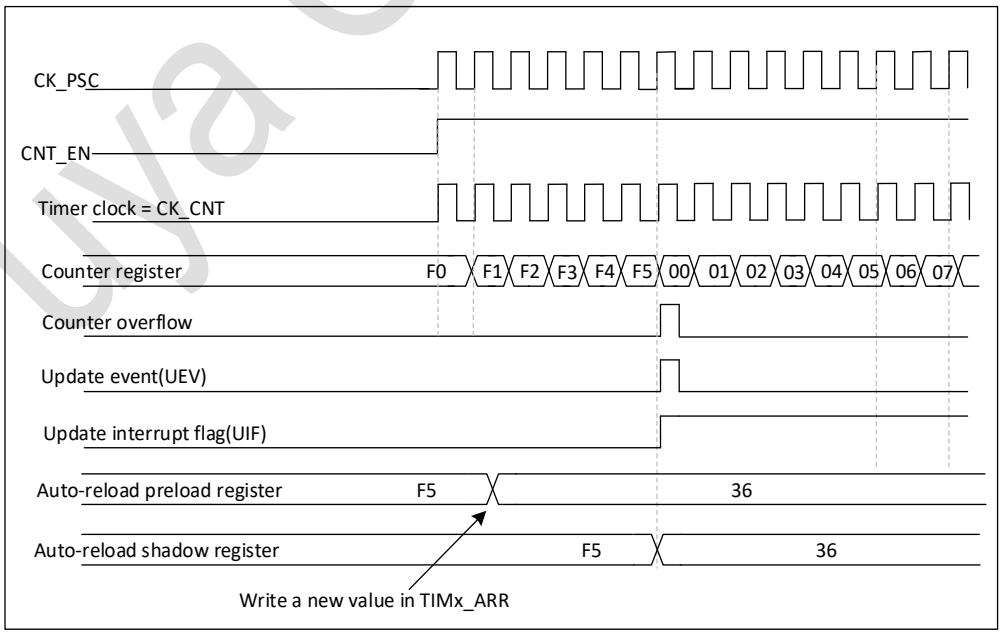


Figure 22-9 Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded)

Puya Confidential

22.3.2. Clock sources

The counter clock is provided by the internal clock (CK_INT) source. The CEN (in the TIMx_CR1 register) and UG bits (in the TIM14_EGR register) are actual control bits and can be changed only by software (except for UG that remains cleared automatically). As soon as the CEN bit is written to 1, the prescaler is clocked by the internal clock CK_INT.

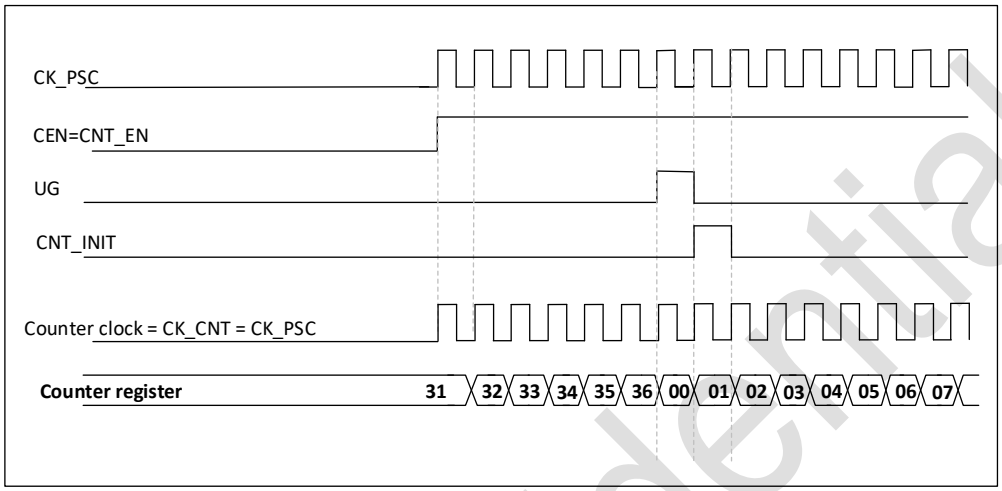


Figure 22-10 Control circuit in normal mode, internal clock divided by 1

22.3.3. Capture/Compare channels

Each Capture/Compare channel is built around a capture/compare register (including a shadow register), an input stage for capture (with digital filter, multiplexing and prescaler) and an output stage (with comparator and output control).

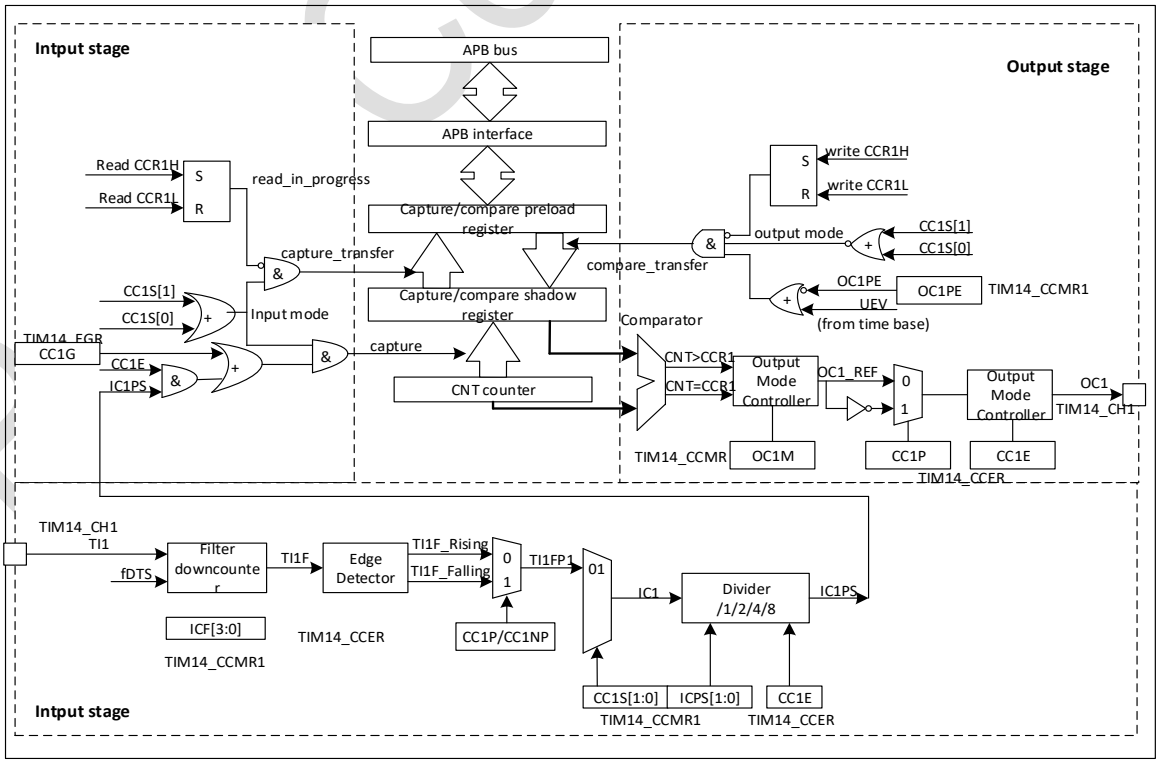


Figure 22-11 TIM14 capture/compare channel

Confidential

The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detector with polarity selection generates a signal (TIxFPx) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (ICxPS).

The output stage generates an intermediate waveform (active high) which is then used for reference. The polarity acts at the end of the chain.

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register.

In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

22.3.4. Input capture mode:

In input capture mode, the capture/compare registers (TIMx_CCRx) are used to latch the value of the counter after a transition detected by the corresponding ICx signal. When a capture occurs, the corresponding CCXIF flag (TIM14_SR register) is set. If a capture occurs while the CCxIF flag was already high, then the over-capture flag CCxOF (TIMx_SR register) is set. CCxIF can be cleared by software by writing it to ‘0’ or by reading the captured data stored. CCxOF is cleared when you write it to ‘0’.

The following example shows how to capture the counter value in TIM14_CCR1 when TI1 input rises. To do this, use the following procedure:

- Select the active output:TIM14_CCR1 must be linked to the TI1 input, so write the CC1S bits to 01 in the TIM14_CCMR1 register. As soon as CC1S becomes different from 00, the channel is configured in input and the TIM14_CCR1 register becomes read-only.
- Program the input filter duration you need with respect to the signal you connect to the timer (when the input is one of the TIx (ICxF bits in the TIM14_CCMRx register)). Let’s imagine that, when toggling, the input signal is not stable during at least 5 internal clock cycles. We must program a filter duration longer than these 5 clock cycles. We can validate a transition on TI1 when 8 consecutive samples with the new level have been detected (sampled at f_{DTS} frequency). Then write IC1F bits to ‘0011’ in the TIM14_CCMR1 register.
- Select the edge of the active transition on the TI1 channel by programming CC1P and CC1NP bits to ‘00’ in the TIMx_CCER register (rising edge in this case).
- Program the input prescaler. In our example, we wish the capture to be performed at each valid transition, so the prescaler is disabled (write IC1PS bits to ‘00’ in the TIMx_CCMR1 register).
- Enable capture from the counter into the capture register by setting the CC1E bit in the TIMx_CCER register.

If needed, enable the related interrupt request by setting the CC1IE bit in the TIMx_DIER register.

When an input capture occurs:

- The TIM14_CCR1 register gets the value of the counter on the active transition.

Puya Confidential

- CC1IF flag is set (interrupt flag). CC1OF is also set if at least two consecutive captures occurred whereas the flag was not cleared.
- An interrupt is generated depending on the CC1IE bit.

In order to handle the overcapture, it is recommended to read the data before the overcapture flag. This is to avoid missing an overcapture which could happen after reading the flag and before reading the data.

Note:IC interrupt requests can be generated by software by setting the corresponding CCxG bit in the TIMx_EGR register.

22.3.5. Forced output mode

In output mode (CCxS bits = 00 in the TIM14_CCMRx register), each output compares signal (OCxREF and then OCx) can be forced to active or inactive level directly by software, independently of any comparison between the output compare register and the counter.

To force an output compare signal (OCXREF/OCx) to its active level, you just need to write101 in the OCxM bits in the corresponding TIM14_CCMRx register. Thus OCXREF is forced high (OCxREF is always active high) and OCx get opposite value to CCxP polarity bit.

For example:CCxP=0 (OCx active high) => OCx is forced to high level.

The OCxREF signal can be forced low by writing the OCxM bits to 100 in the TIM14_CCMRx register.

Anyway, the comparison between the TIM14_CCRx shadow register and the counter is still performed and allows the flag to be set. This is described in the output compare mode section below.

22.3.6. Output compare mode

This function is used to control an output waveform or indicating when a period of time has elapsed.

When a match is found between the capture/compare register and the counter, the output compare function:

- Assigns the corresponding output pin to a programmable value defined by the output compare mode (OCxM bits in the TIM14_CCMRx register) and the output polarity (CCxP bit in the TIMx_CCER register). The output pin can keep its level (OCXM=000), be set active (OCxM=001), be set inactive (OCxM=010) or can toggle (OCxM=011) on match.
- Sets a flag in the interrupt status register (CCxIF bit in the TIMx_SR register).
- Generates an interrupt if the corresponding interrupt mask is set (CCxIE bit in the TIM14_DIER register).

The TIM14_CCRx registers can be programmed with or without preload registers using the OCxPE bit in the TIM14_CCMRx register.

In output compare mode, the update event UEV has no effect on OCxREF and OCx output. The timing resolution is one count of the counter. Output compare mode can also be used to output a single pulse (in one pulse mode).

Puya Confidential

The TIMx_CCRx register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled (OCxPE= ’0’, else TIMx_CCRx shadow register is updated only at the next update event UEV). An example is given in the figure below.

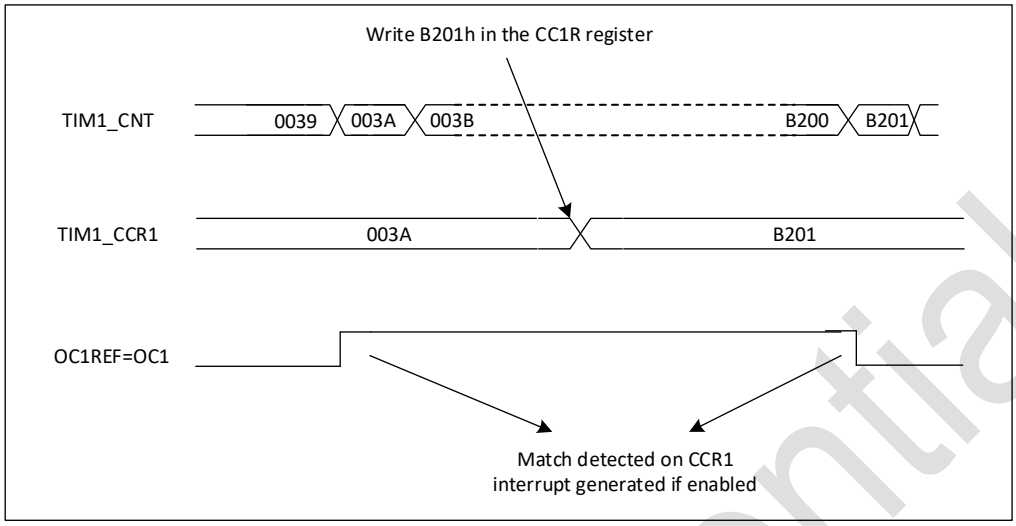


Figure 22-12 Output compare mode, toggle on OC1

22.3.7. PWM mode

Pulse width modulation (PWM) mode allows you to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register.

The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing ‘110’ (PWM mode 1) or ‘111’ (PWM mode 2) in the OCxM bits in the TIM14_CCMRx register. You must enable the corresponding preload register by setting the OCxPE bit in the TIM14_CCMRx register, and eventually the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIM14_CR1 register.

As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, all registers must be initialized by setting the UG bit in the TIM14_EGR register.

OCx polarity is software programmable using the CCxP bit in the TIM14_CCER register. It can be programmed as active high or active low. The OCx output is enabled by the CCxE bit in the TIM14_CCER register.

In PWM mode (1 or 2), TIM14_CNT and TIM14_CCRx are always compared to determine whether TIM14_CNT ≤ TIM14_CCRx.

The timer is able to generate PWM in edge-aligned mode.

PWM edge-aligned mode

In the following example, we consider PWM mode 1. The reference PWM signal OCxREF is high as long as TIM14_CNT < TIMx_CCRx else it becomes low. If the compare value in TIM14_CCRx is greater than the auto-reload value (in TIM14_ARR) then OCxREF is held at ‘1’. If the compare value is 0 then OCxRef is held at ‘0’.

Puya Confidential

Figure below shows some edge-aligned PWM waveforms in an example where TIMx_ARR=8.

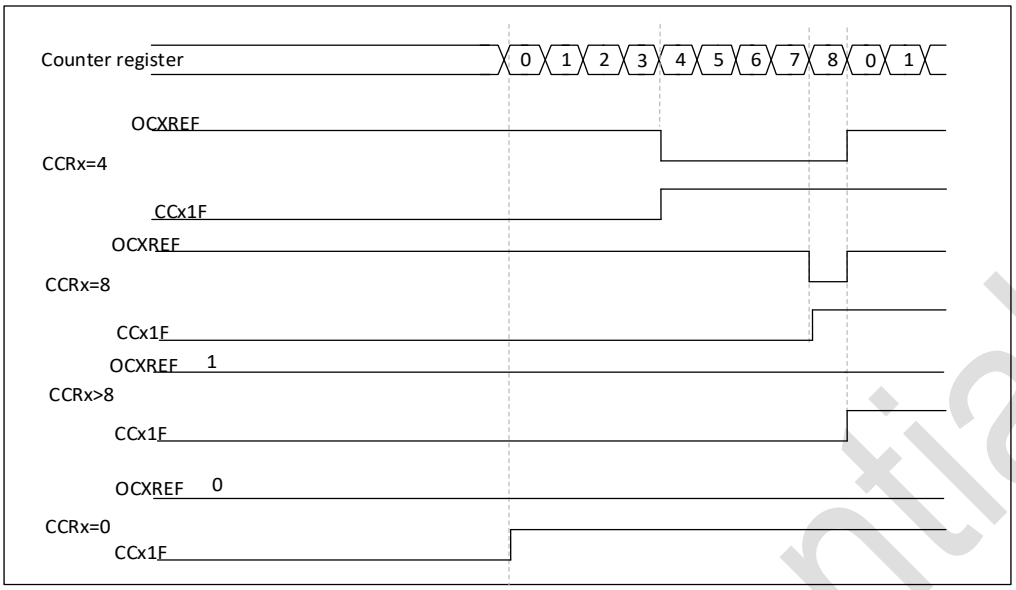


Figure 22-13 Edge-aligned PWM waveforms (ARR=8)

22.3.8. One-pulse mode

One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. One-pulse mode is selected by setting the OPM bit in the TIMx_CR1 register. This makes the counter stop automatically at the next update event UEV.

A pulse can be correctly generated only if the compare value is different from the counter initial value. Before starting (when the timer is waiting for the trigger), the configuration must be:

In upcounting:CNT < CCRx ≤ ARR (in particular, 0 < CCRx)

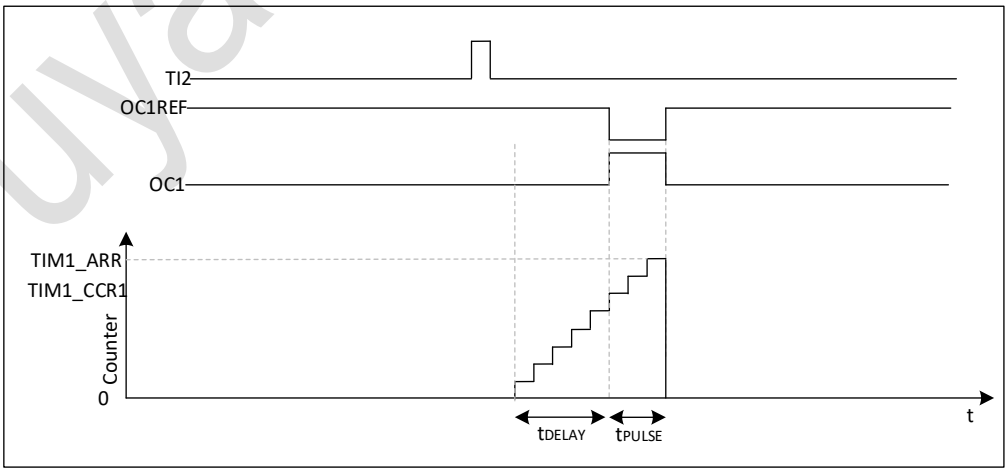


Figure 22-14 Example of one pulse mode.

For example, one may want to generate a positive pulse on OC1 with a length of t_{PULSE} and after a delay of t_{DELAY} as soon as a positive edge is detected on the TI2 input pin.

Assume TI2FP2 as the trigger:

Puya Confidential

- Map TI2FP2 on TI2 by writing CC2S=01 in the TIMx_CCMR1 register.
- TI2FP2 must detect a rising edge, write CC2P=0 in the TIMx_CCER register.
- Configure TI2FP2 as trigger for the slave mode controller (TRGI) by writing TS=110 in the TIMx_SMCR register.
- TI2FP2 is used to start the counter by writing SMS to '110' in the TIMx_SMCR register (trigger mode).

The OPM waveform is defined by writing the compare registers (taking into account the clock frequency and the counter prescaler).

- The t_{DELAY} is defined by the value written in the TIMx_CCR1 register.
- The t_{PULSE} is defined by the difference between the auto-reload value and the compare value (TIMx_ARR - TIMx_CCR1).
- Let's say one want to build a waveform with a transition from '0' to '1' when a compare match occurs and a transition from '1' to '0' when the counter reaches the auto-reload value. To do this PWM mode 2 must be enabled by writing OC1M=111 in the TIMx_CCMR1 register. Optionally the preload registers can be enabled by writing OC1PE='1' in the TIMx_CCMR1 register and ARPE in the TIMx_CR1 register. In this case one has to write the compare value in the TIMx_CCR1 register, the auto-reload value in the TIMx_ARR register, generate an update by setting the UG bit and wait for external trigger event on TI2. CC1P is written to '0' in this example.

In our example, the DIR and CMS bits in the TIMx_CR1 register should be low.

The user only wants one pulse (Single mode), so '1' must be written in the OPM bit in theTIMx_CR1 register to stop the counter at the next update event (when the counter rolls over from the auto-reload value back to 0). When OPM bit in the TIMx_CR1 register is set to '0', so the repetitive mode is selected.

22.3.9. Timer synchronization

The TIMx timers are linked together internally for timer synchronization or chaining. When one Timer is configured in Master mode, it can reset, start, stop or clock the counter of another Timer configured in Slave mode.

22.3.10. Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the TIMx counter either continues to work normally or stops, depending on DBG_TIMx_STOP configuration bit in DBG module.

22.4. TIM14 registers

22.4.1. TIM14 control register 1 (TIM14_CR1)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Puya Confidential

Res	Res	Res	Res	Res	Res	CKD [1:0]	ARPE	Res	Res	Res	OPM	URS	UDIS	CEN
-	-	-	-	-	-	RW	RW	-	-	-	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9:8	CKD [1:0]	RW	00	Clock division This bit-field indicates the division ratio between the timer clock (CK_INT) frequency and sampling clock (t_{DTS}) 00:t_{DTS} = t_{CK_INT} 01:t_{DTS} = 2 x t_{CK_INT} 10:t_{DTS} = 4 x t_{CK_INT} 11:Reserved, do not program this value
7	ARPE	RW	0	Auto-reload preload enable 0:TIM14_ARR register is not buffered 1:TIM14_ARR register is buffered
6:4	Reserved	-	-	Reserved
3	OPM	RW	0	One pulse mode 0:Counter is not stopped at update event 1:Counter stops counting at the next update event (clearing the bit CEN)
2	URS	RW	0	Update request source This bit is set and cleared by software to select the UEV event sources. 0:Any of the following events generate an update interrupt or DMA request if enabled. These events can be: − Counter overflow/underflow − Setting the UG bit 1:Only counter overflow/underflow generates an update interrupt or DMA request if enabled
1	UDIS	RW	0	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0:UEV enabled. The Update (UEV) event is generated by one of the following events: − Counter overflow/underflow − Setting the UG bit Buffered registers are then loaded with their preload values. 1:UEV disabled. The update event is not generated, shadow registers keep their value (ARR, PSC, CCRx).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				However, the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	RW	0	Counter enable 0:Counter disabled 1:Counter enabled Note:external clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However, trigger mode can set the CEN bit automatically by hardware.

22.4.2. TIM14 DMA/interrupt enable register (TIM14_DIER)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res														CC1IE	UIE
-														RW	RW

Bit	Name	R/W	Reset Value	Function
31:2	Reserved	-	-	Reserved
1	CC1IE	RW	0	CC1IE:Capture/Compare 1 interrupt enable 0:Capture/Compare 1 interrupt disabled 1:Capture/Compare 1 interrupt enabled
0	UIE	RW	0	UIE:Update interrupt enable 0:Update interrupt disabled. 1:Update interrupt enabled

22.4.3. TIM14 status register (TIM14_SR)

Address offset:0x010

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res											IC1IF	Res	Res	Res	IC1IR
-											RC_W0	-	-	-	RC_W0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res						CC1OF	Res							CC1F	UIF
-						RC_W0	-							RC_W0	RC_W0

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:21	Reserved	-	-	Reserved
20	IC1IF	RC_W0	0	Falling edge capture 1 flag This flag is set by hardware only when the corresponding channel is configured in input capture mode and triggered by falling edge. It is cleared by software or reading TIMx_CCR1. 0:No falling edge capture occurs; 1:A falling edge capture event occurs.
19:17	Reserved	-	-	Reserved
16	IC1IR	RC_W0	0	Rising edge capture 1 flag This flag is set by hardware only when the corresponding channel is configured in input capture mode and triggered by rising edge. It is cleared by software or reading TIMx_CCR1. 0:No rising edge capture occurs; 1:A rising edge capture event occurs.
15:10	Reserved	-	-	Reserved
9	CC1OF	RC_W0	0	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to ‘0’. 0:No overcapture has been detected. 1:The counter value has been captured in TIM1_CCR1 register while CC1IF flag was already set.
8:2	Reserved	-	-	Reserved
1	CC1IF	RC_W0	0	Capture/Compare 1 interrupt flag If channel CC1 is configured as output: This flag is set by hardware when the counter matches the compare value. It is cleared by software. 0:No match; 1:The content of the counter TIM14_CNT matches the content of the TIM14_CCR1 register. If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading the TIM14_CCR1 register. 0:No input capture occurred

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:The counter value has been captured in TIM14_CCR1 register (An edge has been detected on IC1 which matches the selected polarity)
0	UIF	RC_W0	0	Update interrupt flag. This bit is set by hardware on an update event. It is cleared by software. 0:No update occurred. 1:Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow and if UDIS '0' in the TIMx_CR1 register. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0 in the TIMx_CR1 register.

22.4.4. TIM14 event generation register (TIM14_EGR)

Address offset:0x14

Reset value:0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res														CC1G	UG
-														W	W

Bit	Name	R/W	Reset Value	Function
31:2	Reserved	-	-	Reserved
1	CC1G	W	0	Capture/compare 1 generation. This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A capture/compare event is generated on channel CC1: If channel CC1 is configured as input: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input: The current value of the counter is captured in TIM14_CCR1 register. The CC1IF flag is set, the corresponding interrupt is sent if enabled. The CC1OF flag is set if theCC1IF flag was already set.

Puya Confidential

0	UG	W	0	Update generation. This bit can be set by software, it is automatically cleared by hardware. 0:No action 1:Reinitialize the counter and generates an update of the registers. Note that the counter of the prescaler is also cleared to 0 (but the prescaler coefficient is unchanged).

22.4.5. TIM14 capture/compare mode register 1 (TIM1_CCMR1)

Address offset:0x18

Reset value:0x00000000

Output compare mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res								Res	OC1M [2:0]			OC1PE	Res	CC1S [1:0]	
-								-	RW	RW	RW	RW	-	RW	RW

Bit	Name	R/W	Reset Value	Function
31:7	Reserved	-	-	Reserved
6:4	OC1M [2:0]	RW	00	Output compare 1 mode These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000:Frozen. The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on the outputs. 001:Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 010:Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 011:Toggle OC1REF toggles when TIMx_CNT=TIMx_CCR1. 100:Force inactive level OC1REF is forced low. 101:Force active level OC1REF is forced high. 110:PWM mode 1 In upcounting, channel 1 is active as long as TIMx_CNT<TIMx_CCR1 else inactive. In downcounting, channel 1

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				is inactive (OC1REF= ‘0’) as long as TIMx_CNT>TIMx_CCR1 else active (OC1REF=’1’). 111:PWM mode 2 Channel 1 is inactive as long as TIMx_CNT < TIMx_CCR1 else active. Note:In PWM mode 1 or 2, the OCREF level changes when the result of the comparison changes or when the output compare mode switches from frozen to PWM mode.
3	OC1PE	RW	0	Output Compare 1 preload enable 0:Preload register on TIM14_CCR1 disabled. TIM14_CCR1 can be written at any time, the new value is taken in account immediately. 1:Preload register on TIM14_CCR1 enabled. Read/Write operations access the preload register. TIM14_CCR1 preload value is loaded in the active register at each update event.
2	Res	-	-	Reserved
1:0	CC1S [1:0]	RW	00	Capture/Compare 1 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:Reserved; 11:Reserved. Note:CC1S bits are writable only when the channel is OFF (CC1E = ‘0’ in TIM14_CCER).

Input capture mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res								IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
-								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:21	Reserved	-	-	Reserved
7:4	IC1F [3:0]	RW	0000	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter

Priya Confidential

Bit	Name	R/W	Reset Value	Function
				in which N consecutive events are needed to validate a transition on the output: 0000:No filter, sampling is done at f_{DTS} 0001:$f_{SAMPLING}$ = f_{CK_INT}, N = 2 0010:$f_{SAMPLING}$ = f_{CK_INT}, N = 4 0011:$f_{SAMPLING}$ = f_{CK_INT}, N = 8 0100:$f_{SAMPLING}$ = f_{DTS}/2, N = 6 0101:$f_{SAMPLING}$ = f_{DTS}/2, N = 8 0110:$f_{SAMPLING}$ = f_{DTS}/4, N = 6 0111:$f_{SAMPLING}$ = f_{DTS}/4, N = 8 1000:$f_{SAMPLING}$ = f_{DTS}/8, N = 6 1001:$f_{SAMPLING}$ = f_{DTS}/8, N = 8 1010:$f_{SAMPLING}$ = f_{DTS}/16, N = 5 1011:$f_{SAMPLING}$ = f_{DTS}/16, N = 6 1100:$f_{SAMPLING}$ = f_{DTS}/16, N = 8 1101:$f_{SAMPLING}$ = f_{DTS}/32, N = 5 1110:$f_{SAMPLING}$ = f_{DTS}/32, N = 6 1111:$f_{SAMPLING}$ = f_{DTS}/32, N = 8
3:2	IC1PSC [1:0]	RW	00	Input capture 1 prescaler This bit-field defines the ratio of the prescaler acting on CC1 input (IC1). The prescaler is reset as soon as CC1E='0' (TIMx_CCER register). 00:no prescaler, capture is done each time an edge is detected on the capture input; 01:capture is done once every 2 events 10:capture is done once every 4 events 11:capture is done once every 8 events
1:0	CC1S [1:0]	RW	00	CC1S [1:0]:capture/compare 1 selection. This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:Reserved 11:Reserved Note:CC1S bits are writable only when the channel is OFF (CC1E = ‘0’ in TIM14_CCER).

22.4.6. TIM14 capture/compare enable register (TIM14_CCER)

Address offset:0x20

Puya Confidential

Reset value:0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC1NP	Res	CC1P	CC1E
-	-	-	-	-	-	-	-	-	-	-	-	RW	-	RW	RW

Bit	Name	R/W	Reset Value	Function
31:4	Reserved	-	-	Reserved
3	CC1NP	RW	0	Capture/compare 1 complementary output polarity. CC1 channel configured as output: CC1NP must be kept cleared. CC1 channel configured as input: CC1NP bit is used in conjunction with CC1P to defineTI1FP1 polarity (refer to CC1P description).
2	Reserved	-	-	Reserved
1	CC1P	RW	0	Capture/compare 1 output polarity CC1 channel configured as output: 0:OC1 active high. 1:OC1 active low. CC1 channel configured as input: CC1NP/CC1P bits select the active polarity of TI1FP1 and TI2FP1 for trigger or capture operations. 00:Non-inverted/rising edge: TIxFP1 rising edge valid (triggered in capture mode); 01:Inverted/falling edge: TIxFP1 falling edge active (triggered in capture mode); 10:reserved, do not use this configuration. 11:non-inverted/double edge TIxFP1 both rising and falling edges are valid (triggered in capture mode);
0	CC1E	RW	0	Capture/Compare 1 output enable **CC1 channel configured as output:** 0:Off - OC1 is not active 1:On - OC1 signal is output on the corresponding output pin. **CC1 channel configured as input:** This bit determines if a capture of the counter value can actually be done into the input capture/compare register 1 (TIMx_CCR1) or not.

Puya Confidential

				0:Capture disabled. 1:Capture enabled.

CcxE bit	OCx output state
0	Output Disabled (OCx=0, OCx_EN=0)
1	OCx= OCxREF+Polarity, OCx_EN=1

22.4.7. TIM14 counter (TIM14_CNT)

Address offset:0x24

Reset value:0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CNT [15:0]	RW	0	Counter value

22.4.8. TIM14 prescaler (TIM14_PSC)

Address offset:0x28

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSC [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PSC [15:0]	RW	0	Prescaler value The counter clock frequency (CK_CNT) is equal to fCK_PSC / (PSC [15:0] +1). PSC contains the value to be loaded in the active pre-scaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in reset mode).

Puya Confidential

22.4.9. TIM14 auto-reload register (TIM14_ARR)

Address offset:0x2c

Reset value:0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ARR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	ARR [15:0]	RW	FFFF	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. Refer to 24.3.1:Time-base unit for more details about ARR update and behavior. The counter is blocked while the auto-reload value is null.

22.4.10. TIM14 capture/compare register 1 (TIM14_CCR1)

Address offset:0x34

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR1 [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CCR1 [15:0]	RW	0	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).

22.4.11. TIM14 option register (TIM14_OR)

Address offset:0x50

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res														TI1_RMP	
-														RW	

Bit	Name	R/W	Reset Value	Function
31:2	Reserved	-	-	Reserved
1:0	TI1_RMP	RW	0	Timer input 1 remap Set and cleared by software. 00:TIM14 channel 1 is connected to GPIO. For details, refer to the multiplexing function of the data sheet. 01:TIM14 channel 1 is connected to RTCCLK. 10:TIM14 channel 1 is connected to HSE/32 clock 11:TIM14 channel 1 is connected to the MCU clock output (MCO). This configuration is determined by setting the MCO [2:0] of the RCC_CFG register.

Puya Confidential

23. General-purpose timers (TIM15/16/17)

23.1. Introduction

The TIM15/TIM16/TIM17 timers consist of a 16-bit auto-reload counter driven by a programmable prescaler. They may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM, complementary PWM with dead-time insertion).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers. The TIM15/TIM16/TIM17 timers are completely independent, and do not share any resources. TIM15_16_17 and other Timers may be synchronized.

23.2. TIM15 main features

- 16-bit up auto-reload upcounter
- 16-bit programmable prescaler used to divide (also on the fly) the counter clock frequency by any factor between 1 and 65535
- Up to 2 independent channels
  - Input capture
  - Output compare
  - PWM generation (edge-aligned mode)
  - One-pulse mode output
- Complementary outputs with programmable dead-time (for channel 1 only)
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Repetition counter to update the timer registers only after a given number of cycles of the counter.
- Break input to put the timer’s output signals in reset state or in a known state.
- Interrupt/DMA generation on the following events:
  - Update:counter overflow, counter initialization (by software or internal/external trigger)
  - Trigger event
  - Input capture
  - Output compare
  - Break input

Puya Confidential

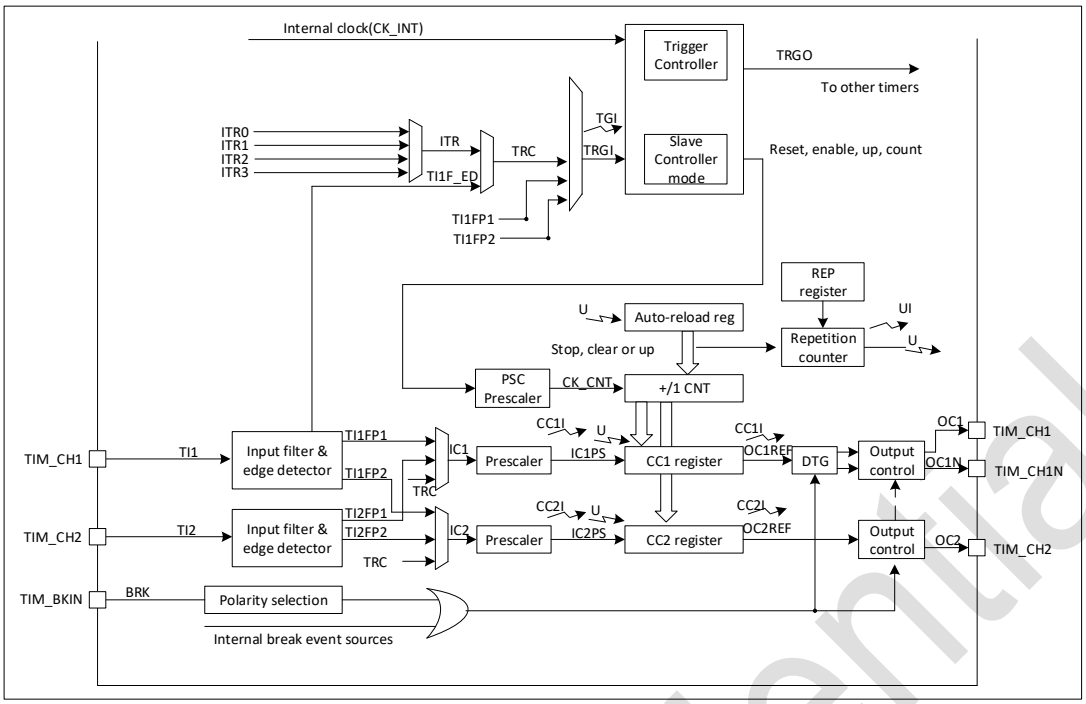


Figure 23-1 General-purpose timers (TMI15) block diagram

23.3. TIM16/17 main features

- 16-bit up auto-reload upcounter
- 16-bit programmable prescaler used to divide (also on the fly) the counter clock frequency by any factor between 1 and 65535
- An independent channel for
  - Input capture
  - Output compare
  - PWM generation (edge-aligned mode)
  - One-pulse mode output
- Complementary outputs with programmable dead-time
- Repetition counter to update the timer registers only after a given number of cycles of the counter.
- Break input to put the timer's output signals in reset state or in a known state.
- Interrupt/DMA generation on the following events:
  - Update:counter overflow
  - Input capture
  - Output compare
  - Break input

Puya Confidential

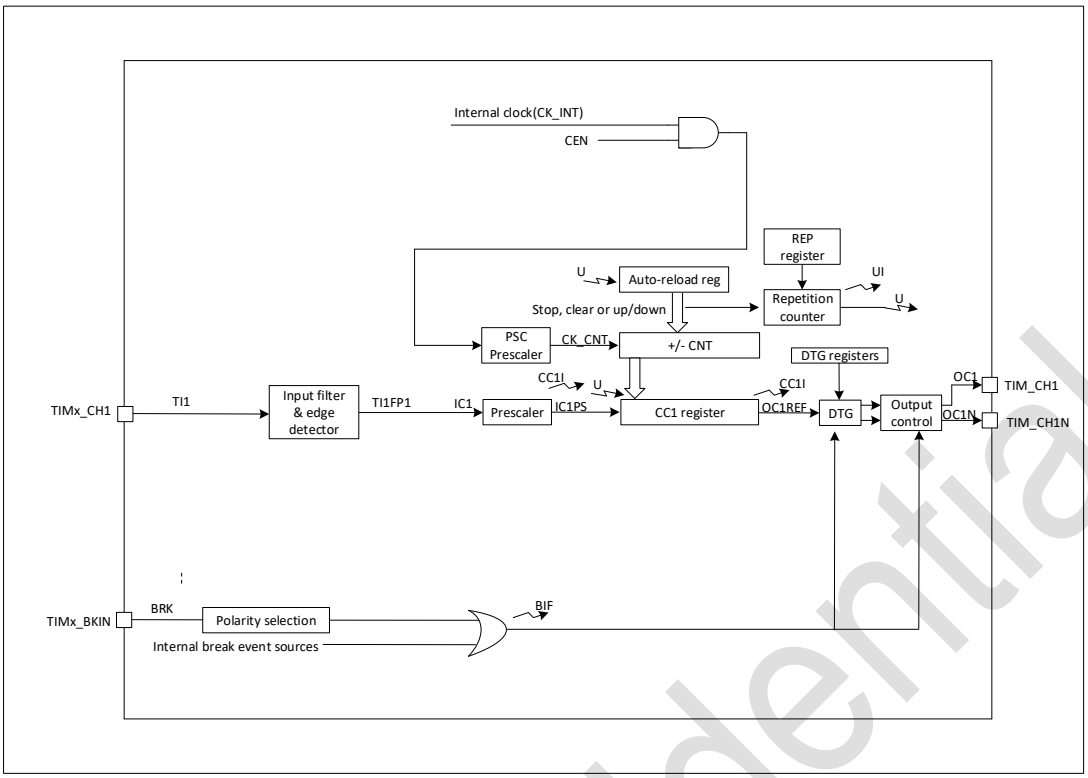


Figure 23-2 General-purpose timers (TMI16/17) block diagram

23.4. TIM15/16/17 functional description

23.4.1. Time-base unit

The main block of the programmable advanced-control timer is a 16-bit counter with its related auto-reload register. This counter can count up. The counter clock can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.

The time-base unit includes:

- Counter register (TIMx_CNT)
- Prescaler register (TIMx_PSC)
- Auto-reload register (TIMx_ARR)
- Repetition counter register (TIMx_RCR)

The auto-load register is preloaded. Writing to or reading from the auto-load register accesses the preload register. The content of the preload register is transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload preload enable bit (ARPE) in TIMx_CR1 register. The update event is sent when the counter reaches the overflow value and if the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software.

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in TIMx_CR1 register is set.

Note that the counter starts counting 1 clock cycle after setting the CEN bit in the TIMx_CR register.

Puya Confidential

Prescaler description

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIMx_PSC register). It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

Figures below give some examples of the counter behavior when the prescaler ratio is changed on the fly:

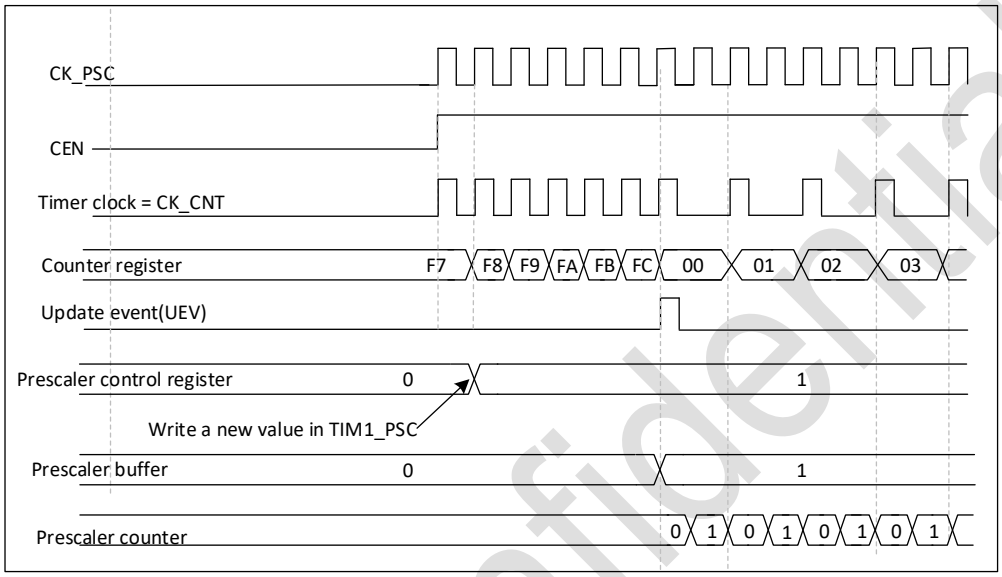


Figure 23-3 Counter timing diagram with prescaler division change from 1 to 2

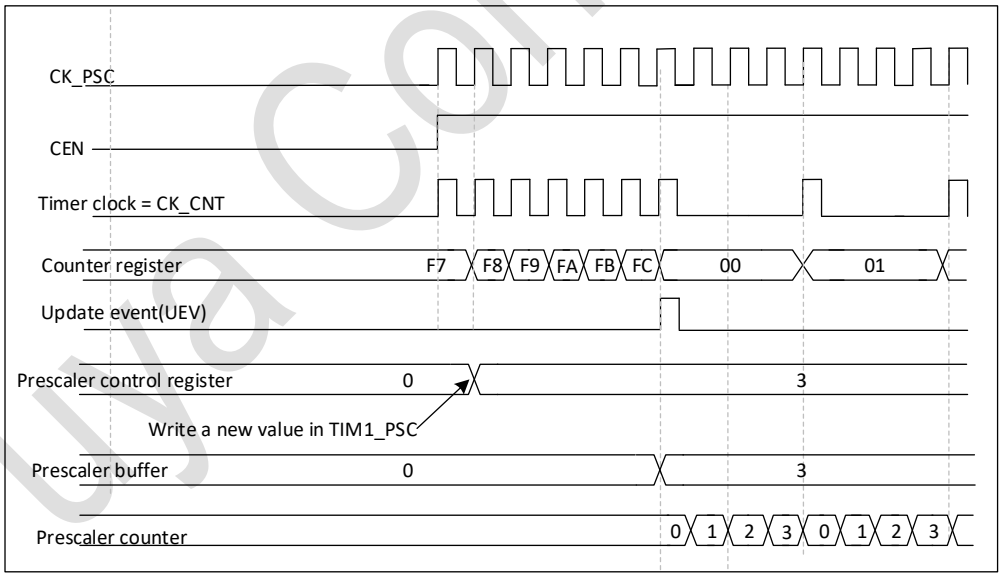


Figure 23-4 Counter timing diagram with prescaler division change from 1 to 4

23.4.2. Counter mode

Upcounting mode

In upcounting mode, the counter counts from 0 to the auto-reload value, then restarts from 0 and generates a counter overflow event.

Puya Confidential

If the repetition counter is used, the update event (UEV) is generated after upcounting is repeated for the number of times programmed in the repetition counter register plus one (TIMx_RCR+1). Else the update event is generated at each counter overflow.

Setting the UG bit in the TIMx_EGR register (by software or by using the slave mode controller) also generates an update event.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts from 0, as well as the counter of the prescaler (but the prescale rate does not change). In addition, if the URS bit (update request selection) inTIMx_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag (thus no interrupt or DMA request is sent). This is to avoid generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, all the registers are updated and the update flag (UIF bit in TIMx_SR register) is set (depending on the URS bit).

- The repetition counter is reloaded with the content of TIMx_RCR register.
- The auto-reload shadow register is updated with the preload value (TIMx_ARR).
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register).

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR=0x36.

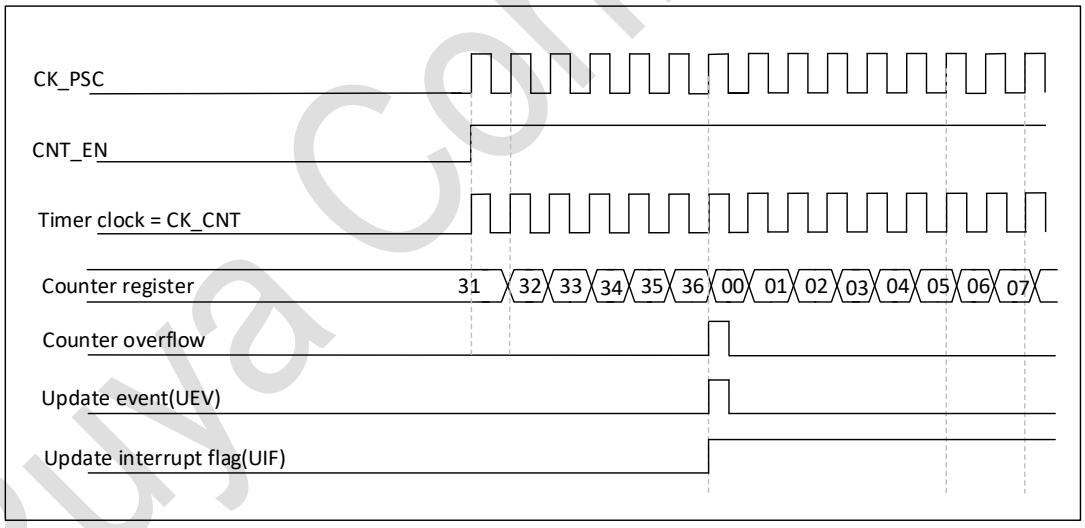


Figure 23-5 Counter timing diagram, internal clock divided by 1

Puya Confidential

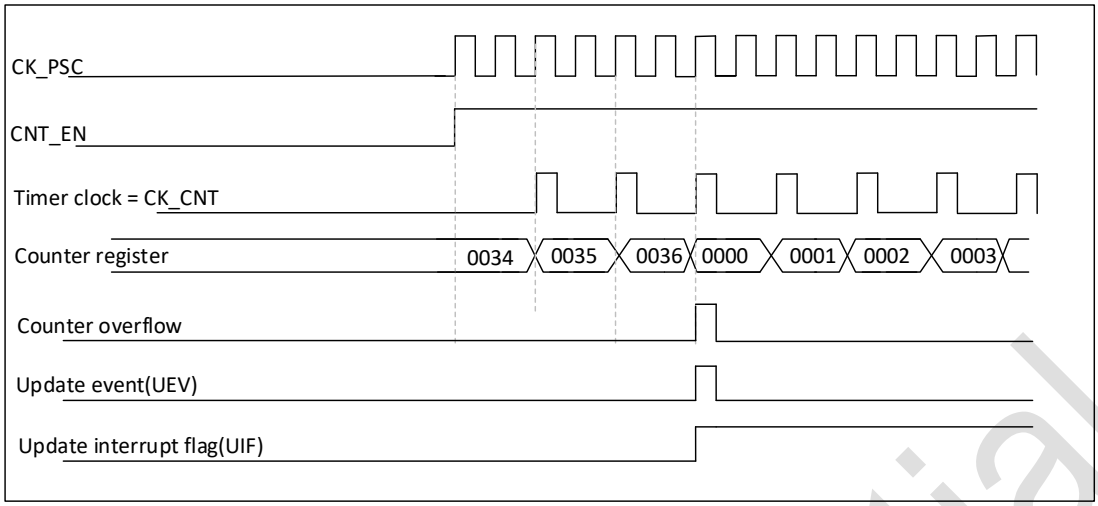


Figure 23-6 Counter timing diagram, internal clock divided by 2

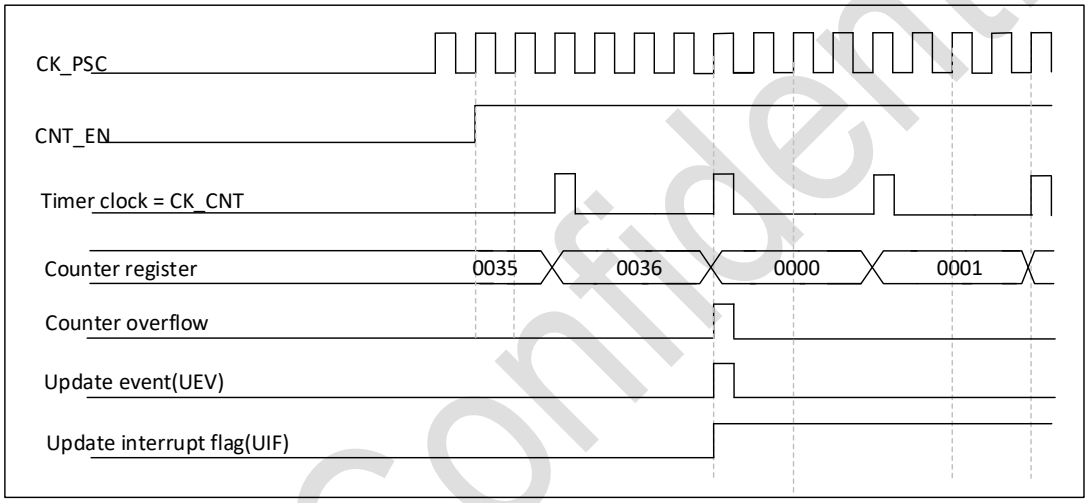


Figure 23-7 Counter timing diagram, internal clock divided by 4

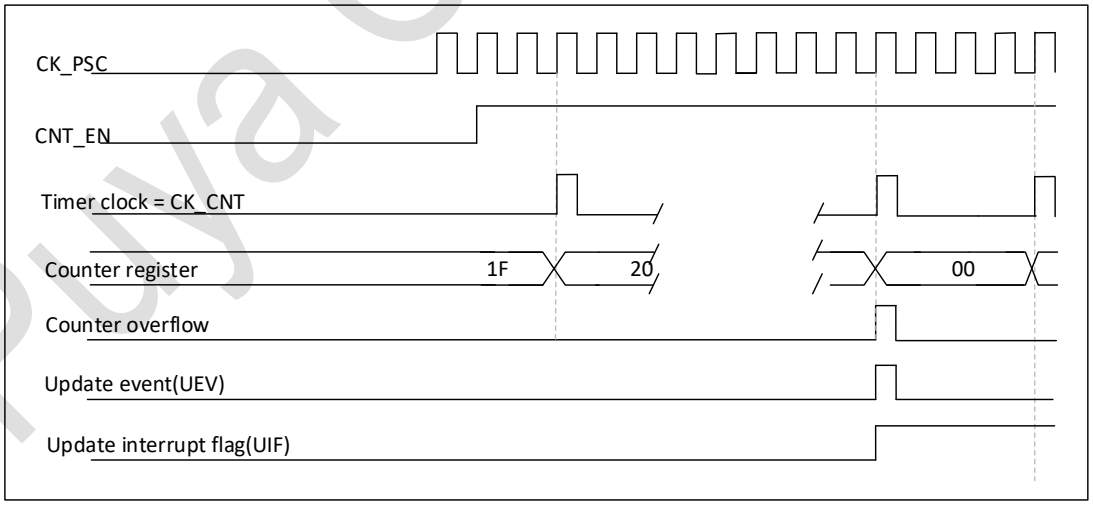


Figure 23-8 Counter timing diagram, internal clock divided by N

Puya Confidential

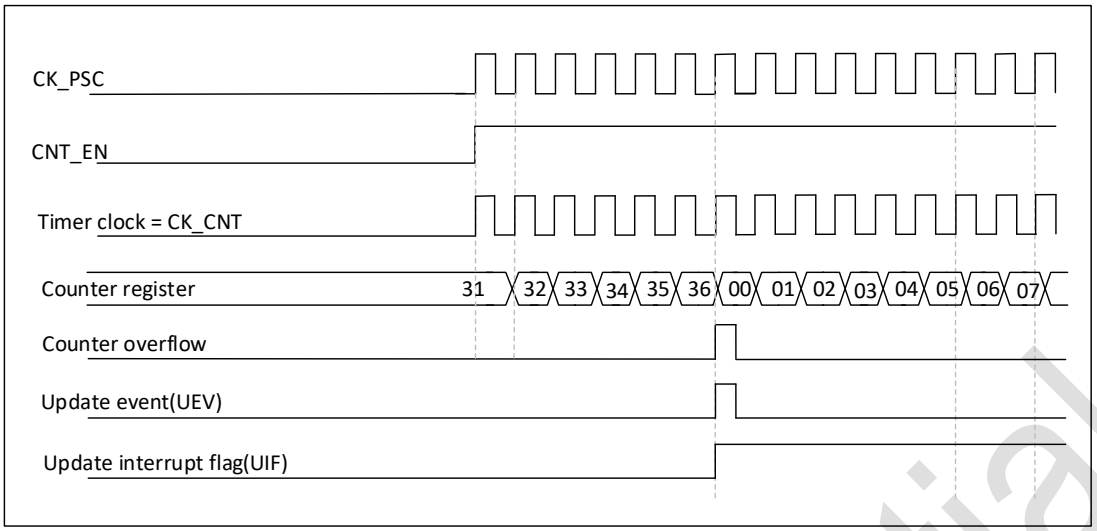


Figure 23-9 Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded)

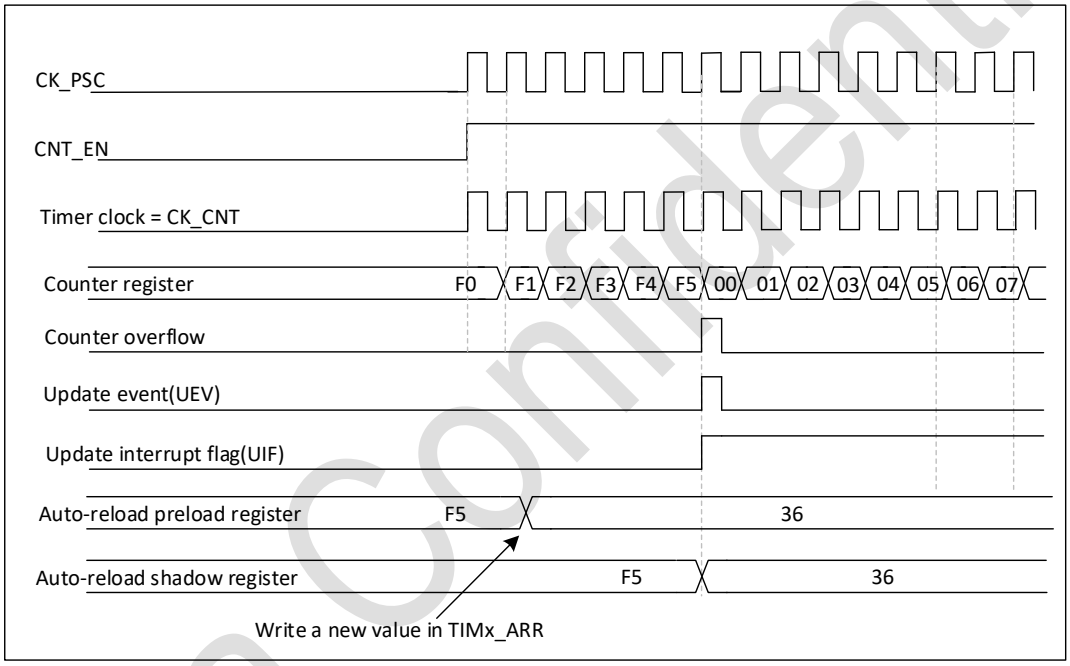


Figure 23-10 Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded)

23.4.3. Repetition counter

Time-base unit describes how the update event (UEV) is generated with respect to the counter overflows. It is actually generated only when the repetition counter has reached zero. This can be useful when generating PWM signals.

This means that data are transferred from the preload registers to the shadow registers (TIMx_ARR auto-reload register, TIMx_PSC prescaler register, but also TIMx_CCRx capture/compare registers in compare mode) every N + 1 counter overflows, where N is the value in the TIMx_RCR repetition counter register.

The repetition counter is decremented when the following conditions are true:

- At each counter overflow in upcounting mode

The repetition counter is an auto-reload type; the repetition rate is maintained as defined by the TIMx_RCR register value. When the update event is generated by software (by setting the UG bit in

Puya Confidential

TIMx_EGR register) or by hardware through the slave mode controller, it occurs immediately whatever the value of the repetition counter is, and the repetition counter is reloaded with the content of the TIMx_RCR register.

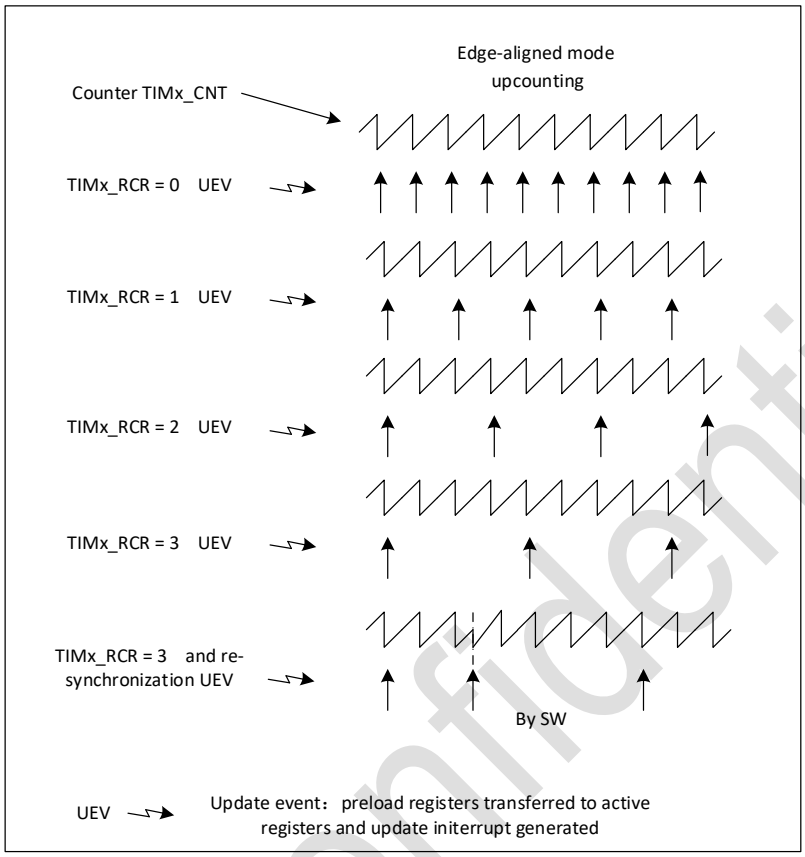


Figure 23-11 Update rate examples depending on mode and TIMx_RCR register settings

23.4.4. Clock sources

The counter clock can be provided by the following clock sources:

- Internal clock (CK_INT)
- External clock mode 1:External input pin (TMI15 only)
- Internal trigger inputs (ITRx):using one timer as prescaler for another timer. For example, the user can configure Timer 1 to act as a prescaler for Timer 2. (TMI15 only)

Internal clock source (CK_INT)

If the slave mode controller is disabled, then the CEN, DIR (in the TIMx_CR1 register) and UG bits (in the TIMx_EGR register) are actual control bits and can be changed only by software. As soon as the CEN bit is written to 1, the prescaler is clocked by the internal clock CK_INT.

Puya Confidential

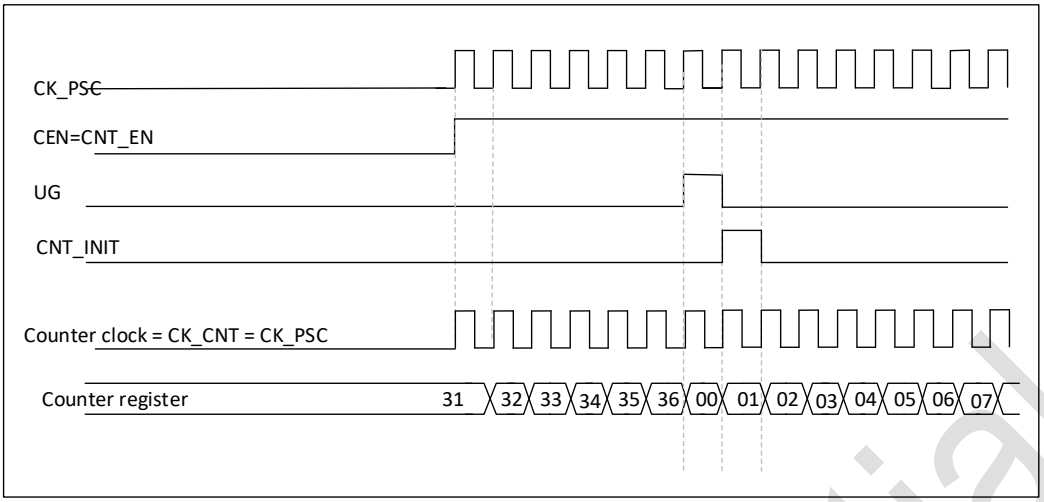


Figure 23-12 Control circuit in normal mode, internal clock divided by 1

External clock source mode 1 (TIM15 only)

This mode is selected when SMS=111 in the TIMx_SMCR register. The counter can count at each rising or falling edge on a selected input.

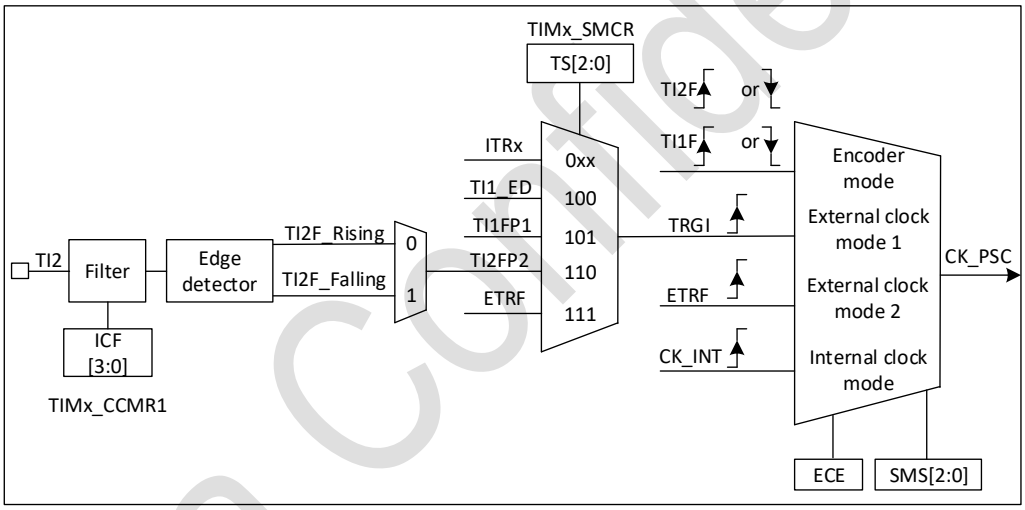


Figure 23-13 TI2 external clock connection example

For example, to configure the upcounter to count in response to a rising edge on the TI2 input, use the following procedure:

1. Configure channel 2 to detect rising edges on the TI2 input by writing CC2S = ‘01’ in the TIMx_CCMR1 register.

2. Configure the input filter duration by writing the IC2F [3:0] bits in the TIMx_CCMR1 register (if no filter is needed, keep IC2F=0000).

3. Select rising edge polarity by writing CC2P=0 in the TIMx_CCER register.

4. Configure the timer in external clock mode 1 by writing SMS=111 in the TIMx_SMCR register.

5. Select TI2 as the trigger input source by writing TS=110 in the TIMx_SMCR register;

6. Enable the counter by writing CEN=1 in the TIMx_CR1 register.

Note:The capture prescaler is not used for triggering, so it does not need to be configured.

When a rising edge occurs on TI2, the counter counts once and the TIF flag is set.

Puya Confidential

The delay between the rising edge of TI2 and the actual clock of the counter depends on the resynchronization circuit at the input of TI2.

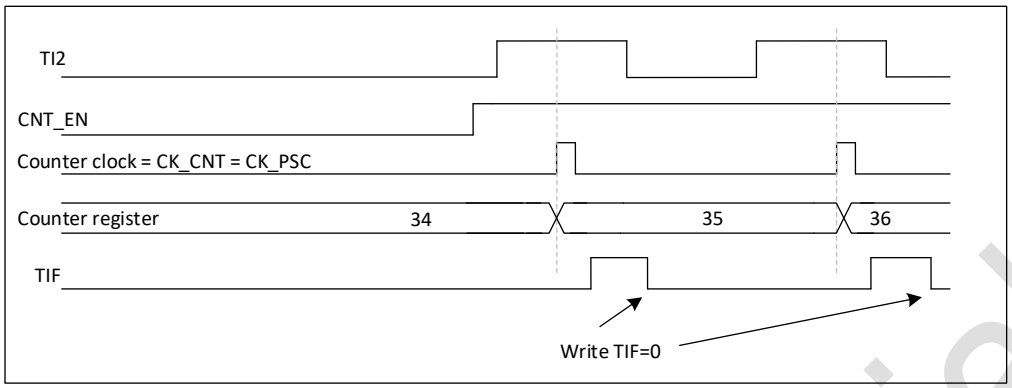


Figure 23-14 Control circuit in external clock mode 1

23.4.5. Capture/Compare channels

Each Capture/Compare channel is built around a capture/compare register (including a shadow register), an input stage for capture (with digital filter, multiplexing and prescaler) and an output stage (with comparator and output control).

The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detector with polarity selection generates a signal (TIxFPx) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (ICxPS).

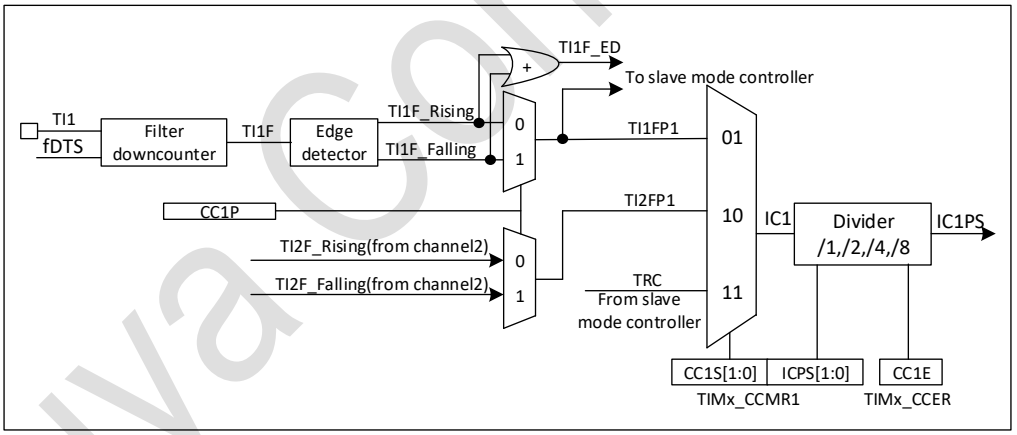


Figure 23-15 Capture/compare channel (example:channel 1 input stage)

The output stage generates an intermediate waveform which is then used for reference:OCxREF (active high). The polarity acts at the end of the chain.

Puya Confidential

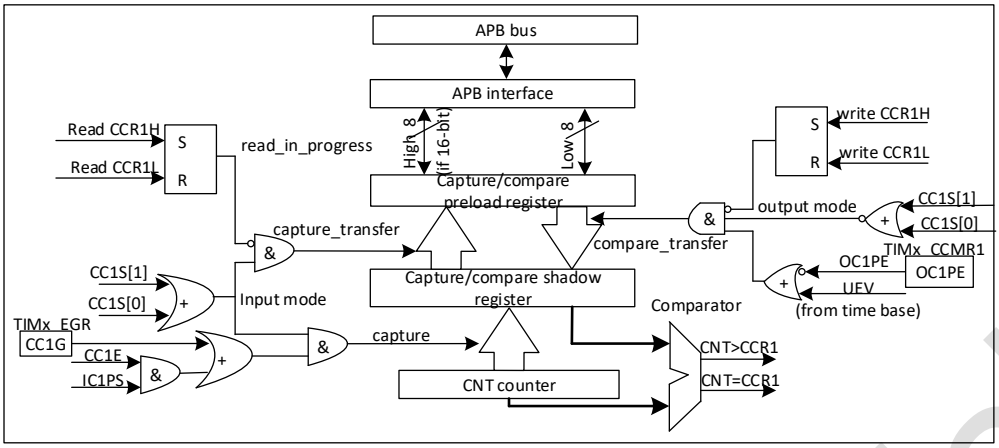


Figure 23-16 Capture/compare channel 1 main circuit

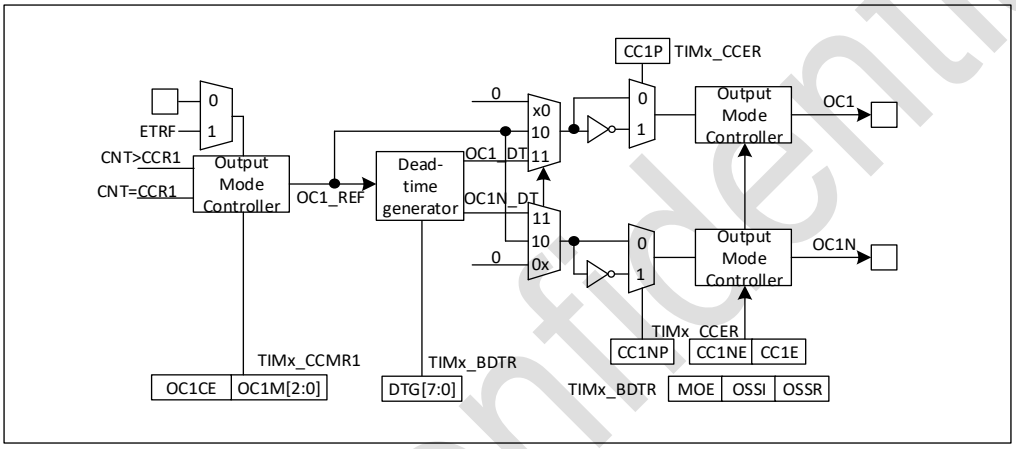


Figure 23-17 Output stage of capture/compare channel (channel 1 to 3)

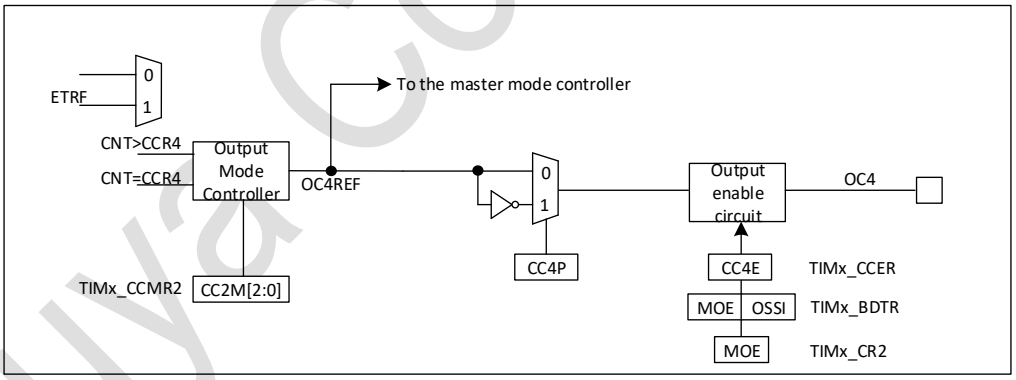


Figure 23-18 Output stage of capture/compare channel (channel 4)

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register.

In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

Puya Confidential

23.4.6. Input capture mode:

In Input capture mode, the capture/compare registers are used to latch the value of the counter after a transition detected by the corresponding ICx signal. When a capture occurs, the corresponding CCxIF flag (TIMx_SR register) is set and an interrupt or a DMA request can be sent if they are enabled. If a capture occurs while the CCxIF flag was already high, then the over-capture flag CCxOF (TIMx_SR register) is set. CCxIF can be cleared by software by writing it to '0' or by reading the captured data stored in the TIMx_CCRx register. CCxOF is cleared when you write it to '0'.

The following example shows how to capture the counter value in TIMx_CCR1 when TI1input rises. To do this, use the following procedure:

- Select the active input:TIMx_CCR1 must be linked to the TI1 input, so write the CC1S bits to 01 in the TIMx_CCMR1 register. As soon as CC1S becomes different from 00, the channel is configured in input and the TIMx_CCR1 register becomes read-only.
- Program the needed input filter duration with respect to the signal connected to the timer (by programming ICxF bits in the TIMx_CCMRx register if the input is a TIx input). Let's imagine that, when toggling, the input signal is not stable during at most 5 internal clock cycles. We must program a filter duration longer than these 5 clock cycles. We can validate a transition on TI1 when 8 consecutive samples with the new level have been detected (sampled at fDTS frequency). Then write IC1F bits to 0011 in the TIMx_CCMR1 register.
- Select the edge of the active transition on the TI1 channel by writing CC1P bit to 0 in the TIMx_CCER register (rising edge in this case).
- Program the input prescaler. In our example, we wish the capture to be performed at each valid transition, so the prescaler is disabled (write IC1PS bits to '00' in the TIMx_CCMR1 register).

When an input capture occurs:

- The TIMx_CCR1 register gets the value of the counter on the active transition.
- CC1IF flag is set (interrupt flag). CC1OF is also set if at least two consecutive captures occurred whereas the flag was not cleared.
- An interrupt is generated depending on the CC1IE bit.
- A DMA request is generated depending on the CC1DE bit.

In order to handle the overcapture, it is recommended to read the data before the overcapture flag. This is to avoid missing an overcapture which could happen after reading the flag and before reading the data.

Note:IC interrupt and/or DMA requests can be generated by software by setting the corresponding CCxG bit in the TIMx_EGR register.

23.4.7. PWM input mode (only for TIM15)

This mode is a particular case of input capture mode. The procedure is the same except:

- Two ICx signals are mapped on the same TIx input.
- These 2 ICx signals are active on edges with opposite polarity.

Puya Confidential

- One of the two TIxFP signals is selected as trigger input and the slave mode controller is configured in reset mode.

For example, user can measure the period (in TIMx_CCR1 register) and the duty cycle (in TIMx_CCR2 register) of the PWM applied on TI1 using the following procedure (depending on CK_INT frequency and prescaler value):

- Select the active input for TIMx_CCR1:write the CC1S bits to 01 in the TIMx_CCMR1 register (TI1 selected).
- Select the active polarity for TI1FP1 (used both for capture in TIMx_CCR1 and counter clear):write the CC1P bit to '0' (active on rising edge).
- Select the active input for TIMx_CCR2:write the CC2S bits to 10 in the TIMx_CCMR1 register (TI1 selected).
- Select the active polarity for TI1FP2 (used for capture in TIMx_CCR2):write the CC2Pbit to '1' (active on falling edge).
- Select the valid trigger input:write the TS bits to 101 in the TIMx_SMCR register (TI1FP1 selected).
- Configure the slave mode controller in reset mode:write the SMS bits to 100 in the TIMx_SMCR register.
- Enable the captures:write the CC1E and CC2E bits to '1' in the TIMx_CCER register.

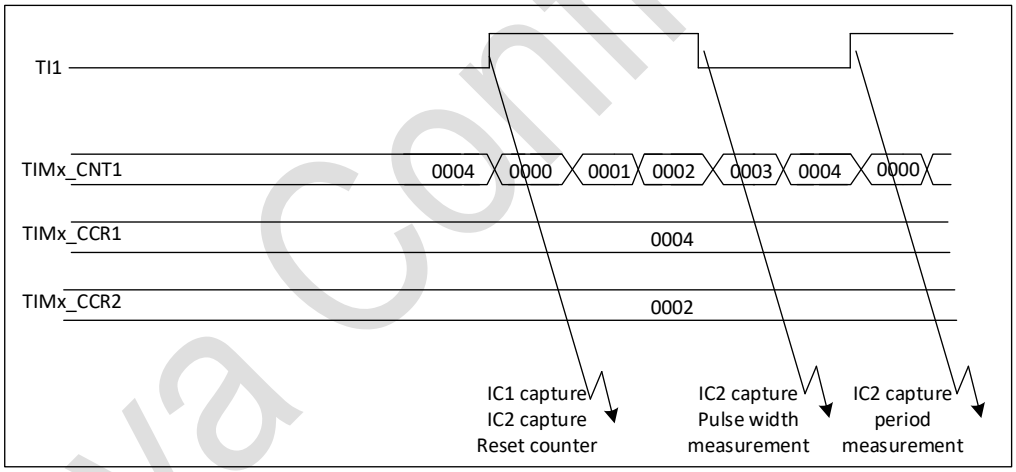


Figure 23-19 PWM input mode timing

23.4.8. Forced output mode

In output mode (CCxS bits = 00 in the TIMx_CCMRx register), each output compares signal (OCxREF and then OCx/OCxN) can be forced to active or inactive level directly by software, independently of any comparison between the output compare register and the counter. To force an output compare signal (OCXREF/OCx) to its active level, you just need to write 101 in the OCxM bits in the corresponding TIMx_CCMRx register. Thus OCxREF is forced high (OCxREF is always active high) and OCx get opposite value to CCxP polarity bit.

For example:CCxP=0 (OCx active high) => OCx is forced to high level. The OCxREF signal can be forced low by writing the OCxM bits to 100 in the TIMx_CCMRx register.

Puya Confidential

Anyway, the comparison between the TIMx_CCRx shadow register and the counter is still performed and allows the flag to be set. Interrupt and DMA requests can be sent accordingly. This is described in the output compare mode section below.

23.4.9. Output compare mode

This function is used to control an output waveform or indicating when a period of time has elapsed. When a match is found between the capture/compare register and the counter, the output compare function:

- Assigns the corresponding output pin to a programmable value defined by the output compare mode (OCxM bits in the TIMx_CCMRx register) and the output polarity (CCxP bit in the TIMx_CCER register). The output pin can keep its level (OCXM=000), be set active (OCxM=001), be set inactive (OCxM=010) or can toggle (OCxM=011) on match.
- Sets a flag in the interrupt status register (CCxIF bit in the TIMx_SR register).
- Generates an interrupt if the corresponding interrupt mask is set (CCXIE bit in the TIMx_DIER register).
- Sends a DMA request if the corresponding enable bit is set (CCxDE bit in the TIMx_DIER register, CCDS bit in the TIMx_CR2 register for the DMA request selection).

The TIMx_CCRx registers can be programmed with or without preload registers using the OCxPE bit in the TIMx_CCMRx register. In output compare mode, the update event UEV has no effect on OCxREF and OCx output.

The timing resolution is one count of the counter. Output compare mode can also be used to output a single pulse (in one pulse mode).

Procedure:

1. Select the counter clock (internal, external, prescaler).
2. Write the desired data in the TIMx_ARR and TIMx_CCRx registers.
3. Set the CCxIE bit if an interrupt request is to be generated.
4. Select the output mode. For example:
   - Write OCxM = 011 to toggle OCx output pin when CNT matches CCRx
   - Write OCxPE = 0 to disable preload register
   - Write CCxP = 0 to select active high polarity
   - Write CCxE = 1 to enable the output
5. Enable the counter by setting the CEN bit in the TIMx_CR1 register.

The TIMx_CCRx register can be updated at any time by software to control the output waveform, provided that the preload register is not enabled (OCxPE= ’0’, else TIMx_CCRx shadow register is updated only at the next update event UEV). An example is given in the figure below.

Puya Confidential

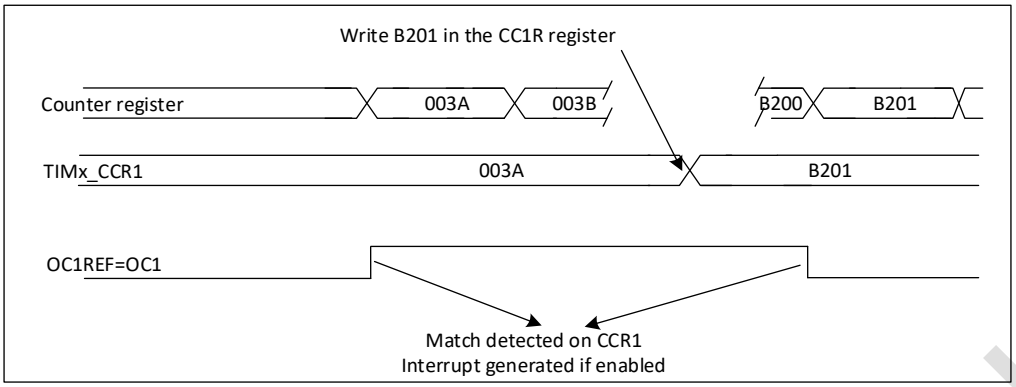


Figure 23-20 Output compare mode, toggle on OC1

23.4.10. PWM mode

Pulse Width Modulation mode allows you to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register.

The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing '110' (PWM mode 1) or '111' (PWM mode 2) in the OCxM bits in the TIMx_CCMRx register. You must enable the corresponding preload register by setting the OCxPE bit in the TIMx_CCMRx register, and eventually the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIMx_CR1 register.

As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, all registers must be initialized by setting the UG bit in the TIMx_EGR register.

OCx polarity is software programmable using the CCxP bit in the TIMx_CCER register. It can be programmed as active high or active low. OCx output is enabled by a combination of the CCxE, CCxNE, MOE, OSSI and OSSR bits (TIMx_CCER and TIMx_BDTR registers). Refer to the TIMx_CCER register description for more details.

In PWM mode (1 or 2), TIMx_CNT and TIMx_CCRx are always compared to determine whether TIMx_CCRx ≤ TIMx_CNT or TIMx_CNT ≤ TIMx_CCRx (depending on the direction of the counter).

The timer is able to generate PWM in edge-aligned mode or center-aligned mode depending on the CMS bits in the TIMx_CR1 register.

PWM edge-aligned mode

- Upcounting configuration

Upcounting is active when the DIR bit in the TIMx_CR1 register is low. In the following example, we consider PWM mode 1. The reference PWM signal OCxREF is high as long as TIMx_CNT < TIMx_CCRx else it becomes low. If the compare value in TIMx_CCRx is greater than the auto-reload value (in TIMx_ARR) then OCxREF is held at '1'. If the compare value is 0 then OCxRef is held at '0'. Figure below shows some edge-aligned PWM waveforms in an example where TIMx_ARR=8.

Puya Confidential

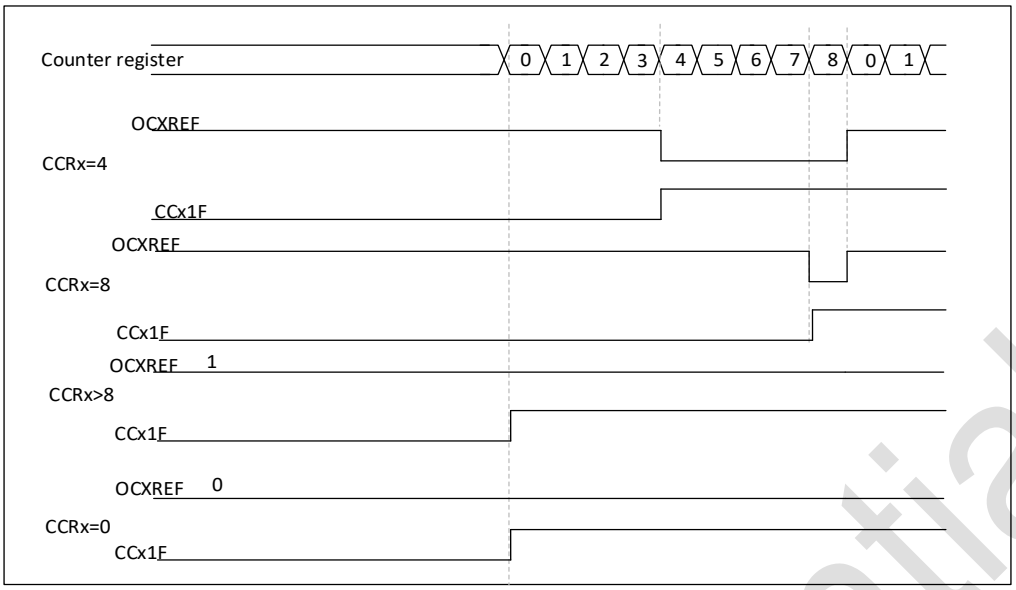


Figure 23-21 Edge-aligned PWM waveforms (ARR=8)

23.4.11. Complementary outputs and dead-time insertion

The general-purpose timers (TIM15/16/17) can output two complementary signals and manage the switching-off and the switching-on instants of the outputs. This time is generally known as dead-time, and you have to adjust it depending on the devices you have connected to the outputs and their characteristics (intrinsic delays of level-shifters, delays due to power switches...)

You can select the polarity of the outputs (main output OCx or complementary OCxN) independently for each output. This is done by writing to the CCxP and CCxNP bits in the TIMx_CCER register.

The complementary signals OCx and OCxN are activated by a combination of several control bits:the CCxE and CCxNE bits in the TIMx_CCER register and the MOE, OISx, OISxN, OSSI and OSSR bits in the TIMx_BDTR and TIMx_CR2 registers. Refer to output control bits for complementary OCx and OCxN channels with break feature for more details. In particular, the dead-time is activated when switching to the IDLE state (MOE falling down to 0).

Dead-time insertion is enabled by setting both CCxE and CCxNE bits, and the MOE bit if the break circuit is present. There is one 8-bit dead-time generator for each channel. From a reference waveform OCxREF, it generates 2 outputs OCx and OCxN. If OCx and OCxN are active high:

- The OCx output signal is the same as the reference signal except for the rising edge, which is delayed relative to the reference rising edge.
- The OCxN output signal is the opposite of the reference signal except for the rising edge, which is delayed relative to the reference falling edge.

If the delay is greater than the width of the active output (OCx or OCxN) then the corresponding pulse is not generated.

The following figures show the relationships between the output signals of the dead-time generator and the reference signal OCxREF. (we suppose CCxP=0, CCxNP=0, MOE=1, CCxE=1 and CCxNE=1 in these examples)

Puya Confidential

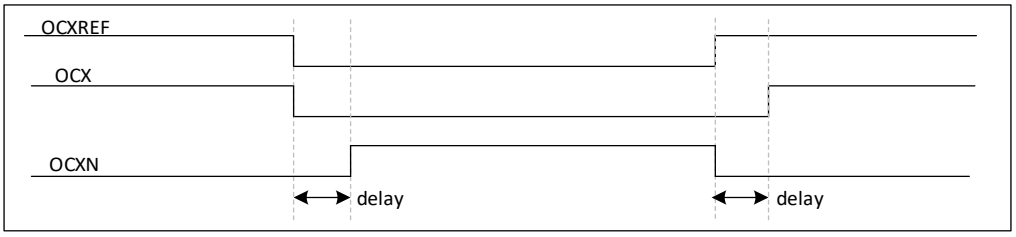


Figure 23-22 Complementary output with dead-time insertion

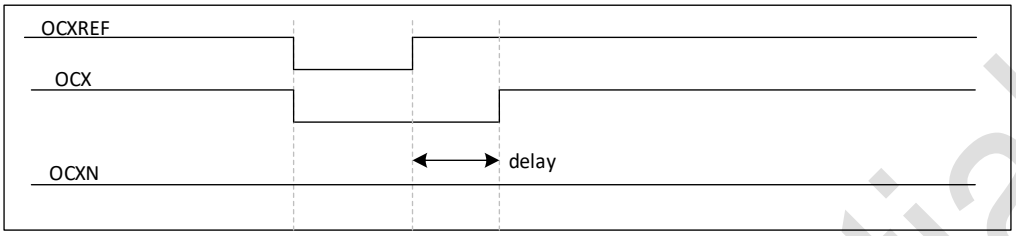


Figure 23-23 Dead-time waveforms with delay greater than the negative pulse

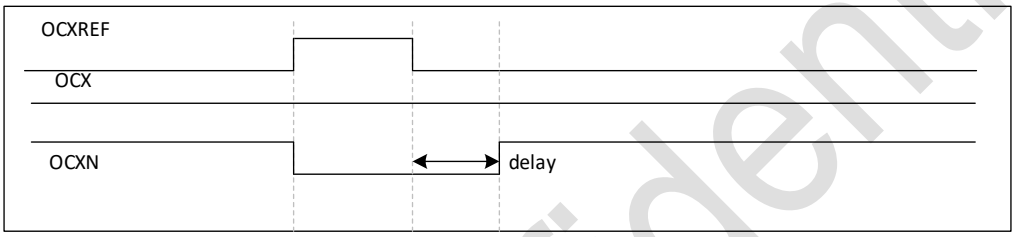


Figure 23-24 Dead-time waveforms with delay greater than the positive pulse.

The dead-time delay is the same for each of the channels and is programmable with the DTG bits in the TIMx_BDTR register.

Re-directing OCxREF to OCx or OCxN

In output mode (forced, output compare or PWM), OCxREF can be re-directed to the OCx output or to OCxN output by configuring the CCxE and CCxNE bits in the TIMx_CCER register. This allows you to send a specific waveform (such as PWM or static active level) on one output while the complementary remains at its inactive level. Other alternative possibilities are to have both outputs at inactive level or both outputs active and complementary with dead-time.

Note:When only OCxN is enabled (CCxE=0, CCxNE=1), it is not complemented and becomes active as soon as OCxREF is high. For example, if CCxNP=0 then OCxN=OCxREF. On the other hand, when both OCx and OCxN are enabled (CCxE=CCxNE=1) OCx becomes active when OCxREF is high whereas OCxN is complemented and becomes active when OCxREF is low.

23.4.12. Using the break function

When the brake function is used, both the output enable signal and the invalid level signal are modified according to additional control bits (MOE, OSSI and OSSR bits in the TIMx_BDTR register, OISx and OISxN bits in the TIMx_CR2 register). In any case, the OCx and OCxN outputs cannot be set both to active level at a given time.

The source for break (BRK) channel can be an external source connected to the BKIN pin or one of the following internal sources:

- the Cortex®-M0+ LOCKUP output
- the PVD output

Puya Confidential

- a clock failure event generated by the CSS detector
- the output from a comparator

When exiting from reset, the break circuit is disabled and the MOE bit is low. You can enable the break function by setting the BKE bit in the TIMx_BDTR register. The break input polarity can be selected by configuring the BKP bit in the same register. BKE and BKP can be modified at the same time. When the BKE and BKP bits are written, a delay of 1 APB clock cycle is applied before the writing is effective. Consequently, it is necessary to wait 1APB clock period to correctly read back the bit after the write operation.

MOE falling edge can be asynchronous, thus a resynchronization circuit has been inserted between the actual signal (acting on the outputs) and the synchronous control bit (accessed in the TIMx_BDTR register). It results in some delays between the asynchronous and the synchronous signals. In particular, if you write MOE to 1 whereas it was low, you must insert a delay (dummy instruction) before reading it correctly. This is because the write acts on the asynchronous signal whereas the read reflects the synchronous signal.

When a break occurs (selected level on the break input):

- The MOE bit is cleared asynchronously, putting the outputs in inactive state, idle state or in reset state (selected by the OSSI bit). This feature functions even if the MCU oscillator is off.
- Each output channel is driven with the level programmed in the OISx bit in the TIMx_CR2 register as soon as MOE=0. If OSSI=0 then the timer releases the enable output else the enable output remains high.
- When complementary outputs are used:
  - The outputs are first put in reset state inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
  - If the timer clock is still present, then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the OISx and OISxN bits after a dead-time. Even in this case, OCx and OCxN cannot be driven to their active level together. Note that because of the resynchronization on MOE, the dead-time duration is a bit longer than usual (around 2 ck_tim clock cycles).
  - If OSSI=0 then the timer releases the enable outputs else the enable outputs remain or become high as soon as one of the CCxE or CCxNE bits is high.
- The break status flag (BIF bit in the TIMx_SR register) is set. An interrupt can be generated if the BIE bit in the TIMx_DIER register is set. If the BDE bit in the TIMx_DIER register is set, a DMA request is generated.
- If the AOE bit in the TIMx_BDTR register is set, the MOE bit is automatically set again at the next update event UEV. This can be used to perform a regulation, for instance. Else, MOE remains low until it is written to ‘1’ again. In this case, it can be used for security and the break input can be connected to an alarm from power drivers, thermal sensors or any security components.

Note:The break inputs are active on level. Thus, the MOE cannot be set while the break input is active (neither automatically nor by software). In the meantime, the status flag BIF cannot be cleared.

Puya Confidential

The break can be generated by the BRK input which has a programmable polarity and an enable bit BKE in the TIMx_BDTR register. The brake can also be generated by software setting the BG bit in the TIMx_EGR register.

In addition to the break input and the output management, a write protection has been implemented inside the break circuit to safeguard the application. It allows to freeze the configuration of several parameters (dead-time duration, OCx/OCxN polarities and state when disabled, OCxM configurations, break enable and polarity). The application can choose from 3 levels of protection selected by the LOCK bits in the TIMx_BDTR register. The LOCK bits can be written only once after an MCU reset.

The figure below shows an example of behavior of the outputs in response to a break.

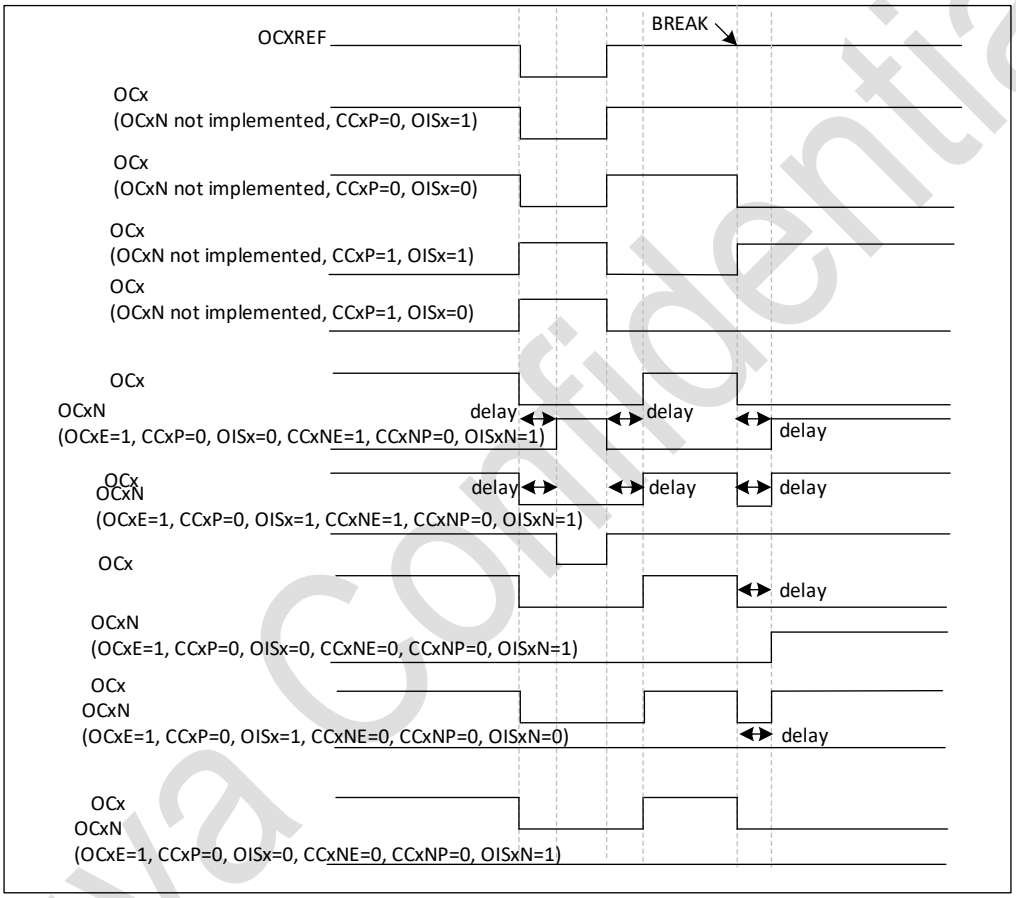


Figure 23-25 Output behavior in response to a break

23.4.13. One-pulse mode

One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. One-pulse mode is selected by setting the OPM bit in the TIMx_CR1 register. This makes the counter stop automatically at the next update event UEV.

A pulse can be correctly generated only if the compare value is different from the counter initial value. Before starting (when the timer is waiting for the trigger), the configuration must be:

- In upcounting:CNT < CCRx ≤ ARR (in particular, 0 < CCRx)

Puya Confidential

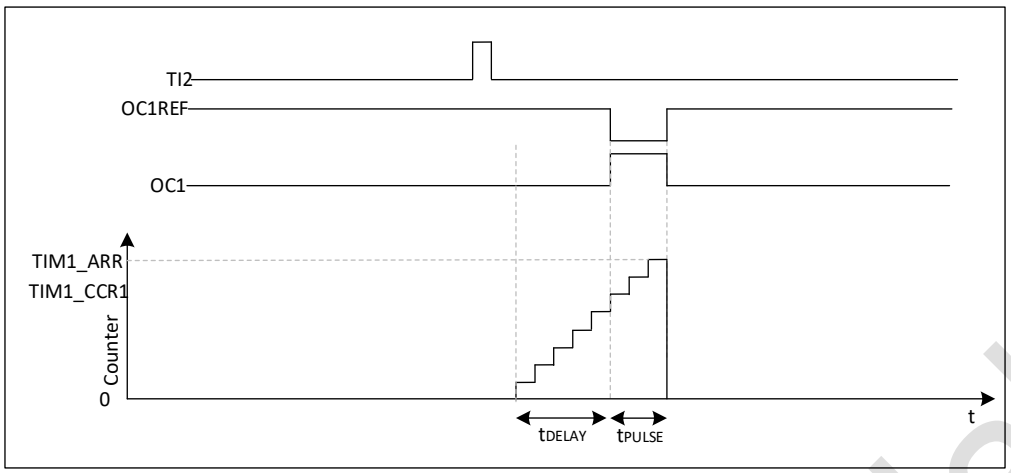


Figure 23-26 Example of one pulse mode

For example, one may want to generate a positive pulse on OC1 with a length of t_{PULSE} and after a delay of t_{DELAY} as soon as a positive edge is detected on the TI2 input pin.

Let's use TI2FP2 as trigger 1:

- Map TI2FP2 on TI2 by writing CC2S=01 in the TIMx_CCMR1 register.
- TI2FP2 must detect a rising edge, write CC2P=0 in the TIMx_CCER register.
- Configure TI2FP2 as trigger for the slave mode controller (TRGI) by writing TS=110 in the TIMx_SMCR register.
- TI2FP2 is used to start the counter by writing SMS to '110' in the TIMx_SMCR register (trigger mode).

The OPM waveform is defined by writing the compare registers (taking into account the clock frequency and the counter prescaler).

- The t_{DELAY} is defined by the value written in the TIMx_CCR1 register.
- The t_{PULSE} is defined by the difference between the auto-reload value and the compare value (TIMx_ARR - TIMx_CCR1).
- Let's say one want to build a waveform with a transition from '0' to '1' when a compare match occurs and a transition from '1' to '0' when the counter reaches the auto-reload value. To do this PWM mode 2 must be enabled by writing OC1M=111 in the TIMx_CCMR1 register. Optionally the preload registers can be enabled by writing OC1PE='1' in the TIMx_CCMR1 register and ARPE in the TIMx_CR1 register. In this case one has to write the compare value in the TIMx_CCR1 register, the auto-reload value in the TIMx_ARR register, generate an update by setting the UG bit and wait for external trigger event on TI2. CC1P is written to '0' in this example.

In our example, the DIR and CMS bits in the TIMx_CR1 register should be low.

The user only wants one pulse (Single mode), so '1' must be written in the OPM bit in theTIMx_CR1 register to stop the counter at the next update event (when the counter rolls over from the auto-reload value back to 0).

Puya Confidential

Particular case:OCx fast enable:

In One-pulse mode, the edge detection on TIx input set the CEN bit which enables the counter. Then the comparison between the counter and the compare value makes the output toggle. But several clock cycles are needed for these operations, and it limits the minimum delay tDELAY min we can get.

If one wants to output a waveform with the minimum delay, the OCxFE bit can be set in the TIMx_CCMRx register. Then OCxREF (and OCx) is forced in response to the stimulus, without taking in account the comparison. Its new level is the same as if a compare match had occurred. OCxFE acts only if the channel is configured in PWM1 or PWM2 mode.

23.5. External trigger synchronization (only for TIM15))

The TIMx timer can be synchronized with an external trigger in several modes:Reset mode, Gated mode and Trigger mode.

23.5.1. Slave mode:Reset mode

The counter and its prescaler can be reinitialized in response to an event on a trigger input. Moreover, if the URS bit from the TIMx_CR1 register is low, an update event UEV is generated. Then all the preloaded registers (TIMx_ARR, TIMx_CCRx) are updated.

In the following example, the upcounter is cleared in response to a rising edge on TI1 input:

- Configure the channel 1 to detect rising edges on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=0 in TIMx_CCER register to validate the polarity (and detect rising edges only).
- Configure the timer in reset mode by writing SMS=100 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.
- Enable the counter by writing CEN=1 in the TIMx_CR1 register.

The counter starts counting on the internal clock, then behaves normally until TI1 rising edge. When TI1 rises, the counter is cleared and restarts from 0. In the meantime, the trigger flag is set (TIF bit in the TIMx_SR register) and an interrupt request, or a DMA request can be sent if enabled (depending on the TIE and TDE bits in TIMx_DIER register).

The following figure shows this behavior when the auto-reload register TIMx_ARR=0x36. The delay between the rising edge on TI1 and the actual reset of the counter is due to the resynchronization circuit on TI1 input.

Puya Confidential

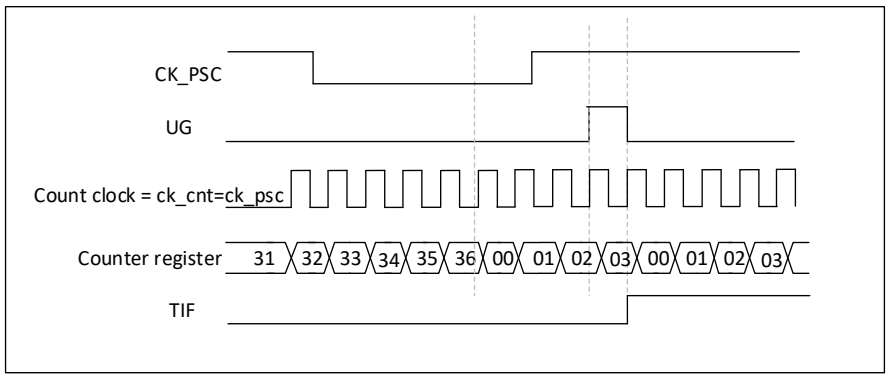


Figure 23-27 Control circuit in reset mode

23.5.2. Slave mode:Gated mode

The counter can be enabled depending on the level of a selected input.

In the following example, the upcounter counts only when TI1 input is low:

- Configure the channel 1 to detect low levels on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=1 in TIMx_CCER register to validate the polarity (and detect low level only).
- Configure the timer in gated mode by writing SMS=101 in TIMx_SMCR register. SelectTI1 as the input source by writing TS=101 in TIMx_SMCR register.
- Enable the counter by writing CEN=1 in the TIMx_CR1 register. In gated mode, the counter doesn't start if CEN=0, whatever is the trigger input level.

The counter starts counting on the internal clock as long as TI1 is low and stops as soon as TI1 becomes high. The TIF flag in the TIMx_SR register is set both when the counter starts or stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

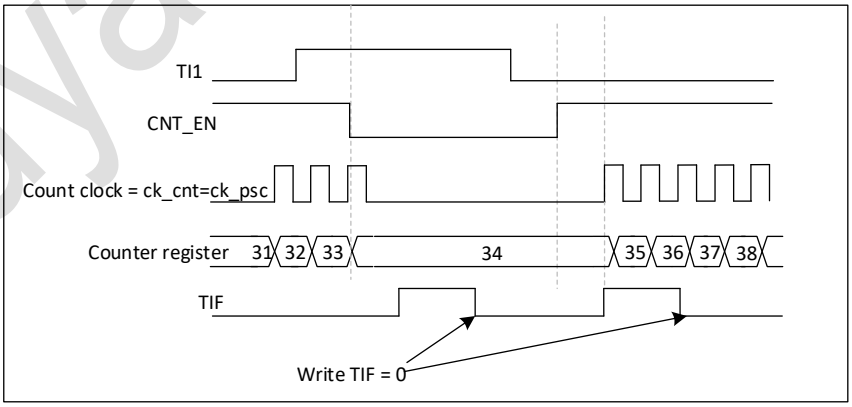


Figure 23-28 Control circuit in Gated mode

23.5.3. Slave mode:Trigger mode

The selected event on the input enables the counter.

In the following example, the upcounter starts in response to a rising edge on TI2 input:

Puya Confidential

- Configure the channel 2 to detect rising edges on TI2. Configure the input filter duration (in this example, we do not need any filter, so we keep IC2F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC2S bits are configured to select the input capture source only, CC2S=01 in TIMx_CCMR1 register. Write CC2P=1 in TIMx_CCER register to validate the polarity (and detect low level only).
- Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI2 as the input source by writing TS=110 in TIMx_SMCR register.

When a rising edge occurs on TI2, the counter starts counting on the internal clock and the TIF flag is set.

The delay between the rising edge on TI2 and the actual start of the counter is due to the resynchronization circuit on TI2 input.

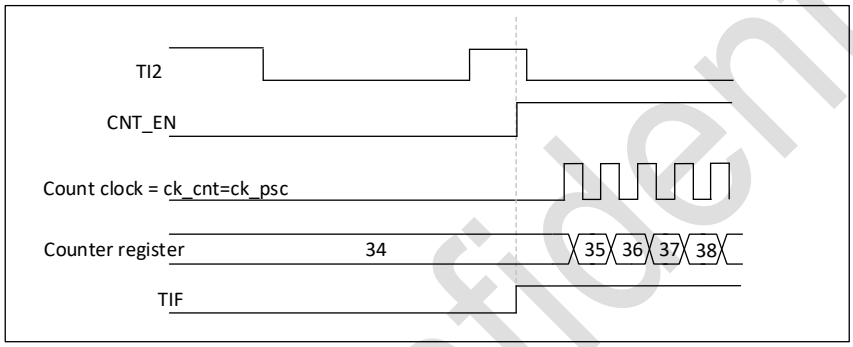


Figure 23-29 Control circuit in trigger mode

23.5.4. Slave mode:external clock mode 2 + trigger mode

The external clock mode 2 can be used in addition to another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as external clock input, and another input can be selected as trigger input when operating in reset mode, gated mode or trigger mode. It is recommended not to select ETR as TRGI through the TS bits of TIMx_SMCR register.

In the following example, the upcounter is incremented at each rising edge of the ETR signal as soon as a rising edge of TI1 occurs:

1. Configure the external trigger input circuit by programming the TIMx_SMCR register as follows:
   - ETF = 0000:no filter
   - ETPS = 00:prescaler disabled
   - ETP = 0:detection of rising edges on ETR and ECE=1 to enable the external clock mode 2.
2. Configure the channel 1 as follows, to detect rising edges on TI:
   - IC1F = 0000:no filter
   - The capture prescaler is not used for triggering, so it does not need to be configured.
   - CC1S=01 in TIMx_CCMR1 register to select only the input capture source
   - Write CC1P=0 in TIMx_CCER register to validate the polarity (and detect rising edges only).
3. Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.

Puya Confidential

A rising edge on TI1 enables the counter and sets the TIF flag. The counter then counts on ETR rising edges.

The delay between the rising edge of the ETR signal and the actual reset of the counter is due to the resynchronization circuit on ETRP input.

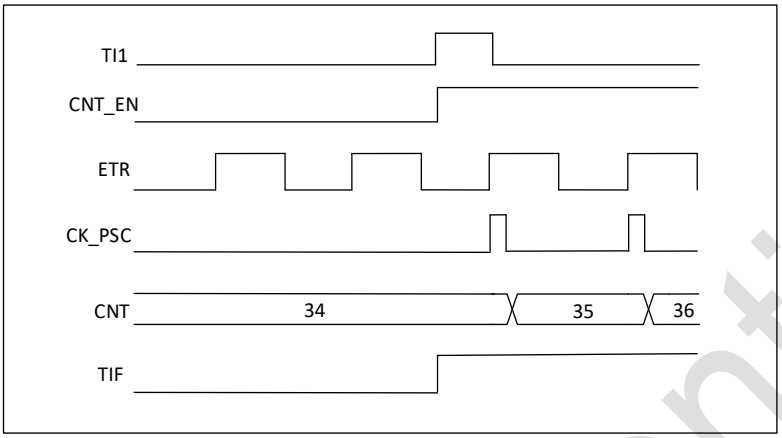


Figure 23-30 Control circuit in external clock mode 2 + trigger mode

23.5.5. TIM and external trigger synchronization

The TIMx timer can be synchronized with an external trigger in several modes:Reset mode, Gated mode and Trigger mode.

Slave mode:Reset mode

The counter and its prescaler can be reinitialized in response to an event on a trigger input. Moreover, if the URS bit from the TIMx_CR1 register is low, an update event UEV is generated. Then all the preloaded registers (TIMx_ARR, TIMx_CCRx) are updated.

In the following example, the upcounter is cleared in response to a rising edge on TI1 input:

- Configure the channel 1 to detect rising edges on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=0 in TIMx_CCER register to validate the polarity (and detect rising edges only).
- Configure the timer in reset mode by writing SMS=100 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.
- Enable the counter by writing CEN=1 in the TIMx_CR1 register.

The counter starts counting on the internal clock, then behaves normally until TI1 rising edge. When TI1 rises, the counter is cleared and restarts from 0. In the meantime, the trigger flag is set (TIF bit in the TIMx_SR register) and an interrupt request, or a DMA request can be sent if enabled (depending on the TIE and TDE bits in TIMx_DIER register).

The following figure shows this behavior when the auto-reload register TIMx_ARR=0x36. The delay between the rising edge on TI1 and the actual reset of the counter is due to the resynchronization circuit on TI1 input.

Puya Confidential

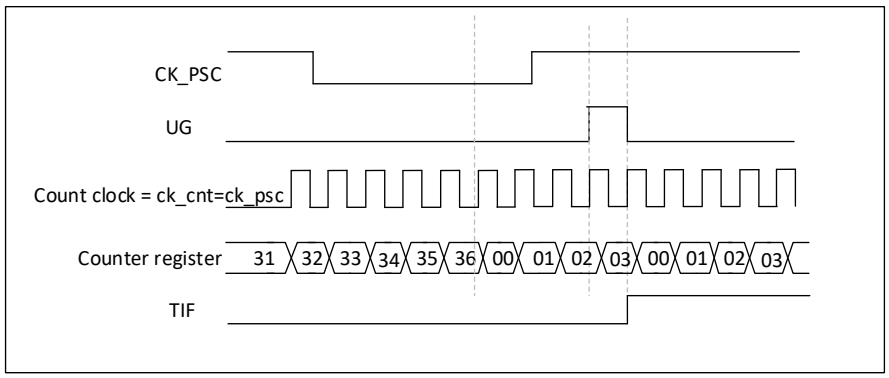


Figure 23-31 Control circuit in reset mode

Slave mode:Gated mode

The counter can be enabled depending on the level of a selected input.

In the following example, the upcounter counts only when TI1 input is low:

- Configure the channel 1 to detect low levels on TI1. Configure the input filter duration (in this example, we do not need any filter, so we keep IC1F=0000). The capture prescaler is not used for triggering, so it does not need to be configured. The CC1S bits select the input capture source only, CC1S=01 in TIMx_CCMR1 register. Write CC1P=1 in TIMx_CCER register to validate the polarity (and detect low level only).
- Configure the timer in gated mode by writing SMS=101 in TIMx_SMCR register. SelectTI1 as the input source by writing TS=101 in TIMx_SMCR register.
- Enable the counter by writing CEN=1 in the TIMx_CR1 register. In gated mode, the counter doesn't start if CEN=0, whatever is the trigger input level.

The counter starts counting on the internal clock as long as TI1 is low and stops as soon as TI1 becomes high. The TIF flag in the TIMx_SR register is set both when the counter starts or stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

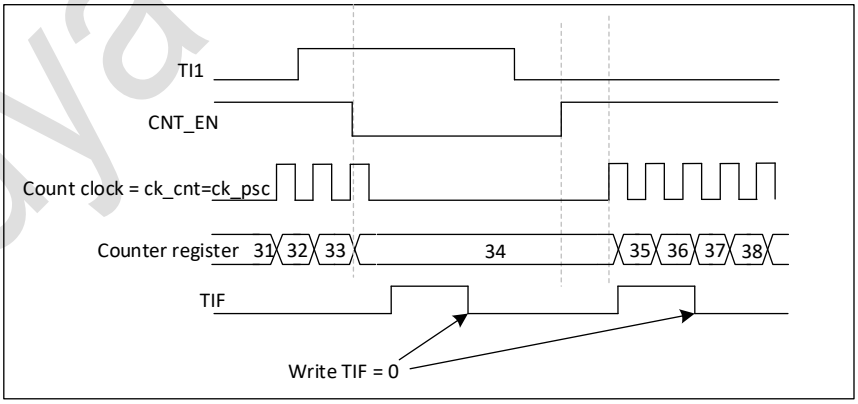


Figure 23-32 Control circuit in Gated mode

The selected event on the input enables the counter.

In the following example, the upcounter starts in response to a rising edge on TI2 input:

a) Configure the channel 2 to detect rising edges on TI2. Configure the input filter duration (in this example, we do not need any filter, so we keep IC2F=0000). The capture prescaler is not

Puya Confidential

used for triggering, so it does not need to be configured. The CC2S bits are configured to select the input capture source only, CC2S=01 in TIMx_CCMR1 register. Write CC2P=1 in TIMx_CCER register to validate the polarity (and detect low level only).

b) Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI2 as the input source by writing TS=110 in TIMx_SMCR register.

When a rising edge occurs on TI2, the counter starts counting on the internal clock and the TIF flag is set. The delay between the rising edge on TI2 and the actual start of the counter is due to the re-synchronization circuit on TI2 input.

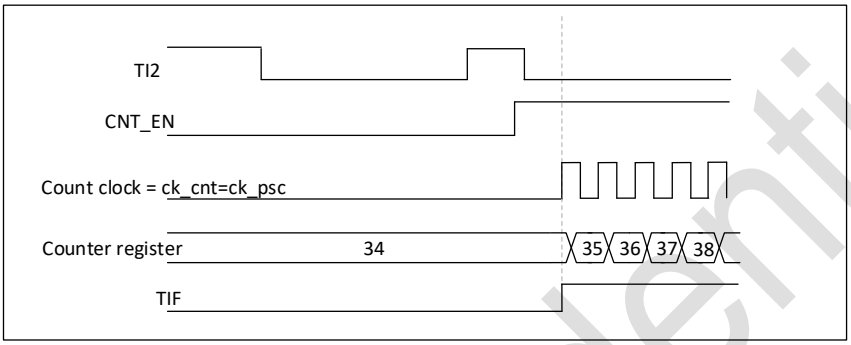


Figure 23-33 Control circuit in Gated mode

Slave mode:external clock mode 2 + trigger mode

The external clock mode 2 can be used in addition to another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as external clock input, and another input can be selected as trigger input when operating in reset mode, gated mode or trigger mode. It is recommended not to select ETR as TRGI through the TS bits of TIMx_SMCR register.

In the following example, the upcounter is incremented at each rising edge of the ETR signal as soon as a rising edge of TI1 occurs:

- Configure the external trigger input circuit by programming the TIMx_SMCR register as follows:
  - ETF = 0000:no filter
  - ETPS = 00:prescaler disabled
  - ETP = 0:detection of rising edges on ETR and ECE=1 to enable the external clock mode 2.
- Configure the channel 1 as follows, to detect rising edges on TI:
  - IC1F = 0000:no filter
  - The capture prescaler is not used for triggering, so it does not need to be configured.
  - CC1S=01 in TIMx_CCMR1 register to select only the input capture source
  - Write CC1P=0 in TIMx_CCER register to validate the polarity (and detect rising edges only).
- Configure the timer in trigger mode by writing SMS=110 in TIMx_SMCR register. Select TI1 as the input source by writing TS=101 in TIMx_SMCR register.

A rising edge on TI1 enables the counter and sets the TIF flag. The counter then counts on ETR rising edges. The delay between the rising edge of the ETR signal and the actual reset of the counter is due to the resynchronization circuit on ETRP input.

Puya Confidential

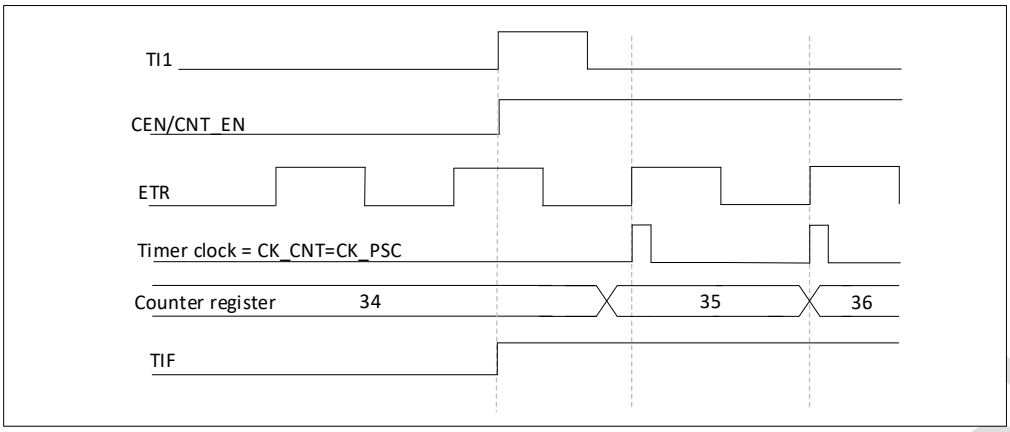


Figure 23-34 Control circuit in external clock mode 2 + trigger mode

23.6. Timer synchronization (only for TIM15))

The TIMx timers are linked together internally for timer synchronization or chaining. When one Timer is configured in Master mode, it can reset, start, stop or clock the counter of another Timer configured in Slave mode.

The following presents an overview of the trigger selection and the master mode selection blocks.

23.6.1. Using one timer as prescaler for another timer

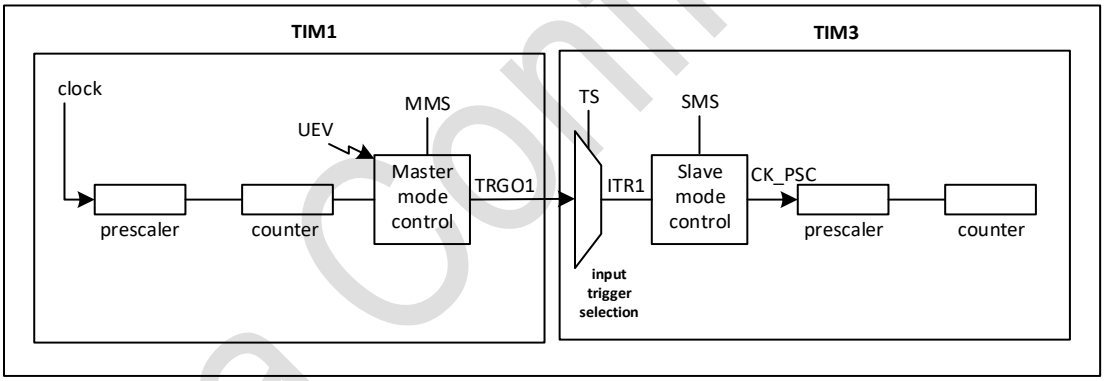


Figure 23-35 Master/Slave timer example

For example, the user can configure Timer 1 to act as a prescaler for Timer 2. The following is performed:

- Configure Timer 1 in master mode so that it outputs a periodic trigger signal on each update event UEV. If you write MMS=010 in the TIM/15/16/17_TIM1_CR2 register, a rising edge is output on TRGO each time an update event is generated.
- To connect the TRGO output of Timer 1 to Timer 2, Timer 2 must be configured in slave mode using ITR0 as internal trigger. You select this through the TS bits in the TIM2_SMCR register (writing TS=000).
- Then you put the slave mode controller in external clock mode 1 (write SMS=111 in the TIM2_SMCR register). This causes Timer 2 to be clocked by the rising edge of the periodic Timer 1 trigger signal (which correspond to the timer 1 counter overflow).
- Finally, the CEN bit of the corresponding (TIMx_CR1 register) must be set to start the two timers respectively.

Puya Confidential

Note:If OCx is selected on Timer 1 as trigger output (MMS=1xx), its rising edge is used to clock the counter of timer 2.

23.6.2. Using one timer to enable another timer

In this example, we control the enable of Timer 2 with the output compare 1 of Timer 1. Timer 2 counts on the divided internal clock only when OC1REF of Timer 1 is high. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_INT ($f_{CK\_CNT} = f_{CK\_INT}/3$).

- Configure Timer 1 master mode to send its Output compare 1 reference (OC1REF) signal as trigger output (MMS=100 in the TIM15/16/17_CR2 register).
- Configure the Timer 1 OC1REF waveform (TIM15/16/17_CCMR1 register).
- Configure Timer 2 to get the input trigger from Timer 1 (TS=000 in the TIM2_SMCR register).
- Configure Timer 2 in gated mode (SMS=101 in TIM2_SMCR register).
- Enable Timer 2 by writing ‘1 in the CEN bit (TIM2_CR1 register).
- Start Timer 1 by writing ‘1 in the CEN bit (TIM15/16/17_CR1 register).

Note:The counter 2 clock is not synchronized with counter 1, this mode only affects the Timer 2 counter enable signal.

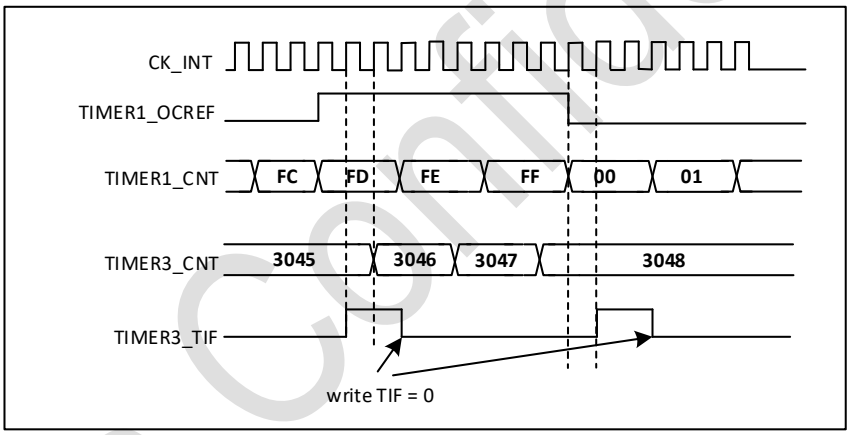


Figure 23-36 Gating timer 2 with OC1REF of timer 1

The Timer 2 counter and prescaler are not initialized before being started. So they start counting from their current value. It is possible to start from a given value by resetting both timers before starting Timer 1. You can then write any value you want in the timer counters. The timers can easily be reset by software using the UG bit in the TIMx_EGR registers.

In the next example, we synchronize Timer 1 and Timer 2. Timer 1 is the master and starts from 0. Timer 2 is the slave and starts from 0xE7. The prescaler ratio is the same for both timers. Timer 2 stops when Timer 1 is disabled by writing 0 to the CEN bit in the TIM15/16/17_CR1 register.

- Configure Timer 1 master mode to send its Output compare 1 reference (OC1REF) signal as trigger output (MMS=100 in the TIM15/16/17_CR2 register).
- Configure the Timer 1 OC1REF waveform (TIM15/16/17_CCMR1 register).
- Configure Timer 2 to get the input trigger from Timer 1 (TS=000 in the TIM2_SMCR register).
- Configure Timer 2 in gated mode (SMS=101 in TIM2_SMCR register).
- Reset Timer 1 by writing 1 in UG bit (TIM15/16/17_EGR register).

Puya Confidential

- Reset Timer 1 by writing 1 in UG bit (TIM2_EGR register).
- Initialize Timer 2 to 0XE7 by writing '0xE7' in the timer 2 counter (TIM2_CNT).
- Enable Timer 2 by writing 1 in the CEN bit (TIM2_CR1 register).
- Start Timer 1 by writing 1 in the CEN bit (TIM15/16/17_CR1 register).
- Stop Timer 1 by writing 0 in the CEN bit (TIM15/16/17_CR1 register).

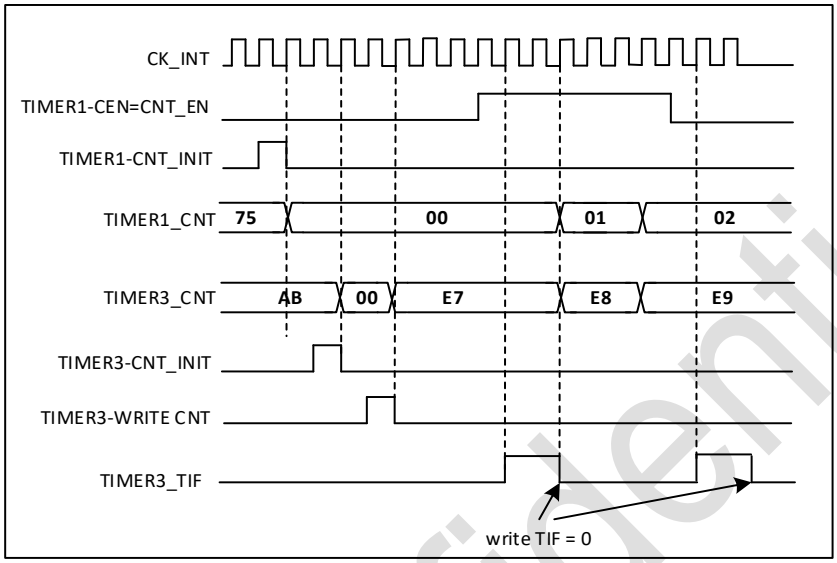


Figure 23-37 Gating timer 2 with enable of timer 1

23.6.3. Using one timer to start another timer

In this example, we set the enable of Timer 2 with the update event of Timer 1. Timer 2 starts counting from its current value (which can be nonzero) on the divided internal clock as soon as the update event is generated by Timer 1. When Timer 2 receives the trigger signal its CEN bit is automatically set and the counter counts until we write 0 to the CEN bit in the TIM2_CR1 register. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_INT ($f_{CK\_CNT} = f_{CK\_INT}/3$).

- Configure Timer 1 master mode to send its update event (UEV) as trigger output (MMS=010 in the TIM15/16/17_CR2 register).
- Configure the Timer 1 period (TIM15/16/17_ARR registers)
- Configure Timer 2 to get the input trigger from Timer 1 (TS=000 in the TIM2_SMCR register).
- Configure Timer 2 in trigger mode (SMS=110 in TIM2_SMCR register).
- Start Timer 1 by writing '1 in the CEN bit (TIM15/16/17_CR1 register).

Puya Confidential

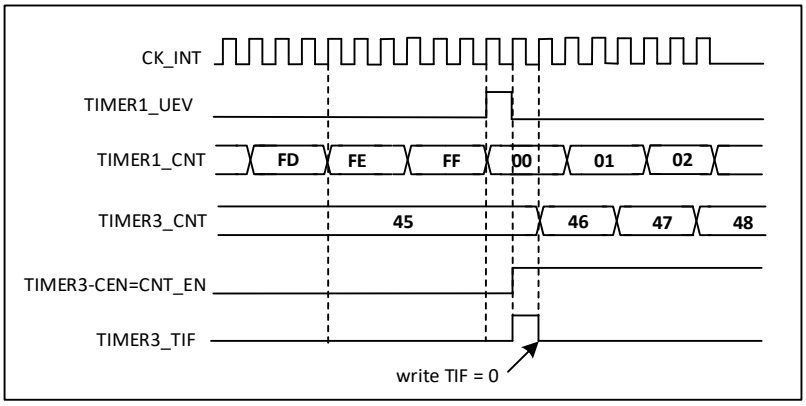


Figure 23-38 Triggering timer 2 with update of timer 1

As in the previous example, the user can initialize both counters before starting counting. The following figure shows the behavior with the same configuration as in Figure above but in trigger mode instead of gated mode (SMS=110 in the TIM3_SMCR register).

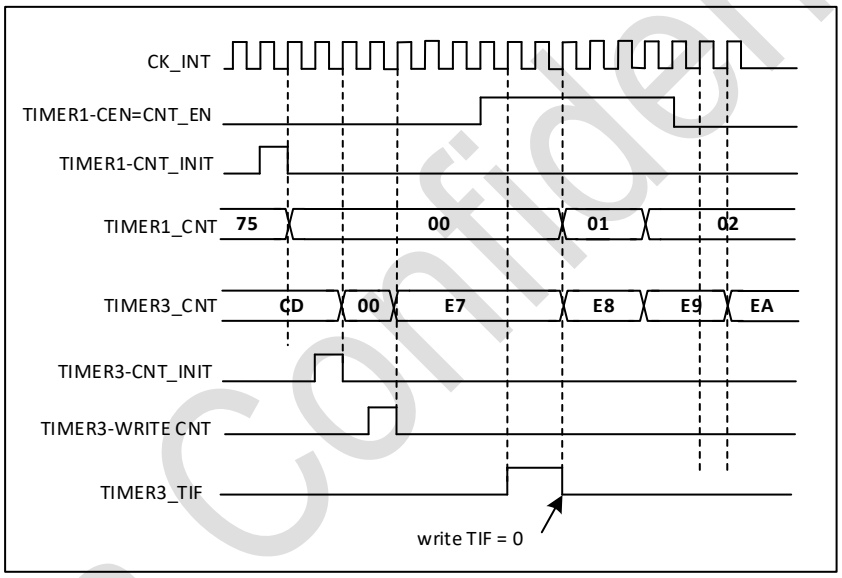


Figure 23-39 Triggering timer 2 with enable of timer 1

23.6.4. Starting 2 timers synchronously in response to an external trigger

In this example, we set the enable of timer 1 when its TI1 input rises, and the enable of Timer 3 with the enable of Timer 1. To ensure the counters are aligned, Timer 1 must be configured in Master/Slave mode (slave with respect to TI1, master with respect to Timer 2):

- Configure Timer 1 master mode to send its enable as trigger output (MMS=001 in the TIM15/16/17_CR2 register).
- Configure Timer 1 slave mode to get the input trigger from TI1 (TS=100 in the TIM15/16/17_SMCR register).
- Configure Timer 1 in trigger mode (SMS=110 in the TIM15/16/17_SMCR register).
- Configure the Timer 1 in Master/Slave mode by writing MSM=1 (TIM15/16/17_SMCR register).
- Configure Timer 2 to get the input trigger from Timer 1 (TS=000 in the TIM2_SMCR register).
- Configure Timer 2 in trigger mode (SMS=110 in TIM2_SMCR register).

Puya Confidential

When a rising edge occurs on TI1 (Timer 1), both counters starts counting synchronously on the internal clock and both TIF flags are set.

Note:In this example both timers are initialized before starting (by setting their respective UG bits). Both counters start from 0, but you can easily insert an offset between them by writing any of the counter registers (TIMx_CNT). You can see that the master/slave mode insert a delay between CNT_EN and CK_PSC on timer 1.

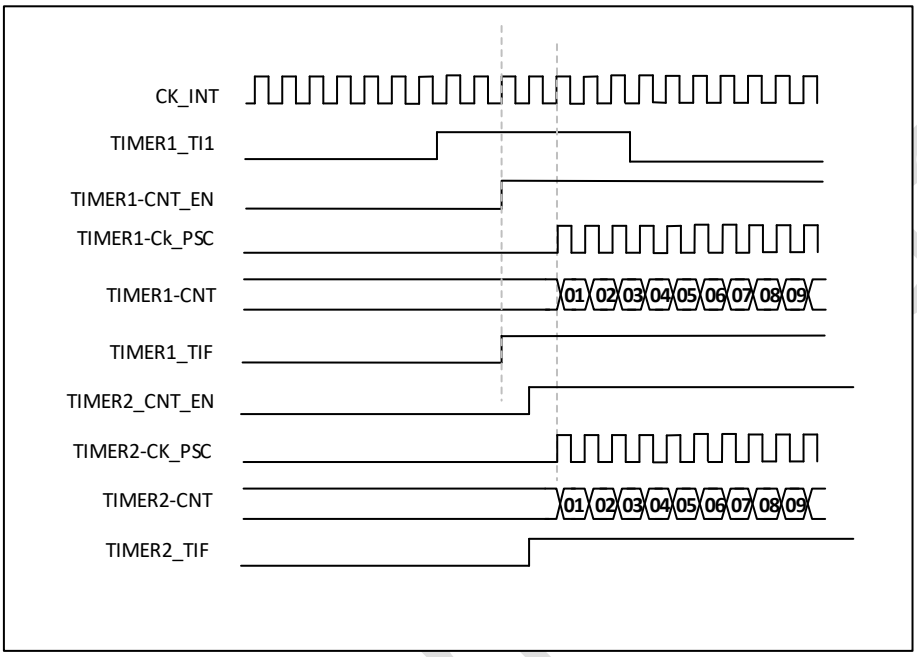


Figure 23-40 Triggering Timer 1 and Timer 2 using the TI1 input of Timer 1

23.6.5. Debug mode

When the microcontroller enters debug mode, the TIMx counter either continues to work normally or stops, depending on DBG_TIMx_STOP configuration bit in DBG module.

23.7. TIM15 registers

0x40014800-0x40014BFFTIM17

0x40014400-0x400147FFTIM16

0x40014000-0x400143FFTIM15

23.7.1. TIM15 control register 1 (TIMx_CR1)

Address offset:0x00

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	CKD [1:0]		ARPE	Res	Res	Res	OPM	URS	UDIS	CEN
-	-	-	-	-	-	RW		RW	-	-	-	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:10	Reserved	-	-	Reserved

Bit	Name	R/W	Reset Value	Function
9:8	CKD [1:0]	RW	00	Clock division This bit-field indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock (t_{DTS}) used by the dead-time generators and the digital filters (ETR, TIx) 00:t_{DTS} = t_{CK_INT} 01:t_{DTS} = 2 x t_{CK_INT} 10:t_{DTS} = 4 x t_{CK_INT} 11:Reserved, do not program this value
7	ARPE	RW	0	Auto-reload preload enable 0:TIMx_ARR register is not buffered 1:TIMx_ARR register is buffered
6:4	Reserved	-	-	Reserved
3	OPM	RW	0	One-pulse mode 0:Counter is not stopped at update event 1:Counter stops counting at the next update event (clearing the bit CEN)
2	URS	RW	0	Update request source This bit is set and cleared by software to select the UEV event sources. 0:Any of the following events generate an update interrupt or DMA request if enabled. These events can be: − Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller 1:Only counter overflow/underflow generates an update interrupt or DMA request if enabled
1	UDIS	RW	0	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0:UEV enabled. The Update (UEV) event is generated by one of the following events: − Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1:UEV disabled. The update event is not generated, shadow registers keep their value (ARR, PSC, CCRx).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				However, the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	RW	0	Counter enable 0:Counter disabled 1:Counter enabled Note:external clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However, trigger mode can set the CEN bit automatically by hardware.

23.7.2. TIM15 control register 2 (TIMx_CR2)

Address offset:0x04

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res					OIS2	OIS1N	OIS1	Res	MMS [2:0]			CCDS	CCUS	Res	CCPC
-					RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW

Bit	Name	R/W	Reset Value	Function
15:11	Reserved	-	-	Reserved
10	OIS2	RW	0	Output idle state 2 (OC2 output) Refer to OIS1 bit
9	OIS1N	RW	0	Output idle state 1 (OC1N output) 0:OC1N=0 after a dead-time when MOE=0 1:OC1N=1 after a dead-time when MOE=0 Note:this bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).
8	OIS1	RW	0	Output idle state 1 (OC1 output) 0:OC1=0 (after a dead-time if OC1N is implemented) when MOE=0 1:OC1=1 (after a dead-time if OC1N is implemented) when MOE=0 Note:this bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).
7	Reserved	-	-	Reserved
6:4	MMS [2:0]	RW	000	Master mode selection

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				These three bits are used to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000:Reset - the UG bit from the TIMx_EGR register is used as trigger output (TRGO). If the reset is generated by the trigger input (slave mode controller configured in reset mode) then the signal on TRGO is delayed compared to the actual reset. 001:Enable - the counter enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enabled. The Counter Enable signal is generated by a logic AND between CEN control bit and the trigger input when configured in gated mode. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected (see the MSM bit description in TIMx_SMCR register). 010:Update - The update event is selected as trigger output (TRGO). For instance, a master timer can then be used as a prescaler for a slave timer. 011:Compare pulse - The trigger output sends a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or a compare match occurred (TRGO). 100:Compare - OC1REF signal is used as trigger output (TRGO) 101:Compare - OC2REF signal is used as trigger output (TRGO) Note: 1. The clock of the slave timer or ADC must be enabled prior to receive events from the master timer, and must not be changed. 2. If the master and slave timers are not on the same bus, the master mode should be configured to the width that can be picked by the slave timer.
3	CCDS	RW	0	Capture/compare DMA selection 0:CCx DMA request sent when CCx event occurs

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:CCx DMA requests sent when update event occurs
2	CCUS	RW	0	Capture/compare control update selection 0:When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COM bit only. 1:When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COM bit or when a rising edge occurs on TRGI. Note:This bit acts only on channels that have a complementary output.
1	Reserved	-	-	Reserved
0	CCPC	RW	0	Capture/compare preloaded control 0:CCxE, CCxNE and OCxM bits are not preloaded. 1:CCxE, CCxNE, and OCxM bits are preloaded; Once this bit is set, they are only set when COM Note:This bit acts only on channels that have a complementary output.

23.7.3. TIM15 slave mode control register (TIMx_SMCR)

Address offset:0x08

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								MSM	TS [2:0]			Res	SMS [2:0]		
-								RW	RW			-	RW		

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7	MSM	RW	0	Master/slave mode 0:No action 1:The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.
6:4	TS [2:0]	RW	000	Trigger selection. This bit-field selects the trigger input to be used to synchronize the counter. 000:TIM2 (ITR0) 001:TIM3 (ITR1) 010:TIM16 (ITR2) 011:TIM17 (ITR3) 100:edge detector of TI1 (TI1F_ED)

Puya Confidential

				101:Filtered timer input 1 (TI1FP1) 110:Filtered timer input 2 (TI2FP2) 111 external trigger input (ETRF) Note:These bits must be changed only when they are not used to avoid wrong edge detections at the transition.
3	Reserved	-	-	Reserved
2:0	SMS [2:0]	RW	000	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input (see Input Control register and Control Register description). 000:Slave mode disabled If CEN = '1' then the prescaler is clocked directly by the internal clock. 100:Reset mode Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101:Gated mode The counter clock is enabled when the trigger input (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop of the counter are controlled. 110:Trigger mode The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only the start of the counter is controlled. 111:External clock mode 1 Rising edges of the selected trigger (TRGI) clock the counter. Note:The gated mode must not be used if TI1F_EN is selected as the trigger input (TS=100). Indeed, TI1F_ED outputs 1 pulse for each transition on TI1F, whereas the gated mode checks the level of the trigger signal. Note:In encoder mode, do not use uev as trgo output signal, (i.e. mms cannot be configured to 010)

TIM15 internal trigger connection

Slave TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM15	TIM2_TRGO	TIM3_TRGO	TIM16_OC1	TIM17_OC

23.7.4. TIM15 DMA/interrupt enable register (TIMx_DIER)

Address offset:0x0C

Reset value:0x000 00000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	TDE	COMDE	Res		CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res		CC2IE	CC1IE	UIE

Puya Confidential

-	RW	RW	-	-	RW	-	-	RW

Bit	Name	R/W	Reset Value	Function
15	Reserved	-	-	Reserved
14	TDE	RW	0	TDE:Trigger DMA request enable 0:Trigger DMA request disabled 1:Trigger DMA request enabled
13	COMDE	RW	0	COMDE:COM DMA request enable 0:COM DMA request disabled 1:COM DMA request enabled
11:12	Reserved	-	-	Reserved
10	CC2DE	RW	0	CC2DE:Capture/Compare 2 DMA request enable 0:CC2 DMA request disabled 1:CC2 DMA request enabled
9	CC1DE	RW	0	CC1DE:Capture/Compare 1 DMA request enable 0:CC1 DMA request disabled 1:CC1 DMA request enabled
8	UDE	RW	0	UDE:Update DMA request enable 0:Update DMA request disabled 1:Update DMA request enabled
7	BIE	RW	0	BIE:Break interrupt enable 0:Break interrupt disabled 1:Break interrupt enabled
6	TIE	RW	0	TIE:Trigger interrupt enable 0:Trigger interrupt disabled. 1:Trigger interrupt enabled
5	COMIE	RW	0	COMIE:COM interrupt enable 0:COM interrupt disabled 1:COM interrupt enabled
4:3	Reserved	-	-	Reserved
2	CC2IE	RW	0	CC2IE:Capture/Compare 2 interrupt enable 0:Capture/Compare 2 interrupt disabled 1:Capture/Compare 2 interrupt enabled
1	CC1IE	RW	0	CC1IE:Capture/Compare 1 interrupt enable 0:Capture/Compare 1 interrupt disabled 1:Capture/Compare 1 interrupt enabled
0	UIE	RW	0	UIE:Update interrupt enable 0:Update interrupt disabled. 1:Update interrupt enabled

Puya Confidential

23.7.5. TIM15 status register (TIMx_SR)

Address offset:0x010

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	CC2OF	CC1OF	Res	BIF	TIF	COMIF	Res	Res	CC2IF	CC1F	UIF
-	-	-	-	-	RC_W0		-	RC_W0			-	-	RC_W0		

Bit	Name	R/W	Reset Value	Function
15:11	Reserved	-	-	Reserved
10	CC2OF	RC_W0	0	Capture/Compare 2 overcapture flag Refer to CC1OF description
9	CC1OF	RC_W0	0	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'. 0:No overcapture has been detected. 1:The counter value has been captured in TIMx_CCR1 register while CC1OF flag was already set.
8	Reserved	-	-	Reserved
7	BIF	RC_W0	0	Break interrupt flag This flag is set by hardware as soon as the break input goes active. It can be cleared by software if the break input is not active. 0:No break event occurred. 1:An active level has been detected on the break input.
6	TIF	RC_W0	0	Trigger interrupt flag This flag is set by hardware on trigger event (active edge detected on TRGI input when the slave mode controller is enabled in all modes but gated mode). It is set when the counter starts or stops when gated mode is selected. It is cleared by software. 0:No trigger event occurred. 1:Trigger interrupt pending.
5	COMIF	RC_W0	0	COM interrupt flag This flag is set by hardware on COM event (when Capture/compare Control bits - CCxE, CCxNE, OCxM - have been updated). It is cleared by software. 0:No update occurred. 1:COM interrupt pending.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
4:3	Reserved	-	-	Reserved
2	CC2IF	RC_W0	0	Capture/Compare 2 interrupt flag Refer to CC1IF description
1	CC1IF	RC_W0	0	Capture/Compare 1 interrupt flag If channel CC1 is configured as output: This bit is set to 1 by hardware when the counter value matches the comparison value, except in centrosymmetric mode (refer to TIMx_CR1 register It is cleared by software. 0:No match; 1:The content of the counter TIMx_CNT matches the content of the TIMx_CCR1 register. If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading the TIMx_CCR1 register. 0:No input capture occurred 1:The counter value has been captured in TIMx_CCR1 register (An edge has been detected on IC1 which matches the selected polarity) Note:When CEN is on, this bit will also be set.
0	UIF	RC_W0	0	Update interrupt flag This bit is set by hardware on an update event. It is cleared by software. 0:No update occurred. 1:Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow regarding the repetition counter value (update if REP_CNT = 0) and if the UDIS=0 in the TIMx_CR1 register. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0 in the TIMx_CR1 register. − If UDIS = 0 and URS = 0 of the TIMx_CR1 register, an update event will be generated when the CNT is reinitialized by the trigger event. (Refer to TIM1 slave mode control register (TIM1_SMCR))

23.7.6. TIM15 event generation register (TIMx_EGR)

Address offset:0x14

Reset value:0x00000000

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	BG	TG	COMG	Res	Res	CC2G	CC1G	UG
-	-	-	-	-	-	-	-	W	W	W	-	-	W	W	W

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7	BG	W	0	Break generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A break event is generated. MOE bit is cleared and BIF flag is set. Related interrupt or DMA transfer can occur if enabled.
6	TG	W	0	Trigger generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:The TIF flag is set in TIMx_SR register. Related interrupt or DMA transfer can occur if enabled.
5	COMG	W	0	Capture/Compare control update generation This bit can be set by software, it is automatically cleared by hardware. 0:No action 1:When CCPC bit is set, it allows to update CCxE, CCxNE and OCxM bits Note:This bit acts only on channels that have a comple-mentary output.
4:3	Reserved	-	-	Reserved
2	CC2G	W	0	Capture/Compare 2 generation Refer to CC1G description
1	CC1G	W	0	Capture/Compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A capture/compare event is generated on channel CC1: If channel CC1 is configured as input: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input:

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				The current value of the counter is captured in TIMx_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if theCC1IF flag was already set.
0	UG	W	0	Update generation This bit can be set by software; it is automatically cleared by hardware. 0:No action 1:Reinitialize the counter and generate an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected).

23.7.7. TIM15 capture/compare mode register 1 (TIMx_CCMR1)

Address offset:0x18

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	OC2M [2:0]			OC2PE	CO2FE	CC2S [1:0]		Res	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
IC2F [3:0]				IC2PSC [1:0]				IC1F [3:0]				IC1PSC [1:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Output compare mode

Bit	Name	R/W	Reset Value	Function
15	Reserved	-	-	Reserved
14:12	OC2M [2:0]	RW	0	Output Compare 2 mode
11	OC2PE	RW	0	Output Compare 2 preload enable
10	OC2FE	RW	0	Output Compare 2 fast enable
9:8	CC2S [1:0]	RW	00	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC2 channel is configured as output; 01:CC2 channel is configured as input, IC2 is mapped on TI2; 10:CC2 channel is configured as input, IC2 is mapped on TI1; 11:CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC2S bits are writable only when the channel is OFF (CC2E = ‘0’ in TIMx_CCER).
7	Reserved	-	-	Reserved
6:4	OC1M [2:0]	RW	00	Output compare 1 mode These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000:Frozen. The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on the outputs. 001:Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 010:Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 011:Toggle OC1REF toggles when TIMx_CNT=TIMx_CCR1. 100:Force inactive level OC1REF is forced low. 101:Force active level OC1REF is forced high. 110:PWM mode 1 - Channel 1 is active as long as TIMx_CNT<TIMx_CCR1 else inactive; 111:PWM mode 2 - Channel 1 is inactive as long as TIMx_CNT<TIMx_CCR1 else active. Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output). Note:In PWM mode 1 or 2, the OCREF level changes when the result of the comparison changes or when the output compare mode switches from frozen to PWM mode.
3	OC1PE	RW	0	Output Compare 1 preload enable 0:Preload register on TIMx_CCR1 disabled. TIMx_CCR1 can be written at any time, the new value is taken in account immediately. 1:Preload register on TIMx_CCR1 enabled. Read/Write operations access the preload register. TIMx_CCR1 preload value is loaded in the active register at each update event. Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output). Note:The PWM mode can be used without validating the preload register only in one pulse mode. Else the behavior is not guaranteed.
2	OC1FE	RW	0	Output Compare 1 fast enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0:CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1:An active edge on the trigger input acts like a compare match on OC1 output. Then, OC1 is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OCFE acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S [1:0]	RW	00	Capture/Compare 1 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:CC1 channel is configured as input, IC1 is mapped on TI2; 11:CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC1S bits are writable only when the channel is OFF (CC1E = '0' in TIMx_CCER).

Input capture mode:

Bit	Name	R/W	Reset Value	Function
15:12	IC2F	RW	0	Input capture 2 filter
11:10	IC2PSC [1:0]	RW	0	Capture/Compare 2 prescaler
9:8	CC2S [1:0]	RW	0	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC2 channel is configured as output; 01:CC2 channel is configured as input, IC2 is mapped on TI2; 10:CC2 channel is configured as input, IC2 is mapped on TI1; 11:CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC2S bits are writable only when the channel is OFF (CC2E = '0' in TIMx_CCER).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
7:4	IC1F [3:0]	RW	0000	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output: 0000:No filter, sampling is done at f_{DTS} 0001:$f_{SAMPLING} = f_{CK\_INT}$, N = 2 0010:$f_{SAMPLING} = f_{CK\_INT}$, N = 4 0011:$f_{SAMPLING} = f_{CK\_INT}$, N = 8 0100:$f_{SAMPLING} = f_{DTS}/2$, N = 6 0101:$f_{SAMPLING} = f_{DTS}/2$, N = 8 0110:$f_{SAMPLING} = f_{DTS}/4$, N = 6 0111:$f_{SAMPLING} = f_{DTS}/4$, N = 8 1000:$f_{SAMPLING} = f_{DTS}/8$, N = 6 1001:$f_{SAMPLING} = f_{DTS}/8$, N = 8 1010:$f_{SAMPLING} = f_{DTS}/16$, N = 5 1011:$f_{SAMPLING} = f_{DTS}/16$, N = 6 1100:$f_{SAMPLING} = f_{DTS}/16$, N = 8 1101:$f_{SAMPLING} = f_{DTS}/32$, N = 5 1110:$f_{SAMPLING} = f_{DTS}/32$, N = 6 1111:$f_{SAMPLING} = f_{DTS}/32$, N = 8
3:2	IC1PSC [1:0]	RW	00	Input capture 1 prescaler 00:no prescaler, capture is done each time an edge is detected on the capture input; 01:capture is done once every 2 events 10:capture is done once every 4 events 11:capture is done once every 8 events
1:0	CC1S [1:0]	RW	00	CC1S [1:0]:capture/compare 1 selection. This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:CC1 channel is configured as input, IC1 is mapped on TI2; 11:CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC1S bits are writable only when the channel is OFF (CC1E = '0' in TIMx_CCER).

Puya Confidential

23.7.8. TIM15 capture/compare enable register (TIMx_CCER)

Address offset:0x20

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								CC2NP	Res	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
-								RW	-	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7	CC2NP	RW	0	Capture/compare 2 complementary output polarity. Refer to CC1NP description.
6	Reserved	-	-	Reserved
5	CC2P	RW	0	Capture/compare 2 output polarity Refer to CC1P description.
4	CC2E	RW	0	Capture/Compare 2 output enable Refer to CC1E description.
3	CC1NP	RW	0	Capture/compare 1 complementary output polarity. 0:OC1N active high. 1:OC1N active low. Note:Once the LOCK level (LOCK bit in the TIMx_BDTR register) is set to 3 or 2 and CC1S = 00 (channel configuration as output), this bit cannot be modified.
2	CC1NE	RW	0	Capture/Compare 1 complementary output enable 0:OC1N output disabled 1:OC1N signal output to corresponding output pin When the CC1 channel is configured as an output, the OC1N output level is determined by the MOE, OSSI, OSSR, OIS1, OIS1N, CC1E, and CC1NE bits, as shown in the table below
1	CC1P	RW	0	Capture/compare 1 output polarity CC1 channel configured as output: 0:OC1 active high. 1:OC1 active low. CC1 channel configured as input: CC1NP/CC1P bits select the active polarity of TI1FP1and TI2FP1 for trigger or capture operations. 00:Non-inverted/rising edge: The circuit is sensitive to TIxFP1 rising edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode or encoder mode). 01:Inverted/falling edge:

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				The circuit is sensitive to TIxFP1 falling edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is inverted (trigger operation in gated mode or encoder mode). 10:reserved, do not use this configuration. 11:non-inverted/double edge The circuit is sensitive to both TIxFP1 rising and falling edges (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode). This configuration must not be used in encoder mode. Notes: 1. For complementary output channels, this bit is preloaded. If the CCPC bit is set in the TIMx_CR2 register, then the CC1P active bit takes the new value from the preloaded bit only when a Commutation event is generated. 2. Once the LOCK level (LOCK bit in the TIMx_BDTR register) is set to 3 or 2, this bit cannot be modified
0	CC1E	RW	0	Capture/Compare 1 output enable 0:Off - OC1 is not active. 1:On - OC1 signal is output on the corresponding output pin depending on MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits. Notes: For complementary output channels, this bit is preloaded. If the CCPC bit is set in the TIMx_CR2 register then the CC1E active bit takes the new value from the preloaded bit only when a Commutation event is generated.

Table 23-1 Output control bits for complementary OCx and OCxN channels with break feature

Control bits					Output state	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx output state	OCxN output state
1	X	0	0	0	Output disabled (not driven by the timer), OCx=0, OCx_EN=0	Output disabled (not driven by the timer), OCxN=0, OCxN_EN=0
		0	0	1	Output disabled (not driven by the timer), OCx=0, OCx_EN=0	OCxREF + Polarity OCxN = OCxREF xor CCxNP, OCxN_EN = 1

Puya Confidential

Control bits					Output state	
MOE	**OSSI**	**OSSR**	**CCxE**	**CCxNE**	**OCx output state**	**OCxN output state**
		0	1	0	OCxREF+Polarity OCx = OCREF xor CCxP, OCx_EN=1	Output disabled (not driven by the timer), OCxN=0, OCxN_EN=0
		0	1	1	OCREF + Polarity + dead-time OCx_EN = 1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
		1	0	0	Output disabled (not driven by the timer), OCx=CCxP, OCx_EN=0	Output disabled (not driven by the timer), OCxN=CCxNP, OCxN_EN=0
		1	0	1	Output disabled (not driven by the timer), OCx=CCxP, OCx_EN=1	OCxREF+Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF+Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state), OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF + Polarity + dead-time OCx_EN = 1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
0	0	X	0	0	Output disabled (not driven by the timer anymore).	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1	Off-State (output enabled with inactive state) Asynchronously:OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1 If the clock is present:OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state.	
	1		1	0		
	1		1	1		

When both outputs of a channel are not used (CCxE = CCxNE = 0), the OISx, OISxN, CCxP and CCxNP bits must be kept cleared.

Note:The state of the external I/O pins connected to the complementary OCx and OCxN channels depends on the OCx and OCxN channel state and AFIO registers.

Puya Confidential

23.7.9. TIM15 counter (TIMx_CNT)

Address offset:0x24

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	CNT [15:0]	RW	0	Counter value

23.7.10. TIM15 prescaler (TIMx_PSC)

Address offset:0x28

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	PSC [15:0]	RW	0	Prescaler value The counter clock frequency (CK_CNT) is equal to fCK_PSC / (PSC [15:0] +1). PSC contains the value to be loaded in the active pre-scaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in reset mode).

23.7.11. TIM15 auto-reload register (TIMx_ARR)

Address offset:0x2c

Reset value:0x0000FFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	ARR [15:0]	RW	FFFF	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. The counter is blocked while the auto-reload value is null.

Puya Confidential

23.7.12. TIM15 repetition counter register (TIMx_RCR)

Address offset:0x30

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	REP [7:0]							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7:0	REP [7:0]	RW	0	Repetition counter value These bits allow the user to set-up the update rate of the compare registers (i.e. periodic transfers from preload to active registers) when preload registers are enabled, as well as the update interrupt generation rate, if this interrupt is enabled. Each time the REP_CNT related downcounter reaches zero, an update event is generated, and it restarts counting from REP value. As REP_CNT is reloaded with REP value only at the repetition update event U_RC, any write to the TIMx_RCR register is not taken in account until the next repetition update event. It means in PWM mode (REP+1) corresponds to: - the number of PWM periods in edge-aligned mode

23.7.13. TIM15 capture/compare register 1 (TIMx_CCR1)

Address offset:0x34

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
RW/R															

Bit	Name	R/W	Reset Value	Function
15:0	CCR1 [15:0]	RW/R	0	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs.

Puya Confidential

				The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).

23.7.14. TIM15 capture/compare register 2 (TIM1_CCR2)

Address offset:0x38

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
RW/R															

Bit	Name	R/W	Reset Value	Function
15:0	CCR2 [15:0]	RW/R	0	Capture/Compare 2 value If channel CC2 is configured as output: CCR2 is the value to be loaded in the actual capture/compare 2 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC2PE). Else the preload value is copied in the active capture/compare 2 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC output. If channel CC2 is configured as input: CCR2 is the counter value transferred by the last input capture 2 event (IC2).

23.7.15. TIM15 break and dead-time register (TIMx_BDTR)

Address offset:0x44

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DTG [7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Note:As the AOE, BKP, BKE, OSSI, OSSR and DTG [7:0] bits maybe write-locked depending on the LOCK configuration, it may be necessary to configure all of them during the first write access to the TIMx_BDTR register.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
15	MOE	RW	0	Main output enable This bit is cleared asynchronously by hardware as soon as one of the break inputs is active. It is cleared by software or automatically setting depending on the AOE bit. It is acting only on the channels which are configured in output. 0:OC and OCN outputs are disabled or forced to idle state. 1:If the corresponding enable bit (CCxE, CCxNE bits of the TIMx_CCER register) is set, turn on OC and OCN output.
14	AOE	RW	0	Automatic output enable 0:MOE can be set only by software; 1:MOE can be set by software or automatically at the next update event (if none of the break inputs is active). Note:This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIM1_BDTR register).
13	BKP	RW	0	Break polarity 0:Break input BRK is active low; 1:Break input BRK is active high Note:This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIM1_BDTR register). Any write operation to this bit takes a delay of 1 APB clock cycle to become effective.
12	BKE	RW	0	Break enable 0:Break function disabled 1:Break function enabled Note:This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIM1_BDTR register). Any write operation to this bit takes a delay of 1 APB clock cycle to become effective.
11	OSSR	RW	0	Off-state selection for Run mode This bit is used when MOE=1 on channels having a complementary output which are configured as outputs. OSSR bit is not implemented if no complementary output is implemented in the timer. See OC/OCN enable description for more details (TIMx capture/compare enable register (TIMx_CCER)).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal = 0). 1:When inactive, OC/OCN outputs are enabled with their inactive level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1 Note:This bit cannot be modified as long as LOCK level 2 has been set (LOCK bits in TIMx_BDTR register).
10	OSSI	RW	0	Off-state selection for Idle mode This bit is used when MOE=0 and on channels configured as outputs. See OC/OCN enable description for more details (TIMx capture/compare enable register (TIMx_CCER)). 0:When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal = 0). 1:When inactive, OC/OCN outputs are first forced to their idle level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1 Note:This bit cannot be modified as long as LOCK level 2 has been set (LOCK bits in TIMx_BDTR register).
9:8	LOCK [1:0]	RW	00	Lock configuration These bits offer a write protection against software errors. 00:LOCK OFF, no bit is write protected. 01:LOCK Level 1, DTG, BKE, BKP, AOE bits in TIMx_BDTR register and OISx and OISxN bits in TIMx_CR2 register can no longer be written. 10:LOCK Level 2 = LOCK Level 1 + CC Polarity bits (CCxP/CCxNP bits in TIMx_CCER register, as long as the related channel is configured in output through the CCxS bits) as well as OSSR and OSSI bits can no longer be written. 11:LOCK Level 3 = LOCK Level 2 + CC Control bits (OCxM and OCxPE bits in TIMx_CCMRx registers, as long as the related channel is configured in output through the CCxS bits) can no longer be written. Note:The LOCK bits can be written only once after the re-set. Once the TIMx_BDTR register has been written, their content is frozen until the next reset.
7:0	DTG [7:0]	RW	00000000	Dead-time generator setup

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit-field defines the duration of the dead-time inserted between the complementary outputs. DT correspond to this duration. DTG [7:5] = 0xx = > DT = DTG [7:0] × Tdtg, Tdtg = TDTS; DTG [7:5] = 10x = > DT = (64 + DTG [5:0]) × Tdtg, Tdtg = 2 × TDTS; DTG [7:5] = 110 = > DT = (32 + DTG [4:0]) × Tdtg, Tdtg = 8 × TDTS; DTG [7:5] = 111 = > DT = (32 + DTG [4:0]) × Tdtg, Tdtg = 16 × TDTS; Example:If TDTS = 125ns (8MHz), the dead time possible value is: 0 to 15875 ns by 125 ns steps; 16 µs to 31750 ns by 250 ns steps; 32 µs to 63 µs by 1 us steps; 64 µs to 126 µs by 2 us steps Note:This bit-field cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).

23.7.16. TIM15 DMA control register (TIMx_DCR)

Address offset:0x48

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	DBL [4:0]					Res			DBA [4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:13	Reserved	-	-	Reserved
12:8	DBL [4:0]	RW	00000	DMA burst length These bits define the transfer length of the DMA in continuous mode (when reading or writing to the address of the TIMx_DMAR register address is performed). 00000:1 transfer 00001:2 transfers 00010:3 transfers .... 10001:18 transfers

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Example:We consider such a transmission:DBL = 7, DBA = TIM2_CR1 If DBL = 7, DBA = TIM2_CR1 represents the address of the data to be transmitted, then the transmitted address is given by: (address of TIMx_CR1) + DBA + (DMA index), where DMA index = DBL Where (address of TIMx_CR1) + DBA plus 7 gives the address where data will be written or read out, so that data transfer will occur in 7 registers starting from address (address of TIMx_CR1) + DBA. Depending on the setting of the DMA data length, the following can occur: -If the data is set to half word (16 bits), then the data will be transferred to all 7 registers. -If the data is set to bytes, the data will still be transferred to all 7 registers:the first register contains the first MSB byte, the second register contains the first LSB byte, and so on. Therefore, for the timer, the user must specify the data width to be transmitted by the DMA.
7:5	Reserved	-	-	Reserved
4:0	DBA [4:0]	RW	00000	DBA [4:0]:DMA base address These bits define the base address of the DMA in continuous mode (when read/write access are done through the TIMx_DMAR address). DBA is defined as an offset starting from the address of the TIMx_CR1 register. 00000:TIMx_CR1 00001:TIMx_CR2 00010:TIMx_SMCR .....

23.7.17. TIM15 DMA address for full transfer (TIMx_DMAR)

Address offset:0x4C

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB [15:0]															

Bit	Name	R/W	Reset Value	Function
15:0	DMAB [31:0]	RW	0	DMA register for burst accesses

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				A read or write operation to the DMAR register accesses the register located at the address: (TIMx_CR1 address) + (DBA + DMA index) x 4 where: TIMx_CR1 address is the address of the control register 1; DBA is the DMA base address configured in TIMx_DCR register; DMA index is automatically controlled by the DMA transfer, and ranges from 0 to DBL (DBL configured in TIMx_DCR).

Note:When using the DMA continuous transfer function, the value of the CNDTR register of the corresponding channel in the DMA must be corresponded to the value of the DBL in the TIMx_DCR register, otherwise it will not be used properly.

23.8. TIM16/17 registers

0x40014800-0x40014BFFTIM17

0x40014400-0x400147FFTIM16

0x40014000-0x400143FFTIM15

23.8.1. TIM16/17 control register 1 (TIMx_CR1)

Address offset:0x00

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	CKD [1:0]		ARPE	Res	Res	Res	OPM	URS	UDIS	CEN
-	-	-	-	-	-	RW		RW	-	-	-	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:10	Reserved	-	-	Reserved
9:8	CKD [1:0]	RW	00	Clock division This bit-field indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock (tDTS) used by the dead-time generators and the digital filters (ETR, TIx) 00:t_{DTS} = t_{CK_INT} 01:t_{DTS} = 2 x t_{CK_INT} 10:t_{DTS} = 4 x t_{CK_INT} 11:Reserved, do not program this value
7	ARPE	RW	0	Auto-reload preload enable 0:TIMx_ARR register is not buffered 1:TIMx_ARR register is buffered
6:4	Reserved	-	-	Reserved
3	OPM	RW	0	One-pulse mode 0:Counter is not stopped at update event

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:Counter stops counting at the next update event (clearing the bit CEN)
2	URS	RW	0	Update request source This bit is set and cleared by software to select the UEV event sources. 0:Any of the following events generate an update interrupt or DMA request if enabled. These events can be: − Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller 1:Only counter overflow/underflow generates an update interrupt or DMA request if enabled
1	UDIS	RW	0	Update disable This bit is set and cleared by software to enable/disable UEV event generation. 0:UEV enabled. The Update (UEV) event is generated by one of the following events: − Counter overflow/underflow − Setting the UG bit − Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1:UEV disabled. The update event is not generated, shadow registers keep their value (ARR, PSC, CCRx). However, the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	RW	0	Counter enable 0:Counter disabled 1:Counter enabled Note:external clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However, trigger mode can set the CEN bit automatically by hardware.

23.8.2. TIM16/17 control register 2 (TIMx_CR2)

Address offset:0x04

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						OIS1N	OIS1	Res				CCDS	CCUS	Res	CCPC

Puya Confidential

-	RW	RW	-	RW	RW	-	RW

Bit	Name	R/W	Reset Value	Function
15:10	Reserved	-	-	Reserved
9	OIS1N	RW	0	Output idle state 1 (OC1N output) 0:OC1N=0 after a dead-time when MOE=0 1:OC1N=1 after a dead-time when MOE=0 Note:this bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).
8	OIS1	RW	0	Output idle state 1 (OC1 output) 0:OC1=0 (after a dead-time if OC1N is implemented) when MOE=0 1:OC1=1 (after a dead-time if OC1N is implemented) when MOE=0 Note:this bit cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).
7:4	Reserved	-	-	Reserved
3	CCDS	RW	0	Capture/compare DMA selection 0:CCx DMA request sent when CCx event occurs 1:CCx DMA requests sent when update event occurs
2	CCUS	RW	0	Capture/compare control update selection 0:When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COM bit only. 1:When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COM bit or when a rising edge occurs on TRGI. Note:This bit acts only on channels that have a comple-mentary output.
1	Reserved	-	-	Reserved
0	CCPC	RW	0	Capture/compare preloaded control 0:CCxE, CCxNE and OCxM bits are not preloaded. 1:CCxE, CCxNE, and OCxM bits are preloaded; Once this bit is set, they are only set when COM Note:This bit acts only on channels that have a comple-mentary output.

23.8.3. TIM16/17 DMA/interrupt enable register (TIMx_DIER)

Address offset:0x0C

Puya Confidential

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						CC1DE	UDE	BIE	Res	COMIE	Res	Res	Res	CC1IE	UIE
-						RW	RW	RW	-	RW	-	-	-	RW	RW

Bit	Name	R/W	Reset Value	Function
15:10	Reserved	-	-	Reserved
9	CC1DE	RW	0	CC1DE:Capture/Compare 1 DMA request enable 0:CC1 DMA request disabled 1:CC1 DMA request enabled
8	UDE	RW	0	UDE:Update DMA request enable 0:Update DMA request disabled 1:Update DMA request enabled
7	BIE	RW	0	BIE:Break interrupt enable 0:Break interrupt disabled 1:Break interrupt enabled
6	Reserved	-	-	Reserved
5	COMIE	RW	0	COMIE:COM interrupt enable 0:COM interrupt disabled 1:COM interrupt enabled
4:2	Reserved	-	-	Reserved
1	CC1IE	RW	0	CC1IE:Capture/Compare 1 interrupt enable 0:Capture/Compare 1 interrupt disabled 1:Capture/Compare 1 interrupt enabled
0	UIE	RW	0	UIE:Update interrupt enable 0:Update interrupt disabled. 1:Update interrupt enabled

23.8.4. TIM16/17 status register (TIMx_SR)

Address offset:0x010

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						CC1OF	Res	BIF	Res	COMIF	Res			CC1F	UIF
-						RC_W0	-	RC_W0	-	RC_W0	-			RC_W0	RC_W0

Bit	Name	R/W	Reset Value	Function
15:10	Reserved	-	-	Reserved
9	CC1OF	RC_W0	0	Capture/Compare 1 overcapture flag

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'. 0:No overcapture has been detected. 1:The counter value has been captured in TIMx_CCR1 register while CC1OF flag was already set.
8	Reserved	-	-	Reserved
7	BIF	RC_W0	0	Break interrupt flag This flag is set by hardware as soon as the break input goes active. It can be cleared by software if the break input is not active. 0:No break event occurred. 1:An active level has been detected on the break input.
6	Reserved	-	-	Reserved
5	COMIF	RC_W0	0	COM interrupt flag This flag is set by hardware on COM event (when Capture/compare Control bits - CCxE, CCxNE, OCxM - have been updated). It is cleared by software. 0:No update occurred. 1:COM interrupt pending.
4:2	Reserved	-	-	Reserved
1	CC1IF	RC_W0	0	Capture/Compare 1 interrupt flag If channel CC1 is configured as output: This bit is set to 1 by hardware when the counter value matches the comparison value, except in centrosymmetric mode (refer to CMS bit in TIMx_CR1 register) It is cleared by software. 0:No match; 1:The content of the counter TIMx_CNT matches the content of the TIMx_CCR1 register. If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading the TIMx_CCR1 register. 0:No input capture occurred 1:The counter value has been captured in TIMx_CCR1 register (An edge has been detected on IC1 which matches the selected polarity)
0	UIF	RC_W0	0	Update interrupt flag

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is set by hardware on an update event. It is cleared by software. 0:No update occurred. 1:Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow regarding the repetition counter value (update if REP_CNT = 0) and if the UDIS=0 in the TIMx_CR1 register. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0 in the TIMx_CR1 register. − If UDIS = 0 and URS = 0 of the TIMx_CR1 register, an update event will be generated when the CNT is reinitialized by the trigger event. (Refer to TIM1 slave mode control register (TIM1_SMCR))

23.8.5. TIM16/17 event generation register (TIMx_EGR)

Address offset:0x14

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	BG	Res	COMG	Res	Res	Res	CC1G	UG
-	-	-	-	-	-	-	-	W	-	W	-	-	-	W	W

Bit	Name	R/W	Reset Value	Function
15:8	Res	-	-	Reserved
7	BG	W	0	Break generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A break event is generated. MOE bit is cleared and BIF flag is set. Related interrupt or DMA transfer can occur if enabled.
6	Res	-	-	Reserved
5	COMG	W	0	Capture/Compare control update generation This bit can be set by software, it is automatically cleared by hardware. 0:No action

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:When CCPC bit is set, it allows to update CCxE, CCxNE and OCxM bits Note:This bit acts only on channels that have a complementary output.
4:2	Res	-	-	Reserved
1	CC1G	W	0	Capture/Compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0:No action 1:A capture/compare event is generated on channel CC1: If channel CC1 is configured as input: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input: The current value of the counter is captured in TIMx_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if theCC1IF flag was already set.
0	UG	W	0	Update generation This bit can be set by software; it is automatically cleared by hardware. 0:No action 1:Reinitialize the counter and generate an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected). The counter is cleared if the center-aligned mode is selected or if DIR=0 (upcounting), else it takes the auto-reload value (TIMx_ARR) if DIR=1 (downcounting).

23.8.6. TIM16/17 capture/compare mode register 1 (TIMx_CCMR1)

Address offset:0x18

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								Res	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
								IC1F [3:0]				IC1PSC [1:0]			
-								RW	RW	RW	RW	RW	RW	RW	RW

Output compare mode:

Bit	Name	R/W	Reset Value	Function
15:7	Reserved	-	-	Reserved
6:4	OC1M [2:0]	RW	00	Output compare 1 mode

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits. 000:Frozen. The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on the outputs. 001:Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 010:Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 011:Toggle OC1REF toggles when TIMx_CNT=TIMx_CCR1. 100:Force inactive level OC1REF is forced low. 101:Force active level OC1REF is forced high. 110:PWM mode 1 - Channel 1 is active as long as TIMx_CNT<TIMx_CCR1 else inactive. 111:PWM mode 2 - Channel 1 is inactive as long as TIMx_CNT<TIMx_CCR1 else active. Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output). Note:In PWM mode 1 or 2, the OCREF level changes when the result of the comparison changes or when the output compare mode switches from frozen to PWM mode.
3	OC1PE	RW	0	Output Compare 1 preload enable 0:Preload register on TIMx_CCR1 disabled. TIMx_CCR1 can be written at any time, the new value is taken in account immediately. 1:Preload register on TIMx_CCR1 enabled. Read/Write operations access the preload register. TIMx_CCR1 preload value is loaded in the active register at each update event. Note:These bits cannot be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output).

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Note:The PWM mode can be used without validating the preload register only in one pulse mode. Else the behavior is not guaranteed.
2	OC1FE	RW	0	Output Compare 1 fast enable This bit is used to accelerate the effect of an event on the trigger in input on the CC output. 0:CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. The minimum delay to activate CC1 output when an edge occurs on the trigger input is 5 clock cycles. 1:An active edge on the trigger input acts like a compare match on OC1 output. Then, OC1 is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate CC1 output is reduced to 3 clock cycles. OCFE acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	CC1S [1:0]	RW	00	Capture/Compare 1 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:CC1 channel is configured as input, IC1 is mapped on TI2; 11:CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register) Note:CC1S bits are writable only when the channel is OFF (CC1E = ‘0’ in TIMx_CCER).

Input capture mode:

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7:4	IC1F [3:0]	RW	0000	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter applied to TI1. The digital filter is made of an event counter

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				in which N consecutive events are needed to validate a transition on the output: 0000:No filter, sampling is done at f_{DTS} 0001:$f_{SAMPLING}$ = f_{CK_INT}, N = 2 0010:$f_{SAMPLING}$ = f_{CK_INT}, N = 4 0011:$f_{SAMPLING}$ = f_{CK_INT}, N = 8 0100:$f_{SAMPLING}$ = f_{DTS}/2, N = 6 0101:$f_{SAMPLING}$ = f_{DTS}/2, N = 8 0110:$f_{SAMPLING}$ = f_{DTS}/4, N = 6 0111:$f_{SAMPLING}$ = f_{DTS}/4, N = 8 1000:$f_{SAMPLING}$ = f_{DTS}/8, N = 6 1001:$f_{SAMPLING}$ = f_{DTS}/8, N = 8 1010:$f_{SAMPLING}$ = f_{DTS}/16, N = 5 1011:$f_{SAMPLING}$ = f_{DTS}/16, N = 6 1100:$f_{SAMPLING}$ = f_{DTS}/16, N = 8 1101:$f_{SAMPLING}$ = f_{DTS}/32, N = 5 1110:$f_{SAMPLING}$ = f_{DTS}/32, N = 6 1111:$f_{SAMPLING}$ = f_{DTS}/32, N = 8
3:2	IC1PSC [1:0]	RW	00	Input capture 1 prescaler 00:no prescaler, capture is done each time an edge is detected on the capture input; 01:capture is done once every 2 events 10:capture is done once every 4 events 11:capture is done once every 8 events
1:0	CC1S [1:0]	RW	00	CC1S [1:0]:capture/compare 1 selection. This bit-field defines the direction of the channel (input/output) as well as the used input. 00:CC1 channel is configured as output; 01:CC1 channel is configured as input, IC1 is mapped on TI1. 10:CC1 channel is configured as input, IC1 is mapped on TI2; 11:CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through the TS bit (TIMx_SMCR register)

23.8.7. TIM16/17 capture/compare enable register (TIMx_CCER)

Address offset:0x20

Reset value:0x00000000

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC1NP	CC1NE	CC1P	CC1E
-	-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:4	Reserved	-	-	Reserved
3	CC1NP	RW	0	Capture/compare 1 complementary output polarity. 0:OC1N active high. 1:OC1N active low. Note:This bit is not writable as soon as LOCK level 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (channel configured as output).
2	CC1NE	RW	0	Capture/Compare 1 complementary output enable 0:OC1N output disabled 1:OC1N signal output to corresponding output pin When the CC1 channel is configured as an output, the OC1N output level is determined by the MOE, OSSI, OSSR, OIS1, OIS1N, CC1E, and CC1NE bits, as shown in the table below
1	CC1P	RW	0	Capture/compare 1 output polarity CC1 channel configured as output: 0:OC1 active high. 1:OC1 active low. CC1 channel configured as input: CC1NP/CC1P bits select the active polarity of TI1FP1and TI2FP1 for trigger or capture operations. 00:Non-inverted/rising edge: The circuit is sensitive to TIxFP1 rising edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode or encoder mode). 01:Inverted/falling edge: The circuit is sensitive to TIxFP1 falling edge (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is inverted (trigger operation in gated mode or encoder mode). 10:reserved, do not use this configuration. 11:non-inverted/double edge

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				The circuit is sensitive to both TIxFP1 rising and falling edges (capture or trigger operations in reset, external clock or trigger mode). TIxFP1 is not inverted (trigger operation in gated mode). This configuration must not be used in encoder mode. Notes: Once the LOCK level (LOCK bit in the TIMx_BDTR register) is set to 3 or 2, this bit cannot be modified
0	CC1E	RW	0	Capture/Compare 1 output enable 0:Off - OC1 is not active. 1:On - OC1 signal is output on the corresponding output pin depending on MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits. Notes: For complementary output channels, this bit is preloaded. If the CCPC bit is set in the TIMx_CR2 register then the CC1E active bit takes the new value from the preloaded bit only when a Commutation event is generated.

Table 23-2 Output control bits for complementary OCx and OCxN channels with break feature

Control bits					Output state	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx output state	OCxN output state
1	X	0	0	0	Output disabled (not driven by the timer), OCx=0, OCx_EN=0	Output disabled (not driven by the timer), OCxN=0, OCxN_EN=0
		0	0	1	Output disabled (not driven by the timer), OCx=0, OCx_EN=0	OCxREF + Polarity OCxN = OCxREF xor CCxNP, OCxN_EN = 1
		0	1	0	OCxREF+Polarity OCx = OCREF xor CCxP, OCx_EN=1	Output disabled (not driven by the timer), OCxN=0, OCxN_EN=0
		0	1	1	OCREF + Polarity + dead-time OCx_EN = 1	Complementary to OC-REF (not OCREF) + Po-larity + dead-time OCxN_EN=1
		1	0	0	Output disabled (not driven by the timer), OCx=CCxP, OCx_EN=0	Output disabled (not driven by the timer), OCxN=CCxNP, OCxN_EN=0

Puya Confidential

Control bits					Output state	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx output state	OCxN output state
		1	0	1	Output disabled (not driven by the timer), OCx=CCxP, OCx_EN=1	OCxREF+Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF+Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state), OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF + Polarity + dead-time OCx_EN = 1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
0	0	X	0	0	Output disabled (not driven by the timer anymore).	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1	Off-State (output enabled with inactive state) Asynchronously:OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1 If the clock is present:OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state.	
	1		1	0		
	1		1	1		

When both outputs of a channel are not used (CCxE = CCxNE = 0), the OISx, OISxN, CCxP and CCxNP bits must be kept cleared.

Note:The state of the external I/O pins connected to the complementary OCx and OCxN channels depends on the OCx and OCxN channel state and AFIO registers.

23.8.8. TIM16/17 counter (TIMx_CNT)

Address offset:0x24

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	CNT [15:0]	RW	0	Counter value

23.8.9. TIM16/17 prescaler (TIMx_PSC)

Address offset:0x28

Puya Confidential

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	PSC [15:0]	RW	0	Prescaler value The counter clock frequency (CK_CNT) is equal to fCK_PSC / (PSC [15:0] +1). PSC contains the value to be loaded in the active pre-scaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in reset mode).

23.8.10. TIM16/17 auto-reload register (TIMx_ARR)

Address offset:0x2c

Reset value:0x0000FFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	ARR [15:0]	RW	FFFF	Auto-reload value ARR is the value to be loaded in the actual auto-reload register. The counter is blocked while the auto-reload value is null.

23.8.11. TIM16/17 repetition counter register (TIMx_RCR)

Address offset:0x30

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	REP [7:0]							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7:0	REP [7:0]	RW	0	Repetition counter value These bits allow the user to set-up the update rate of the compare registers (i.e. periodic transfers from preload to active registers) when preload registers are enabled, as

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				well as the update interrupt generation rate, if this interrupt is enabled. Each time the REP_CNT related downcounter reaches zero, an update event is generated, and it restarts counting from REP value. As REP_CNT is reloaded with REP value only at the repetition update event U_RC, any write to the TIMx_RCR register is not taken in account until the next repetition update event. It means in PWM mode (REP+1) corresponds to: - the number of PWM periods in edge-aligned mode

23.8.12. TIM16/17 capture/compare register 1 (TIMx_CCR1)

Address offset:0x34

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
RW/R															

Bit	Name	R/W	Reset Value	Function
15:0	CCR1 [15:0]	RW/R	0	Capture/Compare 1 value If channel CC1 is configured as output: CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC1 output. If channel CC1 is configured as input: CCR1 is the counter value transferred by the last input capture 1 event (IC1).

23.8.13. TIM16/17 break and dead-time register (TIMx_BDTR)

Address offset:0x44

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DTG [7:0]							

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Note:As the AOE, BKP, BKE, OSSI, OSSR and DTG [7:0] bits maybe write-locked depending on the LOCK configuration, it may be necessary to configure all of them during the first write access to the TIMx_BDTR register.

Bit	Name	R/W	Reset Value	Function
15	MOE	RW	0	Main output enable This bit is cleared asynchronously by hardware as soon as one of the break inputs is active. It is cleared by software or automatically setting depending on the AOE bit. It is acting only on the channels which are configured in output. 0:OC and OCN outputs are disabled or forced to idle state. 1:If the corresponding enable bit (CCxE, CCxNE bits of the TIMx_CCER register) is set, turn on OC and OCN output.
14	AOE	RW	0	Automatic output enable 0:MOE can be set only by software; 1:MOE can be set by software or automatically at the next update event (if none of the break inputs is active). Note:This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIM1_BDTR register).
13	BKP	RW	0	Break polarity 0:Break input BRK is active low; 1:Break input BRK is active high Note:This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIM1_BDTR register). Any write operation to this bit takes a delay of 1 APB clock cycle to become effective.
12	BKE	RW	0	Break enable 0:Break function disabled 1:Break function enabled Note:This bit cannot be modified as long as LOCK level 1 has been set (LOCK bits in TIM1_BDTR register). Any write operation to this bit takes a delay of 1 APB clock cycle to become effective.
11	OSSR	RW	0	Off-state selection for Run mode This bit is used when MOE=1 on channels having a complementary output which are configured as outputs. OSSR bit is not implemented if no complementary output is implemented in the timer.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				See OC/OCN enable description for more details (TIMx capture/compare enable register (TIMx_CCER)). 0:When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal = 0). 1:When inactive, OC/OCN outputs are enabled with their inactive level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1 Note:This bit cannot be modified as long as LOCK level 2 has been set (LOCK bits in TIMx_BDTR register).
10	OSSI	RW	0	Off-state selection for Idle mode This bit is used when MOE=0 and on channels configured as outputs. See OC/OCN enable description for more details (TIMx capture/compare enable register (TIMx_CCER)). 0:When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal = 0). 1:When inactive, OC/OCN outputs are first forced to their idle level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1 Note:This bit cannot be modified as long as LOCK level 2 has been set (LOCK bits in TIMx_BDTR register).
9:8	LOCK [1:0]	RW	00	Lock configuration These bits offer a write protection against software errors. 00:LOCK OFF, no bit is write protected. 01:LOCK Level 1, DTG, BKE, BKP, AOE bits in TIMx_BDTR register and OISx and OISxN bits in TIMx_CR2 register can no longer be written. 10:LOCK Level 2 = LOCK Level 1 + CC Polarity bits (CCxP/CCxNP bits in TIMx_CCER register, as long as the related channel is configured in output through the CCxS bits) as well as OSSR and OSSI bits can no longer be written. 11:LOCK Level 3 = LOCK Level 2 + CC Control bits (OCxM and OCxPE bits in TIMx_CCMRx registers, as long as the related channel is configured in output through the CCxS bits) can no longer be written. Note:The LOCK bits can be written only once after the re-set. Once the TIMx_BDTR register has been written, their content is frozen until the next reset.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
7:0	DTG [7:0]	RW	00000000	Dead-time generator setup This bit-field defines the duration of the dead-time inserted between the complementary outputs. DT correspond to this duration. DTG [7:5] = 0xx = > DT = DTG [7:0] × Tdtg, Tdtg = TDTS; DTG [7:5] = 10x = > DT = (64 + DTG [5:0]) × Tdtg, Tdtg = 2 × TDTS; DTG [7:5] = 110 = > DT = (32 + DTG [4:0]) × Tdtg, Tdtg = 8 × TDTS; DTG [7:5] = 111 = > DT = (32 + DTG [4:0]) × Tdtg, Tdtg = 16 × TDTS; Example:If TDTS = 125ns (8MHZ), the dead time possible value is: 0 to 15875 ns by 125 ns steps; 16 µs to 31750 ns by 250 ns steps; 32 µs to 63µs by 1 us steps; 64 µs to 126 µs by 2 us steps Note:This bit-field cannot be modified as long as LOCK level 1, 2 or 3 has been programmed (LOCK bits in TIMx_BDTR register).

23.8.14. TIM16/17 DMA control register (TIMx_DCR)

Address offset:0x48

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	DBL [4:0]					Res			DBA [4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:13	Reserved	-	-	Reserved
12:8	DBL [4:0]	RW	00000	DMA burst length These bits define the transfer length of the DMA in continuous mode (when reading or writing to the address of the TIMx_DMAR register address is performed). 00000:1 transfer 00001:2 transfers 00010:3 transfers ....

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				10001:18 transfers Example:We consider such a transmission:DBL = 7, DBA = TIM2_CR1 If DBL = 7, DBA = TIM2_CR1 represents the address of the data to be transmitted, then the transmitted address is given by: (address of TIMx_CR1) + DBA + (DMA index), where DMA index = DBL Where (address of TIMx_CR1) + DBA plus 7 gives the address where data will be written or read out, so that data transfer will occur in 7 registers starting from address (address of TIMx_CR1) + DBA. Depending on the setting of the DMA data length, the following can occur: -If the data is set to half word (16 bits), then the data will be transferred to all 7 registers. -If the data is set to bytes, the data will still be transferred to all 7 registers:the first register contains the first MSB byte, the second register contains the first LSB byte, and so on. Therefore, for the timer, the user must specify the data width to be transmitted by the DMA.
7:5	Reserved	-	-	Reserved
4:0	DBA [4:0]	RW	00000	DBA [4:0]:DMA base address These bits define the base address of the DMA in continuous mode (when read/write access aredone through the TIMx_DMAR address). DBA is defined as an offset starting from the address of the TIMx_CR1 register. 00000:TIMx_CR1 00001:TIMx_CR2 00010:TIMx_SMCR .....

23.8.15. TIM16/17 DMA address for full transfer (TIMx_DMAR)

Address offset:0x4C

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB [15:0]															
RW															

Puya Confidential

Bit	Name	R/W	Reset Value	Function
15:0	DMAB [31:0]	RW	0	DMA register for burst accesses A read or write operation to the DMAR register accesses the register located at the address: (TIMx_CR1 address) + (DBA + DMA index) x 4 where: TIMx_CR1 address is the address of the control register 1; DBA is the DMA base address configured in TIMx_DCR register; DMA index is automatically controlled by the DMA transfer, and ranges from 0 to DBL (DBL configured in TIMx_DCR).

Note:When using the DMA continuous transfer function, the value of the CNDTR register of the corresponding channel in the DMA must be corresponded to the value of the DBL in the TIMx_DCR register, otherwise it will not be used properly.

Puya Confidential

24. Low-power timer (LPTIM)

24.1. Introduction

The LPTIM is a 16-bit timer. The ability of LPTIM to wake up the system from low power mode makes it suitable for implementing low power applications.

LPTIM can count using high frequency clocks (PCLK) and low frequency clocks (LSI/LSE), providing the required functionality and performance while minimizing power consumption.

24.2. LPTIM main features

- 16 bit upcounter
- 3-bit prescaler with 8 possible dividing factors (1, 2, 4, 8, 16, 32, 64, 128)
- Selectable clock
  - Internal clock sources:LSE, LSI or APB clock (PCLK)
- 16 bit ARR reload register
- Continuous/One-shot mode

24.3. LPTIM functional description

24.3.1. LPTIM block diagram

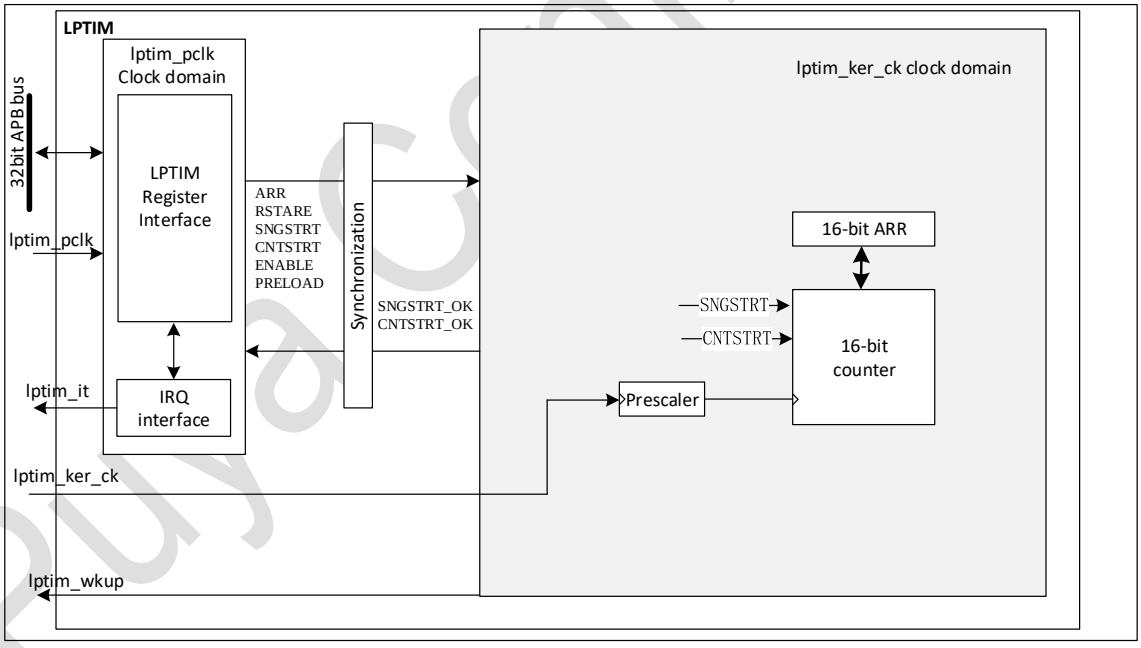


Figure 24-1 Low power timer block diagram

24.3.2. LPTIM pins and internal signals

Table 24-1 LPTIM pins and internal signals

Names	Signal type	Description
lptim_pclk	Input	LPTIM APB clock
lptim_ker_ck	Input	LPTIM internal clock source

Puya Confidential

Names	Signal type	Description
lptim_it	Output	LPTIM global interrupt
lptim_wkup	Output	LPTIM Wakeup event

24.3.3. LPTIM reset and clocks

The LPTIM can be clocked using several clock sources.

It can be clocked using an internal clock signal which can be chosen among PCLK, LSI and LSE sources through the reset and clock controller (RCC).

24.3.4. Prescaler

The LPTIM 16-bit counter is preceded by a configurable power-of-2 prescaler. The prescaler division ratio is controlled by the PRESC [2:0].

The table below lists all the possible division ratios:

Table 24-2 Prescaler division ratios

Programming	Dividing factor
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

Note:When the LPTIM internal clock source selects PCLK (configured via RCC_CCIPR.LPTIMSEL), the prescalation factor cannot be configured to 3 'b000.

24.3.5. Operating mode

The LPTIM has two operating modes:

- **One-shot mode:**the timer is started from a trigger event (writing to the SNGSTRT register) and stops when reaching the ARR value.

To enable the one-shot counting, the SNGSTRT bit must be set.

A new trigger event will re-start the timer. Any trigger event occurring after the counter starts and before the counter reaches ARR will be discarded.

- **Continuous mode:**the timer is free running, the timer is started from a trigger event (writing to the CNTSTRT register) and never stops until the timer is disabled

To enable the continuous counting, the LPTIM_CR. CNTSTRT bit must be set.

LPTIM_CR. CNTSTRT is set will start the counter for continuous counting.

It is possible to change on the fly from One-shot mode to Continuous mode:

Puya Confidential

- If the Continuous mode was previously selected, setting SNGSTRT will switch the LPTIM to the One-shot mode. The counter (if active) will stop as soon as it reaches ARR.
- If the One-shot mode was previously selected, setting LPTIM_CR. CNTSTRT will switch the LPTIM to the Continuous mode. The counter (if active) will restart as soon as it reaches ARR.

24.3.6. Register update

The PRELOAD bit controls how the LPTIM_ARR register is updated:

- When the PRELOAD bit is reset to ‘0’, the LPTIM_ARR and the LPTIM_CMP registers are immediately updated after any write access.
- When the PRELOAD bit is set to ‘1’, the LPTIM_ARR and the LPTIM_CMP registers are updated at the end of the current period, if the timer has been already started.

The LPTIM APB interface and the LPTIM kernel logic use different clocks, so there is some latency between the APB write and the moment when these values are available to the counter comparator. Within this latency period, any additional write into these registers must be avoided.

The ARROK flag in the LPTIM_ISR register indicates when the write operation is completed to the LPTIM_ARR register.

After a write to the LPTIM_ARR register, a new write operation to the same register can only be performed when the previous write operation is completed. Any successive write before respectively the ARROK flag be set, will lead to unpredictable results.

24.3.7. Counter mode

The LPTIM counter can be used to count internal clock cycles only.

The ENABLE bit located in the LPTIM_CR register is used to enable/disable the LPTIM kernel logic. After setting the ENABLE bit, a delay of three counter clock is needed before the LPTIM is actually enabled.

The LPTIM_CFGR and LPTIM_IER registers must be modified only when the LPTIM is disabled.

24.3.8. Timer counter reset

In order to reset the content of LPTIM_CNT register to zero, reset mechanism is implemented:

The synchronous reset mechanism:

The synchronous reset is controlled by the COUNTRST bit in the LPTIM_CR register. After setting the COUNTRST bit-field to '1', the reset signal is propagated in the LPTIM kernel clock domain. So it is important to note that a few clock pulses of the LPTIM kernel logic will elapse before the reset is taken into account. This will make the LPTIM counter count few extra pluses between the time when the reset is trigger, and it becomes effective.

Since the COUNTRST bit is located in the APB clock domain and the LPTIM counter is located in the LPTIM kernel clock domain, a delay of 3 clock cycles of the kernel clock is needed to synchronize the reset signal issued by the APB clock domain when writing '1' to the COUNTRST bit.

The asynchronous reset mechanism:

Puya Confidential

The asynchronous reset is controlled by the RSTARE bit located in the LPTIM_CR register. When this bit is set to '1', any read access to the LPTIM_CNT register will reset its content to zero. Asynchronous reset should be triggered within a timeframe in which no LPTIM core clock is provided.

It should be noted that to read reliably the content of the LPTIM_CNT register two successive read accesses must be performed and compared. A read access can be considered reliable when the value of the two read accesses is equal. Unfortunately, when asynchronous reset is enabled there is no possibility to read twice the LPTIM_CNT register.

Note:After RSTARE is enabled, reading the LPTIM_CNT register requires 4 LPTIM Kernel clocks apart.

24.3.9. Debug mode

When the microcontroller enters debug mode, the LPTIM counter either continues to work normally or stops, depending on the DBG_LPTIM_STOP configuration bit in the DBG module.

24.4. LPTIM low-power modes

Table 24-3 Effect of low-power modes on the LPTIM

Mode	Description
Sleep	No effect. LPTIM interrupts cause the device to exit Sleep mode.
Stop	No effect when LPTIM is clocked by LSE or LSI. LPTIM interrupts cause the device to exit Stop mode.

24.5. LPTIM interrupts

The following events generate an interrupt/wake-up event, if they are enabled through the LPTIM_IER register:

- Auto-reload match

Note:If any bit in the LPTIM_IER register (interrupt enable register) is set after that its corresponding flag in the LPTIM_ISR register (status register) is set, the interrupt is not asserted.

Table 24-4 LPTIM interrupt events

Interrupt event	Description
Auto-reload match	Interrupt flag is raised when the content of the Counter register (LPTIM_CNT) matches the content of the Auto-reload register (LPTIM_ARR).
Auto-reload register update OK	Interrupt flag is raised when the write operation to the LPTIM_ARR register is complete.

24.6. LPTIM registers

24.6.1. LPTIM interrupt and status register (LPTIM_ISR)

Address offset:0x000

Reset value:0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARROK	Res	Res	ARRM	Res
-	-	-	-	-	-	-	-	-	-	-	R	-	-	R	-

Bit	Name	R/W	Reset Value	Function
31:5	Reserved	-	-	Reserved
4	ARROK	R	0	Auto-reload register update OK ARROK is set by hardware to inform application that the APB bus write operation to the LPTIM_ARR register has been successfully completed. ARROK flag can be cleared by writing 1 to the ARROKCF bit in the LPTIM_ICR register.
3:2	Reserved	-	-	Reserved
1	ARRM	R	0	Auto-reload match ARRM is set by hardware to inform application that LPTIM_CNT register's value reached the LPTIM_ARR register's value. ARRM flag can be cleared by writing 1 to the ARRMCF bit in the LPTIM_ICR register.
0	Reserved	-	-	Reserved

24.6.2. LPTIM interrupt clear register (LPTIM_ICR)

Address offset:0x004

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARROKCF	Res	Res	ARRMCF	Res
-	-	-	-	-	-	-	-	-	-	-	RW	-	-	RW	-

Bit	Name	R/W	Reset Value	Function
31:5	Reserved	-	-	Reserved
4	ARROKCF	W	0	Auto-reload register update OK clear flag Writing 1 to this bit clears the ARROK flag in the LPTIM_ISR register
3:2	Reserved	-	-	Reserved
1	ARRMCF	W	0	Auto-reload match clear flag

Dave Confidential

				Writing 1 to this bit clears the ARRM flag in the LPTIM_ISR register
0	Reserved	-	-	Reserved

24.6.3. LPTIM interrupt enable register (LPTIM_IER)

Address offset:0x008

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARROKIE	Res	Res	ARRMIE	Res
-	-	-	-	-	-	-	-	-	-	-	RW	-	-	RW	-

Bit	Name	R/W	Reset Value	Function
31:5	Reserved	-	-	Reserved
4	ARROKIE	RW	0	Auto-reload register update OK interrupt enable 0:ARROK interrupt disabled 1:ARROK interrupt enabled
3:2	Reserved	-	-	Reserved
1	ARRMIE	RW	0	Auto-reload match interrupt enable 0:ARRM interrupt disabled 1:ARRM interrupt enabled
0	Reserved	-	-	Reserved

24.6.4. LPTIM configuration register (LPTIM_CFGR)

Address offset:0x00C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	PRELOAD	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	RW	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	PRESC [2:0]			Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	RW	RW	RW	-	-	-	-	-	-	-	-	-

Bit	Name	R/W	Reset Value	Function
31:23	Reserved	-	-	Reserved
22	PRELOAD	RW	0	Register update mode

Buya Confidential

				The preload bit controls the LPTIM_ARR register update modality 0:Registers are updated after each APB bus write access 1:Registers are updated at the end of the current LPTIM period
21:12	Reserved	-	-	Reserved
11:9	PRESC [2:0]	RW	0	Clock prescaler The PRESC bits configure the prescaler division factor. It can be one among the following division factors: 000:/1 001:/2 010:/4 011:/8 100:/16 101:/32 110:/64 111:/128
8:0	Reserved	-	-	Reserved

24.6.5. LPTIM control register (LPTIM_CR)

Address offset:0x010

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
-	-	-	-	-	-	-	-	-	-	-	RW	RS	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:5	Reserved	-	-	Reserved
4	RSTARE	RW	0	Reset after read enable This bit is set and cleared by software. When RSTARE is set to '1', any read access to LPTIM_CNT register will asynchronously reset LPTIM_CNT register content.
3	COUNTRST	RS	0	Timer counter reset

Buyer Confidential

Bit	Name	R/W	Reset Value	Function
				This bit is set by software and cleared by hardware. When set to '1' this bit will trigger a synchronous reset of the LPTIM_CNT counter register. Due to the synchronous nature of this reset, it only takes place after a synchronization delay of 3 LPTIM core clock cycles (LPTIM core clock may be different from APB clock). Caution:COUNTRST must never be set to '1' by software before it is already cleared to '0' by hardware. Software should consequently check that COUNTRST bit is already cleared to '0' before attempting to set it to '1'.
2	CNTSTRT	RW	0	Timer start in Continuous mode This bit is set by software and setting this bit starts the LPTIM in continuous mode. If this bit is set when a single pulse mode counting is ongoing, then the timer will not stop at the next match between the LPTIM_ARR and LPTIM_CNT registers. The LPTIM counter keeps counting in continuous mode. Note:This bit can be set only when the LPTIM is enabled. It will be automatically reset by hardware.
1	SNGSTRT	RW	0	LPTIM start in One-shot mode This bit is set by software and cleared by hardware. Setting this bit starts the LPTIM in single pulse mode Note:This bit can be set only when the LPTIM is enabled. It will be automatically reset by hardware.
0	ENABLE	RW	0	LPTIM enable. The ENABLE bit is set and cleared by software. 0:LPTIM disabled 1:LPTIM enabled

24.6.6. LPTIM auto-reload register (LPTIM_ARR)

Address offset:0x018

Reset value:0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	ARR	RW	0x0001	Auto reload value ARR is the auto-reload value for the LPTIM. The LPTIM_ARR register must only be modified when the LPTIM is enabled.

24.6.7. LPTIM counter register (LPTIM_CNT)

Address offset:0x01C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	CNT	R	0	Counter value When the LPTIM is running with an asynchronous clock, reading the LPTIM_CNT register may return unreliable values. So in this case it is necessary to perform two consecutive read accesses and verify that the two returned values are identical. A read access can be considered reliable when the values of the two consecutive read accesses are equal.

Puya Confidential

25. Independent watchdog (IWDG)

25.1. Introduction

The devices have an embedded watchdog peripheral which offers a combination of high safety level, timing accuracy and flexibility of use. The independent watchdog peripheral detects and solves malfunctions due to software failure, and triggers system reset when the counter reaches a given timeout value.

The IWDG is clocked by LSI, so even if the main clock fails, it can keep working.

The IWDG is best suited for applications that require the watchdog to run as a totally independent process outside the main application, but have lower timing accuracy constraints.

25.2. IWDG main features

- Free-running downcounter
- clocked from an independent RC oscillator (can operate in Stop mode)
- After starting IWDG in hardware or software mode, the RCC module automatically enables LSI as the IWDG clock.
- When the watchdog is activated, a reset occurs when the counter counts to 0x000

25.3. IWDG functional description

25.3.1. IWDG block diagram

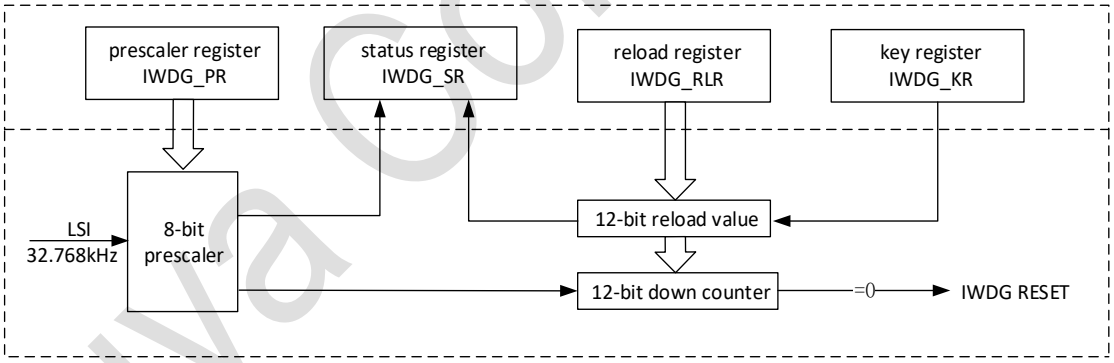


Figure 25-1 IWDG block diagram

Note:The watchdog function is located in the V_{DDD} voltage domain, still functional in Stop and Standby modes.

When the independent watchdog is started by writing the value 0xCCCC in the IWDG_KR register, the counter starts counting down from the reset value of 0xFFF. When it reaches the end of count value (0x000) a reset signal is generated (IWDG_RESET).

Whenever the key value 0x0000 AAAA is written in the IWDG_KR register, the IWDG_RLR value is reloaded in the counter and the watchdog reset is prevented.

Puya Confidential

Table 25-1 Watchdog timeout 32.768 kHz input clock (LSI)

Prescaler division ratios	PR [2:0] bit	Minimum time (ms) RL [11:0] = 0x000	Maximum time (ms) RL [11:0] = 0xFFF
/4	0	0.125	511.875
/8	1	0.25	1023.75
/16	2	0.5	2047.5
/32	3	1	4095
/64	4	2	8190
/128	5	4	16380
/256	(6 or 7)	8	32760

Note:These times are given as per the 32.768 kHz clock. Furthermore, even if the frequency of the RC oscillator is accurate, the exact timing still depends on the phase difference between the APB interface clock and the RC oscillator clock, so there will always be a complete RC cycle that is uncertain. A relatively accurate watchdog timeout can be obtained by calibrating the LSI.

25.3.2. Hardware watchdog

If the Hardware watchdog feature is enabled through the device option bits, the watchdog is automatically enabled at power-on, and will generate a reset unless the IWDG_KR register is written by the software before the counter reaches end of count.

25.3.3. Register access protection

Write access to the IWDG_PR and IWDG_RLR registers is protected. To modify them, first write the code 0x5555 in the IWDG_KR register. A write access to this register with a different value will break the sequence and register access will be protected again. This implies that it is the case of the reload operation (writing 0xAAAA).

A status register is available to indicate that an update of the prescaler or of the downcounter reload value is ongoing.

25.3.4. Debug mode and Stop mode

This function only exists when the system supports debug mode.

When the device enters Debug mode, the IWDG counter either continues to work normally or stops, depending on the configuration of the DBG_IWDG_STOP bit in DBG module.

If the CPU enters the STOP low power mode, there is an IWDG_STOP bit in the Option byte in Flash. The IWDG_STOP bit can control whether the IWDG continues to count normally or freezes the timer when the CPU enters the STOP low power mode.

The configuration of IWDG_STOP in the option byte is as follows:

Set the running status of IWDG timer in STOP mode:

0:Freeze timer

Puya Confidential

1:Normal operation

25.4. IWDG registers

The peripheral registers have to be accessed by half-words (16 bits) or words (32 bits).

25.4.1. IWDG key register (IWDG_KR)

Address offset:0x00

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEY [15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	KEY [15:0]	W	0x00	Key value These bits must be written by software at regular intervals with the key value 0xAAAA, otherwise the watchdog generates a reset when the counter reaches 0. Writing the key value 0x5555 to enable access to the IWDG_PR and IWDG_RLR registers; Writing the key value 0xCCCC starts the watchdog (except if the hardware watchdog option is selected).

25.4.2. IWDG prescaler register (IWDG_PR)

Address offset:0x04

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PR [2:0]		
-	-	-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:3	Reserved	-	-	Reserved
2:0	PR [2:0]	RW	0	Prescaler divider They are written by software to select the prescaler divider feeding the counter clock.

Puya Confidential

				PVU bit of the IWDG status register (IWDG_SR) must be reset in order to be able to change the prescaler divider. 000:divider /4 001:divider /8 010:divider /16 011:divider /32 100:divider /64 101:divider /128 110:divider /256 111:divider /256

25.4.3. IWDG reload register (IWDG_RLR)

Address offset:0x08

Reset value:0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	RL [11:0]											
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:12	Res	-	-	Reserved
11/0	RL [11:0]	RW	0xFFF	IWDG counter reload value They are written by software to define the value to be loaded in the watchdog counter each time the value0xAAAA is written in the IWDG key register (IWDG_KR). The watchdog counter counts down from this value. The timeout period is a function of this value and the clock prescaler. The RVU bit in the IWDG status register (IWDG_SR) must be reset to be able to change the reload value. In addition, it should be noted that when the reload value is written to the RLR register, it needs to wait for three LSI clocks to read it out.

25.4.4. IWDG status register (IWDG_SR)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

Puya Confidential

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RVU	PVU
-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R

Bit	Name	R/W	Reset Value	Function
31:2	Reserved	-	-	Reserved
1	RVU	R	0	Watchdog counter reload value update Hardware and software must start IWDG before the RVU can work properly This bit is set by hardware to indicate that an update of the window value is ongoing. It is reset by hardware when the re-load value update operation is completed.
0	PVU	R	0	Watchdog prescaler value update Hardware and software must start IWDG before the PVU can work properly This bit is set by hardware to indicate that an update of the prescaler value is ongoing. It is reset by hardware when the prescaler update operation is completed.

Note:It is mandatory to wait until RVU bit is reset before changing the reload value, to wait until PVU bit is reset before changing the prescaler value. However, after updating the prescaler and/or the reload value, it is not necessary to wait until RVU or PVU is reset before continuing code execution.

Puya Confidential

26. Window watchdog (WWDG)

26.1. Introduction

The window watchdog (WWDG) is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the contents of the downcounter before the T6 bit becomes cleared. An MCU reset is also generated if the 7-bit downcounter value (in the control register) is refreshed before the downcounter has reached the window register value. This implies that the counter must be refreshed in a limited window.

The WWDG clock is prescaled from the APB clock and has a configurable time-window that can be programmed to detect abnormally late or early application behavior.

The WWDG is best suited for applications which require the watchdog to react within an accurate timing window.

26.2. WWDG main features

- Programmable free-running downcounter
- Reset conditions
  - Reset (if watchdog activated) when the downcounter value becomes lower than 0x40
  - Reset (if watchdog activated) if the downcounter is reloaded outside the window
- Early wakeup interrupt (EWI):triggered (if enabled and the watchdog activated) when the downcounter is equal to 0x40.

26.3. WWDG functional description

If the watchdog is activated (the WDGA bit is set in the WWDG_CR register) and when the 7-bit downcounter (T[6:0] bits) is decremented from 0x40 to 0x3F (T6 becomes cleared), it initiates a reset. If the software reloads the counter while the counter is greater than the value stored in the window register, then a reset is generated.

The application program must write in the WWDG_CR register at regular intervals during normal operation to prevent an MCU reset. This operation must occur only when the counter value is lower than the window register value and higher than 0x3F. The value to be stored in the WWDG_CR register must be between 0xFF and 0xC0.

Puya Confidential

26.3.1. WWDG block diagram

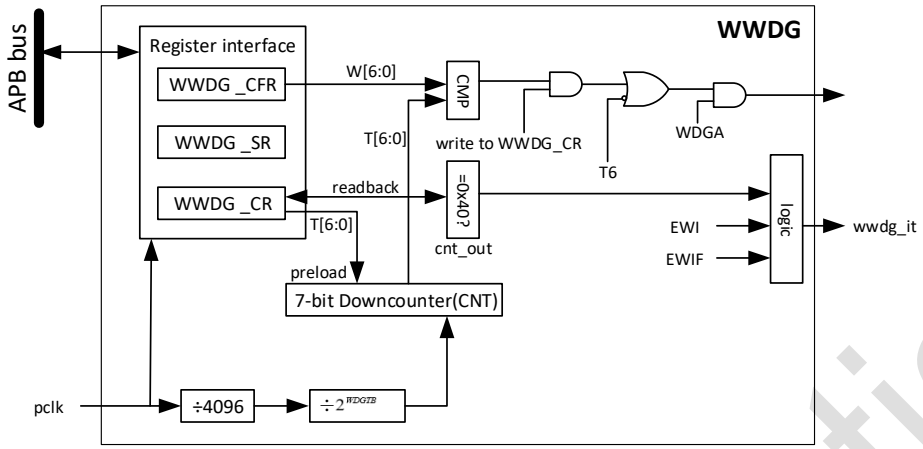


Figure 26-1 Window watchdog block diagram

26.3.2. Enabling the watchdog

After the system is reset, the watchdog is always turned off. Setting the WDGA bit or option byte of the WWDG_CR register can turn on the watchdog, and then it cannot be turned off again unless a reset operation is performed.

There is a WWDG_SW register bit in the option byte, and you can also start a watchdog, whose value is:

0:Hardware window watchdog

1:Software window watchdog

As long as one of the hardware boot and software boot is set, the watchdog will be started.

26.3.3. Controlling the downcounter

This downcounter is free-running, counting down even if the watchdog is disabled. When the watchdog is enabled, the T6 bit must be set to prevent generating an immediate reset.

The T [5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset. The timing varies between a minimum and a maximum value due to the unknown status of the prescaler when writing to the WWDG_CR register. The configuration register (WWDG_CFR) contains the high limit of the window:To prevent a reset, the downcounter must be reloaded when its value is lower than the window register value and greater than 0x3F. Figure below describes the window watchdog process.

Another way to reload the counter is to utilize an early wake-up interrupt (EWI). Setting the WEI bit in the WWDG_CFR register turns on the interrupt. When the decrement counter reaches 0x40, this interrupt is generated, and the corresponding interrupt service routine (ISR) can be used to reload the counter to prevent WWDG from resetting. This interrupt can be cleared by writing '0' in the WWDG_SR register.

Note:A software reset can be generated with the T6 bit (set the WDGA bit to '1' and the T6 bit to '0').

Puya Confidential

26.3.4. Advanced watchdog interrupt feature

The early wakeup interrupt (EWI) can be used if specific safety operations or data logging must be performed before the actual reset is generated. The EWI interrupt is enabled by setting the EWI bit in the WWDG_CFR register. When the downcounter reaches the value 0x40, an EWI interrupt is generated. And the corresponding interrupt service routine (ISR) can be used to trigger specific actions (such as communications or data logging), before resetting the device.

In some applications, the EWI interrupt can be used to manage a software system check and/or system recovery/graceful degradation, without generating a WWDG reset. In this case, the corresponding interrupt service routine (ISR) should reload the WWDG counter to avoid the WWDG reset, then trigger the required actions.

The EWI interrupt is cleared by writing '0' to the EWIF bit in the WWDG_SR register.

Note:When the EWI interrupt cannot be served, e.g. due to a system lock in a higher priority task, the WWDG reset is eventually generated.

26.3.5. How to program the watchdog timeout

When writing to the WWDG_CR register, always write 1 in the T6 bit to avoid generating an immediate reset.

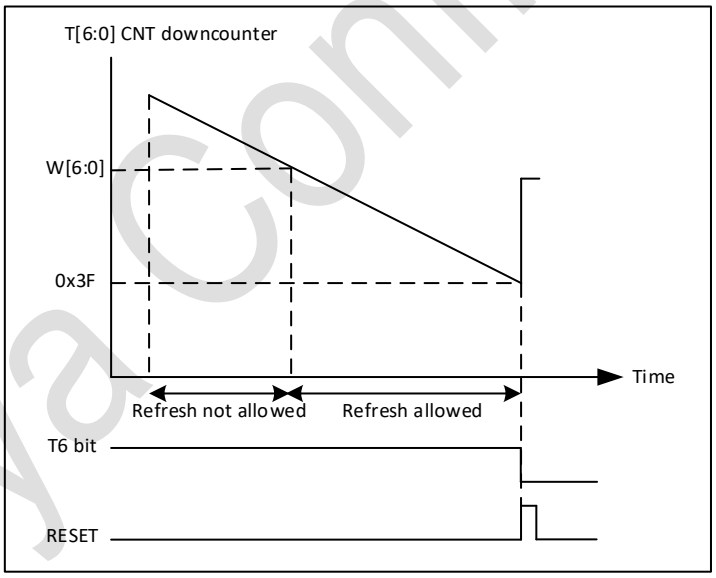


Figure 26-2 Window watchdog timing diagram

The formula to calculate the WWDG timeout value is given by:

$$t_{WWDG} = t_{PCLK} \times 4096 \times 2^{WDGTB[1:0]} \times (T[5:0] + 1) \text{ (ms)}$$

with:

1. t_{WWDG}:WWDG timeout
2. t_{PCLK}:APB clock period measured in ms
3. 4096:value corresponding to internal divider

Puya Confidential

26.3.6. Debug mode

When the microcontroller enters debug mode (Cortex®-M0+ core halted), the WWDG counter either continues to work normally or stops, depending on DBG_WWDG_STOP configuration bit in DBG module. See the Debug support for details.

26.4. WWDG registers

26.4.1. WWDG control register (WWDG_CR)

Address offset:0x00

Reset value:0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	WDGA	T [6:0]						
-	-	-	-	-	-	-	-	RS	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved	-	-	Reserved
7	WDGA	RS	0	WDGA:Activation bit. This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset. 0:Interrupt is disabled 1:Enabled
6:0	T [6:0]	RW	7F	7-bit counter (MSB to LSB). These bits contain the value of the watchdog counter. It is decremented every (4096 x2WDGTB [1:0]) PCLK1 cycles. A reset is produced when it rolls over from 0x40 to 0x3F (T[6] becomes cleared).

26.4.2. WWDG configuration register (WWDG_CFR)

Address offset:0x04

Reset value:0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	EWI	WDGTB [1:0]		T [6:0]						
-	-	-	-	-	-	RS	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9	EWI	RS	0	Early wake-up interrupt When set, an interrupt occurs whenever the counter reaches the value 0x40. This interrupt is only cleared by hardware after a reset.
8:7	WDGTB [1:0]	RW	0	Time base The time base of the prescaler can be modified as follows: 00:CK counter clock (PCLK divided by 4096) divided by 1 01:CK counter clock (PCLK divided by 4096) divided by 2 10:CK counter clock (PCLK divided by 4096) divided by 4 11:CK counter clock (PCLK divided by 4096) divided by 8
6:0	W [6:0]	RW	7F	7-bit window value. These bits contain the window value to be compared to the downcounter.

26.4.3. WWDG status register (WWDG_SR)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EWIF
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RC_W0

Bit	Name	R/W	Reset Value	Function
31:1	Reserved	-	-	Reserved
0	EWIF	RC_W0	0	Early wake-up interrupt flag. This bit is set by hardware when the counter has reached the value 0x40. It must be cleared by software by writing ‘0’. A write of ‘1’ has no effect. This bit is also set if the interrupt is not enabled.

Puya Confidential

27. Real-time clock (RTC)

27.1. Introduction

The real time clock is an independent timer. The RTC provides a set of continuously running counters which can be used, with suitable software, to provide a clock-calendar function. The counter values can be written to set the current time/date of the system.

27.2. RTC main features

- Programmable prescaler:division factor up to 2^{20}
- 32-bit programmable counter for long-term measurement
- Two separate clocks:PCLK for the APB interface and RTC clock (must be at least four times slower than the PCLK clock)
- The RTC clock source could be any of the following ones:
  - HSE clock divided by 128
  - LSI oscillator clock
  - LSE oscillator clock
- Two separate reset types:
  - The APB interface is reset by the system;
  - The RTC Core (Prescaler, Alarm, Counter and Divider) is reset only by a Backup domain reset.
- Three dedicated maskable interrupt lines:
  Alarm interrupt, for generating a software programmable alarm interrupt.
  Seconds interrupt, for generating a periodic interrupt signal with a programmable period length (up to 1 second).
  Overflow interrupt, to detect when the internal programmable counter rolls over to zero.

27.3. RTC functional description

27.3.1. Overview

The RTC consists of two main units. The first one (APB Interface) is used to interface with the APB bus. This unit also contains a set of 16-bit registers (RTC_CRL and RTC_CRH) accessible from the APB bus in read or write mode. This chapter adds the suffix x to the descriptions of all such register banks, such as RTC_CRx), which can be read and written through the APB bus. The APB interface is clocked by the APB bus clock in order to interface with the APB bus.

The other unit (RTC Core) consists of a chain of programmable counters made of two main blocks.

The first block is the RTC prescaler block, which generates the RTC time base TR_CLK that can be programmed to have a period of up to 1 second. It includes a 20-bit programmable divider (RTC pre-scaler). Every TR_CLK period, the RTC generates an interrupt (second interrupt) if it is enabled in the RTC_CR register.

Puya Confidential

The second block is a 32-bit programmable counter that can be initialized to the current system time. The system time is incremented at the TR_CLK rate and compared with a programmable date (stored in the RTC_ALR register) in order to generate an alarm interrupt, if enabled in the RTC_CR control register.

All interrupts can be used as wake-up signals for Stop mode, as shown in the figure below. In Stop mode, the corresponding interrupt of RTC as wake-up requires pre-configuration of the corresponding interrupt enable bit and the corresponding register of EXTI line19.

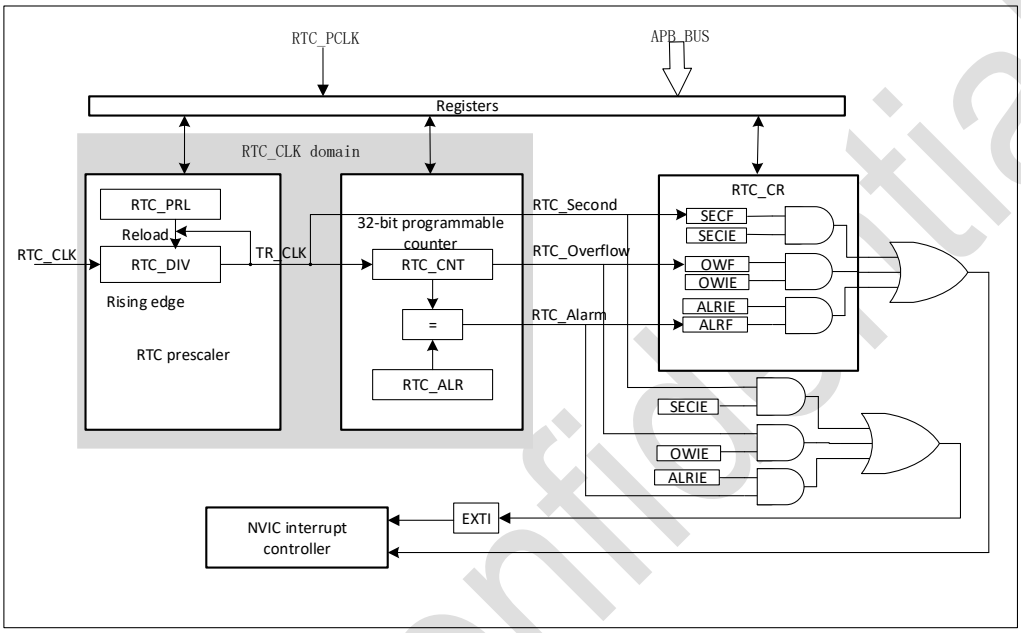


Figure 27-1 RTC block diagram

The DBP bit of the PWR_CR1 register is used to control write protection inhibition of the RTC register. By default, DBP = 0, and write access to the RTC register cannot be performed. Only after the software sets DBP can the write operation to the RTC register be realized.

27.3.2. Resetting RTC registers

The RTC_PRLx, RTC_ALRx, RTC_CNTx, and RTC_DIVx registers can only be reset by an RTC module soft reset or power reset signal. Other system registers (RTC_CRx) are reset asynchronously by system reset or power reset.

Note that when the reset source for reset cannot be generated for the RTC module, not only the RTC module cannot be reset, but also the clock source enable signal sent to the RTC module cannot be reset, and the clock source selection control sent to the RTC module cannot be reset.

27.3.3. Reading RTC registers

The RTC core is completely independent from the RTC APB interface. Software accesses the RTC prescaler, counter and alarm values through the APB interface. But the associated readable registers are internally updated at each rising edge of the RTC clock resynchronized by the RTC APB clock. This is also true for the RTC flags.

Puya Confidential

This means that the first read to the RTC APB registers may be corrupted (generally read as 0) if the APB interface has previously been disabled and the read occurs immediately after the APB interface is enabled but before the first internal update of the registers. This can occur if:

1) A system reset or power reset has occurred
2) The system has just woken up from Stop mode (in this case the count value just won't update because the CPU doesn't work and the system clock stops but the RTC counts normally and the count value won't sync to the V_{DDD} domain)

In all the above cases, the RTC core has been kept running while the APB interface was disabled (reset, not clocked or unpowered).

Consequently, when reading the RTC registers, after having disabled the RTC APB interface, the software must first wait for the RSF bit (register synchronized flag) in the RTC_CRL register to be set by hardware.

Note:

- CPU readable registers include RTC_CRx, RTC_CNTx, RTC_DIVx, and RTC_ALRx;
- The RTC_CRx register is the RTC_PCLK field, and the CPU can read a stable value at any time;
- RTC_CNTx and RTC_DIVx are derived from the RTC_CLK domain. After RTC operation, the RTC_DIVx register is updated at the rising edge of each RTC_CLK; RTC_CNTx and the flag bits originating from the RTC_CLK clock domain also use the same update signal as the RTC_DIVx register, although the value of RTC_CNTx does not change every time it is updated;
- In the RTC_PCLK domain, the RSF is set when the pulse signal after the RTC_CLK is synchronized to the RTC_PCLK is valid;
- RSF only controls the read timing of RTC_CNTx and RTC_DIVx (hardware does not control it)

27.3.4. Configuring RTC registers

To write in the RTC_PRL, RTC_CNT, RTC_ALR registers, the peripheral must enter configuration mode. This is done by setting the CNF bit in the RTC_CRL register

In addition, writing to any RTC register is only enabled if the previous write operation is finished. To enable the software to detect this situation, the RTOFF status bit is provided in the RTC_CRL register to indicate that an update of the registers is in progress. A new value can be written to the RTC registers only when the RTOFF status bit value is ’1’.

Configuration procedure:

1) Poll RTOFF, wait until its value goes to ‘1’;
2) Set the CNF bit to enter configuration mode;
3) Write to one or more RTC registers;
4) Clear the CNF bit to exit configuration mode;
5) Poll RTOFF, wait until its value goes to ‘1’ to check the end of the write operation.

Note:The write operation only executes when the CNF bit is cleared; it takes at least three RTC_CLK cycles to complete. (three RTC_CLK cannot restart the configuration after the CNF flag is cleared, otherwise a configuration error will occur (in this case, it is controlled by RTOFF = 0))

Puya Confidential

Note:

- In this process, RTOFF = 1 when the CPU writes to the register;
- The write cycle of the CPU is from CNF = 1 to CNF = 0, and other registers are configured between these two operations;
- First write CNF to 1 and then clear CNF. This operation clears RTOFF; RTOFF will not be cleared after only writing CNF = 0 or not clearing after writing CNF = 1;
- First write the CNF to 1 and then clear the CNF. This operation writes the cache register value to the RTC_CLK field register;
- RTOFF is implemented in the RTC_PCLK domain.

27.3.5. Setting of RTC flags

Before changing the RTC counter for each RTC_CLK clock cycle, the hardware sets the RTC second flag (SECF). The RTC Overflow Flag (OWF) is set for the last RTC clock cycle before the counter reaches 0x0000.

The RTC_Alarm and RTC alarm flag (ALRF) are set for the RTC clock cycle before the value of the counter reaches the value of the alarm register plus 1 (ALR+1). Writes to RTC alarms must be synchronized with the RTC second flag using one of the following procedures:

(1) Use the RTC alarm interrupt and modify the RTC alarm and/or the RTC counter in the interrupt handler.

(2) Wait for the SECF bit in the RTC control register to be set, and then change the RTC alarm clock and/or RTC counter.

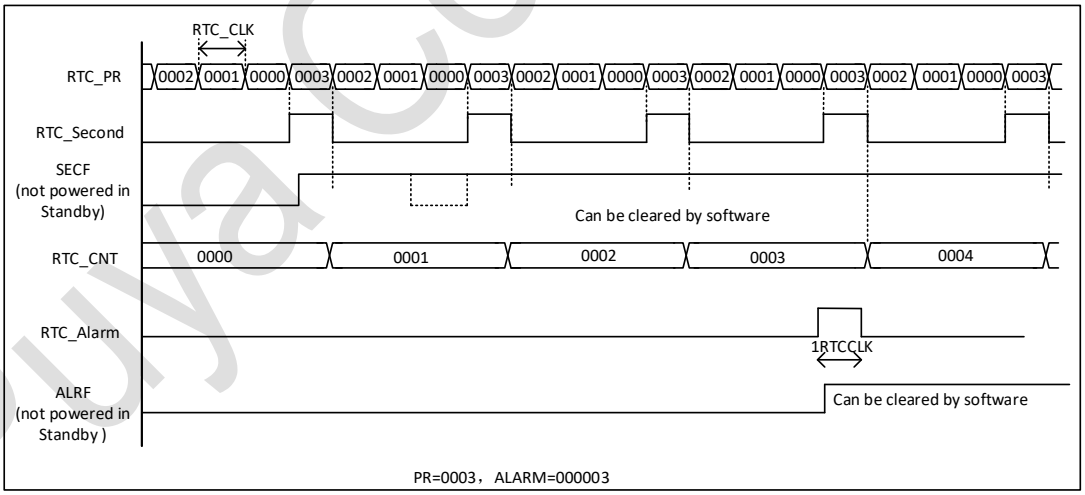


Figure 27-2 RTC second and alarm clock waveform diagram example, PR = 0003, ALR = 00003

Puya Confidential

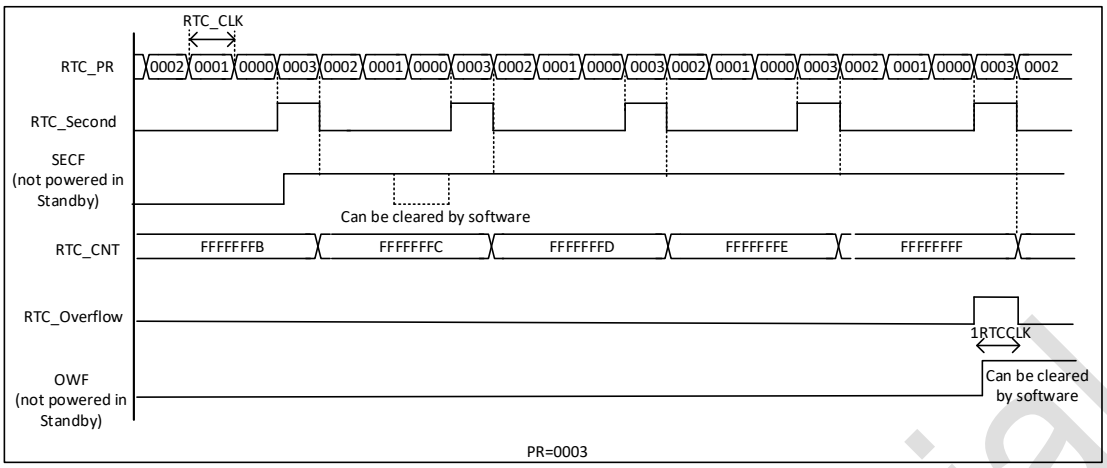


Figure27-3 Example of RTC overflow waveform diagram, PR = 0003

27.3.6. RTC calibration

For measurement purposes, the 64 division of the RTC clock can be output to the IO pin (PF5). This function is implemented by setting CCO bit (BKP_RTCCR register).

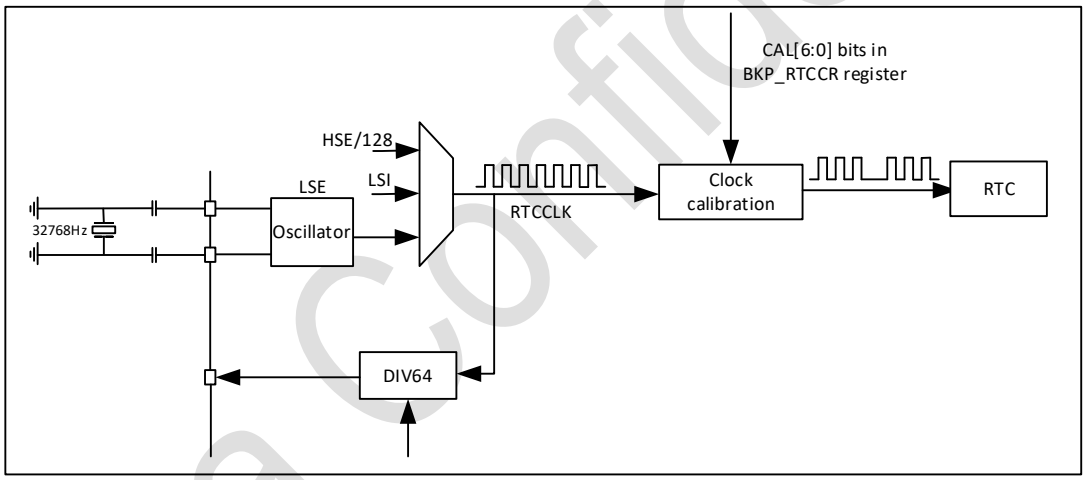


Figure 27-4 RTC calibration diagram

27.4. RTC registers

27.4.1. RTC control register high (RTC_CRH)

Address offset:0x00

Reset value:0x0000 0000

This register is reset by a system reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OWIE	ALRIE	SECIE

Puya Confidential

-	-	-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:3	Reserved	-	-	Reserved
2	OWIE	RW	0	Overflow interrupt enable 0:overflow interrupt disabled 1:overflow interrupt enabled
1	ALRIE	RW	0	Alarm interrupt enable 0:Alarm interrupt disabled 1:Alarm interrupt enabled
0	SECIE	RW	0	Second interrupt enable 0:second interrupt disabled 1:second interrupt enabled

These bits are used to mask interrupt requests. Note that at reset all interrupts are disabled, so it is possible to write to the RTC registers to ensure that no interrupt requests are pending after initialization. It is not possible to write to the RTC_CRH register when the peripheral is completing a previous write operation.

The RTC functions are controlled by this control register. Some bits must be written using a specific configuration procedure.

27.4.2. RTC control register low (RTC_CRL)

Address offset:0x04

Reset value:0x0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res										RTOFF	CNF	RSF	OWF	ALRF	SECF
-										R	RW	RC_W0	RC_W0	RC_W0	RC_W0

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	RTOFF	R	1	RTC operation off, read-only With this bit the RTC reports the status of the last write operation performed on its registers, indicating if it has been completed or not. If its value is ‘0’ then it is not possible to write to any of the RTC registers. 0:Last write operation on RTC registers is still ongoing. 1:Last write operation on RTC registers terminated.

Puya Confidential

4	CNF	RW	0	Configuration flag This bit must be set by software to enter in configuration mode so as to allow new values to be written in the RTC_CNTx, RTC_ALRx or RTC_PRLx registers. The write operation is only executed when the CNF bit is reset by software after has been set. 0:Exit configuration mode (start update of RTC registers). 1:Enter configuration mode
3	RSF	RC_W0	0	Registers synchronized flag This bit is set by hardware at each time the RTC_CNTx and RTC_DIVx registers are updated and cleared by software. After an APB reset or an APB clock stop, this bit must be cleared by software. Before any read operation, the user application must wait until it is set to be sure that the RTC_CNTx, RTC_ALRx or RTC_PRLx registers are synchronized. 0:Registers not yet synchronized. 1:Registers synchronized.
2	OWF	RC_W0	0	Overflow flag This bit is set by hardware when the 32-bit programmable counter overflows. An interrupt is generated if OWIE=1 in the RTC_CRH register. It can be cleared only by software. Writing ‘1’ has no effect. 0:Overflow not detected; 1:32-bit programmable counter overflow occurred.
1	ALRF	RC_W0	0	Alarm flag This bit is set by hardware when the 32-bit programmable counter reaches the threshold set in the RTC_ALRx register. An interrupt is generated if ALRIE=1 in the RTC_CRH register. It can be cleared only by software. Writing ‘1’ has no effect. 0:Alarm not detected 1:Alarm detected
0	SECF	RC_W0	0	Second flag This bit is set by hardware when the 32-bit programmable prescaler overflows, thus incrementing the RTC counter. Hence this flag provides a periodic signal with a period corresponding to the resolution programmed for the RTC counter (usually one second). Aninterrupt is generated if SECIE=1 in the RTC_CRH register. It can be cleared only by software. Writing ‘1’ has no effect. 0:Second flag condition not met. 1:Second flag condition met

Puya Confidential

The functions of the RTC are controlled by this control register. It is not possible to write to the RTC_CRx register while the peripheral is completing a previous write operation (flagged by RTOFF=0).

Notes:

- Any flag remains pending until the appropriate RTC_CR request bit is reset by software, indicating that the interrupt request has been granted.
- At reset the interrupts are disabled, no interrupt requests are pending and it is possible to write to the RTC registers (referring to the write operation from the cache register to the RTC_CLK field register).
- When the APB clock is not running, the OWF, ALRF, SECF and RSF bits are not updated (cannot be synchronized).
- The OWF, ALRF, SECF and RSF bits can only be set by hardware and cleared by software.

27.4.3. RTC prescaler load register (RTC_PRLH)

The prescaler load registers keep the period counting value of the RTC prescaler. They are write-protected by the RTOFF bit in the RTC_CR register, and a write operation is allowed if the RTOFF value is '1' (write to the buffer register).

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRL [19:16]			
-	-	-	-	-	-	-	-	-	-	-	-	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:4	Reserved	-	-	Reserved
3:0	PRL [19:16]	W	0	RTC prescaler reload value high These bits are used to define the counter clock frequency according to the following formula:: fTR_CLK = fRTC_CLK/(PRL [19:0] +1) Note:0 value is not recommended, otherwise RTC interrupt and flag bits cannot be generated correctly

27.4.4. RTC prescaler divider register (RTC_PRLL)

Address offset:0x0C

Reset value:0x8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

Puya Confidential

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRL [15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	PRL [15:0]	W	0x8000	RTC prescaler reload value high These bits are used to define the counter clock frequency according to the following formula:fTR_CLK = fRTC_CLK/(PRL[19:0] +1) Note:0 value is not recommended, otherwise RTC interrupt and flag bits cannot be generated correctly

27.4.5. RTC prescaler divider register high (RTC_DIVH)

At each TR_CLK cycle, the value of the RTC_PRLx register is reloaded into the RTC prescalation counter. The user can obtain an accurate time measurement by reading the RTC_DIVx register to obtain the current value of the pre-division counter without stopping the operation of the division counter.

This register is read-only and will be reloaded by the hardware when there is any change in the value of the RTC_PRLx or RTC_CNTx register.

Address offset:0x10

Reset value:0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RTC_DIV [19:16]			
-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R

Bit	Name	R/W	Reset Value	Function
31:4	Reserved	-	-	Reserved
3:0	RTC_DIV [19:16]	R	0	RTC clock divider high

27.4.6. RTC prescaler divider register low (RTC_DIVL)

Address offset:0x14

Reset value:0x0000 8000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

Puya Confidential

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV [15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	DIV [15:0]	R	0x8000	RTC clock divider low

27.4.7. RTC counter register high (RTC_CNTH)

The RTC core has one 32-bit programmable counter, accessed through two 16-bit registers; the count rate is based on the TR_CLK time reference, generated by the prescaler.

RTC_CNTx registers keep the counting value of this counter. They are write-protected by bit RTOFF in the RTC_CR register, and a write operation is allowed if the RTOFF value is ‘1’. A write operation on the upper (RTC_CNTH) or lower (RTC_CNTL) registers directly loads the corresponding programmable counter and reloads the RTC Prescaler. When reading, the current value in the counter (system date) is returned.

Address offset:0x18

Reset value:0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTC_CNT [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	RTC_CNT [31:16]	RW	0x0000	16 bits of RTC counter high Reading the RTC_CNTH register, the current value of the high part of the RTC counter register is returned. To write to this register it is necessary to enter configuration mode.

27.4.8. RTC counter register low (RTC_CNTL)

Address offset:0x1C

Reset value:0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

Puya Confidential

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTC_CNT [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	RTC_CNT [15:0]	RW	0x0000	16 bit of RTC counter low Reading the RTC_CNTL register, the current value of the lower part of the RTC counter register is returned. To write to this register it is necessary to enter configuration mode.

27.4.9. RTC alarm register high (RTC_ALRH)

When the programmable counter reaches the 32-bit value stored in the RTC_ALRx register, an alarm is triggered. This register is write-protected by the RTOFF bit, and a write operation is allowed if the RTOFF value is ‘1’.

Address offset:0x20

Reset value:0x0000 FFFF

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTC_ALR [31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	ALR [31:16]	RW	0xFFFF	RTC alarm high The high part of the alarm time is written by software in this register. To write to this register it is necessary to enter configuration mode.

27.4.10. RTC alarm register high (RTC_ALRL)

Address offset:0x24

Reset value:0x0000 FFFF

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Puya Confidential

RTC_ALR [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	ALR [15:0]	RW	0xFFFF	RTC alarm low The low part of the alarm time is written by software in this register. To write to this register it is necessary to enter configuration mode.

27.4.11. RTC clock calibration and output configuration register (BKP_RTCCR)

Address offset:0x2C

Reset value:0x0000 0000 (can only be reset by por and bdcr soft reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	ASOS	ASOE	CCO	CAL [6:0]						
-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9	ASOS	RW	0	Second/Alarm pulse output selection When that ASOE bit is set, the ASOS bit can be used to select whether the output on the Pin is an RTC second pulse or an alarm signal 0:RTC alarm clock pulse signal 1:RTC second pulse signal
8	ASOE	RW	0	Second/alarm pulse output enable bit When this bit is set, the ASOS bit determines whether the output on the RTC_OUT pin is an RTC second pulse or an alarm clock pulse signal.
7	CCO	RW	0	Calibration clock output bit When the ASOE is not set, the division of 64 of the CCO output RTC clock can be set 0:No action 1:64 division of RTC clock output on pin
6:0	CAL [6:0]	RW	0	Calibration value

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This value shows the negligible number of clock pulses per 2^{20} clocks, which allows the RTC to calibrate to slow down the clock in steps of $1000000/2^{20}$ PPM. The RTC_CLK may be slowed down from 0 to 121 PPM.

Puya Confidential

28. Inter-integrated circuit (I^2C) interface

28.1. Introduction

The I^2C (inter-integrated circuit) bus interface handles communications between the microcontroller and the serial I^2C bus. It provides multimaster capability, and controls all I^2C bus-specific sequencing, protocol, arbitration and timing. It supports Standard-mode (Sm) and Fast-mode (Fm).

For specific needs, DMA can be used to reduce the burden on the CPU.

28.2. I^2C main features

- Slave and master modes
- Multimaster capability
- Support different communication speeds
  - Standard mode (Sm):up to 100 kHz
  - Fast mode (Fm):up to 400 kHz
- As Master
  - Clock generation
  - Start and Stop generation
- As Slave
  - Programmable I^2C address detection
  - Dual-address capability that responds to two secondary addresses
  - Discovery of the Stop bit
- 7-bit/10-bit addressing mode
- General call
- Status flag
  - Transmit/receive mode flags
  - Byte transfer complete flag
  - I^2C busy flag bit
- I^2S error flag
  - Master arbitration loss
  - ACK failure after address/data transfer
  - Start/Stop error
  - Overrun/Underrun (clock stretching function disable)
- Optional clock stretching
- 1-byte buffer with DMA capability
- Software reset
- Analog noise filter function
- Configurable PEC (packet error checking) generation or verification
  - PEC value can be transmitted as last byte in Tx mode

Puya Confidential

- PEC error checking for last received byte

- Support SMBus
  - 25 ms clock low timeout delay
  - 10 ms master cumulative clock low extend time
  - 25 ms slave cumulative clock low extend time
  - Hardware PEC generation/verification with ACK control
  - Address resolution protocol (ARP) supported

28.3. I²C functional description

28.3.1. I²C block diagram

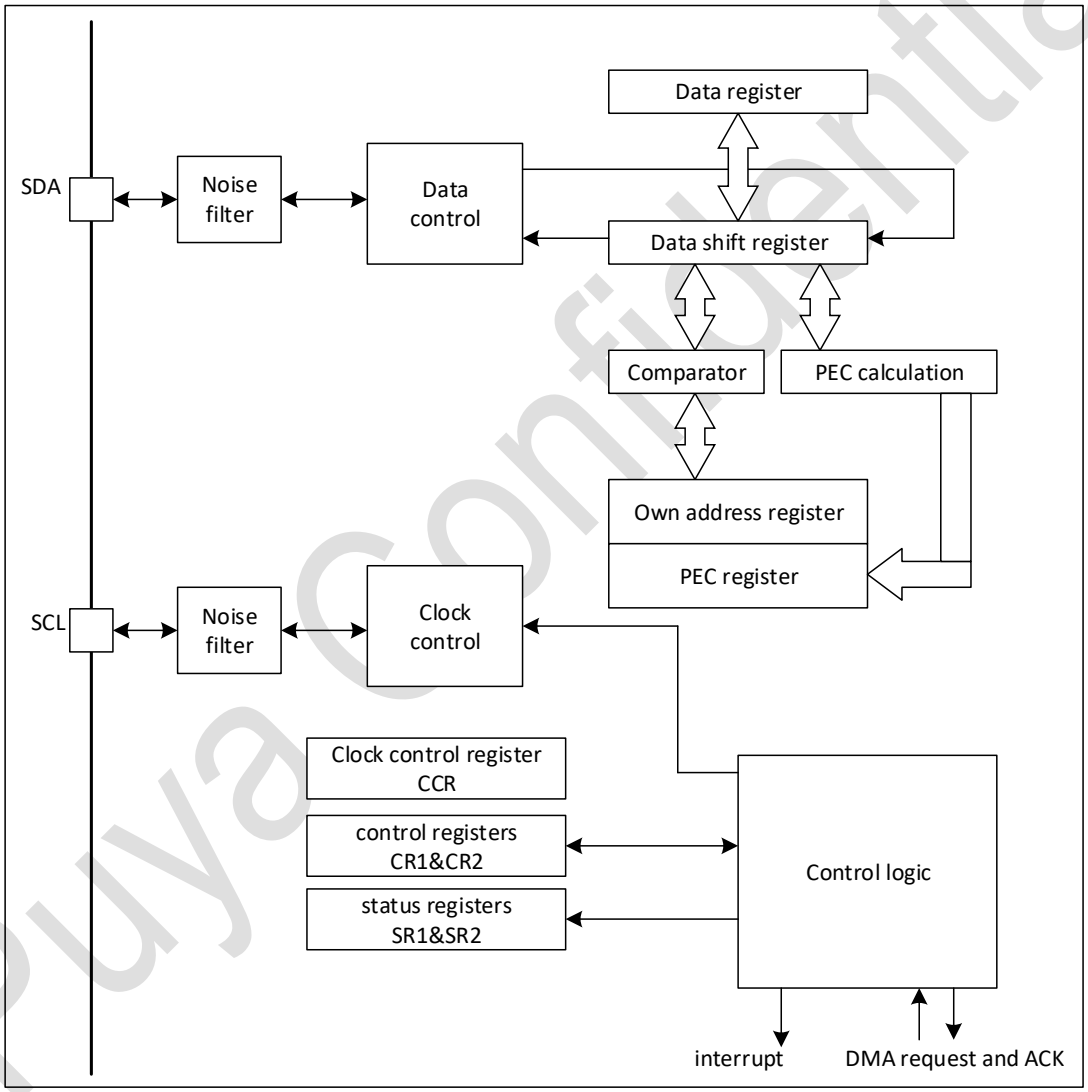


Figure 28-1 I²C block diagram

28.3.2. Mode selection

The interface can operate in one of the four following modes:

- Slave transmitter
- Slave receiver
- Master transmitter

Puya Confidential

- Master receiver

By default, it operates in slave mode. The interface automatically switches from slave to master when it generates a START condition, and from master to slave if an arbitration loss or a STOP generation occurs, allowing multimaster capability.

28.3.2.1. Communication flow

In Master mode, the I²C interface initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated in master mode by software.

In Slave mode, the interface is capable of recognizing its own addresses (7 or 10-bit), and the General Call address The general call address detection may be enabled or disabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the start condition contain the address. The address is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledged bit to the transmitter. Refer to the following figure.

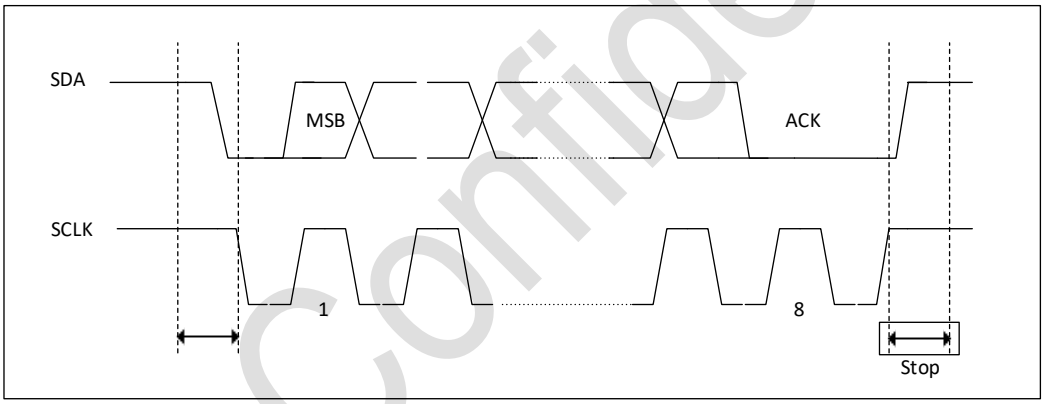


Figure 28-2 I²C bus protocol

Acknowledge may be enabled or disabled by software. The I²C interface addresses (dual addressing 7-bit/10-bit and/or general call address) can be selected by software.

28.3.3. I²C initialization

28.3.3.1. Enable/disable I²C module

The I²C peripheral clock must be configured and enabled by the I2C_EN bit in the RCC_APBENR1 register. Then the I²C can be enabled by setting the PE bit in the I2C_CR1 register.

28.3.3.2. I²C timings

The holding and setup time of the data signal (SDA) is required to meet the I²C standard protocol, and the I²C timing needs to be set. This is achieved by writing to the I2C_CCR and I2C_TRISE registers.

Puya Confidential

28.3.4. I²C slave mode

By default, the I²C interface operates in Slave mode. To switch from default Slave mode to Master mode a Start condition generation is needed.

The peripheral input clock must be programmed in the I2C_CR2 register in order to generate correct timings. The peripheral input clock frequency must be at least:

- 2 MHz in Sm mode
- 4 MHz in Fm mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register. Then it is compared with the address of the interface (OAR1) or the General Call address (if ENGC = 1).

Header or address not matched:

The interface ignores it and waits for another Start condition.

Address matched:

The interface generates in sequence:

- An acknowledged pulse if the ACK bit is set
- The ADDR bit is set by hardware and an interrupt is generated if the ITEVFEN bit is set

The TRA bit indicates whether the slave is in Receiver or Transmitter mode.

28.3.4.1. Slave transmitter

Following the address reception and after clearing ADDR, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave stretches SCL low until ADDR is cleared and DR filled with the data to be sent.

When the acknowledge pulse is received:The TxE bit is set by hardware with an interrupt if the ITEVFEN and the ITBUFEN bits are set.

If TxE is set and some data were not written in the I2C_DR register before the end of the next data transmission, the BTF bit is set. The interface waits until BTF is cleared by a read to I2C_SR1 followed by a write to the I2C_DR register, stretching SCL low.

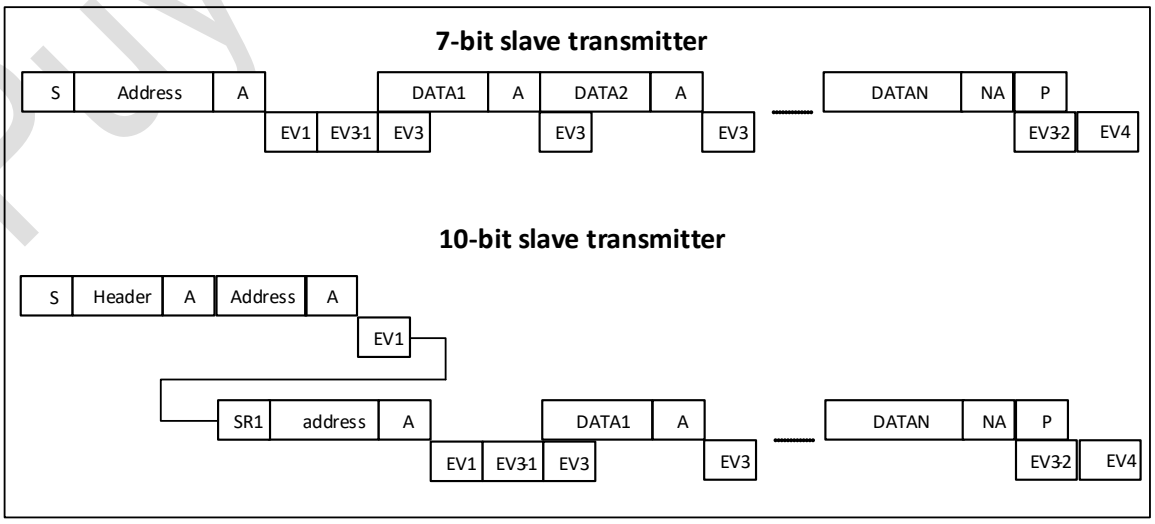


Figure 28-3 Transfer sequence diagram for slave transmitter

Puya Confidential

Legend:S= Start, Sr = Repeated Start, P= Stop, A= Acknowledge, NA= Non-acknowledge, EVx= Event (with interrupt if ITEVFEN=1)

EV1:ADDR=1, cleared by reading SR1 followed by reading SR2

EV3_1: TxE=1, shift register empty, data register empty, write Data1 in DR.

EV3: TxE=1, shift register not empty, data register empty, cleared by writing DR

EV3-2:AF=1; AF is cleared by writing '0'.

EV4:STOPF=1, cleared by reading SR1 register followed by writing to the CR1 register.

Notes:

1) The EV1 and EV3_1 event stretch SCL low until the end of the corresponding software sequence.

2) The EV3 software sequence must be completed before the end of the current byte transfer.

28.3.4.2. Slave receiver

Following the address reception and after clearing ADDR, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- An acknowledge pulse if the ACK bit is set
- The RxNE bit is set by hardware. An interrupt is generated if the ITEVTEN and ITBUFEN bit is set.

If RxNE is set and the data in the DR register is not read before the end of the next data reception, the BTF bit is set, and the interface waits until BTF is cleared by a read fromI2C_SR1 followed by a read from the I2C_DR register, stretching SCL low.

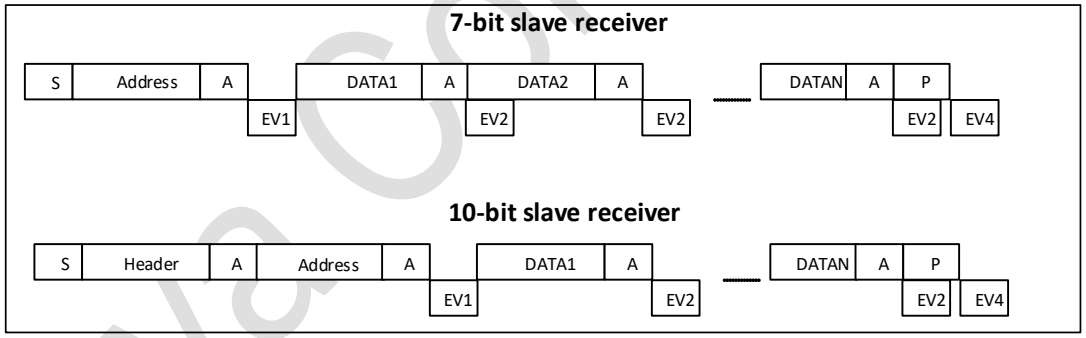


Figure 28-4 Transfer sequence diagram for slave receiver

Legend:S= Start, Sr = Repeated Start, P= Stop, A= Acknowledge, EVx= Event (with interrupt if ITEVFEN=1)

EV1:ADDR=1, cleared by reading SR1 followed by reading SR2

EV2: RxNE=1 cleared by reading DR register

EV4:STOPF=1, cleared by reading SR1 register followed by writing to the CR1 register.

Notes:

— The EV1 event stretches SCL low until the end of the corresponding software sequence.

— The EV2 software sequence must be completed before the end of the current byte transfer.

— After checking the SR1 register content, the user should perform the complete clearing sequence for each flag found set. Thus, for ADDR and STOPF flags, the following sequence is required inside the I^2C interrupt routine:

PUYA Confidential

If ADDR = 1, reading SR1 followed by reading SR2; if STOPF =1, reading SR1 followed by reading CR1. The purpose is to make sure that both ADDR and STOPF flags are cleared if both are found set.

28.3.4.3. Closing slave communication

After the last data byte is transferred, a Stop Condition is generated by the master. The interface detects this condition and sets:

- The STOPF bit is set and generates an interrupt if the ITEVFEN bit is set.
- The STOPF bit is cleared by a read of the SR1 register followed by a write to the CR1 register.

28.3.5. I²C master mode

In Master mode, the I²C interface initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition.

Master mode is selected as soon as the Start condition is generated on the bus with a START bit.

The following is the required sequence in master mode:

- Program the peripheral input clock in I2C_CR2 register in order to generate correct timings
- Configure the clock control registers
- Configure the rise time register
- Program the I2C_CR1 register to enable the peripheral
- Set the START bit in the I2C_CR1 register to generate a Start condition

  The peripheral input clock frequency must be at least:
- 2 MHz in Sm mode
- 4 MHz in Fm mode

28.3.5.1. SCL master clock generation

The CCR bits are used to generate the high and low level of the SCL clock, starting from the generation of the rising edge and falling edge. As a slave may stretch the SCL line, the peripheral checks the SCL input from the bus at the end of the time programmed in TRISE register after rising edge generation.

- If the SCL line is low, it means that a slave is stretching the bus, and the high level counter stops until the SCL line is detected high. This allows to guarantee the minimum HIGH period of the SCL clock parameter.
- If the SCL line is high, the high level counter keeps on counting.

Indeed, the feedback loop from the SCL rising edge generation by the peripheral to the SCL rising edge detection by the peripheral takes time even if no slave stretches the clock. This loop back duration is linked to the SCL rising time (impacting SCL VIH input detection), plus delay due to the noise filter present on the SCL input path, plus delay due to internal SCL input synchronization with APB clock. The maximum time used by the feedback loop is programmed in the TRISE bits, so that the SCL frequency remains stable whatever the SCL rising time.

Puya Confidential

28.3.5.2. Start condition

Setting the START bit causes the interface to generate a Start condition and to switch to Master mode (MSL bit set) when the BUSY bit is cleared.

Note:In master mode, setting the START bit causes the interface to generate a Restart condition at the end of the current byte transfer.

Once the Start condition is sent:

- The SB bit is set by hardware and an interrupt is generated if the ITEVFEN bit is set.

Then the master waits for a read of the I2C_SR1 register followed by a write in the I2C_DR register with the Slave address.

28.3.5.3. Slave address transmission

Then the slave address is sent to the SDA line via the internal shift register.

- In 10-bit addressing mode, sending the header sequence causes the following event:

— The ADDR bit is set by hardware and an interrupt is generated if the ITEVFEN bit is set.

Then the master waits for a read of the I2C_SR1 register followed by a write in the DR register with the second address byte.

The ADDR bit is set by hardware and an interrupt is generated if the TEVFEN bit is set.

Then the master waits for a read of the I2C_SR1 register followed by a read of the I2C_SR2 register.

- In 7-bit addressing mode, one address byte is sent.

As soon as the address byte is sent,

— The ADDR bit is set by hardware and an interrupt is generated if the ITEVTEN bit is set.

Then the master waits for a read of the I2C_SR1 register followed by a read of the I2C_SR2 register.

The master can decide to enter Transmitter or Receiver mode depending on the LSB of the slave address sent.

- In 7-bit addressing mode,

— To enter Transmitter mode, a master sends the slave address with LSB reset.

— To enter Receiver mode, a master sends the slave address with LSB set.

The TRA bit indicates whether the master is in Receiver or Transmitter mode.

- In 10-bit addressing mode,

— To enter Transmitter mode, a master sends the header (11110xx0) and then the slave address, (where xx denotes the two most significant bits of the address).

— To enter Receiver mode, a master sends the header (11110xx0) and then the slave address. Then it should send a repeated Start condition followed by the header (11110xx1),

(where xx denotes the two most significant bits of the address). The TRA bit indicates whether the master is in Receiver or Transmitter mode.

Puya Confidential

28.3.5.4. Master transmitter

Following the address transmission and after clearing ADDR, the master sends bytes from the DR register to the SDA line via the internal shift register.

The master waits until the first data byte is written into DR register (see EV8_1).

When the acknowledge pulse is received, the TxE bit is set by hardware and an interrupt is generated if the ITEVFEN and ITBUFEN bits are set.

If TxE is set and some data were not written in the DR register before the end of the next data transmission, the BTF bit is set. The interface waits until BTF is cleared by a read from I2C_SR1 followed by a write to I2C_DR, stretching SCL low.

Closing slave communication

After the last byte is written to the DR register, the STOP bit is set by software to generate a Stop condition (see EV8_2 in figure below). The interface automatically goes back to slave mode (MSL bit cleared).

Note:Stop condition should be programmed during EV8_2 event, when either TxE or BTF is set.

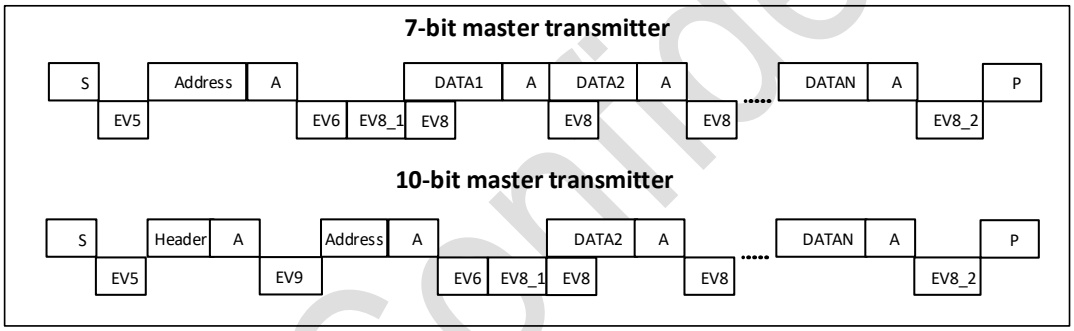


Figure 28-5 Transfer sequence diagram for master transmitter

Legend:S= Start, Sr = Repeated Start, P= Stop, A= Acknowledge, EVx= Event (with interrupt if ITEVFEN=1)

EV5:SB = 1, cleared by reading SR1 register followed by writing DR register with Address.

EV6:ADDR=1, cleared by reading SR1 followed by reading SR2

EV8_1:TXE = 1, shift register empty

EV8:TXE = 1, cleared by writing DR register

EV8_2:TXE = 1, BTF = 1, cleared by hardware by the Stop condition

EV9:ADDR10 = 1, cleared by reading SR1 register followed by writing DR register.

Notes:

1） The EV5, EV6, EV8_1 and EV8_2 events stretch SCL low until the end of the corresponding software sequence.

2） The EV8 software sequence must complete before the end of the current byte transfer. In case EV8 software sequence cannot be managed before the current byte end of transfer, it is recommended to use BTF instead of TXE with the drawback of slowing the communication.

Puya Confidential

28.3.5.5. Master receiver

Following the address transmission and after clearing ADDR, the I²C interface enters Master Receiver mode. In this mode the interface receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- An acknowledge pulse if the ACK bit is set
- The RxNE bit is set and an interrupt is generated if the INEVFEN and ITBUFEN bits are set.

If the RxNE bit is set and the data in the DR register is not read before the end of the last data reception, the BTF bit is set by hardware and the interface waits until BTF is cleared by a read in the I2C_SR1 register followed by a read in the I2C_DR register, stretching SCL low.

Closing slave communication

Method 1:This method is for the case when the I²C is used with interrupts that have the highest priority in the application.

The master sends a NACK for the last byte received from the slave. After receiving this NACK, the slave releases the control of the SCL and SDA lines. Then the master can send a Stop/Restart condition.

1） To generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just after reading the second last data byte (after second last RxNE event).

2） To generate the Stop/Restart condition, software must set the STOP/START bit just after reading the second last data byte (after the second last RxNE event).

3） In case a single byte has to be received, the Acknowledge disable and the Stop condition generation are made just after EV6 (in EV6_1, just after ADDR is cleared).

After the Stop condition generation, the interface goes automatically back to slave mode (MSL bit cleared).

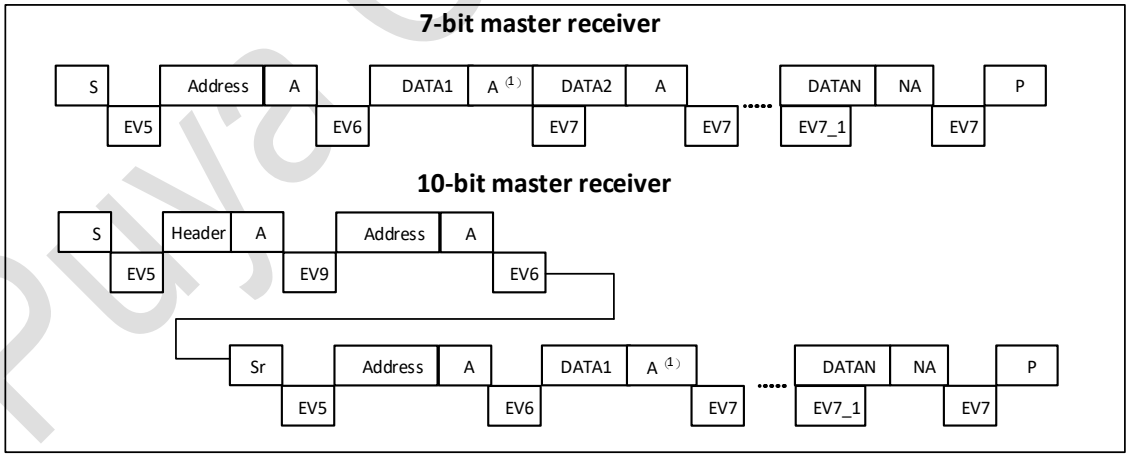


Figure 28-6 Method 1:transfer sequence diagram for master receiver

Legend:S= Start, Sr = Repeated Start, P= Stop, A= Acknowledge, EVx= Event (with interrupt if ITEVFEN=1)

EV5:SB=1, cleared by reading SR1 register followed by writing DR register.

EV6:ADDR=1, cleared by reading SR1 followed by reading SR2

EV6_1:no associated flag event, used for 1 byte reception only.

Puya Confidential

EV7: RxNE=1 cleared by reading DR register.

EV7_1: RxNE=1 cleared by reading DR register, program ACK=0 and STOP request

EV9:ADDR10 = 1, cleared by reading SR1 register followed by writing DR register.

1) If a single byte is received, it is NA.
2) The EV5 and EV6 events stretch SCL low until the end of the corresponding software sequence.
3) The EV7 software sequence must complete before the end of the current byte transfer. In case EV7 software sequence cannot be managed before the current byte end of transfer, it is recommended to use BTF instead of RXNE with the drawback of slowing the communication.
4) The EV6_1 or EV7_1 software sequence must complete before the ACK pulse of the current byte transfer.

Method 2:This method is for the case when the I2C is used with interrupts that do not have the highest priority in the application or when the I2C is used with polling.

With this method, DataN_2 is not read, so that after DataN_1, the communication is stretched (both RxNE and BTF are set). Then, clear the ACK bit before reading DataN-2 in DR register to ensure it is cleared before the DataN Acknowledge pulse. After that, just after reading DataN_2, set the STOP/ START bit and read DataN_1. After RxNE is set, read DataN.

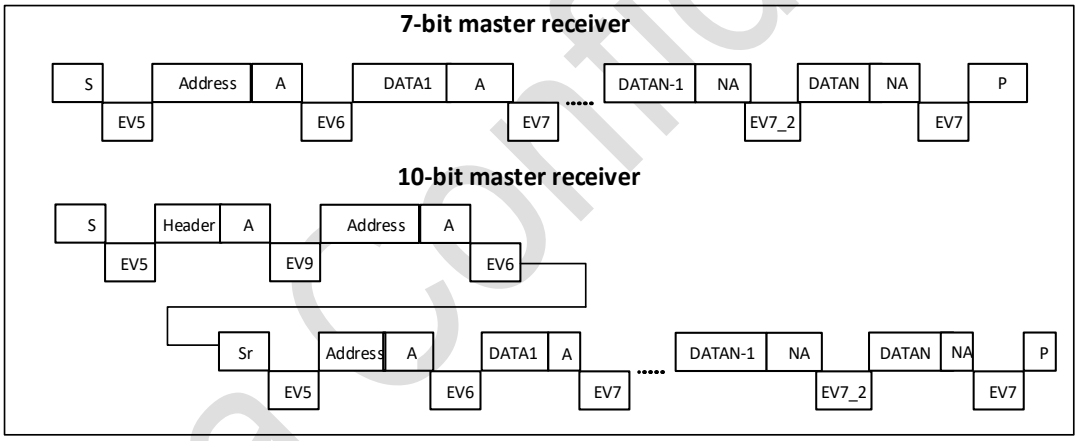


Figure 28-7 Method 2:transfer sequence diagram for master receiver when N>2

Legend:S= Start, Sr = Repeated Start, P= Stop, A= Acknowledge, EVx= Event (with interrupt if ITEVFEN=1)

EV5:SB=1, cleared by reading SR1 register followed by writing the DR register.

EV6:ADDR=1, cleared by reading SR1 followed by reading SR2

EV7: RxNE=1 cleared by reading DR register

EV7_2:BTF = 1, DataN-2 in DR register and DataN-1 in shift register, program ACK = 0, Read DataN-2 in DR. Program STOP = 1, read DataN-1.

EV9:ADDR10 = 1, cleared by reading SR1 register followed by writing DR register.

Notes:

- The EV5 and EV6 events stretch SCL low until the end of the corresponding software sequence.

Puya Confidential

- The EV7 software sequence must complete before the end of the current byte transfer. In case EV7 software sequence cannot be managed before the current byte end of transfer, it is recommended to use BTF instead of RXNE with the drawback of slowing the communication.

- **When 3 bytes remain to be read:**
  - RxNE = 1 => Nothing (DataN-2 not read).
  - DataN-1 received
  - BTF = 1 because both shift and data registers are full:DataN-2 in DR and DataN-1 in the shift register => SCL tied low:no other data will be received on the bus.
  - Clear ACK bit
  - Read DataN-2 in DR => This will launch the DataN reception in the shift register
  - DataN received (with a NACK)
  - Program START/STOP bit
  - Read DataN-1
  - RxNE=1
  - Read DataN

The procedure described above is valid for N>2. The cases where a single byte or two bytes are to be received should be handled differently, as described below:

- **Case of two bytes to be received:**
  - Set POS and ACK bit
  - Wait for the ADDR flag to be set
  - Clear ADDR bit
  - Clear ACK bit
  - Wait for BTF to be set
  - Program STOP bit
  - Read DR twice

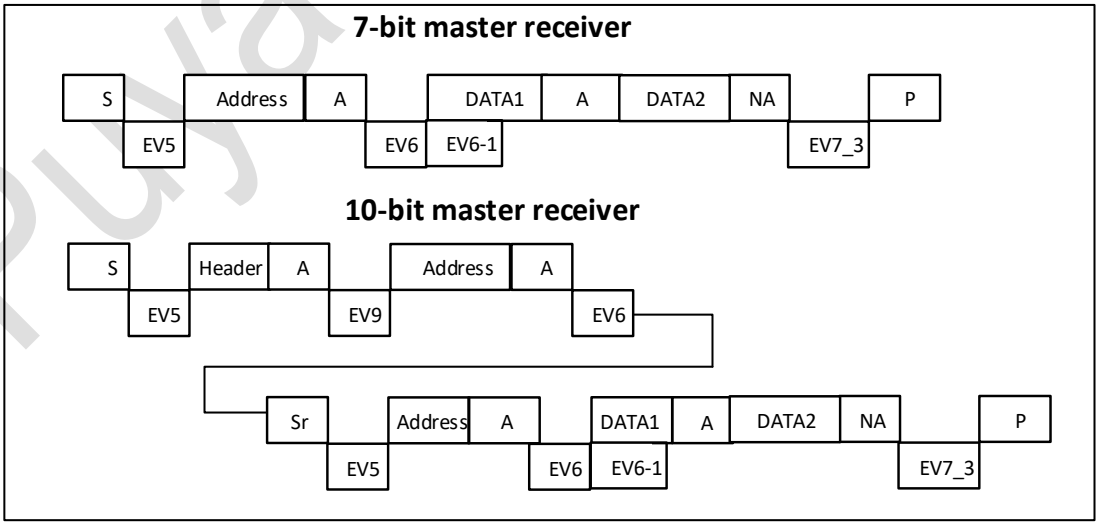


Figure 28-8 Method 2:transfer sequence diagram for master receiver when N=2

Legend:S= Start, Sr = Repeated Start, P= Stop, A= Acknowledge, EVx= Event (with interrupt if ITEVFEN=1)

Puya Confidential

EV5:SB=1, cleared by reading SR1 register followed by writing the DR register.

EV6:ADDR=1, cleared by reading SR1 followed by reading SR2

EV6_1: No associated flag event. The acknowledge disable should be done just after EV6, that is after ADDR is cleared.

EV7_3:BTF = 1, program STOP = 1, read DR twice (Read Data1 and Data2) just after programming the STOP.

EV9:ADDR10 = 1, cleared by reading SR1 register followed by writing DR register.

Notes:

- The EV5 and EV6 events stretch SCL low until the end of the corresponding software sequence.
- The EV6_1 software sequence must complete before the ACK pulse of the current byte transfer.

- **Case of a single byte to be received:**
  - In the ADDR event, clear the ACK bit
  - Clear ADDR
  - Program STOP/START bit
  - Read the data after the RxNE flag is set.

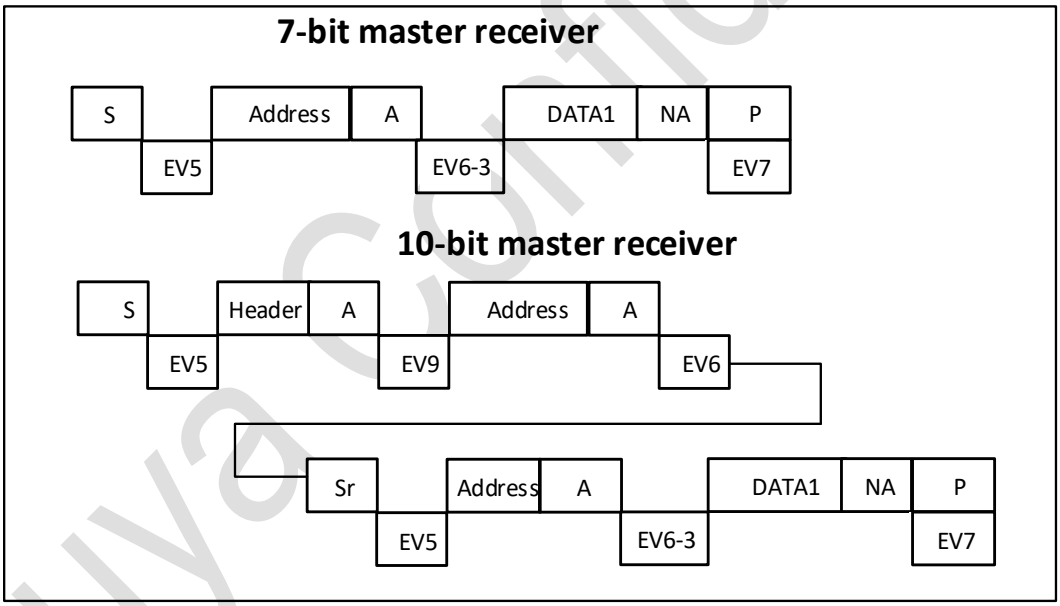


Figure 28-9 Method 2:transfer sequence diagram for master receiver when N=1

Legend:S= Start, Sr = Repeated Start, P= Stop, A= Acknowledge, EVx= Event (with interrupt if ITEVFEN=1)

EV5:SB=1, cleared by reading SR1 register followed by writing the DR register.

EV6_3:ADDR = 1, program ACK = 0. Clear ADDR by reading SR1 register followed by reading SR2 register. Program STOP =1 just after ADDR is cleared.

EV7: RxNE=1 cleared by reading DR register

EV9:ADDR10 = 1, cleared by reading SR1 register followed by writing DR register.

Notes:

Puya Confidential

The EV5, EV6, EV8_1 and EV8_2 events stretch SCL low until the end of the corresponding software sequence.

28.3.6. Error conditions

28.3.6.1. Bus error (BERR)

This error occurs when the I²C interface detects an external Stop or Start condition during an address or a data transfer. In this case:

- The BERR bit is set, and an interrupt is generated if the ITERREN bit is set;
- In Slave mode:data are discarded, and the lines are released by hardware:

  ─ In case of a misplaced Start, the slave considers it is a restart and waits for an address, or a Stop condition

  ─ In case of a misplaced Stop, the slave behaves like for a Stop condition and the lines are released by hardware

- In Master mode:the lines are not released and the state of the current transmission is not affected. It is up to the software to abort or not the current transmission.

28.3.6.2. Acknowledge failure (AF)

This error occurs when the interface detects a non-acknowledge bit. In this case:

- The AF bit is set, and an interrupt is generated if the ITERREN bit is set
- A transmitter which receives a NACK must reset the communication:

  ─ If Slave:lines are released by hardware.

  ─ If Master:a Stop or repeated Start condition must be generated by software

28.3.6.3. Arbitration lost (ARLO)

This error occurs when the I²C interface detects an arbitration lost condition. In this case:

- The ARLO bit is set by hardware (and an interrupt is generated if the ITERREN bit is set)
- The I²C Interface goes automatically back to slave mode (the MSL bit is cleared). When the I²C loses the arbitration, it is not able to acknowledge its slave address in the same transfer, but it can acknowledge it after a repeated Start from the winning master.
- Lines are released by hardware

28.3.6.4. Overrun/underrun error (OVR)

An overrun error can occur in slave mode when clock stretching is disabled and the I²C interface is receiving data. The interface has received a byte (RxNE=1) and the data in DR register has not been read, before the next byte is received by the interface.

In this case:

- The last received byte is lost.
- In case of Overrun error, software should clear the RxNE bit and the transmitter should re-transmit the last received byte.

Underrun error can occur in slave mode when clock stretching is disabled and the I²C interface is transmitting data. The interface has not updated the DR with the next byte (TxE=1), before the clock comes for the next byte. In this case:

Puya Confidential

- The same byte in the DR register will be sent again
- The user should make sure that data received on the receiver side during an underrun error are discarded. The next bytes are written within the clock low time specified in the I2C bus standard.

For the first byte to be transmitted, the DR must be written after ADDR is cleared and before the first SCL rising edge. If not possible, the receiver must discard the first data.

28.3.7. SDA/SCL line control

- If clock stretching is enabled:

— Transmitter mode:If TxE=1 and BTF=1:the interface holds the clock line low before transmission to wait for the microcontroller to read SR1 and then write the byte in the Data register (both DR and shift register are empty).

— Receiver mode:If RxNE=1 and BTF=1:the interface holds the clock line low after reception to wait for the microcontroller to read SR1 and then read the byte in the Data register (both DR and shift register are full).

- If clock stretching is disabled in Slave mode:

— Overrun Error in case of RxNE=1 and no read of DR has been done before the next byte is received. The last received byte is lost.

— Underrun Error in case TxE=1 and no write into DR register has been done before the next byte must be transmitted. The same byte will be sent again.

— Write collision not managed.

28.3.8. DMA request

DMA requests (when enabled) are generated only for data transfer. DMA requests are generated by data register becoming empty in transmission and data register becoming full in reception. The DMA must be initialized and enabled before the I^2C data transfer. The DMAEN bit must be set in the I2C_CR2 register before the ADDR event.

In master mode or in slave mode when clock stretching is enabled, the DMAEN bit can also be set during the ADDR event, before clearing the ADDR flag. The DMA request must be served before the end of the current byte transfer. When the number of data transfers which has been programmed for the corresponding DMA stream is reached, the DMA controller sends an end of transfer (EOT) signal to the I^2C interface and generates a transfer complete interrupt if enabled:

- Master transmitter:In the interrupt routine after the EOT interrupt, disable DMA requests then wait for a BTF event before programming the Stop condition.
- Master receiver:when the number of bytes to be received is equal to or greater than two, the DMA controller sends a hardware signal, EOT_1, corresponding to the last but one data byte (number_of_bytes – 1). If, in the I2C_CR2 register, the LAST bit is set, I2C automatically sends a NACK after the next byte following EOT_1. The user can generate a Stop condition in the DMA transfer complete interrupt routine if enabled.

Puya Confidential

28.3.8.1. DMA transmission

DMA mode can be enabled for transmission by setting the DMAEN bit in the I2C_CR2 register. Data will be loaded from a Memory area configured using the DMA peripheral to the I2C_DR register whenever the TxE bit is set. To map a DMA stream x for I^2C transmission (where x is the stream number), perform the following sequence:

1. Set the I2C_DR register address in the DMA_CPARx register. The data will be moved to this address from the memory after each TxE event.

2. Set the memory address in the DMA_CMARx register. The data will be loaded into I2C_DR from this memory after each TxE event.

3. Configure the total number of bytes to be transferred in the DMA_CNDTRx register. After each TxE event, this value will be decremented.

4. Configure the DMA stream priority using the PL [0:1] bits in the DMA_CCRx register

5. Set the DIR bit in the DMA_CCRx register and configure interrupts after half transfer or full transfer depending on application requirements.

6. Activate the stream by setting the EN bit in the DMA_CCRx register.

When the number of data transfers which has been programmed in the DMA controller registers is reached, the DMA controller sends an EOT/ EOT_1 signal to their 2C interface. The DMA generates an interrupt, if enabled, on the DMA stream interrupt vector.

Note:Do not enable the ITBUFEN bit in the I2C_CR2 register if DMA is used for transmission.

28.3.8.2. DMA reception

DMA mode can be enabled for reception by setting the DMAEN bit in the I2C_CR2 register. Data will be loaded from the I2C_DR register to a Memory area configured using the DMA peripheral (refer to the DMA specification) whenever a data byte is received. To map a DMA stream x for I^2C reception (where x is the stream number), perform the following sequence:

1. Set the I2C_DR register address in the DMA_CPARx register. The data will be moved from this address to the memory after each RxNE event.

2. Set the memory address in the DMA_CMARx register. The data will be loaded from the I2C_DR register to this memory area after each RxNE event.

3. Configure the total number of bytes to be transferred in the DMA_CNDTRx register. After each RxNE event, this value will be decremented.

4. Configure the DMA stream priority using the PL [0:1] bits in the DMA_CCRx register

5. Reset the DIR bit in the DMA_CCRx register and configure interrupts after half transfer or full transfer depending on application requirements.

6. Activate the stream by setting the EN bit in the DMA_CCRx register.

When the number of data transfers which has been programmed in the DMA controller registers is reached, the DMA controller sends an EOT/ EOT_1 signal to the I^2C interface. The DMA generates an interrupt, if enabled, on the DMA stream interrupt vector.

Puya Confidential

Note:Do not enable the ITBUFEN bit in the I2C_CR2 register if DMA is used for reception.

28.3.9. SMBus

The system management bus (SMBus) is a two-wire interface. Through it, various devices can communicate with each other and with the rest of the system. It is based on I²C principles of operation. SMBus provides a control bus for system and power management related tasks. A system may use SMBus to pass messages to and from devices instead of toggling individual control lines.

The system management bus specification refers to three types of devices. A slave is a device that is receiving or responding to a command. A master is a device that issues commands, generates the clocks, and terminates the transfer. A host is a specialized master that provides the main interface to the system's CPU. A host must be a master-slave and must support the SMBus host notify protocol. Only one host is allowed in a system.

The I²C1 module of this project supports SMbus/PMbus function

28.3.9.1. Similarities between SMBus and I²C

1) 2-wire bus protocol (1 Clk, 1 Data) + SMBus Alert line optional
2) Master-slave communication, Master provides clock
3) Multimaster capability
4) SMBus data format similar to I²C 7-bit addressing format

28.3.9.2. Similarities between SMBus and I²C

The following table describes the differences between SMBus and I²C.

Table 28-1 Comparison of SMBus to I²C

SMBus	I²C
Maximum speed 100kHz	Maximum speed 400kHz
Minimum clock speed 10kHz	No minimum clock speed
35 ms clock low timeout	No timeout
Logic levels are fixed	Logic levels are V_{DD} dependent
Different address types (reserved, dynamic etc.)	7-bit, 10-bit and general call slave address types
Different bus protocols (quick command, process call etc.)	No bus protocols

28.3.9.3. SMBus application usage

With system management bus, a device can provide manufacturer information, tell the system what its model/part number is, save its state for a suspend event, report different types of errors, accept control parameters, and return its status. SMBus provides a control bus for system and power management related tasks.

28.3.9.4. Address resolution protocol (ARP)

SMBus slave address conflicts can be resolved by dynamically assigning a new unique address to each slave device.

The address resolution protocol (ARP) has the following attributes:

Puya Confidential

- Address assignment uses the standard SMBus physical layer arbitration mechanism
- Assigned addresses remain constant while device power is applied; address retention through device power loss is also allowed.
- No additional SMBus packet overhead is incurred after address assignment. (i.e. subsequent accesses to assigned slave addresses have the same overhead as accesses to fixed address devices.)
- Any SMBus master can enumerate the bus.

28.3.9.5. SMBus alert mode

SMBus alert is an optional signal with an interrupt line for devices that want to trade their ability to master for a pin. SMBALERT is a wired-and signal just as the SCL and SDA signals are. SMBALERT is used in conjunction with the SMBus general call address.

A slave-only device can signal the host through SMBALERT that it wants to talk by setting ALERT bit in I2C_CR1 register The host processes the interrupt and simultaneously accesses all SMBALERT devices through the alert response address (known as ARA having a value 0001100X). Only the device(s) which pulled SMBALERT low will acknowledge the alert response address. This status is identified using SMBALERT Status flag in I2C_SR1 register The host performs a modified receive byte operation. The 7-bit device address provided by the slave transmit device is placed in the 7 most significant bits of the byte. The eighth bit can be a zero or one.

If more than one device pulls SMBA low, the highest priority (lowest address) device will win communication rights via standard arbitration during the slave address transfer. After acknowledging the slave address the device must disengage its SMBALERT pull-down. If the host still sees SMBALERT low when the message transfer is complete, it knows to read the ARA again. A host which does not implement the SMBALERT signal may periodically access the ARA. For more details on SMBus alert mode, refer to SMBus specification version 2.0.

28.3.9.6. Bus protocols

The SMBus specification supports up to nine bus protocols. For more details of these protocols and SMBus address types, refer to SMBus specification version. 2.0. These protocols should be implemented by the user software.

28.3.9.7. Timeout error

There are differences in the timing specifications between I2C and SMBus.

SMBus defines a clock low timeout, TIMEOUT of 35 Ms. SMBus specifies TLOW:SEXT as the cumulative clock low extend time for a slave device. SMBus specifies TLOW:MEXT as the cumulative clock low extend time for a master device. For more details on these timeouts, refer to SMBus specification version 2.0.

The status flag Timeout in I2C_SR1 shows the status of this feature.

Puya Confidential

28.3.9.8. PMBus

PMbus is based on SMbus, and the transmission logic is exactly the same as SMbus. The difference is that PMbus defines some functions related to power management (done by software).

28.4. I²C interrupts

Table 28-2 I²C interrupt requests

Interrupt event	Event flag	Enable control bit
Start bit sent (Master)	SB	ITEVTEN
Address sent (Master) or Address matched (Slave)	ADDR	
10-bit header sent	ADDR10	
Stop received (Slave)	STOPF	
Data byte transfer finished	BTF	
Receive buffer not empty	RxNE	ITEVTEN and ITBUFEN
Transmit buffer empty	TxE	
Bus error (BERR)	BERR	ITERREN
Arbitration loss (Master)	ARLO	
Acknowledge failure	AF	
Overrun/Underrun	OVR	
PEC error	PECERR	
Timeout/Tlow error	TIMEOUT	
SMBus Alert	SMBALERT	

28.5. I²C registers

The peripheral registers have to be accessed by half-words (16 bits) or words (32 bits).

28.5.1. I²C control register 1 (I2C_CR1)

Address offset:0x00

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Res	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	SMB-TYPE	Res	SMBUS	PE
RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	

Bit	Name	R/W	Reset Value	Function
15	SWRST	RW	0	Software reset

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				When set, the I²C is under reset state. Before resetting this bit, make sure the I²C lines are released and the bus is free. 0:I²C Peripheral not under reset 1:I²C Peripheral under reset state Note:This bit can be used to reinitialize the peripheral after an error or a locked state. As an example, if the BUSY bit is set and remains locked due to a glitch on the bus, the SWRST bit can be used to exit from this state.
14	Reserved	-	-	Reserved
13	ALERT	RW	0	SMBus alert 0:Releases SMBALERT pin high. Alert response address header followed by NACK. 1:Drives SMBALERT pin low. Alert response address header followed by ACK. When PE = 0, it is cleared by the hardware.
12	PEC	RW	0	Packet error checking This bit is set and cleared by software, and cleared by hardware when PEC is transferred or by a START or Stop condition or when PE=0. 0:No PEC transfer 1:PEC transfer (in Tx or Rx mode) Note:PEC calculation is corrupted by an arbitration loss.
11	POS	RW	0	ACK/PEC (for data reception). This bit is set and cleared by software and cleared by hardware when PE=0. 0:ACK bit controls the (N)ACK of the current byte being received in the shift register. The PEC bit indicates that current byte in shift register is a PEC. 1:ACK bit controls the (N)ACK of the next byte which will be received in the shift register. The PEC bit indicates that the next byte in the shift register is a PEC. Note:The POS bit is used when the procedure for reception of 2 bytes is followed. It must be configured before data reception starts. To NACK the 2nd byte, the ACK bit must be cleared just after ADDR is cleared.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				To check the 2nd byte as PEC, the PEC bit must be set during the ADDR stretch event after configuring the POS bit.
10	ACK	RW	0	Acknowledge enable This bit is set and cleared by software and cleared by hardware when PE=0. 0:No acknowledge returned 1:Acknowledge returned after a byte is received. (matched address or data)
9	STOP	RW	0	Stop generation. The bit is set and cleared by software, cleared by hardware when a Stop condition is detected, set by hardware when a timeout error is detected. In Master mode: 0:No Stop generation. 1:Stop generation after the current byte transfer or after the current Start condition is sent. In slave mode: 0:No Stop generation. 1:Release the SCL and SDA lines after the current byte transfer.
8	START	RW	0	Start generation This bit is set and cleared by software and cleared by hardware when start is sent or PE=0. In Master mode: 0:No Start generation 1:Repeated start generation In Slave mode: 0:No Start generation 1:Start generation when the bus is free (automatically switching to Master mode by hardware)
7	NOSTRETCH	RW	0	Clock stretching disable (Slave mode) This bit is used to disable clock stretching in slave mode when ADDR or BTF flag is set, until it is reset by software. 0:Clock stretching enabled 1:Clock stretching disabled
6	ENGC	RW	0	General call enable 0:General call disabled. Address 00h is NACKed. 1:General call enabled. Address 00h is ACKed.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
5	ENPEC	RW	0	PEC enable 0:PEC calculation disabled 1:PEC calculation enabled
4	ENARP	RW	0	ARP enable 0:ARP disabled; 1:ARP enabled; SMBus device default address recognized if SMB-TYPE=0; SMBus host address recognized if SMBTYPE=1.
3	SMBTYPE	RW	0	SMBus type. 0:SMBus device 1:SMBus host
2	Reserved	-	-	Reserved
1	SMBUS	RW	0	SMBus mode. 0:I2C mode 1:SMBus mode
0	PE	RW	0	I2C enable 0:Disabled 1:Enabled Note:If this bit is reset while a communication is ongoing, the peripheral is disabled at the end of the current communication, when back to IDLE state. All bit resets due to PE=0 occur at the end of the communication. In master mode, this bit must not be reset before the end of the communication.

28.5.2. I²C control register 2 (I2C_CR2)

Address offset:0x04

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	LAST	DMAEN	IT-BUFEN	ITEV-TEN	ITER-REN	Res	Res	FREQ [5:0]					
-	-	-	RW	RW	RW	RW	RW	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:13	Reserved	-	-	Reserved
12	LAST	RW	0	DMA last transfer

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Next DMA EOT is not the last transfer 1:Next DMA EOT is the last transfer Note:This bit is used in master receiver mode to permit the generation of a NACK on the last received data.
11	DMAEN	RW	0	DMA request enable. 0:DMA request disabled; 1:DMA request enabled when TxE=1 or RxNE =1.
10	ITBUFEN	RW	0	Buffer interrupt enable 0:TxE = 1 or RxNE = 1 does not generate any interrupt. 1:TxE = 1 or RxNE = 1 generates event interrupt (whatever the state of DMAEN)
9	ITEVTEN	RW	0	Event interrupt enable 0:Disabled 1:Event interrupt enabled This interrupt is generated when: 1) SB = 1 (Master) 2) ADDR = 1 (Master/Slave) 3) STOPF = 1 (Slave) 4) BTF = 1 with no TxE or RxNE event 5) TxE event to 1 if ITBUFEN = 1 6) RxNE event to 1if ITBUFEN = 1
8	ITERREN	RW	0	Error interrupt enable. 0:Error interrupt disabled; 1:Error interrupt enabled; This interrupt is generated when: — BERR = 1 — ARLO = 1 — AF = 1 — OVR = 1 — PECERR = 1 — TIMEOUT = 1 — SMBAlert = 1
7:6	Reserved	-	-	Reserved
5:0	FREQ	RW	0	I^2C clock frequency The FREQ bits must be configured with the APB clock frequency value. The FREQ field is used

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				by the peripheral to generate data setup and hold times compliant with the I²C specifications. The minimum allowed frequency is 4 MHz (Standard mode that is 100k) and 8 MHz (400k) the maximum frequency is limited by the maximum APB frequency. 000000:Disabled 000001:Disabled 000100:4 MHz ….. 011000:24 MHz ….. > 100100:Disabled.

28.5.3. I²C own address register 1 (I2C_OAR1)

Address offset:0x08

Reset value:0x0000 4000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMODE	Res					ADD [9:8]		ADD [7:1]							ADD0
RW	-					RW									

Bit	Name	R/W	Reset Value	Function
15	ADDMODE	RW	0	Addressing mode (slave mode). 0:7-bit slave address (10-bit address not acknowledged) 1:10 bit slave address (7-bit address not acknowledged)
14:10	Reserved	-	-	Reserved
9:8	ADD[9:8]	RW	0	Interface address. 7-bit addressing mode:don't care 10-bit addressing mode:bits 9:8 of address
7:1	ADD[7:1]	RW	0	Bits 7:1 of interface address
0	ADDR0	RW	0	Interface address. 7-bit addressing mode:don't care 10-bit addressing mode:bit 0 of address

28.5.4. I²C own address register 2 (I2C_OAR2)

Address offset:0x0C

Reset value:0x0000 0000

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	ADD2 [7:1]							ENDUAL
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7:1	ADD2 [7:1]	RW	0	Bits 7:1 of interface address bits 7:1 of address in dual addressing mode
0	ENDUAL	RW	0	Dual addressing mode enable 0:Only OAR1 is recognized in 7-bit addressing mode 1:Both OAR1 and OAR2 are recognized in 7-bit addressing mode

28.5.5. I²C data register (I2C_DR)

Address offset:0x10

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	DR [7:0]							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7:0	DR [7:0]	RW	0	The 8-bit data register, inside the device, actually two independent buffers share an address, which are used to store the received data (RX_DR) and place the data to be sent to the bus (TX_DR). **Transmitter mode:** Byte transmission starts automatically when a byte is written in the DR (TX_DR actually) register. A continuous transmit stream can be maintained if the next data to be transmitted is put in DR once the transmission is started (TxE=1). **Receiver mode:** Received byte is copied into DR (RX_DR actually) register (RxNE=1). A continuous transmit stream can be maintained if DR is read before the next data byte is received (RxNE=1). Notes:

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1. In slave mode, the address is not copied into DR. 2. Write collision is not managed (DR can be written if TxE=0). 3. If an ARLO event occurs on ACK pulse, the received byte is not copied into DR and so cannot be read.

28.5.6. I²C status register 1 (I2C_SR1)

Address offset:0x14

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALERT	TIMEOUT	Res	PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Res	STOPF	ADD10	BTF	ADDR	SB
RC_W0		-	RC_W0					R	R	-	R	R	R	R	R

Bit	Name	R/W	Reset Value	Function
15	SMBALERT	RC_W0	0	SMBus alert **SMBus host mode:** 0:no SMBALERT 1:SMBALERT event occurred on pin **SMBus slave mode:** 0:no SMBALERT response address header 1:SMBALERT response address header to SMBALERT LOW received Cleared by software writing 0, or by hardware when PE=0.
14	TIMEOUT	RC_W0	0	Timeout or Tlow error. 0:No timeout error; 1:SCL remained LOW for 25 ms (Timeout); Or Master cumulative clock low extend time more than 10 ms (Tlow:mext); Or Slave cumulative clock low extend time more than 25 ms (Tlow:sext). When set in slave mode:slave resets the communication and lines are released by hardware; When set in master mode:Stop condition sent by hardware; Cleared by software writing 0, or by hardware when PE=0. Note:This functionality is available only in SMBus mode.
13	Reserved	-	-	Reserved
12	PECERR	RC_W0	0	PEC Error in reception

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:no PEC error:receiver returns ACK after PEC reception (if ACK=1) 1:PEC error:receiver returns NACK after PEC reception (whatever ACK) Cleared by software writing 0, or by hardware when PE=0.
11	OVR	RC_W0	0	Overrun/Underrun 0:No overrun/underrun 1:Overrun or underrun Set by hardware in slave mode when NOSTRETCH=1 and: In reception when a new byte is received (including ACK pulse) and the DR register has not been read yet. New received byte is lost. In transmission when a new byte should be sent and the DR register has not been written yet. The same byte is sent twice. Cleared by software writing 0, or by hardware when PE=0. Note:If the DR write occurs very close to SCL rising edge, the sent data is unspecified and a hold timing error occurs.
10	AF	RC_W0	0	Acknowledge failure 0:No acknowledge failure 1:Acknowledge failure Set by hardware when no acknowledge is returned. Cleared by software writing 0, or by hardware when PE=0.
9	ARLO	RC_W0	0	Arbitration lost (master mode) 0:No arbitration lost detected 1:Arbitration lost detected Set by hardware when the interface loses the arbitration of the bus to another master. Cleared by software writing 0, or by hardware when PE=0. After an ARLO event the interface switches back automatically to Slave mode (MSL=0).
8	BERR	RC_W0	0	Bus error 0:No misplaced Start or Stop condition 1:Misplaced Start or Stop condition Set by hardware when the interface detects wrong start or end conditions during a byte transfer. Cleared by software writing 0, or by hardware when PE=0.
7	TxE	R	0	Data register empty (transmitters)

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Data register not empty 1:Data register empty Set when DR is empty in transmission. TxE is not set during address phase. Cleared by software writing to the DR register or by hardware after a start or a stop condition or when PE=0. TxE is not set if either a NACK is received, or if next byte to be transmitted is PEC (PEC=1) Note:TxE is not cleared by writing the first data being transmitted, or by writing data when BTF is set, as in both cases the data register is still empty.
6	RxNE	R	0	Data register not empty (receivers) 0:Data register empty 1:Data register not empty Set when data register is not empty in receiver mode. RxNE is not set during address phase. Cleared by software reading or writing the DR register or by hardware when PE=0. Note:RxNE is not cleared by reading data when BTF is set, as the data register is still full.
5	Reserved	-	-	Reserved
4	STOPF	R	0	Stop detection (slave mode) 0:No Stop condition detected 1:Stop condition detected Set by hardware when a Stop condition is detected on the bus by the slave after an acknowledge (if ACK=1). Cleared by software reading the I2C_SR1 register followed by a write in the I2C_CR1 register, or by hardware when PE=0.
3	ADD10	R	0	10-bit header sent (Master mode). 0:No ADD10 event occurred. 1:Master has sent first address byte. Set by hardware when the master has sent the first byte in 10-bit address mode. Cleared by software reading the I2C_SR1 register followed by a write in the I2C_DR register, or by hardware when PE=0. Note:ADDR is not set after a NACK reception.
2	BTF	R	0	Byte transfer complete flag

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Data byte transfer not done 1:Data byte transfer succeeded Set by hardware when NOSTRETCH=0 and: ✓ In reception when a new byte is received (including ACK pulse) and DR has not been read yet (RxNE=1). ✓ In transmission when a new byte should be sent and DR has not been written yet (TxE=1). Cleared by software reading I2C_SR1 followed by either a read or write in the DR register or by hardware after a start or a stop condition in transmission or when PE=0. Notes: The BTF bit is not set after a NACK reception.
1	ADDR	R	0	Address sent (master mode)/matched (slave mode) Cleared by software reading I2C_SR1 followed by either a read or write in the DR register or by hardware after a start or a stop condition in transmission or when PE=0. Address matched (slave): 0:Address mismatched or not received. 1:Received address matched. Set by hardware as soon as the received slave address matched with the OAR registers content or a general call or a SMBus device default address or SMBus host or SMBus alert is recognized. (when enabled depending on configuration). Note:In slave mode, it is recommended to perform the complete clearing sequence (READ SR1 then READ SR2) after ADDR is set. Address sent (Master): 0:No end of address transmission 1:End of address transmission For 10-bit addressing, the bit is set after the ACK of the 2nd byte. For 7-bit addressing, the bit is set after the ACK of the byte. Note:ADDR is not set after a NACK reception.
0	SB	R	0	Start bit (Master mode) 0:No Start condition 1:Start condition generated. —Set when a Start condition generated.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				—Cleared by software by reading the SR1 register followed by writing the DR register, or by hardware when PE=0

28.5.7. I²C status register 2 (I2C_SR2)

Address offset:0x18

Reset value:0x0000 0000

Note:Reading I2C_SR2 after reading I2C_SR1 clears the ADDR flag, even if the ADDR flag was set after reading I2C_SR1. Consequently, I2C_SR2 must be read only when ADDR is found set in I2C_SR1 or when the STOPF bit is cleared.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC [7:0]								DUALF	SMBHOST	SMBDEFAULT	GENCALL	Res	TRA	BUSY	MSL
R	R	R	R	R	R	R	R	R	R	R	R	-	R	R	R

Bit	Name	R/W	Reset Value	Function
15:8	PEC	R	0	Packet error checking register. This register contains the internal PEC when ENPEC=1.
7	DUALF	R	0	Dual address flag (Slave mode). 0:Received address matched with OAR1; 1:Received address matched with OAR2. Cleared by hardware after a Stop condition or repeated Start condition, or when PE=0.
6	SMBHOST	R	0	SMBus host header (Slave mode) 0:No SMBus Host address; 1:SMBus host address received when SMBTYPE=1 and ENARP=1. Cleared by hardware after a Stop condition or repeated Start condition, or when PE=0.
5	SMBDEFAULT	R	0	SMBus device default address (Slave mode). 0:No SMBus device default address; 1:SMBus device default address received when ENARP=1. Cleared by hardware after a Stop condition or repeated Start condition, or when PE=0.
4	GENCALL	R	0	General call address (Slave mode) 0:No General Call; 1:General Call Address received when ENGC=1.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Cleared by hardware after a Stop condition or repeated Start condition, or when PE=0.
3	Reserved	-	-	Reserved
2	TRA	R	0	Transmit/receive flags 0:Data bytes received 1:Data bytes transmitted This bit is set depending on the R/W bit of the address byte, at the end of total address phase. It is also cleared by hardware after detection of Stop condition (STOPF=1), repeated Start condition, loss of bus arbitration (ARLO=1), or when PE=0.
1	BUSY	R	0	Bus busy 0:No communication on the bus 1:Communication ongoing on the bus Set by hardware on detection of SDA or SCL low Cleared by hardware on detection of a Stop condition. It indicates a communication in progress on the bus. This information is still updated when the interface is disabled (PE=0).
0	MSL	R	0	Master/slave 0:Slave mode 1:Master mode - Set by hardware as soon as the interface is in Master mode (SB=1). - Cleared by hardware after detecting a Stop condition on the bus or a loss of arbitration (ARLO=1), or by hardware when PE=0.

28.5.8. I²C clock control register (I2C_CCR)

Address offset:0x1C

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Res	Res	CCR [11:0]											
RW	RW	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15	F/S	RW	0	I²C master mode selection

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Sm mode I^2C 1:Fm mode I^2C
14	DUTY	RW	0	Fm mode duty cycle 0:Fm mode t_{low}/t_{high} = 2 1:Fm mode t_{low}/t_{high} = 16/9
13:12	Reserved	-	-	Reserved
11:0	CCR [11:0]	RW	0	Clock control register in Fm/Sm mode (Master mode) Controls the SCL clock in master mode. 1. Standard mode: ■ $T_{high} = CCRxT_{pclk}$ ■ $T_{low} = CCRxT_{pclk}$ 2. Fast mode: ■ DUTY=0: $T_{high} = CCRxT_{pclk}$ $T_{low} = 2xCCRxT_{pclk}$ ■ DUTY = 1 (up to 400 kHz): $T_{high} = 9xCCRxT_{pclk}$ $T_{low} = 16xCCRxT_{pclk}$ Notes: ■ The minimum allowed value is 0x04, except in DUTY mode where the minimum allowed value is 0x01. ■ $T_{high} = t_{r\,(SCL)} + t_{w\,(SCLH)}$ ■ $T_{low} = t_{r\,(SCL)} + t_{w\,(SCLL)}$ ■ These delays have no filters ■ This register can only be configured when PE = 0;

28.5.9. I^2C TRISE register (I2C_TRISE)

Address offset:0x20

Reset value:0x00000002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TRISE [5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:6	Reserved	-	-	Reserved
5:0	TRISE	RW	0x0002	Maximum rise time in Fm/Sm mode (Master mode)

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				These bits should provide the maximum duration of the SCL feedback loop in master mode. The purpose is to keep a stable SCL frequency whatever the SCL rising edge duration. These bits must be programmed with the maximum SCL rise time given in the I^2C bus specification, incremented by 1. For instance:in Sm mode, the maximum allowed SCL rise time is 1000 ns. If, in the I2C_CR2 register, the value of FREQ [5:0] bits are equal to 0x08 and TPCLK1 = 125 ns therefore the TRISE [5:0] bits must be programmed with 09h. (1000 ns / 125 ns = 8 + 1 = 9). The filter value can also be added to TRISE [5:0]. If the result is not an integer, TRISE [5:0] must be programmed with the integer part, in order to respect the t_{HIGH} parameter. Note:TRISE[5:0] must be configured only when the I^2C is disabled (PE = 0).

Puya Confidential

29. Serial peripheral interface (SPI)

In this project, two SPI modules are designed and implemented, and the functions of both modules are exactly the same.

29.1. Introduction

The SPI interface may be configured to support the SPI protocol or to support the I^2S audio protocol. The SPI interface works in SPI mode by default, and functions can be switched from SPI mode to I^2S mode through software.

The Serial Peripheral Interface (SPI) allows the chip to communicate with external devices in half/full duplex, synchronous, serial mode. This interface may be configured in master mode and provide serial clocks (SCK) for external slave devices. The interface is also capable of operating in multimaster configuration. It can be used for a variety of purposes, including two-wire simplex synchronous transmission using one bi-directional data line, and reliable communication using CRC verification.

I^2S is also a 3-pin synchronous serial interface communication protocol. It supports four audio standards, including the Philips I^2S standard, the MSB and LSB alignment standards, and the PCM standard. It is in half-duplex communication and can work in master and slave 2 modes. When it acts as a master device, it supplies a clock signal to an external slave device through an interface.

29.2. SPI main features

29.2.1. SPI main features

- Master or slave operation
- 3-wire full-duplex simultaneous transmission
- 2-wire half-duplex synchronous transmission (with bidirectional data line)
- 2-wire simplex synchronous transmission (no bidirectional data line)
- 8-bit or 16-bit transfer frame format selection
- Support multi-master mode
- 8 Master mode baud rate prescalers (Max $f_{PCLK}/2$)
- Slave mode frequency (Max $f_{PCLK}/4$)
- Both Master and Slave modes can be managed by software or hardware NSS:dynamic change of Master/Slave operating mode
- Programmable clock polarity and phase
- Programmable data order, MSB first or LSB first
- Dedicated transmit and receive flags that can trigger interrupts
- SPI bus busy status flag
- Hardware CRC feature for reliable communication
  - In transmit mode, the CRC value can be transmitted as last byte
  - In full-duplex mode, the last received byte is automatically checked for CRC.

Puya Confidential

- Interrupt flags:Master mode fault, overrun and CRC error flags
- Two embedded Rx and Tx FIFOs with DMA capability, depth of four, and width of 16 bits (8 bits when data frame is set to 8-bit)

29.2.2. I²S features

- Half-duplex communication (only transmitter or receiver)
- Master or slave operations
- 8-bit programmable linear prescaler to reach accurate audio sample frequencies (from 8 kHz to 96 kHz)
- Data format may be 16-bit, 24-bit or 32-bit
- Packet frame is fixed to 16-bit (16-bit data frame) or 32-bit (16-bit, 24-bit, 32-bit data frame) by audio channel
- Programmable clock polarity (steady state)
- Underflow flag in slave transmit mode and Overflow flag in master/slave receive mode
- 16-bit register for transmission and reception with one data register for both channel sides
- Support I²S protocols:
  - I²S Philips standard
  - MSB-justified standard (left-justified)
  - LSB-justified standard (right-justified)
  - PCM standard (with short and long frame synchronization on 16-bit channel frame or 16-bit data frame extended to 32-bit channel frame)
- Data direction is always MSB first
- DMA capability for transmission and reception (16-bit wide)
- Master clock may be output to drive an external audio component. Ratio is fixed at 256 × f_S (where f_S is the audio sampling frequency)

Puya Confidential

29.3. SPI functional description

29.3.1. Overview

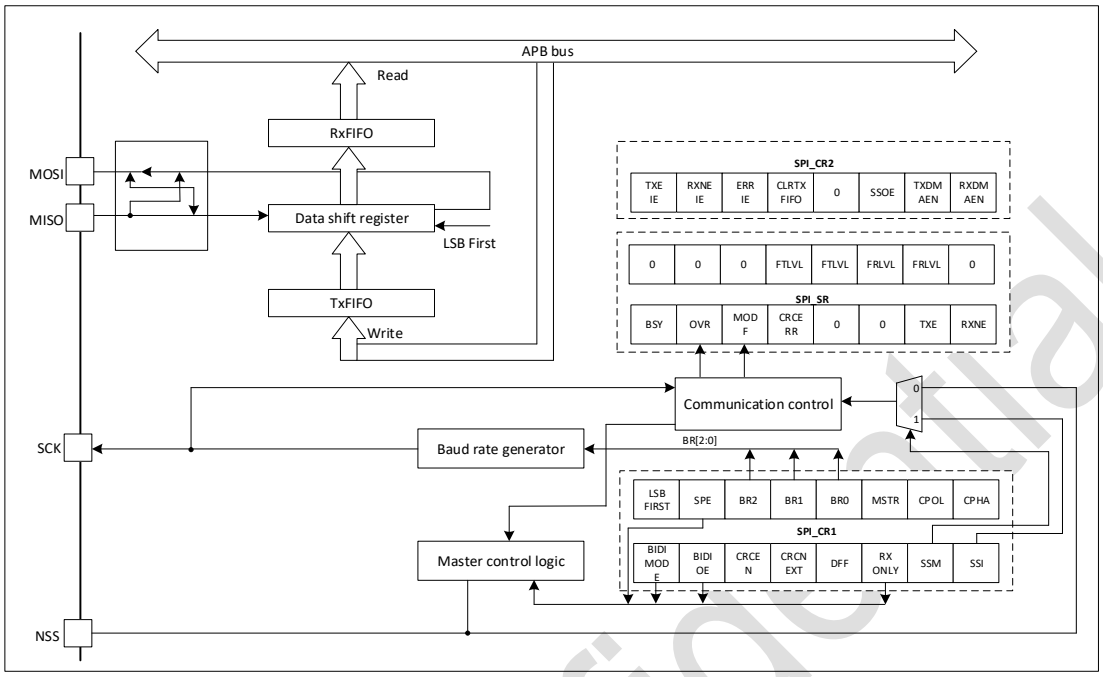


Figure29-1SPI block diagram

Four I/O pins are dedicated to SPI communication with external devices.

MISO:Master In / Slave Out data. In the general case, this pin is used to transmit data in slave mode and receive data in master mode.

MOSI:Master Out / Slave In data. In the general case, this pin is used to transmit data in master mode and receive data in slave mode.

SCK:Serial Clock output pin for SPI masters and input pin for SPI slaves.

NSS:Slave select pin. Depending on the SPI and NSS settings, this pin can be used to either:

─ Select an individual slave device for communication

─ Synchronize the data frame

─ Detect a conflict between multiple masters

The SPI bus allows the communication between one master device and one or more slave devices. The bus consists of at least two wires:one for the clock signal and the other for synchronous data transfer. Other signals can be added depending on the data exchange between SPI nodes and their slave select signal management.

29.3.2. Communications between one master and one slave

The SPI allows the MCU to communicate using different configurations, depending on the device targeted and the application requirements. These configurations use 2 or 3 wires (with software NSS management) or 4 wires (with hardware NSS management). Communication is always initiated by the host.

Puya Confidential

29.3.2.1. Full-duplex communication

By default, the SPI is configured for full-duplex communication. In this configuration, the shift registers of the master and slave are linked using two unidirectional lines between the MOSI and the MISO pins. During SPI communication, data is shifted synchronously on the SCK clock edges provided by the master. The master transmits the data to be sent to the slave via the MOSI line and receives data from the slave via the MISO line. When the data frame transfer is complete (all the bits are shifted) the information between the master and slave is exchanged.

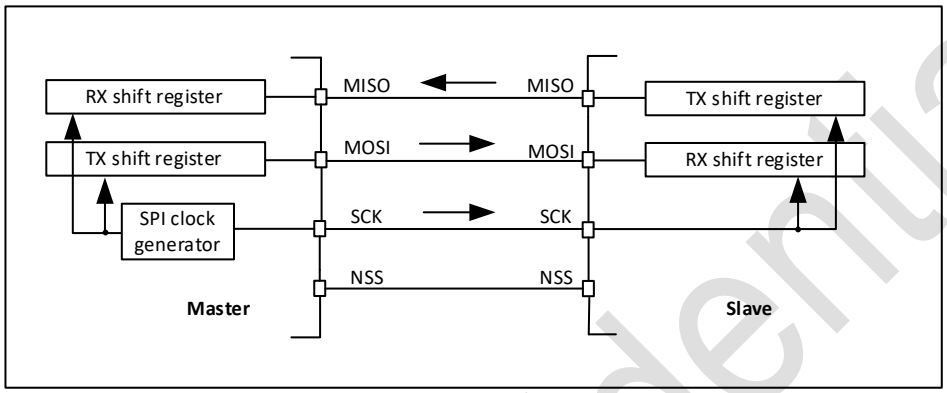


Figure 29-2 Full-duplex single master/ single slave application

29.3.2.2. Half-duplex communication

The SPI can communicate in half-duplex mode by setting the BIDIMODE bit in the SPI_CR1 register. In this configuration, one single cross connection line is used to link the shift registers of the master and slave together. During this communication, the data is synchronously shifted between the shift registers on the SCK clock edge in the transfer direction selected reciprocally by both master and slave with the BDIOE bit in their SPI_CR1 registers. In this configuration, the master's MISO pin and the slave's MOSI pin are free for other application uses and act as GPIOs.

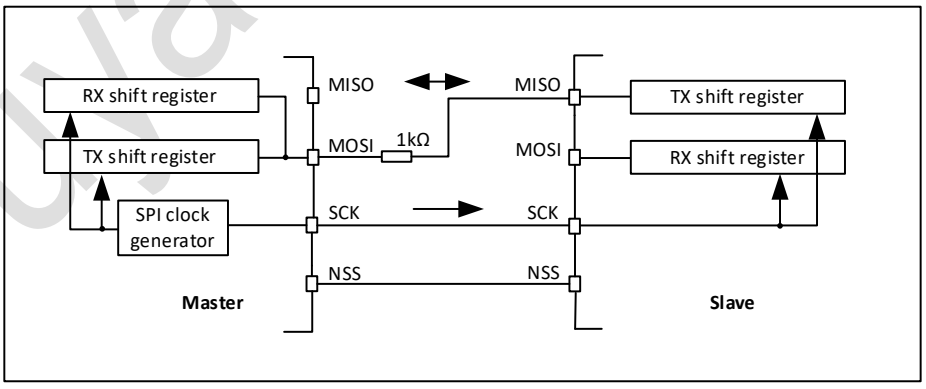


Figure 29-3 Half-duplex single master/ single slave application

The NSS pins can be used to provide a hardware control flow between master and slave. Optionally, NSS may not be used, and the process will then be handled internally.

In this configuration, the MISO pin of the master and the MOSI pin of the slave can be used as GPIO.

Puya Confidential

A critical situation can happen when communication direction is changed not synchronously between two nodes working at bidirectional mode and new transmitter accesses the common data line while former transmitter still keeps an opposite value on the line (the value depends on SPI configuration and communication data). Both nodes then fight while providing opposite output levels on the common line temporary till next node changes its direction settings correspondingly, too. It is suggested to insert a serial resistance between MISO and MOSI pins at this mode to protect the outputs and limit the current blowing between them at this situation.

29.3.2.3. Simplex communications

The SPI can communicate in simplex mode by setting the SPI in transmit-only or in receive-only using the RXONLY bit in the SPI_CR1 register. In this configuration, only one line is used for the transfer between the shift registers of the master and slave. The remaining MISO and MOSI pins pair is not used for communication and can be used as standard GPIOs.

- Transmit-only mode (RXONLY=0):The configuration settings are the same as for full-duplex. The application has to ignore the information captured on the unused input pin. This pin can be used as a standard GPIO.
- Receive-only mode (RXONLY=1):The application can disable the SPI output function by setting the RXONLY bit. In slave configuration, the MISO output is disabled and the pin can be used as a GPIO. The slave continues to receive data from the MOSI pin while its slave select signal is active. Received data events appear depending on the data buffer configuration. In the master configuration, the MOSI output is disabled and the pin can be used as a GPIO. The clock signal is generated continuously as long as the SPI is enabled. The only way to stop the clock is to clear the RXONLY bit or the SPE bit and wait until the incoming pattern from the MISO pin is finished. And then fill the data buffer based on the corresponding configuration.

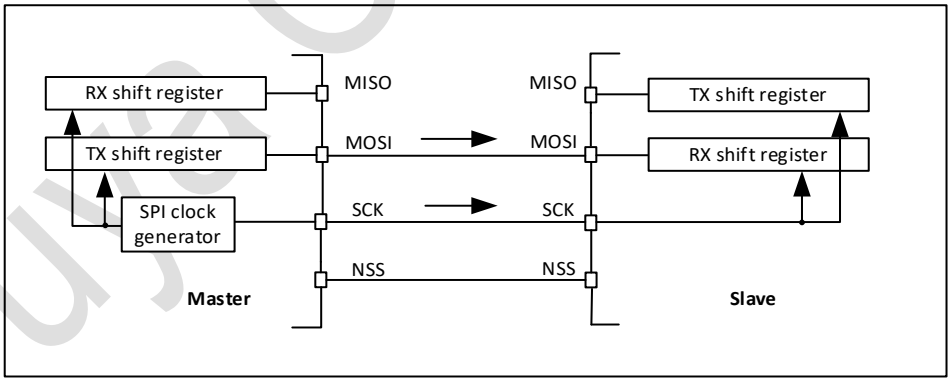


Figure 29-4 Single master/ single slave application
(master in transmit-only/slave in receive-only mode)

(1) The NSS pins can be used to provide a hardware control flow between master and slave. Optionally, NSS may not be used. Then the flow has to be handled internally.

(2) An accidental input information is captured at the input of transmitter Rx shift register. All the events associated with the transmitter receive flow must be ignored in standard transmit only mode.

(3) In this configuration, both the MISO pins can be used as GPIOs.

Puya Confidential

Any simplex communication can be alternatively replaced by a variant of the half-duplex communication with a constant setting of the transaction direction (bidirectional mode is enabled while BIDIOE bit is not changed).

29.3.3. Multi-slave communication

In a configuration with two or more independent slaves, the master uses GPIO to manage NSS for each slave. The master must select one of the slaves individually by pulling low the GPIO connected to the slave NSS input. When this is done, a standard master and dedicated slave communication is established.

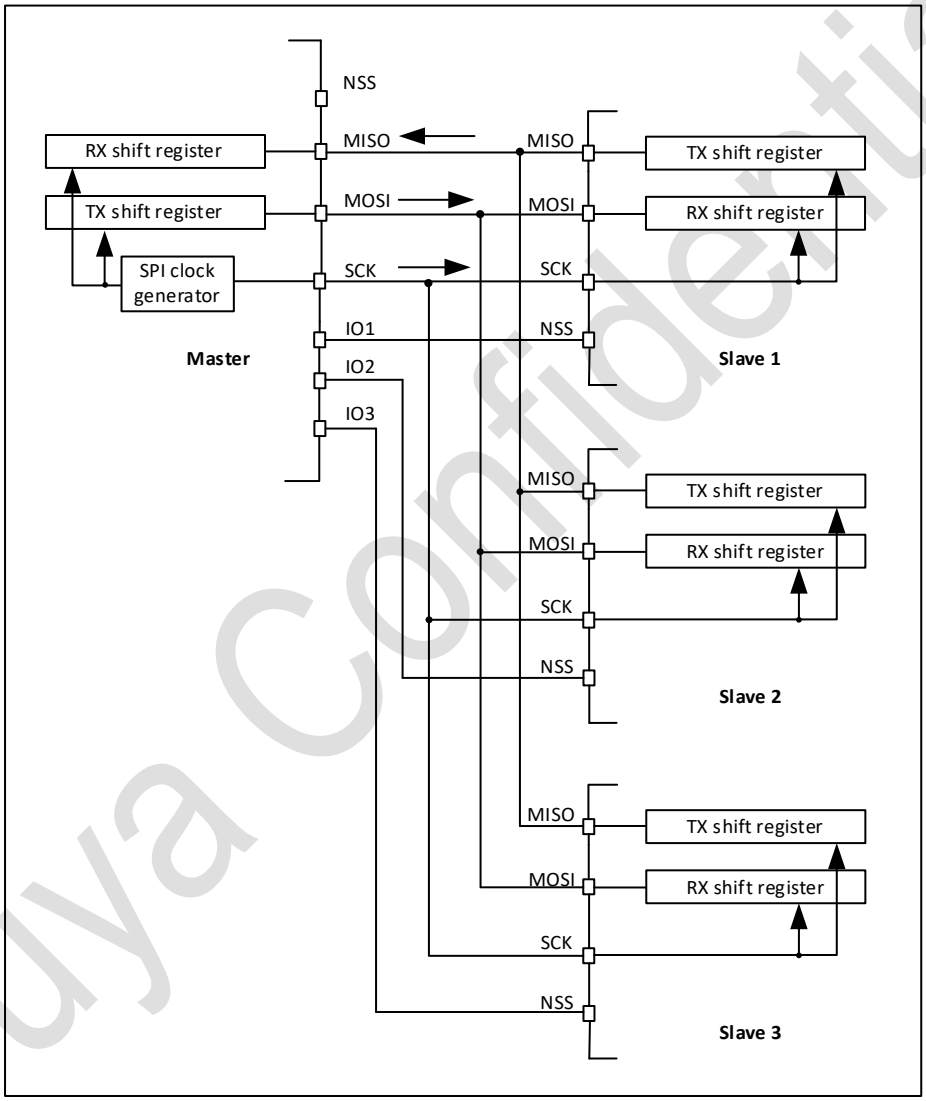


Figure 29-5 Master and three independent slaves

In this configuration, the host does not use NSS pins. It has to be managed internally (SSM=1, SSI=1) to prevent any MODF error.

As MISO pins of the slaves are connected together, all slaves must have the GPIO configuration of their MISO pin set as alternate function open-drain.

Puya Confidential

29.3.4. Multi-master communication

Unless SPI bus is not designed for a multi-master capability primarily, the user can use build in feature which detects a potential conflict between two nodes trying to master the bus at the same time. For this detection, NSS pin is used configured at hardware input mode.

The connection of more than two SPI nodes working at this mode is impossible as only one node can apply its output on a common data line at time.

When nodes are non-active, both stay at slave mode by default. Once one node wants to overtake control on the bus, it switches itself into master mode and applies active level on the slave select input of the other node via dedicated GPIO pin. After the session is completed, the active slave select signal is released and the node mastering the bus temporary returns back to passive slave mode waiting for next session start.

If potentially both nodes raised their mastering request at the same time a bus conflict event appears (see mode fault MODF event). Then the user can apply some simple arbitration process (e.g. to post-pone next attempt by predefined different time-outs applied at both nodes).

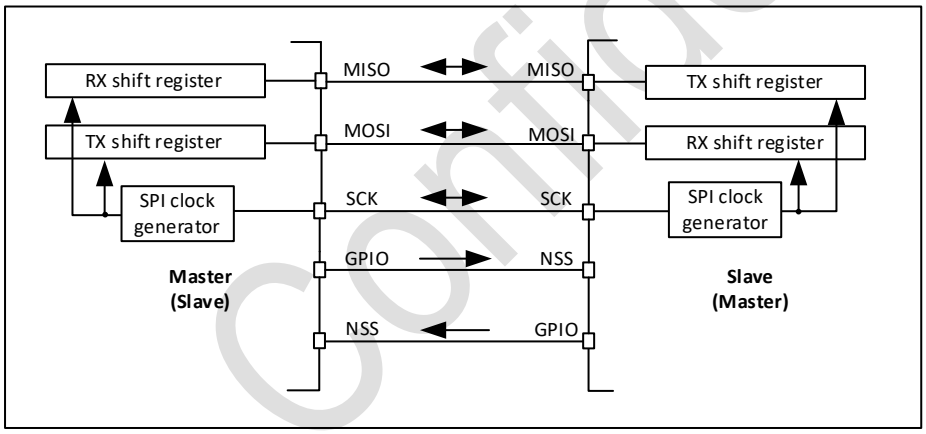


Figure 29-6 Multi-master application

The NSS is configured in a hardware input mode at both nodes. Its active level enables the MISO line output control as the passive node is configured as a slave.

29.3.5. Slave Select (NSS) pin management

In slave mode, the NSS works as a standard “chip select” input and lets the slave communicate with the master. In master mode, NSS can be used either as output or input. As an input it can prevent multimaster bus collision, and as an output it can drive a slave select signal of a single slave.

Hardware or software slave select management can be set using the SSM bit in the SPI_CR1 register:

- Software NSS management (SSM = 1):in this configuration, slave select information is driven internally by the SSI bit value in register SPI_CR1. The external NSS pin is free for other application uses.
- Hardware NSS management (SSM = 0):in this case, there are two possible configurations.

  1) NSS output enabled (SSM = 0, SSOE = 1):This configuration is used only when the device

Puya Confidential

operates in master mode. The NSS pin is managed by the hardware. The NSS signal is driven low as soon as the SPI is enabled in master mode (SPE=1), and is kept low until the SPI is disabled (SPE =0). The SPI cannot work in multimaster configuration with this NSS setting.

2) NSS output disable (SSM=0, SSOE = 0):if the microcontroller is acting as the master on the bus, this configuration allows multimaster capability. If the NSS pin is pulled low in this mode, the SPI enters master mode fault state and the device is automatically reconfigured in slave mode. In slave mode, the NSS pin works as a standard “chip select” input and the slave is selected while NSS line is at low level.

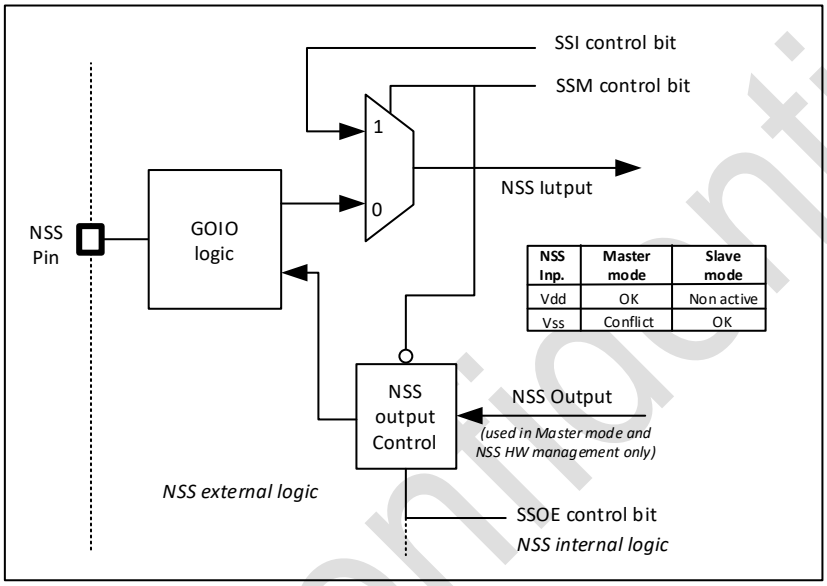


Figure 29-7 Hardware/Software slave NSS management

29.3.6. Communication formats

During SPI communication, receive and transmit operations are performed simultaneously serial clock (SCK) synchronizes the information shift and sampling operation on the data line. The communication format depends on the clock phase, the clock polarity and the data frame format. To be able to communicate together, the master and slaves devices must follow the same communication format.

29.3.6.1. Clock phase and polarity controls

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits in the SPI_CR1 register. The CPOL (clock polarity) bit controls the steady state value of the clock when no data is being transferred. This bit affects both master and slave modes. If CPOL is reset, the SCK pin has a low-level idle state. If CPOL is set, the SCK pin has a high-level idle state.

If the CPHA bit is set, the second edge on the SCK pin captures the first data bit transacted (falling edge if the CPOL bit is reset, rising edge if the CPOL bit is set). Data are latched on each occurrence of this clock transition type. If the CPHA bit is reset, the first edge on the SCK pin captures the first data bit transacted (falling edge if the CPOL bit is set, rising edge if the CPOL bit is reset). Data are latched on each occurrence of this clock transition type.

Puya Confidential

The combination of CPOL (clock polarity) and CPHA (clock phase) bits selects the data capture clock edge.

Prior to changing the CPOL/CPHA bits the SPI must be disabled by resetting the SPE bit.

The idle state of SCK must correspond to the polarity selected in the SPI_CR1 register (by pulling up SCK if CPOL=1 or pulling down SCK if CPOL=0).

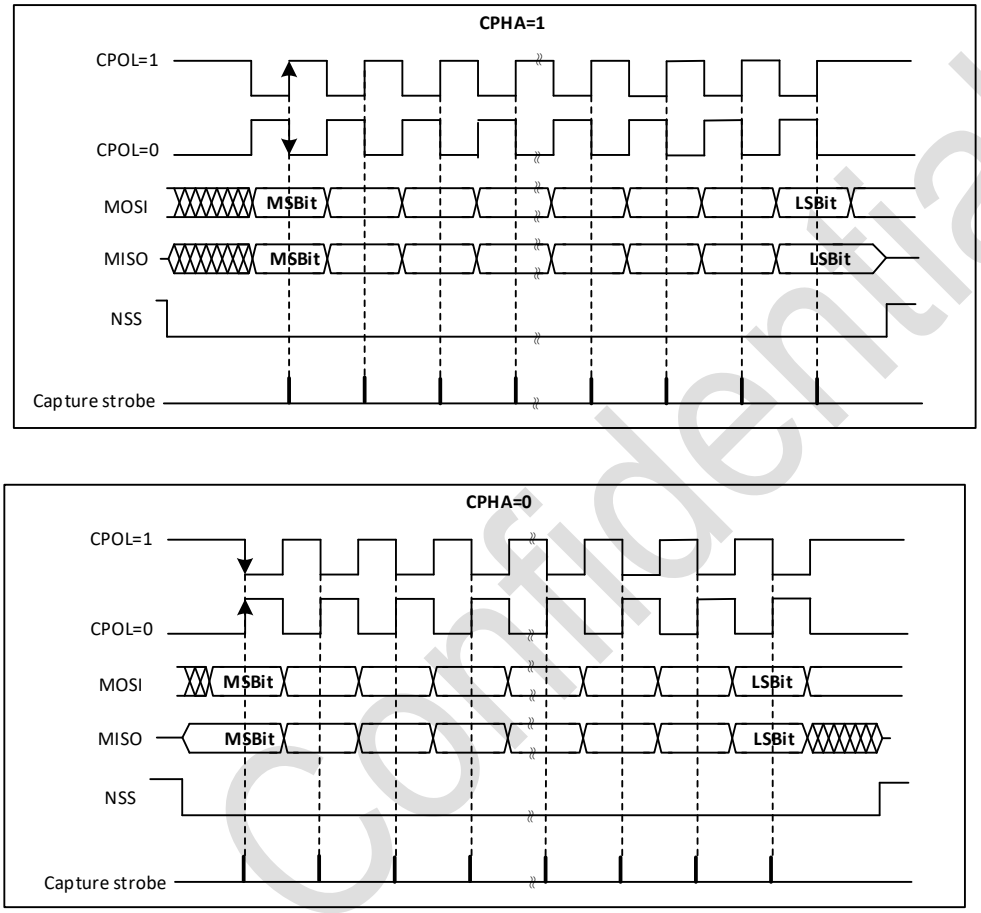


Figure 29-8 Data clock timing diagram

The order of data bits depends on LSBFIRST bit setting.

29.3.6.2. Data frame format

The SPI shift register can be set up to shift out MSB-first or LSB-first, depending on the value of the LSBFIRST bit in SPI_CR1 register. The data frame size is chosen by using the DFF bit in SPI_CR1 register. It can be set 8-bit or 16-bit length and the setting applies for both transmission and reception.

29.3.7. Configuration of SPI

The configuration procedure is almost the same for master and slave. For specific pattern establishment, follow the special chapter introduction. When a standard communication is to be initialized, perform these steps:

1. Write proper GPIO registers:Configure GPIO for MOSI, MISO and SCK pins.
2. Write to the SPI_CR1 register:
   1) Configure the serial clock baud rate using the BR [2:0] bits (not required in slave mode)
   2) Configuring CPOL and CPHA defines the relationship between data transfer and serial clock

Puya Confidential

(one of four relationships)

3) Select simplex or half-duplex mode by configuring RXONLY or BIDIMODE and BIDIOE
4) Configure LSBFIRST to define the frame format
5) Configure SSM and SSI
6) Configure the MSTR bit (in multimaster NSS configuration, avoid conflict state on NSS if master is configured to prevent MODF error).
7) Configure DFF, select the number of data frame bits

3. Write to SPI_CR2 register:
   1) Configure SSOE (not required for slave mode)
4. Write to SPI_CRCPR register:Configure the CRC polynomial if needed.
5. Write the corresponding DMA register:enable TXDMAEN or RXDMAEN, configure the SYSCFG_CFGR3.DMAx_MAP and SYSCFG_CFGR4.DMAx_MAP registers to select the corresponding DMA channel of SPI.

29.3.8. Procedure for enabling SPI

It is recommended to enable the SPI slave before the master sends the clock. If this is not done, the data transfer may not be proper. The data register of the slave must contain the data to be sent before it starts communication with the master (either at the first edge of the communication clock or, if the clock signal is continuous, before the end of the ongoing communication). The SCK signal must be settled at an idle state level corresponding to the selected polarity before the SPI slave is enabled.

The master at full-duplex (or in any transmit-only mode) starts to communicate when the SPI is enabled and TXFIFO is not empty, or with the next write to TXFIFO.

In any master receive only mode (RXONLY=1 or BIDIMODE=1 & BIDIOE=0), master starts to communicate, and the clock starts running immediately after SPI is enabled.

For DMA processing, follow the specific sections.

29.3.9. Data transmission and reception procedures

29.3.9.1. RXFIFO and TXFIFO

SPI All data communication is through FIFO, with a depth of 4 and a width of 16 bits/8 bits. This enables the SPI to work in a continuous flow, and prevents overruns when the data frame size is short. Each direction has its own FIFO called TXFIFO and RXFIFO. These FIFOs are used in all SPI modes.

The handling of FIFOs depends on the data exchange mode (duplex, simplex) and data frame format.

A read access to the SPI_DR register returns the oldest value stored in RXFIFO that has not been read yet. A write access to the SPI_DR stores the written data in the TXFIFO at the end of a send queue. Read and write access must be aligned to the data width. The FTLVL [1:0] and FRLVL [1:0] bits show the current occupancy levels of the two FIFOs.

A read access to the SPI_DR register must be managed by the RXNE event. This event is triggered when data is stored in RXFIFO. When RXNE is cleared, RXFIFO is considered to be empty.

Puya Confidential

In a similar way, write access of a data frame to be transmitted is managed by the TXE event. This event is triggered when the TXFIFO level is less than or equal to half of its capacity. Otherwise, TXE is cleared and the TXFIFO is considered as full.

In this way, RXFIFO and TXFIFO can both store 4 data frames.

Both TXE and RXNE events can be polled or handled by interrupts.

Another way to manage data exchange is to use DMA

If the next data is received when the RXFIFO is full, an overrun event occurs. An overrun event can be polled or handled by an interrupt.

The BSY bit being set indicates ongoing transaction of a current data frame. When the clock signal runs continuously, the BSY flag stays
set between data frames at master but becomes low for a minimum duration of one SPI clock at slave between each data frame transfer.

In some application scenarios, when writing data to TXFIFO, you can clear the TXFIFO data by setting the CLRTXFIFO bit, so as to write new data to TXFIFO again and communicate again.

29.3.9.2. Sequence handling

A few data frames can be passed at single sequence to complete a message. After transmission is enabled, when there is any data in the TXFIFO of the host, the sequence starts and continues as long as there is data in the TXFIFO of the host device. The clock signal is provided continuously by the master until TXFIFO becomes empty, then it stops waiting for additional data.

In receive-only modes, half-duplex (BIDIMODE=1, BIDIOE=0) or simplex (BIDIMODE=0, RXONLY=1) the master starts the sequence immediately when both SPI is enabled and receive-only mode is activated. The clock signal is provided by the master, and it does not stop until either SPI or receive-only mode is disabled by the master. The master receives data frames continuously up to this moment.

While the master can provide all the transactions in continuous mode (SCK signal is continuous) it has to respect slave capability to handle data flow and its content at any time. When necessary, the master must slow down the communication and provide either a slower clock or separate frames or data sessions with sufficient delays. Be aware there is no underflow error signal for master or slave in SPI mode, and data from the slave is always transacted and processed by the master even if the slave could not prepare it correctly in time. For slaves, a better method is to use DMA, especially when the data frame is small and the baud rate is high.

Each sequence must be encased by the NSS pulse in parallel with the multi-slave system to select just one of the slaves for communication. In a single slave system, it is not necessary to control the slave with NSS, but it is often better to provide the NSS pulse here too, to synchronize the slave with the beginning of each data sequence. NSS can be managed by both software and hardware.

When the BSY bit is set it signifies an ongoing data frame transaction. When the dedicated frame transaction is finished, the RXNE flag is raised. The last bit is just sampled and the complete data frame is stored in the RXFIFO.

Puya Confidential

29.3.9.3. Procedure for disabling the SPI

The SPI must be shut down following a specific shutdown process. It is important to do this before the system enters a low-power mode when the peripheral clock is stopped. Ongoing transactions can be corrupted in this case. In some modes the disable procedure is the only way to stop continuous communication running.

Master in full-duplex or transmit only mode can finish any transaction when it stops providing data for transmission. In this case, the clock stops after the last data transaction. Before the SPI is disabled in these modes, the user must follow standard disable procedure. When the SPI is disabled at the master transmitter while a frame transaction is ongoing or next data frame is stored in TXFIFO, the SPI behavior is not guaranteed.

When the master is in any receive only mode, the only way to stop the continuous clock is to disable the peripheral by SPE=0. This must occur in specific time window within last data frame transaction just between the sampling time of its first bit and before its last bit transfer starts (in order to receive a complete number of expected data frames and to prevent any additional “dummy” data reading after the last valid data frame). If this cannot be done, the host will always generate a clock. Specific procedure must be followed when disabling SPI in this mode.

Data received but not read remains stored in RXFIFO when the SPI is disabled, and must be processed the next time the SPI is enabled, before starting a new sequence. To prevent unread data, make sure that RXFIFO is empty when SPI is turned off (use the correct shutdown process, or by resetting with software to initialize all SPI registers).

Standard disable procedure is based on pulling BSY status together with FTLVL [1:0] to check if a transmission session is fully completed. This check can be done in specific cases, too, when it is necessary to identify the end of ongoing transactions, for example:

1. When NSS signal is managed by software and master has to provide proper end of NSS pulse for slave.
2. When transactions’ streams from DMA or FIFO are completed while the last data frame or CRC frame transaction is still ongoing in the peripheral bus.

The correct disable procedure is (except when receive only mode is used):

- Wait until FTLVL [1:0] = 00 (no more data to transmit)
- Wait until BSY=0 (the last data frame is processed)
- Disable the SPI (SPE=0).
- Read data until FRLVL [1:0] = 00 (read all the received data)

The correct disable procedure for certain receive only modes is:

- Interrupt the receive flow by disabling SPI (SPE=0) in the specific time window while the last data frame is ongoing.
- Wait until BSY=0 (the last data frame is processed).
- Read data until FRLVL [1:0] = 00 (read all the received data)

Puya Confidential

29.3.9.4. SPI communication using DMA

To operate at its maximum speed, the SPI needs to be fed with the data for transmission and the data received on the Rx buffer should be read to avoid overrun.

When TXE or RXNE is set, a DMA request is generated. The Tx and Rx buffers have independent requests.

1. When sending, each time TXE is set to 1, a DMA request is generated. The DMA then writes data to the SPI_DR register.
2. Upon reception, each time RXNE is set to 1, a DMA request is generated. The DMA then reads the data of the SPI_DR register.

When the SPI is used only for transmission data, only the SPITxDMA channel may be enabled. In this case, the OVR flag bit is set because the received data is not read away. When the SPI is used only for receiving data, the SPIRxDMA channel may be enabled.

At the time of transmission, when the DMA has written all the data to be transmitted (the TCIF flag bit of the DMA_ISR register is set), it is possible to ensure that the SPI communication is complete by detecting the BSY flag. This is used to avoid the final transmission being corrupted when the SPI is turned off or the stop low power mode is entered. The software must wait for FTLVL [1:0] = 00 before waiting for BSY = 0.

When starting to use DMA communication, to avoid error events in the DMA channel, you must follow the following steps:

1. Enable DMARx buffer (RXDMAEN bit of SPI_CR2) (if RxDMA is used)
2. Configure the SYSCFG_CFGR3.DMAx_MAP and SYSCFG_CFGR4.DMAx_MAP registers to select the DMA channel used by the SPI; And configure the registers of DMA (refer to the description of DMA module)
3. Enable DMATx buffer (at TXDMAEN bit of SPI_CR2 register) (if TxDMA is used)
4. Enable SPI via SPE bit

To force the communication shut down, you must follow the following steps:

- If DMA is enabled, turn off the DMA module by operating the DMA register (refer to the description of the DMA module)
- Close SPI via SPI close process
- Close the DMATx and Rx buffers (if DMATx and Rx are used) by clearing the TXDMAEN and RXDMAEN (SPI_CR2 registers)

29.3.9.5. Communication timing

This section introduces some typical timings that are valid for queries, interrupts, or DMA. For simplification, assume LSBFIRST = 0, CPOL = 0, and CPHA = 1. Full DMA operation configuration is not available.

- After activating NSS and enabling SPI, the slave starts to control MISO; The slave loses control of the MISO when the NSS is released or the SPI is turned off. Slave machines must be provided sufficient time to prepare master-specific data before the transfer begins.

Puya Confidential

On the host side, SPI peripherals control the MOSI and SCK signals (also NSS signals) only after SPI is enabled. If the SPI is turned off, the SPI peripherals are disconnected from the GPIO, and the level values on these lines depend on the GPIO setting.

- At the master, BSY stays active between frames if the communication is continuous. On the slave side, the BSY signal typically remains low for at least one clock cycle between data frames.
- The TXE signal is cleared only if TXFIFO is full.
- After the TXDMAEN bit is set, the DMA arbitration process begins. The TXE interrupt is generated just after the TXEIE is set. When the TXE signal is active, data transmission to the TxFIFO is started until the TxFIFO becomes full, or the DMA transmission is complete.
- If all the data to be sent can be loaded into the TxFIFO, the DMATxTCIF flag will be pulled up before the SPI bus transmission. This flag is always high until the SPI interaction is completed.

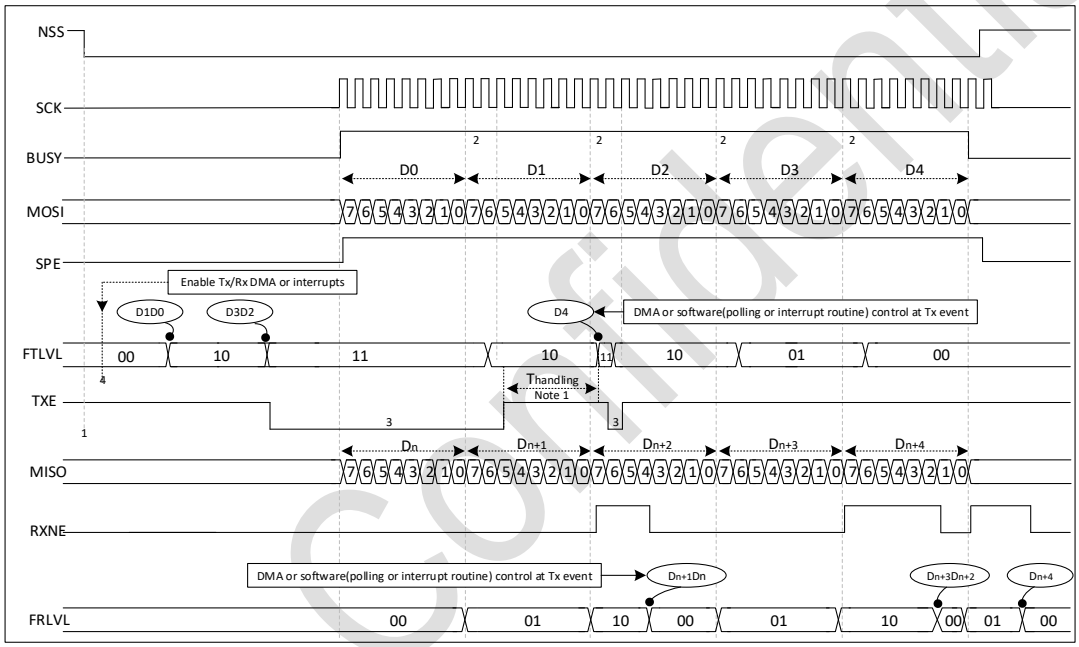


Figure 29-9 Host full-duplex communication timing (data width = 8, FRXTH = 0)

29.3.10. Status flags

Three status flags are provided for the application to completely monitor the state of the SPI bus.

29.3.10.1. Tx buffer empty flag (TXE)

The TXE flag is set when transmission TXFIFO has enough space to store data to send. TXE flag is linked to the TXFIFO level. The flag goes high and stays high until the TXFIFO level is lower or equal to 1/2 of the FIFO depth. An interrupt can be generated if the TXEIE bit in the SPI_CR2 register is set. The bit is cleared automatically when the TXFIFO level becomes greater than 1/2.

29.3.10.2. Rx buffer not empty (RXNE)

The RXNE bit is set.

- When data is received, RXNE is set.

The RXNE is cleared by hardware automatically when the above conditions are no longer true.

Puya Confidential

29.3.10.3. Busy flag (BSY)

This BSY flag is set and cleared by hardware (writing to this flag has no effect).

When it is set to '1', it indicates that the SPI is busy communicating with one exception:in the bidirectional reception mode of the main mode (MSTR = 1, BDM = 1, and BDOE = 0), the BSY flag remains low during reception.

The BSY flag is useful to detect the end of a transfer if the software wants to disable the SPI and enter Stop mode (or disable the peripheral clock). This avoids corrupting the last transfer. For this, the procedure described below must be strictly respected.

The BSY flag is also useful for preventing write collisions in a multimaster system.

Except for the bidirectional reception mode of the master mode (MSTR = 1, BDM = 1, and BDOE = 0), the BSY flag is set to '1' when the transmission starts.

The BSY flag is cleared under any one of the following conditions:

- When the SPI is correctly disabled
- When a fault is detected in Master mode (MODF bit set to 1)
- In Master mode, when it finishes a data transmission and no new data is ready to be sent
- In Slave mode, when the BSY flag is set to '0' for at least one SPI clock cycle between each data transfer.

Note:Do not use the BSY flag to handle each data transmission or reception. It is better to use the TXE and RXNE flags instead.

29.3.11. Error flags

29.3.11.1. Master mode fault (MODF)

Mode fault occurs when the master device has its internal NSS signal (NSS pin in NSS hardware mode) pulled low. Or when the SSI bit is set to '0' under software mode management of the NSS pin. This automatically sets the MODF bit.

Master mode fault affects the SPI interface in the following ways:

- The MODF bit is set and an SPI interrupt is generated if the ERRIE bit is set.
- The SPE bit is cleared. This blocks all output from the device and disables the SPI interface.
- The MSTR bit is cleared, thus forcing the device into slave mode.

Use the following software sequence to clear the MODF bit:

1. Make a read or write access to the SPI_SR register while the MODF bit is set.

2. Then write to the SPI_CR1 register.

To avoid any multiple slave conflicts in a system comprising several MCUs, the NSS pin must be pulled high during the MODF bit clearing sequence. The SPE and MSTR bits can be restored to their original state after this clearing sequence.

As a security, hardware does not allow the SPE and MSTR bits to be set while the MODF bit is set.

Puya Confidential

Normally, the MODF bit of the slave cannot be set to '1'. However, in a multi-master configuration, a device may be in slave mode with MODF bits set; At this point, the MODF bit indicates that a multi-master conflict may have occurred. The interrupt program may perform a reset or return to the default state to recover from the error state.

29.3.11.2. Overrun flag (OVR)

An overrun condition occurs when data is received by a master or slave and the RXFIFO has not enough space to store this received data. This can happen if the software or DMA did not have enough time to read the previously received data (stored in the RXFIFO).

When an overrun condition occurs, the newly received value does not overwrite the previous one in the RXFIFO. The newly received value is discarded, and all data transmitted subsequently is lost.

Clearing the OVR bit is done by a read access to the SPI_DR register followed by a read access to the SPI_SR register.

29.3.11.3. CRC error (CRCERR)

This flag is used to verify the validity of the value received when the CRCEN bit in the SPIx_CR1 register is set. The CRCERR flag in the SPIx_SR register is set if the value received in the shift register does not match the receiver SPIx_RXCRCR value. The flag is cleared by the software.

29.3.12. SPI interrupts

Table 29-1 SPI interrupt requests

Interrupt event	Event flag	Enable control bit
Transmit TXFIFO ready to be loaded	TXE	TXEIE
Data received in RXFIFO	RXNE	RXNEIE
Master Mode fault event	MODF	ERRIE
Overrun error	OVR	ERRIE
CRC protocol error	CRCERR	ERRIE

29.3.13. CRC calculation

Two separate CRC calculators are implemented in order to check the reliability of transmitted and received data. The CRC is calculated using a programmable polynomial serially on each bit. The SPI offers CRC8 or CRC16 calculation independently of the frame data length, which can be fixed to 8-bit or 16-bit.

29.3.13.1. CRC principle

CRC calculation is enabled by setting the CRCEN bit in the SPIx_CR1 register before the SPI is enabled (SPE = 1). The CRC value is calculated using an odd programmable polynomial on each bit. The calculation is processed on the sampling clock edge defined by the CPHA and CPOL bits in the SPIx_CR1 register. The calculated CRC value is checked automatically at the end of the data block as well as for transfer managed by CPU or by the DMA. When a mismatch is detected between the CRC calculated internally on the received data and the CRC sent by the transmitter, a CRCERR flag

Puya Confidential

is set to indicate a data corruption error. The right procedure for handling the CRC calculation depends on the SPI configuration and the chosen transfer management.

Note:The polynomial value should only be odd. No even values are supported.

29.3.13.2. CRC transfer managed by CPU

Communication starts and continues normally until the last data frame has to be sent or received in the SPIx_DR register. Then CRCNEXT bit has to be set in the SPIx_CR1 register to indicate that the CRC frame transaction will follow after the transaction of the currently processed data frame. The CRCNEXT bit must be set before the end of the last data frame transaction. CRC calculation is frozen during CRC transaction.

The received CRC is stored in the RXFIFO like a data byte or word. That is why in CRC mode only, the reception buffer has to be considered as a single 16-bit buffer used to receive only one data frame at a time. A CRC-format transaction usually takes one more data frame to communicate at the end of data sequence. However, when setting an 8-bit data frame checked by 16-bit CRC, two more frames are necessary to send the complete CRC.

When the last CRC data is received, an automatic check is performed comparing the received value and the value in the SPIx_RXCRC register. Software has to check the CRCERR flag in the SPIx_SR register to determine if the data transfers were corrupted or not. Software clears the CRCERR flag by writing '0' to it. After the CRC reception, the CRC value is stored in the RXFIFO and must be read in the SPIx_DR register in order to clear the RXNE flag.

29.3.13.3. CRC transfer managed by DMA

When SPI communication is enabled with CRC communication and DMA mode, the transmission and reception of the CRC at the end of communication is automatic (with the exception of reading CRC data in receive only mode). The CRCNEXT bit does not have to be handled by the software. The counter for the SPI transmission DMA channel has to be set to the number of data frames to transmit excluding the CRC frame. On the receiver side, the received CRC value is handled automatically by DMA at the end of the transaction but user must take care to flush out received CRC information from RXFIFO as it is always loaded into it. In full-duplex mode, the counter of the reception DMA channel can be set to the number of data frames to receive including the CRC.

In receive only mode, the DMA reception channel counter should contain only the amount of data transferred, excluding the CRC calculation. Then based on the complete transfer from DMA, all the CRC values must be read back by software from FIFO as it works as a single buffer in this mode. At the end of the data and CRC transfers, the CRCERR flag in the SPIx_SR register is set if corruption occurred during the transfer.

Puya Confidential

29.4. I²S functional description

29.4.1. Overview

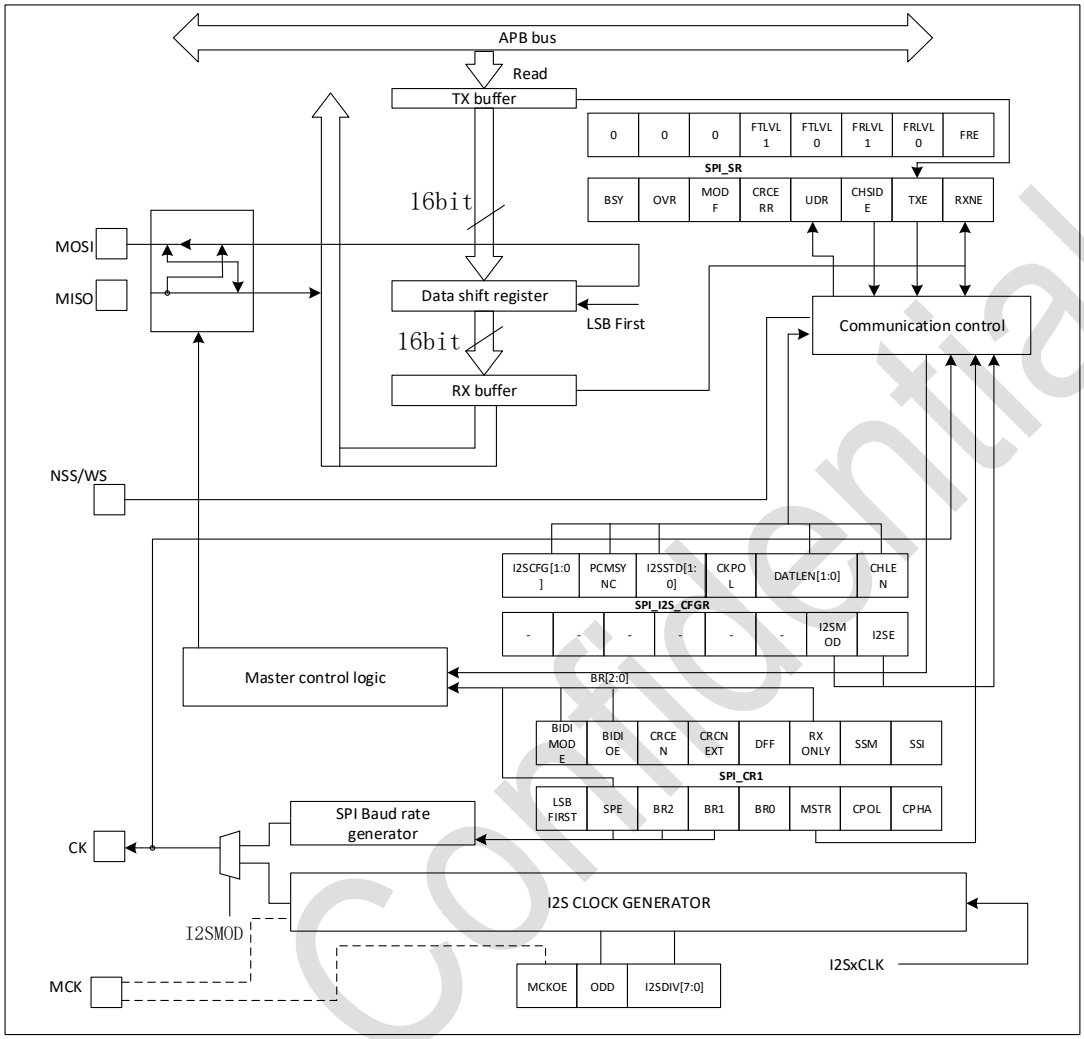


Figure 29-10 I²S block diagram

The I²S function can be enabled by setting the I2SMOD bit of the register SPI_I2SCFGR to '1'. At this time, the SPI module may be used as an I²S audio interface. The I²S interface uses roughly the same pins, flags, and interrupts as the SPI interface.

The I²S shares three common pins with the SPI:

- SD:Serial Data (mapped on the MOSI pin) to transmit or receive the two time-multiplexed data channels;
- WS:Word Select (mapped on the NSS pin) is the data control signal output in master mode and input in slave mode;
- CK:Serial Clock (mapped on the SCK pin) is the serial clock output in master mode and serial clock input in slave mode.

  An additional pin can be used when a master clock output is needed for some external audio devices:

- MCK:Master clock (mapped to MISO pin), used as an extra clock signal pin for outputting when I²S is configured in master mode and the MCKOE bit of register SPI_I2SPR is '1'. The frequency

Puya Confidential

of the output clock signal is preset to 256 × fs, where fs is the sampling frequency of the audio signal.

The I²S uses its own clock generator to produce the communication clock when it is set in master mode. This clock generator is also the source of the master clock output. Two additional registers are available in I²S mode. One is linked to the clock generator configuration SPI_I2SPR and the other one is a generic I²S configuration register SPI_I2SCFGR (audio standard, slave/master mode, data format, packet frame, clock polarity, etc.).

The SPI_CR1 register and all CRC registers are not used in the I²S mode. Likewise, the SSOE bit in the SPI_CR2 register and the MODF and CRCERR bits in the SPI_SR are not used.

The I²S uses the same SPI register for data transfer (SPI_DR) in 16-bit wide mode.

29.4.2. Supported audio protocols

The three-line bus has to handle only audio data generally time-multiplexed on two channels:the right channel and the left channel. However, there is only one 16-bit register for transmission or reception. So, it is up to the software to write into the data register the appropriate value corresponding to each channel side, or to read the data from the data register and to identify the corresponding channel by checking the CHSIDE bit in the SPIx_SR register. Channel left is always sent first followed by the channel right (CHSIDE has no meaning for the PCM protocol). Four data and packet frames are available. Data may be sent with a format of:

- 16-bit data packed in a 16-bit frame
- 16-bit data packed in a 32-bit frame
- 24-bit data packed in a 32-bit frame
- 32-bit data packed in a 32-bit frame

When using 16-bit data extended on 32-bit packet, the first 16 bits (MSB) are the significant bits, the 16-bit LSB is forced to 0 without any need for software action or DMA request (only one read/write operation). The 24-bit and 32-bit data frames need two CPU read or write operations to/from the SPI_DR register or two DMA operations if the DMA is preferred for the application. For 24-bit data frame specifically, the 8 non-significant bits are extended to 32 bits with 0-bits (by hardware). For all data formats and communication standards, the most significant bit is always sent first (MSB first).

The I²S interface supports four audio standards, configurable using the I2SSTD [1:0] and PCMSYNC bits in the SPI_I2SCFGR register.

29.4.2.1. I²S Philips standard

For this standard, the WS signal is used to indicate which channel is being transmitted. It is activated one CK clock cycle before the first bit (MSB) is available.

The 16/32-bit full-precision timing diagram is shown in the figure below. The sender changes data on the falling edge of the clock signal (CK), and the receiver reads data on the rising edge. The WS signal is also latched on the falling edge of CK.

Puya Confidential

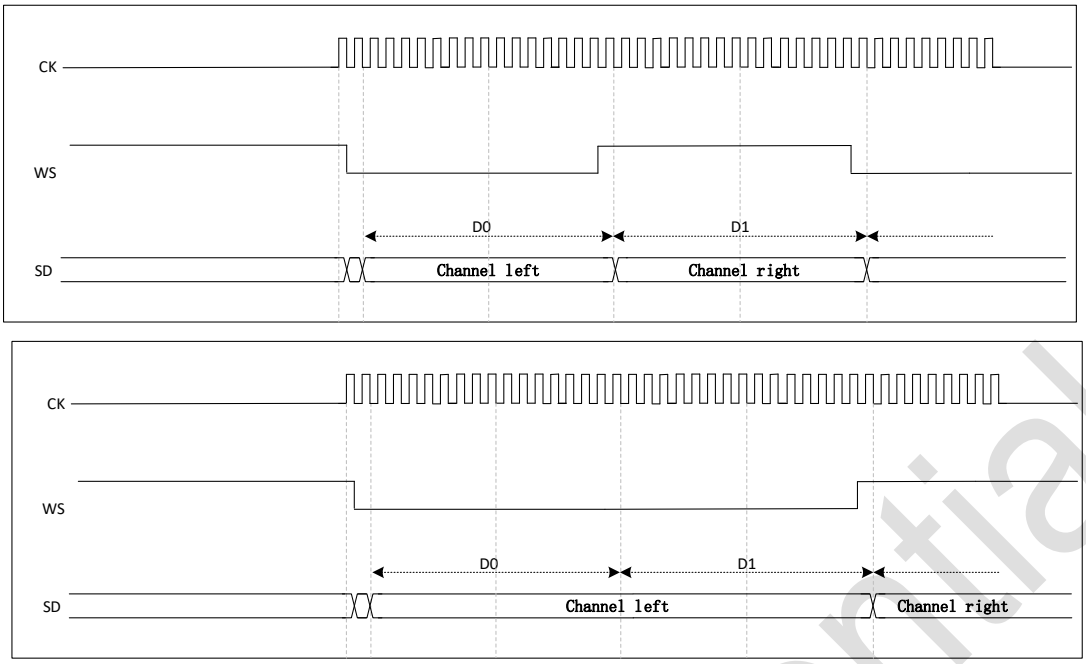


Figure 29-11 I2S Philips protocol waveform (16/32 bit full accuracy, CKPOL = 0)

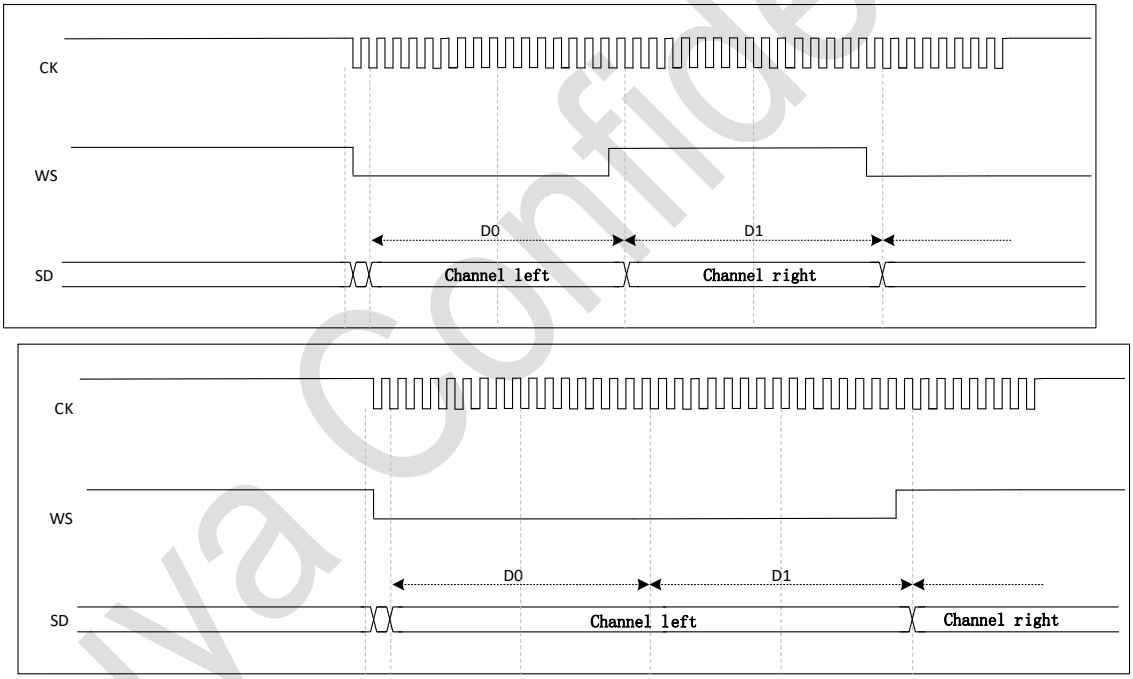


Figure 29-12 I²S Philips protocol waveform (16/32 bit full accuracy, CKPOL = 0)

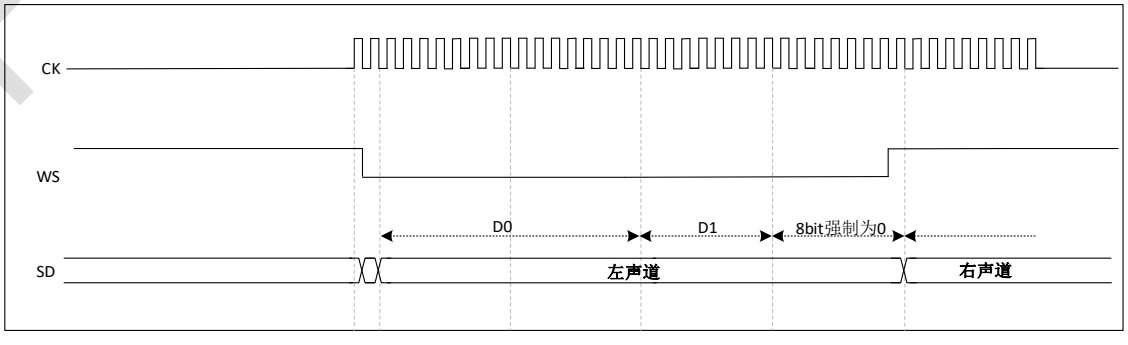


Figure 29-13 I²S Philips protocol standard waveform (24-bit frame, CKPOL = 0)

Puya Confidential

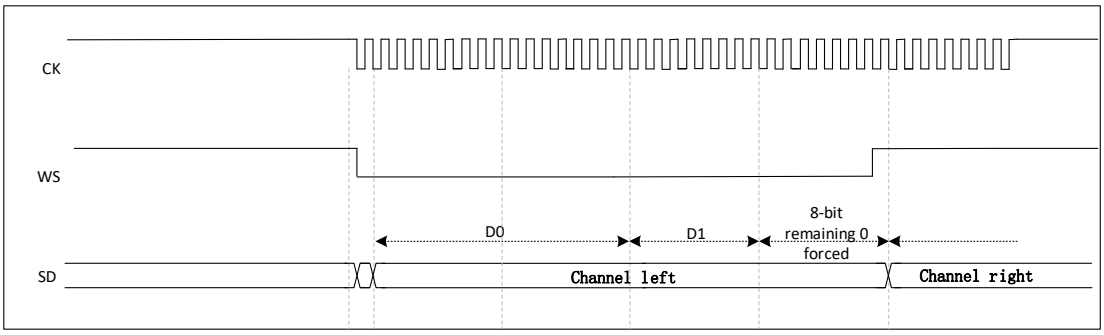


Figure 29-14 I²S Philips protocol standard waveform (24-bit frame, CKPOL = 1)

This mode needs two write or read operations to/from the SPI_DR register.

- In transmission mode:If you need to send 0x8EAA33 (24 bits), write 0x8EAA the first time, write 0x33xx the second time, the lower 8 bits are meaningless
- In reception mode:if you receive 0x8EAA33 (24 bits), receive 0x8EAA for the first time, read 0x3300 for the second time, only the upper 8 bits are meaningful data, and the lower 8 bits are always 0

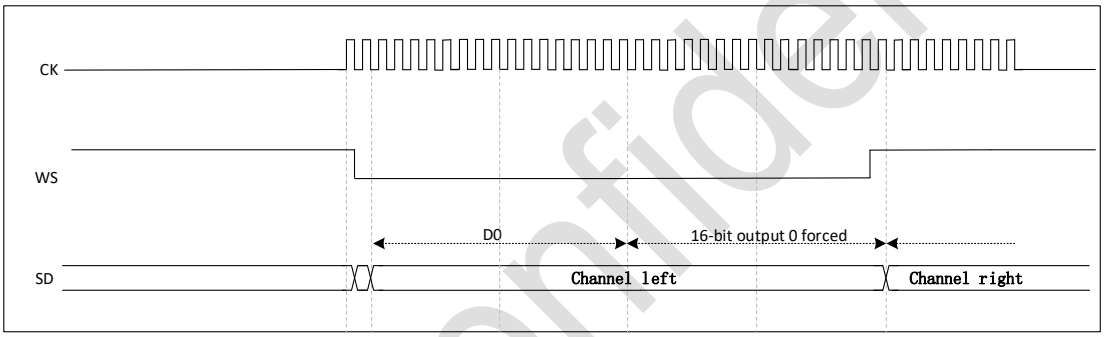


Figure 29-15 I²S Philips standard (16-bit extended to 32-bit packet frame with CPOL = 0)

When 16-bit data frame extended to 32-bit channel frame is selected during the I2S configuration phase, only one access to the SPI_DR register is required. The 16 remaining bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

If the data to transmit or the received data are 0x76A3 (0x76A30000 extended to 32-bit), you only need to operate SPI_DR once and write the data 0x76A3.

For transmission, each time an MSB is written to SPI_DR, the TXE flag is set and its interrupt, if allowed, is generated to load the SPI_DR register with the new value to send. This takes place even if 0x0000 have not yet been sent because it is done by hardware. For reception, the RXNE flag is set and its interrupt, if allowed, is generated when the first 16 MSB half-word is received.

In this way, more time is provided between two write or read operations, which prevents underrun or overrun conditions (depending on the direction of the data transfer).

29.4.2.2. MSB justified standard

For this standard, the WS signal is generated at the same time as the first data bit, which is the MSBit.

The 16/32-bit full-precision timing diagram is shown in the figure below. The sender changes data on the falling edge of the clock signal; The receiver is reading the data on the rising edge.

Puya Confidential

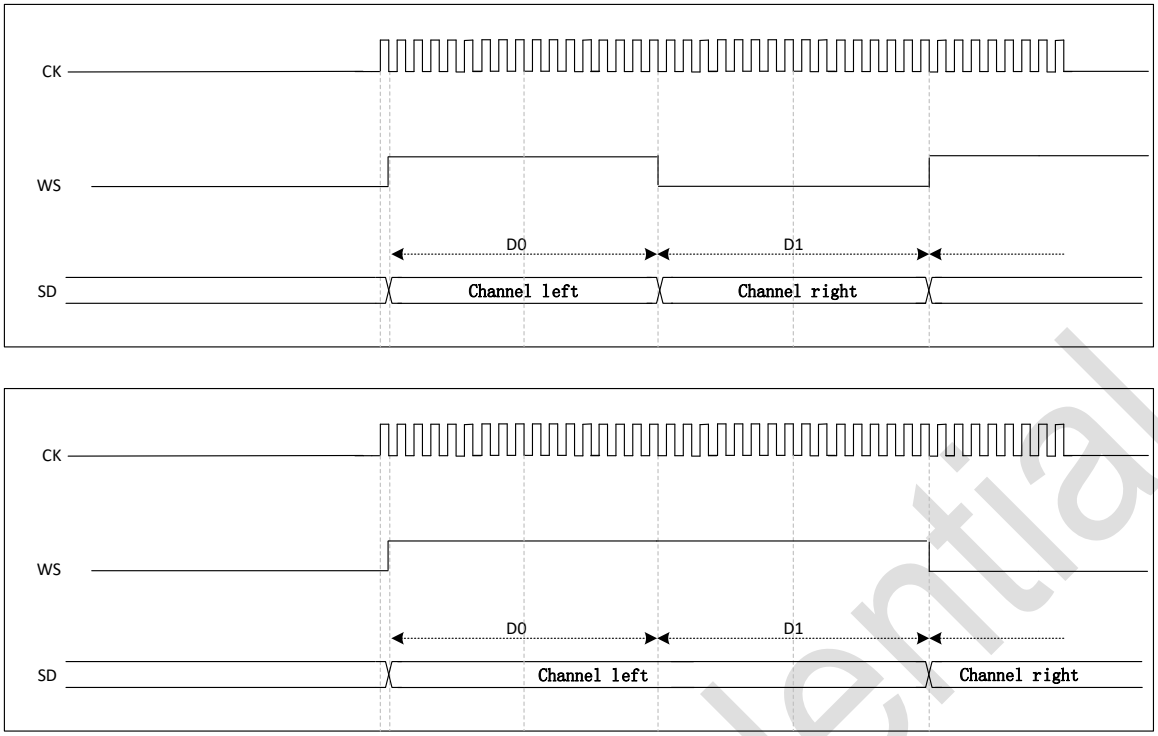


Figure 29-16 MSB justified 16-bit or 32-bit full accuracy, CPKOL = 0

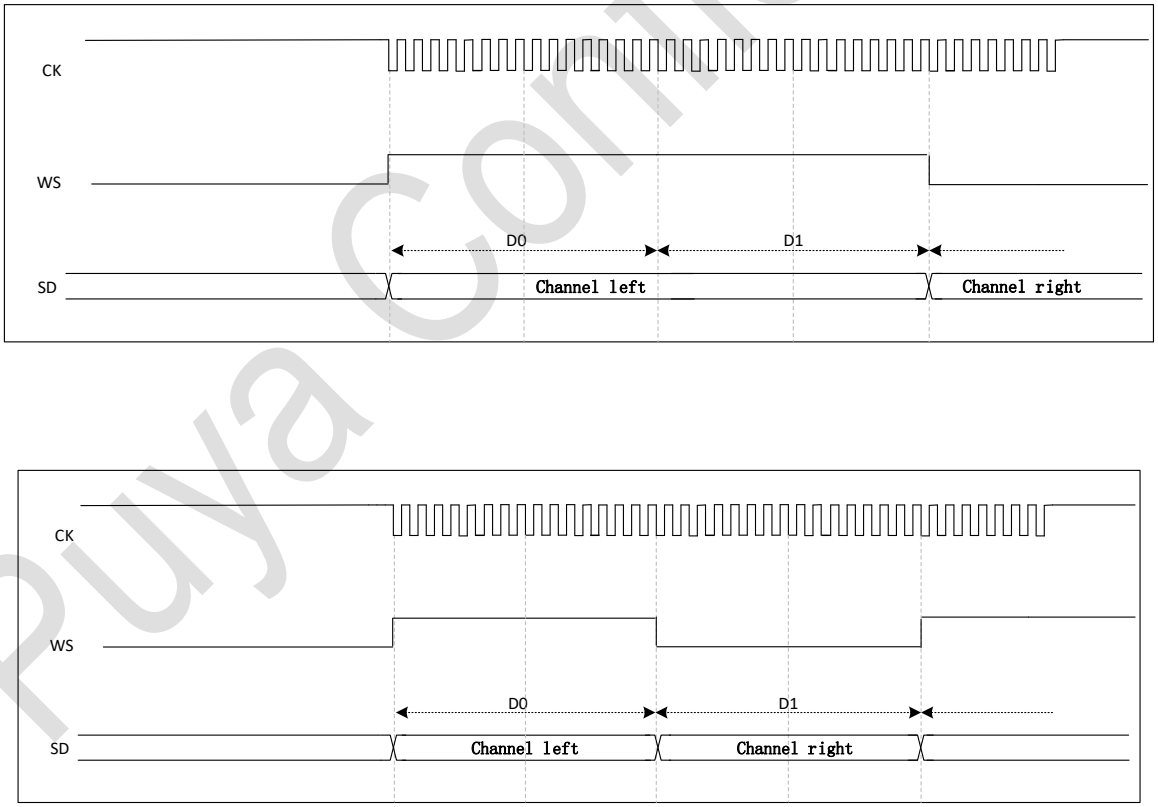


Figure 29-17 MSB justified 16-bit or 32-bit full accuracy, CPKOL = 1

The 24-bit frame timing diagram is as follows:

Puya Confidential

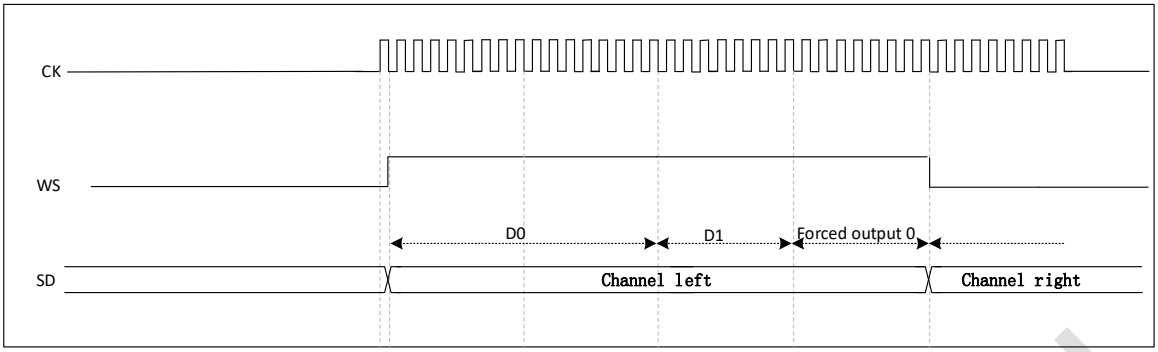


Figure 29-18 MSB justified 24-bit frame, CPKOL = 0

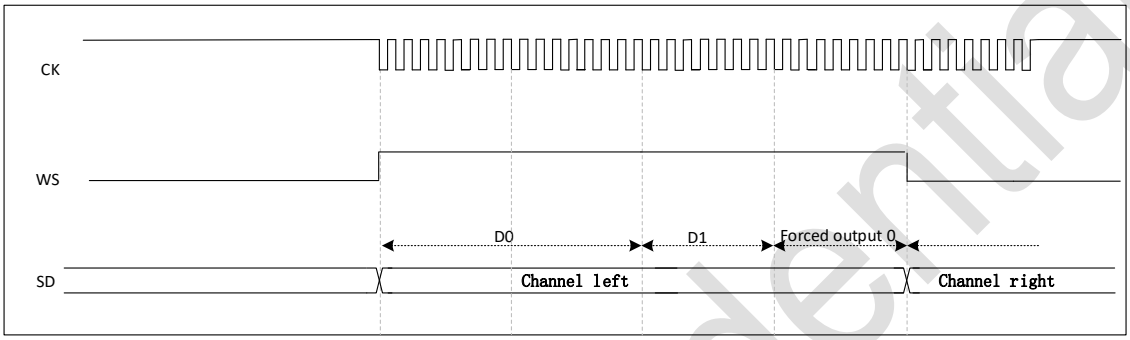


Figure 29-19 MSB justified 24-bit frame, CPKOL = 1

The following figure shows the timing of 16-bit expansion to 32-bit channel frames:

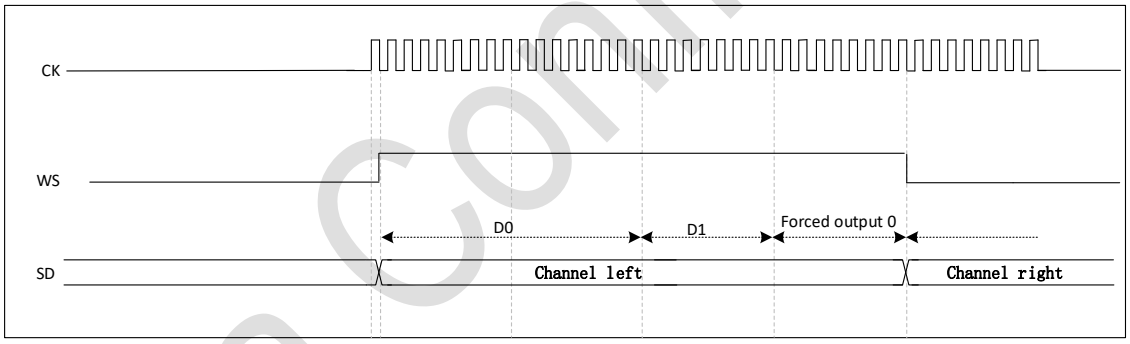


Figure 29-20 MSB justified standard, 16-bit extension to 32-bit, CPKOL = 0

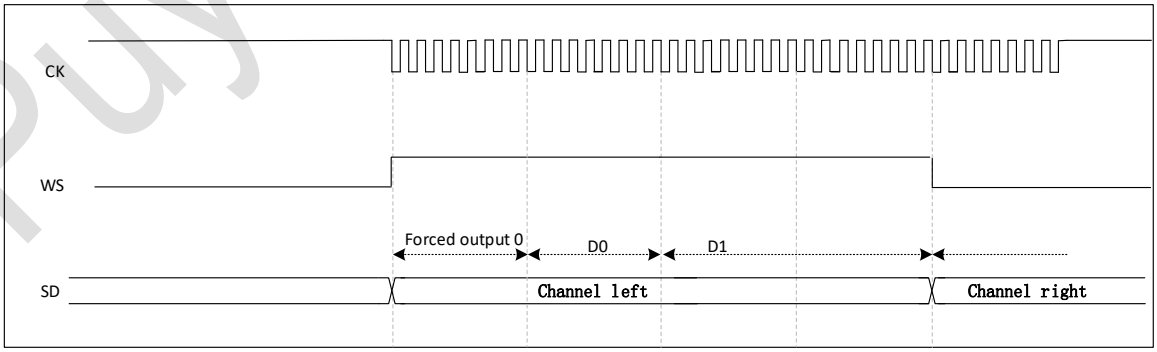


Figure 29-21 MSB justified standard, 16-bit extension to 32-bit, CPKOL = 1

The next TXE event occurs as soon as valid data begins to be sent from the SD pin. In reception mode, RXNE is asserted as soon as the significant half-word is received (and not the 0x0000 field).

Puya Confidential

29.4.2.3. LSB justified standard

This standard is similar to the MSB justified standard (no difference for the 16-bit and 32-bit full-accuracy frame formats).

The 16/32-bit full-precision timing diagram is shown in the figure below. The sender changes data on the falling edge of the clock signal; The receiver is reading the data on the rising edge.

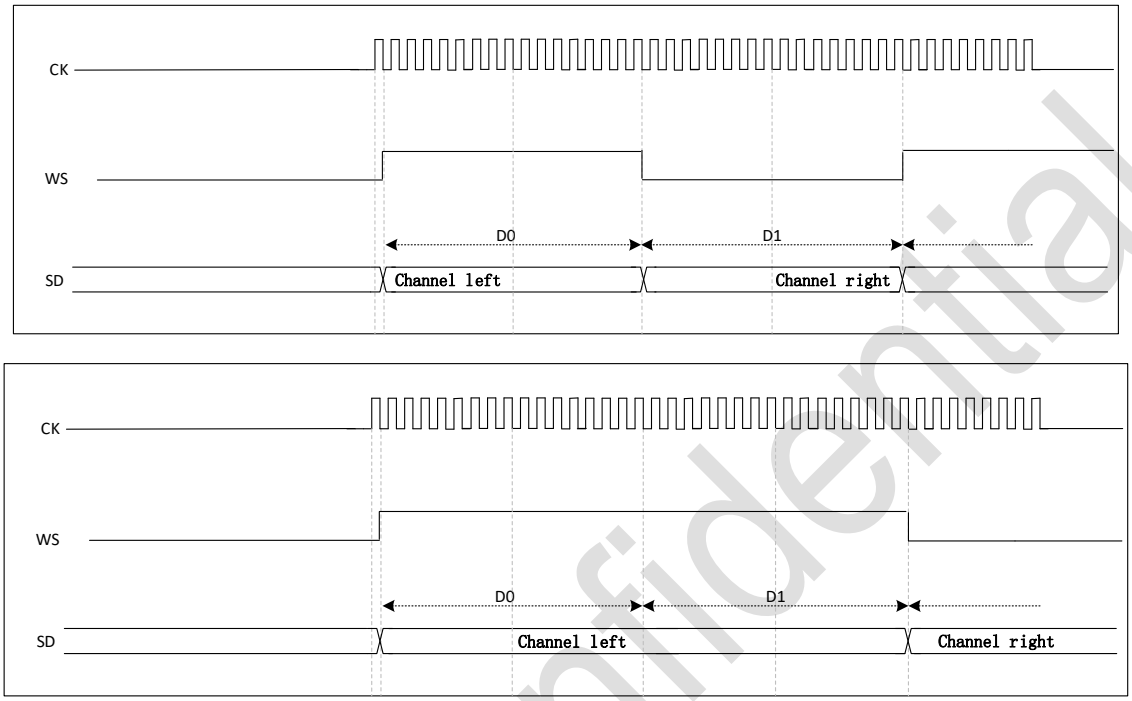


Figure 29-22 LSB justified 16-bit or 32-bit full accuracy, CKPOL = 0

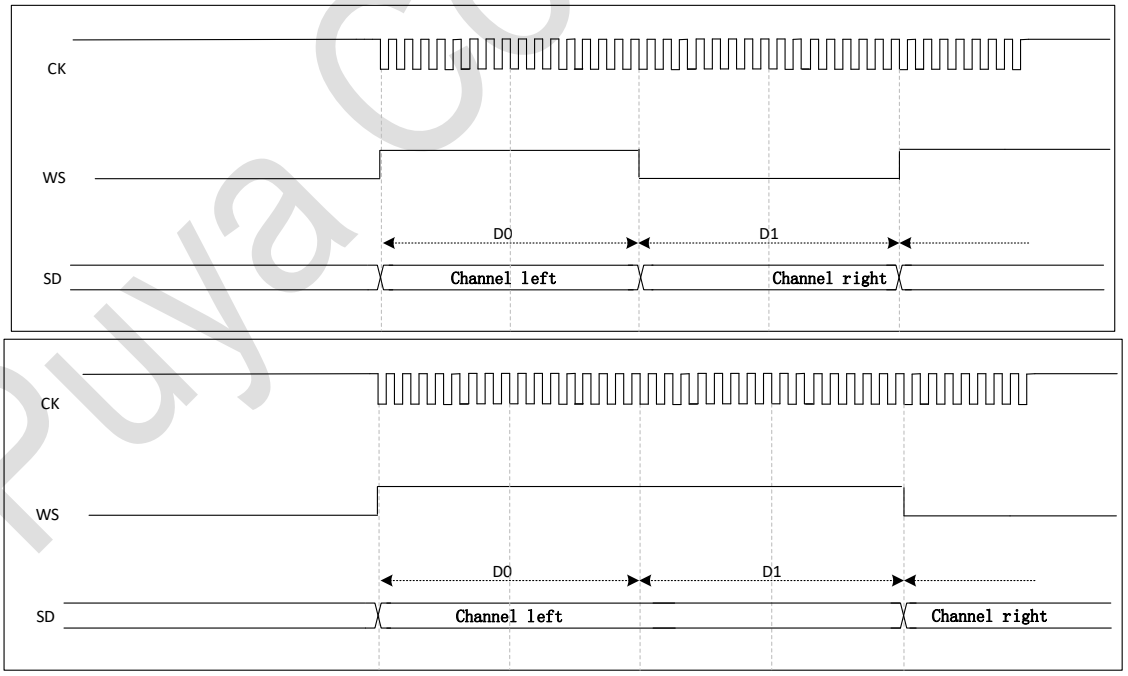


Figure 29-23 LSB justified 16-bit or 32-bit full accuracy, CKPOL = 0

The 24-bit frame timing diagram is as follows:

Puya Confidential

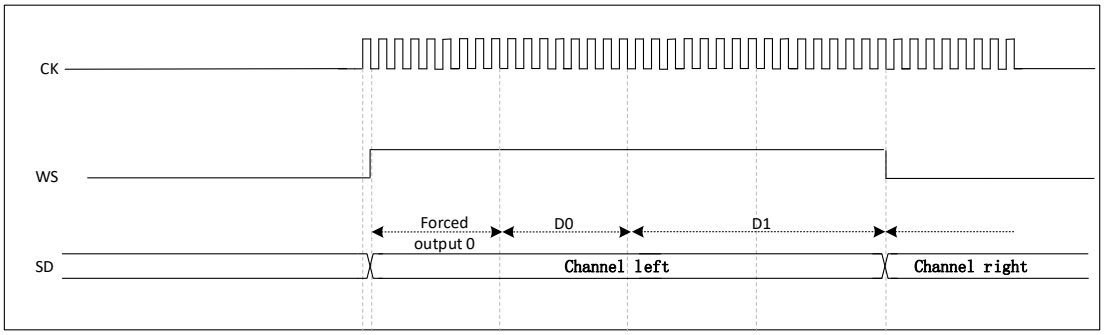


Figure 29-24 LSB justified 24-bit data, CKPOL = 0

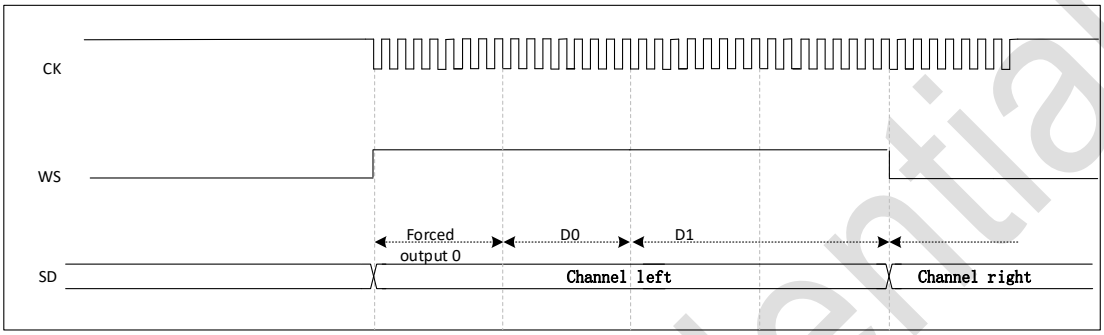


Figure 29-25 LSB justified 24-bit data, CKPOL = 1

- In transmission mode:

  If data 0x3478AE have to be transmitted, two write operations to the SPI_DR register are required by software or by DMA. Data register 0xXX34 is written for the first time and data register 0x78AE is written for the second time.

- In reception mode:

  If you want to receive data 0x3478AE, you need to read the register SPI_DR once when two consecutive RXNE events occur. The first time I read 0x0034, only the lower 8 bits make sense; Second readout 0x78AE.

The following figure shows the timing of 16-bit expansion to 32-bit channel frames:

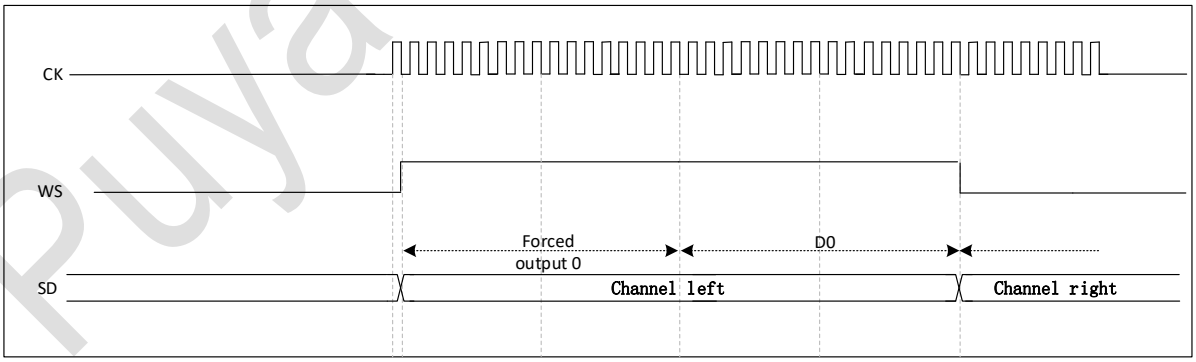


Figure 29-26 LSB justified 16-bit extended to 32-bit packet frame, CKPOL = 0

Puya Confidential

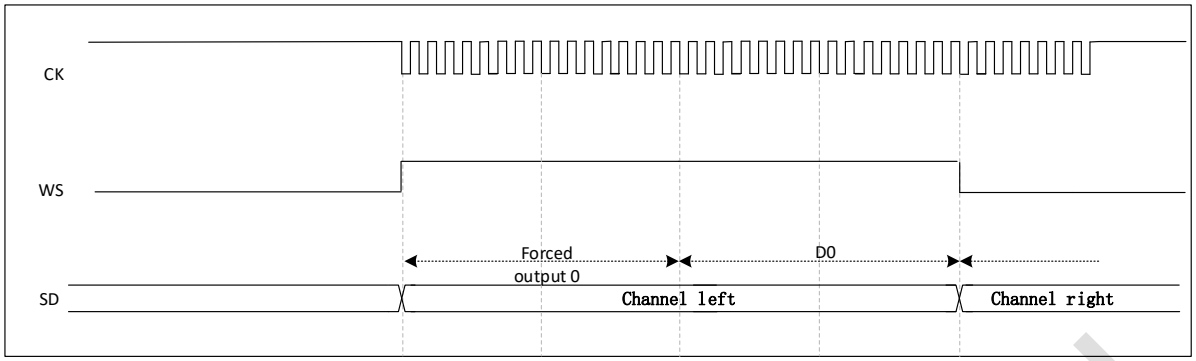


Figure 29-27 LSB justified 16-bit extended to 32-bit packet frame, CKPOL = 1

When 16-bit data frame extended to 32-bit channel frame is selected during the I2S configuration phase, only one access to the SPI_DR register is required. The 16 remaining bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

If the data to transmit or the received data are 0x76A3 (0x000076A3 extended to 32-bit), you only need to operate SPI_DR once and write the data 0x76A3.

In transmission mode, when a TXE event occurs, the application has to write the data to be transmitted (in this case 0x76A3). The 0x0000 used to expand to 32 bits is partially sent out by the hardware first. Once valid data starts to be sent out from the SD pin, the next TXE event occurs.

In reception mode, RXNE is asserted as soon as the significant half-word is received (and not the 0x0000 field).

In this way, more time is provided between two write or read operations to prevent underrun or overrun conditions.

29.4.2.4. PCM standard

For the PCM standard, there is no need to use channel-side information. The two PCM modes (short and long frame) are available and configurable using the PCMSYNC bit in SPI_I2SCFGR register.

In PCM mode, the output signals (WS, SD) are sampled on the rising edge of CK signal. The input signals (WS, SD) are captured on the falling edge of CK.

Note that CK and WS are configured as output in MASTER mode.

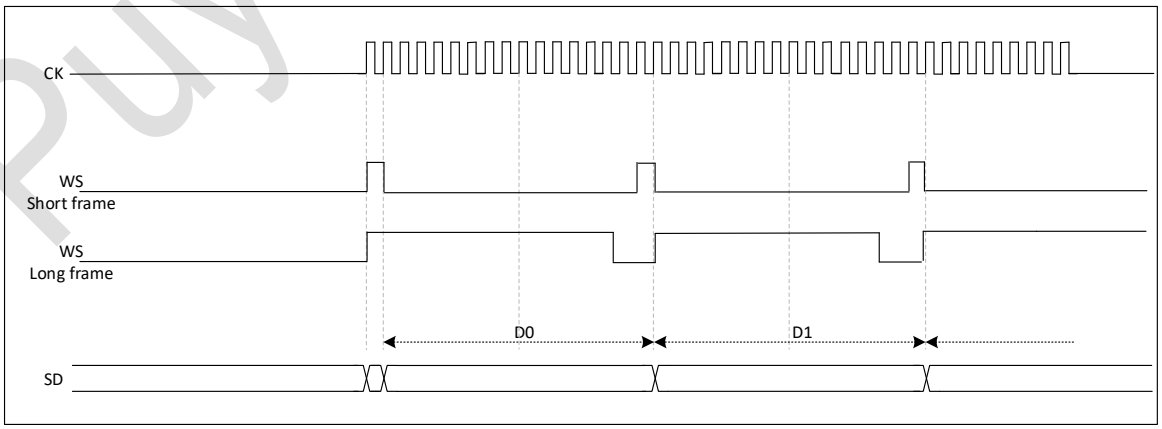


Figure 29-28 PCM standard waveform (16 bits, CKPOL = 0)

Puya Confidential

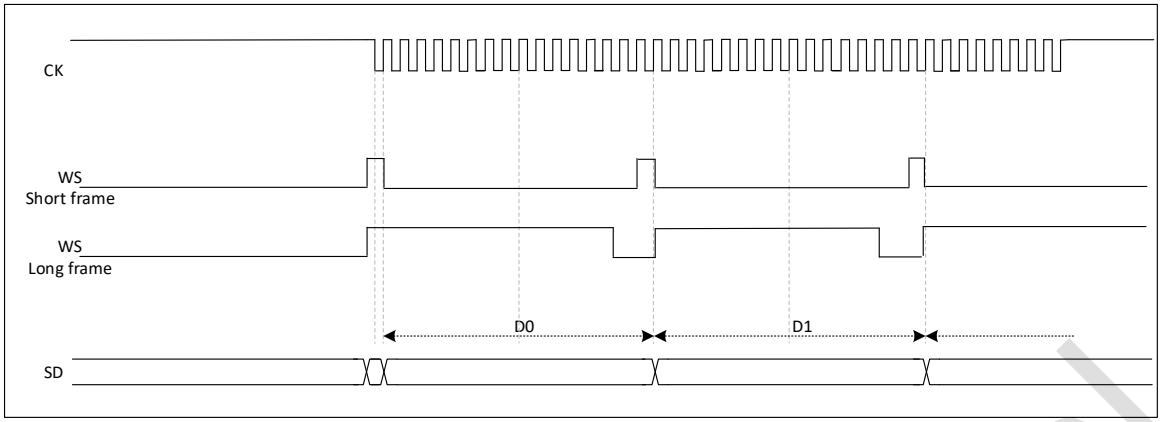


Figure 29-29 PCM standard waveform (16 bits, CKPOL = 1)

For long frame synchronization, the WS signal assertion time is fixed to 13 bits in master mode.

For short frame synchronization, the WS synchronization signal is only one cycle long.

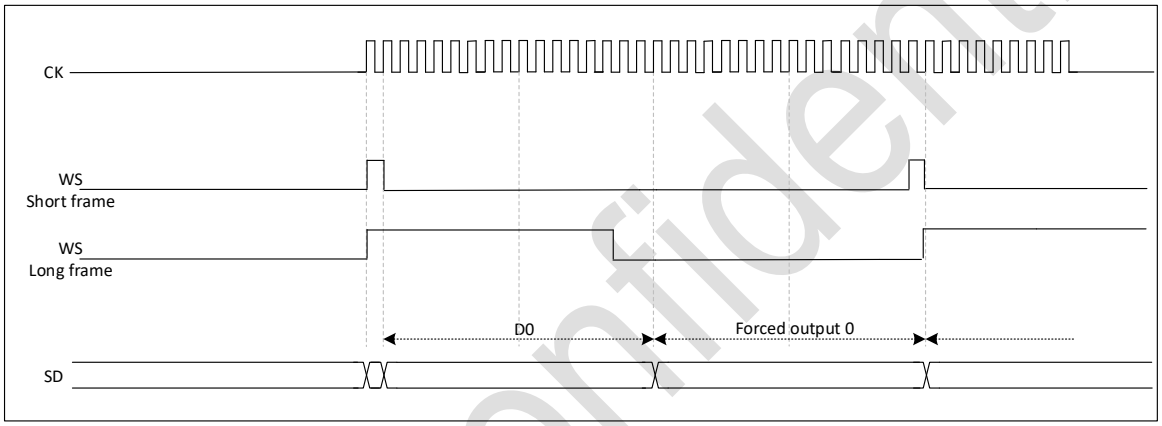


Figure 29-30 PCM standard waveform (16-bit extended to 32-bit packet frame, CKPOL = 0)

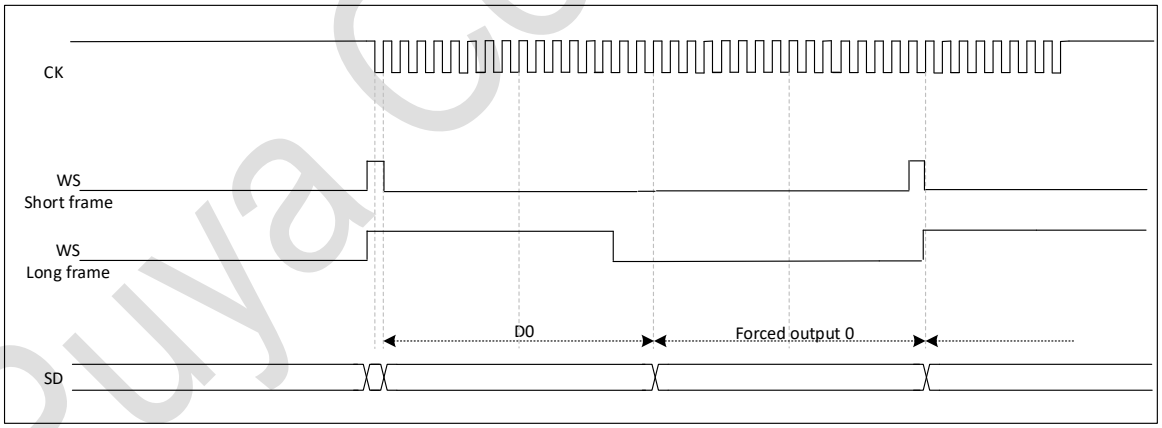


Figure 29-31 PCM standard waveform (16-bit extended to 32-bit packet frame, CKPOL = 1)

Note:For both modes (master and slave) and for both synchronizations (short and long), the number of bits between two consecutive pieces of data (and so two synchronization signals) needs to be specified (DATLEN and CHLEN bits in the SPI_I2SCFGR register) even in slave mode.

29.4.3. Clock generator

The I^2S bit rate determines the data flow on the I^2S data line and the I^2S clock signal frequency. I.e.

I^2S bit rate = number of bits per channel × number of channels × sampling audio frequency

For a 16-bit audio, left and right channel, the I^2S bit rate is calculated as follows:

Puya Confidential

I^2S bit rate = 16 × 2 × f_S

It will be:I^2S bit rate = 32 x 2 x f_S if the packet length is 32-bit wide

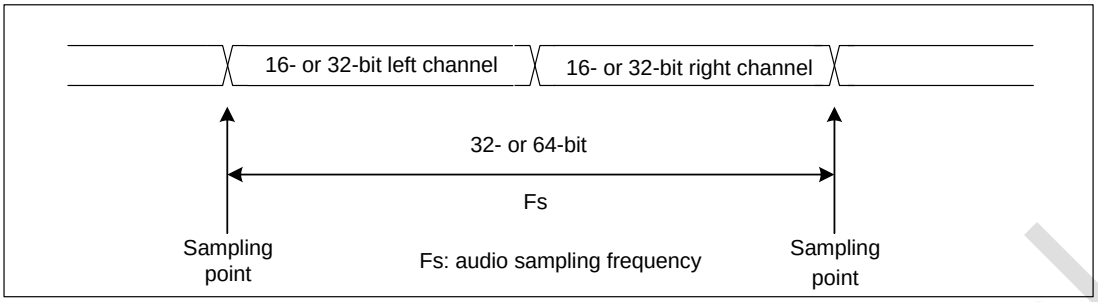


Figure 29-32 Audio sampling frequency definition

When the master mode is configured, a specific action needs to be taken to properly program the linear divider in order to communicate with the desired audio frequency.

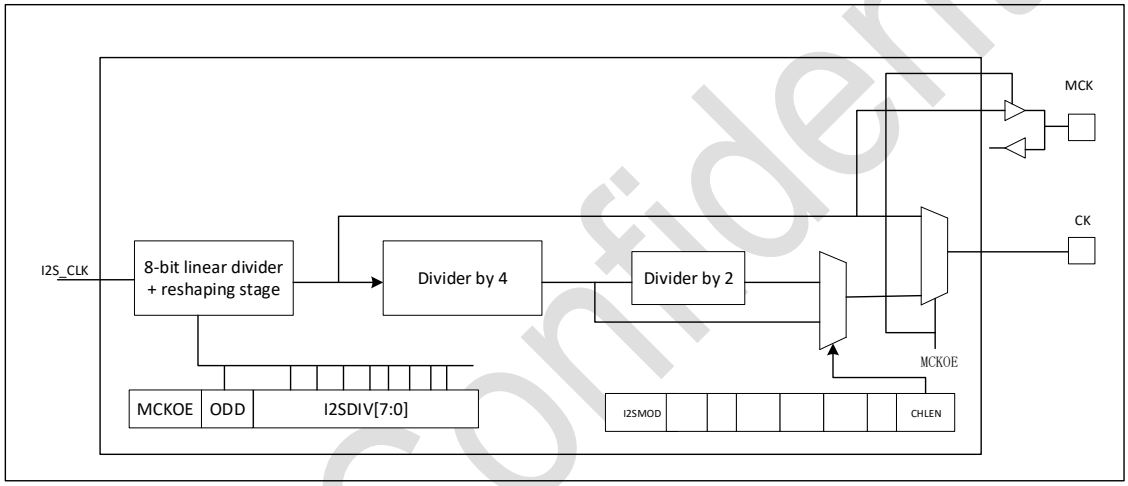


Figure 29-33 I^2S clock generator architecture

The audio sampling frequency may be 96 kHz, 48 kHz, 44.1 kHz, 32 kHz, 22.05 kHz, 16 kHz, 11.025 kHz or 8 kHz (or any other value within this range). In order to reach the desired frequency, the linear divider needs to be programmed according to the formulas below (the MCKOE bit of the register SPI_I2SPR is' 1 '):

When the frame length of the channel is 16 bits, fs = I^2SxCLK/ [(16 * 2) * ((2 * I2SDIV) + ODD) * 8]

When the frame length of the channel is 32 bits, fs = I^2SxCLK/ [(32 * 2) * ((2 * I2SDIV) + ODD) * 4]

When the main clock is turned off (MCKOE bit is' 0 '):

When the frame length of the channel is 16 bits, fs = I^2SxCLK/ [(16 * 2) * ((2 * I2SDIV) + ODD)]

When the frame length of the channel is 32 bits, fs = I^2SxCLK/ [(32 * 2) * ((2 * I2SDIV) + ODD)]

Use the standard 8MHz HSE clock to get the accurate audio frequency, see the table below:

Table 29-2 Audio frequency accuracy using 8 MHz HSE

SYSCLK (MHz)	I2S_DIV		I2S_ODD		MCLK	Target fs (Hz)	Real fs (Hz)		Error	
	16-bit wide	32-bit wide	16-bit wide	32-bit wide			16-bit wide	32-bit wide	16-bit wide	32-bit wide
72	11	6	1	0	None	96000	97826.09	93750	1.90 %	2.34 %

Puya Confidential

72	23	11	1	1	None	48000	47872.34	48913.04	0.27 %	1.90 %
72	25	13	1	0	None	44100	44117.65	43269.23	0.04 %	1.88 %
72	35	17	0	1	None	32000	32142.86	32142.86	0.45 %	0.45 %
72	51	25	0	1	None	22050	22058.82	22058.82	0.04 %	0.04 %
72	70	35	1	0	None	16000	15957.45	16071.43	0.27 %	0.45 %
72	102	51	0	0	None	11025	11029.41	11029.41	0.04 %	0.04 %
72	140	70	1	1	None	8000	8007.117	7978.723	0.09 %	0.27 %
72	2	2	0	0	Yes	96000	70312.5	70312.5	26.76 %	26.76 %
72	3	3	0	0	Yes	48000	46875	46875	2.34 %	2.34 %
72	3	3	0	0	Yes	44100	46875	46875	6.29 %	6.29 %
72	4	4	1	1	Yes	32000	31250	31250	2.34 %	2.34 %
72	6	6	1	1	Yes	22050	21634.62	21634.62	1.88 %	1.88 %
72	9	9	0	0	Yes	16000	15625	15625	2.34 %	2.34 %
72	13	13	0	0	Yes	11025	10817.31	10817.31	1.88 %	1.88 %
72	17	17	1	1	Yes	8000	8035.714	8035.714	0.45 %	0.45 %

29.4.4. I²S master mode

The I²S can be configured in master mode. This means that the serial clock is generated on the CK pin as well as the Word Select signal WS. Master clock (MCK) may be output or not, controlled by the MCKOE bit in the SPI_I2SPR register.

Procedure:

1. Select the I2SDIV [7:0] bits in the SPI_I2SPR register to define the serial clock baud rate to reach the proper audio sample frequency. The ODD bit in the SPIx_I2SPRregister also has to be defined.
2. Select the CKPOL bit to define the steady level for the communication clock. Set the MCKOE bit in the SPI_I2SPR register if the master clock MCK needs to be provided to the external ADC audio component (the I2SDIV and ODD values should be computed depending on the state of the MCK output).
3. Set the I2SMOD bit in the SPI_I2SCFGR register to activate the I²S functions and choose the I²S standard through the I2SSTD [1:0] and PCMSYNC bits, the data length through the DATLEN [1:0] bits and the number of bits per channel by configuring the CHLEN bit. Select also the I²S master mode and direction (transmitter or receiver) through the I2SCFG [1:0] bits in the SPI_I2SCFGR register.
4. If needed, select all the potential interrupt sources and the DMA capabilities by writing the SPI_CR2 register.
5. The I2SE bit in SPIx_I2SCFGR register must be set.
6. WS and CK are configured in output mode. MCK is also an output, if the MCKOE bit in SPI_I2SPR is set.

1) Transmission sequence

The transmission sequence begins when a half-word is written into the Tx buffer.

Let's assume the first data written into the Tx buffer corresponds to the left channel data. When data are transferred from the Tx buffer to the shift register, TXE is set and data corresponding to the right

Puya Confidential

channel have to be written into the Tx buffer. The CHSIDE flag indicates which channel is to be transmitted. It has a meaning when the TXE flag is set because the CHSIDE flag is updated when TXE goes high. A full frame has to be considered as a left channel data transmission followed by a right channel data transmission. It is not possible to have a partial frame where only the left channel is sent.

The data half-word is parallel loaded into the 16-bit shift register during the first bit transmission, and then shifted out, serially, to the MOSI/SD pin, MSB first. The TXE flag is set after each transfer from the Tx buffer to the shift register and an interrupt is generated if the TXEIE bit in the SPI_CR2 register is set.

The operation of writing data depends on the selected I^2S standard. To ensure a continuous audio data transmission, it is mandatory to write the SPI_DR register with the next data to transmit before the end of the current transmission.

To switch off the I^2S, by clearing I2SE, it is mandatory to wait for TXE = 1 and BSY = 0.

2) Reception sequence

The operating mode is the same as for transmission mode except for the point 3 (refer to the procedure above), where the configuration should set the master reception mode through the I2SCFG [1:0] bits.

Whatever the data or channel length, the audio data are received by 16-bit packets. This means that each time the Rx buffer is full, the RXNE flag is set and an interrupt is generated if the RXNEIE bit is set in SPI_CR2 register. Depending on the data and channel length configuration, the audio value received for a right or left channel may result from one or two receptions into the Rx buffer. Clearing the RXNE bit is performed by reading the Spider register. CHSIDE is updated after each reception. It is sensitive to the WS signal generated by the I^2S cell. The operation of reading the data depends on the selected I^2S standard.

If data are received while the previously received data have not been read yet, an overrun is generated and the OVR flag is set. If the ERRIE bit is set in the SPI_CR2 register, an interrupt is generated to indicate the error.

To switch off the I^2S, specific actions are required to ensure that the I^2S completes the transfer cycle properly without initiating a new data transfer. The sequence depends on the configuration of the data and channel lengths, and on the audio protocol mode selected.

- 16-bit data length extended on 32-bit channel length (DATLEN = 00 and CHLEN = 1) using the LSB justified mode (I2SSTD = 10)

a) Wait for the second to last RXNE = 1 (n – 1);

b) Then wait 17 I^2S clock cycles (using a software loop);

c) Disable the I^2S (I2SE = 0).

- 16-bit data length extended on 32-bit channel length (DATLEN = 00 and CHLEN = 1) in MSB justified, I2S or PCM modes (I2SSTD = 00, I2SSTD = 01 or I2SSTD = 11, respectively)

a) Wait for the last RXNE

b) Then wait 1 I^2S clock cycles (using a software loop)

c) Disable the I^2S (I2SE = 0).

Puya Confidential

- For all other combinations of DATLEN and CHLEN, whatever the audio mode selected through the I2SSTD bits, carry out the following sequence to switch off the I2S:

a) Wait for the second to last RXNE = 1 (n – 1);

b) Then wait one I^2S clock cycles (using a software loop)

c) Disable the I^2S (I2SE = 0).

Note:The BSY flag is kept low during transfers.

29.4.5. I^2S slave mode

For the slave configuration, the I^2S can be configured in transmission or reception mode. The operating mode is following mainly the same rules as described for the I^2S master configuration. In slave mode, there is no clock to be generated by the I^2S interface. The clock and WS signals are input from the external master connected to the I^2S interface. There is then no need, for the user, to configure the clock.

1. Set the I2SMOD bit in the SPI_I2SCFGR register to select I^2S mode and choose the I^2S standard through the I2SSTD [1:0] bits, the data length through the DATLEN [1:0] bits and the number of bits per channel for the frame configuring the CHLEN bit. Select also the mode (transmission or reception) for the slave through the I2SCFG [1:0] bits in SPI_I2SCFGR register

2. If needed, select all the potential interrupt sources and the DMA capabilities by writing the SPI_CR2 register.

3. The I2SE bit in SPI_I2SCFGR register must be set.

1) Transmission sequence

The transmission sequence begins when the external master device sends the clock and when the NSS_WS signal requests the transfer of data The slave has to be enabled before the external master starts the communication. The I^2S data register has to be loaded before the master initiates the communication.

For the I^2S, MSB justified and LSB justified modes, the first data item to be written into the data register corresponds to the data for the left channel. When the communication starts, the data are transferred from the Tx buffer to the shift register. The TXE flag is then set in order to request the right channel data to be written into the I^2S data register.

The CHSIDE flag indicates which channel is to be transmitted. Compared to the master transmission mode, in slave mode, CHSIDE is sensitive to the WS signal coming from the external master. This means that the slave needs to be ready to transmit the first data before the clock is generated by the master. WS assertion corresponds to left channel transmitted first.

Note: The I2SE has to be written at least two PCLK cycles before the first clock of the master comes on the CK line.

The data half-word is parallel-loaded into the 16-bit shift register (from the internal bus) during the first bit transmission, and then shifted out serially to the MOSI/SD pin MSB first. The TXE flag is set after each transfer from the Tx buffer to the shift register and an interrupt is generated if the TXEIE bit in the SPI_CR2 register is set. Note that the TXE flag should be checked to be at 1 before attempting to write the Tx buffer. The operation of writing data depends on the selected I^2S standard.

Puya Confidential

To ensure a continuous audio data transmission, it is mandatory to write the SPI_DR register with the next data to transmit before the end of the current transmission. An underrun flag is set and an interrupt may be generated if the data are not written into the SPI_DR register before the first clock edge of the next data communication. This indicates to the software that the transferred data are wrong. If the ERRIE bit is set into the SPI_CR2 register, an interrupt is generated when the UDR flag in the SPI_SR register goes high. In this case, it is mandatory to switch off the I²S and to restart a data transfer starting from the left channel.

To switch off the I²S, by clearing the I2SE bit, it is mandatory to wait for TXE = 1 and BSY = 0.

2) Reception sequence

The operating mode is the same as for the transmission mode except for the point 1. The slave receive mode needs to be selected by configuring I2SCFG [1:0].

Whatever the data length or the channel length, the audio data are received by 16-bit packets. This means that each time the RX buffer is full, the RXNE flag in the SPI_SR register is set and an interrupt is generated if the RXNEIE bit is set in the SPI_CR2 register. Depending on the data length and channel length configuration, the audio value received for a right or left channel may result from one or two receptions into the RX buffer. The CHSIDE flag is updated each time data are received to be read from the SPI_DR register. It is sensitive to the external WS line managed by the external master component. Clearing the RXNE bit is performed by reading the SPI_DR register. The operation of reading data depends on the selected I²S standard.

Three status flags are provided for the application to fully monitor the state of the I²S bus.

29.4.5.1. Busy flag (BSY)

The BSY flag is set and cleared by hardware (writing to this flag has no effect). It indicates the state of the communication layer of the I²S. When BSY is set, it indicates that the I²S is busy communicating. There is one exception in master receive mode (I2SCFG = 11) where the BSY flag is kept low during reception. The BSY flag is useful to detect the end of a transfer if the software needs to disable the I²S. This avoids corrupting the last transfer. For this, the procedure described below must be strictly respected. The BSY flag is set when a transfer starts, except when the I²S is in master receiver mode.

The BSY flag is cleared:

When a transfer completes (except in master transmit mode, in which the communication is supposed to be continuous);

When the I²S is disabled.

When communication is continuous:

In master transmit mode, the BSY flag is kept high during all the transfers;

In slave mode, the BSY flag goes low for one I²S clock cycle between each transfer

Note:Do not use the BSY flag to handle each data transmission or reception. It is better to use the TXE and RXNE flags instead.

Puya Confidential

29.4.5.2. Tx buffer empty flag (TXE)

When set, this flag indicates that the Tx buffer is empty and the next data to be transmitted can then be loaded into it. The TXE flag is reset when the Tx buffer already contains data to be transmitted. It is also reset when the I²S is disabled (I2SE bit is reset).

29.4.5.3. Rx buffer not empty (RXNE)

When set, this flag indicates that there are valid received data in the RX Buffer. It is reset when SPI_DR register is read.

29.4.5.4. Channel side flag (CHSIDE)

In transmission mode, this flag is refreshed when TXE goes high. It indicates the channel side to which the data to transfer on SD has to belong. In case of an underrun error event in slave transmission mode, this flag is not reliable and I²S needs to be switched off and switched on before resuming the communication. In reception mode, this flag is refreshed when data are received into SPI_DR. It indicates from which channel side data have been received. Note that in case of error (like OVR) this flag becomes meaningless and the I²S should be reset by disabling and then enabling it (with configuration if it needs changing).

This flag has no meaning in the PCM standard (for both Short and Long frame modes).

When the OVR or UDR flag in the SPI_SR is set and the ERRIE bit in SPI_CR2 is also set, an interrupt is generated. This interrupt can be cleared by reading the SPI_SR status register.

29.4.6. I²S error flag

There are two error flags for the I²S cell.

29.4.6.1. Underrun flag (UDR)

In slave transmission mode, this flag is set when the first clock for data transmission appears while the software has not yet loaded any value into SPI_DR. It is available when the I2SMOD bit in the SPI_I2SCFGR register is set. An interrupt may be generated if the ERRIE bit in the SPI_CR2 register is set. The UDR bit is cleared by a read operation on the SPI_SR register.

29.4.6.2. Overrun flag (OVR)

This flag is set when data are received and the previous data have not yet been read from the SPI_DR register. As a result, the incoming data are lost. An interrupt may be generated if the ERRIE bit is set in the SPI_CR2 register. In this case, the receive buffer contents are not updated with the newly received data from the transmitter device. A read operation to the SPI_DR register returns the previous correctly received data. All other subsequently transmitted half-words are lost. Clearing the OVR bit is done by a read operation on the SPI_DR register followed by a read access to the SPI_SR register.

Puya Confidential

29.4.7. I²S interrupts

I²S interrupt requests

Interrupt event	Event flag	Enable control bit
Transmit buffer empty flag	TXE	TXEIE
Receive buffer not empty flag	RXNE	RXNEIE
Overrun error	OVR	ERRIE
Underrun error	UDR	ERRIE

29.4.8. DMA functionality

Except for CRC function, same as SPI. Because there is no data transmission protection system in I²S mode.

29.5. SPI and I²S registers

The peripheral registers have to be accessed by half-words (16 bits) or words (32 bits), while the DR register supports 32-bit, 16-bit and 8-bit access.

29.5.1. SPI control register 1 (SPI_CR1) (not used in I²S mode)

Address offset:0x00

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
RW															

Bit	Name	R/W	Reset Value	Function
15	BIDIMODE	RW	0	Bidirectional data mode enable 0:2-line unidirectional data mode selected; 1:1-line bidirectional data mode selected. Note:It is not used in I²S mode.
14	BIDIOE	RW	0	Output enable in bidirectional mode This bit combined with the BIDIMODE bit selects the direction of transfer in bidirectional mode. 0:Output disabled (receive-only mode) 1:Output enabled (transmit-only mode) In master mode, the MOSI pin is used while the MISO pin is used in slave mode. Note:It is not used in I²S mode.
13	CRCEN	RW	0	Hardware CRC calculation enable 0:CRC calculation disabled; 1:CRC calculation enabled.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Note:This bit should be written only when SPI is disabled (SPE = '0') for correct operation. Note:It is not used in I²S mode.
12	CRCNEXT	RW	0	Transmit CRC next 0:Next transmit value is from Tx buffer. 1:Next transmit value is from Tx CRC register. Note:This bit has to be written as soon as the last data is written in the SPI_DR register. Note:It is not used in I²S mode.
11	DFF	RW	0	Data frame format 0:8-bit data frame format is selected for transmission/reception 1:16-bit data frame format is selected for transmission/reception Note:This bit should be written only when SPI is disabled (SPE = '0') for correct operation. Note:It is not used in I²S mode.
10	RXONLY	RW	0	Receive only This bit combined with the BIDIMODE bit selects the direction of transfer in 2-line unidirectional mode. This bit is also useful in a multislave system in which this particular slave is not accessed, the output from the accessed slave is not corrupted. 0:Full-duplex (Transmit and receive) 1:Output disabled (receive-only mode) Note:It is not used in I²S mode.
9	SSM	RW	0	Software slave management When the SSM bit is set, the NSS pin input is replaced with the value from the SSI bit. 0:Software slave management disabled 1:Software slave management enabled Note:It is not used in I²S mode.
8	SSI	RW	0	Internal slave select This bit has an effect only when the SSM bit is set. The value of this bit is forced onto the NSS pin and the IO value of the NSS pin is ignored. Note:It is not used in I²S mode.
7	LSBFIRST	RW	0	Frame format 0:MSB transmitted first

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:LSB transmitted first Note:This bit should not be changed when communication is ongoing. Note:It is not used in I²S mode.
6	SPE	RW	0	SPI enabled 0:Peripheral disabled 1:Peripheral enabled Note:It is not used in I²S mode.
5:3	BR [2:0]	RW	0	Baud rate control 000:f_{PCLK}/2 001:f_{PCLK}/4 010:f_{PCLK}/8 011:f_{PCLK}/16 100:f_{PCLK}/32 101:f_{PCLK}/64 110:f_{PCLK}/128 111:f_{PCLK}/256 Note:This bit should not be changed when communication is ongoing. Note:It is not used in I²S mode. In slave mode, the fastest baud rate only supports f_{PCLK}/4.
2	MSTR	RW	0	Master selection 0:Slave configuration 1:Master configuration Note:This bit should not be changed when communication is ongoing. Note:It is not used in I²S mode.
1	CPOL	RW	0	Clock polarity 0:In idle state, SCK remains low 1:SCK maintains high level in idle state Note:This bit should not be changed when communication is ongoing. Note:It is not used in I²S mode.
0	CPHA	RW	0	Clock phase 0:The first clock transition is the first data capture edge 1:The second clock transition is the first data capture edge Note:This bit should not be changed when communication is ongoing. Note:It is not used in I²S mode.

Puya Confidential

29.5.2. SPI control register 2 (SPI_CR2)

Address offset:0x04

Reset value:0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								TXEIE	RXNEIE	ERRIE	CLRTXFIFO	Res	SSOE	TXDMAEN	RXDMAEN
-								RW	RW	RW	RW	-	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
15:8	Reserved	-	-	Reserved
7	TXEIE	RW	0	Tx buffer empty interrupt enable 0:TXE interrupt masked 1:TXE interrupt not masked. Used to generate an interrupt request when the TXE flag is set.
6	RXNEIE	RW	0	Rx buffer not empty interrupt enable 0:RXNE interrupt masked 1:RXNE interrupt not masked. Used to generate an interrupt request when the RXNE flag is set.
5	ERRIE	RW	0	Error interrupt enable This bit controls the generation of an interrupt when an error condition occurs (CRCERR, OVR and MODF) 0:Error interrupt disabled 1:Error interrupt enabled
4	CLRTXFIFO	RW	0	Clear TXFIFO Set by software and reset by hardware 0:No action 1:Clear TXFIFO Note:This bit should be written only when SPI is disabled (SPE = ‘0’) for valid operation.
3	Res	-	-	Reserved
2	SSOE	RW	0	SS output enable 0:SS output is disabled in master mode and the SPI interface can work in multimaster configuration. 1:SS output is enabled in master mode and when the SPI interface is enabled. The SPI interface cannot work in a multimaster environment. Note:It is not used in I^2S mode.
1	TXDMAEN	RW	0	Tx buffer DMA enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				When this bit is set, a DMA request is generated whenever the TXE flag is set. 0:Tx buffer DMA disabled 1:Tx buffer DMA enabled
0	RXDMAEN	RW	0	Rx buffer DMA enable When this bit is set, a DMA request is generated whenever the RXNE flag is set. 0:Rx buffer DMA disabled 1:Rx buffer DMA enabled

29.5.3. SPI status register (SPI_SR)

Address offset:0x08

Reset value:0x00000002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			FTLVL [1:0]		FRLVL [1:0]		Res	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
-			R	R	R	R	-	R	R	R	RC_W0	R	R	R	R

Bit	Name	R/W	Reset Value	Function
15:13	Reserved	-	-	Reserved
12:11	FTLVL	R	0	FIFO transmission level Set and cleared by hardware 00:FIFO empty 01:1/4 FIFO 10:1/2 FIFO 11:FIFO full (considered as FULL when the FIFO threshold is greater than 1/2) Note:This bit is not used in I^2S mode.
10:9	FRLVL	R	0	FIFO reception level. Set and cleared by hardware 00:FIFO empty 01:1/4 FIFO 10:1/2 FIFO 11:FIFO full Note:These bits are not used in I^2S mode and in SPI receive-only mode while CRC calculation is enabled.
8	Reserved	-	-	Reserved
7	BSY	R	0	Busy flag 0:SPI not busy 1:SPI is busy in communication or Tx buffer is not empty This bit is set or reset by hardware.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
6	OVR	R	0	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set by hardware and reset by a software sequence. (Overrun and underrun sequences are different).
5	MODF	R	0	Mode fault 0:No mode fault occurred 1:Mode fault occurred This flag is set by hardware and reset by a software sequence. Note:It is not used in I^2S mode.
4	CRCERR	RC_W0	0	CRC error flag 0:CRC value received matches the SPIx_RXCRCR value 1:CRC value received does not match the SPIx_RXCRCR value This flag is set by hardware and cleared by software writing 0. Note:It is not used in I^2S mode.
3	UDR	R	0	Underrun flag 0:No underflow occurred 1:Underrun occurred This flag is set by hardware and reset by a software sequence. Note:This bit is not used in SPI mode.
2	CHSIDE	R	0	Channel side 0:Channel Left has to be transmitted or has been received 1:Channel Right has to be transmitted or has been received Note:This bit is not used in SPI mode. It has no significance in PCM mode.
1	TXE	R	1	Transmit buffer empty 0:Tx buffer not empty 1:Tx buffer empty
0	RXNE	R	0	Receive buffer not empty 0:Rx buffer not empty 1:Rx buffer empty

29.5.4. SPI data register (SPI_DR)

Puya Confidential

Address offset:0x0C

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	DR [15:0]	RW	0	Data register. Data received or to be transmitted. The data register serves as an interface between the Rx and Tx FIFOs. When the data register is read, RxFIFO is accessed while the write to data register accesses TxFIFO. Note:Data is always right-aligned. Unused bits are ignored when writing to the register, and read as zero when the register is read. The Rx threshold setting must always correspond with the read access currently used.

29.5.5. SPI CRC polynomial register (SPI_CRCPR)

Address offset:0x10

Reset value:0x00000007

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY [15:0]															
RW															

Bit	Name	R/W	Reset Value	Function
15:0	CRCPOLY [15:0]	RW	0x7	CRC polynomial register This register contains the polynomial for the CRC calculation. The CRC polynomial (0x0007) is the reset value of this register. Another polynomial can be configured as required. Note:These bits are not used in I^2S mode. Note:The polynomial value should only be odd. No even values are supported.

29.5.6. SPI Rx CRC register (SPI_RXCRCR)

Address offset:0x14

Reset value:0x00000000

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxCRC [15:0]															
R															

Bit	Name	R/W	Reset Value	Function
15:0	RxCRC [15:0]	R	0	Rx CRC register When CRC calculation is enabled, the RXCRC [15:0] bits contain the computed CRC value of the subsequently received bytes. This register is reset when the CRCEN bit in SPI_CR1 register is written to 1. The CRC is calculated serially using the polynomial programmed in the SPI_CRCPR register. Only the 8 LSB bits are considered when the CRC frame format is set to be 8-bit length. CRC calculation is done based on any CRC8 standard. The entire 16-bits of this register are considered when a 16-bit CRC frame format is selected. CRC calculation is done based on any CRC16 standard. Note:A read to this register when the BSY Flag is set could return an incorrect value. Note:These bits are not used in I^2S mode.

29.5.7. SPI Tx CRC register (SPI_TXCRCR)

Address offset:0x18

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxCRC [15:0]															
R															

Bit	Name	R/W	Reset Value	Function
15:0	TxCRC [15:0]	R	0	Tx CRC register When CRC calculation is enabled, the TXCRC [15:0] bits contain the computed CRC value of the subsequently transmitted bytes. This register is reset when the CRCEN bit in SPI_CR1 register is written to 1. The CRC is calculated serially using the polynomial programmed in the SPI_CRCPR register.

Puya Confidential

				Only the 8 LSB bits are considered when the CRC frame format is set to be 8-bit length. CRC calculation is done based on any CRC8 standard. The entire 16-bits of this register are considered when a 16-bit CRC frame format is selected. CRC calculation is done based on any CRC16 standard. Note:A read to this register when the BSY Flag is set could return an incorrect value. Note:These bits are not used in I^2S mode.

29.5.8. SPI_I2S configuration register (SPI_I2S_CFGR)

Address offset:0x1C

Reset value:0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				I2SMOD	I2SE	I2SCFG		PCMSYNC	Res	I2SSTD		CKPOL	DATLEN		CHLEN
-				RW	RW	RW		RW	-	RW		RW	RW		RW

Bit	Name	R/W	Reset Value	Function
15:12	Reserved	-	-	Reserved
11	I2SMOD	RW	0	I^2S mode selection 0:SPI mode is selected 1:I^2S mode is selected Note:This bit can only be configured when SPI or I2S is disabled.
10	I2SE	RW	0	I^2S enable 0:I^2S is disabled; 1:I^2S is enabled. Note:This bit is not used in SPI mode.
9:8	I2SCFG	RW	0	I2S configuration mode 00:Slave - transmit 01:Slave - receive 10:Master - transmit 11:Master - receive Note:This bit can only be configured when I^2S is disabled. This bit is not used in SPI mode.
7	PCMSYNC	RW	0	PCM frame synchronization 0:Short frame synchronization 1:Long frame synchronization

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Note:This bit has a meaning only if I2SSTD = 11 (PCM standard is used). This bit is not used in SPI mode.
6	Reserved	-	-	Reserved
5:4	I2SSTD	RW	0	I^2S standard selection 00:I^2S Philips standard 01:MSB justified standard (left justified) 10:LSB justified standard (right justified) 11:PCM standard Note:For correct operation, these bits should be configured when the I^2S is disabled. This bit is not used in SPI mode.
3	CKPOL	RW	0	Inactive state clock polarity 0:I^2S clock inactive state is low level 1:I^2S clock inactive state is high level Note:For correct operation, these bits should be configured when the I^2S is disabled. This bit is not used in SPI mode. The bit CKPOL does not affect the CK edge sensitivity used to receive or transmit the SD and WS signals.
2:1	DATLEN	RW	0	Data length to be transferred 00:16-bit data length 01:24-bit data length 10:32-bit data length 11:Not allowed Note:For correct operation, these bits should be configured when the I^2S is disabled. This bit is not used in SPI mode.
0	CHLEN	RW	0	Channel length (number of bits per audio channel) 0:16-bit wide 1:32-bit wide The bit write operation has a meaning only if DATLEN = 00 otherwise the channel length is fixed to 32-bit by hardware whatever the value filled in. Note:For correct operation, these bits should be configured when the I^2S is disabled. This bit is not used in SPI mode.

29.5.9. SPI_I2S prescaler register (SPI_I2SPR)

Puya Confidential

Address offset:0x20

Reset value:0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						MCKOE	ODD	I2SDIV							
-						RW	RW	RW							

Bit	Name	R/W	Reset Value	Function
15:10	Reserved	-	-	Reserved
9	MCKOE	RW	0	Master clock output enable 0:Master clock output is disabled 1:Master clock output is enabled Note:For correct operation, these bits should be configured when the I^2S is disabled. It is used only when the I^2S is in master mode. This bit is not used in SPI mode.
8	ODD	RW	0	Odd factor for the prescaler 0:Real divider value is = I2SDIV *2 1:Real divider value is = (I2SDIV * 2) + 1 Note:For correct operation, these bits should be configured when the I^2S is disabled. It is used only when the I^2S is in master mode. This bit is not used in SPI mode.
7:0	I2SDIV	RW	0x2	I^2S linear prescaler I2SDIV [7:0] = 0 or I2SDIV [7:0] = 1 are forbidden values. Note:For correct operation, these bits should be configured when the I^2S is disabled. It is used only when the I^2S is in master mode. This bit is not used in SPI mode.

Puya Confidential

30. Universal synchronous/asynchronous receiver transmitter (USART)

30.1. Introduction

The USARTs provide a flexible method for full-duplex data exchange with external devices using the industry-standard NRZ asynchronous serial data format. The USART utilizes a fractional baud rate generator to provide a wide range of baud rate options.

It supports synchronous one-way communication and half-duplex single wire communication. It also supports the LIN (local interconnection network), Smartcard Protocol and IrDA (infrared data association) SIR ENDEC specifications, and modem operations (CTS/RTS). It also supports multiprocessor communications.

High-speed data communication can be achieved by using the DMA method of the multi-buffer configuration.

30.2. USART main features

- 2 USARTs that support full features (USART1 and USART2), and 2 USARTs that do not support LIN, SCEN, and IRDA (USART3 and USART4)
- Full-duplex asynchronous communication
- Programmable baud rate shared by transmit and receive, up to 4.5 Mbit/s
- Programmable data length of 8 or 9 bits
- Configurable stop bits (0.5, 1, 1.5 or 2 bits)
- The transmitter provides a clock for synchronous transmission
- Single-wire Half-duplex communications
- Separate enable bits for transmitter and receiver
- Parity control
  - — Transmit parity bit
  - — Check the received data
- Transfer detection flag
  - — Receive buffer full
  - — Transmit buffer empty
  - — End of transmission flags
- Support LIN master transmit sync break and slave detect break
  - — Generates 13-bit break and detects 10/11-bit breaks when configured for LIN
- IRDA SIR encoder/decoder
  - — Support a 3/16 - bit duration in Run mode
- Smart card emulation function
  - — Smart card interface supports ISO7816 - 3 asynchronous protocol
  - — 0.5 and 1.5 stop bits for the smart card
- Four error detection flags

Puya Confidential

— Overrun error
— Noise error
— Frame error
— Parity error

- Ten interrupt sources with flags
  — CTS change
  — LIN break detection
  — Transmit data register empty
  — Transmission complete
  — Receive full data register
  — Bus idle detected
  — Overrun error
  — Frame error
  — Noise error
  — Parity error
- Multiprocessor communication. enter into mute mode if address match does not occur.
- Wake up from mute mode (by idle line detection or address mark detection)

Two receiver wake-up modes:Address bit (MSB, 9th bit), Idle line

30.3. USART functional description

The interface is externally connected to another device by three pins. Any USART bidirectional communications require a minimum of two pins:receive data in (RX) and transmit data out (TX).

RX: Receive data input. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

TX: Transmit data output. When the transmitter is disabled, the output pin returns to its IO port configuration. When the transmitter is enabled and nothing is to be transmitted, the T_X pin is at high level. In single-wire and smartcard mode, this IO is used to transmit and receive the data.

1) An idle line prior to transmission or reception
2) A start bit
3) A data word (8 or 9 bits) the least significant bit first
4) 0.5, 1, 1.5, 2 Stop bits indicating that the frame is complete
5) This interface uses a fractional baud rate generator - with a 12-bit mantissa and 4-bit fraction
6) A status register (USART_SR)
7) USART data register (USART_DR)
8) A baud rate register (USART_BRR), 12-bit mantissa and 4-bit fraction.
9) A guard-time register (USART_GTPR) in case of Smartcard mode.

The following pin is required to interface in synchronous mode:

CK: Transmitter clock output.

This pin outputs the transmitter data clock for synchronous transmission (no clock pulses on start bit and stop bit, and a software option to send a clock pulse on the last data bit). In parallel data can be

Puya Confidential

received synchronously on R_X. This can be used to control peripherals that have shift registers (e.g. LCD drivers). The clock phase and polarity are software programmable.

The following pins are required in hardware flow control mode:

- **nCTS:**Clear to send blocks the data transmission at the end of the current transfer when high
- **nRTS:**Request to send indicates that the USART is ready to receive a data (when low).

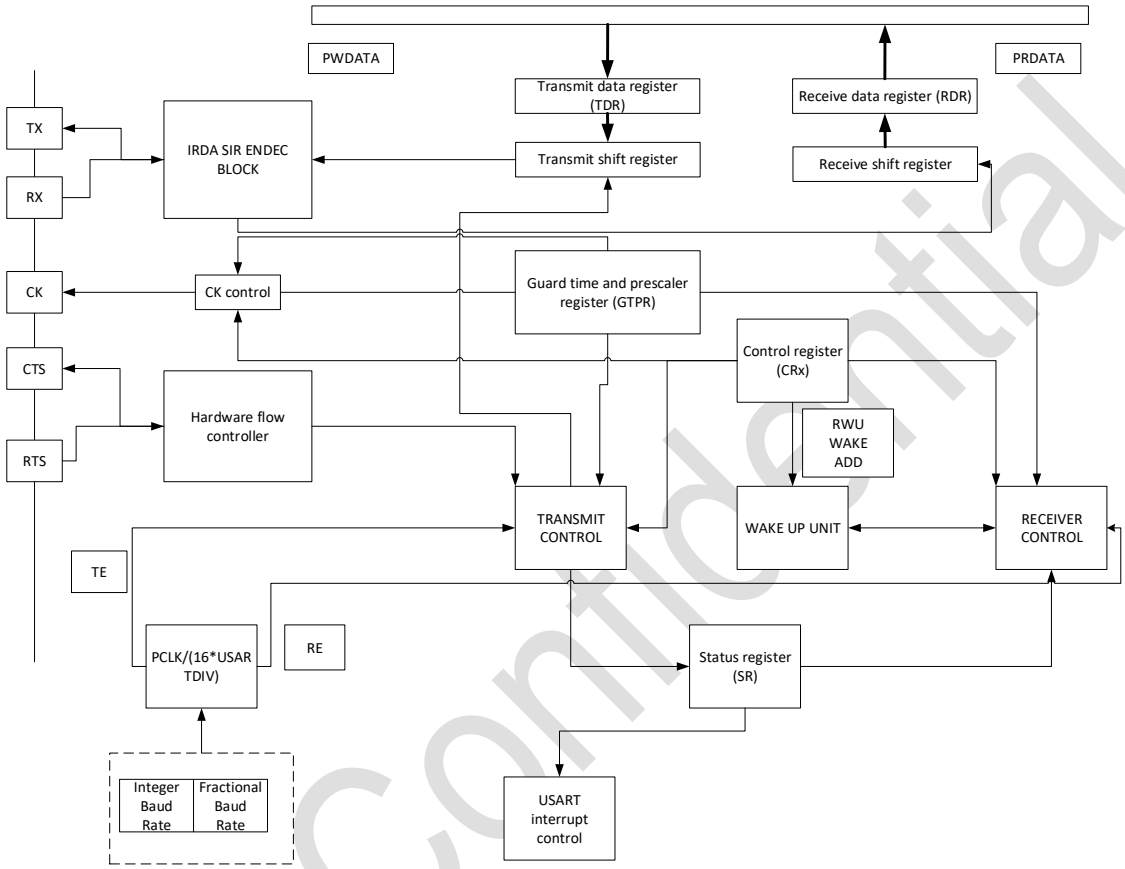


Figure 30-1 USART block diagram

30.3.1. USART character description

Word length may be selected as being either 8 or 9 bits by programming the M bit in the USART_CR1 register. The T_X pin is in low state during the start bit. It is in high state during the stop bit.

An Idle character is interpreted as an entire frame of 1 followed by the start bit of the next frame which contains data (The number of 1's will include the number of stop bits).

A Break character is interpreted on receiving “0” s for a frame period. At the end of the break frame the transmitter inserts either 1 or 2 stop bits (logic “1” bit) to acknowledge the start bit.

Transmission and reception are driven by a common baud rate generator, the clock for each is generated when the enable bit is set respectively for the transmitter and receiver.

Puya Confidential

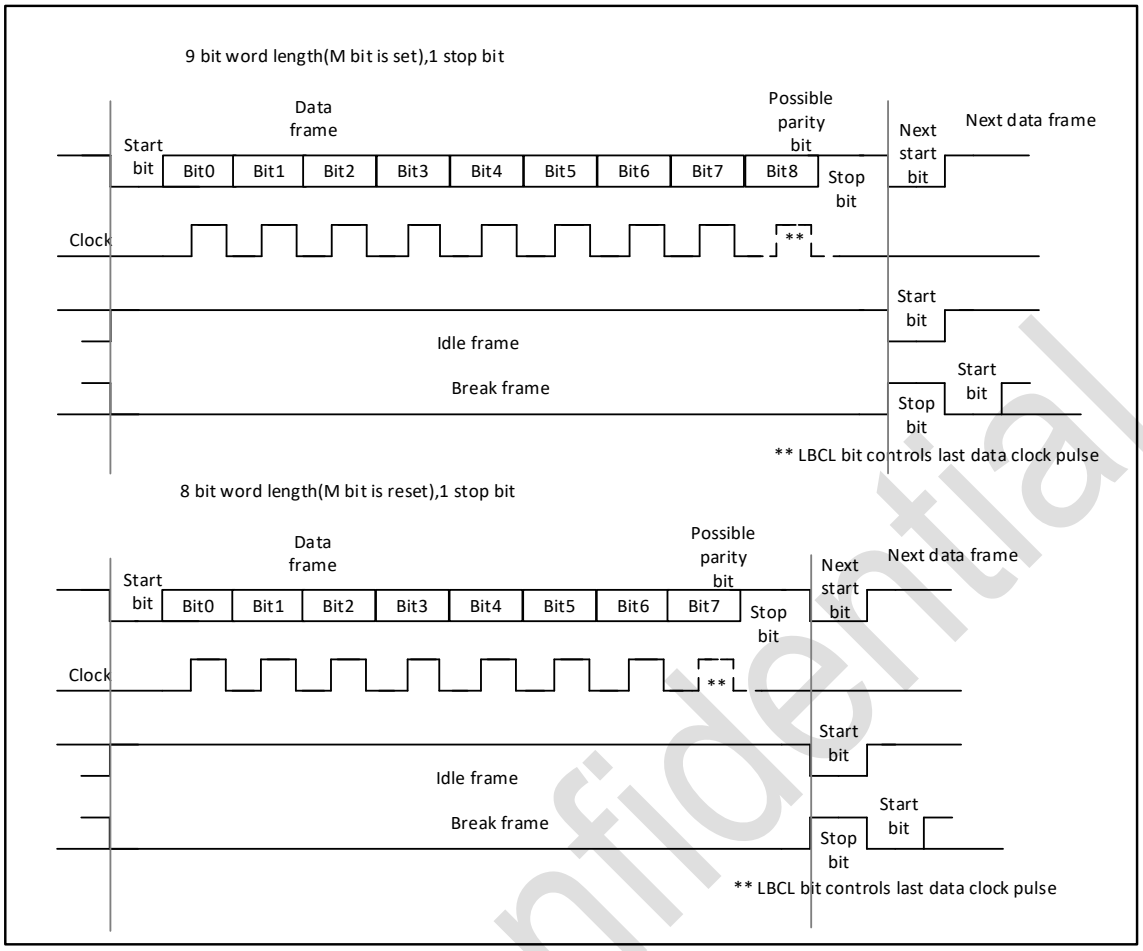


Figure 30-2 Word length programming

30.3.2. Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. When the transmit enable bit (TE) is set, the data in the transmit shift register is output on the TX pin and the corresponding clock pulses are output on the CK pin.

30.3.2.1. Character transmission

During a USART transmission, data shifts out the least significant bit first on the T_X pin. In this mode, the USART_DR register consists of a buffer (TDR) between the internal bus and the transmit shift register.

Every character is preceded by a start bit, which is a logic level low for one-bit period. The character is terminated by a configurable number of stop bits. The following stop bits are supported by USART:0.5, 1, 1.5 and 2 stop bits.

Notes:

The TE bit should not be reset during transmission of data. Resetting the TE bit during the transmission will corrupt the data on the TX pin as the baud rate counters will get frozen. The current data being transmitted will be lost.

An idle frame will be sent after the TE bit is enabled.

Puya Confidential

30.3.2.2. Configurable stop bits

The number of stop bits to be transmitted with every character can be programmed in Control register 2, bits 13, 12.

1) 1 stop bit:This is the default value.

2) 2 stop bits:This is supported by normal USART, single-wire and modem modes.

An idle frame transmission will include the stop bits.

A break transmission will be 10 low bits followed by the configured number of stop bits (when m = 0) and 11 low bits followed by the configured number of stop bits (when m = 1). It is not possible to transmit long breaks (break of length greater than 10/11 low bits)

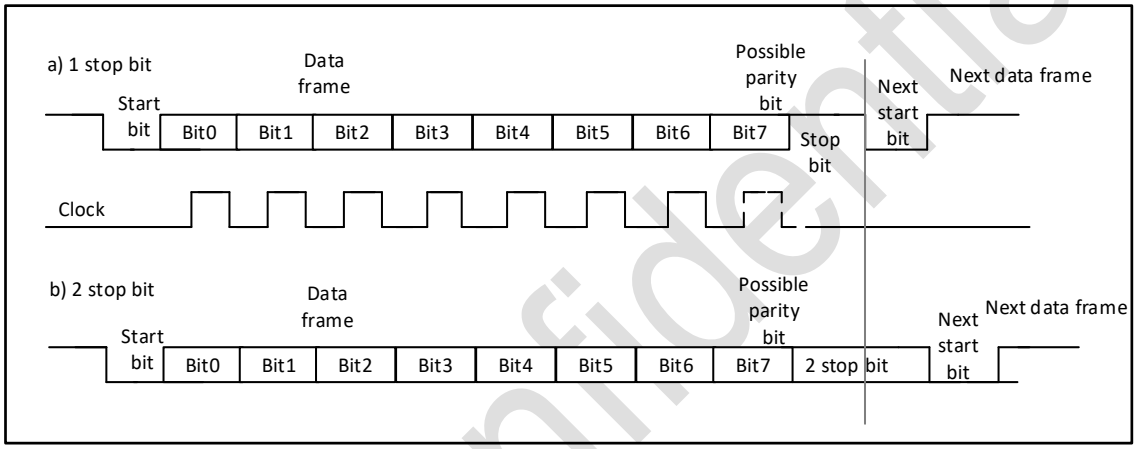


Figure 30-3 Configurable stop bits

Procedure:

1) Enable the USART by writing the UE bit in USART_CR1 register to 1.

2) Program the M bit in USART_CR1 to define the word length.

3) Program the number of stop bits in USART_CR2.

4) Select DMA enable (DMAT) in USART_CR3 if multibuffer communication must take place. Configure the DMA register as explained in multibuffer communication.

5) Select the desired baud rate using the USART_BRR register.

6) Set the TE bit in USART_CR1 to send an idle frame as first transmission.

7) Write the data to send in the USART_DR register (this clears the TXE bit). Repeat this for each data to be transmitted in case of single buffer.

8) After writing the last data into the USART_DR register, wait until TC=1. This indicates that the transmission of the last frame is complete. This is required for instance when the USART is disabled or enters the Halt mode to avoid corrupting the last transmission.

30.3.2.3. Single byte communication

The TXE bit is always cleared by a write to the data register. The TXE bit is set by hardware and it indicates:

- The data has been moved from TDR to the shift register and the data transmission has started.

Puya Confidential

- The TDR register is empty.
- The next data can be written in the USART_DR register without overwriting the previous data.

This flag generates an interrupt if the TXEIE bit is set.

When a transmission is taking place, a write instruction to the USART_DR register stores the data in the TDR register and which is copied in the shift register at the end of the current transmission.

When no transmission is taking place, a write instruction to the USART_DR register places the data directly in the shift register, the data transmission starts, and the TXE bit is immediately set.

If a frame is transmitted (after the stop bit) and the TXE bit is set, the TC bit goes high. An interrupt is generated if the TCIE bit is set in the USART_CR1 register.

After writing the last data into the USART_DR register, it is mandatory to wait for TC=1 before disabling the USART or causing the microcontroller to enter the low-power mode.

The TC bit is cleared by the following software sequence:

1. A read from the USART_SR register;

2. A write to the USART_DR register.

Note:The TC bit is cleared by writing 0 to it. This clearing sequence is recommended only for Multi-buffer communication.

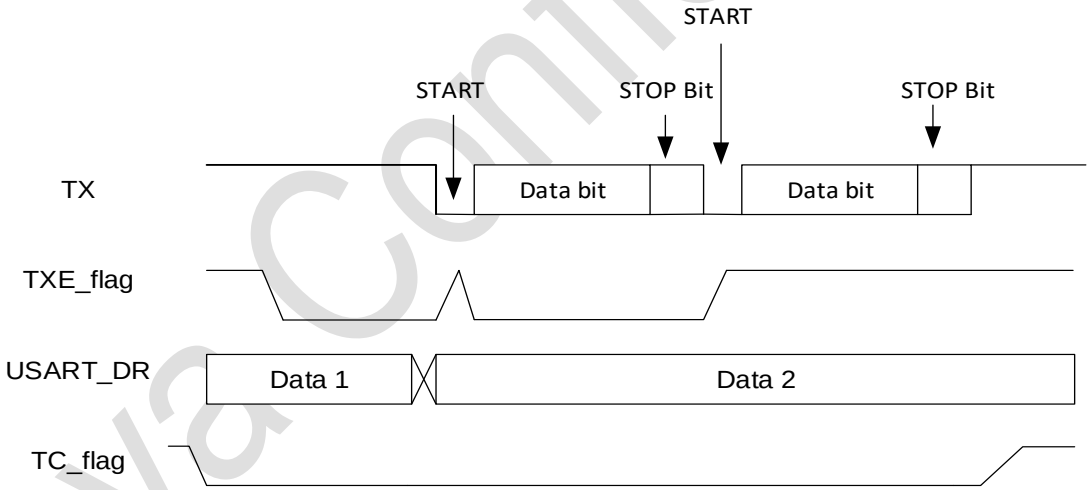


Figure 30-4 TC/TXE behavior when transmitting

30.3.2.4. Break characters

Setting the SBK bit transmits a break character. The break frame length depends on the Mbit. If the SBK bit is set to ‘1’ a break character is sent on the TX line after completing the current character transmission. This bit is reset by hardware when the break character is completed (during the stop bit of the break character). The USART inserts a logic 1 bit at the end of the last break frame to guarantee the recognition of the start bit of the next frame.

Note:If the software resets the SBK bit before the commencement of break transmission, the break character will not be transmitted. For two consecutive breaks, the SBK bit should beset after the stop bit of the previous break.

Puya Confidential

30.3.2.5. Idle characters

Setting the TE bit drives the USART to send an idle frame before the first data frame.

30.3.3. Receiver

The USART can receive data words of either 8 or 9 bits depending on the M bit in the USART_CR1 register.

30.3.3.1. Start bit detection

In the USART, the start bit is detected when a specific sequence of samples is recognized. The sequence is:1110X0X0X0000

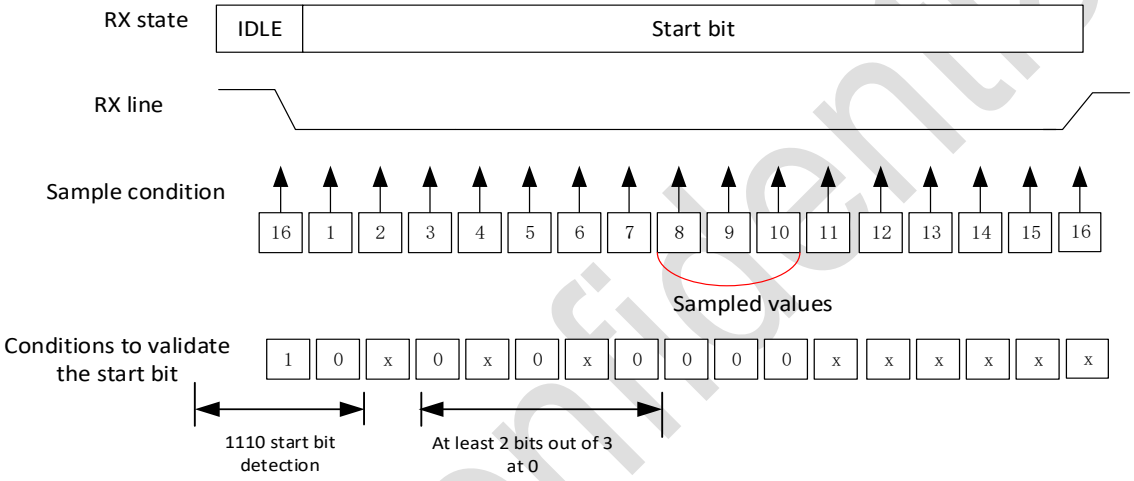


Figure 30-5 Start bit detection

If the sequence is not complete, the start bit detection aborts and the receiver returns to the idle state (no flag is set) where it waits for a falling edge. The start bit is confirmed (RXNE flag set, interrupt generated if RXNEIE=1) if the 3 sampled bits are at 0 (first sampling on the 3rd, 5th and 7th bits finds the 3 bits at 0 and second sampling on the 8th, 9th and 10th bits also finds the 3 bits at 0).

The start bit is validated (RXNE flag set, interrupt generated if RXNEIE=1) but the NE noise flag is set if, for both samplings, at least 2 out of the 3 sampled bits are at 0 (sampling on the 3rd, 5th and 7th bits and sampling on the 8th, 9th and 10th bits). If this condition is not met, the start detection aborts and the receiver returns to the idle state (no flag is set).

If, for one of the samplings (sampling on the 3rd, 5th and 7th bits or sampling on the 8th, 9th and 10th bits), 2 out of the 3 bits are found at 0, the start bit is validated but the NE noise flag bit is set.

30.3.3.2. Character reception

During a USART reception, data shifts in least significant bit first through the RX pin. In this mode, the USART_DR register consists of a buffer (RDR) between the internal bus and the received shift register.

Procedure:

1. Enable the USART by writing the UE bit in USART_CR1 register to 1.

2. Program the M bit in USART_CR1 to define the word length.

Puya Confidential

3. Program the number of stop bits in USART_CR2.

4. Select DMA enable (DMAR) in USART_CR3 if multibuffer communication must take place. Configure the DMA register as explained in multibuffer communication.

5. Select the desired baud rate using the baud rate register USART_BRR

6. Set the RE bit USART_CR1. This enables the receiver which begins searching for a start bit.

When a character is received:

- The RXNE bit is set. It indicates that the content of the shift register is transferred to the RDR. In other words, data has been received and can be read (as well as its associated error flags).
- An interrupt is generated if the RXNEIE bit is set.
- The error flags can be set if a frame error, noise or an overrun error has been detected during reception.
- In multibuffer, RXNE is set after every byte received and is cleared by the DMA read to the Data register.
- In single buffer mode, clearing the RXNE bit is performed by a software read to the USART_DR register. The RXNE flag can also be cleared by writing a zero to it. The RXNE bit must be cleared before the end of the reception of the next character to avoid an overrun error.
  Note:The RE bit should not be reset while receiving data. If the RE bit is disabled during reception, the reception of the current byte will be aborted.

30.3.3.3. Break characters

When a break character is received, the USART handles it as a framing error

30.3.3.4. Idle characters

When an idle frame is detected, there is the same procedure as a data received character plus an interrupt if the IDLEIE bit is set.

30.3.3.5. Overrun error

An overrun error occurs when a character is received when RXNE has not been reset. Data cannot be transferred from the shift register to the RDR register until the RXNE bit is cleared. The RXNE flag is set after every byte received. An overrun error occurs if RXNE flag is set when the next data is received or the previous DMA request has not been serviced.

When an overrun error occurs:The ORE bit is set.

- The RDR content will not be lost. The previous data is available when a read to USART_DR is performed.
- The shift register will be overwritten. After that point, any data received during overrun is lost.
- An interrupt is generated if either the RXNEIE bit is set or both the EIE and DMAR bits are set.
- The ORE bit is reset by a read to the USART_SR register followed by a USART_DR register read operation.

Note:The ORE bit, when set, indicates that at least 1 data has been lost. There are two possibilities:

Puya Confidential

- If RXNE=1, then the last valid data is stored in the receive register RDR and can be read.
- If RXNE=0, then it means that the last valid data has already been read and thus there is nothing to be read in the RDR. This case can occur when the last valid data is read in the RDR at the same time as the new (and lost) data is received. It may also occur when the new data is received during the reading sequence (between the USART_SR register read access and the USART_DR read access).

30.3.3.6. Noise error

Over-sampling techniques are used (except in synchronous mode) for data recovery by discriminating between valid incoming data and noise.

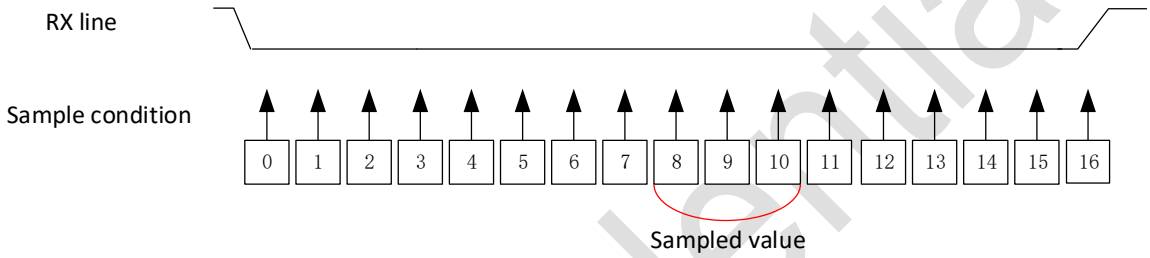


Figure 30-6 Data sampling for noise detection

Table 30-1 Data sampling for noise detection

Sampled value	NE status	Received bit value	Data validity
000	0	0	Valid
001	1	0	Not valid
010	1	0	Not valid
011	1	1	Not valid
100	1	0	Not valid
101	1	1	Not valid
110	1	1	Not valid
111	0	1	Valid

When noise is detected in a frame:

- The NE is set at the rising edge of the RXNE bit.
- The invalid data is transferred from the Shift register to the USART_DR register.
- No interrupt is generated in case of single byte communication. However, this bit rises at the same time as the RXNE bit which itself generates an interrupt. In case of multibuffer communication an interrupt will be issued if the EIE bit is set in the USART_CR3 register.
- The NE bit is reset by a USART_SR register read operation followed by a USART_DR register read operation.

30.3.3.7. Framing error

A framing error is detected when:

The stop bit is not recognized on reception at the expected time, following either a desynchronization or excessive noise.

Puya Confidential

When the framing error is detected:

- The FE bit is set by hardware
- The invalid data is transferred from the Shift register to the USART_DR register.
- No interrupt is generated in case of single byte communication. However, this bit rises at the same time as the RXNE bit which itself generates an interrupt. In case of multibuffer communication an interrupt is issued if the EIE bit is set in the USART_CR3 register.
- The FE bit is reset by a read to the USART_SR register followed by a USART_DR register read operation.

30.3.3.8. Configurable stop bits during reception

The number of stop bits to be received can be configured through the control bits of Control register 2 - it can be either 1 or 2 in normal mode and 0.5 or 1.5 in Smartcard mode.

- 0.5 stop bit (reception in Smartcard mode):no sampling is done for 0.5 stop bit. As a consequence, no framing error and no break frame can be detected when 0.5 stop bit is selected.
- 1 stop bit:sampling for 1 stop bit is done on the 8th, 9th and 10th samples.
- 1.5 stop bits (Smartcard mode):When transmitting in Smartcard mode, the device must check that the data is correctly sent. Thus the receiver block must be enabled (RE =1 in the USART_CR1 register) and the stop bit is checked to test if the smartcard has detected a parity error. In the event of a parity error, the smartcard forces the data signal low during the sampling - NACK signal-, which is flagged as a framing error. Then, the FE flag is set with the RXNE at the end of the 1.5 stop bit. Sampling for 1.5stop bits is done on the 16th, 17th and 18th samples. The 1.5 stop bit can be decomposed into 2 parts:one 0.5 baud clock period during which nothing happens, followed by 1 normal stop bit period during which sampling occurs halfway through.
- 2 stop bits:sampling for 2 stop bits is done on the 8th, 9th and 10th samples of the first stop bit. The framing error flag is set if a framing error is detected during the first stop bit. The second stop bit is not checked for framing error. The RXNE flag is set at the end of the first stop bit.

30.3.4. Fractional baud rate generation

The baud rate for the receiver and transmitter (Rx and Tx) are both set to the same value as programmed in the Mantissa and Fraction values of USARTDIV.

Tx / Rx baud = fCK /(16*USARTDIV)

Here fCK is the clock given to the peripheral USARTDIV is an unsigned fixed-point number. The 12-bit value is set in the USART_BRR register.

Note:The baud counters are updated with the new value of the Baud registers after a write to USART_BRR. Hence, the Baud rate register value should not be changed during communication.

How to derive USARTDIV from USART_BRR register values

Example 1:

If DIV_Mantissa = 27 and DIV_Fraction = 12 (USART_BRR = 0x1BC),

Then:

Mantisa (USARTDIV) = 27

Puya Confidential

Fraction (USARTDIV) = 12/16 = 0.75

So USARTDIV = 27.75

Example 2:

USARTDIV = 25.62 is required,

There leads to:

DIV_Fraction = 16 * 0.62 = 9.92

The nearest integer is:10 = 0x0A

DIV_mantissa = mantissa (25.620) = 25 = 0x19

Thus, USART_BRR = 0x19A

Example 3:

USARTDIV = 50.99 is required

There leads to:

DIV_Fraction = 16 * 0.99 = 15.84

The nearest integer is:16 = 0x10 = > DIV_frac [3:0] overflow = > carry must be added to the mantissa part

DIV_mantissa = mantissa (50.990 + carry) = 51 = 0x33

Thus:USART_BRR = 0x330, USARTDIV = 51

30.3.5. Tolerance of the USART receiver to clock deviation

The USART asynchronous receiver operates correctly only if the total clock system deviation is less than the tolerance of the USART receiver. The causes which contribute to the total deviation are:

1. DTRA:deviation due to the transmitter error (which also includes the deviation of the transmitter's local oscillator)
2. DQUANT:error due to the baud rate quantization of the receiver
3. DREC:deviation of the receiver local oscillator
4. DTCL:deviation due to the transmission line (generally due to the transceivers which can introduce an asymmetry between the low-to-high transition timing and the high-to-low transition timing).

DTRA + DQUANT + DREC + DTCL < USART receiver tolerance

The USART receiver can receive data correctly at up to the maximum tolerated deviation specified in Tables below, depending on the following settings:

1. 10- or 11-bit character length defined by the M bit in the USART_CR1 register
2. Oversampling by 8 or 16 defined by the OVER8 bit in the USART_CR3 register
3. Bits BRR [3:0] of USART_BRR register are equal to or different from 0000.

Puya Confidential

Table 30-2 Tolerance of the USART receiver when DIV_Fraction is 0

M bit	OVR8=0		OVR8=1	
	NF is an error	NF doesn't care	NF is an error	NF doesn't care
0	1.2 %	5.8 %	1.27 %	1.23 %
1	1.16 %	4.98 %	1.13 %	1.11 %

Table 30-3 Tolerance of the USART receiver when DIV_Fraction is different from 0

M bit	OVR8=0		OVR8=1	
	NF is an error	NF doesn't care	NF is an error	NF doesn't care
0	1.25 %	5.71 %	1.24 %	1.23 %
1	1.16 %	4.79 %	1.15 %	1.15 %

30.3.6. USART Auto baud rate detection

The USART can detect and automatically set the USARTx_BRR register value based on the reception of one character. Automatic baud rate detection is useful under following circumstances:

- The communication speed of the system is not known in advance.
- The system is using a relatively low accuracy clock source and this mechanism allows the correct baud rate to be obtained without measuring the clock deviation.

  The clock source frequency must be compatible with the expected communication speed. (You must select 16x oversampling and choose between fCK/65535 and fCK/16)

Before activating the auto baud rate detection, the auto baud rate detection mode must be selected through the ABRMOD [1:0] field in the USART_CR3 register. There are various modes based on different character patterns.

In these auto baud rate modes, the baud rate is measured several times during the synchronization data reception and each measurement is compared to the previous one.

These modes are the following:

Mode 0:Any character starting with a bit at '1'. In this case the USART measures the duration of the start bit (falling edge to rising edge).

Mode 1: Any character starting with a 10xx bit pattern. In this case, the USART measures the duration of the Start and of the 1st data bit. The measurement is done falling edge to falling edge, to ensure a better accuracy in the case of slow signal slopes.

In parallel, another check is performed for each intermediate RX line transition. An error is generated if the transitions on RX are not sufficiently synchronized with the receiver (the receiver being based on the baud rate calculated on bit 0).

Prior to activating the auto baud rate detection, the USARTx_BRR register must be initialized by writing a non-zero baud rate value.

The automatic baud rate detection is activated by setting the ABREN bit in the USARTx_CR3 register. The USART will then wait for the first character on the RX line. The auto baud rate operation completion is indicated by the setting of the ABRF flag in the USARTx_ISR register. If the line is noisy, the correct baud rate detection cannot be guaranteed. In this case the BRR value may be corrupted and

Puya Confidential

the ABRE error flag will be set. This also happens if the communication speed is not compatible with the automatic baud rate detection range (bit duration not between 16 and 65536 clock periods).

The RXNE interrupt will show the completion of the operation. The auto baud rate detection can be re-launched later by resetting the ABRF flag (by writing a ‘0’).

Note:If the UE is cleared during auto-baud, the BRR value may be corrupted.

30.3.7. Multiprocessor communication

It is possible to perform USART multiprocessor communications (with several USARTs connected in a network). For instance, one of the USARTs can be the master with its TX output connected to the RX inputs of the other USARTs, while the others are slaves with their respective TX outputs logically ANDed together and connected to the RX input of the master.

In multiprocessor configurations, it is often desirable that only the intended message recipient actively receives the full message contents, thus reducing redundant USART service overhead for all non-addressed receivers.

The non-addressed devices can be placed in Mute mode. When the Mute mode is enabled:

- None of the reception status bits can be set;
- All the receive interrupts are disabled.
- The RWU bit in the USARTx_CR1 register is set to 1. RWU can be controlled automatically by hardware or by software under certain conditions.

The USARTx can enter or exit from Mute mode using one of two methods, depending on the WAKE bit in the USARTx_CR1 register:

- Idle Line detection if the WAKE bit is reset,
- Address mark detection if the WAKE bit is set.

30.3.7.1. Idle bus detection (WAKE=0)

The USART enters mute mode when the RWU bit is written to 1. It wakes up when an Idle frame is detected. Then the RWU bit is cleared by hardware, but the IDLE bit is not set in the USART_SR register. RWU can also be written to 0 by software. An example of mute mode behavior using idle line detection is given in the figure below.

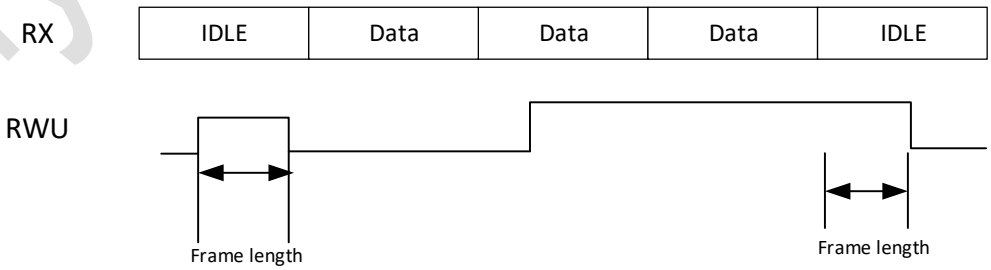


Figure 30-7 Mute mode using Idle line detection

30.3.7.2. Address mark detection (WAKE=1)

In this mode, bytes are recognized as addresses if their MSB is a ‘1’ else they are considered as data. In an address byte, the address of the targeted receiver is put in the 4 LSBs. This 4-bit word is

Puya Confidential

compared by the receiver with its own address which is programmed in the ADD bits in the USART_CR2 register.

The USART enters mute mode when an address character is received which does not match its programmed address. In this case, the RWU bit is set by hardware.

The RXNE flag is not set for this address byte and no interrupt nor DMA request is issued as the USART would have entered mute mode.

It exits from mute mode when an address character is received which matches the programmed address. Then the RWU bit is cleared, and subsequent bytes are received normally. The RXNE bit is set for the address character since the RWU bit has been cleared.

The RWU bit can be written to as 0 or 1 when the receiver buffer contains no data (RXNE=0 in the USART_SR register). Otherwise, the write attempt is ignored. An example of mute mode behavior using idle line detection is given in the figure below.

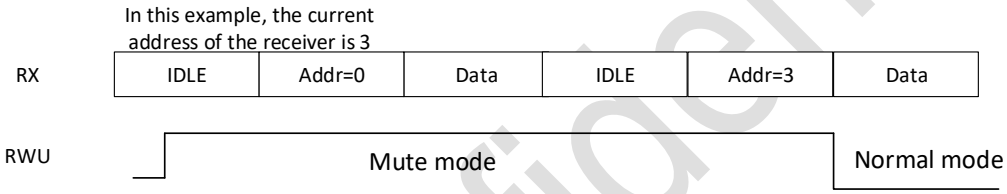


Figure 30-8 Mute mode using address mark detection

30.3.7.3. Parity control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the USART_CR1 register. Depending on the frame length defined by the M bit, the possible USART frame formats are as listed in the table below.

Table 30-4 Frame formats

M bit	PCE bit	USART frame
0	0	SB—8 bit data—STB
0	1	SB—7 bit data—PB—STB
1	0	SB—9 bit data—STB
1	1	SB—8 bit data—PB—STB

In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit. (MSB is the last one issued in the data bit, followed by the check bit or stop bit)

30.3.7.4. Even parity

The parity bit is calculated to obtain an even number of “1s” inside the frame made of the 7 or 8 LSB bits and the parity bit.

As an example, if data = 00110101 and 4 bits are set, the parity bit is equal to 0 if even parity is selected (PS bit in USARTx_CR1 = 0).

30.3.7.5. Odd parity

The parity bit is calculated to obtain an odd number of “1s” inside the frame made of the 7 or 8 LSB bits and the parity bit.

Puya Confidential

As an example, if data = 00110101 and 4 bits set, then the parity bit is equal to 1 if odd parity is selected (PS bit in USARTx_CR1 = 1).

30.3.7.6. Transmission mode:

If the PCE bit is set in USARTx_CR1, then the MSB bit of the data written in the data register is transmitted but is changed by the parity bit (even number of“1s” if even parity is selected or an odd number of “1s” if odd parity is selected). If the parity check fails, the PE flag is set in the USART_SR register and an interrupt is generated if PEIE is set in the USART_CR1 register.

30.3.8. LIN (local interconnection network) mode

Configure USART_CR2. LINEN = 1 to select LIN mode. In LIN mode, the following bits must be kept cleared:

- CLKEN in the USARTx_CR2 register;
- STOP, SCEN, HDSEL and IREN in the USART_CR3 register.

30.3.8.1. Transmission

Compared with general USART transmission, LIN transmission has the following differences:

Clear the M bit to configure 8-bit word length;

USART_CR2. LINEN = 1 needs to be configured. Setting the SBK bit sends 13 ‘0’ bits as a break character Then a bit of value ‘1’ is sent to allow the next start detection.

30.3.8.2. Reception

When the LIN mode is enabled, a break detection circuit is implemented in the USART. The detection is totally independent from the normal USART receiver. A break can be detected whenever it occurs, during idle state or during a frame.

When the receiver is enabled (RE=1 in USART_CR1), the circuit looks at the RX input for a start signal. The method for detecting start bits is the same when searching break characters or data. After a start bit has been detected, the circuit samples the next bits exactly like for the data (on the 8th, 9th and 10th samples). If 10 (when the LBDL = 0 in USART_CR2) or 11 (when LBDL=1 in USART_CR2) consecutive bits are detected as ‘0’, and are followed by a delimiter character, the LBD flag is set in USART_SR. If the LBDIE bit=1, an interrupt is generated. Before validating the break, the delimiter is checked for as it signifies that the RX line has returned to a high level.

If a ‘1’ is sampled before the 10 or 11 have occurred, the break detection circuit cancels the current detection and searches for a start bit again. If the LIN mode is disabled (LINEN=0), the receiver continues working as normal USART, without taking into account the break detection.

If LIN mode is not activated (LINEN = 0), the receiver still operates normally in USART mode and does not perform disconnect detection.

If the LIN mode is enabled (LINEN=1), as soon as a framing error occurs (i.e. stop bit detected at ‘0’, which will be the case for any break frame), the receiver stops until the break detection circuit receives either a ‘1’, if the break word was not complete, or a delimiter character if a break has been detected.

Puya Confidential

Case1: break signal not long enough => break discarded, LBD is not set

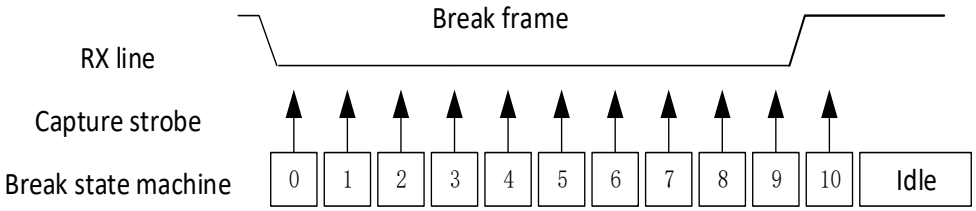
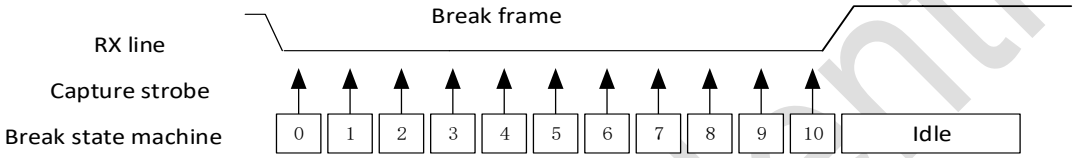


Figure 30-9 Break detection in LIN mode (break signal not long enough)

Case 2: break signal just long enough => break detected, LBD is set



Case 3: break signal long enough => break detected, LBD is set

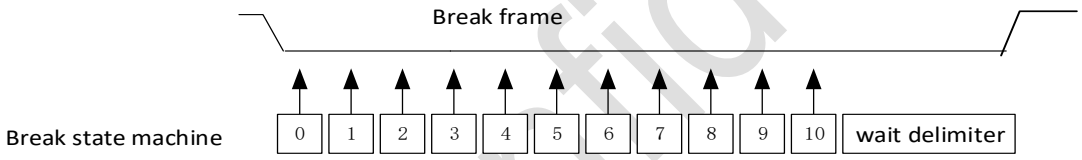
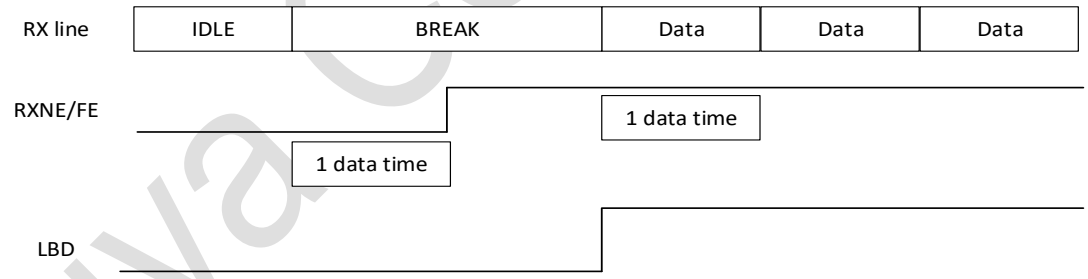


Figure 30-10 Break detection in LIN mode (break signal just long enough)

Case 1: break occurring after an Idle



Case 2: break occurring while a data is being received

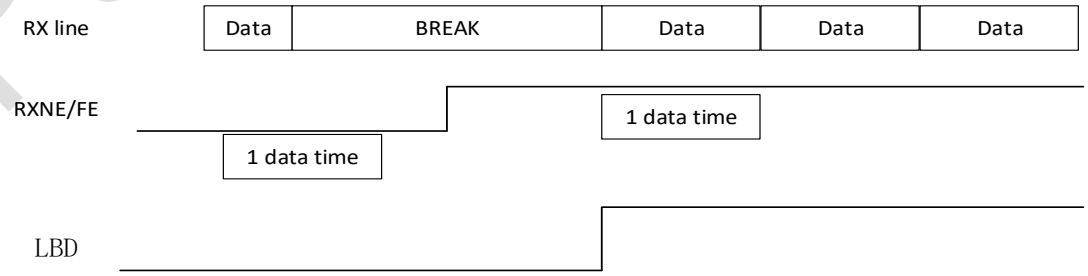


Figure 30-11 Break detection in LIN mode vs. Framing error detection

Puya Confidential

30.3.9. USART synchronous mode

The synchronous master mode is selected by programming the CLKEN bit in the USART_CR2 register to '1'. In synchronous mode, the following bits must be kept cleared:

LINEN bit in the USART_CR2 register

SCEN, HDSEL and IREN bits in the USART_CR3 register.

The USART can be used to control bidirectional synchronous serial communications in master mode. The CK pin is the output of the USART transmitter clock. No clock pulses are sent to the CK pin during start bit and stop bit. Depending on the state of the LBCL bit in the USART_CR2 register, clock pulses are, or are not, generated during the last valid data bit. The CPOL bit in the USART_CR2 register is used to select the clock polarity, and the CPHA bit in the USART_CR2 register is used to select the phase of the external clock.

During the Idle state, preamble and send break, the external CK clock is not activated.

In synchronous master mode, the USART transmitter operates exactly like in asynchronous mode. However, since CK is synchronized with TX (according to CPOL and CPHA), the data on TX is synchronous.

In synchronous master mode, the USART receiver operates in a different way compared to asynchronous mode. If RE is set to 1, the data are sampled on CK (rising or falling edge, depending on CPOL and CPHA), without any oversampling. A given setup and a hold time must be respected (which depends on the baud rate:1/16 bit time).

Note:

The CK pin operates in conjunction with the TX pin. Thus, the clock is provided only if the transmitter is enabled (TE = 1) and data are being transmitted (USART_TDR data register written). This means that it is not possible to receive synchronous data without transmitting data.

The LBCL, CPOL and CPHA bits have to be selected when both the transmitter and the receiver are disabled to ensure that the clock pulses function correctly. These bits should not be changed while the transmitter or the receiver is enabled.

It is advised that TE and RE are set in the same instruction to minimize the setup and the hold time of the receiver.

The USART supports master mode only:it cannot receive or send data related to an input clock (CK is always an output).

Puya Confidential

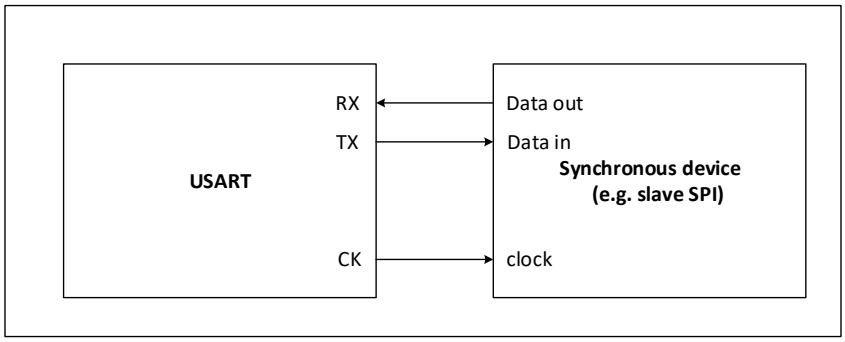


Figure 30-12 USART example of synchronous transmission

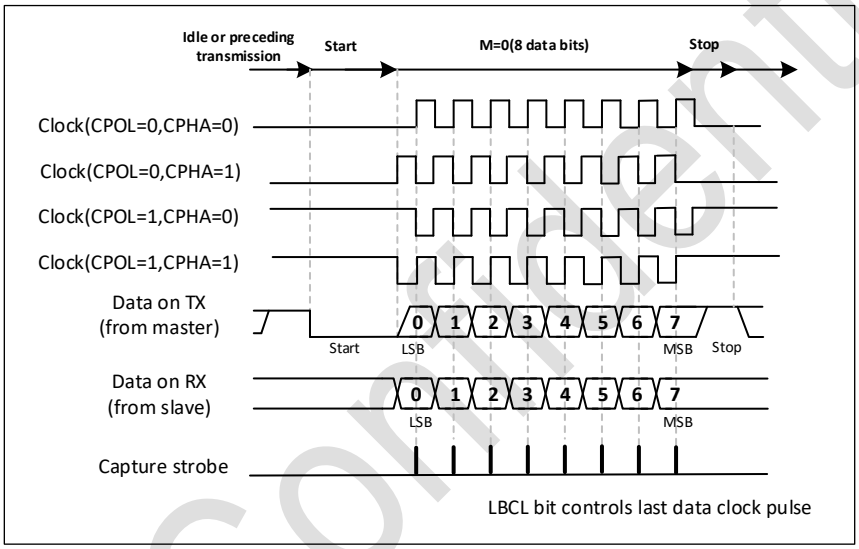


Figure 30-13 USART data clock timing diagram (M=0)

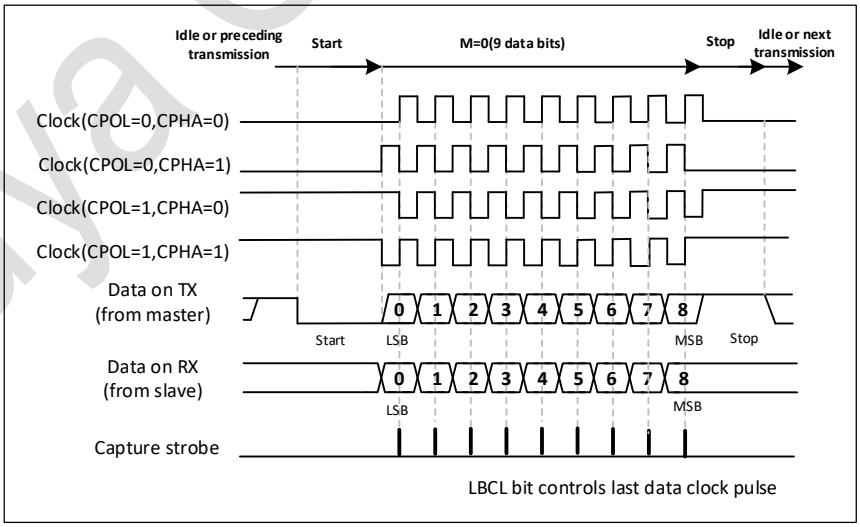


Figure 30-14 USART data clock timing diagram (M=1)

Puya Confidential

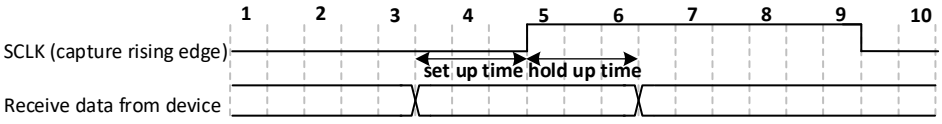


Figure 30-15 RX data setup/hold time

30.3.10. Single-wire Half-duplex communications

The single-wire half-duplex mode is selected by setting the HDSEL bit in the USARTx_CR3 register. In this mode, the following bits must be kept cleared:

- CLKEN in the USARTx_CR2 register,
- SCEN and IREN bits in the USART_CR3 register.

The USART can be configured to follow a Single-wire Half-duplex protocol. In single-wire half-duplex mode, the TX and RX pins are connected internally. The selection between half- and full-duplex communication is made with a control bit ‘HALF DUPLEX SEL’ (HDSEL in USART_CR3).

As soon as HDSEL is written to 1:

- RX is no longer used
- The TX pin is always released when no data is transmitted. Thus, it acts as a standard I/O in idle or in reception. It means that the I/O must be configured so that TX is configured as alternate function open-drain with an external pull-up.

Apart from this, the communication protocol is similar to normal USART mode. The conflicts on the line must be managed by the software (by the use of a centralized arbiter, for instance). In particular, the transmission is never blocked by hardware. The transmission is never blocked by hardware and continues as soon as data are written in the data register while the TE bit is set.

30.3.11. Smartcard

Smartcard mode is selected by setting the SCEN bit in the USART_CR3 register. In Smartcard mode, the following bits must be kept cleared:

- LINEN in the USART_CR2 register,
- HDSEL and IREN bits in the USART_CR3 register.

Moreover, the CLKEN bit may be set in order to provide a clock to the smartcard.

Note: When smart card mode is supported, the TE and RE register bits must be configured after SCEN is enabled.

The Smartcard interface is designed to support asynchronous protocol Smartcards as defined in the ISO 7816-3 standard. The USART should be configured as:

1. 8 bits plus parity:where M=1 and PCE=1 in the USART_CR1 register
2. 1.5 stop bits when transmitting and receiving:where STOP=’11’ in the USART_CR2 register.

Note:It is also possible to choose 0.5 stop bit for receiving but it is recommended to use 1.5 stop bits for both transmitting and receiving to avoid switching between the two configurations.

Figure below shows examples of what can be seen on the data line with and without parity error.

Puya Confidential

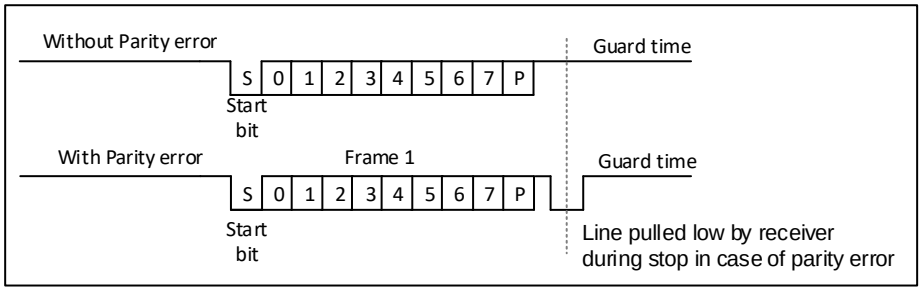


Figure 30-16 ISO 7816-3 asynchronous protocol

When connected to a smartcard, the USART TX output drives a bidirectional line that is also driven by the smartcard. In order to do this, SW_RX must be connected to the same I/O port as TX. The output enable bit TX_EN of the transmitter is set during the transmission of the start bit and the data byte, and is released (weak pull-up) during the transmission of the stop bit, so the receiver can pull the data line low in the event that a check error is found. If TX_EN is not used, TX is pulled high during the stop bit:in this case, the receiver can drive this line as long as TX is configured as an open drain.

Smartcard is a single wire half duplex communication protocol.

1) Transmission of data from the transmit shift register is guaranteed to be delayed by a minimum of 1/2 baud clock. In normal operation a full transmit shift register will start shifting on the next baud clock edge. In Smartcard mode this transmission is further delayed by a guaranteed 1/2 baud clock.
2) If a parity error is detected during reception of a frame programmed with a 0.5 or 1.5 stop bit period, the transmit line is pulled low for a baud clock period after the completion of the receive frame. This is to indicate to the Smartcard that the data transmitted to USART has not been correctly received. This NACK signal (pulling transmit line low for 1 baud clock) will cause a framing error on the transmitter side (configured with 1.5 stop bits). The application can handle re-sending of data according to the protocol. A parity error is ‘NACK’ed by the receiver if the NACK control bit is set, otherwise a NACK is not transmitted.
3) The assertion of the TC flag can be delayed by programming the Guard Time register. In normal operation, TC is asserted when the transmit shift register is empty and no further transmit requests are outstanding. In Smartcard mode an empty transmit shift register triggers the guard time counter to count up to the programmed value in the Guard Time register. TC is forced low during this time. When the guard time counter reaches the programmed value TC is asserted high.
4) The de-assertion of TC flag is unaffected by Smartcard mode.
5) If a framing error is detected on the transmitter end (due to a NACK from the receiver), the NACK will not be detected as a start bit by the receive block of the transmitter. According to the ISO protocol, the duration of the received NACK can be 1 or 2 baud clock periods.
6) On the receiver side, if a parity error is detected and a NACK is transmitted the receiver will not detect the NACK as a start bit.

Note:

1) A break character is not significant in Smartcard mode. A 0x00 data with a framing error will be treated as data and not as a break.

Puya Confidential

2) No IDLE frame is transmitted when toggling the TE bit. The IDLE frame is not defined by the ISO protocol.

Figure below details how the NACK signal is sampled by the USART. In this example, the USART transmits a data and is configured with 1.5 stop bits. The receiver part of the USART is enabled in order to check the integrity of the data and the NACK signal.

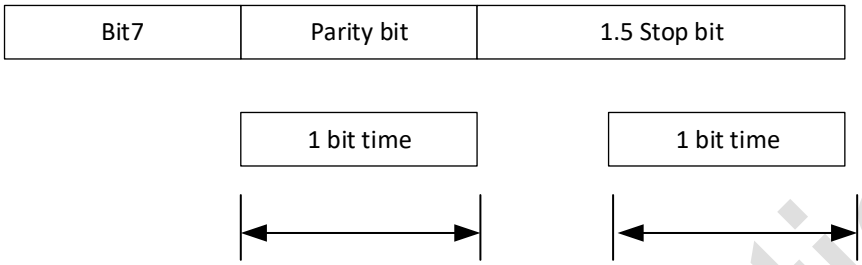


Figure 30-17 Parity error detection using the 1.5 stop bits

The USART can provide a clock to the Smartcard through the CK output. In Smartcard mode, CK is not associated to the communication but is simply derived from the internal peripheral input clock through a 5-bit prescaler. The division ratio is configured in the prescaler register USART_GTPR. The CK frequency can be programmed from fCK/2 to fCK/62, where fCK is the peripheral input clock.

30.3.12. IrDA SIR ENDEC block

IrDA mode is selected by setting the IREN bit in the USART_CR3 register. In IrDA mode, the following bits must be kept cleared:

- LINEN, STOP and CLKEN bits in the USART_CR2 register,
- SCEN and HDSEL bits in the USART_CR3 register.

30.3.12.1. IrDA normal mode

The IrDA SIR physical layer specifies use of a Return to Zero, inverted (RZI) modulation scheme that represents logic 0 as an infrared light pulse. The SIR Transmit encoder modulates the Non Return to Zero (NRZ) transmit bit stream output from USART. The output pulse stream is transmitted to an external output driver and infrared LED. USART supports only bit rates up to 115.2Kbps for the SIR ENDEC. In normal mode the transmitted pulse width is specified as 3/16 of a bit period.

The SIR receive decoder demodulates the return-to-zero-bit stream from the infrared detector and outputs the received NRZ serial bit stream to USART. The decoder input is normally HIGH (marking state) in the idle state. The transmit encoder output has the opposite polarity to the decoder input. A start bit is detected when the decoder input is low.

- IrDA is a half-duplex communication protocol. If the Transmitter is busy (i.e. the USART sends data to the IrDA encoder), any data on the IrDA receive line is ignored by the IrDA decoder. If the Receiver is busy (USART receives decoded data from the USART), data on the TX from the USART to IrDA is not encoded by IrDA. While receiving data, transmission should be avoided as the data to be transmitted could be corrupted.
- A ‘0’ is transmitted as a high pulse and a ‘1’ is transmitted as a ‘0’. The width of the pulse is specified as 3/16th of the selected bit period in normal mode.
- The SIR receive logic interprets a high state as a logic one and low pulses as logic zeros.

Puya Confidential

- The transmit encoder output has the opposite polarity to the decoder input. The SIR output is in low state when idle.
- The SIR decoder converts the IrDA compliant received signal into a bit stream for the USART.
- The IrDA specification requires the acceptance of pulses greater than 1.41 us. The acceptable pulse width is programmable. Glitch detection logic on the receiver end filters out pulses of width less than 2 PSC periods (PSC is the prescaler value programmed in the USART_GTPR). Pulses of width less than 1 PSC period are always rejected, but those of width greater than one and less than two periods may be accepted or rejected, those greater than 2 periods will be accepted as a pulse. The IrDA encoder/decoder doesn't work when PSC=0.
- The receiver can communicate with a low-power transmitter.
- In IrDA mode, the stop bits in the USART_CR2 register must be configured to '1 stop bit'.

30.3.12.2. IrDA low-power mode

Transmitter:

In low-power mode the pulse width is not maintained at 3/16 of the bit period. Instead, the width of the pulse is 3 times the low-power baud rate which can be a minimum of 1.42 MHz. Generally, this value is 1.8432 MHz (1.42 MHz < PSC< 2.12 MHz). A low- power mode programmable divisor divides the system clock to achieve this value.

Receiver:

Receiving in low-power mode is similar to receiving in normal mode. For glitch detection, the USART should discard pulses of duration shorter than 1/PSC. A valid low is accepted only if its duration is greater than 2 periods of the IrDA low-power Baud clock (PSC value in USART_GTPR).

Note:

1. A pulse of width less than two and greater than one PSC period(s) may or may not be rejected.

2. The receiver set up time should be managed by software. The IrDA physical layer specification specifies a minimum of 10 ms delay between transmission and reception (IrDA is a half-duplex protocol).

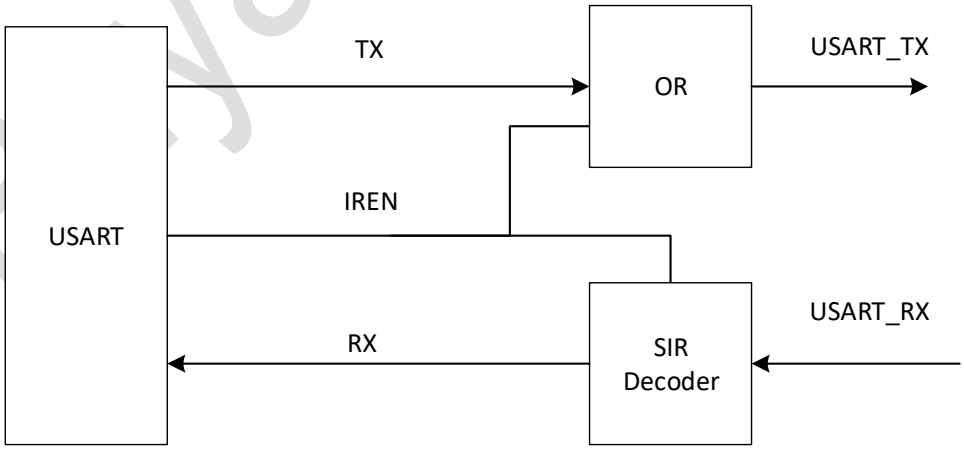


Figure 30-18 IrDA SIR ENDEC block diagram

Puya Confidential

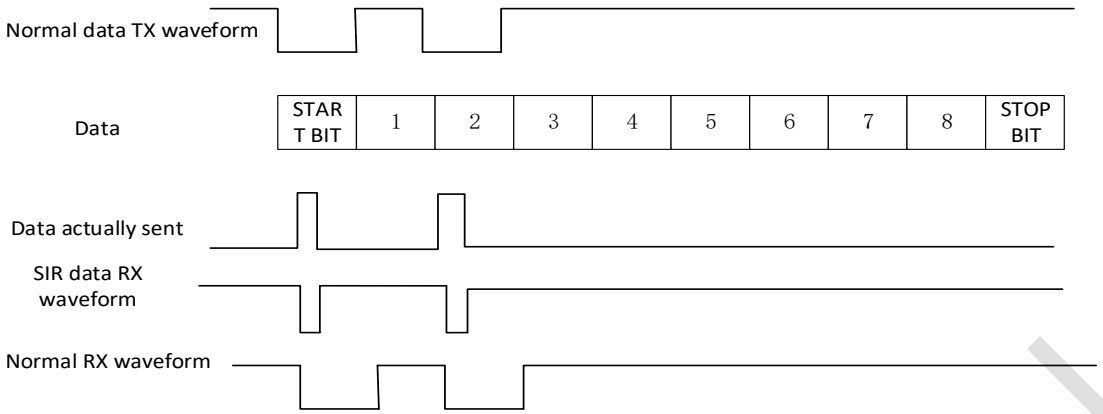


Figure 30-19 IrDA data modulation (3/16)-normal Mode

30.3.13. Continuous communication using DMA

The USART is capable of performing continuous communications using the DMA. The DMA requests for Rx buffer and Tx buffer are generated independently.

30.3.13.1. Transmission using DMA

DMA mode can be enabled for transmission by setting DMAT bit in the USART_CR3 register. Data is loaded from a SRAM area configured using the DMA peripheral to the USART_DR register whenever the TXE bit is set. To map a DMA channel for USART transmission, use the following procedure:

1) Write the USART_DR register address in the DMA control register to configure it as the destination of the transfer. The data will be moved to this address from memory after each TXE event.

2) Write the memory address in the DMA control register to configure it as the source of the transfer. The data will be loaded into the USART_DR register from this memory area after each TXE event.

3) Configure the total number of bytes to be transferred to the DMA control register.

4) Configure the channel priority in the DMA register

5) Configure DMA interrupt generation after half/full transfer as required by the application.

6) Clear the TC bit by writing 0 to it.

7) Activate the channel in the DMA register.

When the number of data transfers programmed in the DMA Controller is reached, the DMA controller generates an interrupt on the DMA channel interrupt vector.

In transmission mode, once the DMA has written all the data to be transmitted (the TCIF flag is set in the DMA_ISR register), the TC flag can be monitored to make sure that the USART communication is complete. This is required to avoid corrupting the last transmission before disabling the USART or entering the Stop mode. The software must wait until TC=1. The TC flag remains cleared during all data transfers and it is set by hardware at the last frame's end of transmission.

Puya Confidential

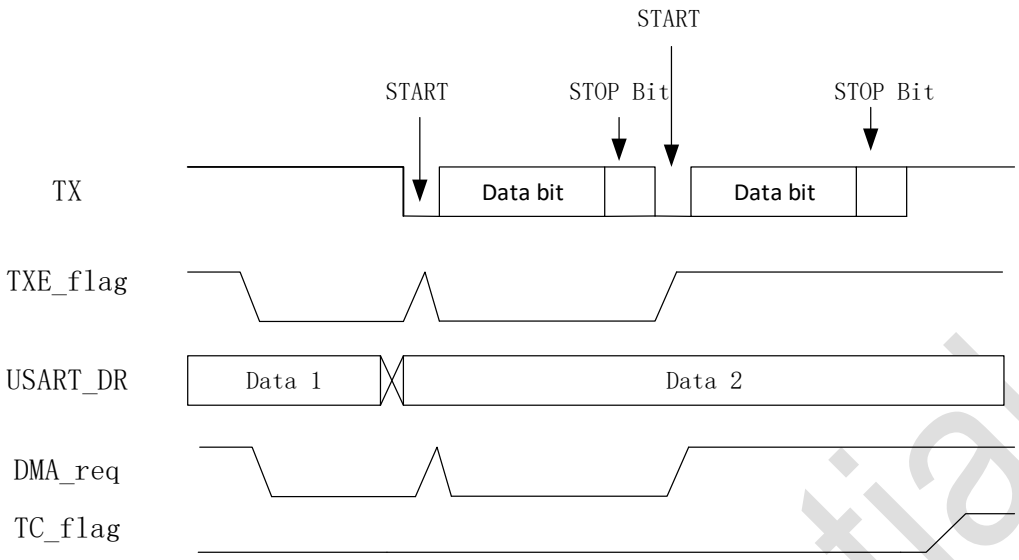


Figure 30-20 Transmission using DMA

30.3.13.2. Reception using DMA

DMA mode can be enabled for reception by setting the DMAR bit in USART_CR3 register. Data is loaded from the USART_DR register to a SRAM area configured using the DMA peripheral (refer to the DMA specification) whenever a data byte is received. To map a DMA channel for USART reception, use the following procedure:

1) Write the USART_DR register address in the DMA control register to configure it as the source of the transfer. The data will be moved from this address to the memory after each RXNE event.

2) Write the memory address in the DMA control register to configure it as the destination of the transfer. The data will be loaded from USART_DR to this memory area after each RXNE event.

3) Configure the total number of bytes to be transferred to the DMA control register.

4) Configure the channel priority in the DMA register

5) Configure DMA interrupt generation after half/full transfer as required by the application.

6) Activate the channel in the DMA control register.

When the number of data transfers programmed in the DMA Controller is reached, the DMA controller generates an interrupt on the DMA channel interrupt vector.

30.3.13.3. Error flagging and interrupt generation in multibuffer communication

In case of multibuffer communication if any error occurs during the transaction the error flag will be asserted after the current byte. An interrupt will be generated if the interrupt enable flag is set. For framing error, overrun error and noise flag which are asserted with RXNE in case of single byte reception, there will be separate error flag interrupt enable bit, which if set will issue an interrupt after the current byte with either of these errors.

30.3.14. Hardware flow control

It is possible to control the serial data flow between 2 devices by using the nCTS input and the nRTS output. The figure below shows how to connect two devices in this mode.

Puya Confidential

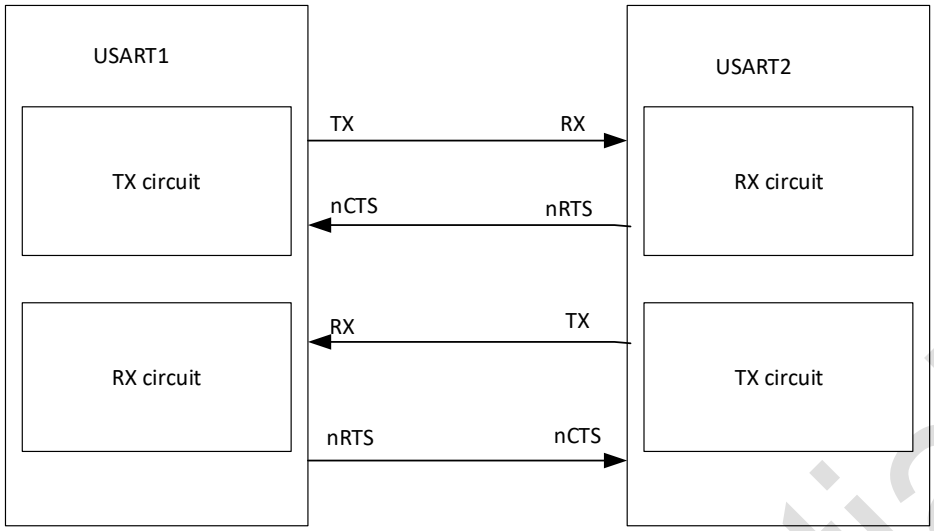


Figure 30-21 Hardware flow control between two USARTs

30.3.14.1. RTS flow control

If the RTS flow control is enabled (RTSE=1), then nRTS is asserted (tied low) as long as the USART receiver is ready to receive a new data. When the receive register is full, nRTS is deasserted, indicating that the transmission is expected to stop at the end of the current frame.

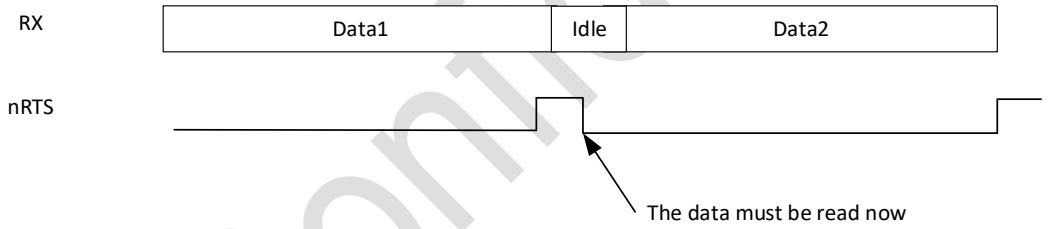


Figure 30-22 RTS flow control

30.3.14.2. CTS flow control

If the CTS flow control is enabled (CTSE=1), then the transmitter checks the nCTS input before transmitting the next frame. If nCTS is asserted (tied low), then the next data is transmitted (assuming that a data is to be transmitted, in other words, if TXE=0), else the transmission does not occur. When CTS is deasserted during a transmission, the current transmission is completed before the transmitter stops.

When CTSE = 1, the CTSIF status bit is automatically set by hardware as soon as the nCTS input toggles. It indicates when the receiver becomes ready or not ready for communication. An interrupt is generated if the CTSIE bit in the USART_CR3 register is set.

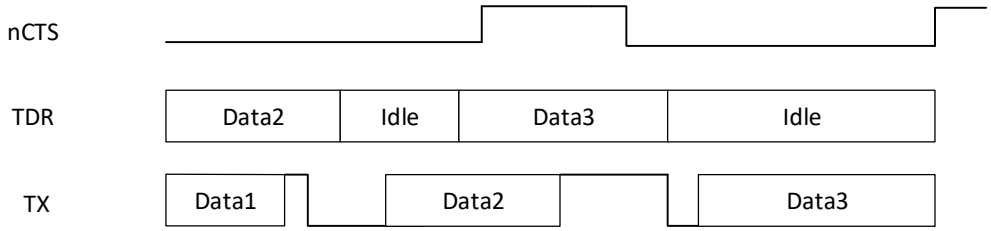


Figure 30-23 CTS flow control

Puya Confidential

30.4. USART interrupt requests

Table 30-5 USART interrupt requests

No.	Interrupt event	Event flag	Enable Control bit	Transmission/ Reception
1	Transmit data register empty	TXE	TXEIE	Transmission
2	CTS (Clear to Send) Interrupt	CTSIF	CTSIE	Transmission
3	Transmission complete	TC	TCIE	Transmission
4	Receive data register not empty (data ready to be read)	RXNE	RXNEIE	Reception
5	Overrun error	ORE		Reception
6	Idle line detected	IDLE	IDLEIE	Reception
7	Parity error	PE	PEIE	Reception
8	Break flag	LBD	LBDIE	Reception
9	Noise error/Overrun error/Framing Error in multibuffer communication	NE/ORE/FE	EIE	Reception

Note:The USART interrupt events are connected to the same interrupt vector.

Puya Confidential

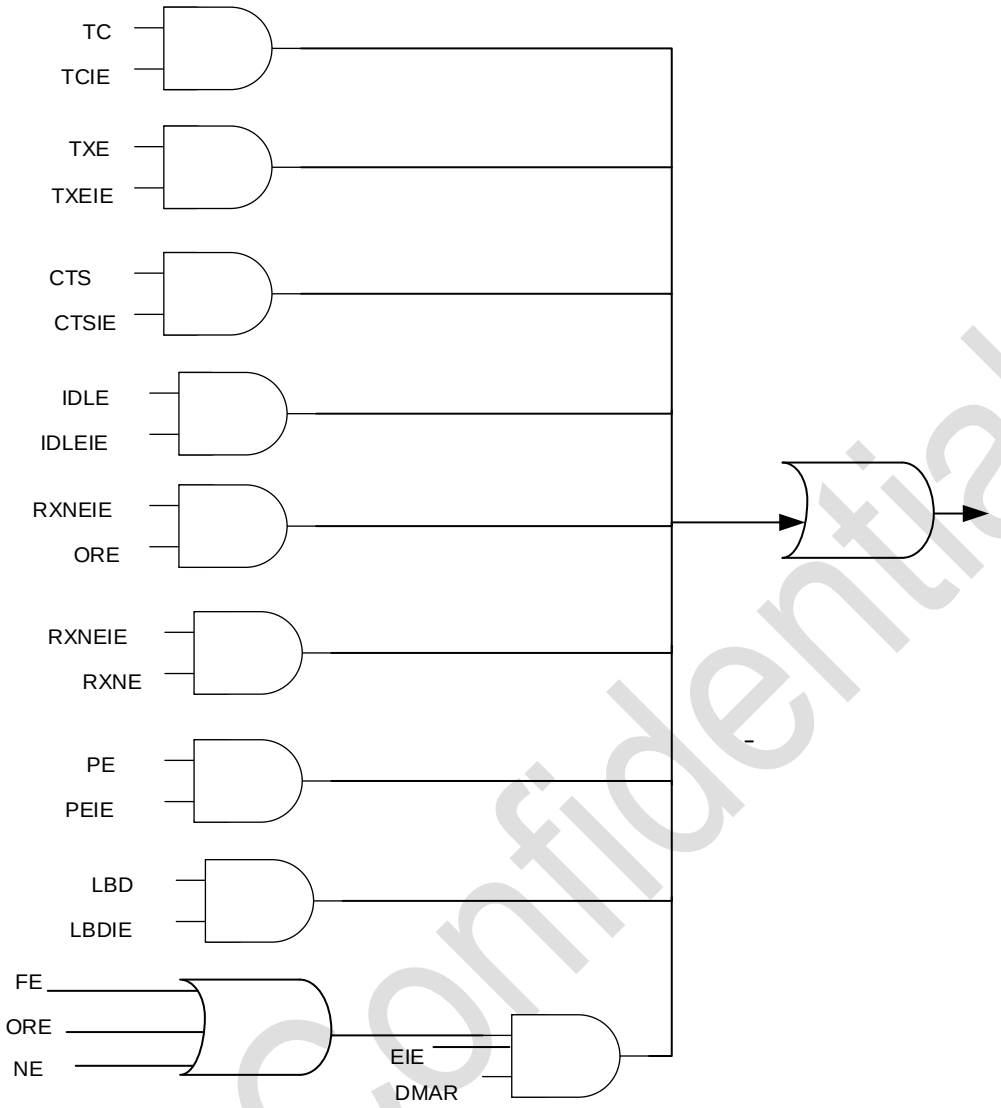


Figure 30-24 USART interrupt mapping diagram

30.5. USART register

30.5.1. USART status register (USART_SR)

Address offset:0x00

Reset value:0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res			ABRRQ	ABRE	ABRF	CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
-			W	R		RC_W0		R	RC_W0		R				

Bit	Name	R/W	Reset Value	Function
31:13	Reserved	-	-	Reserved
12	ABRRQ	W	0	Auto baud rate request

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Writing 1 to this bit resets the ABRF flag and requests an automatic baud rate measurement on the next received data frame.
11	ABRE	R	0	Auto baud rate error This bit is set by hardware if the baud rate measurement failed (baud rate out of range or character comparison failed). It is cleared by software, by writing 1 to the ABRRQ bit.
10	ABRF	R	0	Auto baud rate detection flag. This bit is set by hardware when the automatic baud rate has been set (RXNE is also set, generating an interrupt if RXNEIE = 1) or when the auto baud rate operation was completed without success (ABRE = 1) (ABRE, RXNE and FE are also set in this case). It is cleared by software, by writing 1 to the ABRRQ in the USART_SR register.
9	CTS	RC_W0	0	CTS flag This bit is set by hardware when the CTS input toggles, if the CTSE bit is set. It is cleared by software (by writing it to 0). An interrupt is generated on CTS flag if the CTSIE bit is set. 0:No change occurred on the CTS status line 1:A change occurred on the CTS status line
8	LBD	RC_W0	~~0~~	LBD:LIN break detection flag This bit is set by hardware when the LIN break is detected. It is cleared by software (by writing it to 0). An interrupt is generated if LBDIE = 1 in the USART_CR3 register. 0:LIN Break not detected; 1:LIN break detected. Note:An interrupt is generated when LBD=1 if LBDIE=1
7	TXE	R	1	Transmit data register empty This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				TXEIE bit =1. It is cleared by a write to the USART_DR register. 0:Data is not transferred to the shift register 1:Data is transferred to the shift register
6	TC	RC_W0	1	Transmission complete This bit is set by hardware if the transmission of a frame containing data is complete and if TXE is set. An interrupt is generated if TCIE=1. It is cleared by a software sequence (a read from the USART_SR register followed by a write to the USART_DR register). This clearing sequence is recommended only for multibuffer communication. The TC bit can also be cleared by writing a '0' to it. 0:Transmission not complete 1:Transmission is complete
5	RXNE	RC_W0	0	Read data register not empty This bit is set by hardware when the content of the shift register has been transferred to the USART_DR register. It is cleared by a read to the USART_DR register. An interrupt is generated if the RXNEIE bit =1. 0:Data is not received 1:Received data is ready to be read.
4	IDLE	R	0	DLE line detected This bit is set by hardware when an IDLE line is detected. An interrupt is generated if the IDLEIE bit =1. It is cleared by a software sequence (a read from the USART_SR register followed by a write to the USART_DR register). 0:No Idle Line is detected 1:Idle Line is detected
3	ORE	R	0	Overrun error. This bit is set by hardware when the word currently being received in the shift register is ready to be transferred into the RDR register while RXNE=1.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				It is cleared by a software sequence (a read from the USART_SR register followed by a write to the USART_DR register). An interrupt is generated if the RXNEIE bit =1. 0:No Overrun error 1:Overrun error is detected Note:When this bit is set, the RDR register content will not be lost, but the shift register will be overwritten. An interrupt is generated on ORE flag if the EIE bit is set.
2	NE	R	0	Noise error flag This bit is set by hardware when noise is detected on a received frame. It is cleared by a software sequence (a read from the USART_SR register followed by a write to the USART_DR register). 0:No noise is detected 1:Noise is detected Note:This bit does not generate interrupt as it appears at the same time as the RXNE bit which itself generates an interrupting. Interrupt is generated on NE flag in case of MultiBuffer communication if the EIE bit is set.
1	FE	R	0	Framing error This bit is set by hardware when a de-synchronization, excessive noise or abort characters is detected. It is cleared by a software sequence (a read from the USART_SR register followed by a write to the USART_DR register). 0:No frame error detected 1:Framing error or break character is detected Note:This bit does not generate interrupt as it appears at the same time as the RXNE bit which itself generates an interrupting. If the word currently being transferred causes both frame error and overrun error, it will be transferred, and only the ORE bit will be set. Interrupt is generated on

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				FE flag in case of MultiBuffer communication if the EIE bit is set.
0	PE	R	0	Parity error This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by a software sequence (a read from the USART_SR register followed by a write to the USART_DR register). The software must wait for the RXNE flag to be set before clearing the PE bit. An interrupt is generated if the PEIE bit =1. 0:No parity error 1:Parity error

30.5.2. USART data register (USART_DR)

Address offset:0x04

Reset value:0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res							DR [8:0]								
-							RW								

Bit	Name	R/W	Reset Value	Function
31:9	Reserved	-	-	Reserved
8:0	DR [8:0]	RW	0x0	Data value Contains the Received or Transmitted data character, depending on whether it is read from or written to. The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR). The TDR register provides the parallel interface between the internal bus and the output shift register. The RDR register provides the parallel interface between the input shift register and the internal bus. When transmitting with the parity enabled, the value written in the MSB (bit 7 or bit 8 depending on the data length) has no effect because it is replaced by the parity.

Puya Confidential

				When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.

30.5.3. USART baud rate register (USART_BRR)

Address offset:0x08

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIV_Mantissa [11:0]												DIV_Faction [3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Attention:This register can only be written when USART is disabled (USART_CR1.UE=0). In automatic baud rate detection mode, the hardware updates this register.

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:4	DIV_Mantissa [11:0]	RW	0	12-bit integer
3:0	DIV_Fraction [3:0]	RW	0	4bit decimal

30.5.4. USART control register 1 (USART_CR1)

Address offset:0x0C

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
-	-	RW													

Bit	Name	R/W	Reset Value	Function
31:14	Reserved	-	-	Reserved
13	UE	RW	0	USART enable When this bit is cleared the USART prescalers are stopped. This bit is set and cleared by software. 0:USART prescaler and outputs disabled, low-power mode 1:USART enabled The software needs to wait for USART_SR.TC to be set before it can clear the UE bit and enter the low power mode;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				At the same time, the DMA channel needs to be disabled before the UE bit is cleared.
12	M	RW	0	0:1 Start bit, 8 Data bits, n Stop bit 1:1 Start bit, 9 Data bits, n Stop bit
11	WAKE	RW	0	Receive wakeup mode. Wake up mode from mute mode. Set and cleared by software. 0:Idle line wakes up 1:Address Mark
10	PCE	RW	0	Parity control enable 0:Parity control disabled 1:Parity control enabled Parity bit:9th bit of 9bit; The 8th bit of 8bit.
9	PS	RW	0	Parity selection Set and cleared by software. 0:Even parity 1:odd parity
8	PEIE	RW	0	PE interrupt enable Set and cleared by software. 0:Disabled 1:PE interrupt enabled
7	TXEIE	RW	0	TXE interrupt enable Set and cleared by software. 0:Disabled 1:TXE interrupt enabled
6	TCIE	RW	0	Transmission complete interrupt enable Set and cleared by software. 0:Disabled 1:TC interrupt enabled
5	RXNEIE	RW	0	RXNE interrupt enable; Set and cleared by software. 0:Disabled 1:ORE or RXNE interrupt enabled
4	IDLEIE	RW	0	IDLE interrupt enable Set and cleared by software. 0:Disabled 1:IDLE interrupt enabled
3	TE	RW	0	Transmitter enable 0:Transmitter is disabled 1:Transmitter is enabled
2	RE	RW	0	Receiver enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Receiver is disabled 1:Receiver is enabled and begins searching for a start bit
1	RWU	RW	0	Receiver wakeup This bit indicates if the USART is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wakeup sequence is recognized. The Mute mode control sequence (address or IDLE) is selected by the WAKE bit in the USART_CR1 register. 0:Receiver in active mode 1:Receiver in mute mode Note1:Before selecting Mute mode, the USART must first receive a data byte, otherwise it cannot function in Mute mode with wakeup by Idle line detection. Note:2:In Address Mark Detection wakeup configuration (WAKE bit=1) the RWU bit cannot be modified by software while the RXNE bit is set.
0	SBK	RW	0	Send break This bit set is used to send break characters. It should be set by software, and will be reset by hardware during the stop bit of break. 0:No break character is transmitted 1:Break character will be transmitted

30.5.5. USART control register 2 (USART_CR2)

Address offset:0x10

Reset value:0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res	LBDIE	LBDL	Res	ADD [3:0]			
-	RW							-	RW	RW	-	RW			

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved

Puya Confidential

Bit	Name	R/W	Reset Value	Function
14	LINEN	RW	0	LIN mode enable. Set and cleared by software. 0:LIN mode disabled 1:LIN mode enabled The LIN mode enables the capability to send LIN Synch Breaks (13 low bits) using the SBK bit in the USART_CR1 register, and to detect LIN Sync breaks.
13:12	STOP [1:0]	RW	0	Stop bits 00:1 stop bit; 01:0.5 stop bit; 10:2 stop bits; 11:1.5 stop bit
11	CLKEN	RW	0	clock enable 0:Interrupt is disabled 1:CK pin enabled This bit is reserved when synchronous mode is not supported.
10	CPOL	RW	0	Clock polarity This bit allows the user to select the polarity of the clock output on the CK pin in synchronous mode. 0:Steady low value on CK pin outside transmission window 1:Steady high value on CK pin outside transmission window Attention:This register can only be written when USART is disabled (USART_CR1.UE=0).
9	CPHA	RW	0	This bit allows the user to select the phase of the clock output on the CK pin in synchronous mode. It works in conjunction with the CPOL bit to produce the desired clock/data relationship. 0:The first clock transition is the first data capture edge 1:The second clock transition is the first data capture edge Attention:This register can only be written when USART is disabled (USART_CR1.UE=0).
8	LBCL	RW	0	Whether the clock pulse of the last bit of data is output at CK pin. 0:The clock pulse of the last data bit is not output to the CK pin 1:The clock pulse of the last data bit is output to the CK pin

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				Attention:This register can only be written when USART is disabled (USART_CR1.UE=0).
7	Reserved	-	-	Reserved
6	LBDIE	RW	0	LIN break detection interrupt enable. 0:Interrupt is disabled 1:Interrupt generated An interrupt is generated by using this bit to set the LBD bit in USART_SR register.
5	LBDL	RW	0	LIN break detection length. 0:10-bit break detection 1:11 bit break detection
4	Reserved	-	-	Reserved
3:0	ADD [3:0]	RW	4'b0	Address of the USART node This is used in multiprocessor communication during mute mode, for a 4-bit wake up with address mark detection.

30.5.6. USART control register 3 (USART_CR3)

Address offset:0x14

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	ABR-MOD [1:0]		ABREN	OVER8	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
-	RW														

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14:13	ABRMOD [1:0]	RW	2'b0	Auto baud rate mode 00:Measurement of the start bit is used to detect the baud rate 01:Falling edge to falling edge measurement 10:Reserved 11:Reserved This bit field can only be written when ABREN = 0 or the USART is disabled (UE=0).
12	ABREN	RW	0	Auto baud rate enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Disabled 1:Auto baud rate detection is enabled
11	OVER8	RW	0	Oversampling mode 0:Oversampling by 16 1:Oversampling by 8 This bit is only be written when the USART is disabled (UE=0).
10	CTSIE	RW	0	CTS interrupt enable 0:Interrupt is disabled 1:CTSIF interrupt enable;
9	CTSE	RW	0	CTS enable 0:CTS hardware flow control disabled 1:CTS mode enabled Data is only transmitted when the CTS input is asserted (tied to 0). If a data is written into the data register while CTS is deasserted, the transmission is postponed until CTS is asserted.
8	RTSE	RW	0	RTS enable 0:RTS hardware flow control disabled 1:RTS interrupt enabled, data is only requested when there is space in the receive buffer. The transmission of data is expected to cease after the current character has been transmitted. The RTS output is asserted (tied to 0) when a data can be received.
7	DMAT	RW	0	DMA enable transmitter. 0:Interrupt is disabled 1:DMA mode is enabled for transmission
6	DMAR	RW	0	DMA enable receiver 0:Interrupt is disabled 1:DMA mode is enabled for reception
5	SCEN	RW	0	Smartcard mode enable. 0:Interrupt is disabled 1:Enabled This bit is only be configured when the USART is disabled (UE=0).
4	NACK	RW	0	Smartcard NACK enable 0:NACK transmission in case of parity error is disabled;

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:NACK transmission during parity error is enabled;
3	HDSEL	RW	0	Half-duplex selection 0:Half duplex mode is not selected 1:Half duplex mode is selected This bit is only be configured when the USART is disabled (UE=0).
2	IRLP	RW	0	IrDA low power 0:normal mode 1:IrDA low power mode This bit is only be configured when the USART is disabled (UE=0).
1	IREN	RW	0	IrDA mode enable This bit is set and cleared by software. 0:IrDA disabled 1:IrDA enabled This bit is only be configured when the USART is disabled (UE=0).
0	EIE	RW	0	Error interrupt enable 0:Interrupt is disabled 1:Frame error FE, overrun error ORE, noise NE interrupt enabled.

30.5.7. USART guard time and prescaler (USART_GTPR)

Address offset:0x18

Reset value:0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GT [7:0]								PSC [7:0]							
RW								RW							

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	0	Reserved
15:8	GT [7:0]	RW	0	Guard time value.

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				This bit-field gives the Guard time value in terms of number of baud clocks. This is used in Smart-card mode. The Transmission Complete flag is set after this guard time value.
7:0	PSC [7:0]	RW	0	Prescaler value. ■ In IrDA Low-power mode: PSC [7:0] = IrDA Low power baud rate Used for programming the prescaler for dividing the system clock to achieve the low-power frequency: The source clock is divided by the value given in the register (8 significant bits): 00000000:Reserved - do not program this value; 00000001:divides the source clock by 1 00000010:divides source clock by 2 …… ■ In normal IrDA mode:PSC must be set to 00000001. ■ In smartcard mode: PSC [4:0]:prescaler value Used for programming the prescaler for dividing the system clock to provide the smartcard clock. The value given in the register (5 significant bits) is multiplied by 2 to give the division factor of the source clock frequency: 00000:Reserved - do not program this value; 00001:divides the source clock by 2 00010:divides the source clock by 4 00011:divides the source clock by 6 …… Note:Bit [7:5] have no effect if Smartcard mode is used.

Puya Confidential

31. Debug support (DBG)

31.1. Overview

The device is built around a Cortex®-M0+ core which contains hardware extensions for advanced debugging features. The debug extensions allow the core to be stopped either on a given instruction fetch (breakpoint) or data access (watchpoint). When stopped, the core's internal state and the system's external state may be examined. Once examination is complete, the core and the system may be restored and program execution resumed.

The debug features are used by the debugger host when connecting to and debugging the MCUs. The debugging interface is serial wire. The debug features embedded in the Cortex®-M0+ core are a subset of the Arm® CoreSight Design Kit.

M0 + provides integrated on-chip debug support. It is comprised of:

- SW-DP:serial wire
- BPU:Break point unit
- DWT:Data watchpoint trigger

It also includes debug features dedicated to the device:

- Flexible debug pinout assignment, SWIO @ PA13, SWCLK @ PA14
- MCU debug box (support for low-power modes, control over peripheral clocks, etc.)

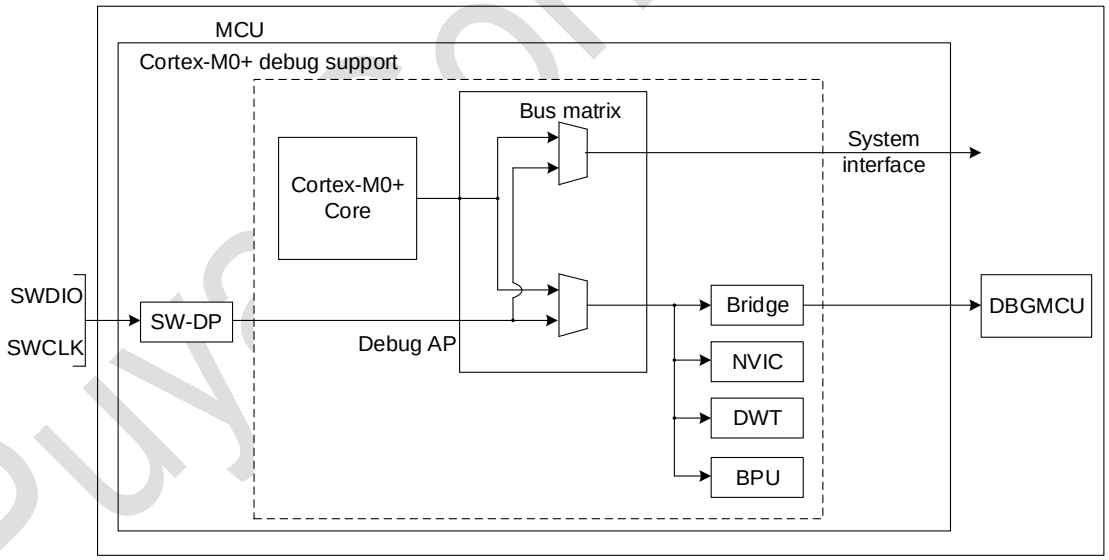


Figure 31-1 DBG block diagram

31.2. Pinout and debug port pins

31.2.1. SWD port

Two pins are related to debugging functions which are available on all packages.

Puya Confidential

Table 31-1 DBG block diagram

SW-DP pin name	SWD debug port pins		Pin assignment
	Type	Debug assignment	
SWDIO	I/O	Serial Wire Data Input/Output	PA13
SWCLK	I	Serial Wire Clock	PA14

31.2.2. Flexible SWJ-DP pin assignment

After RESET (SYSRESETn or PORESETn), all pins used for the SW-DP are assigned as dedicated pins immediately usable by the debugger host.

In addition, the MCU offers the possibility to disable the SWD port and can then release the associated pins for general-purpose I/O (GPIO) usage.

31.2.3. Internal pull-up & pull-down on SWD pins

Once the SW I/O is released by the user software, the GPIO controller takes control of these pins. The reset states of the GPIO control registers put the I/Os in the equivalent states:

- SWDIO:input pull-up
- SWCLK:input pull-down

Having embedded pull-ups and pull-downs removes the need to add external resistors.

31.3. ID codes and locking mechanism

There are several ID codes inside the MCU. It is strongly recommended the tool manufacturers (for example Keil and IAR) to lock their debugger using the MCU device ID located at address 0x40 015 800.

After the device is powered on, the hardware reads the 0x1FFF33F0 address of the reserved area of Flash and loads it into the DBG_IDCODE register.

31.4. SWD port

31.4.1. SWD protocol introduction

This synchronous serial protocol uses two pins:

- SWCLK:clock from host to target
- SWDIO:bidirectional

The protocol allows two banks of registers (DPACC registers and APACC registers) to be read and written to. Bits are transferred LSB-first on the wire. For SWDIO bidirectional management, the line must be pulled-up on the board (100 kΩ recommended by Arm).

Each time the direction of SWDIO changes in the protocol, a turnaround time is inserted where the line is not driven by the host nor the target. By default, this turnaround time is one bit time, however this can be adjusted by configuring the SWCLK frequency.

31.4.2. SWD protocol sequence

Each sequence consists of three phases:

Puya Confidential

- Packet request (8 bits) transmitted by the host
- Acknowledge response (3 bits) transmitted by the target
- Data transfer phase (33 bits) transmitted by the host or the target

Table 31-2 Request packet (8bits)

Bit	Pin name	Description
0	Start	Must be “1”
1	ApnDP	0:DP Access 1:AP Access
2	RnW	0:Write Request 1:Read request
4:3	A[3:2]	Address field of the DP or AP registers
5	Parity	Single bit parity of preceding bits
6	Stop	0
7	Park	Not driven by the host. Must be read as “1” by the target because of the pull-up

The packet request is always followed by the turnaround time (default 1 bit) where neither the host nor target drive the line.

Table 31-3 ACK response (3bits)

Bit	Pin name	Description
[2:0]	ACK	001:FAULT 010:WAIT 100:OK

The ACK Response must be followed by a turnaround time only if it is a READ transaction or if a WAIT or FAULT acknowledge has been received.

Table 31-4 DATA transfer (33bits)

Bit	Pin name	Description
[31:0]	WDATA or RDATA	Write or Read data
32	Parity	Single parity of the 32 data bits

The DATA transfer must be followed by a turnaround time only if it is a READ transaction.

31.4.3. SW-DP state machine (reset, idle states, ID code)

The State Machine of the SW-DP has an internal ID code which identifies the SW-DP. It follows the JEP-106 standard. This ID code is the default Arm one and is set to 0x0BB11477 (corresponding to Cortex®-M0+).

31.4.4. DP and AP read/write accesses

- Read accesses to the DP are not posted:the target response can be immediate (if ACK=OK) or can be delayed (if ACK=WAIT).
- Read accesses to the AP are posted. This means that the result of the access is returned on the next transfer. If the next access to be done is NOT an AP access, then the DP-RDBUFF register must be read to obtain the result.

Puya Confidential

The READOK flag of the DP-CTRL/STAT register is updated on every AP read access or RDBUFF read request to know if the AP read access was successful.

- The SW-DP implements a write buffer (for both DP or AP writes), that enables it to accept a write operation even when other transactions are still outstanding. If the write buffer is full, the target acknowledge response is WAIT. With the exception of IDCODE read or CTRL/STAT read or ABORT write which are accepted even if the write buffer is full.
- Because of the asynchronous clock domains SWCLK and HCLK, two extra SWCLK cycles are needed after a write transaction (after the parity bit) to make the write effective internally. These cycles should be applied while driving the line low.

This is particularly important when writing the CTRL/STAT for a power-on request. If the next transaction (requiring a power-on) occurs immediately, it will fail.

31.4.5. SW-DP registers

Access to these registers are initiated when APnDP=0

A [3:2]	R/W	CTRLSEL bit or SELECT registers	Register
00	Read		IDCODE
00	Write		ABORT
01	Read/Write	0	DP-CTRL/STAT
01	Read/Write	1	WIRECONTROL
10	Read		READRESEND
10	Write		SELECT
11	Read/Write		READBUFFER

31.4.6. SW-AP registers

Address	A [3:2]	Description
0x0	00	Reserved
0x4	01	DP CTRL/STAT register, used as ■ Request a system or debug power-up ■ Configure the transfer operation for AP accesses ■ Control the pushed compare and pushed verify operations ■ Read some status flags (overflow, power-up acknowledges)
0x8	10	DP SELECTRION register:Used to select the current access port and the active 4-words register window. ■ Bits 31:24:APSEL:select the current AP ■ Bits 23:8:Reserved ■ Bits 7:4:APBANKSEL:select the active 4-words register window on the current AP ■ Bits 3:0:Reserved
0xC	11	DP RDBUFF register:Used to allow the debugger to get the final result after a sequence of operations (without requesting new JTAG-DP operation)

Puya Confidential

31.5. Core debug

Core debug is accessed through the core debug registers. Debug access to these registers is by means of the debug access port. It consists of the following four registers

Table 31-5 Core debug registers

Register	Description
DHCSR	The 32-bit Debug halting control and status register
DCRSR	The 17-bit Debug core register selector register
DHRDR	The 32-bit Debug core register data register
DEMCR	The 32-bit Debug exception and monitor control register

These registers are not reset by a system reset. They can only be reset by power-on reset, to halt on reset, it is necessary to:

- Enable the bit0 (VC_CORRESET) of the Debug exception and monitor control register.
- Enable the bit0 (C_DEBUGEN) of the Debug halting control and status register

31.6. Breakpoint unit (BPU)

The Cortex®-M0+ BPU implementation provides four breakpoint registers. The BPU is a set of the Flash Patch and Breakpoint (FPB) block available in ARMv7-M (Cortex-M3 &Cortex-M4).

31.6.1. BPU functionality

The processor breakpoints implement PC based breakpoint functionality.

Refer to the ARMv6-M ARM and ARM coresight components technical reference manual for more information on BPU Coresight's identity registers and their addresses and access kinds.

31.7. Data watchpoint (DWT)

The Cortex-M0 DWT provides 2 watchpoint registers.

31.7.1. DWT functionality

The processor watchpoints implement PC based watchpoint functionality.

31.7.2. DWT program counter sample register

A processor that implements the data watchpoint unit also implements the ARMv6-M optional DWT program counter sample register (DWT_PCSR). This register permits a debugger to periodically sample the PC without halting the processor. This provides coarse grained profiling.

The Cortex®-M0+ DWT_PCSR records both instructions that pass their condition codes and those that fail.

31.8. MCU debug component (DBG)

The MCU debug component helps the debugger provide support for:

- Low-power modes
- Clock control for timers, watchdog and I^2C during a breakpoint

Puya Confidential

- Control of tracking pin assignment

The MCUDBG register also provides chip ID encoding. This ID encoding can be accessed by a JTAG or SW debug interface, or by a user program.

31.8.1. Debug support for low-power modes

To enter low-power mode, the instruction WFI or WFE must be executed. The MCU implements several low-power modes which can either deactivate the CPU clock or reduce the power of the CPU.

The core does not allow FCLK or HCLK to be turned off during a debug session. As these are required for the debugger connection, during a debug, they must remain active. The MCU integrates special means to allow the user to debug software in low-power modes.

For this, the debugger host must first set some debug configuration registers to change the low-power mode behavior:

- In Sleep mode:FCLK and HCLK are still active. Consequently, this mode does not impose any restrictions on the standard debug features.
- In Stop mode, the DBG_STOP bit must be previously set by the debugger.

31.8.2. Debug support for timers, watchdog and I²C

During a breakpoint, it is necessary to choose how the counter of timers and watchdog should behave:

- They can continue to count inside a breakpoint. This is usually required when a PWM is controlling a motor, for example.
- They can stop to count inside a breakpoint. This is required for watchdog purposes.

For the I²C, the user can choose to block the SMBUS timeout during a breakpoint.

31.9. DBG register

31.9.1. DBG device ID code register (DBG_IDCODE)

Address offset:0x00

Only 32-bit access supported. Read-only

This code is accessible by the software debug port (two pins) or by the user software.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBG_IDCODE [31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBG_IDCODE [16:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Reset Value	Function
31:0	DBG_IDCODE [31:0]	R	0x06188061	Revision ID

Puya Confidential

31.9.2. DBG configuration register (DBG_CR)

This register configures the low-power modes of the MCU under debug.

It is asynchronously reset by the POR (and not the system reset). It can be written by the debugger under system reset.

If the debugger host does not support this feature, it is still possible for the user software to write to this register.

Address offset:0x04

Reset value:0x0000 0000 (not reset by system reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
-															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBG_STOP	DBG_SLEEP
-														RW	RW

Bit	Name	R/W	Reset Value	Function
31:2	Reserved	-		Reserved
1	DBG_STOP	RW	0	Debug stop mode. 0:(FCLK = off, HCLK = off). In Stop mode, both HCLK and FCLK are off. When exiting from Stop mode, the clock configuration is identical to the one after RESET (CPU clocked by the internal RC oscillator (HSI)). Consequently, the software must reprogram the clock controller. 1:(FCLK = on, HCLK = on). In this case, when entering Stop mode, FCLK and HCLK are provided by the internal RC oscillator (HSI). When exiting Stop mode, the software must reprogram the clock controller if the clock control needs to be changed.
0	DBG_SLEEP	RW	0	Debug sleep mode. 0:(FCLK ON, HCLK OFF). In Sleep mode, FCLK is provided by the previously configured system clock, and HCLK is turned off. Since sleep mode does not reset the configured clock system, the software does not need to reconfigure the clock after exiting Debug sleep mode. 1:(FCLK=On, HCLK=On). In SLEEP mode, both the FCLK and HCLK clocks are provided by the previously configured system clock.

Puya Confidential

31.9.3. DBG APB freeze register 1 (DBG_APB_FZ1)

This register configures the clocking of timers, RTC, IWDG and WWDG of the MCU under debug. The register is asynchronously reset by the POR (and not the system reset). It can be written by the debugger under system reset.

Address offset:0x08

Power on reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBG_LPTIM_STOP	Res								DBG_I2C2_SMBUS_TIMEOUT	DBG_I2C1_SMBUS_TIMEOUT	Res				
RW	-								R	RW	-				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res			DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res				DBG_TIM7_STOP	DBG_TIM6_STOP	Res		DBG_TIM3_STOP	DBG_TIM2_STOP
-			RW	RW	RW	-				RW	RW	-		RW	RW

Bit	Name	R/W	Reset Value	Function
31	DBG_LPTIM_STOP	RW	0	Clocking of LPTIM counter when the core is halted 0:Enable 1:Disable
30:23	Reserved	-	-	Reserved
22	DBG_I2C2_SMBUS_TIMEOUT	R	0	Fixed at 0
21	DBG_I2C1_SMBUS_TIMEOUT	RW	0	SMBUS timeout when core is halted 0:same behavior as in normal mode; 1:SMBUS timeout is frozen.
20:13	Reserved	-	-	Reserved
12	DBG_IWDG_STOP	RW	0	Clocking of IWDG counter when the core is halted 0:Enable 1:Disable
11	DBG_WWDG_STOP	RW	0	Clocking of WWDG counter when the core is halted 0:Enable 1:Disable
10	DBG_RTC_STOP	RW	0	Clocking of RTC counter when the core is halted

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				0:Enable 1:Disable
9:6	Reserved	-		Reserved
5	DBG_TIM7_STOP	RW	0	Clocking of TIM7 counter when the core is halted 0:clock enabled; 1:Clock disabled;
4	DBG_TIM6_STOP	RW	0	Clocking of TIM6 counter when the core is halted 0:clock enabled; 1:Clock disabled;
3:2	Reserved	-		Reserved
1	DBG_TIM3_STOP	RW	0	Clocking of TIM3 counter when the core is halted 0:Enable 1:Disable
0	DBG_TIM2_STOP	RW	0	Clocking of TIM2 counter when the core is halted 0:clock enabled; 1:Clock disabled;

31.9.4. DBG APB freeze register 2 (DBG_APB_FZ2)

This register configures the clocking of timer counters when the MCU is under debug. The register is asynchronously reset by the POR (and not the system reset). It can be written by the debugger under system reset.

Address offset:0x0C

Power on reset value:0x0000 0000

Only 32-bit access supported. Read-only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res													DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP
-													RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBG_TIM14_STOP	Res			DBG_TIM1_STOP	Res										
RW	-			RW	-										

Bit	Name	R/W	Reset Value	Function
31:19	Reserved	-		Reserved
18	DBG_TIM17_STOP	RW	0	Clocking of TIM17 counter when the core is halted 0:Enable

Puya Confidential

Bit	Name	R/W	Reset Value	Function
				1:Disable
17	DBG_TIM16_STOP	RW	0	Clocking of TIM16 counter when the core is halted 0:Enable 1:Disable
16	DBG_TIM15_STOP	RW	0	Clocking of TIM15 counter when the core is halted 0:clock enabled 1:Clock disabled
15	DBG_TIM14_STOP	RW	0	Clocking of TIM14 counter when the core is halted 0:Enable 1:Disable
14:12	Reserved	-		Reserved
11	DBG_TIM1_STOP	RW	0	Clocking of TIM1 counter when the core is halted 0:Enable 1:Disable
10:0	Reserved	-		Reserved

Puya Confidential

32. Revision history

Version	Date	Descriptions
V0.2	2025.06.10	Initial version
V0.3	2025.07.03	1.LSI frequency adjustable: 32.768/23.552/38.912/39.168/43.008 kHz 2.Supports 8 * 32/4 * 40 LCD (Only external resistor mode can be used at 1/6 or 1/8 duty)
V0.5	2025.09.25	Translated and fixed known issues

Puya Confidential

Puya Semiconductor Co., Ltd.

IMPORTANT NOTICE

Puya reserve the right to make changes, corrections, enhancements, modifications to Puya products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information of Puya products before placing orders.

Puya products are sold pursuant to terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice and use of Puya products. Puya does not provide service support and assumes no responsibility when products that are used on its own or designated third party products.

Puya hereby disclaims any license to any intellectual property rights, express or implied.

Resale of Puya products with provisions inconsistent with the information set forth herein shall void any warranty granted by Puya.

Any with Puya or Puya logo are trademarks of Puya. All other product or service names are the property of their respective owners.

The information in this document supersedes and replaces the information in the previous version.

Puya Semiconductor Co., Ltd. – All rights reserved