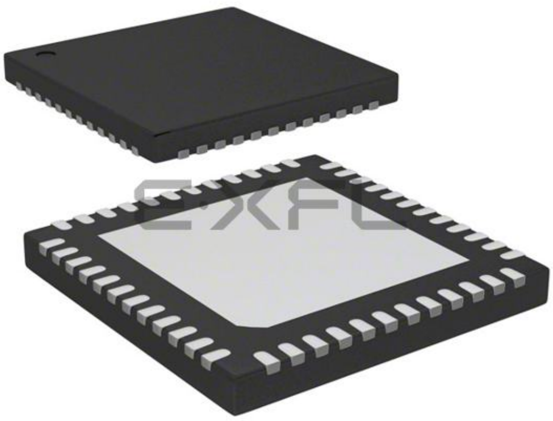


Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	RF4CE, Remote Control
Core Processor	ARM® Cortex®-M3
Program Memory Type	FLASH (256kB)
Controller Series	STM32W
RAM Size	16K x 8
Interface	I²C, SPI, UART/USART
Number of I/O	24
Voltage - Supply	1.18V ~ 3.6V
Operating Temperature	-40°C ~ 105°C
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/stm32w108ccu73tr

List of tables

Table 2. Pin descriptions (continued)

48-Pin Package Pin no.	40-Pin Package Pin no.	Signal	Direction	Description
31	26	PB2	I/O	Digital I/O
		SC1MISO	I	SPI master data in of Serial Controller 1 Select SPI with SC1_CR Select master with SC1_SPICR
		SC1MOSI	I	SPI slave data in of Serial Controller 1 Select SPI with SC1_CR Select slave with SC1_SPICR
		SC1SCL	I/O	I^2C clock of Serial Controller 1 Either disable timer output in TIM2_CCER, or disable remap with TIM2_OR[5] Select I^2C with SC1_CR Select alternate open-drain output function with GPIOB_CRL[11:8]
		SC1RXD	I	UART receive data of Serial Controller 1 Select UART with SC1_CR
		TIM2_CH2 (see also Pin 25)	O	Timer 2 channel 2 output Enable remap with TIM2_OR[5] Enable timer output in TIM2_CCER Select alternate output function with GPIOB_CRL[11:8]
			I	Timer 2 channel 2 input. Enable remap with TIM2_OR[5].
32	27	SWCLK	I/O	Serial Wire clock input/output with debugger Selected when in Serial Wire mode (see JTMS description, Pin 35)
		JTCK	I	JTAG clock input from debugger Selected when in JTAG mode (default mode, see JTMS description, Pin 35) Internal pull-down is enabled
33	28	PC2	I/O	Digital I/O Enable with GPIO_DBGCR[5]
		JTDO	O	JTAG data out to debugger Selected when in JTAG mode (default mode, see JTMS description, Pin 35)
		SWO	O	Serial Wire Output asynchronous trace output to debugger Select asynchronous trace interface in ARM core Enable trace interface in ARM core Select alternate output function with GPIOC_CRL[11:8] Enable Serial Wire mode (see JTMS description, Pin 35) Internal pull-up is enabled

6.2.4 Reset register

Reset status register (RST_SR)

Address offset: 0x4000 002C
Reset value: 0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LKUP	OBFAIL	WKUP	SWRST	WDG	PIN	PWRLV	PWRHV
								r	r	r	r	r	r	r	r

Bits 31:8 Reserved, must be kept at reset value

Bit 7 LKUP:
When set to ‘1’, the reset is due to core lockup.

Bit 6 OBFAIL:
When set to ‘1’, the reset is due to an Option byte load failure (may be set with other bits).

Bit 5 WKUP:
When set to ‘1’, the reset is due to a wake-up from deep sleep.

Bit 4 SWRST:
When set to ‘1’, the reset is due to a software reset.

Bit 3 WDG:
When set to ‘1’, the reset is due to watchdog expiration.

Bit 2 PIN:
When set to ‘1’, the reset is due to an external reset pin signal.

Bit 1 PWRLV:
When set to ‘1’, the reset is due to the application of a Core power supply (or previously failed).

Bit 0 PWRHV:
Always set to ‘1’, Normal power applied.

Output mode

Output mode provides a general purpose output under direct software control. Regardless of whether an output is configured as push-pull or open-drain, the GPIO's bit in the GPIOx_ODR register controls the output. The GPIOx_BSR and GPIOx_BRR registers can atomically set and clear bits within GPIOx_ODR register. These set and clear registers simplify software using the output port because they eliminate the need to disable interrupts to perform an atomic read-modify-write operation of GPIOx_ODR.

When configured in output mode:

- The output drivers are enabled and are controlled by the value written to GPIOx_ODR:
- In open-drain mode: 0 activates the N-MOS current sink; 1 tri-states the pin.
- In push-pull mode: 0 activates the N-MOS current sink; 1 activates the P-MOS current source.
- The internal pull-up and pull-down resistors are disabled.
- The Schmitt trigger input is connected to the pin.
- Reading GPIOx_IDR returns the input at the pin.
- Reading GPIOx_ODR returns the last value written to the register.

Note: Depending on configuration and usage, GPIOx_ODR and GPIOx_IDR may not have the same value.

Alternate output mode

In this mode, the output is controlled by an on-chip peripheral instead of GPIOx_ODR and may be configured as either push-pull or open-drain. Most peripherals require a particular output type - I^2C requires an open-drain driver, for example - but since using a peripheral does not by itself configure a pin, the GPIOx_CRH/L registers must be configured properly for a peripheral's particular needs. As described in *Section 8.1.2: Configuration on page 94*, when more than one peripheral can be the source of output data, registers in addition to GPIOx_CRH/L determine which to use.

When configured in alternate output mode:

- The output drivers are enabled and are controlled by the output of an on-chip peripheral:
- In open-drain mode: 0 activates the N-MOS current sink; 1 tri-states the pin.
- In push-pull mode: 0 activates the N-MOS current sink; 1 activates the P-MOS current source.
- The internal pull-up and pull-down resistors are disabled.
- The Schmitt trigger input is connected to the pin.
- Reading GPIOx_IDR returns the input to the pin.

Note: Depending on configuration and usage, GPIOx_ODR and GPIOx_IDR may not have the same value.

Alternate output SPI SCLK mode

SPI master mode SCLK outputs, PB3 (SC1SCLK) or PA2 (SC2SCLK), use a special output push-pull mode reserved for those signals. Otherwise this mode is identical to alternate output mode.

8.1.7 Wake monitoring

The PWR_WAKEPxR registers specify which GPIOs are monitored to wake the processor. If a GPIO's wake enable bit is set in PWR_WAKEPxR, then a change in the logic value of that GPIO causes the STM32W108xx to wake from deep sleep. The logic values of all GPIOs are captured by hardware upon entering sleep. If any GPIO's logic value changes while in sleep and that GPIO's PWR_WAKEPxR bit is set, then the STM32W108xx will wake from deep sleep. (There is no mechanism for selecting a specific rising-edge, falling-edge, or level on a GPIO: any change in logic value triggers a wake event.) Hardware records the fact that GPIO activity caused a wake event, but not which specific GPIO was responsible. Instead, software should read the state of the GPIOs on waking to determine the cause of the event.

The register PWR_WAKEFILTR contains bits to enable digital filtering of the external wakeup event sources: the GPIO pins, SC1 activity, SC2 activity, and IRQD. The digital filter operates by taking samples based on the (nominal) 10 kHz LSI RC oscillator. If three samples in a row all have the same logic value, and this sampled logic value is different from the logic value seen upon entering sleep, the filter outputs a wakeup event.

In order to use GPIO pins to wake the STM32W108xx from deep sleep, the GPIO_SEL bit in the EXTIx_CR register must be set. Waking up from GPIO activity does not work with pins configured for analog mode since the digital logic input is always set to 1 when in analog mode. Refer to *Section 6: System modules on page 46* for information on the STM32W108xx's power management and sleep modes.

8.2 External interrupts

The STM32W108xx can use up to four external interrupt sources (IRQA, IRQB, IRQC, and IRQD), each with its own top level NVIC interrupt vector. Since these external interrupt sources connect to the standard GPIO input path, an external interrupt pin may simultaneously be used by a peripheral device or even configured as an output. Analog mode is the only GPIO configuration that is not compatible with using a pin as an external interrupt.

External interrupts have individual triggering and filtering options selected using the registers EXTIA_TSR, EXTIB_TSR, EXTIC_TSR, and EXTID_TSR. The bit field INTMOD of the EXTIx_TSR register enables IRQx's second level interrupt and selects the triggering mode: 0 is disabled; 1 for rising edge; 2 for falling edge; 3 for both edges; 4 for active high level; 5 for active low level. The minimum width needed to latch an unfiltered external interrupt in both level- and edge-triggered mode is 80 ns. With the digital filter enabled (the FILTEN bit in the EXTIx_TSR register is set), the minimum width needed is 450 ns.

The register EXTI_PR is the second-level interrupt flag register that indicates pending external interrupts. Writing 1 to a bit in the EXTI_PR register clears the flag while writing 0 has no effect. If the interrupt is level-triggered, the flag bit is set again immediately after being cleared if its input is still in the active state.

Two of the four external interrupts, IRQA and IRQB, have fixed pin assignments. The other two external interrupts, IRQC and IRQD, can use any GPIO pin. The EXTIC_CR and EXTID_CR registers specify the GPIO pins assigned to IRQC and IRQD, respectively. *Table 16* shows how the EXTIC_CR and EXTID_CR register values select the GPIO pin used for the external interrupt.

The SPI master controller uses the three signals:

- MOSI (Master Out, Slave In) - outputs serial data from the master
- MISO (Master In, Slave Out) - inputs serial data from a slave
- SCLK (Serial Clock) - outputs the serial clock used by MOSI and MISO

The GPIO pins used for these signals are shown in *Table 21*. Additional outputs may be needed to drive the nSSEL signals on slave devices.

Table 21. SPI master GPIO usage

Parameter	MOSI	MISO	SCLK
Direction	Output	Input	Output
GPIO configuration	Alternate Output (push-pull)	Input	Alternate Output (push-pull) Special SCLK mode
SC1 pin	PB1	PB2	PB3
SC2 pin	PA0	PA1	PA2

9.3.1 Setup and configuration

Both serial controllers, SC1 and SC2, support SPI master mode. SPI master mode is enabled by the following register settings:

- The serial controller mode register (SCx_CR) is ‘2’.
- The MSTR bit in the SPI configuration register (SCx_SPICR) is ‘1’.
- The ACK bit in the I^2C control register (SCx_I2CCR2) is ‘1’.

The SPI serial clock (SCLK) is produced by a programmable clock generator. The serial clock is produced by dividing down 12 MHz according to this equation:

$$Rate = \frac{12MHz}{(LIN + 1)x2^{EXP}}$$

EXP is the value written to the SCx_CRR2 register and LIN is the value written to the SCx_CRR1 register. The SPI master mode clock may not exceed 6 Mbps, so EXP and LIN cannot both be zero.

The SPI master controller supports various frame formats depending upon the clock polarity (CPOL), clock phase (CPHA), and direction of data (LSBFIRST) (see *SPI master mode formats on page 114*). The bits CPOL, CPHA, and LSBFIRST are defined within the SCx_SPICR register.

9.5.3 Interrupts

I^2C master controller interrupts are generated on the following events:

- Bus command (START/STOP) completed (0 to 1 transition of CMDFIN)
- Character transmitted and slave device responded with NACK
- Character transmitted (0 to 1 transition of BTF)
- Character received (0 to 1 transition of BRF)
- Received and lost character while receive FIFO was full (receive overrun error)
- Transmitted character while transmit FIFO was empty (transmit underrun error)

To enable CPU interrupts, set the desired interrupt bits in the second level SCx_IER register.

9.6 Universal asynchronous receiver/transmitter (UART)

The SC1 UART is enabled by writing 1 to SC1_CR. The SC2 serial controller does not include UART functions.

The UART supports the following features:

- Flexible baud rate clock (300 bps to 921.6 bps)
- Data bits (7 or 8)
- Parity bits (none, odd, or even)
- Stop bits (1 or 2)
- False start bit and noise filtering
- Receive and transmit FIFOs
- Optional RTS/CTS flow control
- Receive and transmit DMA channels

The UART uses two signals to transmit and receive serial data:

- TXD (Transmitted Data) - serial data received by the STM32W108xx
- RXD (Received Data) - serial data sent by the STM32W108xx

If RTS/CTS flow control is enabled, these two signals are also used:

- nRTS (Request To Send) - indicates the STM32W108xx is able to receive data RXD
- nCTS (Clear To Send) - inhibits sending data from the STM32W108xx if not asserted

The GPIO pins assigned to these signals are shown in *Table 28*.

Table 28. UART GPIO usage

Parameter	TXD	RXD	nCTS(1)	nRTS(1)
Direction	Output	Input	Input	Output
GPIO configuration	Alternate Output (push-pull)	Input	Input	Alternate Output (push-pull)
SC1 pin	PB1	PB2	PB3	PB4

1. Only used if RTS/CTS hardware flow control is enabled.

9.8 Serial controller common registers

9.8.1 Serial controller interrupt status register (SCx_ISR)

Address offset: 0xA808 (SC1_ISR) and 0xA80C (SC2_ISR)
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PE	FE	TXUL ODB	TXUL ODA	RXUL ODB	RXUL ODA	NACK	CMD FIN	BTF	BRF	UDR	OVR	IDLE	TXE	RXNE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31:15 Reserved, must be kept at reset value

Bit 14 PE: Parity error pending interrupt
This bit is set by hardware when a parity error occurs in receiver mode.
0: No parity error pending interrupt
1: Parity error pending interrupt
Note: Not used in SC2

Bit 13 FE: Framing error pending interrupt
This bit is set by hardware when a desynchronization or excessive noise is detected.
0: No framing error detected pending interrupt
1: Framing error pending interrupt
Note: Not used in SC2

Bit 12 TXULODB: DMA transmit buffer B unloaded pending interrupt
This bit is set by hardware when DMA load error is detected during transmission.
0: No DMA transmit buffer B unloaded error pending interrupt
1: DMA transmit buffer B unloaded pending interrupt

Bit 11 TXULODA: DMA transmit buffer A unloaded pending interrupt
This bit is set by hardware when DMA load error is detected during transmission.
0: No DMA transmit buffer A unloaded error pending interrupt
1: DMA transmit buffer A unloaded error pending interrupt.

Bit 10 RXULODB: DMA receive buffer B unloaded pending interrupt
This bit is set by hardware when DMA load error is detected during reception.
0: No DMA receive buffer B unloaded error pending interrupt
1: DMA receive buffer B unloaded error pending interrupt

Bit 9 RXULODA: DMA receive buffer A unloaded pending interrupt
This bit is set by hardware when DMA load error is detected during reception.
0: No DMA receive buffer A unloaded error pending interrupt
1: DMA receive buffer A unloaded error pending interrupt

Bit 8 NACK: I^2C not acknowledge received pending interrupt
This bit is set by hardware when a NACK is received after a byte transmission.
0: No NACK detected pending interrupt
1: NACK detected pending interrupt

9.11 Serial controller: Universal asynchronous receiver/ transmitter (UART) registers

9.11.1 Serial controller UART status register (SC1_UARTSR)

Address offset: 0xC848
Reset value: 0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved									IDLE	PE	FE	OVR	TXE	RXNE	CTS
									r	r	r	r	r	r	r

Bits 31:7 Reserved, must be kept at reset value

Bit 6 IDLE: Idle line detected flag

This bit is set by hardware when both the transmit FIFO and the transmit serializer are empty. An interrupt is generated if IDLEIE=1 in the SCx_IER register.

0: No UART idle line is detected

1: UART idle line is detected

Bit 5 PE: Parity error flag

This bit is set when the byte in the data register is received with a parity error. This bit is updated when the data register is read, and is cleared if the receive FIFO is empty. An interrupt is generated if PEIE=1 in the SCx_IER register.

0: No UART parity error

1: UART parity error

Bit 4 FE: Frame error flag

This bit is set when the byte in the data register is received with a frame error. This bit is updated when the data register is read, and is cleared if the receive FIFO is empty. An interrupt is generated if FEIE=1 in the SCx_IER register.

0: No UART frame error

1: UART frame error

Bit 3 OVR: Overrun error flag

This bit is set when the receive FIFO has been overrun. This occurs if a byte is received when the receive FIFO is full. This bit is cleared by reading the data register. An interrupt is generated if OVRIE=1 in the SCx_IER register.

0: No overrun error occurred

1: Overrun error occurred

9.12.9 Serial controller receive DMA counter channel A register (SCx_DMARXCNTAR)

Address offset: 0xC820 (SC1_DMARXCNTAR) and 0xC020 (SC2_DMARXCNTAR)
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CNT[12:0]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:13 Reserved, must be kept at reset value

Bits 12:0 CNT[12:0]:
The offset from the start of DMA receive buffer A at which the next byte is written. This register is set to zero when the buffer is loaded and when the DMA is reset. If this register is written when the buffer is not loaded, the buffer is loaded.

9.12.10 Serial controller receive DMA count channel B register (SCx_DMARXCNTBR)

Address offset: 0xC824 (SC1_DMARXCNTBR) and 0xC024 (SC2_DMARXCNTBR)
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CNT[12:0]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:13 Reserved, must be kept at reset value

Bits 12:0 CNT[12:0]:
The offset from the start of DMA receive buffer B at which the next byte is written. This register is set to zero when the buffer is loaded and when the DMA is reset. If this register is written when the buffer is not loaded, the buffer is loaded.

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.

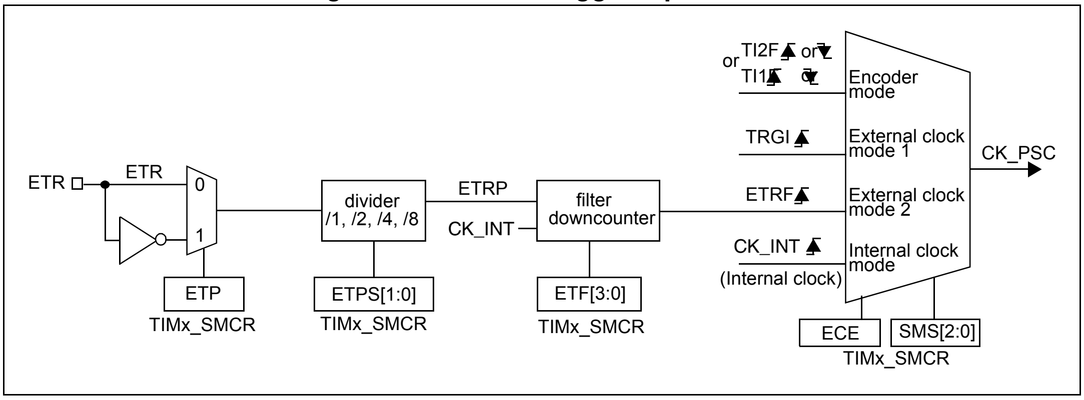
The EXTRIGSEL bits in the TIMx_OR register select a clock signal that drives ETR, as shown in *Table 33*.

Table 33. EXTRIGSEL clock signal selection

EXTRIGSEL bits	Clock signal selection
00	PCLK (peripheral clock). When running from the 24 MHz HSE OSC, the PCLK frequency is 12 MHz. When the 12 MHz HSI RC oscillator is in use, the frequency is 6 MHz.
01	Calibrated 1 kHz internal RC oscillator
10	Optional 32 kHz HSE OSC
11	TIMxCLK pin. If the CLKMSKEN bit in the TIMx_OR register is set, this signal is AND'ed with the TIMxMSK pin providing a gated clock input.

Figure 29 gives an overview of the external trigger input block.

Figure 29. External trigger input block



For example, to configure the up-counter to count each 2 rising edges on ETR, use the following procedure:

- As no filter is needed in this example, write ETF = 0000 in the TIMx_SMCR register.
- Set the prescaler by writing ETPS = 01 in the TIMx_SMCR register.
- Select rising edge detection on ETR by writing ETP = 0 in the TIMx_SMCR register.
- Enable external clock mode 2 by writing ECE = 1 in the TIMx_SMCR register.
- Enable the counter by writing CEN = 1 in the TIMx_CR1 register.

The counter counts once each 2 ETR rising edges.

The delay between the rising edge on ETR and the actual clock of the counter is due to the resynchronization circuit on the ETRP signal.

Table 34. Counting direction versus encoder signals

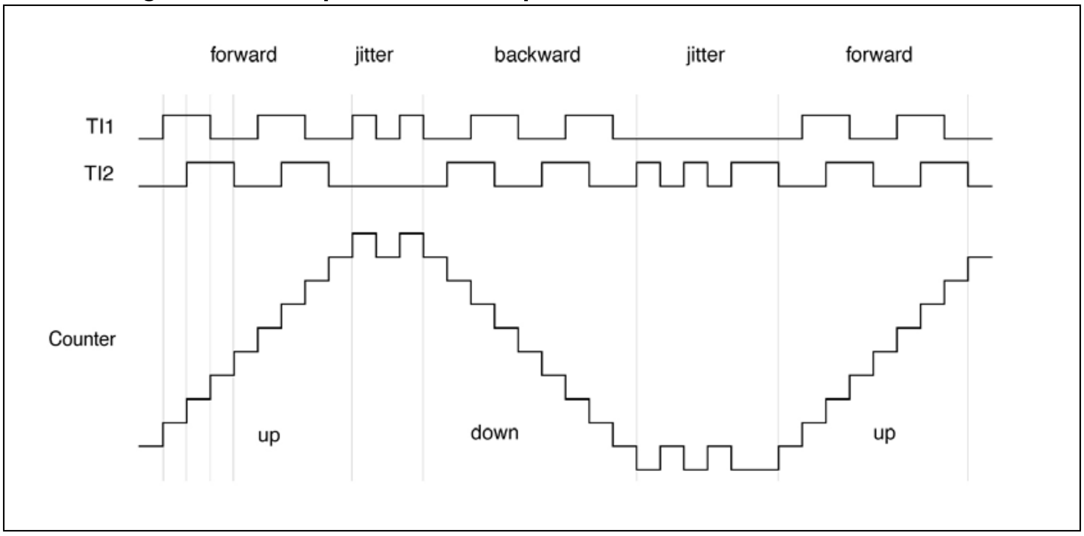
Active edges	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on TI2 only	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting 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 an encoder's differential outputs to digital signals, and this greatly increases noise immunity. If a third encoder output indicates the mechanical zero (or index) position, it may be connected to an external interrupt input and can trigger a counter reset.

Figure 39 gives an example of counter operation, showing count signal generation and direction control. It also shows how input jitter is compensated for when both inputs are used for counting. This might occur if the sensor is positioned near one of the switching points. This example assumes the following configuration:

- CC1S = 01 (TIMx_CCMR1 register, IC1FP1 mapped on TI1).
- CC2S = 01 (TIMx_CCMR2 register, IC2FP2 mapped on TI2).
- CC1P = 0 (TIMx_CCER register, IC1FP1 non-inverted, IC1FP1 = TI1).
- CC2P = 0 (TIMx_CCER register, IC2FP2 non-inverted, IC2FP2 = TI2).
- SMS = 011 (TIMx_SMCR register, both inputs are active on both rising and falling edges).
- CEN = 1 (TIMx_CR1 register, counter is enabled).

Figure 39. Example of counter operation in encoder interface mode



10.3.2 Timer *x* interrupt missed register (TIMx_MISSR)

Address offset: 0xA818 (TIM1) and 0xA81C (TIM2)
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4IM	CC3IM	CC2IM	CC1IM	Reserved		RSVD[6:0]						
			rw	rw	rw	rw			r	r	r	r	r	r	r

Bits 31:13] Reserved, must be kept at reset value

Bit 12 CC4IM: Capture or compare 4 interrupt missed

Bit 11 CC3IM: Capture or compare 3 interrupt missed

Bit 10 CC2IM: Capture or compare 2 interrupt missed

Bit 9 CC1IM: Capture or compare 1 interrupt missed

Bits 8:7] Reserved, must be kept at reset value

Bits 6:0] RSVD[6:0]: May change during normal operation

10.3.3 Timer *x* interrupt enable register (TIMx_IER)

Address offset: 0xA840 (TIM1) and 0xA844 (TIM2)
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TIE	Reserved	CC4IE	CC3IE	CC2IE	CC1IE	UIE
									rw		rw	rw	rw	rw	rw

Bits 31:7] Reserved, must be kept at reset value

Bit 6 TIE: Trigger interrupt enable

Bit 4 CC4IE: Capture or compare 4 interrupt enable

Bit 3 CC3IE: Capture or compare 3 interrupt enable

Bit 2 CC2IE: Capture or compare 2 interrupt enable

Bit 1 CC1IE: Capture or compare 1 interrupt enable

Bit 0 UIE: Update interrupt enable

Bit 1 UDIS: Update Disable

0: An update event is generated as soon as a counter overflow occurs, a software update is generated, or a hardware reset is generated by the slave mode controller. Shadow registers are then loaded with their buffer register values.

1: An update event is not generated and shadow registers keep their value (TIMx_ARR, TIMx_PSC, TIMx_CCRy). 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.

Bit 0 CEN: 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. Trigger mode sets the CEN bit automatically through hardware.

10.3.5 Timer *x* control register 2 (TIMx_CR2)

Address offset: 0xE004 (TIM1) and 0xF004 (TIM2)
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved								TI1S	MMS[2:0]			Reserved			
								rw	rw	rw	rw				

Bits 31:8 Reserved, must be kept at reset value

Bit 7 TI1S: TI1 Selection
0: TI1M (input of the digital filter) is connected to TI1 input.
1: TI1M is connected to the TI_HALL inputs (XOR combination).

Bits 6:4 MMS[2:0]: Master Mode Selection
This selects the information to be sent in master mode to a slave timer for synchronization using the trigger output (TRGO).
000: Reset - the UG bit in the TIMx_EGR register is trigger output.
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 - counter enable signal CNT_EN is trigger output.
This mode is used to start both timers at the same time or to control a window in which a slave timer is enabled. The counter enable signal is generated by either the CEN control bit or 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 - update event is trigger output
This mode allows a master timer to be 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 occurs.
100: Compare - OC1REF signal is trigger output
101: Compare - OC2REF signal is trigger output
110: Compare - OC3REF signal is trigger output
111: Compare - OC4REF signal is trigger output

Bits 3:0] Reserved, must be kept at reset value

14.4 SPI interface characteristics

Unless otherwise specified, the parameters given in Table 55 for the SPI are derived from tests performed under ambient temperature and V_{DD} supply voltage conditions summarized in Table 48: General operating conditions.

Table 55. SPI characteristics

Symbol	Parameter	Conditions	Min	Max[1]	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master mode	-	12	MHz
		Slave mode	-	5	
Duty(SCK)	Duty cycle of SPI clock frequency	Slave mode	30	70	%
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time	Capacitive load: C = 15 pF	-	8	ns
$t_{su(NSS)}$	NSS setup time	Slave mode	$2t_{PCLK}$	-	
$t_{h(NSS)}$	NSS hold time	Slave mode	$2t_{PCLK}$	-	
$t_{w(SCKH)}$ $t_{w(SCKL)}$	SCK high and low time	Master mode	$T_{CSCK}/2-2$	$T_{CSCK}/2+2$	
$t_{su(MI)}$ $t_{su(SI)}$	Data input setup time	Master mode	1	-	
		Slave mode	0	-	
$t_{h(MI)}$ $t_{h(SI)}$	Data input hold time	Master mode	1	-	
		Slave mode	0.25	-	
$t_{a(SO)}$[2]	Data output access time	Slave mode, f_{PCLK} = 12 MHz	0	$2t_{PCLK}$	
$t_{dis(SO)}$[3]	Data output disable time	Slave mode	0	36	
$t_{v(SO)}$[1] $t_{v(MO)}$[1]	Data output valid time	Slave mode (after enable edge)	-	22	
		Master mode (after enable edge)	-	14.25	
$t_{h(SO)}$[1] $t_{h(MO)}$[1]	Data output hold time	Slave mode (after enable edge)	15.6	-	
		Master mode (after enable edge)	0	-	

1. Based on characterization, not tested in production.
2. Min time is for the minimum time to drive the output and max time is for the maximum time to validate the data.
3. Min time is for the minimum time to invalidate the output and max time is for the maximum time to put the data in Hi-Z.

Characterization values are related to STM32W108CC.

14.6 Clock frequencies

14.6.1 High frequency internal clock characteristics

Table 60. High-frequency RC oscillator characteristics

Parameter	Test conditions	Min.	Typ.	Max.	Unit
Frequency at reset	-	6	12	20	MHz
Frequency Steps	-	-	0.5	-	MHz
Duty cycle	-	40	-	60	%
Supply dependence	Change in supply = 0.1 V	-	-	-	-
Test at supply changes: 1.8 V to 1.7 V	-	-	5	-	%

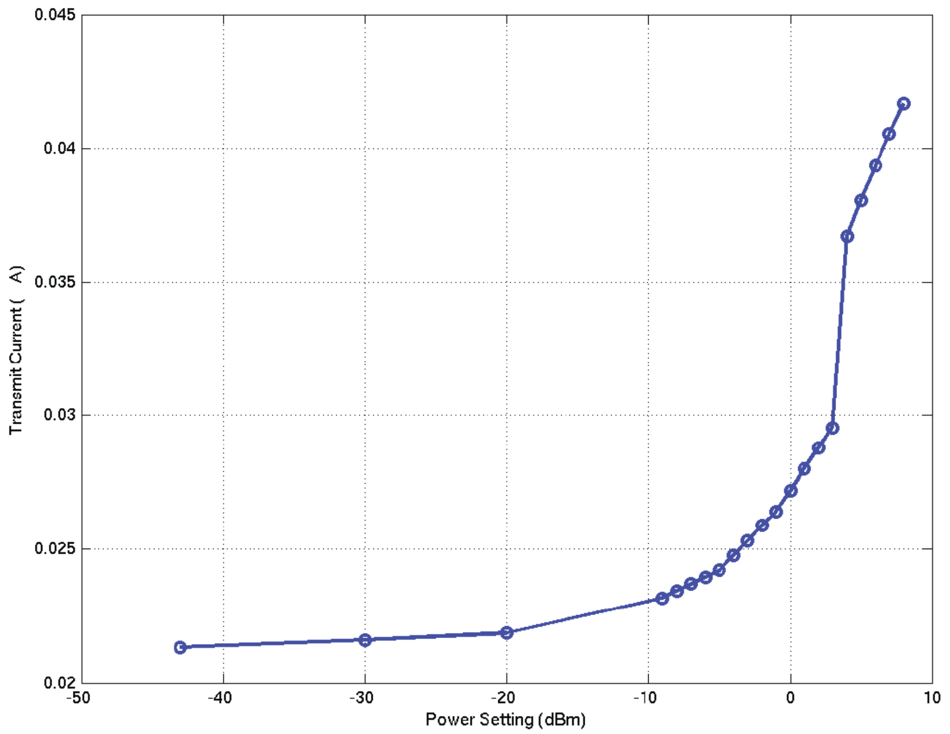
14.6.2 High frequency external clock characteristics

Table 61. High-frequency crystal oscillator characteristics

Parameter	Test conditions	Min.	Typ.	Max.	Unit
Frequency	–	–	24	–	MHz
Accuracy	–	-40	–	+40	ppm
Duty cycle	–	40	–	60	%
Phase noise (at 100 kHz offset)	–	–	–	-120	dBc/Hz
Start-up time at max bias	–	–	–	1	ms
Start up time at optimal bias	–	–	–	2	ms
Current consumption	–	–	200	300	µA
Current consumption at max bias	–	–	–	1	mA
Crystal with high ESR	–	–	–	100	Ω
– Load capacitance	–	–	–	10	pF
– Crystal capacitance	–	–	–	7	pF
– Crystal power dissipation	–	–	–	200	µW
Crystal with low ESR	–	–	–	60	Ω
– Load capacitance	–	–	–	18	pF
– Crystal capacitance	–	–	–	7	pF
– Crystal power dissipation	–	–	–	1	mW

Figure 58 shows the variation of current in Transmit mode (with the ARM® Cortex®-M3 running at 12 MHz).

Figure 58. Transmit power consumption



16 Ordering information scheme

Example: STM32 W 108 C B U 6 x

Device family

STM32 = ARM®-based 32-bit microcontroller

Product type

W = wireless system-on-chip

Sub-family

108 = IEEE 802.15.4 specification

Pin count

H = 40 pins
C = 48 pins

Code size

B = 128 Kbyte of Flash memory

C = 256 Kbyte of Flash memory

Z = 192 Kbyte of Flash memory

Package

U = FQFPN

Temperature range

6 = –40 °C to +85 °C

7 = –40 °C to +105 °C

Enabled protocol stack

“Blank” = Development sample platform (1)

1 = Ember ZigBee stack(2)

3 = RF4CE stack

4 = IEEE 802.15.4 media access control

1. This P/N is under specific ordering conditions. Please refer to your nearest ST sales office.
2. The Ember ZigBee stack is available on 128 Kbyte devices only.