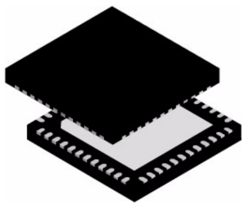
STM32W108C8

High-performance, IEEE 802.15.4 wireless system-on-chip with 64-Kbyte Flash memory

Datasheet - production data

Features

- Complete system-on-chip
  - 32-bit ARM® Cortex™-M3 processor
  - 2.4 GHz IEEE 802.15.4 transceiver & lower MAC
  - 8-Kbyte RAM and 64-Kbyte Flash memory
  - AES128 encryption accelerator
  - Flexible ADC, SPI/UART/I^{2}C serial communications, and general-purpose timers
  - 24 highly configurable GPIOs with Schmitt trigger inputs
- Industry-leading ARM® Cortex™-M3 processor
  - Leading 32-bit processing performance
  - Highly efficient Thumb®-2 instruction set
  - Operation at 6, 12 or 24 MHz
  - Flexible nested vectored interrupt controller
- Low power consumption, advanced management
  - Receive current (w/ CPU): 27 mA
  - Transmit current (w/ CPU, +3 dBm TX): 31 mA
  - Low deep sleep current, with retained RAM and GPIO: 400 nA/800 nA with/without sleep timer
  - Low-frequency internal RC oscillator for low-power sleep timing
  - High-frequency internal RC oscillator for fast (100 µs) processor start-up from sleep
- Exceptional RF performance
  - Normal mode link budget up to 102 dB; configurable up to 107 dB
  - -99 dBm normal RX sensitivity; configurable to -100 dBm (1% PER, 20 byte packet)
  - +3 dB normal mode output power; configurable up to +8 dBm
  - Robust WiFi and Bluetooth coexistence


VFQFPN48
(7 x 7 mm)

- Innovative network and processor debug
  - Non-intrusive hardware packet trace
  - Serial wire/JTAG interface
  - Standard ARM debug capabilities: Flash patch & breakpoint; data watchpoint & trace; instrumentation trace macrocell
- Application flexibility
  - Single voltage operation: 2.1-3.6 V with internal 1.8 V and 1.25 V regulators
  - Optional 32.768 kHz crystal for higher timer accuracy
  - Low external component count with single 24 MHz crystal
  - Support for external power amplifier
  - Small 7x7 mm 48-pin VFQFPN package

Applications

- RF4CE products and remote controls
- 6LoWPAN and custom protocols
- 802.15.4 based network protocols (standard and proprietary)

Contents

List of tables

List of figures

1 Description

The STM32W108 is a fully integrated System-on-Chip that integrates a 2.4 GHz, IEEE 802.15.4-compliant transceiver, 32-bit ARM® Cortex™-M3 microprocessor, Flash and RAM memory, and peripherals of use to designers of 802.15.4-based systems.

The transceiver utilizes an efficient architecture that exceeds the dynamic range requirements imposed by the IEEE 802.15.4-2003 standard by over 15 dB. The integrated receive channel filtering allows for robust co-existence with other communication standards in the 2.4 GHz spectrum, such as IEEE 802.11 and Bluetooth. The integrated regulator, VCO, loop filter, and power amplifier keep the external component count low. An optional high performance radio mode (boost mode) is software-selectable to boost dynamic range.

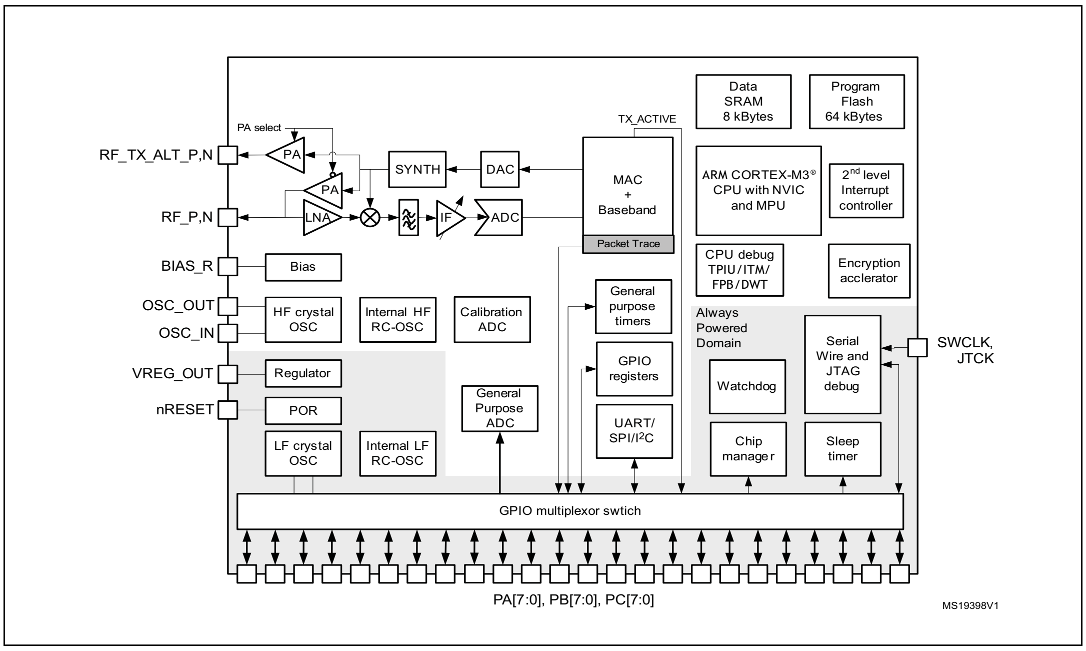
The integrated 32-bit ARM® Cortex™-M3 microprocessor is highly optimized for high performance, low power consumption, and efficient memory utilization. Including an integrated MPU, it supports two different modes of operation: Privileged mode and Unprivileged mode. This architecture could be used to separate the networking stack from the application code and prevent unwanted modification of restricted areas of memory and registers resulting in increased stability and reliability of deployed solutions.

The STM32W108 has 64 Kbytes of embedded Flash memory and 8 Kbytes of integrated RAM for data and program storage. The STM32W108 HAL software employs an effective wear-leveling algorithm that optimizes the lifetime of the embedded Flash.

To maintain the strict timing requirements imposed by the IEEE 802.15.4-2003 standards, the STM32W108 integrates a number of MAC functions into the hardware. The MAC hardware handles automatic ACK transmission and reception, automatic backoff delay, and clear channel assessment for transmission, as well as automatic filtering of received packets. A packet trace interface is also integrated with the MAC, allowing complete, non-intrusive capture of all packets to and from the STM32W108.

The STM32W108 offers a number of advanced power management features that enable long battery life. A high-frequency internal RC oscillator allows the processor core to begin code execution quickly upon waking. Various deep sleep modes are available with less than 1 µA power consumption while retaining RAM contents. To support user-defined applications, on-chip peripherals include UART, SPI, I^{2}C, ADC and general-purpose timers, as well as up to 24 GPIOs. Additionally, an integrated voltage regulator, power-on-reset circuit, and sleep timer are available.

Figure 1. STM32W108 block diagram



1.1 Development tools

The STM32W108 implements both the ARM Serial Wire and JTAG debug interfaces. These interfaces provide real time, non-intrusive programming and debugging capabilities. Serial Wire and JTAG provide the same functionality, but are mutually exclusive. The Serial Wire interface uses two pins; the JTAG interface uses five. Serial Wire is preferred, since it uses fewer pins.

The STM32W108 also integrates the standard ARM system debug components: Flash Patch and Breakpoint (FPB), Data Watchpoint and Trace (DWT), and Instrumentation Trace Macrocell (DWT).

1.2 Overview

1.2.1 Functional description

The STM32W108 radio receiver is a low-IF, super-heterodyne receiver. The architecture has been chosen to optimize co-existence with other devices in the 2.4 GHz band (namely, WIFI and Bluetooth), and to minimize power consumption. The receiver uses differential signal paths to reduce sensitivity to noise interference. Following RF amplification, the signal is downconverted by an image-rejecting mixer, filtered, and then digitized by an ADC.

The radio transmitter uses an efficient architecture in which the data stream directly modulates the VCO frequency. An integrated power amplifier (PA) provides the output power. Digital logic controls Tx path and output power calibration. If the STM32W108 is to be used with an external PA, use the TX_ACTIVE or nTX_ACTIVE signal to control the timing of the external switching logic.

The integrated 4.8 GHz VCO and loop filter minimize off-chip circuitry. Only a 24 MHz crystal with its loading capacitors is required to establish the PLL local oscillator signal.

The MAC interfaces the on-chip RAM to the Rx and Tx baseband modules. The MAC provides hardware-based IEEE 802.15.4 packet-level filtering. It supplies an accurate symbol time base that minimizes the synchronization effort of the software stack and meets the protocol timing requirements. In addition, it provides timer and synchronization assistance for the IEEE 802.15.4 CSMA-CA algorithm.

The STM32W108 integrates an ARM® Cortex-M3 microprocessor, revision r1p1. This industry-leading core provides 32 bit performance and is very power efficient. It has excellent code density using the ARM® Thumb 2 instruction set. The processor can be operated at 12 MHz or 24 MHz when using the crystal oscillator, or at 6 MHz or 12 MHz when using the integrated high frequency RC oscillator.

The STM32W108 has 64 Kbytes of Flash memory, 8 Kbytes of SRAM on-chip, and the ARM configurable memory protection unit (MPU).

The STM32W108 contains 24 GPIO pins shared with other peripheral or alternate functions. Because of flexible routing within the STM32W108, external devices can use the alternate functions on a variety of different GPIOs. The integrated Serial Controller SC1 can be configured for SPI (master or slave), I^{2}C (master-only), or UART operation, and the Serial Controller SC2 can be configured for SPI (master or slave) or I^{2}C (master-only) operation.

The STM32W108 has a general purpose ADC which can sample analog signals from six GPIO pins in single-ended or differential modes. It can also sample the regulated supply VDD_PADSA, the voltage reference VREF, and GND. The ADC has two selectable voltage ranges: 0 V to 1.2 V for the low voltage (input buffer disabled) and 0.1 V to VDD_PADS minus 0.1 V for the high voltage supply (input buffer enabled). The ADC has a DMA mode to capture samples and automatically transfer them into RAM. The integrated voltage reference for the ADC, VREF, can be made available to external circuitry. An external voltage reference can also be driven into the ADC.

The STM32W108 contains four oscillators: a high frequency 24 MHz external crystal oscillator (24 MHz HSE OSC), a high frequency 12 MHz internal RC oscillator (12 MHz HSI RC), an optional low frequency 32.768 kHz external crystal oscillator (32 kHz HSE OSC), and a 10 kHz internal RC oscillator (10 kHz LSI RC).

The STM32W108 has an ultra low power, deep sleep state with a choice of clocking modes. The sleep timer can be clocked with either the external 32.768 kHz crystal oscillator or with a 1 kHz clock derived from the internal 10 kHz LSI RC oscillator. Alternatively, all clocks can be disabled for the lowest power mode. In the lowest power mode, only external events on GPIO pins will wake up the chip. The STM32W108 has a fast startup time (typically 100 µs) from deep sleep to the execution of the first ARM® Cortex-M3 instruction.

The STM32W108 contains three power domains. The always-on high voltage supply powers the GPIO pads and critical chip functions. Regulated low voltage supplies power the rest of the chip. The low voltage supplies are be disabled during deep sleep to reduce power consumption. Integrated voltage regulators generate regulated 1.25 V and 1.8 V voltages from an unregulated supply voltage. The 1.8 V regulator output is decoupled and routed externally to supply analog blocks, RAM, and Flash memories. The 1.25 V regulator output is decoupled externally and supplies the core logic.

The digital section of the receiver uses a coherent demodulator to generate symbols for the hardware-based MAC. The digital receiver also contains the analog radio calibration routines and controls the gain within the receiver path.

In addition to 2 general-purpose timers, the STM32W108 also contains a watchdog timer to ensure protection against software crashes and CPU lockup, a 32-bit sleep timer dedicated to system timing and waking from sleep at specific times and an ARM® standard system event timer in the NVIC.

The STM32W108 integrates hardware support for a Packet Trace module, which allows robust packet-based debug.

Note: *The STM32W108 is not pin-compatible with the previous generation chip, the SN250, except for the RF section of the chip. Pins 1-11 and 45-48 are compatible, to ease migration to the STM32W108.*

1.2.2 ARM® Cortex™-M3 core

The STM32W108 integrates the ARM® Cortex™-M3 microprocessor, revision r1p1, developed by ARM Ltd, making the STM32W108 a true system-on-a-chip solution. The ARM® Cortex-M3 is an advanced 32-bit modified Harvard architecture processor that has separate internal program and data buses, but presents a unified program and data address space to software. The word width is 32 bits for both the program and data sides. The ARM® Cortex-M3 allows unaligned word and half-word data accesses to support efficiently-packed data structures.

The ARM® Cortex-M3 clock speed is configurable to 6 MHz, 12 MHz, or 24 MHz. For normal operation 12 MHz is preferred over 24 MHz due to its lower power consumption. The 6 MHz operation can only be used when radio operations are not required since the radio requires an accurate 12 MHz clock.

The ARM® Cortex-M3 in the STM32W108 has also been enhanced to support two separate memory protection levels. Basic protection is available without using the MPU, but the usual operation uses the MPU. The MPU protects unimplemented areas of the memory map to prevent common software bugs from interfering with software operation. The architecture could also separate the networking stack from the application code using a fine granularity RAM protection module. Errant writes are captured and details are reported to the developer to assist in tracking down and fixing issues.

2 Documentation conventions

Table 1. Description of abbreviations used for bit field access

Abbreviation	Description(1)
Read/Write (rw)	Software can read and write to these bits.
Read-only (r)	Software can only read these bits.
Write only (w)	Software can only write to this bit. Reading returns the reset value.
Read/Write in (MPU) Privileged mode only (rws)	Software can read and write to these bits only in Privileged mode. For more information, please refer to *RAM memory protection on page 35* and *Memory protection unit on page 40*.

1. The conditions under which the hardware (core) sets or clears this field are explained in details in the bit field description, as well as the events that may be generated by writing to the bit.

3 Pinout and pin description

Figure 2. 48-pin VFQFPN pinout

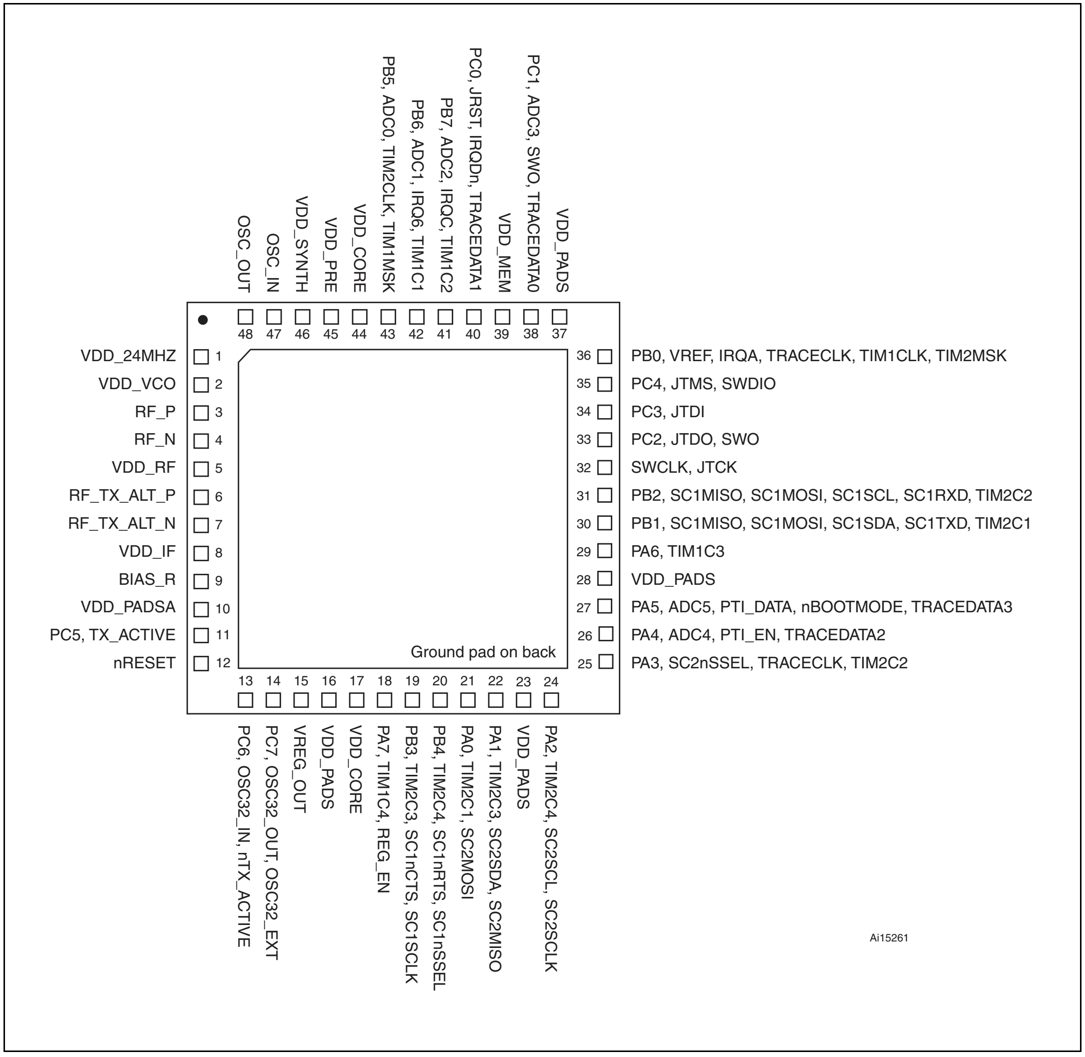

OSC_OUT 48
OSC_IN 47
VDD_SYNTH 46
VDD_PRE 45
VDD_CORE 44
PB5, ADC0, TIM2CLK, TIM1MSK 43
PB6, ADC1, IRQ6, TIM1C1 42
PB7, ADC2, IRQC, TIM1C2 41
PC0, JRST, IRQDn, TRACEDATA1 40
VDD_MEM 39
PC1, ADC3, SWO, TRACEDATA0 38
VDD_PADS 37
VDD_24MHZ 1
VDD_VCO 2
RF_P 3
RF_N 4
VDD_RF 5
RF_TX_ALT_P 6
RF_TX_ALT_N 7
VDD_IF 8
BIAS_R 9
VDD_PADSA 10
PC5, TX_ACTIVE 11
nRESET 12
36 PB0, VREF, IRQA, TRACECLK, TIM1CLK, TIM2MSK
35 PC4, JTMS, SWDIO
34 PC3, JTDI
33 PC2, JTDO, SWO
32 SWCLK, JTCK
31 PB2, SC1MISO, SC1MOSI, SC1SCL, SC1RXD, TIM2C2
30 PB1, SC1MISO, SC1MOSI, SC1SDA, SC1TXD, TIM2C1
29 PA6, TIM1C3
28 VDD_PADS
27 PA5, ADC5, PTI_DATA, nBOOTMODE, TRACEDATA3
26 PA4, ADC4, PTI_EN, TRACEDATA2
25 PA3, SC2nSSEL, TRACECLK, TIM2C2
Ground pad on back
13 PC6, OSC32_IN, nTX_ACTIVE
14 PC7, OSC32_OUT, OSC32_EXT
15 VREG_OUT
16 VDD_PADS
17 VDD_CORE
18 PA7, TIM1C4, REG_EN
19 PB3, TIM2C3, SC1nCTS, SC1SCLK
20 PB4, TIM2C4, SC1nRTS, SC1nSSEL
21 PA0, TIM2C1, SC2MOSI
22 PA1, TIM2C3, SC2SDA, SC2MISO
23 VDD_PADS
24 PA2, TIM2C4, SC2SCL, SC2SCLK
Ai15261

Table 2. Pin descriptions

Pin no.	Signal	Direction	Description
1	VDD_24MHZ	Power	1.8V high-frequency oscillator supply
2	VDD_VCO	Power	1.8V VCO supply
3	RF_P	I/O	Differential (with RF_N) receiver input/transmitter output
4	RF_N	I/O	Differential (with RF_P) receiver input/transmitter output
5	VDD_RF	Power	1.8V RF supply (LNA and PA)
6	RF_TX_ALT_P	O	Differential (with RF_TX_ALT_N) transmitter output (optional)
7	RF_TX_ALT_N	O	Differential (with RF_TX_ALT_P) transmitter output (optional)
8	VDD_IF	Power	1.8V IF supply (mixers and filters)

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
9	BIAS_R	I	Bias setting resistor
10	VDD_PADSA	Power	Analog pad supply (1.8V)
11	PC5	I/O	Digital I/O
	TX_ACTIVE	O	Logic-level control for external Rx/Tx switch. The STM32W108 baseband controls TX_ACTIVE and drives it high (VDD_PADS) when in Tx mode. Select alternate output function with GPIOC_CRH[7:4]
12	nRESET	I	Active low chip reset (internal pull-up)
13	PC6	I/O	Digital I/O
	OSC32_IN	I/O	32.768 kHz crystal oscillator Select analog function with GPIOC_CRH[11:8]
	nTX_ACTIVE	O	Inverted TX_ACTIVE signal (see PC5) Select alternate output function with GPIOC_CRH[11:8]
14	PC7	I/O	Digital I/O
	OSC32_OUT	I/O	32.768 kHz crystal oscillator. Select analog function with GPIOC_CRH[15:12]
	OSC32_EXT	I	Digital 32 kHz clock input source
15	VREG_OUT	Power	Regulator output (1.8 V while awake, 0 V during deep sleep)
16	VDD_PADS	Power	Pads supply (2.1-3.6 V)
17	VDD_CORE	Power	1.25 V digital core supply decoupling
18	PA7	I/O High current	Digital I/O. Disable REG_EN with GPIO_DBGCR[4]
	TIM1_CH4	O	Timer 1 Channel 4 output Enable timer output with TIM1_CCER Select alternate output function with GPIOA_CRH[15:12] Disable REG_EN with GPIO_DBGCR[4]
		I	Timer 1 Channel 4 input. (Cannot be remapped.)
	REG_EN	O	External regulator open drain output. (Enabled after reset.)

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
19	PB3	I/O	Digital I/O
	TIM2_CH3 (see Pin 22)	O	Timer 2 channel 3 output Enable remap with TIM2_OR[6] Enable timer output in TIM2_CCER Select alternate output function with GPIOB_CRL[15:12]
		I	Timer 2 channel 3 input. Enable remap with TIM2_OR[6].
	UART_CTS	I	UART CTS handshake of Serial Controller 1 Enable with SC1_UARTCR[5] Select UART with SC1_CR
	SC1SCLK	O	SPI master clock of Serial Controller 1 Either disable timer output in TIM2_CCER or disable remap with TIM2_OR[6] Enable master with SC1_SPICR[4] Select SPI with SC1_CR Select alternate output function with GPIOB_CRL[15:12]
		I	SPI slave clock of Serial Controller 1 Enable slave with SC1_SPICR[4] Select SPI with SC1_CR
20	PB4	I/O	Digital I/O
	TIM2_CH4 (see also Pin 24)	O	Timer 2 channel 4 output Enable remap with TIM2_OR[7] Enable timer output in TIM2_CCER Select alternate output function with GPIOB_CRH[3:0]
		I	Timer 2 channel 4 input. Enable remap with TIM2_OR[7].
	UART_RTS	O	UART RTS handshake of Serial Controller 1 Either disable timer output in TIM2_CCER or disable remap with TIM2_OR[7] Enable with SC1_UARTCR[5] Select UART with SC1_CR Select alternate output function with GPIOB_CRH[3:0]
	SC1nSSEL	I	SPI slave select of Serial Controller 1 Enable slave with SC1_SPICR[4] Select SPI with SC1_CR

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
21	PA0	I/O	Digital I/O
	TIM2_CH1 (see also Pin 30)	O	Timer 2 channel 1 output Disable remap with TIM2_OR[4] Enable timer output in TIM2_CCER Select alternate output function with GPIOA_CRL[3:0]
		I	Timer 2 channel 1 input. Disable remap with TIM2_OR[4].
	SC2MOSI	O	SPI master data out of Serial Controller 2 Either disable timer output in TIM2_CCER or enable remap with TIM2_OR[4] Enable master with SC2_SPICR[4] Select SPI with SC2_CR Select alternate output function with GPIOA_CRL[3:0]
		I	SPI slave data in of Serial Controller 2 Enable slave with SC2_SPICR[4] Select SPI with SC2_CR
22	PA1	I/O	Digital I/O
	TIM2_CH3 (see also Pin 19)	O	Timer 2 channel 3 output Disable remap with TIM2_OR[6] Enable timer output in TIM2_CCER Select alternate output function with GPIOA_CRL[7:4]
		I	Timer 2 channel 3 input. Disable remap with TIM2_OR[6].
	SC2SDA	I/O	I^{2}C data of Serial Controller 2 Either disable timer output in TIM2_CCER or enable remap with TIM2_OR[6] Select I^{2}C with SC2_CR Select alternate open-drain output function with GPIOA_CRL[7:4]
	SC2MISO	O	SPI slave data out of Serial Controller 2 Either disable timer output in TIM2_CCER or enable remap with TIM2_OR[6] Enable slave with SC2_SPICR[4] Select SPI with SC2_CR Select alternate output function with GPIOA_CRL[7:4]
		I	SPI master data in of Serial Controller 2 Enable slave with SC2_SPICR[4] Select SPI with SC2_CR
23	VDD_PADS	Power	Pads supply (2.1-3.6V)

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
24	PA2	I/O	Digital I/O
	TIM2_CH4 (see also Pin 20)	O	Timer 2 channel 4 output Disable remap with TIM2_OR[7] Enable timer output in TIM2_CCER Select alternate output function with GPIOA_CRL[11:8]
		I	Timer 2 channel 4 input. Disable remap with TIM2_OR[7].
	SC2SCL	I/O	I^{2}C clock of Serial Controller 2 Either disable timer output in TIM2_CCER or enable remap with TIM2_OR[7] Select I^{2}C with SC2_CR Select alternate open-drain output function with GPIOA_CRL[11:8]
	SC2SCLK	O	SPI master clock of Serial Controller 2 Either disable timer output in TIM2_CCER or enable remap with TIM2_OR[7] Enable master with SC2_SPICR[4] Select SPI with SC2_CR Select alternate output function with GPIOA_CRL[11:8]
		I	SPI slave clock of Serial Controller 2 Enable slave with SC2_SPICR[4] Select SPI with SC2_CR
25	PA3	I/O	Digital I/O
	SC2nSSEL	I	SPI slave select of Serial Controller 2 Enable slave with SC2_SPICR[4] Select SPI with SC2_CR
	TRACECLK (see also Pin 36)	O	Synchronous CPU trace clock Either disable timer output in TIM2_CCER or enable remap with TIM2_OR[5] Enable trace interface in ARM core Select alternate output function with GPIOA_CRL[15:12]
	TIM2_CH2 (see also Pin 31)	O	Timer 2 channel 2 output Disable remap with TIM2_OR[5] Enable timer output in TIM2_CCER Select alternate output function with GPIOA_CRL[15:12]
		I	Timer 2 channel 2 input. Disable remap with TIM2_OR[5].
26	PA4	I/O	Digital I/O
	ADC4	Analog	ADC Input 4. Select analog function with GPIOA_CRH[3:0].
	PTI_EN	O	Frame signal of Packet Trace Interface (PTI). Disable trace interface in ARM core. Select alternate output function with GPIOA_CRH[3:0].
	TRACEDATA2	O	Synchronous CPU trace data bit 2. Select 4-wire synchronous trace interface in ARM core. Enable trace interface in ARM core. Select alternate output function with GPIOA_CRH[3:0].

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
27	PA5	I/O	Digital I/O
	ADC5	Analog	ADC Input 5. Select analog function with GPIOA_CRH[7:4].
	PTI_DATA	O	Data signal of Packet Trace Interface (PTI). Disable trace interface in ARM core. Select alternate output function with GPIOA_CRH[7:4].
	nBOOTMODE	I	Embedded serial bootloader activation out of reset. Signal is active during and immediately after a reset on NRST. See *Section 6.2: Resets on page 46* for details.
	TRACEDATA3	O	Synchronous CPU trace data bit 3. Select 4-wire synchronous trace interface in ARM core. Enable trace interface in ARM core. Select alternate output function with GPIOA_CRH[7:4]
28	VDD_PADS	Power	Pads supply (2.1-3.6 V)
29	PA6	I/O High current	Digital I/O
	TIM1_CH3	O	Timer 1 channel 3 output Enable timer output in TIM1_CCER Select alternate output function with GPIOA_CRH[11:8]
		I	Timer 1 channel 3 input (Cannot be remapped.)

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
30	PB1	I/O	Digital I/O
	SC1MISO	O	SPI slave data out of Serial Controller 1 Either disable timer output in TIM2_CCER or disable remap with TIM2_OR[4] Select SPI with SC1_CR Select slave with SC1_SPICR Select alternate output function with GPIOB_CRL[7:4]
	SC1MOSI	O	SPI master data out of Serial Controller 1 Either disable timer output in TIM2_CCER or disable remap with TIM2_OR[4] Select SPI with SC1_CR Select master with SC1_SPICR Select alternate output function with GPIOB_CRL[7:4]
	SC1SDA	I/O	I^2C data of Serial Controller 1 Either disable timer output in TIM2_CCER, or disable remap with TIM2_OR[4] Select I^2C with SC1_CR Select alternate open-drain output function with GPIOB_CRL[7:4]
	SC1TXD	O	UART transmit data of Serial Controller 1 Either disable timer output in TIM2_CCER or disable remap with TIM2_OR[4] Select UART with SC1_CR Select alternate output function with GPIOB_CRL[7:4]
	TIM2_CH1 (see also Pin 21)	O	Timer 2 channel 1 output Enable remap with TIM2_OR[4] Enable timer output in TIM2_CCER Select alternate output function with GPIOA_CRL[7:4]
		I	Timer 2 channel 1 input. Disable remap with TIM2_OR[4].

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
31	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	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	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
34	PC3	I/O	Digital I/O Either Enable with GPIO_DBGCR[5], or enable Serial Wire mode (see JTMS description)
	JTDI	I	JTAG data in from debugger Selected when in JTAG mode (default mode, see JTMS description, Pin 35) Internal pull-up is enabled

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
35	PC4	I/O	Digital I/O Enable with GPIO_DBGCR[5]
	JTMS	I	JTAG mode select from debugger Selected when in JTAG mode (default mode) JTAG mode is enabled after power-up or by forcing NRST low Select Serial Wire mode using the ARM-defined protocol through a debugger Internal pull-up is enabled
	SWDIO	I/O	Serial Wire bidirectional data to/from debugger Enable Serial Wire mode (see JTMS description) Select Serial Wire mode using the ARM-defined protocol through a debugger Internal pull-up is enabled
36	PB0	I/O	Digital I/O
	VREF	Analog O	ADC reference output. Enable analog function with GPIOB_CRL[3:0].
	VREF	Analog I	ADC reference input. Enable analog function with GPIOB_CRL[3:0]. Enable reference output with an ST system function.
	IRQA	I	External interrupt source A.
	TRACECLK (see also Pin 25)	O	Synchronous CPU trace clock. Enable trace interface in ARM core. Select alternate output function with GPIOB_CRL[3:0].
	TIM1CLK	I	Timer 1 external clock input.
	TIM2MSK	I	Timer 2 external clock mask input.
37	VDD_PADS	Power	Pads supply (2.1 to 3.6 V).
38	PC1	I/O	Digital I/O
	ADC3	Analog	ADC Input 3 Enable analog function with GPIOC_CRL[7:4]
	SWO (see also Pin 33)	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[7:4]
	TRACEDATA0	O	Synchronous CPU trace data bit 0 Select 1-, 2- or 4-wire synchronous trace interface in ARM core Enable trace interface in ARM core Select alternate output function with GPIOC_CRL[7:4]
39	VDD_MEM	Power	1.8 V supply (Flash, RAM)

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
40	PC0	I/O High current	Digital I/O Either enable with GPIO_DBGCR[5], or enable Serial Wire mode (see JTMS description, Pin 35) and disable TRACEDATA1
	JRST	I	JTAG reset input from debugger Selected when in JTAG mode (default mode, see JTMS description) and TRACEDATA1 is disabled Internal pull-up is enabled
	IRQD (1)	I	Default external interrupt source D
	TRACEDATA1	O	Synchronous CPU trace data bit 1 Select 2- or 4-wire synchronous trace interface in ARM core Enable trace interface in ARM core Select alternate output function with GPIOC_CRL[3:0]
41	PB7	I/O High current	Digital I/O
	ADC2	Analog	ADC Input 2 Enable analog function with GPIOB_CRH[15:12]
	IRQC (1)	I	Default external interrupt source C
	TIM1_CH2	O	Timer 1 channel 2 output Enable timer output in TIM1_CCER Select alternate output function with GPIOB_CRH[15:12]
		I	Timer 1 channel 2 input (Cannot be remapped)
42	PB6	I/O High current	Digital I/O
	ADC1	Analog	ADC Input 1 Enable analog function with GPIOB_CRH[11:8]
	IRQB	I	External interrupt source B
	TIM1_CH1	O	Timer 1 channel 1 output Enable timer output in TIM1_CCER Select alternate output function with GPIOB_CRH[11:8]
		I	Timer 1 channel 1 input (Cannot be remapped)
43	PB5	I/O	Digital I/O
	ADC0	Analog	ADC Input 0 Enable analog function with GPIOB_CRH[7:4]
	TIM2CLK	I	Timer 2 external clock input
	TIM1MSK	I	Timer 2 external clock mask input
44	VDD_CORE	Power	1.25 V digital core supply decoupling
45	VDD_PRE	Power	1.8 V prescaler supply
46	VDD_SYNTH	Power	1.8 V synthesizer supply
47	OSC_IN	I/O	24 MHz HSE OSC or left open when using external clock input on OSC_OUT

Table 2. Pin descriptions (continued)

Pin no.	Signal	Direction	Description
48	OSC_OUT	I/O	24 MHz HSE OSC or external clock input
49	GND	Ground	Ground supply pad in the bottom center of the package.

1. IRQC and IRQD external interrupts can be mapped to any digital I/O pin using the using the EXTIC_CR and EXTID_CR registers.

4 Embedded memory

4.1 Memory organization and memory map

The bytes are coded in the memory in Little Endian format. The lowest numbered byte in a word is considered the word’s least significant byte and the highest numbered byte the most significant.

For detailed mapping of peripheral registers, please refer to the relevant section.

All the memory areas that are not allocated to on-chip memories and peripherals are considered “Reserved”).

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

Figure 3. STM32W108 memory mapping

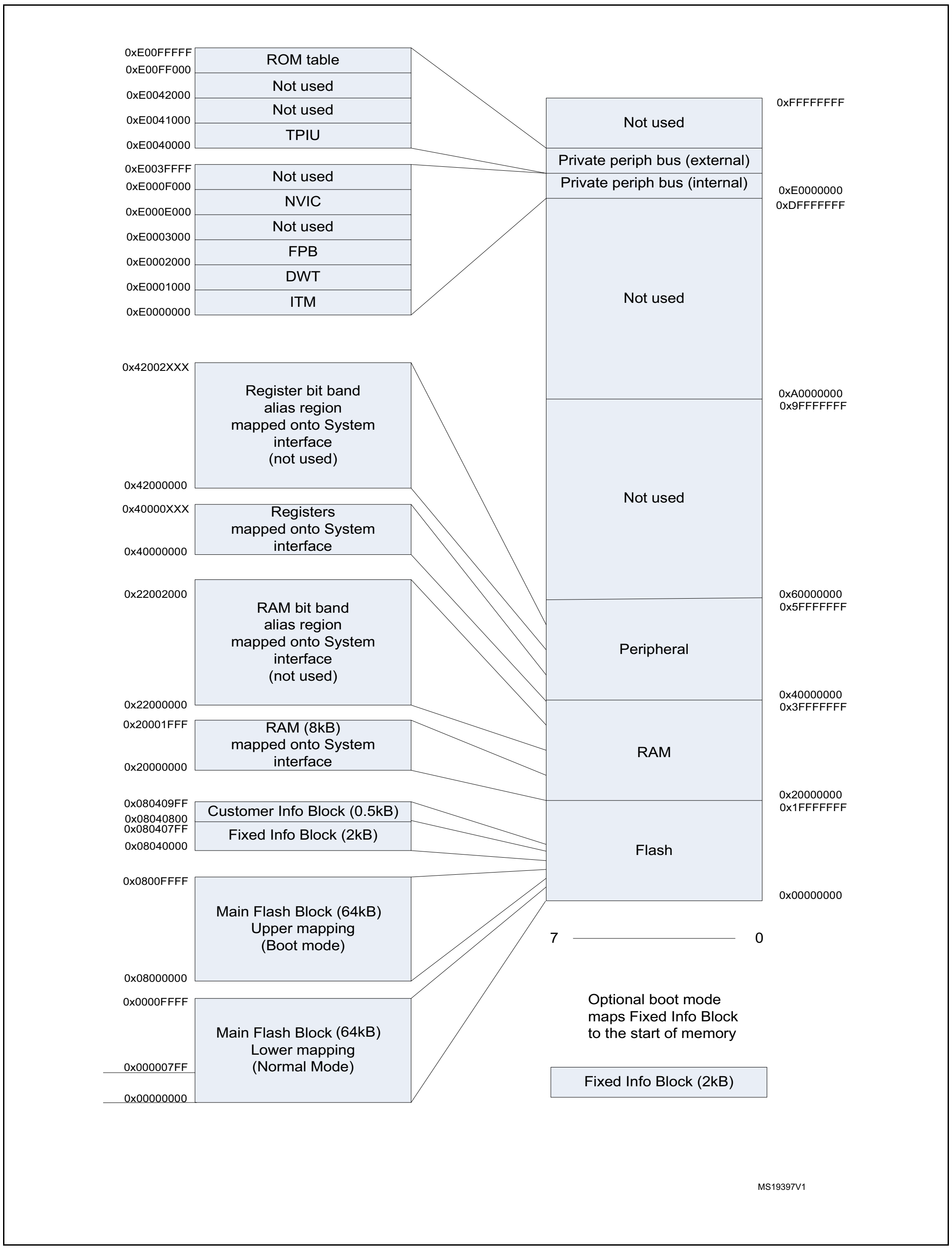


Table 3. STM32W108xx peripheral register boundary addresses

Bus	Boundary address	Peripheral	Register map
APB	0x4000 F000 - 0x4000 FFFF	General-purpose timer 2 (TIM2)	*Table 10.3.20: General-purpose timers 1 and 2 (TIM1/TIM2) register map*
	0x4000 E000 - 0x4000 EFFF	General-purpose timer 1 (TIM1)	*Table 10.3.20: General-purpose timers 1 and 2 (TIM1/TIM2) register map*
	0x4000 D025 - 0x4000 DFFF	Reserved	-
	0x4000 D000 - 0x4000 D024	Analog-to-digital converter (ADC)	*Table 11.3.12: Analog-to-digital converter (ADC) register map*
	0x4000 C871 - 0x4000 CFFF	Reserved	-
	0x4000 C800 - 0x4000 C870	Serial interface (SC1)	*Table 9.12.17: Serial interface (SC1/SC2) register map*
	0x4000 C071 - 0x4000 C7FF	Reserved	-
	0x4000 C000 - 0x4000 C070	Serial interface (SC2)	*Table 9.12.17: Serial interface (SC1/SC2) register map*
	0x4000 B000 - 0x4000 BFFF	General-purpose input/output (GPIO)	*Table 8.5.13: General-purpose input/output (GPIO) register map*
	0x4000 A000 - 0x4000 AFFF	Management interrupt (MGMT)	*Table 12.2.3: Management interrupt (MGMT) register map*
	0x4000 6025 - 0x4000 9FFF	Reserved	-
	0x4000 600C - 0x4000 6024	Sleeptimer (SLPTMR)	*MAC timer (MACTMR)/Watchdog (WDG)/Sleeptimer(SLPTMR) register map*
	0x4000 6009 - 0x4000 600B	Reserved	-
	0x4000 6000 - 0x4000 6008	Watchdog (WDG)	*MAC timer (MACTMR)/Watchdog (WDG)/Sleeptimer(SLPTMR) register map*
	0x4000 5000 - 0x4000 5FFF	Memory controller (MEM)	*Memory controller (MEM) register map*
	0x4000 4021 - 0x4000 4FFF	Reserved	-
	0x4000 4000 - 0x4000 4020	Clock switching (CLK)	*Clock switching (CLK) register map*
	0x4000 3000 - 0x4000 3FFF	Reserved	-

Table 3. STM32W108xx peripheral register boundary addresses (continued)

Bus	Boundary address	Peripheral	Register map
APB	0x4000 2000 - 0x4000 2FFF	MAC timer (MACTMR)	*MAC timer (MACTMR)/Watchdog (WDG)/Sleeptimer(SLP TMR) register map*
	0x4000 1000 - 0x4000 1FFF	Reserved	-
	0x4000 0000 - 0x4000 0FFF	Power management (PWR)	*Power management (PWR) register map*
	0x2000 4000 - 0x3FFF FFFF	Reserved	-
	0x2000 0000 - 0x2000 3FFF	SRAM	-
	0x0804 0000 - 0x1FFF FFFF	Reserved	-
	0x0800 0000 - 0x0803 FFFF	Main Flash memory (256 Kbyte)	-

4.2 Flash memory

The STM32W108 provides a total of 66.5 Kbytes of Flash memory in three separate blocks:

- Main Flash Block (MFB)
- Fixed Information Block (FIB)
- Customer Information Block (CIB)

The MFB is divided into 641024-byte pages. The CIB is a single 512-byte page. The FIB is a single 2048-byte page. The smallest erasable unit is one page and the smallest writable unit is an aligned 16-bit half-word. The Flash is guaranteed to have 10k write/erase cycles. The Flash cell has been qualified for a data retention time of >100 years at room temperature.

Flash may be programmed either through the Serial Wire/JTAG interface or through bootloader software. Programming Flash through Serial Wire/JTAG requires the assistance of RAM-based utility code. Programming through a bootloader requires specific software for over-the-air loading or serial link loading. A simplified, serial-link-only bootloader is also available preprogrammed into the FIB.

4.3 Random-access memory

The STM32W108 has 8 Kbytes of static RAM on-chip. The start of RAM is mapped to address 0x20000000. Although the ARM® Cortex-M3 allows bit band accesses to this address region, the standard MPU configuration does not permit use of the bit-band feature.

The RAM is physically connected to the AHB System bus and is therefore accessible to both the ARM® Cortex-M3 microprocessor and the debugger. The RAM can be accessed for both instruction and data fetches as bytes, half words, or words. The standard MPU configuration does not permit execution from the RAM, but for special purposes, such as programming the main Flash block, the MPU may be disabled. To the bus, the RAM appears as 32-bit wide memory and in most situations has zero wait state read or write access. In the higher CPU clock mode the RAM requires two wait states. This is handled by hardware transparent to the user application with no configuration required.

4.3.1 Direct memory access (DMA) to RAM

Several of the peripherals are equipped with DMA controllers allowing them to transfer data into and out of RAM autonomously. This applies to the radio (802.15.4 MAC), general purpose ADC, and both serial controllers. In the case of the serial controllers, the DMA is full duplex so that a read and a write to RAM may be requested at the same time. Thus there are six DMA channels in total.

The STM32W108 integrates a DMA arbiter that ensures fair access to the microprocessor as well as the peripherals through a fixed priority scheme appropriate to the memory bandwidth requirements of each master. The priority scheme is as follows, with the top peripheral being the highest priority:

1. General Purpose ADC
2. Serial Controller 2 Receive
3. Serial Controller 2 Transmit
4. MAC
5. Serial Controller 1 Receive
6. Serial Controller 1 Transmit

4.3.2 RAM memory protection

The STM32W108 integrates two memory protection mechanisms. The first memory protection mechanism is through the ARM® Cortex-M3 Memory Protection Unit (MPU) described in the Memory Protection Unit section. The MPU may be used to protect any area of memory. MPU configuration is normally handled by software. The second memory protection mechanism is through a fine granularity RAM protection module.This allows segmentation of the RAM into 32-byte blocks where any block can be marked as write protected. An attempt to write to a protected RAM block using a user mode write results in a bus error being signaled on the AHB System bus. A system mode write is allowed at any time and reads are allowed in either mode. The main purpose of this fine granularity RAM protection module is to notify the stack of erroneous writes to system areas of memory. RAM protection is configured using a group of registers that provide a bit map.Each bit in the map represents a 32-byte block of RAM. When the bit is set the block is write protected.

The fine granularity RAM memory protection mechanism is also available to the peripheral DMA controllers. A register bit is provided to enable the memory protection to include DMA writes to protected memory. If a DMA write is made to a protected location in RAM, a management interrupt is generated. At the same time the faulting address and the identification of the peripheral is captured for later debugging. Note that only peripherals capable of writing data to RAM, such as received packet data or a received serial port character, can generate this interrupt.

4.3.3 Memory controller

The STM32W108xx allows the RAM and DMA protection to be controlled using the memory controller interface. The chip contains eight RAM protection registers and two DMA protection registers. In addition, the chip contains a register, RAM_CR, for enabling the protection of the memory.

4.3.4 Memory controller registers

RAM is divided into 32 byte pages. Each page has a register bit that, when set, protects it from being written in user mode. The protection registers (MEM_PROT) are arranged in the register map as a 256-bit vector. Bit 0 of this vector protects page 0 which begins at location 0x2000 0000 and ends at 0x2000 001F. Bit 255 of this vector protects the top page which starts at 0x20001FE0 and ends at 0x2000 1FFF.

Memory RAM protection register x (RAM_PROTRx)

Address: 0x 4000 5000 (RAM_PROTR1), 0x 4000 5004 (RAM_PROTR2),
0x 4000 5008 (RAM_PROTR3), 0x 4000 500C (RAM_PROTR4),
0x 4000 5010 (RAM_PROTR5), 0x 4000 5014 (RAM_PROTR6),
0x 4000 5018(RAM_PROTR7), and 0x 4000 501C (RAM_PROTR8).

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Memory page protection x[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
Memory page protection x[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 Memory page protection x[31:0]:

Bit 0 in the RAM_PROTR1 protects page 0
…
Bit 31 in the RAM_PROTR1 protects page 31
Bit 0 in the RAM_PROTR2 protects page 32
…
Bit 31 in the RAM_PROTR2 protects page 63
Bit 0 in the RAM_PROTR3 protects page 64
…
Bit 31 in the RAM_PROTR3 protects page 95
Bit 0 in the RAM_PROTR4 protects page 96
…
Bit 31 in the RAM_PROTR4 protects page 127
Bit 0 in the RAM_PROTR5 protects page 128
…
Bit 31 in the RAM_PROTR5 protects page 159
Bit 0 in the RAM_PROTR6 protects page 160
…
Bit 31 in the RAM_PROTR6 protects page 191
Bit 0 in the RAM_PROTR7 protects page 192
…
Bit 31 in the RAM_PROTR7 protects page 223
Bit 0 in the RAM_PROTR8 protects page 224
….
Bit 31 in the RAM_PROTR8 protects page 255

Memory DMA protection register 1 (DMA_PROTR1)

Address: 0x 4000 5020
Reset value: 0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Offset[18:3]															
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**
Offset[2:0]			Address[11:0]												Reserved
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

Bits 31:13 Offset[18:0]:
Offset in RAM

Bits 12:1 Offset[11:0]:
DMA protection fault, faulting address.

Bit 0 Reserved, must be kept at reset value

Memory DMA protection register 2 (DMA_PROTR2)

Address: 0x 4000 5024
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													Channel[2:0]		
													r	r	r

Bits 31:3 Reserved, must be kept at reset value

Bits 2:0 Channel[2:0]: Channel encoding
7: Not used
6: Not used
5: SC2_RX
4: Not used
3: ADC
2: Not used
1: SC1_RX
0: Not used

Memory RAM control register (RAM_CR)

Address: 0x 4000 5028
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													WEN	Reserved	
													rw		

Bits 31:3	Reserved, must be kept at reset value
Bit 2	WEN: Makes all RAM writes appear as user mode
Bits 1:0	Reserved, must be kept at reset value

Memory controller (MEM) register map

Table 4 gives the MEM register map and reset values.

Table 4. MEM register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5000	RAM_PROTR1	Memory page protection 1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5004	RAM_PROTR2	Memory page protection 2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5008	RAM_PROTR3	Memory page protection 3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x500C	RAM_PROTR4	Memory protection 4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5010	RAM_PROTR5	Memory protection 5[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5014	RAM_PROTR6	Memory protection 6[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5018	RAM_PROTR7	Memory protection 7[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x501C	RAM_PROTR8	Memory protection 8[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5020	DMA_PROTR1	Offset[18:0]																			Address[11:0]												Res.
	Reset value	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x5024	DMA_PROTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Channel[2:0]			
	Reset value																													0	0	0	0
0x5028	RAM_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WEN	Res.	Res.
	Reset value																														0		

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

4.4 Memory protection unit

The STM32W108 includes the ARM® Cortex-M3 Memory Protection Unit, or MPU. The MPU controls access rights and characteristics of up to eight address regions, each of which may be divided into eight equal sub-regions. Refer to the ARM® Cortex-M3 Technical Reference Manual (DDI 0337A) for a detailed description of the MPU.

ST software configures the MPU in a standard configuration and application software should not modify it. The configuration is designed for optimal detection of illegal instruction or data accesses. If an illegal access is attempted, the MPU captures information about the access type, the address being accessed, and the location of the offending software. This simplifies software debugging and increases the reliability of deployed devices. As a consequence of this MPU configuration, accessing RAM and register bit-band address alias regions is not permitted, and generates a bus fault if attempted.

5 Radio frequency module

The radio module consists of an analog front end and digital baseband as shown in *Figure 1: STM32W108 block diagram*.

5.1 Receive (Rx) path

The Rx path uses a low-IF, super-heterodyne receiver that rejects the image frequency using complex mixing and polyphase filtering. In the analog domain, the input RF signal from the antenna is first amplified and mixed down to a 4 MHz IF frequency. The mixers' output is filtered, combined, and amplified before being sampled by a 12 Msps ADC. The digitized signal is then demodulated in the digital baseband. The filtering within the Rx path improves the STM32W108's co-existence with other 2.4 GHz transceivers such as IEEE 802.15.4, IEEE 802.11g, and Bluetooth radios. The digital baseband also provides gain control of the Rx path, both to enable the reception of small and large wanted signals and to tolerate large interferers.

5.1.1 Rx baseband

The STM32W108 Rx digital baseband implements a coherent demodulator for optimal performance. The baseband demodulates the O-QPSK signal at the chip level and synchronizes with the IEEE 802.15.4-defined preamble. An automatic gain control (AGC) module adjusts the analog gain continuously every ¼ symbol until the preamble is detected. Once detected, the gain is fixed for the remainder of the packet. The baseband despreads the demodulated data into 4-bit symbols. These symbols are buffered and passed to the hardware-based MAC module for packet assembly and filtering.

In addition, the Rx baseband provides the calibration and control interface to the analog Rx modules, including the LNA, Rx baseband filter, and modulation modules. The ST RF software driver includes calibration algorithms that use this interface to reduce the effects of silicon process and temperature variation.

5.1.2 RSSI and CCA

The STM32W108 calculates the RSSI over every 8-symbol period as well as at the end of a received packet. The linear range of RSSI is specified to be at least 40 dB over temperature. At room temperature, the linear range is approximately 60 dB (-90 dBm to -30 dBm input signal).

The STM32W108 Rx baseband provides support for the IEEE 802.15.4-2003 RSSI CCA method, Clear channel reports busy medium if RSSI exceeds its threshold.

5.2 Transmit (Tx) path

The STM32W108 Tx path produces an O-QPSK-modulated signal using the analog front end and digital baseband. The area- and power-efficient Tx architecture uses a two-point modulation scheme to modulate the RF signal generated by the synthesizer. The modulated RF signal is fed to the integrated PA and then out of the STM32W108.

5.2.1 Tx baseband

The STM32W108 Tx baseband in the digital domain spreads the 4-bit symbol into its IEEE 802.15.4-2003-defined 32-chip sequence. It also provides the interface for software to calibrate the Tx module to reduce silicon process, temperature, and voltage variations.

5.2.2 TX_ACTIVE and nTX_ACTIVE signals

For applications requiring an external PA, two signals are provided called TX_ACTIVE and nTX_ACTIVE. These signals are the inverse of each other. They can be used for external PA power management and RF switching logic. In transmit mode the Tx baseband drives TX_ACTIVE high, as described in *Table 16: GPIO signal assignments on page 96*. In receive mode the TX_ACTIVE signal is low. TX_ACTIVE is the alternate function of PC5, and nTX_ACTIVE is the alternate function of PC6. See *Section 8: General-purpose input/output on page 89* for details of the alternate GPIO functions.

5.3 Calibration

The ST RF software driver calibrates the radio using dedicated hardware resources.

5.4 Integrated MAC module

The STM32W108 integrates most of the IEEE 802.15.4 MAC requirements in hardware. This allows the ARM® Cortex-M3 CPU to provide greater bandwidth to application and network operations. In addition, the hardware acts as a first-line filter for unwanted packets. The STM32W108 MAC uses a DMA interface to RAM to further reduce the overall ARM® Cortex-M3 CPU interaction when transmitting or receiving packets.

When a packet is ready for transmission, the software configures the Tx MAC DMA by indicating the packet buffer RAM location. The MAC waits for the backoff period, then switches the baseband to Tx mode and performs channel assessment. When the channel is clear the MAC reads data from the RAM buffer, calculates the CRC, and provides 4-bit symbols to the baseband. When the final byte has been read and sent to the baseband, the CRC remainder is read and transmitted.

The MAC is in Rx mode most of the time. In Rx mode various format and address filters keep unwanted packets from using excessive RAM buffers, and prevent the CPU from being unnecessarily interrupted. When the reception of a packet begins, the MAC reads 4-bit symbols from the baseband and calculates the CRC. It then assembles the received data for storage in a RAM buffer. Rx MAC DMA provides direct access to RAM. Once the packet has been received additional data, which provides statistical information on the packet to the software stack, is appended to the end of the packet in the RAM buffer space.

The primary features of the MAC are:

- CRC generation, appending, and checking
- Hardware timers and interrupts to achieve the MAC symbol timing
- Automatic preamble and SFD pre-pending on Tx packets
- Address recognition and packet filtering on Rx packets
- Automatic acknowledgement transmission
- Automatic transmission of packets from memory
- Automatic transmission after backoff time if channel is clear (CCA)
- Automatic acknowledgement checking
- Time stamping received and transmitted messages
- Attaching packet information to received packets (LQI, RSSI, gain, time stamp, and packet status)
- IEEE 802.15.4 timing and slotted/unslotted timing

5.5 Packet trace interface (PTI)

The STM32W108 integrates a true PHY-level PTI for effective network-level debugging. It monitors all the PHY Tx and Rx packets between the MAC and baseband modules without affecting their normal operation. It cannot be used to inject packets into the PHY/MAC interface. This 500 kbps asynchronous interface comprises the frame signal (PTI_EN, PA4) and the data signal (PTI_DATA, PA5).

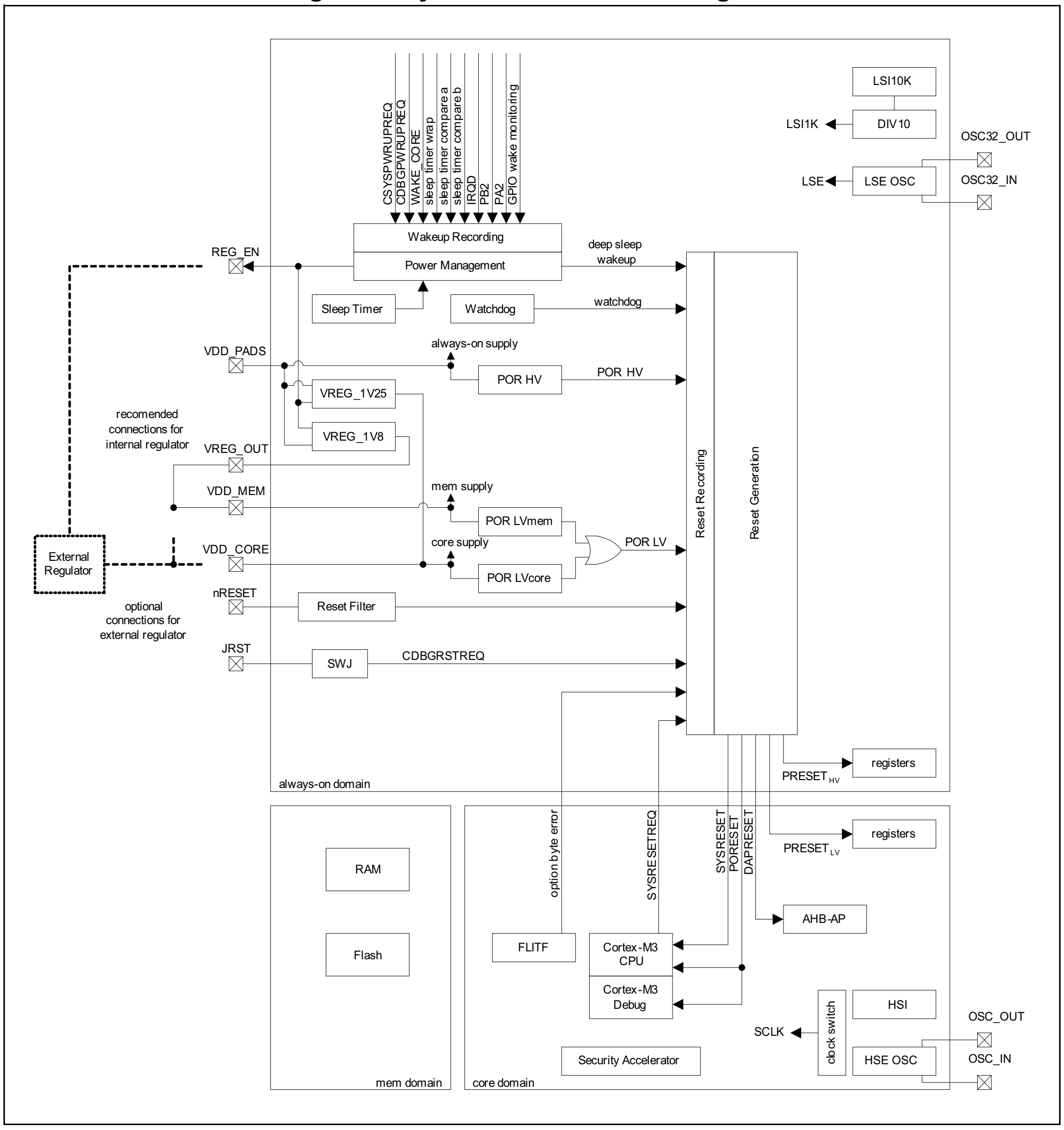
5.6 Random number generator

Thermal noise in the analog circuitry is digitized to provide entropy for a true random number generator (TRNG). The TRNG produces 16-bit uniformly distributed numbers. The Software can use the TRNG to seed a pseudo random number generator (PNRG). The TRNG is also used directly for cryptographic key generation.

6 System modules

System modules encompass power, resets, clocks, system timers, power management, and encryption. *Figure 4* shows these modules and how they interact.

Figure 4. System module block diagram



6.1 Power domains

The STM32W108 contains three power domains:

- An "always on domain" containing all logic and analog cells required to manage the STM32W108's power modes, including the GPIO controller and sleep timer. This domain must remain powered.
- A "core domain" containing the CPU, Nested Vectored Interrupt Controller (NVIC), and peripherals. To save power, this domain can be powered down using a mode called deep sleep.
- A "memory domain" containing the RAM and Flash memories. This domain is managed by the power management controller. When in deep sleep, the RAM portion of this domain is powered from the always-on domain supply to retain the RAM contents while the regulators are disabled. During deep sleep the Flash portion is completely powered down.

6.1.1 Internally regulated power

The preferred and recommended power configuration is to use the internal regulated power supplies to provide power to the core and memory domains. The internal regulators (VREG_1V25 and VREG_1V8) generate nominal 1.25 V and 1.8 V supplies. The 1.25 V supply is internally routed to the core domain and to an external pin. The 1.8 V supply is routed to an external pin where it can be externally routed back into the chip to supply the memory domain. The internal regulators are described in *Section 7: Integrated voltage regulator on page 87*.

When using the internal regulators, the always-on domain must be powered between 2.1 V and 3.6 V at all four VDD_PADS pins.

When using the internal regulators, the VREG_1V8 regulator output pin (VREG_OUT) must be connected to the VDD_MEM, VDD_PADSA, VDD_VCO, VDD_RF, VDD_IF, VDD_PRE, and VDD_SYNTH pins.

When using the internal regulators, the VREG_1V25 regulator output and supply requires a connection between both VDD_CORE pins.

6.1.2 Externally regulated power

Optionally, the on-chip regulators may be left unused, and the core and memory domains may instead be powered from external supplies. For simplicity, the voltage for the core domain can be raised to nominal 1.8 V, requiring only one external regulator. Note that if the core domain is powered at a higher voltage (1.8 V instead of 1.25 V) then power consumption increases. A regulator enable signal, REG_EN, is provided for control of external regulators. This is an open-drain signal that requires an external pull-up resistor. If REG_EN is not required to control external regulators it can be disabled (see *Section 8.1.3: Forced functions on page 92*).

Using an external regulator requires the always-on domain to be powered between 1.8 V and 3.6 V at all four VDD_PADS pins.

When using an external regulator, the VREG_1V8 regulator output pin (VREG_OUT) must be left unconnected.

When using an external regulator, this external nominal 1.8 V supply has to be connected to both VDD_CORE pins and to the VDD_MEM, VDD_PADSA, VDD_VCO, VDD_RF, VDD_IF, VDD_PRE and VDD_SYNTH pins.

6.2 Resets

The STM32W108 resets are generated from a number of sources. Each of these reset sources feeds into central reset detection logic that causes various parts of the system to be reset depending on the state of the system and the nature of the reset event.

6.2.1 Reset sources

For power-on reset (POR HV and POR LV) thresholds, see *Section 14.3.2: Operating conditions at power-up on page 246*.

Watchdog reset

The STM32W108 contains a watchdog timer (see also the Watchdog Timer section) that is clocked by the internal 1 kHz timing reference. When the timer expires it generates the reset source WATCHDOG_RESET to the Reset Generation module.

Software reset

The ARM® Cortex-M3 CPU can initiate a reset under software control. This is indicated with the reset source SYSRESETREQ to the Reset Generation module.

Note: When using certain external debuggers, the chip may lock up require a pin reset or power cycle if the debugger asserts SYSRESETREQ. It is recommended not to write to the SCS_AIRCR register directly from application code. The ST software provides a reset function that should be used instead. This reset function ensures that the chip is in a safe clock mode prior to triggering the reset.

Option byte error

The Flash memory controller contains a state machine that reads configuration information from the information blocks in the Flash at system start time. An error check is performed on the option bytes that are read from Flash and, if the check fails, an error is signaled that provides the reset source OPT_BYTE_ERROR to the Reset Generation module.

If an option byte error is detected, the system restarts and the read and check process is repeated. If the error is detected again the process is repeated but stops on the 3rd failure. The system is then placed into an emulated deep sleep where recovery is possible. In this state, Flash memory readout protection is forced active to prevent secure applications from being compromised.

Debug reset

The Serial Wire/JTAG Interface (SWJ) provides access to the SWJ Debug Port (SWJ-DP) registers. By setting the register bit CDBGRSTREQ in the SWJ-DP, the reset source CDBGRSTREQ is provided to the Reset Generation module.

JTAG reset

One of the STM32W108's pins can function as the JTAG reset, conforming to the requirements of the JTAG standard. This input acts independently of all other reset sources and, when asserted, does not reset any on-chip hardware except for the JTAG TAP. If the STM32W108 is in the Serial Wire mode or if the SWJ is disabled, this input has no effect.

Deep sleep reset

The Power Management module informs the Reset Generation module of entry into and exit from the deep sleep states. The deep sleep reset is applied in the following states: before entry into deep sleep, while removing power from the memory and core domain, while in deep sleep, while waking from deep sleep, and while reapplying power until reliable power levels have been detect by POR LV.

The Power Management module allows a special emulated deep sleep state that retains memory and core domain power while in deep sleep.

6.2.2 Reset recording

The STM32W108 records the last reset condition that generated a restart to the system. The reset conditions recorded are:

• PWRHV	Always-on domain power supply failure
• PWRLV	Core or memory domain power supply failure
• RSTB	NRST pin asserted
• WDG	Watchdog timer expired
• SWRST	Software reset by SYSERSETREQ from ARM® Cortex-M3 CPU
• WKUP	Wake-up from deep sleep
• OBFAIL	Error check failed when reading option bytes from Flash memory

The *Reset status register (RST_SR)* is used to read back the last reset event. All bits are mutually exclusive except the OBFAIL bit which preserves the original reset event when set.

Note: While CPU Lockup is marked as a reset condition in software, CPU Lockup is not specifically a reset event. CPU Lockup is set to indicate that the CPU entered an unrecoverable exception. Execution stops but a reset is not applied. This is so that a debugger can interpret the cause of the error. We recommend that in a live application (i.e. no debugger attached) the watchdog be enabled by default so that the STM32W108 can be restarted.

6.2.3 Reset generation

The Reset Generation module responds to reset sources and generates the following reset signals:

• PORESET	Reset of the ARM® Cortex-M3 CPU and ARM® Cortex-M3 System Debug components (Flash Patch and Breakpoint, Data Watchpoint and Trace, Instrumentation Trace Macrocell, Nested Vectored Interrupt Controller). ARM defines PORESET as the region that is reset when power is applied.
• SYSRESET	Reset of the ARM® Cortex-M3 CPU without resetting the Core Debug and System Debug components, so that a live system can be reset without disturbing the debug configuration.
• DAPRESET	Reset to the SWJ's AHB Access Port (AHB-AP).
• PRESETHV	Peripheral reset for always-on power domain, for peripherals that are required to retain their configuration across a deep sleep cycle.
• PRESETLV	Peripheral reset for core power domain, for peripherals that are not required to retain their configuration across a deep sleep cycle.

Table 5 shows which reset sources generate certain resets.

Table 5. Generated resets

Reset source	Reset generation				
	PORESET	**SYSRESET**	**DAPRESET**	**PRESETHV**	**PRESETLV**
POR HV	X	X	X	X	X
POR LV (in deep sleep)	X	X	X		X
POR LV (not in deep sleep)	X	X	X	X	X
RSTB	X	X		X	X
Watchdog reset		X		X	X
Software reset		X		X	X
Option byte error	X	X			X
Normal deep sleep	X	X	X		X
Emulated deep sleep		X			X
Debug reset		X			

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.

Reset (RST) register map

Table 6 gives the RST register map and reset values.

Table 6. RST register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x002C	RST_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LKUP	OBFAIL	WKUP	SWRST	WDG	PIN	PWRLV	PWRHV
	Reset value																									0	0	0	0	0	0	0	0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

6.3 Clocks

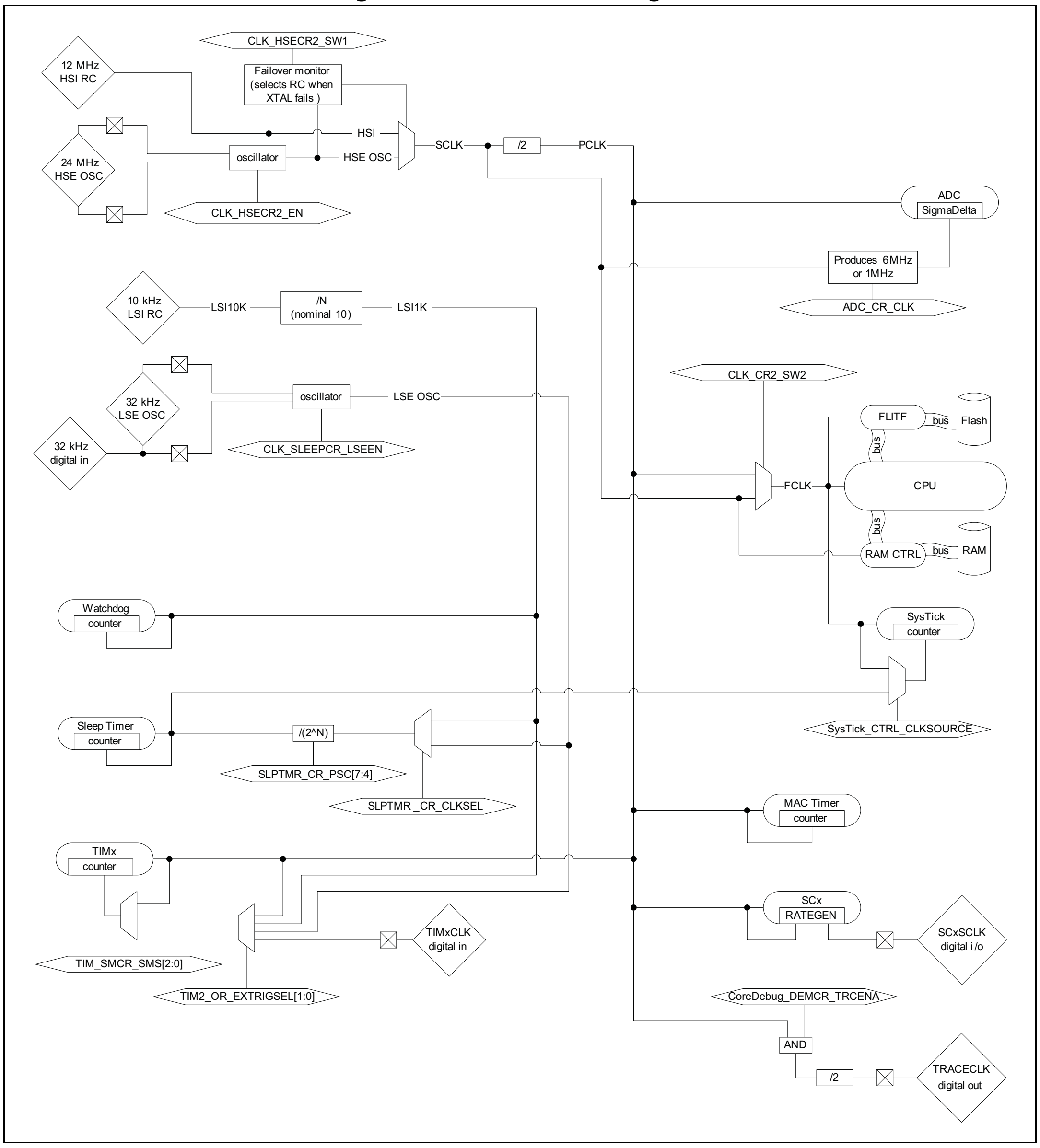
The STM32W108 integrates four oscillators:

- High frequency RC oscillator (HSI)
- 24 MHz crystal oscillator (HSE)
- 10 kHz LSI RC oscillator (LSI10K)
- 32.768 kHz crystal oscillator (LSE)

Note: The LSI1K clock is generated from the 10 kHz LSI RC oscillator (LSI10K). The default value is a divide by 10 for a nominal 1 kHz output clock.

Figure 5 shows a block diagram of the clocks in the STM32W108. This simplified view shows all the clock sources and the general areas of the chip to which they are routed.

Figure 5. Clocks block diagram



6.3.1 High-frequency internal RC oscillator (HSI)

The high-frequency RC oscillator (HSI) is used as the default system clock source when power is applied to the core domain. The nominal frequency coming out of reset is 12 MHz.

Most peripherals, excluding the radio peripheral, are fully functional using the HSI clock source. Application software must be aware that peripherals are clocked at different speeds depending on whether HSI or HSE OSC is being used. Since the frequency step of HSI is 0.5 MHz and the high-frequency crystal oscillator is used for calibration, the calibrated accuracy of HSI is ±250 kHz ±40 ppm. The UART and ADC peripherals may not be usable due to the lower accuracy of the HSI frequency.

See also *Section 14.5.1: High frequency internal clock characteristics on page 253*.

6.3.2 High-frequency crystal oscillator (HSE OSC)

The high-frequency crystal oscillator (HSE OSC) requires an external 24 MHz crystal with an accuracy of ±40 ppm. Based upon the application's bill of materials and current consumption requirements, the external crystal may cover a range of ESR requirements.

The crystal oscillator has a software-programmable bias circuit to minimize current consumption. ST software configures the bias circuit for minimum current consumption.

All peripherals including the radio peripheral are fully functional using the HSE OSC clock source. Application software must be aware that peripherals are clocked at different speeds depending on whether HSI or HSE OSC is being used.

If the 24 MHz crystal fails, a hardware failover mechanism forces the system to switch back to the high-frequency RC oscillator as the main clock source, and a non-maskable interrupt (NMI) is signaled to the ARM® Cortex-M3 NVIC.

See also *Section 14.5.2: High frequency external clock characteristics on page 253*.

6.3.3 Low-frequency internal RC oscillator (LSI10K)

A low-frequency RC oscillator (LSI10K) is provided as an internal timing reference. The nominal frequency coming out of reset is 10 kHz, and ST software calibrates this clock to 10 kHz. From the tuned 10 kHz oscillator (LSI10K) ST software calibrates a fractional-N divider to produce a 1 kHz reference clock, LSI1K.

See also *Section 14.5.3: Low frequency internal clock characteristics on page 254*.

6.3.4 Low-frequency crystal oscillator (LSE OSC)

A low-frequency 32.768 kHz crystal oscillator (LSE OSC) is provided as an optional timing reference for on-chip timers. This oscillator is designed for use with an external watch crystal.

See also *Section 14.5.4: Low frequency external clock characteristics on page 254*.

6.3.5 Clock switching

The STM32W108 has two switching mechanisms for the main system clock, providing four clock modes.

The register bit SW1 in the CLK_HSECR2 register switches between the high-frequency RC oscillator (HSI) and the high-frequency crystal oscillator (HSE OSC) as the main system clock (SCLK). The peripheral clock (PCLK) is always half the frequency of SCLK.

The register bit SW2 in the CLK_CPUCR register switches between PCLK and SCLK to produce the ARM® Cortex-M3 CPU clock (FCLK). The default and preferred mode of operation is to run the CPU at the lower PCLK frequency, 12 MHz, but the higher SCLK frequency, 24 MHz, can be selected to give higher processing performance at the expense of an increase in power consumption.

In addition to these modes, further automatic control is invoked by hardware when Flash programming is enabled. To ensure accuracy of the Flash controller's timers, the FCLK frequency is forced to 12 MHz during Flash programming and erase operations.

Table 7. System clock modes

SW1	**CLK_CPUCR**	**SCLK**	**PCLK**	**f_{CLK}**	
				Flash Program/ Erase Inactive	**Flash Program/ Erase Active**
0 (HSI)	0 (Normal CPU)	12 MHz	6 MHz	6 MHz	12 MHz
0 (HSI)	1 (Fast CPU)	12 MHz	6 MHz	12 MHz	12 MHz
1 (HSE OSC)	0 (Normal CPU)	24 MHz	12 MHz	12 MHz	12 MHz
1 (HSE OSC)	1 (Fast CPU)	24 MHz	12 MHz	24 MHz	12 MHz

6.3.6 Clock switching registers

Clock sleep mode control register (CLK_SLEEPCR)

The sleep timer controls the low power clock gated modes.

Clearing the LSI10KEN bit in the CLK_SLEEPCR register before executing WFI with the SLEEPDEEP bit set to '1' in the SCB_SCR register (for more details refer to the Cortex-M3 Programming manual PM0056) causes deep sleep 2 mode to be entered. Setting the LSI10KEN bit in the CLK_SLEEPCR register causes deep sleep 1 mode to be entered.

Address: 0x4000 0008
Reset value: 0x0000 0002

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														LSI10KEN	LSEEN
														rw	rw

Bits 31:2 Reserved, must be kept at reset value

Bit 1 LSI10KEN:
1: Enables 10 kHz internal RC during deep sleep mode.
2: Disables 10 kHz internal RC during deep sleep mode

Bit 0 LSEEN:
1: Enables 32 kHz external oscillator during deep sleep mode.
2: Disables 32 kHz external oscillator during deep sleep mode.

Low-speed internal 10 KHz clock (LSI10K) control register (CLK_LSI10KCR)

Address: 0x4000 000C
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												TUNE[3:0]			
												rw	rw	rw	rw

Bits 1:4 Reserved, must be kept at reset value

Bits 3:0 TUNE[3:0]:
Tunes the value for the HSI clock.

Low-speed internal 1 KHz clock control register (CLK_LSI1KCR)

Address: 0x4000 0010
Reset value: 0x0000 5000

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**
CALINT[4:0]					CLKFRAC[10:0]										
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	Reserved, must be kept at reset value
Bits 15:11	CALINT[4:0]: Divider value integer portion.
Bits 10:0	CALINT[10:0]: Divider value fractional portion.

High-speed external clock control register 1 (CLK_HSECR1)

Address: 0x4000 4004
Reset value: 0x0000 000F

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												BIASTRIM[3:0]			
												rw	rw	rw	rw

Bits 31:4	Reserved, must be kept at reset value
Bits 3:0	BIASTRIM[3:0]: Bias trim setting for 24-MHz oscillator. Reset to full bias power up. May be overwritten in software.

High-speed internal clock control register (CLK_HSICR)

Address: 0x4000 4008
Reset value: 0x0000 0017

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											TUNE[4:0]				
											rw	rw	rw	rw	rw

Bits 31:5 Reserved, must be kept at reset value

Bits 4:0 TUNE[4:0]:
Frequency trim setting for the high-speed internal oscillator.

High-speed external clock comparator register (CLK_HSECOMPR)

Address: 0x4000 400C
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														HLEVEL	LLEVEL
														r	r

Bits 31:2 Reserved, must be kept at reset value

Bit 1 HLEVEL: High-level comparator output
1: High-level comparator output set.
0: High-level comparator output reset.

Bit 0 LLEVEL: Low-level comparator output
1: Low-level comparator output set.
0: Low-level comparator output reset.

Clock period control register (CLK_PERIODCR)

Address: 0x4000 4010
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														MODE[1:0]	
														rw	rw

Bits 31:2 Reserved, must be kept at reset value

Bits 1:0 MODE[1:0]: Sets the clock to be measured by CLK_PERIOD
3: Not used
2: Measures TUNE_FILTER_RESULT
1: Measures HSI
0: Measures LSI

Clock period status register (CLK_PERIODSR)

Address: 0x4000 4014
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**
PERIOD[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 PERIOD[15:0]: Measures the number of 12-MHz clock cycles in 16 or 256 periods (depending on the MODE bits in the CLK_PERIODCR register) of the selected clock.
16 x 12 MHz clock period in LSI10K mode or 256 x 12 MHz clock period in HSI mode.

Clock dither control register (CLK_DITHERCR)

Address: 0x4000 4018
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															DIS
															rws

Bits 31:1 Reserved, must be kept at reset value

Bit 0 DIS: Dither disable
1: Dither enable
0: Dither disable

High-speed external clock control register 2 (CLK_HSECR2)

Address: 0x4000 401C
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														EN	SW1
														rws	rws

Bits 31:2 Reserved, must be kept at reset value

Bit 1 EN: External high-speed clock enable
When set to “1”, the main clock is 24-MHz HSE OSC.
1: Enables the 24-MHz HSE OSC.
0: Disables the 24-MHz HSE OSC.

Bit 0 SW1: System clock switch
1: HSE (external high-speed clock) is selected.
0: HSI (internal high-speed clock) is selected.

CPU clock control register (CLK_CPUCR)

Address: 0x4000 4020
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															SW2
															rws

Bits 31:1 Reserved, must be kept at reset value

Bit 0 SW2: Switch clock 2
1: 24-MHz CPU clock selected
0: 12-MHz CPU clock selected

Note: Clock selection determines if the RAM controller is running at the same speed as the PCLK (SW2 = ‘1’) or double speed of PCLK (SW2 = ‘0’).

Clock switching (CLK) register map

Table 8 gives the CLK register map and reset values.

Table 8. CLK register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0008	CLK_SLEEPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSI10KEN	LSEEN
	Reset value																															1	0
0x000C	CLK_LSI10KCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TUNE[3:0]			
	Reset value																													0	0	0	0
0x0010	CLK_LSI1KCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALINT[4:0]					CLKFRAC[10:0]										
	Reset value																	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0x0014-0x4000	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x4004	CLK_HSECR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BIASTRIM [3:0]			
	Reset value																													1	1	1	1
0x4008	CLK_HSICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TUNE[4:0]				
	Reset value																												0	1	1	1	1
0x400C	CLK_HSECOMPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HLEVEL	LLEVEL
	Reset value																															0	0
0x4010	CLK_PERIODCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	
	Reset value																															0	0
0x4014	CLK_PERIODSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIOD[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4018	CLK_DITHERCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS
	Reset value																																0
0x401C	CLK_HSECR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN	SW1
	Reset value																															0	0
0x4020	CLK_CPUCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SW2
	Reset value																																0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

6.4 System timers

6.4.1 MAC timer

The STM32W108 devices integrate a 20-bit counter (MACTMR_CNTR register) dedicated to the MAC timer. The counting mode of the MAC timer is controlled using the MACTMR_CR register. This register (MACTMR_CR) integrates two bits: one to enable the counting mode and the other to reset the value of the counter (MACTMR_CNTR register).

6.4.2 Watchdog timer

The STM32W108 integrates a watchdog timer which can be enabled to provide protection against software crashes and ARM® Cortex-M3 CPU lockup. By default, it is disabled at power up of the always-on power domain. The watchdog timer uses the calibrated 1 kHz clock (LSI1K) as its reference and provides a nominal 2.048 s timeout. A low water mark interrupt occurs at 1.792 s and triggers an NMI to the ARM® Cortex-M3 NVIC as an early warning. When enabled, periodically reset the watchdog timer by writing to the WDG_KICKSR register before it expires.

The watchdog timer can be paused when the debugger halts the ARM® Cortex-M3. To enable this functionality, set the bit DBGP bit in the SLPTMR_CR register.

If the low-frequency internal RC oscillator (LSI10K) is turned off during deep sleep, LSI1K stops. As a consequence the watchdog timer stops counting and is effectively paused during deep sleep.

The watchdog enable/disable bits are protected from accidental change by requiring a two step process. To enable the watchdog timer the application must first write the enable code 0xEABE to the WDG_KR register and then set the WDGEN register bit. To disable the timer the application must write the disable code 0xDEAD to the WDG_KR register and then set the WDGDIS register bit.

6.4.3 Sleep timer

The STM32W108 integrates a 32-bit timer dedicated to system timing and waking from sleep at specific times. The sleep timer can use either the calibrated 1 kHz reference(LSI1K), or the 32 kHz crystal clock (LSE). The default clock source is the internal 1 kHz clock. The sleep timer clock source is chosen with the CLKSEL bit in the SLPTMR_CR register.

The sleep timer has a prescaler, a divider of the form 2^N, where N can be programmed from 1 to 2^15. This divider allows for very long periods of sleep to be timed. The timer provides two compare outputs and wrap detection, all of which can be used to generate an interrupt or a wake up event.

The sleep timer is paused when the debugger halts the ARM® Cortex-M3. No additional register bit must be set.

To save current during deep sleep, the low-frequency internal RC oscillator (LSI10K) can be turned off. If LSI10K is turned off during deep sleep and a low-frequency 32.768 kHz crystal oscillator is not being used, then the sleep timer will not operate during deep sleep and sleep timer wake events cannot be used to wakeup the STM32W108.

6.4.4 Event timer

The SysTick timer is an ARM® standard system timer in the NVIC. The SysTick timer can be clocked from either the FCLK (the clock going into the CPU) or the Sleep Timer clock. FCLK is either the SCLK or PCLK as selected by CLK_CPUCR (see *Section 6.3.5: Clock switching on page 53*).

6.4.5 Slow timer (MAC timer, Watchdog, and Sleeptimer) control and status registers

These registers are powered from the always-on power domain.

All registers are only writable when in System mode

MACTimer counter register (MACTMR_CNTR)

Address: 0x4000 2038
Reset value: 0x0000 0000

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

Bits 31:20	Reserved, must be kept at reset value
Bits 19:0	CNT[19:0]: MAC timer counter value

MACTimer counter register (MACTMR_CR)

Address: 0x4000 208C
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														RST	EN
														rw	rw

Bits 31:2	Reserved, must be kept at reset value
Bit 1	RST: MAC timer reset
Bit 0	EN: MAC timer enable

Watchdog control register (WDG_CR)

Register bits for general top level chip functions and protection.

Watchdog bits can only be written after first writing the appropriate code to the WDG_KR register.

Address: 0x4000 6000
Reset value: 0x0000 0002

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														WDG DIS	WDG EN
														rw	rw

Bits 31:2 Reserved, must be kept at reset value

Bit 1 WDGDIS: Watchdog disable

Bit 0 WDGEN: Watchdog enable

Watchdog key register (WDG_KR)

Requires magic number write to arm the watchdog enable or disable function.

Address: 0x4000 6004
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
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 KEY[15:0]:
Write 0xDEAD to disable or 0xEABE to enable.

Watchdog kick-start register (WDG_KICKSR)

Write any value to this register to kick-start the watchdog.

Address: 0x4000 6008
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**
KS[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 KS[15:0]:
Watchdog kick-start value: write any value to restart the watchdog.

Sleep timer configuration register (SLPTMR_CR)

This register sets the various options for the Sleep timer.

Address: 0x4000 600C
Reset value: 0x0000 0400

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			REVERSE	EN	DBGP	Reserved		PSC[3:0]				Reserved			CLK SEL
			rw	rw	rw			rw	rw	rw	rw				rw

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

Bit 12 REVERSE:
0: Count forward
1: Count backwards
Only changes when EN is set to ‘0’

Bit 11 EN:
0: Disable sleep timer
1: Enable sleep timer
To change other register bits (REVERSE, PSC, CLKSEL), this bit must be set to ‘0’.
Enabling/Disabling latency can be up 2 to 3 clock-periods of selected clock.

Bit 10	DBGP: Debug pause 0: The timer continues working in Debug mode. 1: The timer is paused in Debug mode when the CPU is halted.
Bits 9:8	Reserved, must be kept at reset value
Bits 7:4	PSC[3:0]: Sleep timer prescaler setting Divides clock by 2^N where N = 0 to 15. Can only be changed when the EN is set to ‘0’.
Bits 3:1	Reserved, must be kept at reset value
Bit 0	CLKSEL: Clock select 0: Calibrated 1kHz RC clock (default); 1: 32kHz. Can only be changed when the EN is set to ‘0’.

Sleep timer count high register (SLPTMR_CNTH)

Address: 0x4000 6010
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**
CNTH[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16	Reserved, must be kept at reset value
Bits 15:0	CNTH[15:0]: Sleep timer counter high value Reading this register updates the SLEEP_CNTL for subsequent reads.

Sleep timer count low register (SLPTMR_CNTL)

Address: 0x4000 6014
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
CNTL[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 CNTL[15:0]: Sleep timer counter low value
This register is only valid following a read of the SLPTMR_CNTH register.

Sleep timer compare A high register (SLPTMR_CMPAH)

Address: 0x4000 6018
Reset value: 0x0000 FFFF

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
CMPAH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 CMPAH[15:0]: Sleep timer compare A high value
Sleep timer compare value - writing to this register updates the SLPTMR_CMPAH register directly and updates the SLPTMR_CMPAL register from the hold register. This value can only be changed when the sleep timer is disabled (EN bit set to 0 in the SLPTMR_CR register). If the value is changed when the sleep timer is enabled (EN bit set to ‘1’ in the SLPTMR_CR register), a spurious interrupt may be generated. Therefore it is recommended to disable sleep timer interrupts before changing this register.

Sleep timer compare A low register (SLPTMR_CMPAL)

Address: 0x4000 601C
Reset value: 0x0000 FFFF

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**
CMPAL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 CMPAL[15:0]: Sleep timer compare A low value
Writing to this register puts the value in the hold register until a write to the SLPTMR_CMPAH register. The value can only be changed when the sleep timer is disabled (EN bit set to '0' in the SLPTMR_CR register). If the value is changed when the sleep timer is enabled (EN bit set to '1' in the SLPTMR_CR register) a spurious interrupt may be generated. Therefore it is recommended to disable interrupts before changing this register.

Sleep timer compare B high register (SLPTMR_CMPBH)

Address: 0x4000 6020
Reset value: 0x0000 FFFF

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**
CMPBH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 CMPBH[15:0]: Sleep timer compare B high value
Sleep timer compare value - writing to this register updates the SLPTMR_CMPBH register directly and updates the SLPTMR_CMPBL register from the hold register. This value can only be changed when the EN (bit 11 of SLPTMR_CR register) is set to '0'. If the value is changed when the EN bit is set to '1', a spurious interrupt may be generated. Therefore it is recommended to disable interrupts before changing this register.

Sleep timer compare B low register (SLPTMR_CMPBL)

Address:	0x4000 6024
Reset value:	0x0000 FFFF

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**
CMPBL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	Reserved, must be kept at reset value
Bits 15:0	CMPBL[15:0]: Sleep timer compare B low value Writing to this register puts the value in the hold register until a write to the SLPTMR_CMPBH register. The value can only be changed when the sleep timer is enabled (EN bit set to ‘0’ in the SLPTMR_CR register) is set to ‘0’. If the value is changed when the sleep timer is enabled (EN bit set to ‘1’ in the SLPTMR_CR register), a spurious interrupt may be generated. Therefore it is recommended to disable interrupts before changing this register.

Sleep timer interrupt source register (SLPTMR_ISR)

Address:	0x4000 A014
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													CMPB	CMPA	WRAP
													rw	rw	rw

Bits 31:3	Reserved, must be kept at reset value
Bit 2	CMPB: Sleep timer compare B
Bit 1	CMPA: Sleep timer compare A
Bit 0	WRAP: Sleep timer wrap

Sleep timer force interrupt register (SLPTMR_IFR)

Address:	0x4000 A020
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													CMPB	CMPA	WRAP
													rw	rw	rw

Bits 31:3 Reserved, must be kept at reset value

Bit 2 CMPB: Force sleep timer compare B interrupt

Bit 1 CMPA: Force sleep timer compare A interrupt

Bit 0 WRAP: Force sleep timer wrap interrupt

Sleep timer interrupt enable register (SLPTMR_IER)

Address:	0x4000 A054
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													CMPB	CMPA	WRAP
													rw	rw	rw

Bits 31:3 Reserved, must be kept at reset value

Bit 2 CMPB: Sleep timer compare B

Bit 1 CMPA: Sleep timer compare A

Bit 0 WRAP: Sleep timer wrap

MAC timer (MACTMR)/Watchdog (WDG)/Sleeptimer(SLPTMR) register map

Table 9 gives the MACTMR, WDG, and SLPTMR register map and reset values.

Table 9. MACTMR, WDG, and SLPTMR register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2038	MACTMR_CNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[19:0]																			
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x208C	MACTMR_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RST	EN
	Reset value																															0	0
0x6000	WDG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGDIS	WDGEN
	Reset value																															1	0
0x6004	WDG_KR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6008	WDG_KICKSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x600C	SLPTMR_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVERSE	EN	DBGP	Res.	Res.	PSC[3:0]				Res.	Res.	Res.	CLKSEL
	Reset value																				0	0	1			0	0	0	0				0
0x6010	SLPTMR_CNTH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTH[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6014	SLPTMR_CNTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTL[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6018	SLPTMR_CMPAH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPAH[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x601C	SLPTMR_CMPAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPAL[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6020	SLPTMR_CMPBH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPBH[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6024	SLPTMR_CMPBL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPBL[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA014	SLPTMR_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPB	CMPA	WRAP
	Reset value																														0	0	0
0xA020	SLPTMR_IFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPB	CMPA	WRAP
	Reset value																														0	0	0

Table 9. MACTMR, WDG, and SLPTMR register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA054	SLPTMR_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPB	CMPA	WRAP
	Reset value																														0	0	0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

6.5 Power management

The STM32W108's power management system is designed to achieve the lowest deep sleep current consumption possible while still providing flexible wakeup sources, timer activity, and debugger operation. The STM32W108 has four main sleep modes:

- Idle Sleep: Puts the CPU into an idle state where execution is suspended until any interrupt occurs. All power domains remain fully powered and nothing is reset.
- Deep sleep 1: The primary deep sleep state. In this state, the core power domain is fully powered down and the sleep timer is active
- Deep sleep 2: The same as deep sleep 1 except that the sleep timer is inactive to save power. In this mode the sleep timer cannot wakeup the STM32W108.
- Deep sleep 0 (also known as emulated deep sleep): The chip emulates a true deep sleep without powering down the core domain. Instead, the core domain remains powered and all peripherals except the system debug components (ITM, DWT, FPB, NVIC) are held in reset. The purpose of this sleep state is to allow STM32W108 software to perform a deep sleep cycle while maintaining debug configuration such as breakpoints.

6.5.1 Wake sources

When in deep sleep the STM32W108 can be returned to the running state in a number of ways, and the wake sources are split depending on deep sleep 1 or deep sleep 2.

The following wake sources are available in both deep sleep 1 and 2.

- Wake on GPIO activity: Wake due to change of state on any GPIO.
- Wake on serial controller 1: Wake due to a change of state on GPIO Pin PB2.
- Wake on serial controller 2: Wake due to a change of state on GPIO Pin PA2.
- Wake on IRQD: Wake due to a change of state on IRQD. Since IRQD can be configured to point to any GPIO, this wake source is another means of waking on any GPIO activity.
- Wake on setting of CDBGPWRUPREQ: Wake due to setting the CDBGPWRUPREQ bit in the debug port in the SWJ.
- Wake on setting of CSYSPWRUPREQ: Wake due to setting the CSYSPWRUPREQ bit in the debug port in the SWJ.

The following sources are only available in deep sleep 1 since the sleep timer is not active in deep sleep 2.

- Wake on sleep timer compare A.
- Wake on sleep timer compare B.
- Wake on sleep timer wrap.

The following source is only available in deep sleep 0 since the SWJ is required to write memory to set this wake source and the SWJ only has access to some registers in deep sleep 0.

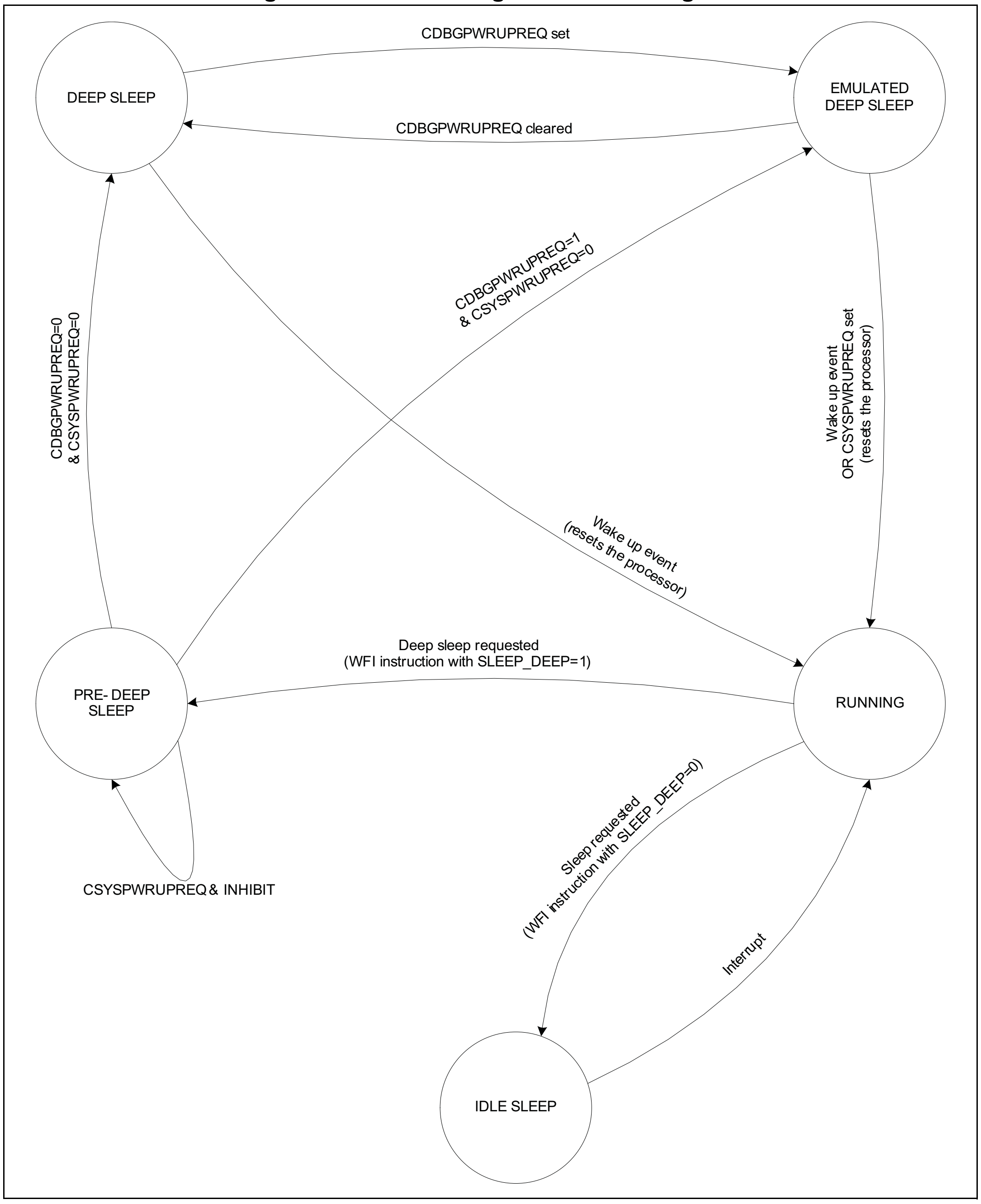
- Wake on write to the COREWAKE bit in the PWR_WAKECR2 register.

The Wakeup Recording module monitors all possible wakeup sources. More than one wakeup source may be recorded because events are continually being recorded (not just in deep-sleep), since another event may happen between the first wake event and when the STM32W108 wakes up.

6.5.2 Basic sleep modes

The power management state diagram in *Figure 6* shows the basic operation of the power management controller.

Figure 6. Power management state diagram


DEEP SLEEP
EMULATED DEEP SLEEP
CDBGPWRUPREQ set
CDBGPWRUPREQ cleared
CDBGPWRUPREQ=1 & CSYSPWRUPREQ=0
CDBGPWRUPREQ=0 & CSYSPWRUPREQ=0
Wake up event OR CSYSPWRUPREQ set (resets the processor)
Wake up event (resets the processor)
Deep sleep requested (WFI instruction with SLEEP_DEEP=1)
PRE- DEEP SLEEP
RUNNING
CSYSPWRUPREQ & INHIBIT
Sleep requested (WFI instruction with SLEEP_DEEP=0)
Interrupt
IDLE SLEEP

In normal operation an application may request one of two low power modes through program execution:

- Idle Sleep is achieved by executing a WFI instruction whilst the SLEEPDEEP bit in the Cortex System Control register (SCS_SCR) is clear (for more details refer to the Cortex-M3 Programming manual PM0056). This puts the CPU into an idle state where execution is suspended until an interrupt occurs. This is indicated by the state at the bottom of the diagram. Power is maintained to the core logic of the STM32W108 during the Idle Sleeping state.
- Deep sleep is achieved by executing a WFI instruction whilst the SLEEPDEEP bit in the Cortex System Control register (SCS_SCR) is set (for more details refer to the Cortex-M3 Programming manual PM0056). This triggers the state transitions around the main loop of the diagram, resulting in powering down the STM32W108's core logic, and leaving only the always-on domain powered. Wake up is triggered when one of the pre-determined events occurs.

If a deep sleep is requested the STM32W108 first enters a pre-deep sleep state. This state prevents any section of the chip from being powered off or reset until the SWJ goes idle (by clearing CSYSPWRUPREQ). This pre-deep sleep state ensures debug operations are not interrupted.

In the deep sleep state the STM32W108 waits for a wake up event which will return it to the running state. In powering up the core logic the ARM® Cortex-M3 is put through a reset cycle and ST software restores the stack and application state to the point where deep sleep was invoked.

6.5.3 Further options for deep sleep

By default, the low-frequency internal RC oscillator (LSI10K) is running during deep sleep (known as deep sleep 1).

To conserve power, LSI10K can be turned off during deep sleep. This mode is known as deep sleep 2. Since the LSI10K is disabled, the sleep timer and watchdog timer do not function and cannot wake the chip unless the low-frequency 32.768 kHz crystal oscillator is used. Non-timer based wake sources continue to function. Once a wake event occurs, the LSI10K restarts and becomes enabled.

6.5.4 Use of debugger with sleep modes

The debugger communicates with the STM32W108 using the SWJ.

When the debugger is connected, the CDBGPWRUPREQ bit in the debug port in the SWJ is set, the STM32W108 will only enter deep sleep 0 (the emulated deep sleep state). The CDBGPWRUPREQ bit indicates that a debug tool is connected to the chip and therefore there may be debug state in the system debug components. To maintain the state in the system debug components only deep sleep 0 may be used, since deep sleep 0 will not cause a power cycle or reset of the core domain. The CSYSPWRUPREQ bit in the debug port in the SWJ indicates that a debugger wants to access memory actively in the STM32W108. Therefore, whenever the CSYSPWRUPREQ bit is set while the STM32W108 is awake, the STM32W108 cannot enter deep sleep until this bit is cleared. This ensures the STM32W108 does not disrupt debug communication into memory.

Clearing both CSYSPWRUPREQ and CDBGPWRUPREQ allows the STM32W108 to achieve a true deep sleep state (deep sleep 1 or 2). Both of these signals also operate as wake sources, so that when a debugger connects to the STM32W108 and begins accessing the chip, the STM32W108 automatically comes out of deep sleep. When the debugger initiates access while the STM32W108 is in deep sleep, the SWJ intelligently holds off the debugger for a brief period of time until the STM32W108 is properly powered and ready.

For more information regarding the SWJ and the interaction of debuggers with deep sleep, contact ST support for Application Notes and ARM® CoreSight documentation.

6.5.5 Power management registers

Power deep sleep control register 1 (PWR_DSLEEPCR1)

Address: 0x4000 0004
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														LVFREEZE	Reserved
														rw	

Bits 31:2 Reserved, must be kept at reset value

Bit 1 LVFREEZE: LV freeze state
1: Enables LV freeze output states
0: Disables GPIO freeze

Bit 0 Reserved, must be kept at reset value

Power deep sleep control register 2 (PWR_DSLEEPCR2)

Address: 0x4000 0014
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															MODE
															rw

Bits 31:1 Reserved, must be kept at reset value

Bit 0 MODE: This bit is used only when the debugger is attached to enable deep sleep mode 0.
1: Enables deep sleep mode 0 when the debugger is attached (default condition).
0: Disables deep sleep mode 0 when the debugger is attached (the CPU is in deep sleep mode 1 or 2).

Power voltage regulator control register (PWR_VREGCR)

Address: 0x4000 0018
Reset value: 0x0000 0204

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				1V8EN	Reserved	1V8TRIM[2:0]			Reserved		1V2EN	Reserved	1V2TRIM[2:0]		
				rws		rw	rw	rw			rws		rw	rw	rw

Bits 31:12 Reserved, must be kept at reset value

Bit 11 1V8EN: 1V8 direct control of regulator on/off
1: 1V8 regulator on
0: 1V8 regulator off

Bit 10 Reserved, must be kept at reset value

Bits 9:7 1V8TRIM: 1V8 regulator trim value

Bits 6:5 Reserved, must be kept at reset value

Bit 4 1V2EN: 1V2 direct control of regulator on/off
1: 1V2 regulator on
0: 1V2 regulator off

Bit 3 Reserved, must be kept at reset value

Bits 2:0 1V2TRIM: 1V2 regulator trim value

Power wakeup event control register 1 (PWR_WAKECR1)

Address: 0x4000 0020
Reset value: 0x0000 0200

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						CRYSP WRUP REQ	CPWR RUP REQ	CORE	WRAP	COMPB	COMP A	IRQD	SC2	SC1	WAK EEN
						rw	rw	rw	rw	rw	rws	rw	rw	rw	rw

Bits 31:10 Reserved, must be kept at reset value

Bit 9 CSYSPWRUPREQ: Wakeup on the CSYSPWRUPREQ event (move to running from deep sleep 0).
1: Enables wakeup on CSYSPWRUPREQ event
0: Disables Wakeup on CSYSPWRUPREQ event

Bit 8 CPWRRUPREQ: Wakeup on the CPWRUPREQ event (move to running from deep sleep 0)
1: Enables wakeup on CPWRUPREQ event
0: Disables wakeup on CPWRUPREQ event

Bit 7 CORE: Wakeup on write to WAKE_CORE bit
1: Enables wakeup on write to WAKE_CORE bit
0: Disables wakeup on write to WAKE_CORE bit

Bit 6 WRAP: Wakeup on sleep timer compare wrap/overflow event
1: Enables wakeup on sleep timer compare wrap/overflow event
0: Disables wakeup on sleep timer compare wrap/overflow event

Bit 5 COMPB: Wake up on sleep timer compare B event
1: Enables wakeup on sleep timer compare B event
0: Disables wakeup on sleep timer compare B event

Bit 4 COMPA: Wakeup on sleep timer compare A event
1: Enables wakeup on sleep timer compare A event
0: Disables wakeup on sleep timer compare A event

Bit 3 IRQD: Wakeup on IRQD event
1: Enables wakeup on IRQD event
0: Disables wakeup on IRQD event

Bit 2 SC2: Wakeup on SC2 event
1: Enables wakeup on SC2 event
0: Disables wakeup on SC2 event

Bit 1 SC1: Wakeup on SC1 event
1: Enables wakeup on SC1 event
0: Disables wakeup on SC1 event

Bit 0 WAKEEN: Enable GPIO wakeup monitoring
1: Enables GPIO wakeup monitoring
0: Disables GPIO wakeup monitoring

Power wakeup event control register 2 (PWR_WAKECR2)

Address: 0x4000 0024
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										CORE WAKE	Reserved				
										w					

Bits 31:10 Reserved, must be kept at reset value

Bit 5 COREWAKE: Power-up controlled by debug port activity. Write to this bit to wake core from deep sleep 0.

Bits 4:0 Reserved, must be kept at reset value

Power wakeup event status register (PWR_WAKESR)

Address: 0x4000 0028
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						CRYSP WRUP REQ	CPWR RUP REQ	CORE	WRAP	COMPB	COMP A	IRQD	SC2	SC1	GPIO PIN
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:10 Reserved, must be kept at reset value

Bit 9 CSYSPWRUPREQ: Indicates that a Debug Access Port (DAP) access to the SYS registers triggered the wake event.
0: Wakeup on CSYSPWRUPREQ event not detected
1: Wakeup on CSYSPWRUPREQ event detected

Bit 8 CPWRRUPREQ: Wake indicates that a DAP access to the DBG registers triggered the wake event.
0: Wakeup on CPWRRUPREQ event not detected
1: Wakeup on CPWRRUPREQ event detected

Bit 7 CORE: Wakeup on debug port activity
0: Wakeup on CORE event not detected
1: Wakeup on CORE event detected

Bit 6 WRAP: Sleep timer wrap
0: Wakeup on WRAP event not detected
1: Wakeup on WRAP event detected

Bit 5 COMPB: Sleep timer compare B
0: Wakeup on COMPB event not detected
1: Wakeup on COMPB event detected

Bit 4 COMPA: Sleep timer compare A
0: Wake up on COMPA event not detected
1: Wake up on COMPA event detected

Bit 3 IRQD: Change of GPIO pin for external interrupt IRQD
0: Wakeup on IRQD event not detected
1: Wakeup on IRQD event detected

Bit 2 SC2: Serial control 2
0: Wakeup on SC2 event not detected
1: Wakeup on SC2 event detected

Bit 1 SC1: Serial control 1
0: Wakeup on SC1 event not detected
1: Wakeup on SC1 event detected

Bit 0 GPIOPIN: Change of programmable GPIO pin (programmable with GPIO wakeup monitoring)
0: Wakeup on GPIO pin not detected
1: Wakeup on GPIO pin detected

Power CPWRUPREQ status register (PWR_CPWRUPREQSR)

Address: 0x4000 0034
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															REQ
															r

Bits 31:1 Reserved, must be kept at reset value

Bit 0 REQ: Status of the SPWRUPREQ

Power CSYSPWRUPREQ status register (PWR_CSYSPWRUPREQSR)

Address: 0x4000 0038
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															REQ
															r

Bits 31:1	Reserved, must be kept at reset value
Bit 0	REQ: Status of the CSYSPWRUPREQ

Power CSYSPWRUPACK status register (PWR_CSYSPWRUPACKSR)

Address: 0x4000 003C
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															ACK
															r

Bits 31:1	Reserved, must be kept at reset value
Bit 0	ACK: Status of the CSYSPWRUPACK

Power CSYSPWRUPACK control register (PWR_CSYSPWRUPACKCR)

Address: 0x4000 0040
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															INHIBIT
															rw

Bits 31:1 Reserved, must be kept at reset value

Bit 0 INHIBIT: Value of CSYSPWRUPACK_INHIBIT (cleared by the power management state machine as part of power-down sequence).
1: Inhibits CSYSPWRUPACK

Power GPIO wakeup monitoring port A register (PWR_WAKEPAR)

Address: 0x4000 BC08
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								PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bit 7 PA7
1: Enables GPIO wakeup on pin GPIO[7] changing state
0: Disables GPIO wakeup on pin GPIO[7] changing state

Bit 6 PA6
1: Enables GPIO wakeup on pin GPIO[6] changing state
0: Disables GPIO wakeup on pin GPIO[6] changing state

Bit 5 PA5
1: Enables GPIO wakeup on pin GPIO[5] changing state
0: Disables GPIO wakeup on pin GPIO[5] changing state

Bit 4 PA4
1: Enables GPIO wakeup on pin GPIO[4] changing state
0: Disables GPIO wakeup on pin GPIO[4] changing state

Bit 3 PA3

1: Enables GPIO wakeup on pin GPIO[3] changing state
0: Disables GPIO wakeup on pin GPIO[3] changing state

Bit 2 PA2

1: Enables GPIO wakeup on pin GPIO[2] changing state
0: Disables GPIO wakeup on pin GPIO[2] changing state

Bit 1 PA1

1: Enables GPIO wakeup on pin GPIO[1] changing state
0: Disables GPIO wakeup on pin GPIO[1] changing state

Bit 0 PA0

1: Enables GPIO wakeup on pin GPIO[0] changing state
0: Disables GPIO wakeup on pin GPIO[0] changing state

Power GPIO wakeup monitoring port B register (PWR_WAKEPBR)

Address:	0x4000 BC0C
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								PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bit 7 PB7

1: Enables GPIO wakeup on pin GPIO[7] changing state
0: Disables GPIO wakeup on pin GPIO[7] changing state

Bit 6 PB6

1: Enables GPIO wakeup on pin GPIO[6] changing state
0: Disables GPIO wakeup on pin GPIO[6] changing state

Bit 5 PB5

1: Enables GPIO wakeup on pin GPIO[5] changing state
0: Disables GPIO wakeup on pin GPIO[5] changing state

Bit 4 PB4

1: Enables GPIO wakeup on pin GPIO[4] changing state
0: Disables GPIO wakeup on pin GPIO[4] changing state

Bit 3 PB3

1: Enables GPIO wakeup on pin GPIO[3] changing state
0: Disables GPIO wakeup on pin GPIO[3] changing state

Bit 2 PB2

1: Enables GPIO wakeup on pin GPIO[2] changing state
0: Disables GPIO wakeup on pin GPIO[2] changing state

Bit 1 PB1

1: Enables GPIO wakeup on pin GPIO[1] changing state
0: Disables GPIO wakeup on pin GPIO[1] changing state

Bit 0 PB0

1: Enables GPIO wakeup on pin GPIO[0] changing state
0: Disables GPIO wakeup on pin GPIO[0] changing state

Power GPIO wakeup monitoring port C register (PWR_WAKEPCR)

Address:	0x4000 BC10
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								PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bit 7 PC7

1: Enables GPIO wakeup on pin GPIO[7] changing state
0: Disables GPIO wakeup on pin GPIO[7] changing state

Bit 6 PC6

1: Enables GPIO wakeup on pin GPIO[6] changing state
0: Disables GPIO wakeup on pin GPIO[6] changing state

Bit 5 PC5

1: Enables GPIO wakeup on pin GPIO[5] changing state
0: Disables GPIO wakeup on pin GPIO[5] changing state

Bit 4 PC4

1: Enables GPIO wakeup on pin GPIO[4] changing state
0: Disables GPIO wakeup on pin GPIO[4] changing state

Bit 3 PC3

1: Enables GPIO wakeup on pin GPIO[3] changing state
0: Disables GPIO wakeup on pin GPIO[3] changing state

Bit 2 PC2

1: Enables GPIO wakeup on pin GPIO[2] changing state
0: Disables GPIO wakeup on pin GPIO[2] changing state

Bit 1 PC1

1: Enables GPIO wakeup on pin GPIO[1] changing state
0: Disables GPIO wakeup on pin GPIO[1] changing state

Bit 0 PC0

1: Enables GPIO wakeup on pin GPIO[0] changing state
0: Disables GPIO wakeup on pin GPIO[0] changing state

Power wakeup filter register (PWR_WAKEFILTR)

Address: 0x4000 BC1C
Reset value: 0x0000 000F

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												IRQD	SC2	SC1	GPIO PIN
												rw	rw	rw	rw

Bits 31:4	Reserved, must be kept at reset value
Bit 3	IRQD_WAKE_FILTER: Enables filter on GPIO wakeup source IRQD
Bit 2	SC2_WAKE_FILTER: Enables filter on GPIO wakeup source SC2 (PA2)
Bit 1	SC1_WAKE_FILTER: Enables filter on GPIO wakeup source SC1 (PB2)
Bit 0	GPIO_WAKE_FILTER: Enables filter on GPIO wakeup sources enabled by the PWR_WAKEPAR, PWR_WAKEPBR, and PWR_WAKEPCR registers.

Power management (PWR) register map

Table 10 gives the PWR register map and reset values.

Table 10. PWR register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0004	PWR_DSLEEPCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVFREEZE	Res.
	Reset value																															0	
0x0008-0x0010	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0014	PWR_DSLEEPCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE
	Reset value																																0
0x0018	PWR_VREGCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	1V8EN	Res.	1V8TRIM [2:0]			Res.	Res.	1V2EN	Res.	1V2TRIM [2:0]		
	Reset value																					0		1	0	0			0		1	1	1
0x001C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Table 10. PWR register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0020	PWR_WAKECR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSYSPWRUPREQ	CPWRRUPREQ	CORE	WRAP	COMPB	COMPA	IRQD	SC2	SC1	WAKEEN
	Reset value																							1	0	0	0	0	0	0	0	0	0
0x0024	PWR_WAKECR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREWAKE	Res.	Res.	Res.	Res.	Res.
	Reset value																											0					
0x0028	PWR_WAKESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSYSPWRUPREQ	CPWRRUPREQ	CORE	WRAP	COMPB	COMPA	IRQD	SC2	SC1	GPIOPIN
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x002C-0x0030	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0034	PWR_CPWRUP REQSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REQ
	Reset value																																0
0x0038	PWR_CSYSPWRUP REQSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REQ
	Reset value																																0
0x003C	PWR_CSYSPWRUP ACKSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACK
	Reset value																																0
0x0040	PWR_CSYSPWRUP ACKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INHIBIT
	Reset value																																0
0x0044-0xBC04	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xBC08	PWR_WAKEPAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Reset value																									0	0	0	0	0	0	0	0
0xBC0C	PWR_WAKEPBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	Reset value																									0	0	0	0	0	0	0	0
0xBC10	PWR_WAKEPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
	Reset value																									0	0	0	0	0	0	0	0
0xBC14-0xBC18	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xBC1C	PWR_WAKEFILTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IRQD	SC2	SC1	GPIOPIN
	Reset value																													0	0	0	0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

6.6 Security accelerator

The STM32W108 contains a hardware AES encryption engine accessible from the ARM® Cortex-M3. NIST-based CCM, CCM, CBC-MAC, and CTR modes are implemented in hardware. These modes are described in the IEEE 802.15.4-2003 specification, with the exception of CCM, which is described in the ZigBee Security Services Specification 1.0.

7 Integrated voltage regulator

The STM32W108 integrates two low dropout regulators to provide 1.8 V and 1.25 V power supplies. The 1V8 regulator supplies the analog and memories, and the 1V25 regulator supplies the digital core. In deep sleep the voltage regulators are disabled.

When enabled, the 1V8 regulator steps down the pads supply voltage (VDD_PADS) from a nominal 3.0 V to 1.8 V. The regulator output pin (VREG_OUT) must be decoupled externally with a suitable capacitor. VREG_OUT should be connected to the 1.8 V supply pins VDDA, VDD_RF, VDD_VCO, VDD_SYNTH, VDD_IF, and VDD_MEM. The 1V8 regulator can supply a maximum of 50 mA.

When enabled, the 1V25 regulator steps down VDD_PADS to 1.25 V. The regulator output pin (VDD_CORE, (Pin 17) must be decoupled externally with a suitable capacitor. It should connect to the other VDD_CORE pin (Pin 44). The 1V25 regulator can supply a maximum of 10 mA.

The regulators are controlled by the digital portion of the chip as described in *Section 6: System modules*.

Table 11. 1.8 V integrated voltage regulator specifications

Parameter	Min.	Typ.	Max.	Units	Comments
Supply range for regulator	2.1		3.6	V	VDD_PADS
1V8 regulator output	-5%	1.8	+5%	V	Regulator output after initialization
1V8 regulator output after reset	-5%	1.75	+5%		Regulator output after reset
1V25 regulator output	-5%	1.25	+5%	V	Regulator output after initialization
1V25 regulator output after reset	-5%	1.45	+5%		Regulator output after reset
1V8 regulator capacitor		2.2		µF	Low ESR tantalum capacitor ESR greater than 2 Ω ESR less than 10 Ω De-coupling less than100 nF ceramic
1V25 regulator capacitor		1.0		µF	Ceramic capacitor (0603)
1V8 regulator output current	0		50	mA	Regulator output current
1V25 regulator output current	0		10	mA	Regulator output current
No load current		600		µA	No load current (bandgap and regulators)
1V8 regulator current limit		200		mA	Short circuit current limit
1V25 regulator current limit		25		mA	Short circuit current limit

Table 11. 1.8 V integrated voltage regulator specifications (continued)

Parameter	Min.	Typ.	Max.	Units	Comments
1V8 regulator start-up time		50		µs	0 V to POR threshold 2.2 µF capacitor
1V25 regulator start-up time		50		µs	0 V to POR threshold 1.0 µF capacitor

An external 1.8 V regulator may replace both internal regulators. The STM32W108 can control external regulators during deep sleep using open-drain GPIO PA7, as described in *Section 8: General-purpose input/output*. The STM32W108 drives PA7 low during deep sleep to disable the external regulator and an external pull-up is required to release this signal to indicate that supply voltage should be provided. Current consumption increases approximately 2 mA when using an external regulator. When using an external regulator the internal regulators should be disabled through software.

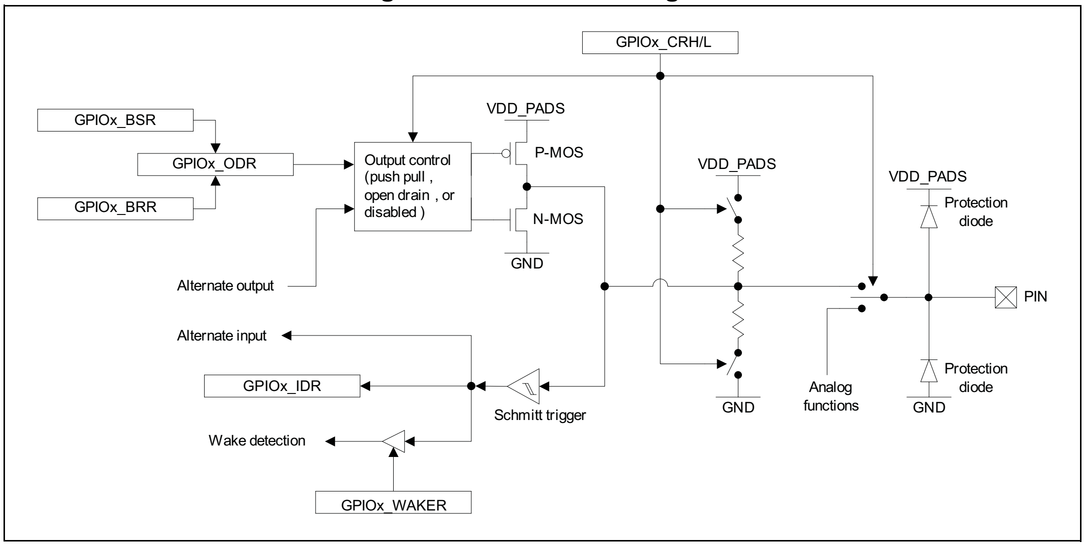
8 General-purpose input/output

The STM32W108 has 24 multi-purpose GPIO pins that may be individually configured as:

- General purpose output
- General purpose open-drain output
- Alternate output controlled by a peripheral device
- Alternate open-drain output controlled by a peripheral device
- Analog
- General purpose input
- General purpose input with pull-up or pull-down resistor

The basic structure of a single GPIO is illustrated in *Figure 7*.

Figure 7. GPIO block diagram



A Schmitt trigger converts the GPIO pin voltage to a digital input value. The digital input signal is then always routed to the GPIOx_IDR register; to the alternate inputs of associated peripheral devices; to wake detection logic if wake detection is enabled; and, for certain pins, to interrupt generation logic. Configuring a pin in analog mode disconnects the digital input from the pin and applies a high logic level to the input of the Schmitt trigger.

Only one device at a time can control a GPIO output. The output is controlled in normal output mode by the GPIOx_ODR register and in alternate output mode by a peripheral device. When in input mode or analog mode, digital output is disabled.

8.1 Functional description

8.1.1 GPIO ports

The 24 GPIO pins are grouped into three ports: PA, PB, and PC. Individual GPIOs within a port are numbered 0 to 7 according to their bit positions within the GPIO registers.

Note: Because GPIO port registers' functions are identical, the notation x is used here to refer to A, B, or C. For example, GPIOx_IDR refers to the registers GPIOA_IDR, GPIOB_IDR, and GPIOC_IDR.

Each of the three GPIO ports has the following registers whose low-order eight bits correspond to the port's eight GPIO pins:

- GPIOx_IDR (input data register) returns the pin level (unless in analog mode).
- GPIOx_ODR (output data register) controls the output level in normal output mode.
- GPIOx_BRR (clear output data register) clears bits in GPIOx_ODR.
- GPIOx_BSR (set output data register) sets bits in GPIOx_ODR.
- PWR_WAKEPxR (wake monitor register) specifies the pins that can wake the STM32W108.

In addition to these registers, each port has a pair of configuration registers, GPIOx_CRH and GPIOx_CRL. These registers specify the basic operating mode for the port's pins. GPIOx_CRL configures the pins CNFMODE3[3:0], CNFMODE2[3:0], CNFMODE1[3:0], and CNFMODE0[3:0]. GPIOx_CRH configures the pins CNFMODE7[3:0], CNFMODE6[3:0], CNFMODE5[3:0], and CNFMODE4[3:0]. Henceforth, the notation GPIOx_CRH/L is used to refer to the pair of configuration registers.

Five GPIO pins (PA6, PA7, PB6, PB7 and PC0) can sink and source higher current than standard GPIO outputs. Refer to *Table 63: Digital I/O characteristics on page 260* for more information.

8.1.2 Configuration

Each pin has a 4-bit configuration value in the GPIOx_CRH/L register. The various GPIO modes and their 4 bit configuration values are shown in *Table 12*.

Table 12. GPIO configuration modes

GPIO mode	GPIOx_CRH/L	Description
Analog	0x0	Analog input or output. When in analog mode, the digital input (GPIOx_IDR) always reads 1.
Input (floating)	0x4	Digital input without an internal pull up or pull down. Output is disabled.
Input (pull-up or pull-down)	0x8	Digital input with an internal pull up or pull down. A set bit in GPIOx_ODR selects pull up and a cleared bit selects pull down. Output is disabled.
Output (push-pull)	0x1	Push-pull output. GPIOx_ODR controls the output.
Output (open-drain)	0x5	Open-drain output. GPIOx_ODR controls the output. If a pull up is required, it must be external.
Alternate Output (push-pull)	0x9	Push-pull output. An onboard peripheral controls the output.
Alternate Output (open-drain)	0xD	Open-drain output. An onboard peripheral controls the output. If a pull up is required, it must be external.
Alternate Output (push-pull) SPI SCLK Mode	0xB	Push-pull output mode only for SPI master mode SCLK pins.

If a GPIO has two peripherals that can be the source of alternate output mode data, then other registers in addition to GPIOx_CRH/L determine which peripheral controls the output.

Several GPIOs share an alternate output with Timer 2 and the Serial Controllers. Bits in Timer 2's TIM2_OR register control routing Timer 2 outputs to different GPIOs. Bits in Timer 2's TIM2_CCER register enable Timer 2 outputs. When Timer 2 outputs are enabled they override Serial Controller outputs. *Table 13* indicates the GPIO mapping for Timer 2 outputs depending on the bits in the register TIM2_OR. Refer to *Section 10: General-purpose timers on page 157* for complete information on timer configuration.

Table 13. Timer 2 output configuration controls

Timer 2 output	Option register bit	GPIO mapping selected by TIM2_OR bit	
		0	1
TIM2_CH1	TIM2_OR[4]	PA0	PB1
TIM2_CH2	TIM2_OR[5]	PA3	PB2
TIM2_CH3	TIM2_OR[6]	PA1	PB3
TIM2_CH4	TIM2_OR[7]	PA2	PB4

For outputs assigned to the serial controllers, the serial interface mode registers (SCx_CR) determine how the GPIO pins are used.

The alternate outputs of PA4 and PA5 can either provide packet trace data (PTI_EN and PTI_DATA), or synchronous CPU trace data (TRACEDATA2 and TRACEDATA3).

If a GPIO does not have an associated peripheral in alternate output mode, its output is set to 0.

8.1.3 Forced functions

For some GPIOs the GPIOx_CRH/L configuration may be overridden. *Table 14* shows the GPIOs that can have different functions forced on them regardless of the GPIOx_CRH/L registers.

Note: The DEBUG_DIS bit in the GPIO_DBGCR register can disable the Serial Wire/JTAG debugger interface. When this bit is set, all debugger-related pins (PC0, PC2, PC3, PC4) behave as standard GPIO.

Table 14. GPIO forced functions

GPIO	Override condition	Forced function	Forced signal
PA7	EXTREGEN bit set in the GPIO_DBGCR register	Open-drain output	REG_EN
PC0	Debugger interface is active in JTAG mode	Input with pull up	JRST
PC2	Debugger interface is active in JTAG mode	Push-pull output	JTDO
PC3	Debugger interface is active in JTAG mode	Input with pull up	JDTI
PC4	Debugger interface is active in JTAG mode	Input with pull up	JTMS
PC4	Debugger interface is active in Serial Wire mode	Bidirectional (push-pull output or floating input) controlled by debugger interface	SWDIO

8.1.4 Reset

A full chip reset is one due to power on (low or high voltage), the NRST pin, the watchdog, or the SYSRESETREQ bit. A full chip reset affects the GPIO configuration as follows:

- The GPIOx_CRH/L configurations of all pins are configured as floating inputs.
- The EXTREGEN bit is set in the GPIO_DBGCR register, which overrides the normal configuration for PA7.
- The DBGDIS bit in the GPIO_DBGCR register is cleared, allowing Serial Wire/JTAG access to override the normal configuration of PC0, PC2, PC3, and PC4.

8.1.5 nBOOTMODE

nBOOTMODE is a special alternate function of PA5 that is active only during a pin reset (NRST) or a power-on-reset of the always-powered domain (POR_HV). If nBOOTMODE is asserted (pulled or driven low) when coming out of reset, the processor starts executing an embedded serial boot loader instead of its normal program.

While in reset and during the subsequent power-on-reset startup delay (512 high-frequency RC oscillator periods), PA5 is automatically configured as an input with a pull-up resistor. At the end of this time, the STM32W108 samples nBOOTMODE: a high level selects normal startup, and a low level selects the boot loader. After nBOOTMODE has been sampled, PA5 is configured as a floating input. The BOOTMODE bit in the GPIO_DBGSR register captures the state of nBOOTMODE so that software may act on this signal if required.

Note: To avoid inadvertently asserting nBOOTMODE, PA5's capacitive load should not exceed 252 pF.

8.1.6 GPIO modes

Analog mode

Analog mode enables analog functions, and disconnects a pin from the digital input and output logic. Only the following GPIO pins have analog functions:

- PA4, PA5, PB5, PB6, PB7, and PC1 can be analog inputs to the ADC.
- PB0 can be an external analog voltage reference input to the ADC, or it can output the internal analog voltage reference from the ADC.
- PC6 and PC7 can connect to an optional 32.768 kHz crystal.

Note: When an external timing source is required, a 32.768 kHz crystal is commonly connected to PC6 and PC7. Alternatively, when PC7 is configured as a digital input, PC7 can accept a digital external clock input.

When configured in analog mode:

- The output drivers are disabled.
- The internal pull-up and pull-down resistors are disabled.
- The Schmitt trigger input is connected to a high logic level.
- Reading GPIOx_IDR returns a constant 1.

Input mode

Input mode is used both for general purpose input and for on-chip peripheral inputs. Input floating mode disables the internal pull-up and pull-down resistors, leaving the pin in a high-impedance state. Input pull-up or pull-down mode enables either an internal pull-up or pull-down resistor based on the GPIOx_ODR register. Setting a bit to 0 in GPIOx_ODR enables the pull-down and setting a bit to 1 enables the pull up.

When configured in input mode:

- The output drivers are disabled.
- An internal pull-up or pull-down resistor may be activated depending on GPIOx_CRH/L and GPIOx_ODR.
- The Schmitt trigger input is connected to the pin.
- Reading GPIOx_IDR returns the input at the pin.
- The input is also available to on-chip peripherals.

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 91*, 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 STM32W108 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 STM32W108 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 STM32W108 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 44* for information on the STM32W108's power management and sleep modes.

8.2 External interrupts

The STM32W108 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 15* shows how the EXTIC_CR and EXTID_CR register values select the GPIO pin used for the external interrupt.

Table 15. IRQC/D GPIO selection

EXTIx_CR	GPIO	EXTIx_CR	GPIO	EXTIx_CR	GPIO
0	PA0	8	PB0	16	PC0
1	PA1	9	PB1	17	PC1
2	PA2	10	PB2	18	PC2
3	PA3	11	PB3	19	PC3
4	PA4	12	PB4	20	PC4
5	PA5	13	PB5	21	PC5
6	PA6	14	PB6	22	PC6
7	PA7	15	PB7	23	PC7

In some cases, it may be useful to assign IRQC or IRQD to an input also in use by a peripheral, for example to generate an interrupt from the slave select signal (nSSEL) in an SPI slave mode interface.

Refer to *Section 12: Interrupts on page 238* for further information regarding the STM32W108 interrupt system.

8.3 Debug control and status

Two GPIO registers are largely concerned with debugger functions. GPIO_DBGCR can disable debugger operation, but has other miscellaneous control bits as well. GPIO_DBGSR, a read-only register, returns status related to debugger activity (FORCEDBG and SWEN), as well a flag (BOOTMODE) indicating whether nBOOTMODE was asserted at the last power-on or NRST-based reset.

8.4 GPIO alternate functions

Table 16 lists the GPIO alternate functions.

Table 16. GPIO signal assignments

GPIO	Analog	Alternate function	Input	Output current drive
PA0		TIM2_CH1(1), SC2MOSI	TIM2_CH1(1), SC2MOSI	Standard
PA1		TIM2_CH3(1), SC2MISO, SC2SDA	TIM2_CH3(1), SC2MISO, SC2SDA	Standard
PA2		TIM2_CH4(1), SC2SCLK, SC2SCL	TIM2_CH4(1), SC2SCLK	Standard
PA3		TIM2_CH2(1), TRACECLK	TIM2_CH2(1), SC2nSSEL	Standard
PA4	ADC4	PTI_EN, TRACEDATA2		Standard

Table 16. GPIO signal assignments (continued)

GPIO	Analog	Alternate function	Input	Output current drive
PA5	ADC5	PTI_DATA, TRACEDATA3	nBOOTMODE(2)	Standard
PA6		TIM1_CH3	TIM1_CH3	High
PA7		TIM1_CH4, REG_EN (3)	TIM1_CH4	High
PB0	VREF	TRACECLK	TIM1CLK, TIM2MSK, IRQA	Standard
PB1		TIM2_CH1(4), SC1TXD, SC1MOSI, SC1MISO, SC1SDA	TIM2_CH1(4), SC1SDA	Standard
PB2		TIM2_CH2(4), SC1SCLK	TIM2_CH2(4), SC1MISO, SC1MOSI, SC1SCL, SC1RXD	Standard
PB3		TIM2_CH3(4), SC1SCLK	TIM2_CH3(4), SC1SCLK, UART_CTS	Standard
PB4		TIM2_CH4(4), UART_RTS	TIM2_CH4(4), SC1nSSEL	Standard
PB5	ADC0		TIM2CLK, TIM1MSK	Standard
PB6	ADC1	TIM1_CH1	TIM1_CH1, IRQB	High
PB7	ADC2	TIM1_CH2	TIM1_CH2	High
PC0		TRACEDATA1	JRST(5)	High
PC1	ADC3	TRACEDATA0, SWO		Standard
PC2		JTDO(6), SWO		Standard
PC3			JTDI(5)	Standard
PC4		SWDIO(7)	SWDIO(7), JTMS(5)	Standard
PC5		TX_ACTIVE		Standard
PC6	OSC32_IN	nTX_ACTIVE		Standard
PC7	OSC32_OUT		OSC32_EXT	Standard

1. Default signal assignment (not remapped).
2. Overrides during reset as an input with pull up.
3. Overrides after reset as an open-drain output.
4. Alternate signal assignment (remapped).
5. Overrides in JTAG mode as an input with pull up.
6. Overrides in JTAG mode as a push-pull output.
7. Overrides in Serial Wire mode as either a push-pull output, or a floating input, controlled by the debugger.

8.5 General-purpose input/output (GPIO) registers

8.5.1 Port x configuration register (Low) (GPIOx_CRL)

Address offset: 0xB000 (GPIOA_CRL), 0xB400 (GPIOB_CRL) and 0xB800 (GPIOC_CRL)
Reset value: 0x0000 4444

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
CNFMODE3[3:0]				CNFMODE2[3:0]				CNFMODE1[3:0]				CNFMODE0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:12 CNFMODE3[3:0]: GPIO configuration control
0x0: Analog, input or output (GPIOx_IDR always reads 1)
0x1: Output, push-pull (GPIOx_ODR controls the output)
0x4: Input, floating
0x5: Output, open-drain (GPIOx_ODR controls the output)
0x8: Input, pulled up or down (selected by GPIOx_ODR: 0 = pull-down, 1 = pull-up)
0x9: Alternate output, push-pull (peripheral controls the output)
0xB: Alternate output SPI SCLK, push-pull (only for SPI master mode SCLK)
0xD: Alternate output, open-drain (peripheral controls the output)

Bits 11:8 CNFMODE2[3:0]: GPIO configuration control and mode
See CNFMODE3 above

Bits 7:4 CNFMODE1[3:0]: GPIO configuration control and mode
See CNFMODE3 above

Bits 3:0 CNFMODE0[3:0]: GPIO configuration control and mode
See CNFMODE3 above

8.5.2 Port x configuration register (High) (GPIOx_CRH)

Address offset: 0xB004 (GPIOA_CRH), 0xB404 (GPIOB_CRH) and 0xB804 (GPIOC_CRH)
Reset value: 0x0000 4444

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**
CNFMODE7[3:0]				CNFMODE6[3:0]				CNFMODE5[3:0]				CNFMODE4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:12 CNFMODE7[3:0]: GPIO configuration control
0x0: Analog, input or output (GPIOx_IDR always reads 1)
0x1: Output, push-pull (GPIOx_ODR controls the output)
0x4: Input, floating
0x5: Output, open-drain (GPIOx_ODR controls the output)
0x8: Input, pulled up or down (selected by GPIOx_ODR: 0 = pull-down, 1 = pull-up)
0x9: Alternate output, push-pull (peripheral controls the output)
0xB: Alternate output SPI SCLK, push-pull (only for SPI master mode SCLK)
0xD: Alternate output, open-drain (peripheral controls the output)

Bits 11:8 CNFMODE6[3:0]: GPIO configuration control and mode
See CNFMODE7 above

Bits 7:4 CNFMODE5[3:0]: GPIO configuration control and mode
See CNFMODE7 above

Bits 3:0 CNFMODE4[3:0]: GPIO configuration control and mode
See CNFMODE7 above

8.5.3 Port x input data register (GPIOx_IDR)

Address offset: 0xB008 (GPIOA_IDR), 0xB408 (GPIOB_IDR) and 0xB808 (GPIOC_IDR)
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								IDRy[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bits 7:0 IDRy[7:0]: Port input data (y = 0...7)

8.5.4 Port x output data register (GPIOx_ODR)

Address offset: 0xB00C (GPIOA_ODR), 0xB40C (GPIOB_ODR)
and 0xB80C (GPIOC_ODR)
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								ODRy[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bits 7:0 ODRy[7:0]: Port output data (y = 0...7)

8.5.5 Port x output set register (GPIOx_BSR)

Address offset: 0xB010 (GPIOA_BSR), 0xB410 (GPIOB_BSR) and 0xB810 (GPIOC_BSR)

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								BSy[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bits 7:0 BSy[7:0]: Port x set bit y (y = 0...7)
0: No action on the corresponding ODRx bit
1: Reset the corresponding ODRx bit

8.5.6 Port x output clear register (GPIOx_BRR)

Address offset: 0xB014 (GPIOA_BRR), 0xB414 (GPIOB_BRR) and 0xB814 (GPIOC_BRR)

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								BRy[7:0]							
								w	w	w	w	w	w	w	w

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

Bits 7:0 BRy[7:0]: Port x reset bit y (y = 0...7)
These bits are write-only and can only be accessed in Word mode.
0: No action on the corresponding ODRx bit
1: Reset the corresponding ODRx bit

8.5.7 External interrupt pending register (EXTI_PR)

Address offset: 0xA814
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												IRQDP	IRQCP	IRQBP	IRQAP
												rw	rw	rw	rw

Bits 31:4 Reserved, must be kept at reset value

Bit 3 IRQDP: EXTI D pending flag

Bit 2 IRQCP: EXTI C pending flag

Bit 1 IRQBP: EXTI B pending flag

Bit 0 IRQAP: EXTI A pending flag

8.5.8 External interrupt x trigger selection register (EXTIx_TSR)

Address offset: 0xA860 (EXTIA_TSR), 0xA864 (EXTIB_TSR), 0xA868 (EXTIC_TSR) and 0xA86C (EXTID_TSR)
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							FILTEN	INTMOD[2:0]			Reserved				
							rw	rw	rw	rw					

Bits 31:9 Reserved, must be kept at reset value

Bit 8 FILTEN:
Set this bit to enable digital filtering on IRQx.

Bits 7:5 INTMOD[2:0]: EXTIx triggering mode
0x0: Disabled
0x1: Rising edge triggered
0x2: Falling edge triggered
0x3: Rising and falling edge triggered.
0x4: Active high level triggered
0x5: Active low level triggered
0x6, 0x7: Reserved, must be kept at reset value

Bits 4:0 Reserved, must be kept at reset value

8.5.9 External interrupt x configuration register (EXTIx_CR)

Address offset: 0xBC14 (EXTIC_CR) and 0xBC18 (EXTID_CR)
Reset value: 0x0000 000F (EXTIC_CR) and 0x0000 0010 (EXTID_CR)

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											GPIO_SEL[4:0]				
											rw	rw	rw	rw	rw

Bits 31:5 Reserved, must be kept at reset value

Bits 4:0 GPIO_SEL[4:0]: Pin assigned to EXTIx

0x00: PA0
0x01: PA1
0x02: PA2
0x03: PA3
0x04: PA4
0x05: PA5
0x06: PA6
0x07: PA7
0x08: PB0
0x09: PB1
0x0A: PB2
0x0B: PB3
0x0C: PB4
0x0D: PB5
0x0E: PB6
0x0F: PB7
0x10: PC0
0x11: PC1
0x12: PC2
0x13: PC3
0x14: PC4
0x15: PC5
0x16: PC6
0x17: PC7
0x18 - 0x1F: Reserved, must be kept at reset value

8.5.10 PC TRACE or debug select register (GPIO_PCTRACECR)

Address offset: 0x4000 4028
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															SEL
															rws

Bits 31:1 Reserved, must be kept at reset value

Bit 0 SEL: Channel encoding
1: PC trace
0: BB debug

8.5.11 GPIO debug configuration register (GPIO_DBGCR)

Address offset: 0xBC00
Reset value: 0x0000 0010

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										DBGDIS	EXTR EGEN	PAD DRIVE	Reserved		
										rw	rw	rw			

Bit 31:6 Reserved, must be kept at reset value

Bit 5 DBGDIS: Disable debug interface override of normal GPIO configuration
0: Permits debug interface to be active.
1: Disables debug interface (if it is not already active).

Bit 4 EXTREGEN: Disable REG_EN override of PA7's normal GPIO configuration
0: Enable override
1: Disable override

Bit 3 PADDRIVE: Global pad drive strength
0: Disables the pad drive strength
1: Enables the pad drive strength

Bit 2:0 Reserved, must be kept at reset value

8.5.12 GPIO debug status register (GPIO_DBGSR)

Address offset: 0xBC04
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												BOOT MODE	Reserved	FORCE DBG	SWEN
												r		r	r

Bit 31:4 Reserved, must be kept at reset value

Bit 3 BOOTMODE: The state of the nBOOTMODE signal sampled at the end of reset
0: nBOOTMODE was not asserted (it read high)
1: nBOOTMODE was asserted (it read low)

Bit 2 Reserved, must be kept at reset value

Bit 1 FORCEDBG: Status of debugger interface
0: Debugger interface not forced active
1: Debugger interface forced active by debugger cable

Bit 0 SWEN: Status of Serial Wire interface
0: Not enabled by SWJ-DP
1: Enabled by SWJ-DP

8.5.13 General-purpose input/output (GPIO) register map

Table 17 gives the GPIO register map and reset values.

Table 17. GPIO register map and reset values

Offset GPIO A/B/C/D	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOA_CRL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNFMODE3 [3:0]				CNFMODE2 [3:0]				CNFMODE1 [3:0]				CNFMODE0 [3:0]			
	Reset value																	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0x04	GPIOx_CRH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNFMODE7 [3:0]				CNFMODE6 [3:0]				CNFMODE5 [3:0]				CNFMODE4 [3:0]			
	Reset value																	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0x08	GPIOx_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
	Reset value																									0	0	0	0	0	0	0	0
0x0C	GPIOx_ODR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
	Reset value																									0	0	0	0	0	0	0	0
0x10	GPIOx_BSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	Reset value																									0	0	0	0	0	0	0	0
0x14	GPIOx_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	Reset value																									0	0	0	0	0	0	0	0
0xA814	EXTI_PR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IRQDP	IRQCP	IRQBP	IRQAP
	Reset value																													0	0	0	0
0xA818-0xA85C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA860	EXTIA_TSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTEN	INT MODE [3:0]			Res.	Res.	Res.	Res.	Res.
	Reset value																								0	0	0	0					
0xA864	EXTIB_TSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTEN	INT MODE [3:0]			Res.	Res.	Res.	Res.	Res.
	Reset value																								0	0	0	0					
0xA868	EXTIC_TSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTEN	INT MODE [3:0]			Res.	Res.	Res.	Res.	Res.
	Reset value																								0	0	0	0					

Table 17. GPIO register map and reset values (continued)

Offset GPIO A/B/C/D	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA86C	EXTID_TSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTEN	INT MODE [3:0]			Res.	Res.	Res.	Res.	Res.
	Reset value																								0	0	0	0					
0xA870-0xBC10	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xBC14	EXTIC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIO_SEL [4:0]				
	Reset value																												0	1	1	1	1
0xBC18	EXTID_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIO_SEL [4:0]				
	Reset value																												0	1	1	1	1
0x4028	GPIO_PCTRACE CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEL
	Reset value																																0
0x402C-0xBBFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xBC00	GPIO_DBGCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGDIS	EXTREGEN	PADDRIVE	Res.	Res.	Res.
	Reset value																											0	1	0			
0xBC04	GPIO_DBGSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOTMODE	Res.	FORCEDBG	SWEN
	Reset value																													0		0	0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

9 Serial interfaces

9.1 Functional description

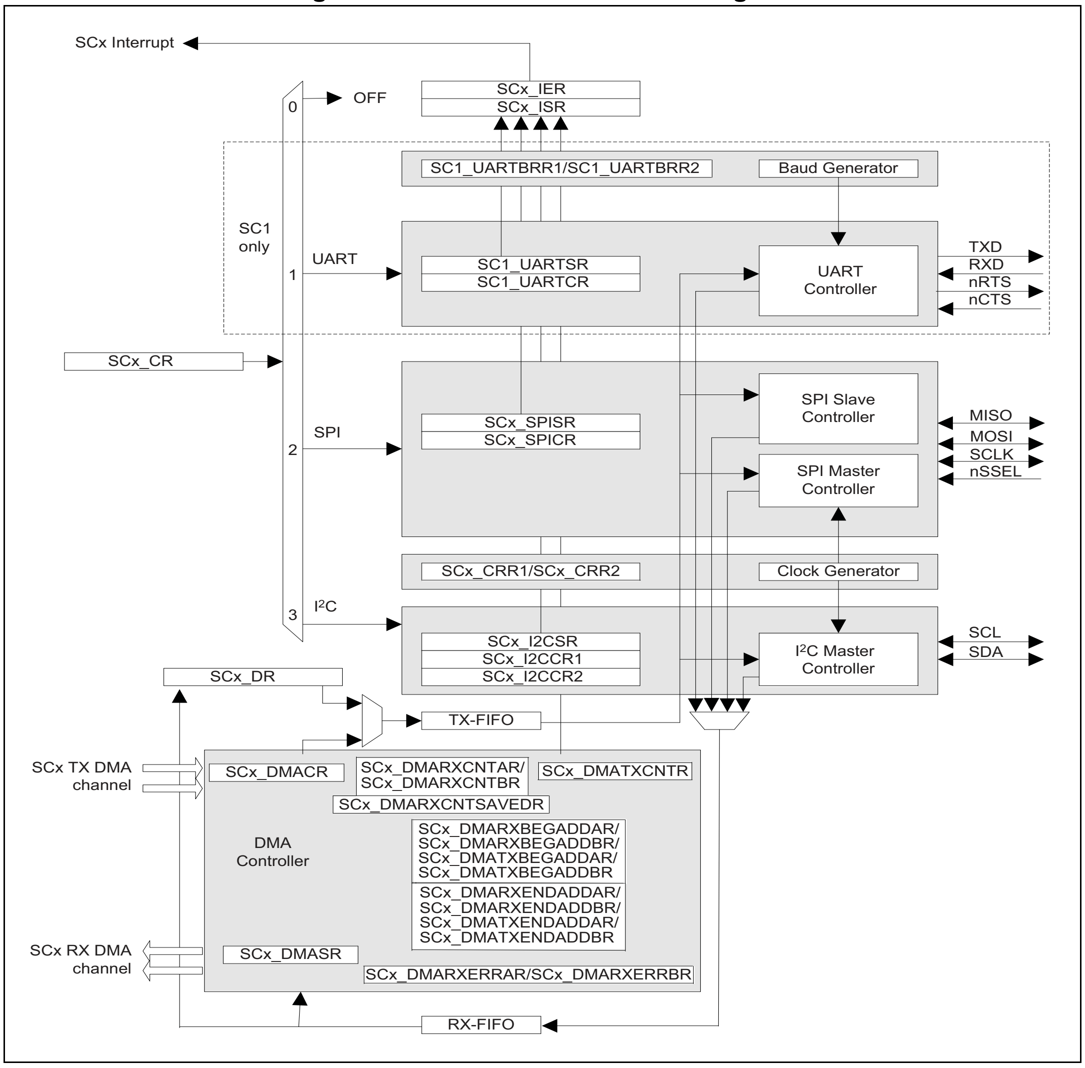
The STM32W108 has two serial controllers, SC1 and SC2, which provide several options for full-duplex synchronous and asynchronous serial communications.

- SPI (Serial Peripheral Interface), master or slave
- I^{2}C (Inter-Integrated Circuit), master only
- UART (Universal Asynchronous Receiver/Transmitter), SC1 only
- Receive and transmit FIFOs and DMA channels, SPI and UART modes

Receive and transmit FIFOs allow faster data speeds using byte-at-a-time interrupts. For the highest SPI and UART speeds, dedicated receive and transmit DMA channels reduce CPU loading and extend the allowable time to service a serial controller interrupt. Polled operation is also possible using direct access to the serial data registers. *Figure 8* shows the components of the serial controllers.

Note: The notation SCx means that either SC1 or SC2 may be substituted to form the name of a specific register or field within a register.

Figure 8. Serial controller block diagram



9.2 Configuration

Before using a serial controller, it should be configured and initialized as follows:

1. Set up the parameters specific to the operating mode (master/slave for SPI, baud rate for UART, etc.).
2. Configure the GPIO pins used by the serial controller as shown in *Table 18* and *Table 19*. *Section 8.1.2: Configuration on page 91* shows how to configure GPIO pins."If using DMA, set up the DMA and buffers. This is described fully in *Section 9.12: Serial controller: Direct memory access (DMA) registers on page 142*.
3. If using interrupts, select edge- or level-triggered interrupts with the SCx_ICR register, enable the desired second-level interrupt sources in the SCx_IER register, and finally enable the top-level SCx interrupt in the NVIC.
4. Write the serial interface operating mode - SPI, I^{2}C, or UART - to the SCx_CR register.

Table 18. SC1 GPIO usage and configuration

Interface	PB1	PB2	PB3	PB4
SPI - Master	SC1MOSI alternate output (push-pull)	SC1MISO input	SC1SCLK alternate output (push-pull); special SCLK mode	(not used)
SPI - Slave	SC1MISO alternate output (push-pull)	SC1MOSI input	SC1SCLK input	SC1nSSEL input
I^2C - Master	SC1SDA alternate output (open-drain)	SC1SCL alternate output (open-drain)	(not used)	(not used)
UART	TXD alternate output (push-pull)	RXD input	nCTS input (1)	nRTS alternate output (push-pull) (1)

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

Table 19. SC2 GPIO usage and configuration

Interface	PA0	PA1	PA2	PA3
SPI - Master	SC2MOSI Alternate Output (push-pull)	SC2MISO Input	SC2SCLK Alternate Output (push-pull), special SCLK mode	(not used)
SPI - Slave	SC2MOSI Alternate Output (push-pull)	SC2MISO Input	SC2SCLK Input	SC2nSSEL Input
I^2C - Master	(not used)	SC2SDA Alternate Output (open-drain)	SC2SCL Alternate Output (open-drain)	(not used)

9.3 SPI master mode

The SPI master controller has the following features:

- Full duplex operation
- Programmable clock frequency (6 MHz max.)
- Programmable clock polarity and phase
- Selectable data shift direction (either LSB or MSB first)
- Receive and transmit FIFOs
- Receive and transmit DMA channels

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 20*. Additional outputs may be needed to drive the nSSEL signals on slave devices.

Table 20. 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'.

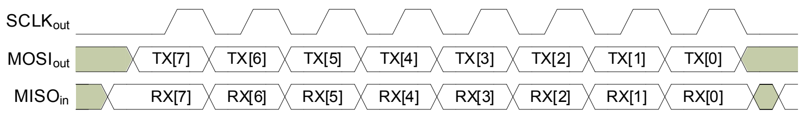
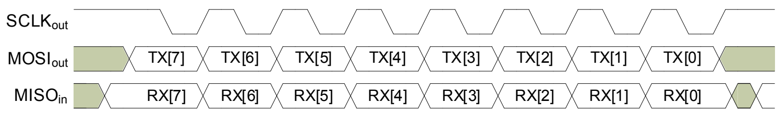
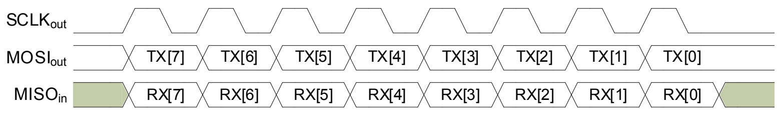
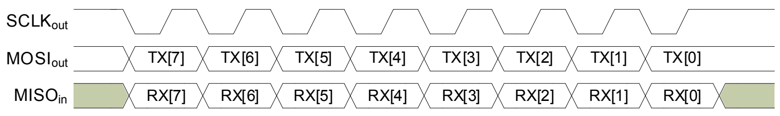
The SPI serial clock (SCLK) is produced by a programmable clock generator. The serial clock is produced by dividing down 12 MHz according to this equation:

$$\text{Rate} = \frac{12\text{MHz}}{(\text{LIN} + 1)\text{x}2^{\text{EXP}}}$$

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 111*). The bits CPOL, CPHA, and LSBFIRST are defined within the SCx_SPICR register.

Table 21. SPI master mode formats

SCx_SPICR				Frame formats
MSTR	LSB FIRST	CPHA	CPOL	
1	0	0	0	 SCLKout MOSIout MISOin TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0] RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0]
1	0	0	1	 SCLKout MOSIout MISOin TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0] RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0]
1	0	1	0	 SCLKout MOSIout MISOin TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0] RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0]
1	0	1	1	 SCLKout MOSIout MISOin TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0] RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0]
1	1	-	-	Same as above except data is sent LSB first instead of MSB first.

9.3.2 Operation

Characters transmitted and received by the SPI master controller are buffered in transmit and receive FIFOs that are both 4 entries deep. When software writes a character to the SCx_DR register, the character is pushed onto the transmit FIFO. Similarly, when software reads from the SCx_DR register, the character returned is pulled from the receive FIFO. If the transmit and receive DMA channels are used, they also write to and read from the transmit and receive FIFOs.

When the transmit FIFO and the serializer are both empty, writing a character to the transmit FIFO clears the IDLE bit in the SCx_SPISR register. This indicates that some characters have not yet been transmitted. If characters are written to the transmit FIFO until it is full, the TXE bit in the SCx_SPISR register is cleared. Shifting out a character to the MOSI pin sets the TXE bit in the SCx_SPISR register. When the transmit FIFO empties and the last character has been shifted out, the IDLE bit in the SCx_SPISR register is set.

Characters received are stored in the receive FIFO. Receiving characters sets the RXNE bit in the SCx_SPISR register, indicating that characters can be read from the receive FIFO. Characters received while the receive FIFO is full are dropped, and the OVF bit in the SCx_SPISR register is set. The receive FIFO hardware generates the OVR, but the DMA register will not indicate the error condition until the receive FIFO is drained. Once the DMA marks a receive error, two conditions will clear the error indication: setting the appropriate TXRST/RXRST bit in the SCx_DMACR register, or loading the appropriate DMA buffer after it has unloaded.

To receive a character, you must transmit a character. If a long stream of receive characters is expected, a long sequence of dummy transmit characters must be generated. To avoid software or transmit DMA initiating these transfers and consuming unnecessary bandwidth, the SPI serializer can be instructed to retransmit the last transmitted character or to transmit a busy token (0xFF), which is determined by the RPTEN bit in the SCx_SPICR register. This functionality can only be enabled or disabled when the transmit FIFO is empty and the transmit serializer is idle, indicated by a cleared IDLE bit in the SCx_SPISR register.

Every time an automatic character transmission starts, a transmit underrun is detected as there is no data in transmit FIFO, and the UDR bit in the SC2_ISR register is set. After automatic character transmission is disabled, no more new characters are received. The receive FIFO holds characters just received.

Note: The Receive DMA complete event does not always mean the receive FIFO is empty.

The DMA Channels section describes how to configure and use the serial receive and transmit DMA channels.

9.3.3 Interrupts

SPI master controller second level interrupts are generated by the following events:

- Transmit FIFO empty and last character shifted out (depending on SCx_ICR, either the 0 to 1 transition or the high level of IDLE)
- Transmit FIFO changed from full to not full (depending on SCx_ICR, either the 0 to 1 transition or the high level of TXE)
- Receive FIFO changed from empty to not empty (depending on SCx_ICR, either the 0 to 1 transition or the high level of RXNE)
- Transmit DMA buffer A/B complete (1 to 0 transition of TXAACK/TXBACK)
- Receive DMA buffer A/B complete (1 to 0 transition of RXAACK/RXBACK)
- 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 SCx_IER register.

9.4 SPI slave mode

Both SC1 and SC2 SPI controllers include a SPI slave controller with these features:

- Full duplex operation
- Up to 5 Mbps data transfer rate
- Programmable clock polarity and clock phase
- Selectable data shift direction (either LSB or MSB first)
- Slave select input

The SPI slave controller uses four signals:

- MOSI (Master Out, Slave In) - inputs serial data from the master
- MISO (Master In, Slave Out) - outputs serial data to the master
- SCLK (Serial Clock) - clocks data transfers on MOSI and MISO
- nSSEL (Slave Select) - enables serial communication with the slave

The GPIO pins that can be assigned to these signals are shown in *Table 22*.

Table 22. SPI slave GPIO usage

Parameter	MOSI	MISO	SCLK	nSSEL
Direction	Input	Output	Input	Input
GPIO configuration	Input	Alternate Output (push-pull)	Input	Input
SC1 pin	PB2	PB1	PB3	PB4
SC2 pin	PA0	PA1	PA2	PA3

9.4.1 Setup and configuration

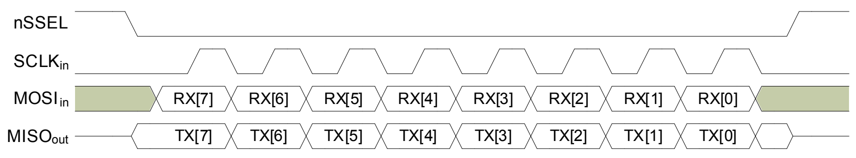
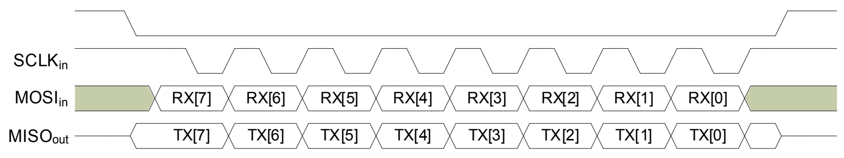
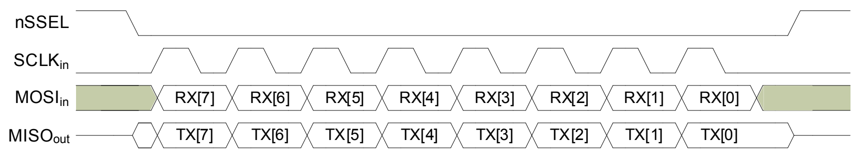
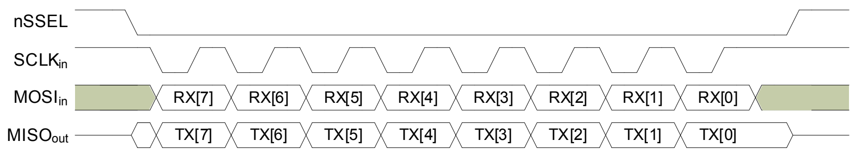
Both serial controllers, SC1 and SC2, support SPI slave mode. SPI slave mode is enabled by the following register settings:

- The serial controller mode register, SCx_CR, is ‘2’.
- The MSTR bit in the SPI configuration register, SCx_SPICR, is ‘0’.

The SPI slave controller receives its clock from an external SPI master device and supports rates up to 5 Mbps.

The SPI slave controller supports various frame formats depending upon the clock polarity (CPOL), clock phase (CPHA), and direction of data (LSBFIRST). The CPOL, CPHA, and LSBFIRST bits are defined within the SCx_SPICR register.

Table 23. SPI slave mode formats

SCx_SPICR				Frame format
MSTR	LSB FIRST	CPHA	CPOL	
0	0	0	0	 nSSEL SCLKin MOSIin MISOout RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0] TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0]
0	0	0	1	 SCLKin MOSIin MISOout RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0] TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0]
0	0	1	0	 nSSEL SCLKin MOSIin MISOout RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0] TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0]
0	0	1	1	 nSSEL SCLKin MOSIin MISOout RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0] TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0]
0	1	-	-	Same as above except LSB first instead of MSB first.

9.4.2 Operation

When the slave select (nSSEL) signal is asserted by the master, SPI transmit data is driven to the output pin MISO, and SPI data is received from the input pin MOSI. The nSSEL pin has to be asserted to enable the transmit serializer to drive data to the output signal MISO. A falling edge on nSSEL resets the SPI slave shift registers.

Characters transmitted and received by the SPI slave controller are buffered in the transmit and receive FIFOs that are both 4 entries deep. When software writes a character to the SCx_DR register, it is pushed onto the transmit FIFO. Similarly, when software reads from the SCx_DR register, the character returned is pulled from the receive FIFO. If the transmit and receive DMA channels are used, the DMA channels also write to and read from the transmit and receive FIFOs.

Characters received are stored in the receive FIFO. Receiving characters sets the RXNE bit in the SCx_SPISR register, to indicate that characters can be read from the receive FIFO. Characters received while the receive FIFO is full are dropped, and the OVF bit in the SCx_SPISR register is set. The receive FIFO hardware generates the OVR interrupt, but the DMA register will not indicate the error condition until the receive FIFO is drained. Once the DMA marks a receive error, two conditions will clear the error indication: setting the appropriate TXRST/RXRST bit in the SCx_DMACR register, or loading the appropriate DMA buffer after it has unloaded.

Receiving a character causes the serial transmission of a character pulled from the transmit FIFO. When the transmit FIFO is empty, a transmit underrun is detected (no data in transmit FIFO) and the UDR bit in the SCx_ISR register is set. Because no character is available for serialization, the SPI serializer retransmits the last transmitted character or a busy token (0xFF), determined by the RPTEN bit in the SCx_SPICR register.

When the transmit FIFO and the serializer are both empty, writing a character to the transmit FIFO clears the IDLE bit in the SCx_SPISR register. This indicates that not all characters have been transmitted. If characters are written to the transmit FIFO until it is full, the TXE bit in the SCx_SPISR register is cleared. Shifting out a transmit character to the MISO pin causes the TXE bit in the SCx_SPISR register to get set. When the transmit FIFO empties and the last character has been shifted out, the IDLE bit in the SCx_SPISR register is set.

The SPI slave controller must guarantee that there is time to move new transmit data from the transmit FIFO into the hardware serializer. To provide sufficient time, the SPI slave controller inserts a byte of padding at the start of every new string of transmit data. After slave select asserts and the RXNE bit in the SCx_SPISR register gets set at least once, the following operation holds true until slave select deasserts. Whenever the transmit FIFO is empty and data is placed into the transmit FIFO, either manually or through DMA, the SPI hardware inserts a byte of padding onto the front of the transmission as if this byte was placed there by software. The value of the byte of padding that is inserted is selected by the RPTEN bit in the SCx_SPICR register.

9.4.3 DMA

The DMA Channels section describes how to configure and use the serial receive and transmit DMA channels.

When using the receive DMA channel and nSSEL transitions to the high (deasserted) state, the active buffer's receive DMA count register (SCx_DMARXCNTAR/SCx_DMARXCNTBR) is saved in the SCx_DMARXCNTSAVEDR register. SCx_DMARXCNTSAVEDR is only written the first time nSSEL goes high after a buffer has been loaded. Subsequent rising edges set a status bit but are otherwise ignored. The 3-bit field NSSS in the SCx_DMASR register records what, if anything, was saved to the SCx_DMARXCNTSAVEDR register, and whether or not another rising edge occurred on nSSEL.

9.4.4 Interrupts

SPI slave controller second level interrupts are generated on the following events:

- Transmit FIFO empty and last character shifted out (depending on SCx_ICR, either the 0 to 1 transition or the high level of IDLE)
- Transmit FIFO changed from full to not full (depending on SCx_ICR, either the 0 to 1 transition or the high level of TXE)
- Receive FIFO changed from empty to not empty (depending on SCx_ICR, either the 0 to 1 transition or the high level of RXNE)
- Transmit DMA buffer A/B complete (1 to 0 transition of TXAACK/TXBACK)
- Receive DMA buffer A/B complete (1 to 0 transition of RXAACK/RXBACK)
- 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 desired interrupt bits in the second level SCx_IER register.

9.5 Inter-integrated circuit interfaces (I^2C)

Both STM32W108 serial controllers SC1 and SC2 include an Inter-integrated circuit interface (I^2C) master controller with the following features:

- Uses only two bidirectional GPIO pins
- Programmable clock frequency (up to 400 kHz)
- Supports both 7-bit and 10-bit addressing
- Compatible with Philips' I^2C-bus slave devices

The I^2C master controller uses just two signals:

- SDA (Serial Data) - bidirectional serial data
- SCL (Serial Clock) - bidirectional serial clock

Table 24 lists the GPIO pins used by the SC1 and SC2 I^2C master controllers. Because the pins are configured as open-drain outputs, they require external pull-up resistors.

Table 24. I^2C Master GPIO Usage

Parameter	SDA	SCL
Direction	Input / Output	Input / Output
GPIO configuration	Alternate Output (open drain)	Alternate Output (open drain)
SC1 pin	PB1	PB2
SC2 pin	PA1	PA2

9.5.1 Setup and configuration

The I^2C controller is enabled by writing 3 to the SCx_CR register. The I^2C controller operates only in master mode and supports both Standard (100 kbps) and Fast (400 kbps) I^2C modes. Address arbitration is not implemented, so multiple master applications are not supported.

The I^2C master controller's serial clock (SCL) is produced by a programmable clock generator. SCL is produced by dividing down 12 MHz according to this equation:

$$\text{Rate} = \frac{12\text{MHz}}{(\text{LIN} + 1)\text{x}2^{\text{EXP}}}$$

EXP is the value written to the SCx_CRR2 register and LIN is the value written to the SCx_CRR1 register. *I2C clock rate programming on page 117* shows the rate settings for Standard-Mode I^2C (100 kbps) and Fast-Mode I^2C (400 kbps) operation.

Table 25. I^2C clock rate programming

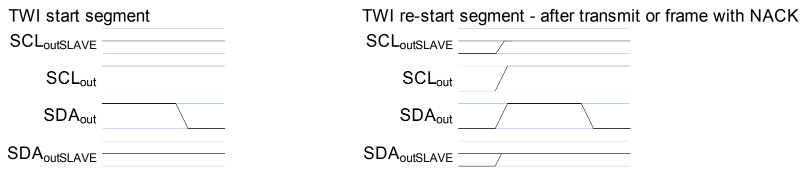
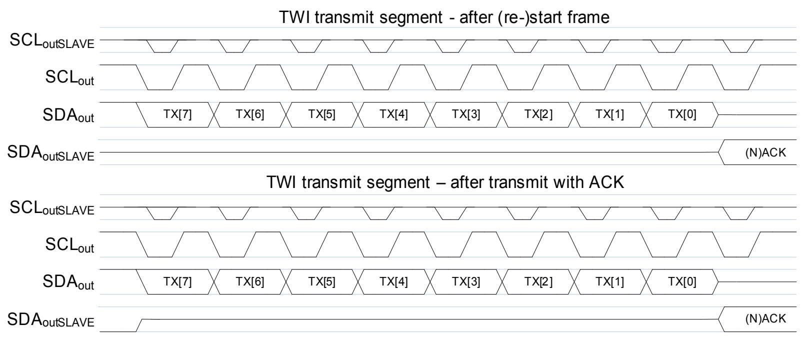
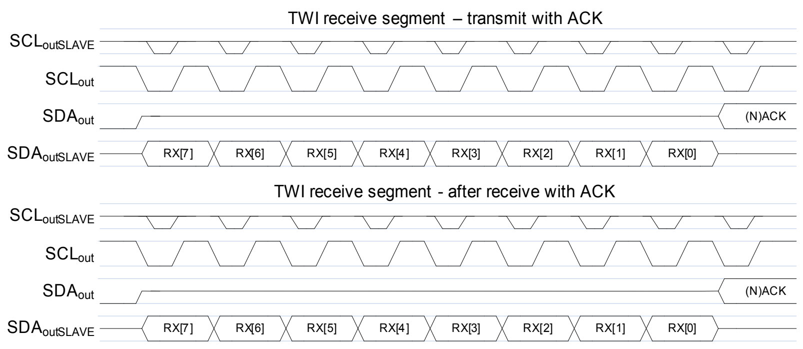
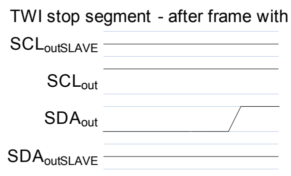
Clock rate	SCx_CRR1	SCx_CRR2
100 kbps	14	3
375 kbps	15	1
400 kbps	14	1

Note: At 400 kbps, the Philips I^2C Bus specification requires the minimum low period of SCL to be 1.3 µs, but on the STM32W108 it is 1.25 µs. If a slave device requires strict compliance with SCL timing, the clock rate must be lowered to 375 kbps.

9.5.2 Constructing frames

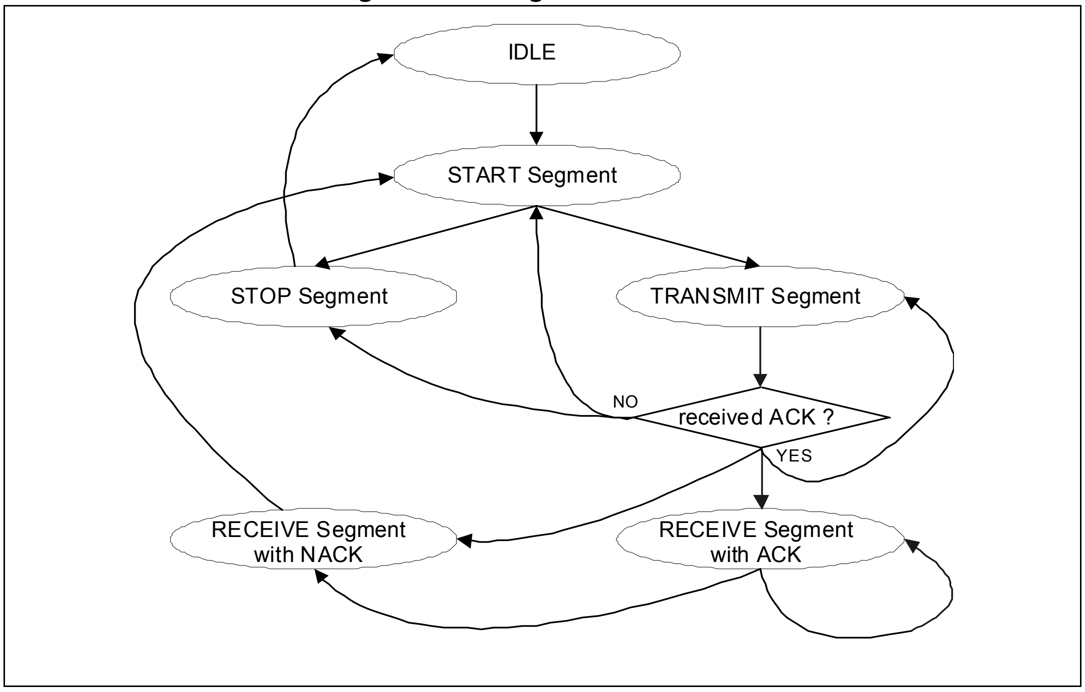
The I^2C master controller supports generating various frame segments by means of the START, STOP, BTE, and BRE bits in the SCx_I2CCR1 registers. *Figure 26* summarizes these frames.

Table 26. I^{2}C master frame segments

SCx_12CCR1				Frame segments
START	BTE	BRE	STOP	
1	0	0	0	 TWI start segment TWI re-start segment - after transmit or frame with NACK $SCL_{outSLAVE}$ SCL_{out} SDA_{out} $SDA_{outSLAVE}$
0	1	0	0	 TWI transmit segment - after (re-)start frame TWI transmit segment – after transmit with ACK $SCL_{outSLAVE}$ SCL_{out} SDA_{out} $SDA_{outSLAVE}$ TX[7] TX[6] TX[5] TX[4] TX[3] TX[2] TX[1] TX[0] (N)ACK
0	0	1	0	 TWI receive segment – transmit with ACK TWI receive segment - after receive with ACK $SCL_{outSLAVE}$ SCL_{out} SDA_{out} $SDA_{outSLAVE}$ (N)ACK RX[7] RX[6] RX[5] RX[4] RX[3] RX[2] RX[1] RX[0]
0	0	0	1	 TWI stop segment - after frame with NACK or stop $SCL_{outSLAVE}$ SCL_{out} SDA_{out} $SDA_{outSLAVE}$
0	0	0	0	No pending frame segment
1 - - 1	1 1 - -	- 1 1 -	- - 1 1	Illegal

Full I^2C frames have to be constructed by software from individual I^2C segments. All necessary segment transitions are shown in *Figure 9*. ACK or NACK generation of an I^2C receive frame segment is determined with the ACK bit in the SCx_I2CCR2 register.

Figure 9. I^2C segment transitions



Generation of a 7-bit address is accomplished with one transmit segment. The upper 7 bits of the transmitted character contain the 7-bit address. The remaining lower bit contains the command type ("read" or "write").

Generation of a 10-bit address is accomplished with two transmit segments. The upper 5 bits of the first transmit character must be set to 0x1E. The next 2 bits are for the 2 most significant bits of the 10-bit address. The remaining lower bit contains the command type ("read" or "write"). The second transmit segment is for the remaining 8 bits of the 10-bit address.

Transmitted and received characters are accessed through the SCx_DR register.

To initiate (re)start and stop segments, set the START or STOP bit in the SCx_I2CCR1 register, then wait until the bit is clear. Alternatively, the CMDFIN bit in the SCx_I2CSR can be used for waiting.

To initiate a transmit segment, write the data to the SCx_DR data register, then set the BTE bit in the SCx_I2CCR1 register, and finally wait until the bit is clear. Alternatively the BTF bit in the SCx_I2CSR register can be used for waiting.

To initiate a receive segment, set the BRE bit in the SCx_I2CCR1 register, wait until it is clear, and then read from the SCx_DR register. Alternatively, the BRF bit in the SCx_ register can be used for waiting. Now the NACK bit in the SCx_I2CSR register indicates if a NACK or ACK was received from an I^2C slave device.

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 STM32W108
- RXD (Received Data) - serial data sent by the STM32W108

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

- nRTS (Request To Send) - indicates the STM32W108 is able to receive data RXD
- nCTS (Clear To Send) - inhibits sending data from the STM32W108 if not asserted

The GPIO pins assigned to these signals are shown in *Table 27*.

Table 27. 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.

9.6.1 Setup and configuration

The UART baud rate clock is produced by a programmable baud generator starting from the 24 Hz clock:

$$baud = \frac{24MHz}{2N + F}$$

The integer portion of the divisor, N, is written to the SC1_UARTBRR1 register and the fractional part, F, to the SC1_UARTBRR2 register. *Table 28* shows the values used to generate some common baud rates and their associated clock frequency error. The UART requires an internal clock that is at least eight times the baud rate clock, so the minimum allowable setting for SC1_UARTBRR1 is ‘8’.

Table 28. UART baud rate divisors for common baud rates

Baud rate (bits/sec)	SC1_UARTBRR1	SC1_UARTBRR2	Baud rate error (%)
300	40000	0	0
2400	5000	0	0
4800	2500	0	0
9600	1250	0	0
19200	625	0	0
38400	312	1	0
57600	208	1	- 0.08
115200	104	0	+ 0.16
230400	52	0	+ 0.16
460800	26	0	+ 0.16
921600	13	0	+ 0.16

Note: The UART may receive corrupt bytes if the interbyte gap is long or there is a baud rate mismatch between receive and transmit. The UART may detect a parity and/or framing error on the corrupt byte, but there will not necessarily be any error detected. As a result, the device should be operated in systems where the other side of the communication link also uses a crystal as its timing reference, and baud rates should be selected to minimize the baud rate mismatch to the crystal tolerance. UART protocols should contain some form of error checking (e.g. CRC) at the packet level to detect, and retry in the event of errors.

The UART character frame format is determined by three bits in the SC1_UARTCR register:

- STOP selects the number of stop bits in transmitted characters. (Only one stop bit is ever required in received characters.) If this bit is clear, characters are transmitted with one stop bit; if set, characters are transmitted with two stop bits.
- PCE controls whether or not received and transmitted characters include a parity bit. If PCE is clear, characters do not contain a parity bit, otherwise, characters do contain a parity bit.
- PS specifies whether transmitted and received parity bits contain odd or even parity. If this bit is clear, the parity bit is even, and if set, the parity bit is odd. Even parity is the exclusive-or of all of the data bits, and odd parity is the inverse of the even parity value. PS has no effect if PCE is clear.

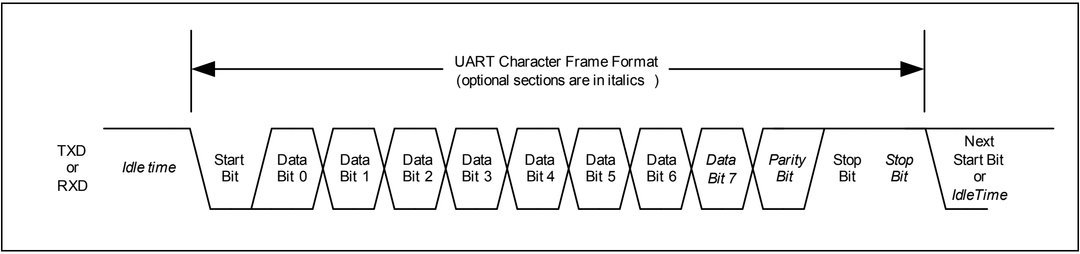
A UART character frame contains, in sequence:

- The start bit
- The least significant data bit
- The remaining data bits
- If parity is enabled, the parity bit
- The stop bit, or bits, if 2 stop bits are selected.

Figure 10 shows the UART character frame format, with optional bits indicated. Depending on the options chosen for the character frame, the length of a character frame ranges from 9 to 12 bit times.

Note that asynchronous serial data may have arbitrarily long idle periods between characters. When idle, serial data (TXD or RXD) is held in the high state. Serial data transitions to the low state in the start bit at the beginning of a character frame.

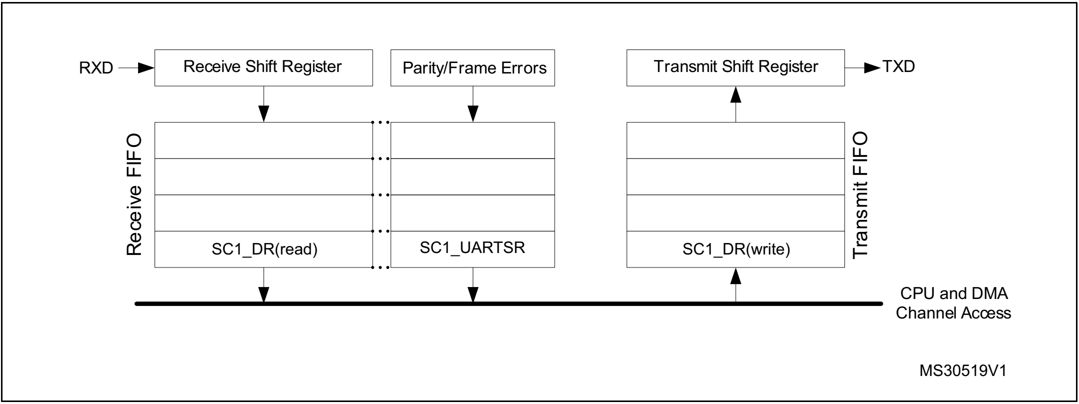
Figure 10. UART character frame format



9.6.2 FIFOs

Characters transmitted and received by the UART are buffered in the transmit and receive FIFOs that are both 4 entries deep (see *Figure 11*). When software writes a character to the SC1_DR register, it is pushed onto the transmit FIFO. Similarly, when software reads from the SC1_DR register, the character returned is pulled from the receive FIFO. If the transmit and receive DMA channels are used, the DMA channels also write to and read from the transmit and receive FIFOs.

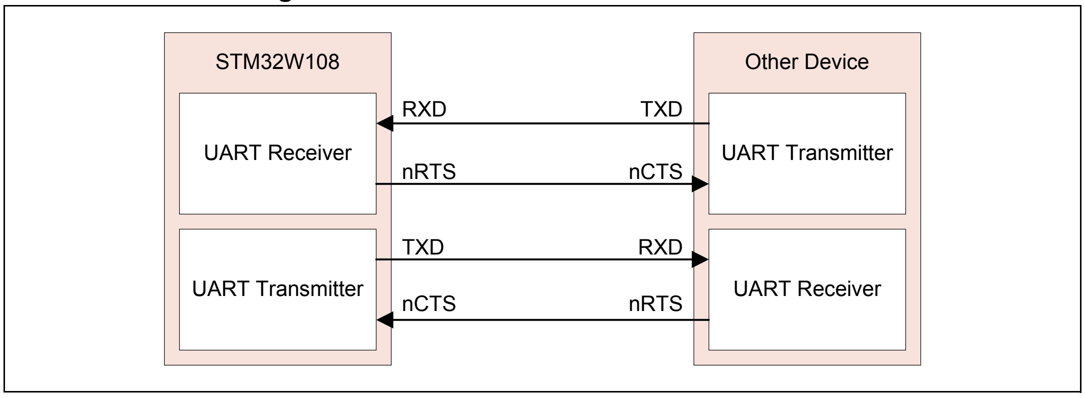
Figure 11. UART FIFOs



9.6.3 RTS/CTS flow control

RTS/CTS flow control, also called hardware flow control, uses two signals (nRTS and nCTS) in addition to received and transmitted data (see *Figure 12*). Flow control is used by a data receiver to prevent buffer overflow, by signaling an external device when it is and is not allowed to transmit.

Figure 12. RTS/CTS flow control connections



The UART RTS/CTS flow control options are selected by the HFCE and AHFCE bits in the SC1_UARTCR register (see *Table 29*). Whenever the HFCE bit is set, the UART will not start transmitting a character unless nCTS is low (asserted). If nCTS transitions to the high state (deasserts) while a character is being transmitted, transmission of that character continues until it is complete.

If the AHFCE bit is set, nRTS is controlled automatically by hardware: nRTS is put into the low state (asserted) when the receive FIFO has room for at least two characters, otherwise is it in the high state (unasserted). If AHFCE is clear, software controls the nRTS output by setting or clearing the nRTS bit in the SC1_UARTCR register. Software control of nRTS is useful if the external serial device cannot stop transmitting characters promptly when nRTS is set to the high state (deasserted).

Table 29. UART RTS/CTS flow control configurations

SC1_UARTCR			Pins used	Operating mode
SC1_UARTxxx[1]				
HFCE	AHFCE	nRTS		
0	-	-	TXD, RXD	No RTS/CTS flow control
1	0	0/1	TXD, RXD, nCTS, nRTS	Flow control using RTS/CTS with software control of nRTS: nRTS controlled by nRTS bit in SC1_UARTCR register
1	1	-	TXD, RXD, nCTS, nRTS	Flow control using RTS/CTS with hardware control of nRTS: nRTS is asserted if room for at least 2 characters in receive FIFO

1. The notation xxx means that the corresponding column header below is inserted to form the field name.

9.6.4 DMA

The DMA Channels section describes how to configure and use the serial receive and transmit DMA channels.

The receive DMA channel has special provisions to record UART receive errors. When the DMA channel transfers a character from the receive FIFO to a buffer in memory, it checks the stored parity and frame error status flags. When an error is flagged, the SC1_DMARXERRAR/SC1_DMARXERRBR register is updated, marking the offset to the first received character with a parity or frame error. Similarly if a receive overrun error occurs, the SC1_DMARXERRAR/SC1_DMARXERRBR registers mark the error offset. The receive FIFO hardware generates the OVR interrupt and DMA status register indicates the error immediately, but in this case the error offset is 4 characters ahead of the actual overflow at the input to the receive FIFO. Two conditions will clear the error indication: setting the appropriate RXRST bit in the SC1_DMACR register, or loading the appropriate DMA buffer after it has unloaded.

9.6.5 Interrupts

UART interrupts are generated on the following events:

- Transmit FIFO empty and last character shifted out (depending on SCx_ICR, either the 0 to 1 transition or the high level of SC1_UARTTXIDLE)
- Transmit FIFO changed from full to not full (depending on SCx_ICR, either the 0 to 1 transition or the high level of SC1_UARTTXFREE)
- Receive FIFO changed from empty to not empty (depending on SCx_ICR, either the 0 to 1 transition or the high level of SC1_UARTRXVAL)
- Transmit DMA buffer A/B complete (1 to 0 transition of TXAACK/TXBACK)
- Receive DMA buffer A/B complete (1 to 0 transition of RXAACK/RXBACK)
- Character received with parity error
- Character received with frame error
- Character received and lost when receive FIFO was full (receive overrun error)

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

9.7 Direct memory access (DMA) channels

The STM32W108 serial DMA channels enable efficient, high-speed operation of the SPI and UART controllers by reducing the load on the CPU as well as decreasing the frequency of interrupts that it must service. The transmit and receive DMA channels can transfer data between the transmit and receive FIFOs and the DMA buffers in main memory as quickly as it can be transmitted or received. Once software defines, configures, and activates the DMA, it only needs to handle an interrupt when a transmit buffer has been emptied or a receive buffer has been filled. The DMA channels each support two memory buffers, labeled A and B, and can alternate ("ping-pong") between them automatically to allow continuous communication without critical interrupt timing.

Note: DMA memory buffer terminology:

- *load - make a buffer available for the DMA channel to use*
- *pending - a buffer loaded but not yet active*
- *active - the buffer that will be used for the next DMA transfer*
- *unload - DMA channel action when it has finished with a buffer*
- *idle - a buffer that has not been loaded, or has been unloaded*

To use a DMA channel, software should follow these steps:

- Reset the DMA channel by setting the TXRST (or RXRST) bit in the SCx_DMACR register.
- Set up the DMA buffers. The two DMA buffers, A and B, are defined by writing the start address to SCx_DMATXBEGADDAR/SCx_DMATXBEGADDBR (or SCx_DMARXBEGADDAR/SCx_DMARXBEGADDBR) and the (inclusive) end address to SCx_DMATXENDADDAR/SCx_DMATXENDADDBR (or SCx_DMARXENDADDAR/SCx_DMARXENDADDBR). Note that DMA buffers must be in RAM.
- Configure and initialize SCx for the desired operating mode.
- Enable second level interrupts triggered when DMA buffers unload by setting the TXULODA/B (or RXULODA/B) bits in the SCx_ISR register.
- Start the DMA by loading the DMA buffers by setting the TXLODA/TXLODB (or RXLODA/RXLODB) bits in the SCx_DMACR register.

A DMA buffer's end address, SCx_DMATXENDADDAR/SCx_DMATXENDADDBR (or SCx_DMARXENDADDAR/SCx_DMARXENDADDBR), can be written while the buffer is loaded or active. This is useful for receiving messages that contain an initial byte count, since it allows software to set the buffer end address at the last byte of the message.

As the DMA channel transfers data between the transmit or receive FIFO and a memory buffer, the DMA count register contains the byte offset from the start of the buffer to the address of the next byte that will be written or read. A transmit DMA channel has a single DMA count register (SCx_DMATXCNTR) that applies to whichever transmit buffer is active, but a receive DMA channel has two DMA count registers (SCx_DMARXCNTAR/SCx_DMARXCNTBR), one for each receive buffer. The DMA count register contents are preserved until the corresponding buffer, or either buffer in the case of the transmit DMA count, is loaded, or until the DMA is reset.

The receive DMA count register may be written while the corresponding buffer is loaded. If the buffer is not loaded, writing the DMA count register also loads the buffer while preserving the count value written. This feature can simplify handling UART receive errors.

The DMA channel stops using a buffer and unloads it when the following is true:

(DMA buffer start address + DMA buffer count) > DMA buffer end address

Typically a transmit buffer is unloaded after all its data has been sent, and a receive buffer is unloaded after it is filled with data, but writing to the buffer end address or buffer count registers can also cause a buffer to unload early.

Serial controller DMA channels include additional features specific to the SPI and UART operation and are described in those sections.

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	TXULODB	TXULODA	RXULODB	RXULODA	NACK	CMDFIN	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^{2}C 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

Bit 7 CMDFIN: I^{2}C command complete detection pending interrupt

This bit is set by hardware when a STOP or START command is generated correctly by the master. It is cleared by software writing it to 0.

0: No CMDFIN detected pending interrupt

1: CMDFIN detected pending interrupt

Bit 6 BTF: I^{2}C byte transmit finished pending interrupt

This bit is set by hardware when the transmit operation is completed.

0: Data byte transmit not done pending interrupt

1: Data byte transmit succeeded pending interrupt

Bit 5 BRF: I^{2}C byte receive finished pending interrupt

This bit is set by hardware when the receive operation is completed.

0: Data byte receive not done pending interrupt

1: Data byte receive succeeded pending interrupt

Bit 4 UDR: Underrun pending interrupt

This bit is set by hardware when data are transmitted and the previous data have not yet left the SCx_DR register.

0: No underrun error pending interrupt

1: Underrun error pending interrupt

Bit 3 OVR: Overrun pending interrupt

This bit is set by hardware when data are received and the previous data have not yet been read from SCx_DR. As a result, the incoming data are lost.

0: No overrun error pending interrupt

1: Overrun error pending interrupt

Bit 2 IDLE: Idle line detected pending interrupt

This bit is set by hardware when an idle line is detected.

0: No idle line detected pending interrupt

1: Idle line detected pending interrupt

Bit 1 TXE: Transmit data register empty (transmitters)

This bit is set by hardware when the SCx_DR register is empty in transmission.

0: Data register not empty pending interrupt

1: Data register empty pending interrupt

Bit 0 RXNE: Data register not empty pending interrupt (receivers)

This bit is set by hardware when the SCx_DR register is not empty in receiver mode.

0: Data is not received pending interrupt

1: Received data is ready to be read pending interrupt

9.8.2 Serial controller interrupt enable register (SCx_IER)

Address offset: 0xA848 (SC1_IER) and 0xA84C (SC2_IER)
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	PEIE	FEIE	TXUL ODRIE	TXUL ODAIE	RXUL ODBIE	RXUL ODAIE	NACK IE	CMD FINIE	BTFIE	BRFIE	UDRIE	OVRIE	IDLEIE	TXEIE	RXNEIE
	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 PEIE: Parity error interrupt enable
This bit is set and cleared by software.
0: Interrupt is inhibited
1: A UART interrupt is generated whenever PE=1 in the SC1_UARTSR register.
Note: Not used in SC2

Bit 13 FEIE: Frame error interrupt enable
This bit is set and cleared by software.
0: Interrupt is inhibited
1: A UART interrupt is generated whenever PE=1 in the SC1_UARTSR register
Note: Not used in SC2

Bit 12 TXULODRIE: DMA transmit buffer B unloaded interrupt enable
This bit is set and cleared by software.
0: Interrupt is inhibited
1: A DMA transmit buffer B unloaded error interrupt is generated whenever TXULOADB=1 in the SC1_UARTSR register.

Bit 11 TXULODAIE: DMA transmit buffer A unloaded interrupt enable
This bit is set and cleared by software.
0: Interrupt is inhibited
1: A DMA transmit buffer A unloaded error interrupt is generated whenever TXULOADA=1 in the SC1_UARTSR register.

Bit 10 RXULODBIE: DMA receive buffer B unloaded interrupt enable
This bit is set and cleared by software.
0: Interrupt is inhibited
1: A DMA receive buffer B unloaded error interrupt is generated whenever RXULOADB=1 in the SCx_ISR register.

Bit 9 RXULODAIE: DMA receive buffer A unloaded interrupt enable
This bit is set and cleared by software.
0: Interrupt is inhibited
1: A DMA receive buffer A unloaded error interrupt is generated whenever RXULOADA=1 in the SCx_ISR register.

Bit 8 NACKIE: I^2C not acknowledge received interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An I^2C not acknowledge interrupt is generated whenever NACK=1 in the SCx_I2CSR register.

Bit 7 CMDFINIE: I^2C command complete detection interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An I^2C command complete interrupt is generated whenever CMDFIN=1 in the SCx_I2CSR register.

Bit 6 BTFIE: I^2C byte transmit finished interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An I^2C byte transmit complete interrupt is generated whenever BTF=1 in the SCx_I2CSR register.

Bit 5 BRFIE: I^2C byte receive finished interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An I^2C byte receive complete interrupt is generated whenever BRF=1 in the SCx_I2CSR register.

Bit 4 UDRIE: Underrun interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An underrun interrupt is generated whenever UND=1 in the SCx_ISR register

Bit 3 OVRIE: Overrun interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An overrun interrupt is generated whenever OVR=1 in the SCx_SPISR register or OVR=1 in the SCx_UARTSR register or OVRA/OVRB in the SCx_DMASR register.

Bit 2 IDLEIE: Line detected interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An idle line detect interrupt is generated whenever IDLE=1 in the SCx_SPISR register or IDLE=1 in the SCx_UARTSR register.

Bit 1 TXEIE: Transmit data register empty interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: A transmit data register empty interrupt is generated whenever TXE=1 in the SCx_SPISR register or TXE=1 in the SCx_UARTSR register.

Bit 0 RXNEIE: Data register not empty interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: A data register not empty interrupt is generated whenever RXNE=1 in the SCx_SPISR register or RXNE=1 in the SCx_UARTSR register.

9.8.3 Serial controller interrupt control register 1 (SCx_ICR)

Address offset: 0xA854 (SC1_ICR) and 0xA858 (SC2_ICR)
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													IDLE LEVEL	TXE LEVEL	RXNE LEVEL
													rw	rw	rw

Bits 31:3 Reserved, must be kept at reset value

Bit 2 IDLELEVEL: Trigger event configuration to generate the IDLE interrupt
This bit is set and cleared by software.
0: Idle interrupt is generated on edge
1: Idle interrupt is generated on level

Bit 1 TXELEVEL: Trigger event configuration to generate the TXE interrupt
This bit is set and cleared by software.
0: TXE interrupt is generated on edge
1: TXE interrupt is generated on level

Bit 0 RNXNELEVEL: Trigger event configuration to generate the RXNE interrupt
This bit is set and cleared by software.
0: RXNE interrupt is generated on edge
1: RXNE interrupt is generated on level

9.8.4 Serial controller data register (SCx_DR)

Address offset: 0xC83C (SC1_DR) and 0xC03C (SC2_DR)
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								DR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bits 7:0 DR[7:0]: Transmit and receive data register
Writing to this register adds a byte to the transmit FIFO. Reading from this register takes the next byte from the receive FIFO and clears the overrun error bit if it was set. In UART mode (SC1 only), reading from this register loads the UART status register with the parity and frame error status of the next byte in the FIFO, and clears these bits if the FIFO is empty.

9.8.5 Serial controller control register 2 (SCx_CR)

Address offset: 0xC854 (SC1_CR) and 0xC054 (SC2_CR)
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														MODE[1:0]	
														rw	rw

Bits 31:2 Reserved, must be kept at reset value

Bits 1:0 MODE[1:0]: Serial controller mode selection
This bit-field specifies the serial control operating mode
00: No mode selected
01: UART mode
10: SPI mode
11: I2C mode
Note: If the UART mode is supported only by SC1

9.8.6 Serial controller clock rate register 1 (SCx_CRR1)

Address offset: 0xC860 (SC1_CRR1) and 0xC060 (SC2_CRR1)
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												LIN[3:0]			
												rw	rw	rw	rw

Bits 31:4 Reserved, must be kept at reset value

Bits 3:0 LIN[3:0]: The linear component of the clock rate in the equation:
Rate = 12 MHz / ((LIN + 1) * (2^EXP))

9.8.7 Serial controller clock rate register 2 (SCx_CRR2)

Address offset: 0xC864 (SC1_CRR2) and 0xC064 (SC2_CRR2)
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												EXP[3:0]			
												rw	rw	rw	rw

Bits 31:4 Reserved, must be kept at reset value

Bits 3:0 EXP[3:0]: The exponential component of the clock rate in the equation:
Rate = 12 MHz / ((LIN + 1) * (2^EXP))

9.9 Serial controller: Serial peripheral interface (SPI) registers

9.9.1 Serial controller SPI status register (SCx_SPISR)

Address offset: 0xC840 (SC1_SPISR) and 0xC040 (SC2_SPISR)
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												IDLE	TXE	RXNE	OVF
												r	r	r	r

Bits 31:4 Reserved, must be kept at reset value

Bit 3 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 SPI idle line is detected
1: SPI idle line is detected

Bit 2 TXE: Transmit data register empty flag (transmitters)
This bit is set by hardware when the transmit FIFO has space to accept at least one byte. An interrupt is generated if TXEIE = 1 in the SCx_IER register.
0: SPI FIFO registers not empty
1: SPI FIFO registers empty

Bit 1 RXNE: Data register not empty flag (receivers)
This bit is set by hardware when the receiver FIFO contains at least one byte. An interrupt is generated if RXNEIE=1 in the SCx_IER register.
0: Data is not received
1: Received data is ready to be read

Bit 0 OVF: Overrun flag
This flag is set by hardware when data are received and the receiver FIFO is full. As a result, the incoming data are lost. It is cleared by software reading the SCx_DR register. An interrupt is generated if OVRIE=1 in the SCx_IER register.
0: No overrun error occurred
1: Overrun error occurred

9.9.2 Serial controller SPI control register (SCx_SPICR)

Address offset: 0xC858 (SC1_SPICR) and 0xC058 (SC2_SPICR)
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										RX MODE	MSTR	RP TEN	LSB FIRST	CPHA	CPOL
										rw	rw	rw	rw	rw	rw

Bits 31:6 Reserved, must be kept at reset value

Bit 5 RXMODE: Receiver-driven mode selection bit (SPI master mode only)
0: Initiate transactions when transmit data is available.
1: Initiate transactions when the receive buffer (FIFO or DMA) has space. Force immediate transmission of busy token or resend last byte (depending on **RPTEN**) and receive data into FIFO until the FIFO is full.

Bit 4 MSTR: Master selection
0: Slave configuration
1: Master configuration

Bit 3 RPTEN: Repeat enable
This bit controls behavior on a transmit buffer underrun condition in slave mode. Clear this bit to send the BUSY token (0xFF) and set this bit to repeat the last byte. Changes to this bit take effect when the transmit FIFO is empty and the transmit serializer is idle.

Bit 2 LSBFIRST: Frame format
0: Most significant bit transmitted first
1: Least significant bit transmitted first

Bit 1 CPHA: Clock phase
0: The first clock transition is the first data capture edge
1: The second clock transition is the first data capture edge

Bit 0 CPOL: Clock polarity
0: CK to 0 when idle
1: CK to 1 when idle

9.10 Serial controller: Inter-integrated circuit (I^2C) registers

9.10.1 Serial controller I^2C status register (SCx_I2CSR)

Address offset: 0xC844 (SC1_I2CSR) and 0xC044 (SC2_I2CSR)
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												CMD FIN	BRF	BTF	NACK
												r	r	r	r

Bits 31:4 Reserved, must be kept at reset value

Bit 3 CMDFIN: Command finished flag

This bit is set when a START or STOP command completes. It is cleared on the next I^2C bus activity.

0: START/ STOP command transmission not done

1: START/ STOP command transmission succeeded

Bit 2 BRF: Byte receive finished flag

This bit is set when a byte is received. It clears on the next I^2C bus activity.

0: Data byte reception not done

1: Data byte reception succeeded

Bit 1 BTF: Byte transfer finished flag

This bit is set when a byte is transmitted. It clears on the next I^2C bus activity.

0: Data byte transmission not done

1: Data byte transmission succeeded

Bit 0 NACK: Not acknowledge flag

This bit is set when a NACK is received from the slave. It clears on the next I^2C bus activity.

0: No NACK received

1: NACK receive succeeded

9.10.2 Serial controller I^2C control register 1 (SCx_I2CCR1)

Address offset: 0xC84C (SC1_I2CCR1) and 0xC04C (SC2_I2CCR1)
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												STOP	START	BTE	BRE
												rw	rw	rw	rw

Bits 31:4 Reserved, must be kept at reset value

Bit 3 STOP: Stop generation
Setting this bit sends the STOP command. It clears when the command completes.
0: No stop condition generation
1: Stop condition generation after current byte transfer

Bit 2 START: Start generation
Setting this bit sends the START or repeated START command. It clears when the command completes.
0: No start generation.
1: Restart/Start generation.

Bit 1 BTE: Byte transmit enable
Setting this bit transmits a byte. It clears when the command completes.
0: Data byte transmission disables
1: Data byte transmission enables

Bit 0 BRE: Byte receive enable
Setting this bit receives a byte. It clears when the command completes.
0: Data byte reception disables
1: Data byte reception enables

9.10.3 Serial controller I^{2}C control register 2 (SCx_I2CCR2)

Address offset: 0xC850 (SC1_I2CCR2) and 0xC050 (SC2_I2CCR2)
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															ACK
															rw

Bits 31:1 Reserved, must be kept at reset value

Bit 0 ACK: Not acknowledge generation
0: A NACK is sent after current received byte
1: An ACK is sent after current received byte

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

Bit 2 TXE: Transmit data register empty flag (transmitters)

This bit is set when the transmit FIFO has space for at least one byte. An interrupt is generated if TXEIE=1 in the SCx_IER register.

0: UART FIFO registers not empty

1: UART FIFO registers empty

Bit 1 RXNE: Receive data register not empty flag (receivers)

This bit is set when the receive FIFO contains at least one byte. An interrupt is generated if RXNEIE=1 in the SCx_IER register.

0: Data is not received

1: Received data is ready to be read

Bit 0 CTS: Clear to send flag

This bit is set by hardware when the nCTS input toggles.

0: No change occurred on the nCTS status line

1: A change occurred on the nCTS status line

9.11.2 Serial controller UART control register (SC1_UARTCR)

Address offset: 0xC85C

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									AHFCE	HFCE	PS	PCE	STOP	M	nRTS
									rw	rw	rw	rw	rw	rw	rw

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

Bit 6 AHFCE: Automatic hardware flow control enable

It is set and cleared by software.

0: Automatic hardware flow control disabled

1: Automatic hardware flow control enabled

Bit 5 HFCE: Hardware flow control enable

It is set and cleared by software.

0: Hardware flow control disabled

1: Hardware flow control enabled

Bit 4 PS: Parity selection

This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software.

0: Even parity

1: Odd parity

Bit 3 PCE: Parity control enable

This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M=1; 8th bit if M=0) and parity is checked on the received data. This bit is set and cleared by software.

0: Parity control disabled.

1: Parity control enabled.

Bit 2 STOP: Number of stop bits t

This bit is used for programming the stop bits.

0: 1 stop bit

1: 2 stop bits

Bit 1 M: Word length

This bit determines the word length. It is set or cleared by software.

0: 1 start bit, 7 data bits, parity bit if enabled, n stop bit

1: 1 start bit, 8 data bits, parity bit if enabled, n stop bit

Bit 0 nRTS: Request to send

This bit controls the flow of the serial data received from another device. This bit directly controls the output at the nRTS pin (HFCE bit must be set and AHFCE bit must be cleared). It is set or cleared by software.

0: nRTS is deasserted (pin is high, 'XOFF', RS232 negative voltage); the other device's transmission is inhibited.

1: nRTS is asserted (pin is low, 'XON', RS232 positive voltage); the other device's transmission is enabled.

9.11.3 Serial controller UART baud rate register 1 (SC1_UARTBRR1)

Address offset: 0xC868
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
N[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31:16 Reserved, must be kept at reset value

Bits 15:0 N[15:0]: The integer part of baud rate period (N) in the equation:

Rate = 24 MHz / ((2 * N) + F)

9.11.4 Serial controller UART baud rate register 2 (SC1_UARTBRR2)

Address offset: 0xC86C
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															F
															rw

Bits 31:1 Reserved, must be kept at reset value

Bit 0 F: The fractional part of the baud rate period (F) in the equation:
Rate = 24 MHz / ((2 * N) + F)

9.12 Serial controller: Direct memory access (DMA) registers

9.12.1 Serial controller receive DMA begin address channel A register (SCx_DMARXBEGADDAR)

Address offset: 0xC800 (SC1_DMARXBEGADDAR) and 0xC000 (SC2_DMARXBEGADDAR)
Reset value: 0x2000 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			ADD[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 ADD[12:0]: DMA receive buffer A start address

9.12.2 Serial controller receive DMA end address channel A register (SCx_DMARXENDADDAR)

Address offset: 0xC804 (SC1_DMARXENDADDAR) and 0xC004 (SC2_DMARXENDADDAR)
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			ADD[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 ADD[12:0]:
Address of the last byte that is written in the DMA receive buffer A.

9.12.3 Serial controller receive DMA begin address channel B register (SCx_ DMARXBEGADDBR)

Address offset: 0xC808 (SC1_DMARXBEGADDBR) and 0xC008 (SC2_DMARXBEGADDBR)
Reset value: 0x2000 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			ADD[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 ADD[12:0]: DMA receive buffer B start address

9.12.4 Serial controller receive DMA end address channel B register (SCx_DMARXENDADDBR)

Address offset: 0xC80C (SC1_DMARXENDADDBR) and 0xC00C (SC2_DMARXENDADDBR)
Reset value: 0x2000 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			ADD[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 ADD[12:0]:
Address of the last byte that is written in the DMA receive buffer B.

9.12.5 Serial controller transmit DMA begin address channel A register (SCx_DMATXBEGADDAR)

Address offset: 0xC810 (SC1_DMATXBEGADDAR) and 0xC010 (SC2_DMATXBEGADDAR)
Reset value: 0x2000 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			ADD[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 ADD[12:0]: DMA transmit buffer A start address

9.12.6 Serial controller transmit DMA end address channel A register (SCx_DMATXENDADDAR)

Address offset: 0xC814 (SC1_DMATXENDADDAR) and 0xC014 (SC2_DMATXENDADDAR)
Reset value: 0x2000 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			ADD[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 ADD[12:0]: Address of the last byte that is read from the DMA transmit buffer A

9.12.7 Serial controller transmit DMA begin address channel B register (SCx_DMATXBEGADDBR)

Address offset: 0xC818 (SC1_DMATXBEGADDBR) and 0xC018 (SC2_DMATXBEGADDBR)
Reset value: 0x2000 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			ADD[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 ADD[12:0]: DMA transmit buffer B start address

9.12.8 Serial controller transmit DMA end address channel B register (SCx_DMATXENDADDBR)

Address offset: 0xC81C (SC1_DMATXENDADDBR) and 0xC01C (SC2_DMATXENDADDBR)
Reset value: 0x2000 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			ADD[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 ADD[12:0]: Address of the last byte that is read from the DMA transmit buffer B

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.

9.12.11 Serial controller transmit DMA counter register (SCx_DMATXCNTR)

Address offset: 0xC828 (SC1_DMATXCNTR) and 0xC028 (SC2_DMATXCNTR)
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]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

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

Bits 12:0 CNT[12:0]:
The offset from the start of the active DMA transmit buffer from which the next byte is read. This register is set to zero when the buffer is loaded and when the DMA is reset.

9.12.12 Serial controller DMA status register (SCx_DMASR)

Address offset: 0xC82C (SC1_DMASR) and 0xC02C (SC2_DMASR)
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			NSSS			FEB	FEA	PEB	PEA	OVRB	OVRA	TX BACK	TX AACK	RX BACK	RX AACK
			r	r	r	r	r	r	r	r	r	r	r	r	r

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

Bits 12:10 NSSS:

Status of the receive count saved in SCx_DMARXCNTSAVEDR (SPI slave mode) when nSSEL deasserts. Cleared when a receive buffer is loaded and when the receive DMA is reset.

0: No count was saved because nSSEL did not deassert
2: Buffer A's count was saved, nSSEL deasserted once
3: Buffer B's count was saved, nSSEL deasserted once
6: Buffer A's count was saved, nSSEL deasserted more than once
7: Buffer B's count was saved, nSSEL deasserted more than once
1, 4, 5: Reserved, must be kept at reset value

Bit 9 FEB: Frame error B flag

This bit is set when DMA receive buffer B reads a byte with a frame error from the receive FIFO. It is cleared the next time buffer B is loaded or when the receive DMA is reset.

0: No DMA buffer B frame error
1: DMA buffer B frame error

Note: Not used in SC2

Bit 8 FEA: Frame error A flag

This bit is set when DMA receive buffer A reads a byte with a frame error from the receive FIFO. It is cleared the next time buffer A is loaded or when the receive DMA is reset.

0: No DMA buffer A frame error
1: DMA buffer A frame error

Note: Not used in SC2

Bit 7 PEB: Parity error B flag

This bit is set when DMA receive buffer B reads a byte with a parity error from the receive FIFO. It is cleared the next time buffer B is loaded or when the receive DMA is reset.

0: No DMA buffer B parity error
1: DMA buffer B parity error

Note: Not used in SC2

Bit 6 PEA: Parity error A flag

This bit is set when DMA receive buffer A reads a byte with a parity error from the receive FIFO. It is cleared the next time buffer A is loaded or when the receive DMA is reset.

0: No DMA buffer A parity error
1: DMA buffer A parity error

Note: Not used in SC2

Bit 5 OVRB: DMA buffer B overrun flag

This bit is set when DMA receive buffer B is passed an overrun error from the receive FIFO. Neither receive buffer is capable of accepting any more bytes (unloaded), and the FIFO fills up. Buffer B is the next buffer to load, and when it drains the FIFO the overrun error is passed up to the DMA and flagged with this bit. It is cleared the next time buffer B is loaded and when the receive DMA is reset.

0: No DMA receive buffer B overrun

1: DMA receive buffer B overrun

Bit 4 OVRA: DMA buffer A overrun flag

This bit is set when DMA receive buffer A is passed an overrun error from the receive FIFO. Neither receive buffer is capable of accepting any more bytes (unloaded), and the FIFO fills up. Buffer A is the next buffer to load, and when it drains the FIFO the overrun error is passed up to the DMA and flagged with this bit. It is cleared the next time buffer A is loaded and when the receive DMA is reset.

0: No DMA receive buffer A overrun

1: DMA receive buffer A overrun

Bit 3 TXBACK: DMA transmit buffer B acknowledge flag

This bit is set/reset by hardware when DMA transmit buffer B is active.

0: DMA transmit buffer B not active

1: DMA transmit buffer B active

Bit 2 TXAACK: DMA transmit buffer A acknowledge flag

This bit is set/reset by hardware when DMA transmit buffer A is active.

0: DMA transmit buffer A not active

1: DMA transmit buffer A active

Bit 1 RXBACK: DMA receive buffer B acknowledge flag

This bit is set/reset by hardware when DMA receive buffer B is active.

0: DMA receive buffer B not active

1: DMA receive buffer B active

Bit 0 RXBACK: DMA receive buffer A acknowledge flag

This bit is set/reset by hardware when DMA receive buffer A is active.

0: DMA receive buffer B not active

1: DMA receive buffer B active

9.12.13 Serial controller DMA control register (SCx_DMACR)

Address offset: 0xC830 (SC1_DMACR) and 0xC030 (SC2_DMACR)
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										TXRST	RXRST	TX LODB	TX LODA	RX LODB	RX LODA
										w	w	rw	rw	rw	rw

Bits 31:6 Reserved, must be kept at reset value

Bit 5 TXRST:

Setting this bit resets the transmit DMA. This bit clears automatically.

Bit 4 RXRST:

Setting this bit resets the receive DMA. This bit clears automatically.

Bit 3 TXLODB:

Setting this bit loads DMA transmit buffer B addresses and allows the DMA controller to start processing transmit buffer B. If both buffer A and B are loaded simultaneously, buffer A is used first. This bit is cleared when DMA completes. Writing a zero to this bit has no effect. Reading this bit returns DMA buffer status:
0: DMA processing is complete or idle
1: DMA processing is active or pending

Bit 2 TXLODA:

Setting this bit loads DMA transmit buffer A addresses and allows the DMA controller to start processing transmit buffer A. If both buffer A and B are loaded simultaneously, buffer A is used first. This bit is cleared when DMA completes. Writing a zero to this bit has no effect. Reading this bit returns DMA buffer status:
0: DMA processing is complete or idle
1: DMA processing is active or pending

Bit 1 RXLODB:

Setting this bit loads DMA receive buffer B addresses and allows the DMA controller to start processing receive buffer B. If both buffer A and B are loaded simultaneously, buffer A is used first. This bit is cleared when DMA completes. Writing a zero to this bit has no effect. Reading this bit returns DMA buffer status:
0: DMA processing is complete or idle
1: DMA processing is active or pending

Bit 0 RXLODA:

Setting this bit loads DMA receive buffer A addresses and allows the DMA controller to start processing receive buffer A. If both buffer A and B are loaded simultaneously, buffer A is used first. This bit is cleared when DMA completes. Writing a zero to this bit has no effect. Reading this bit returns DMA buffer status:
0: DMA processing is complete or idle
1: DMA processing is active or pending

9.12.14 Serial controller receive DMA channel A first error register (SCx_DMARXERRAR)

Address offset: 0xC834 (SC1_DMARXERRAR) and 0xC034 (SC2_DMARXERRAR)
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			ADD[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

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

Bits 12:0 ADD[12:0]:
The offset from the start of DMA receive buffer A of the first byte received with a parity, frame, or overflow error. Note that an overflow error occurs at the input to the receive FIFO, so this offset is 4 bytes before the overflow position. If there is no error, it reads zero. This register is not updated by subsequent errors until the buffer unloads and is reloaded, or the receive DMA is reset.

9.12.15 Serial controller receive DMA channel B first error register (SCx_DMARXERRBR)

Address offset: 0xC838 (SC1_DMARXERRBR) and 0xC038 (SC2_DMARXERRBR)
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			ADD[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

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

Bits 12:0 ADD[12:0]:
The offset from the start of DMA receive buffer B of the first byte received with a parity, frame, or overflow error. Note that an overflow error occurs at the input to the receive FIFO, so this offset is 4 bytes before the overflow position. If there is no error, it reads zero. This register is not updated by subsequent errors until the buffer unloads and is reloaded, or the receive DMA is reset.

9.12.16 Serial controller receive DMA saved counter channel B register (SCx_DMARXCNTSAVEDR)

Address offset: 0xC870 (SC1_DMARXCNTSAVEDR) and 0xC070 (SC2_DMARXCNTSAVEDR)
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]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

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

Bits 12:0 CNT[12:0]:
Receive DMA count saved in SPI slave mode when nSSEL deasserts. The count is only saved the first time nSSEL deasserts.

9.12.17 Serial interface (SC1/SC2) register map

Table 30 gives the SC1/SC2 register map and reset values.

Table 30. SC1/SC2 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA808	SC1_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PE	FE	TXULODB	TXULODA	RXULODB	RXULODA	NACK	CMDFIN	BTF	BRF	UDR	OVR	IDLE	TXE	RXNE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA848	SC1_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEIE	FEIE	TXULODRIE	TXULODAIE	RXULODBIE	RXULODAIE	NACKIE	CMDFINIE	BTFIE	BRFIE	UDRIE	OVRIE	IDLEIE	TXEIE	RXNEIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA854	SC1_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLELEVEL	TXELEVEL	RXNELEVEL
	Reset value																														0	0	0
0xC83C	SC1_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

Table 30. SC1/SC2 register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC854	SC1_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	
	Reset value																															0	0
0xC860	SC1_CRR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LIN[3:0]				
	Reset value																												0	0	0	0	0
0xC864	SC1_CRR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXP[3:0]				
	Reset value																												0	0	0	0	0
0xC840	SC1_SPISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLE	TXE	RXNE	OVF
	Reset value																													0	0	0	0
0xC858	SC1_SPICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXMODE	MSTR	RPTEN	LSBFIRST	CPHA	CPOL
	Reset value																											0	0	0	0	0	0
0xC844	SC1_I2CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDFIN	BRF	BTF	NACK
	Reset value																													0	0	0	0
0xC84C	SC1_I2CRR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP	START	BTE	BRE
	Reset value																													0	0	0	0
0xC850	SC1_I2CRR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACK
	Reset value																																0
0xC848	SC1_UARTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLE	PE	FE	OVR	TXE	RXNE	CTS
	Reset value																										1	0	0	0	0	0	0
0xC85C	SC1_UARTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AHFCE	HFCE	PS	PCE	STOP	M	nRTS
	Reset value																										0	0	0	0	0	0	0
0xC868	SC1_UARTBRR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	N[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xC86C	SC1_UARTBRR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F
	Reset value																																0
0xC800	SC1_DMARX BEGADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC804	SC1_DMARX ENDADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC808	SC1_DMARX BEGADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30. SC1/SC2 register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC80C	SC1_DMARX ENDADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC810	SC1_DMATX BEGADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC814	SC1_DMATX ENDADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC818	SC1_DMATX BEGADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC81C	SC1_DMATX ENDADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC820	SC1_DMARX CNTAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC824	SC1_DMARX CNTBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC828	SC1_DMATX CNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC82C	SC1_DMASR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NSSS [2:0]			FEB	FEA	PEB	PEA	OVRB	OVRA	TXBACK	TXAACK	RXBACK	RXAACK
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC830	SC1_DMACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXRST	RXRST	TXLODB	TXLODA	RXLODB	RXLODA
	Reset value																											0	0	0	0	0	0
0xC834	SC1_DMARX ERRAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC838	SC1_DMARX ERRBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC870	SC1_DMARX CNTSAVEDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xA80C	SC2_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PE	FE	TXULODB	TXULODA	RXULODB	RXULODA	NACK	CMDFIN	BTF	BRF	UDR	OVR	IDLE	TXE	RXNE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA84C	SC2_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEIE	FEIE	TXULODRIE	TXULODAIE	RXULODBIE	RXULODAIE	NACKIE	CMDFINIE	BTFIE	BRFIE	UDRIE	OVRIE	IDLEIE	TXEIE	RXNEIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30. SC1/SC2 register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA858	SC2_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLELEVEL	TXELEVEL	RXNELEVEL
	Reset value																														0	0	0
0xC03C	SC2_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xC054	SC2_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	
	Reset value																															0	0
0xC060	SC2_CRR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LIN[3:0]				
	Reset value																												0	0	0	0	0
0xC064	SC2_CRR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXP[3:0]				
	Reset value																												0	0	0	0	0
0xC040	SC2_SPISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLE	TXE	RXNE	OVF
	Reset value																													0	0	0	0
0xC058	SC2_SPICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXMODE	MSTR	RPTEN	LSBFIRST	CPHA	CPOL
	Reset value																											0	0	0	0	0	0
0xC044	SC2_I2CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDFIN	BRF	BTF	NACK
	Reset value																													0	0	0	0
0xC04C	SC2_I2CRR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP	START	BTE	BRE
	Reset value																													0	0	0	0
0xC050	SC2_I2CRR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACK
	Reset value																																0
0xC000	SC2_DMARX BEGADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC004	SC2_DMARX ENDADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC008	SC2_DMARX BEGADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC00C	SC2_DMARX ENDADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30. SC1/SC2 register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC010	SC2_DMATX BEGADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC014	SC2_DMATX ENDADDAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC018	SC2_DMATX BEGADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC01C	SC2_DMATX ENDADDBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC020	SC2_DMARX CNTAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC024	SC2_DMARX CNTBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC028	SC2_DMATX CNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC02C	SC2_DMASR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NSSS [2:0]			FEB	FEA	PEB	PEA	OVRB	OVRA	TXBACK	TXAACK	RXBACK	RXAACK
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC030	SC2_DMACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXRST	RXRST	TXLODB	TXLODA	RXLODB	RXLODA
	Reset value																											0	0	0	0	0	0
0xC034	SC2_DMARX ERRAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC038	SC2_DMARX ERRBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xC070	SC2_DMARX CNTSAVEDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

10 General-purpose timers

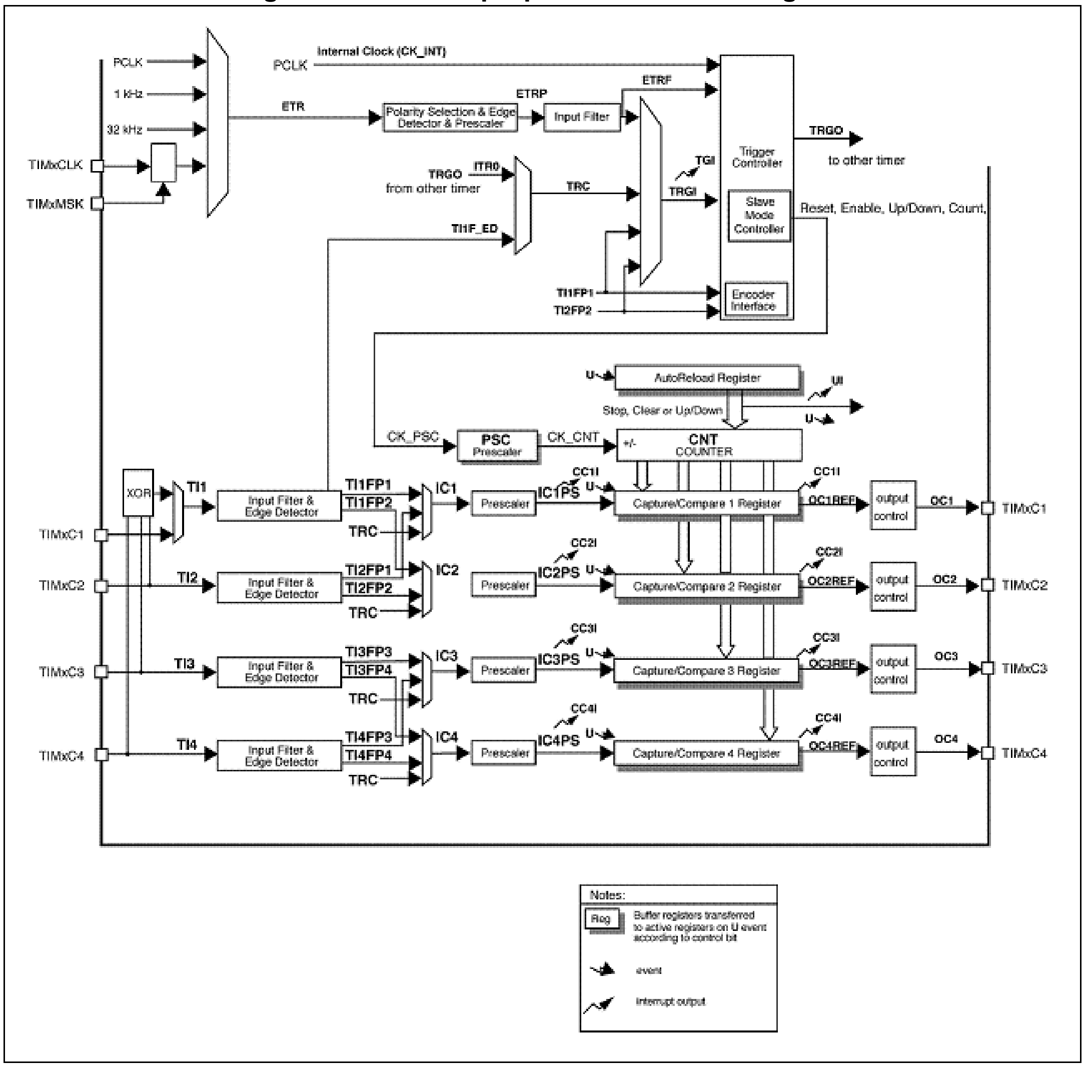
Each of the STM32W108's two general-purpose timers consists of a 16-bit auto-reload counter driven by a programmable prescaler. They may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare and PWM). Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler. The timers are completely independent, and do not share any resources. They can be synchronized together as described in *Section 10.1.14: Timer synchronization on page 187*.

The two general-purpose timers, TIM1 and TIM2, have the following features:

- 16-bit up, down, or up/down auto-reload counter.
- Programmable prescaler to divide the counter clock by any power of two from 1 through 32768.
- 4 independent channels for:
  - Input capture
  - Output compare
- PWM generation (edge- and center-aligned mode)
- One-pulse mode output
- Synchronization circuit to control the timer with external signals and to interconnect the timers.
- Flexible clock source selection:
  - Peripheral clock (PCLK at 6 or 12 MHz)
  - 32 kHz HSE OSC (if available)
  - 1 kHz clock
- GPIO input
- Interrupt generation on the following events:
  - Update: counter overflow/underflow, counter initialization (software or internal/external trigger)
  - Trigger event (counter start, stop, initialization or count by internal/external trigger)
- Input capture
- Output compare
- Supports incremental (quadrature) encoders and Hall sensors for positioning applications.
- Trigger input for external clock or cycle-by-cycle current management.

Note: Because the two timers are identical, the notation TIMx refers to either TIM1 or TIM2. For example, TIMx_PSC refers to both TIM1_PSC and TIM2_PSC. Similarly, "y" refers to any of the four channels of a given timer, so for example, OCy refers to OC1, OC2, OC3, and OC4.

Figure 13. General-purpose timer block diagram



Note: The internal signals shown in Figure 13 are described in Section 10.1.15: Timer signal descriptions on page 193 and are used throughout the text to describe how the timer components are interconnected.

10.1 Functional description

The timers can optionally use GPIOs in the PA and PB ports for external inputs or outputs. As with all STM32W108 digital inputs, a GPIO used as a timer input can be shared with other uses of the same pin. Available timer inputs include an external timer clock, a clock mask, and four input channels. Any GPIO used as a timer output must be configured as an alternate output and is controlled only by the timer.

Many of the GPIOs that can be assigned as timer outputs can also be used by another on-chip peripheral such as a serial controller. Use as a timer output takes precedence over another peripheral function, as long as the channel is configured as an output in the TIMx_CCMR1 register and is enabled in the TIMx_CCER register.

The GPIOs that can be used by Timer 1 are fixed, but the GPIOs that can be used as Timer 2 channels can be mapped to either of two pins, as shown in *Table 31*. The Timer 2 Option Register (TIM2_OR) has four single bit fields (REMAPCy) that control whether a Timer 2 channel is mapped to its default GPIO in port PA, or remapped to a GPIO in PB.

Table 31 specifies the pins that may be assigned to Timer 1 and Timer 2 functions.

Table 31. Timer GPIO use

Signal (direction)	TIMxC1 (in or out)	TIMxC2 (in or out)	TIMxC3 (in or out)	TIMxC4 (in or out)	TIMxCLK (in)	TIMxMSK (in)
Timer 1	PB6	PB7	PA6	PA7	PB0	PB5
Timer 2 (REMAPCy = 0)	PA0	PA3	PA1	PA2	PB5	PB0
Timer 2 (REMAPCy = 1)	PB1	PB2	PB3	PB4	PB5	PB0

The TIMxCLK and TIMxMSK inputs can be used only in the external clock modes: refer to the External Clock Source Mode 1 and External Clock Source Mode 2 sections for details concerning their use.

10.1.1 Time-base unit

The main block of the general purpose timer is a 16-bit counter with its related auto-reload register. The counter can count up, down, or alternate 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 to 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)

Some timer registers cannot be directly accessed by software, which instead reads and writes a "buffer register". The internal registers actually used for timer operations are called "shadow registers".

The auto-reload register is buffered. Writing to or reading from the auto-reload register accesses the buffer register. The contents of the buffer register are transferred into the shadow register permanently or at each update event (UEV), depending on the auto-reload buffer enable bit (ARPE) in the TIMx_CR1 register. The update event is generated when both the counter reaches the overflow (or underflow when down-counting) and when the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software. Update event generation is described in detail for each configuration.

The counter is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) in the TIMx_CR1 register is set. Refer also to the slave mode controller description in the Timers and External Trigger Synchronization section to get more details on counter enabling.

Note that the actual counter enable signal CNT_EN is set one clock cycle after CEN.

Note: When the STM32W108 enters debug mode and the ARM® Cortex-M3 core is halted, the counters continue to run normally.

Prescaler

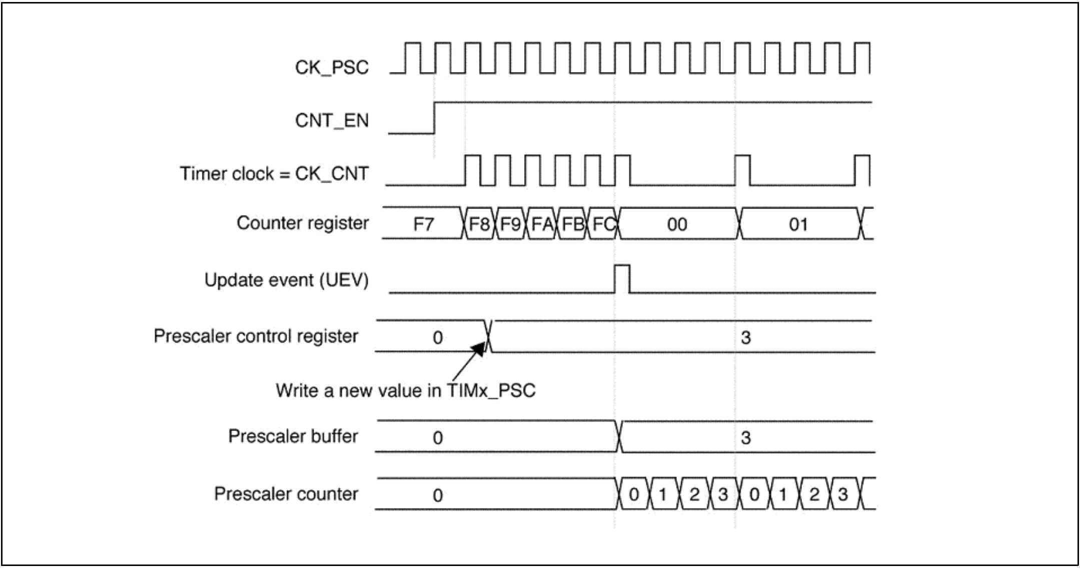
The prescaler can divide the counter clock frequency by power of two from 1 through 32768. It is based on a 16-bit counter controlled through the 4-bit PSC[3:0] in the TIMx_PSC register. The factor by which the internal timer clock frequency (f_{CK_PSC}) is divided is two raised to the power PSC[3:0]:

$$CK\_CNT = f_{CK\_PSC} / (2 \wedge PSC[3:0])$$

It can be changed on the fly as this control register is buffered. The new prescaler ratio is used starting at the next update event.

Figure 14 gives an example of the counter behavior when the prescaler ratio is changed on the fly.

Figure 14. Counter timing diagram with prescaler division change from 1 to 4



10.1.2 Counter modes

Up-counting mode

In up-counting 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.

An update event can be generated at each counter overflow, by setting the UG bit in the TIMx_EGR register, or by using the slave mode controller.

Software can disable the update event by setting the UDIS bit in the TIMx_CR1 register, to avoid updating the shadow registers while writing new values in the buffer registers. No update event will occur until the UDIS bit is written to 0. Both the counter and the prescalar counter restart from 0, but the prescale rate does not change. In addition, if the URS bit in the TIMx_CR1 register is set, setting the UG bit generates an update event but without setting the UIF flag. Thus no interrupt request is sent. This avoids generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, the update flag (the UIF bit in the TIMx_SR register) is set (depending on the URS bit in the TIMx_CR1 register) and the following registers are updated:

- The buffer of the prescaler is reloaded with the buffer value (contents of the TIMx_PSC register).
- The auto-reload shadow register is updated with the buffer value (TIMx_ARR).

Figure 15, *Figure 16*, *Figure 17*, and *Figure 18* show some examples of the counter behavior for different clock frequencies when TIMx_ARR = 0x36.

Figure 15. Counter timing diagram, internal clock divided by 1

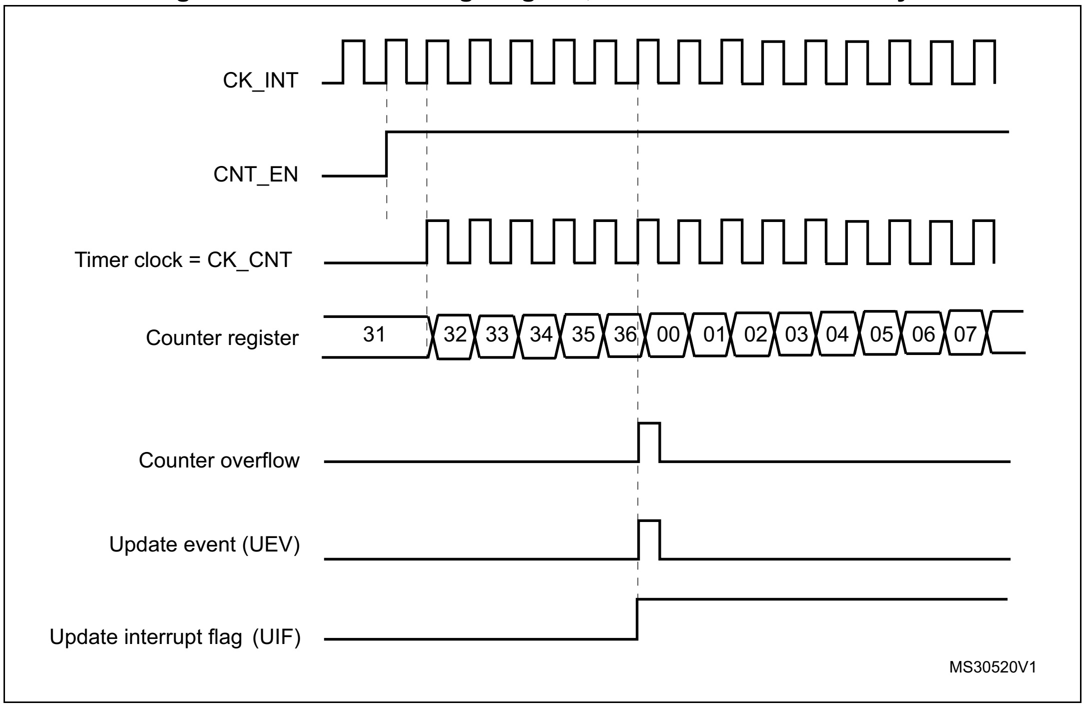


Figure 16. Counter timing diagram, internal clock divided by 4

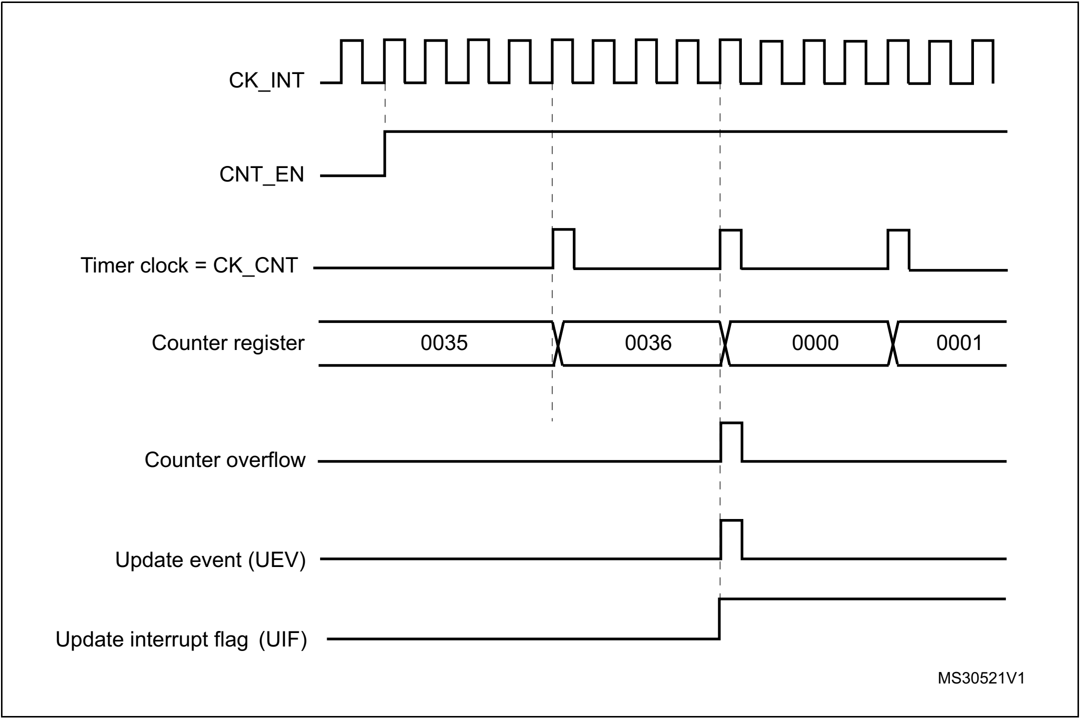


Figure 17. Counter timing diagram, update event when ARPE = 0 (TIMx_ARR not buffered)

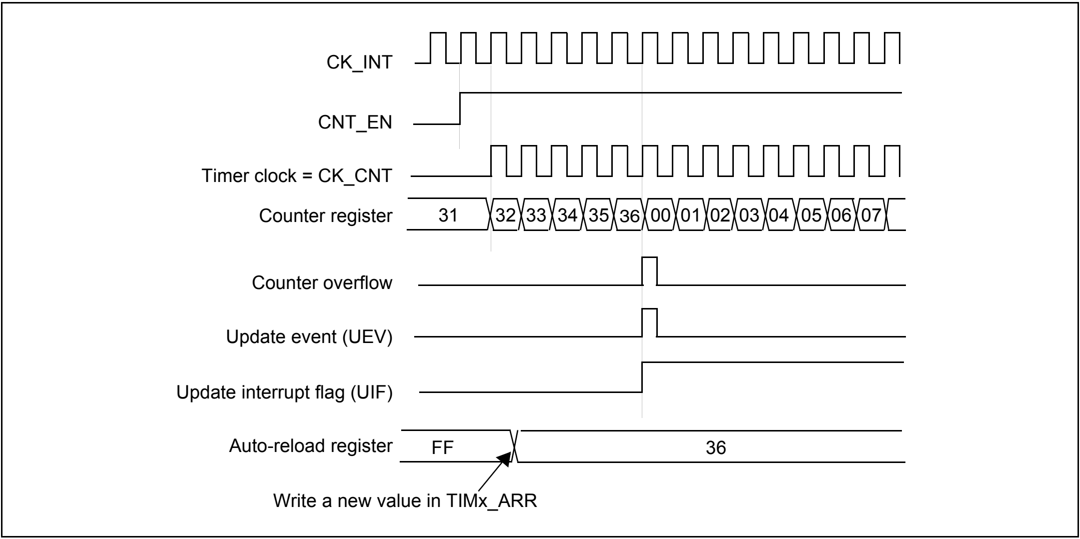
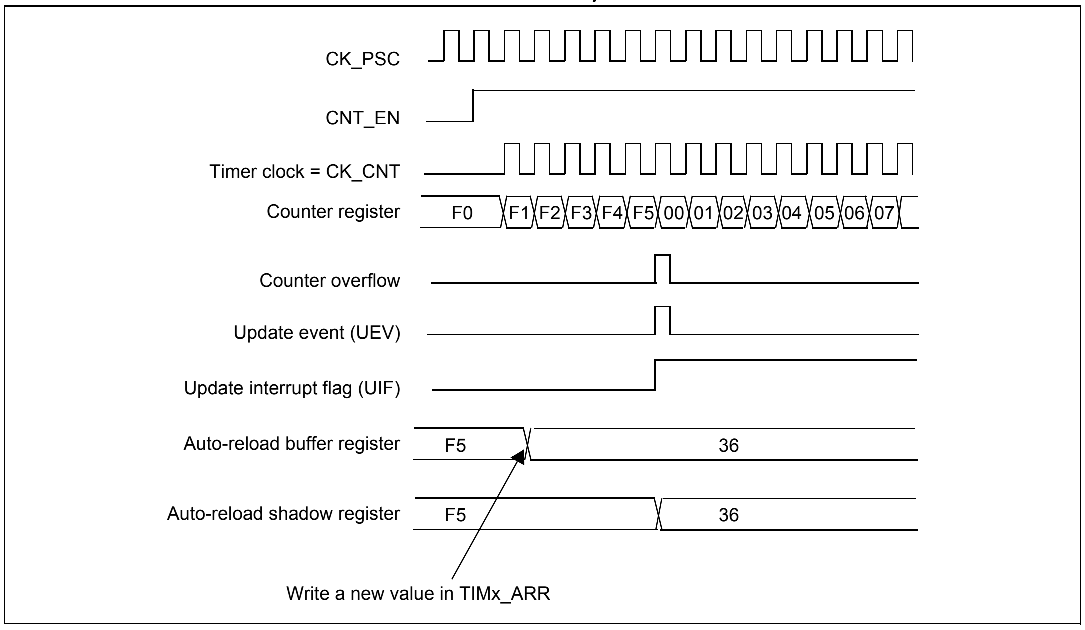


Figure 18. Counter timing diagram, update event when ARPE = 1 (TIMx_ARR buffered)



Down-counting mode

In down-counting mode, the counter counts from the auto-reload value (contents of the TIMx_ARR register) down to 0, then restarts from the auto-reload value and generates a counter underflow event.

An update event can be generated at each counter underflow, by setting the UG bit in the TIMx_EGR register, or by using the slave mode controller). Software can disable the update event by setting the UDIS bit in the TIMx_CR1 register, to avoid updating the shadow registers while writing new values in the buffer registers. No update event occurs until the UDIS bit is written to 0. However, the counter restarts from the current auto-reload value, whereas the prescalar's counter restarts from 0, but the prescale rate doesn't change.

In addition, if the URS bit in the TIMx_CR1 register is set, setting the UG bit generates an update event, but without setting the UIF flag. Thus no interrupt request is sent. This avoids generating both update and capture interrupts when clearing the counter on the capture event.

When an update event occurs, the update flag (the UIF bit in the TIMx_SR register) is set (depending on the URS bit in the TIMx_CR1 register) and the following registers are updated:

- The prescaler shadow register is reloaded with the buffer value (contents of the TIMx_PSC register).
- The auto-reload active register is updated with the buffer value (contents of the TIMx_ARR register). The auto-reload is updated before the counter is reloaded, so that the next period is the expected one.

Figure 19 and *Figure 20* show some examples of the counter behavior for different clock frequencies when TIMx_ARR = 0x36.

Figure 19. Counter timing diagram, internal clock divided by 1

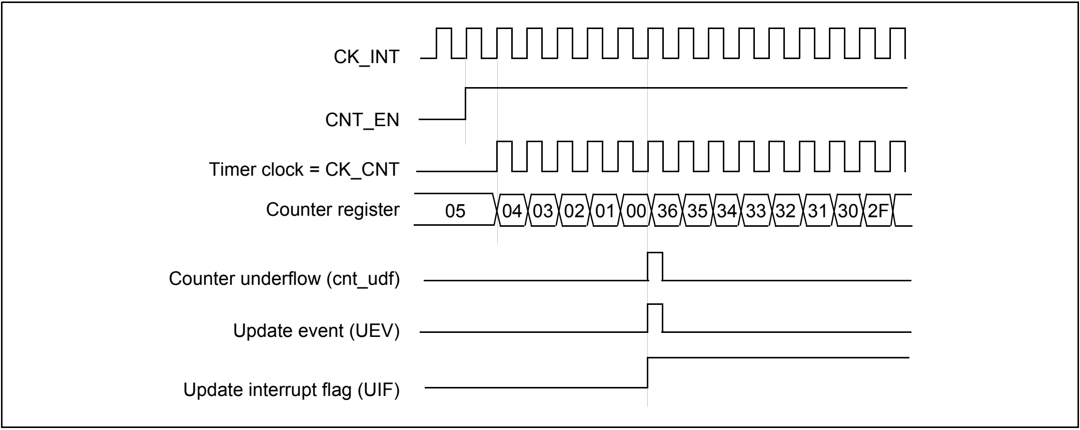
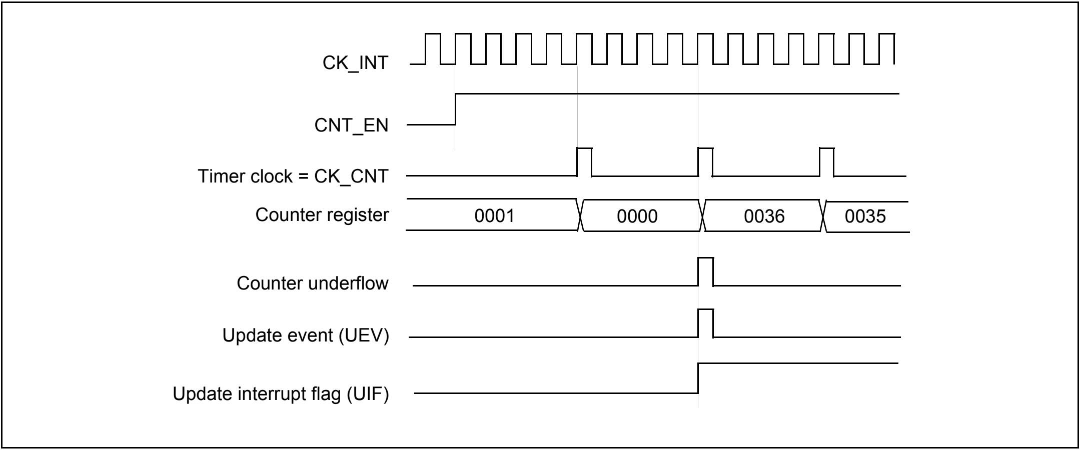


Figure 20. Counter timing diagram, internal clock divided by 4



Center-aligned mode (up/down counting)

In center-aligned mode, the counter counts from 0 to the auto-reload value (contents of the TIMx_ARR register) - 1 and generates a counter overflow event, then counts from the autoreload value down to 1 and generates a counter underflow event. Then it restarts counting from 0.

In this mode, the direction bit (DIR 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. 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 both the counter and the prescalar's counter restart counting from 0.

Software can disable the update event by setting the UDIS bit in the TIMx_CR1 register. This avoids updating the shadow registers while writing new values in the buffer registers. Then no update event occurs until the 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 in the TIMx_CR1 register is set, setting the UG bit generates an update event, but without setting the UIF flag. Thus no interrupt request is sent. This avoids generating both update and capture interrupt when clearing the counter on the capture event.

When an update event occurs, the update flag (the UIF bit in the TIMx_SR register) is set (depending on the URS bit in the TIMx_CR1 register) and the following registers are updated:

- The prescaler shadow register is reloaded with the buffer value (contents of the TIMx_PSC register).
- The auto-reload active register is updated with the buffer value (contents of the TIMx_ARR register). 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.

The following figures show some examples of the counter behavior for different clock frequencies.

Figure 21. Counter timing diagram, internal clock divided by 1, TIMx_ARR = 0x6

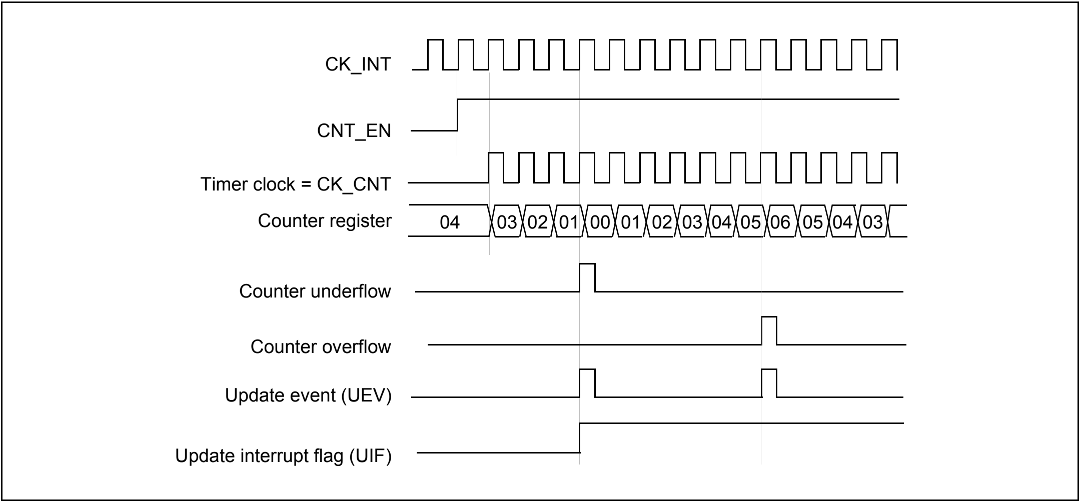


Figure 22. Counter timing diagram, update event with ARPE = 1 (counter underflow)

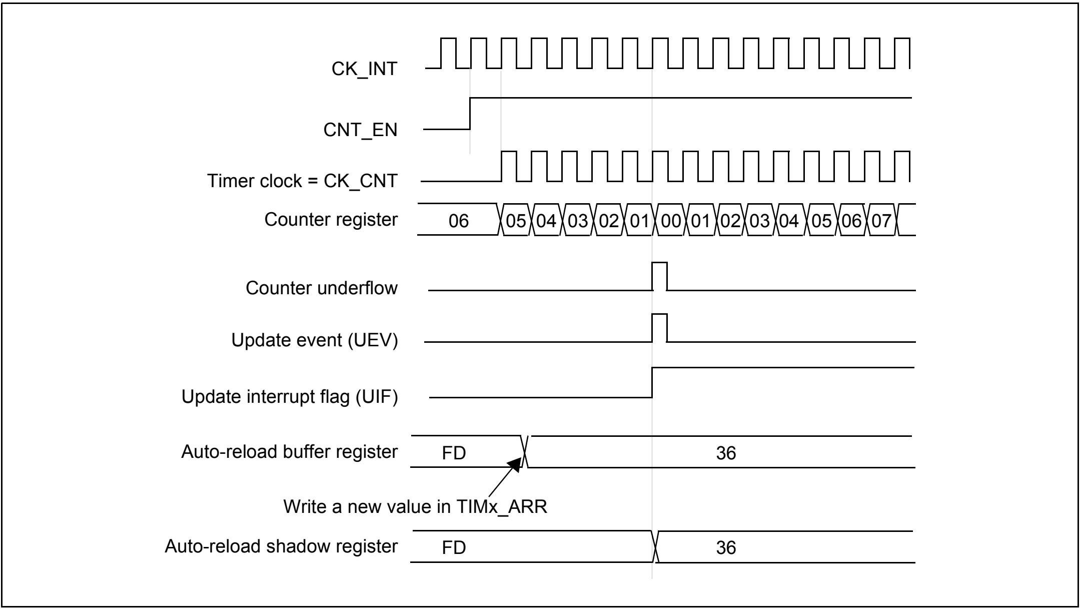
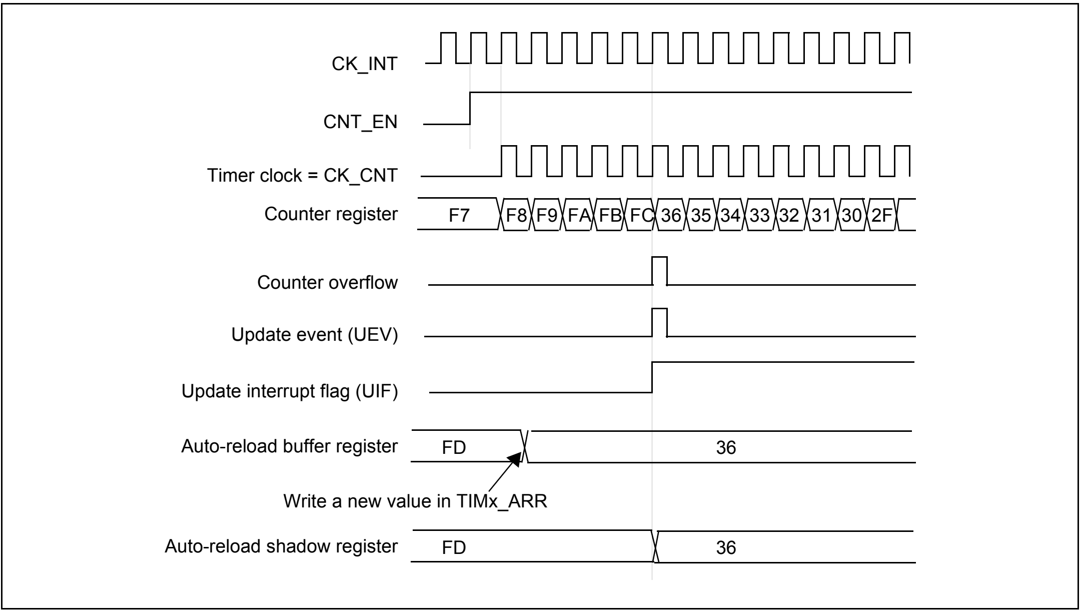


Figure 23. Counter timing diagram, update event with ARPE = 1 (counter overflow)



10.1.3 Clock selection

The counter clock can be provided by the following clock sources:

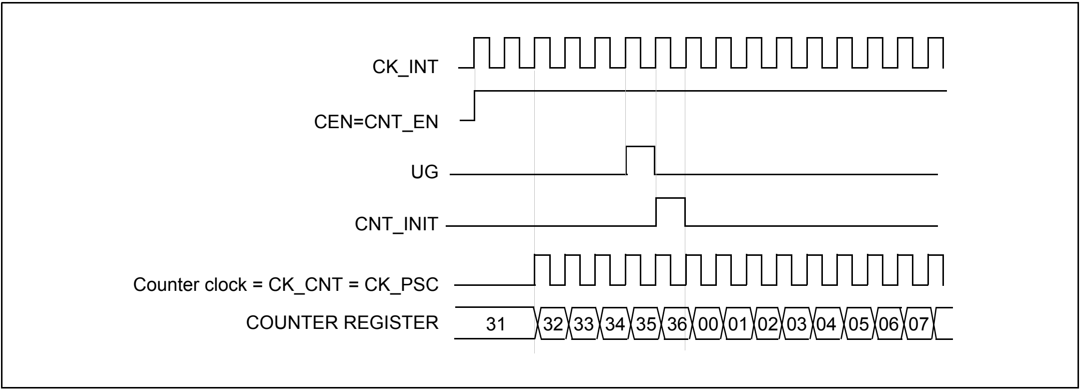
- Internal clock (PCLK)
- External clock mode 1: external input pin (TIy)
- External clock mode 2: external trigger input (ETR)
- Internal trigger input (ITR0): using the other timer as prescaler. Refer to the *Using one timer as prescaler for the other timer* for more details.

Internal clock source (CK_INT)

The internal clock is selected when the slave mode controller is disabled (SMS = 000 in the TIMx_SMCR register). In this mode, 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, except for UG, which remains cleared automatically. As soon as the CEN bit is written to 1, the prescaler is clocked by the internal clock CK_INT.

Figure 24 shows the behavior of the control circuit and the up-counter in normal mode, without prescaling.

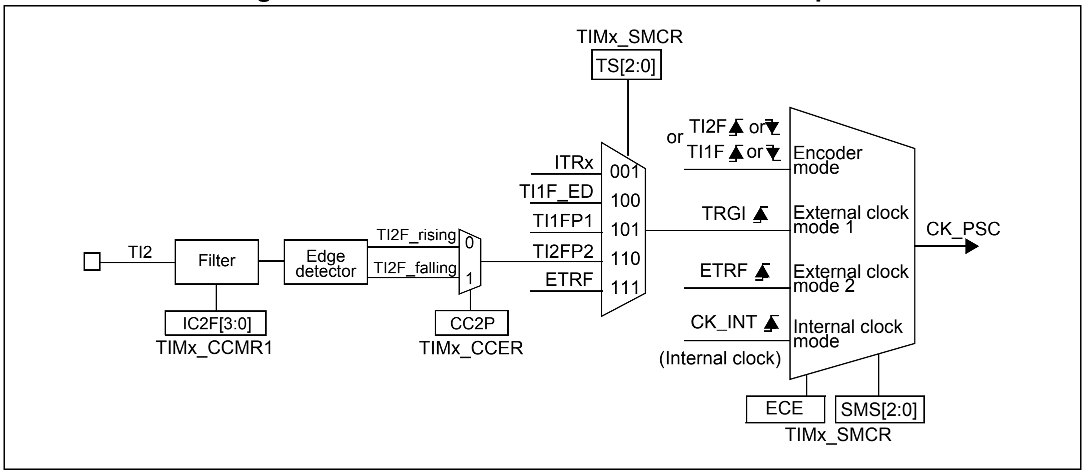
Figure 24. Control circuit in Normal mode, internal clock divided by 1



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 25. TI2 external clock connection example



For example, to configure the up-counter 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 bits in the TIMx_CCMR1 register (if no filter is needed, keep IC2F = 0000).

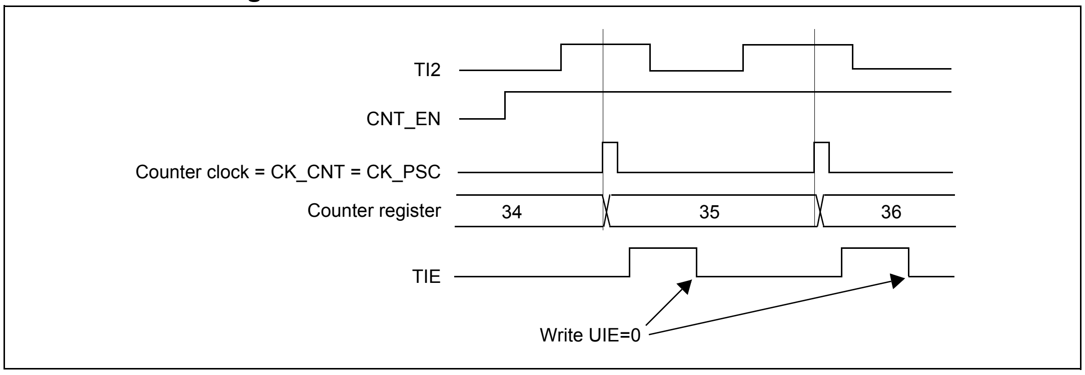
Note: The capture prescaler is not used for triggering, so it does not need to be configured.

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.

When a rising edge occurs on TI2, the counter counts once and the TIE 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 the TI2 input.

Figure 26. 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.

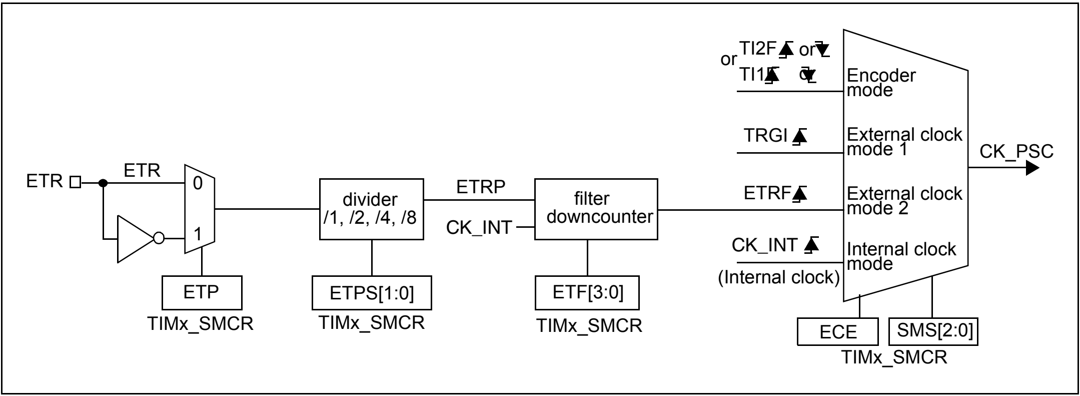
The EXTRIGSEL bits in the TIMx_OR register select a clock signal that drives ETR, as shown in *Table 32*.

Table 32. 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 27 gives an overview of the external trigger input block.

Figure 27. External trigger input block



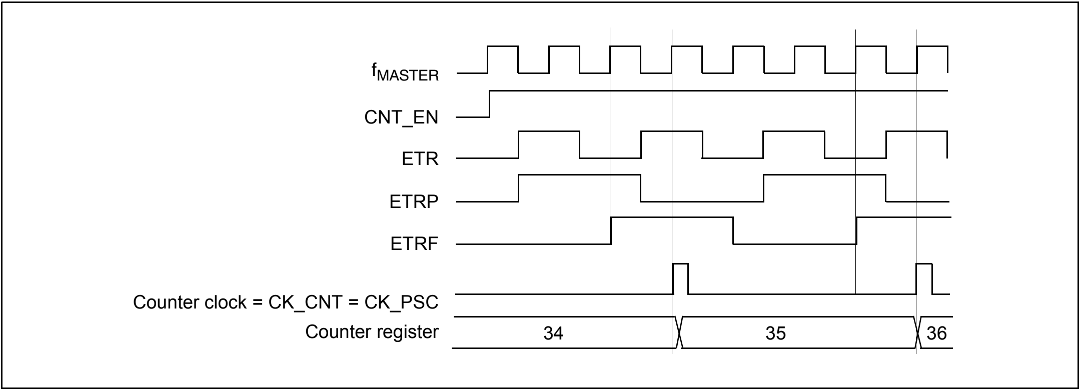
For example, to configure the up-counter to count each 2 rising edges on ETR, use the following procedure:

- 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.

Figure 28. Control circuit in external clock mode 2

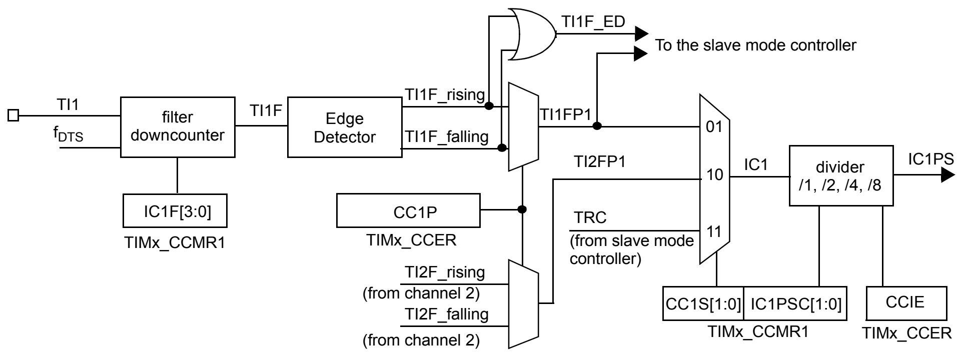


10.1.4 Capture/compare channels

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.

Figure 29 gives an overview of one capture/compare channel. The input stage samples the corresponding TIy input to generate a filtered signal (TIyF). Then an edge detector with polarity selection generates a signal (TIyFPy) which can be used either as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (ICyPS).

Figure 29. Capture/compare channel (example: channel 1 input stage)



The output stage generates an intermediate reference signal, OCyREF, which is only used internally. OCyREF is always active high, but it may be inverted to create the output signal, OCy, that controls a GPIO output.

Figure 30. Capture/compare channel 1 main circuit

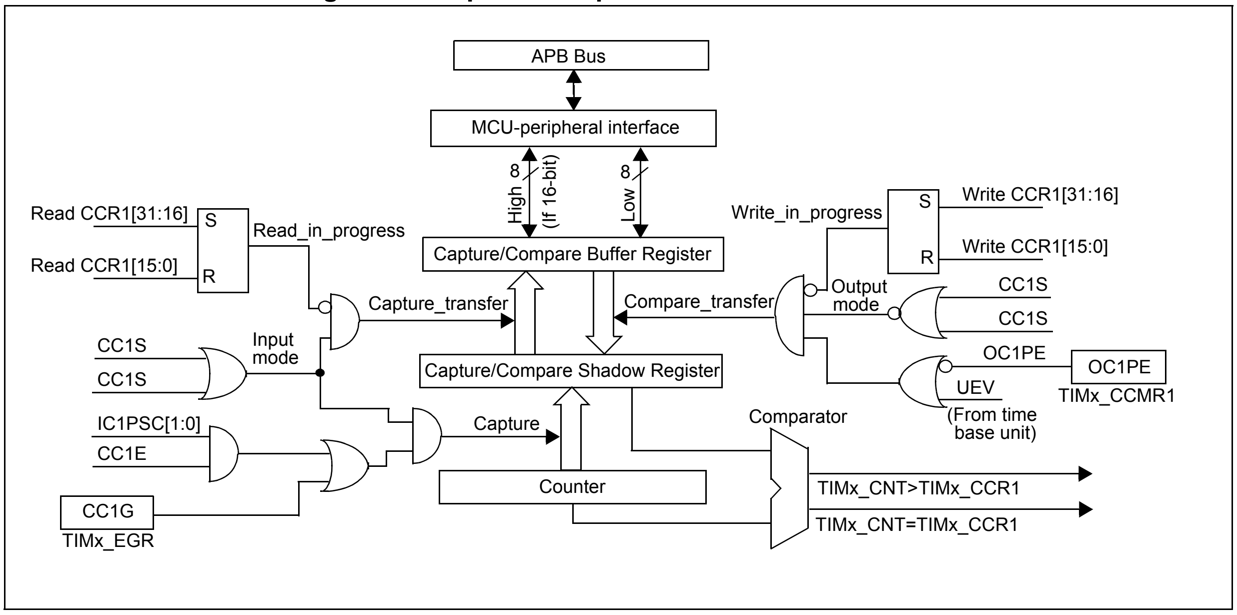
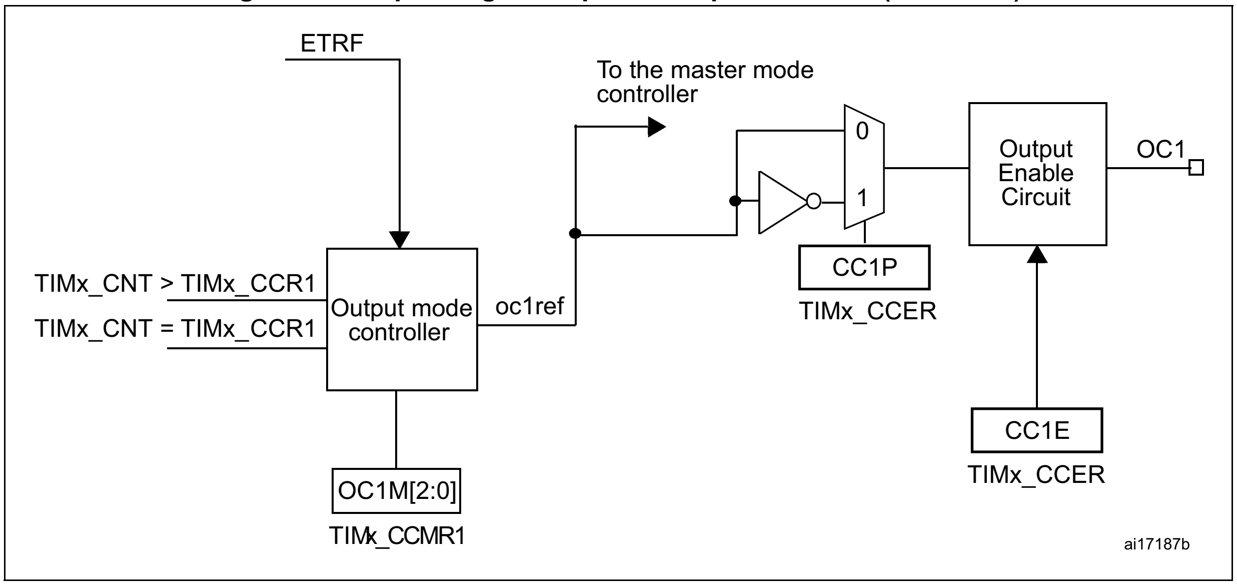


Figure 31. Output stage of capture/compare channel (channel 1)



The capture/compare block is made of a buffer register and a shadow register. Writes and reads always access the buffer register.

In capture mode, captures are first written to the shadow register, then copied into the buffer register.

In compare mode, the content of the buffer register is copied into the shadow register which is compared to the counter.

10.1.5 Input capture mode

In input capture mode, a capture/compare register (TIMx_CCRy) latches the value of the counter after a transition is detected by the corresponding ICy signal. When a capture occurs, the corresponding CCyIF flag in the TIMx_SR register is set, and an interrupt request is sent if enabled.

If a capture occurs when the CCyIF flag is already high, then the missed capture flag CCyIM in the TIMx_MISSR register is set. CCyIF can be cleared by software writing a 1 to its bit or reading the captured data stored in the TIMx_CCRy register. To clear the CCyIF bit, write a 1 to it.

The following example shows how to capture the counter value in the TIMx_CCR1 when the TI1 input rises.

- 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 required input filter duration with respect to the signal connected to the timer, when the input is one of the TIy (ICyF bits in the TIMx_CCMR1 register). Consider a situation in which, when toggling, the input signal is unstable during at most 5 internal clock cycles. The filter duration must be longer than these 5 clock cycles. The transition on TI1 can be validated when 8 consecutive samples with the new level have been detected (sampled at PCLK frequency). To do this, write the IC1F bits to 0011 in the TIMx_CCMR1 register.
- Select the edge of the active transition on the TI1 channel by writing the CC1P bit to 0 in the TIMx_CCER register (rising edge in this case).
- Program the input prescaler: In this example, the capture is to be performed at each valid transition, so the prescaler is disabled (write the IC1PSC 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_IER register.
- When an input capture occurs:
  - The TIMx_CCR1 register gets the value of the counter on the active transition.
  - CC1IF flag is set (capture/compare interrupt flag). The missed capture/compare flag CC1IM in TIMx_MISSR is also set if another capture occurs before the CC1IF flag is cleared.
  - An interrupt may be generated if enabled by the CC1IF bit.

To detect missed captures reliably, read captured data in TIMx_CCRy before checking the missed capture/compare flag. This sequence avoids missing a capture that could happen after reading the flag and before reading the data.

Note: Software can generate IC interrupt requests by setting the corresponding CCyG bit in the TIMx_EGR register.

10.1.6 PWM input mode

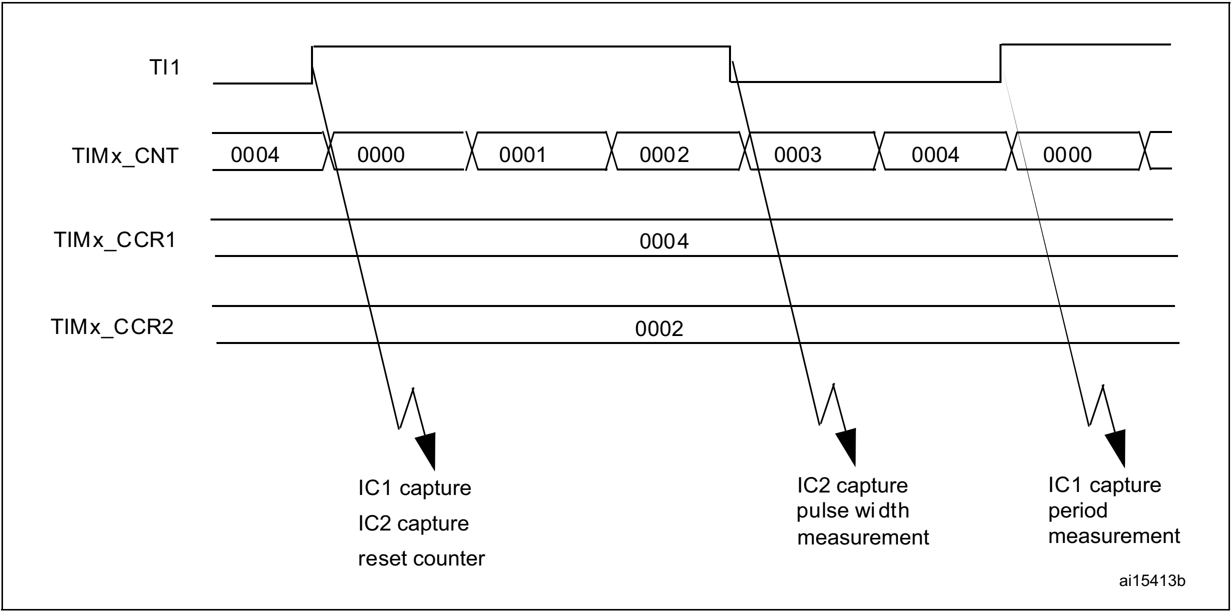
This mode is a particular case of input capture mode. The procedure is the same except:

- Two ICy signals are mapped on the same TIy input.
- These two ICy signals are active on edges with opposite polarity.
- One of the two TIyFP signals is selected as trigger input and the slave mode controller is configured in reset mode.

For example, to measure the period in the TIMx_CCR1 register and the duty cycle in the TIMx_CCR2 register of the PWM applied on TI1, use 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 the TIMx_CCR1 and counter clear, by writing the CC1P bit to 0 (active on rising edge).
- Select the active input for TIMx_CCR2 by writing the CC2S bits to 10 in the TIMx_CCMR1 register (TI1 selected).
- Select the active polarity for TI1FP2 (used for capture in the TIMx_CCR2) by writing the CC2P bit to 1 (active on falling edge).
- Select the valid trigger input by writing the TS bits to 101 in the TIMx_SMCR register (TI1FP1 selected).
- Configure the slave mode controller in reset mode by writing the SMS bits to 100 in the TIMx_SMCR register.
- Enable the captures by writing the CC1E and CC2E bits to 1 in the TIMx_CCER register.

Figure 32. PWM input mode timing



10.1.7 Forced output mode

In output mode (CCyS bits = 00 in the TIMx_CCMR1 register), software can force each output compare signal (OCyREF and then OCy) to an active or inactive level independently of any comparison between the output compare register and the counter.

To force an output compare signal (OCyREF/OCy) to its active level, write 101 in the OCyM bits in the corresponding TIMx_CCMR1 register. OCyREF is forced high (OCyREF is always active high) and OCy gets the opposite value to the CCyP polarity bit. For example, CCyP = 0 defines OCy as active high, so when OCyREF is active, OCy is also set to a high level.

The OCyREF signal can be forced low by writing the OCyM bits to 100 in the TIMx_CCMR1 register.

The comparison between the TIMx_CCRy shadow register and the counter is still performed and allows the CCyIF flag to be set. Interrupt requests can be sent accordingly. This is described in *Section 10.1.8: Output compare mode on page 174*.

10.1.8 Output compare mode

This mode is used to control an output waveform or to indicate 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 (the OCyM bits in the TIMx_CCMR1 register) and the output polarity (the CCyP bit in the TIMx_CCER register). The output can remain unchanged (OCyM = 000), be set active (OCyM = 001), be set inactive (OCyM = 010), or can toggle (OCyM = 011) on the match.
- Sets a flag in the interrupt flag register (the CCyIF bit in the TIMx_SR register).
- Generates an interrupt if the corresponding interrupt mask is set (the CCyIE bit in the TIMx_IER register).

The TIMx_CCRy registers can be programmed with or without buffer registers using the OCyPE bit in the TIMx_CCMR1 register.

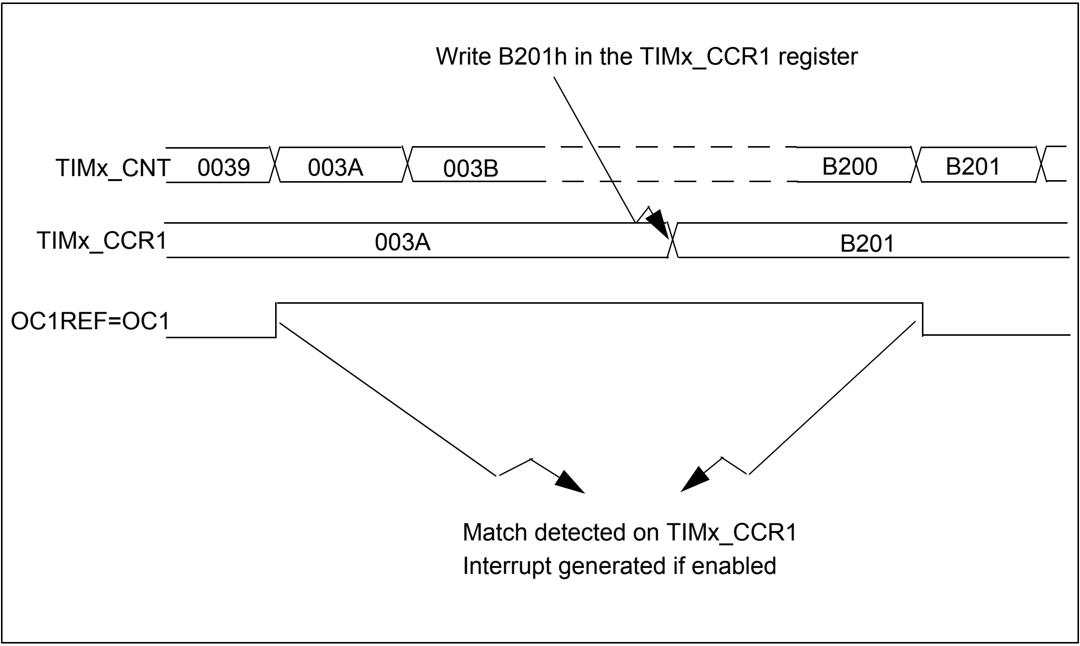
In output compare mode, the update event has no effect on OCyREF or the OCy output. The timing resolution is one count of the counter. Output compare mode can also be used to output a single pulse (in one pulse mode).

Procedure:

1. Select the counter clock (internal, external, and prescaler).
2. Write the desired data in the TIMx_ARR and TIMx_CCRy registers.
3. Set the CCyIE bit in TIMx_IER if an interrupt request is to be generated.
4. Select the output mode. For example, you must write OCyM = 011, OCyPE = 0, CCyP = 0 and CCyE = 1 to toggle the OCy output pin when TIMx_CNT matches TIMx_CCRy, TIMx_CCRy buffer is not used, OCy is enabled and active high.
5. Enable the counter by setting the CEN bit in the TIMx_CR1 register.

To control the output waveform, software can update the TIMx_CCRy register at any time, provided that the buffer register is not enabled (OCyPE = 0). Otherwise TIMx_CCRy shadow register is updated only at the next update event. An example is given in *Figure 33*.

Figure 33. Output compare mode, toggle on OC1


Write B201h in the TIMx_CCR1 register
TIMx_CNT
0039
003A
003B
B200
B201
TIMx_CCR1
003A
B201
OC1REF=OC1
Match detected on TIMx_CCR1
Interrupt generated if enabled

10.1.9 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_CCRy register.

PWM mode can be selected independently on each channel (one PWM per OCy output) by writing 110 (PWM mode 1) or 111 (PWM mode 2) in the OCyM bits in the TIMx_CCMR1 register. The corresponding buffer register must be enabled by setting the OCyPE bit in the TIMx_CCMR1 register. Finally, in up-counting or center-aligned mode the auto-reload buffer register must be enabled by setting the ARPE bit in the TIMx_CR1 register.

Because the buffer registers are only transferred to the shadow registers when an update event occurs, before starting the counter initialize all the registers by setting the UG bit in the TIMx_EGR register.

OCy polarity is software programmable using the CCyP bit in the TIMx_CCER register. It can be programmed as active high or active low. OCy output is enabled by the CCyE bit in the TIMx_CCER register. Refer to the TIMx_CCER register description in the Registers section for more details.

In PWM mode (1 or 2), TIMx_CNT and TIMx_CCRy are always compared to determine whether TIMx_CCRy ≤ TIMx_CNT or TIMx_CNT ≤ TIMx_CCRy,depending on the direction of the counter. The OCyREF signal is asserted only:

- When the result of the comparison changes, or
- When the output compare mode (OCyM bits in the TIMx_CCMR1 register) switches from the "frozen" configuration (no comparison, OCyM = 000) to one of the PWM modes (OCyM = 110 or 111).

This allows software to force a PWM output to a particular state while the timer is running.

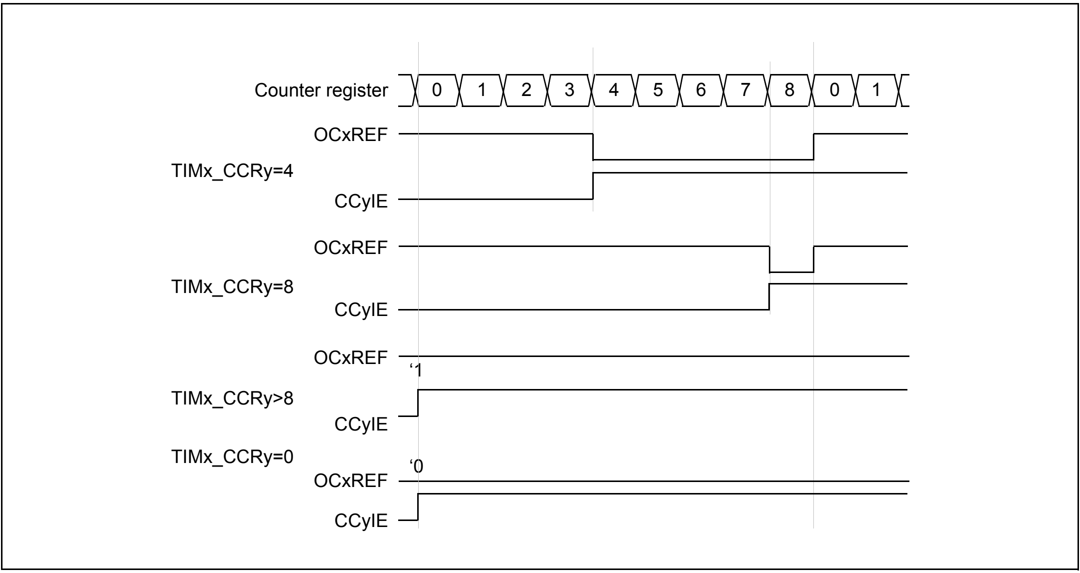
The timer is able to generate PWM in edge-aligned mode or center-aligned mode depending on the CMS bits in the TIMx_CR1 register.

PWM edge-aligned mode: up-counting configuration

Up-counting is active when the DIR bit in the TIMx_CR1 register is low. Refer to *Up-counting mode on page 161*.

The following example uses PWM mode 1. The reference PWM signal OCyREF is high as long as TIMx_CNT < TIMx_CCRy, otherwise it becomes low. If the compare value in TIMx_CCRy is greater than the auto-reload value in TIMx_ARR, then OCyREF is held at 1. If the compare value is 0, then OCyREF is held at 0. *Figure 34* shows some edge-aligned PWM waveforms in an example, where TIMx_ARR = 8.

Figure 34. Edge-aligned PWM waveforms (ARR = 8)



PWM edge-aligned mode: down-counting configuration

Down-counting is active when the DIR bit in the TIMx_CR1 register is high. Refer to *Down-counting mode on page 163* for more information.

In PWM mode 1, the reference signal OCyREF is low as long as TIMx_CNT > TIMx_CCRy, otherwise it becomes high. If the compare value in TIMx_CCRy is greater than the auto-reload value in TIMx_ARR, then OCyREF is held at 1. Zero-percent PWM is not possible in this mode.

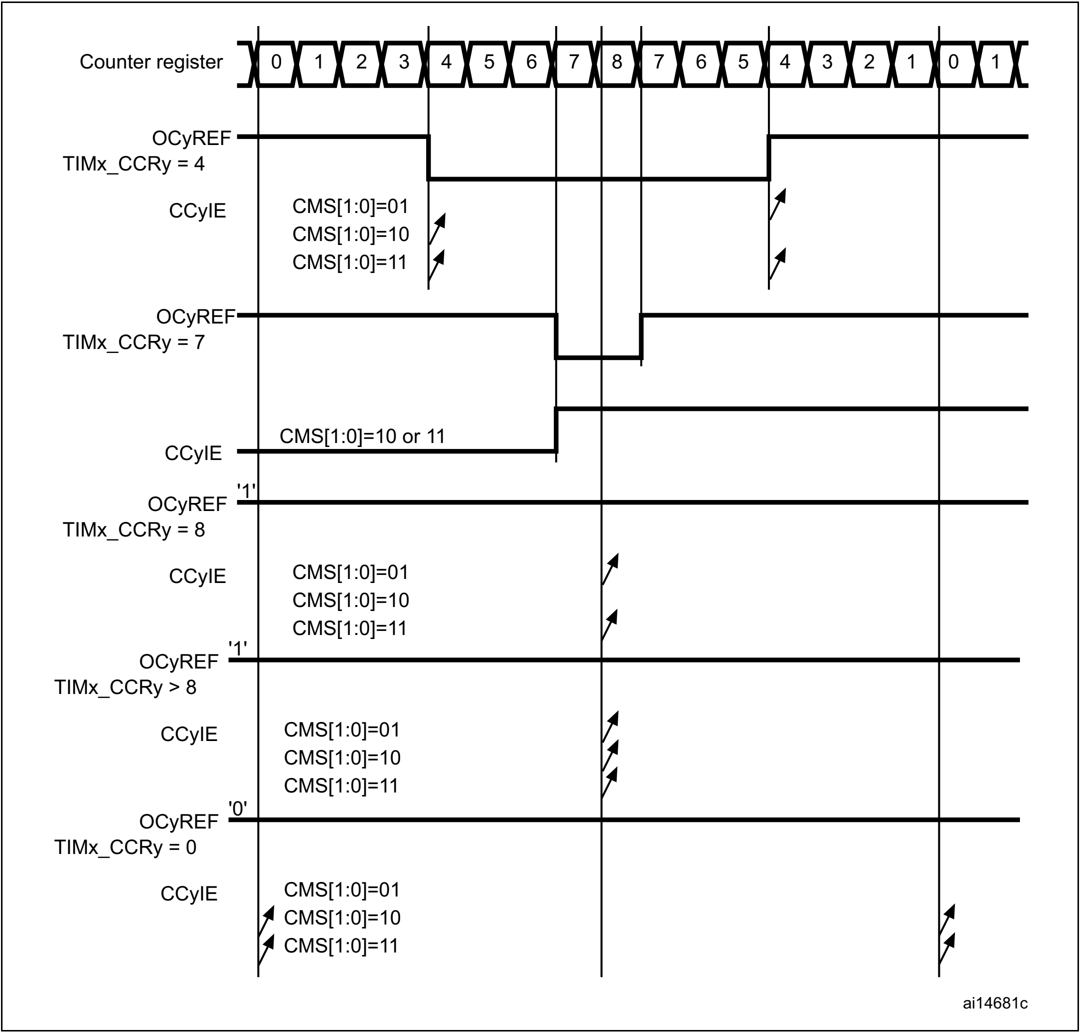
PWM center-aligned mode

Center-aligned mode is active except when the CMS bits in the TIMx_CR1 register are 00 (all configurations where CMS is non-zero have the same effect on the OCyREF/OCy signals). The compare flag is set when the counter counts up, when it counts down, or 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. Refer to *Center-aligned mode (up/down counting) on page 164* for more information.

Figure 35 shows some center-aligned PWM waveforms in an example where:

- TIMx_ARR = 8,
- PWM mode is the PWM mode 1,
- The output compare flag is set when the counter counts down corresponding to the center-aligned mode 1 selected for CMS = 01 in the TIMx_CR1 register.

Figure 35. Center-aligned PWM waveforms (ARR = 8)



<u>Hints on using center-aligned mode:</u>

- When starting in center-aligned mode, the current up-down configuration is used. This means that the counter counts up or down depending on the value written in the DIR bit in the TIMx_CR1 register. 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 the value written to 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 when 0 or the TIMx_ARR value is written to the counter, but no update event 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.

10.1.10 One-pulse mode

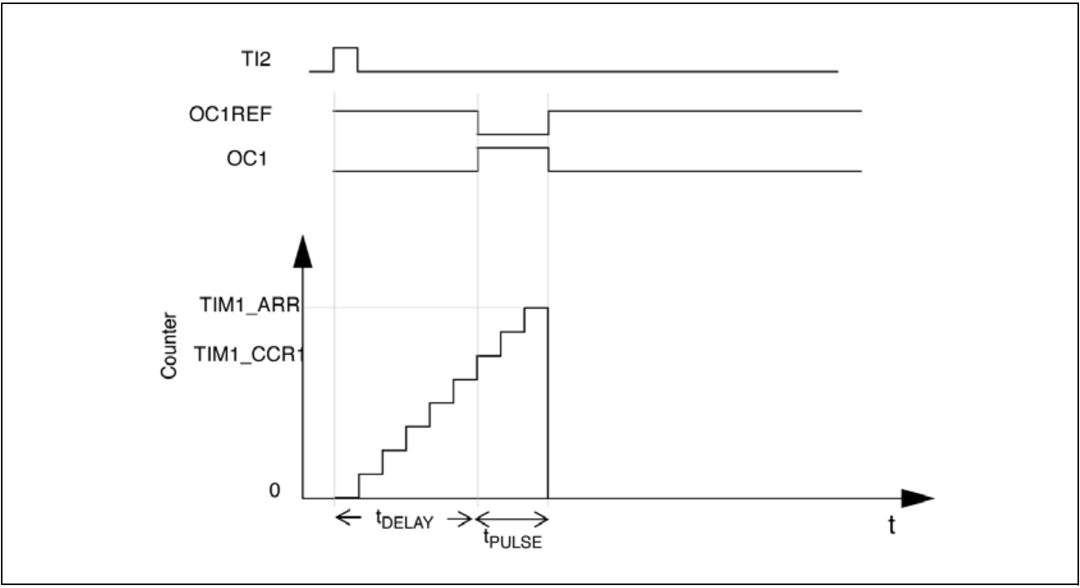
One-pulse mode (OPM) is a special case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. Select OPM by setting the OPM bit in the TIMx_CR1 register. This makes the counter stop automatically at the next update event.

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 up-counting: TIMx_CNT < TIMx_CCRy ≤ TIMx_ARR (in particular, 0 < TIMx_CCRy),
- In down-counting: TIMx_CNT > TIMx_CCRy.

Figure 36. Example of one pulse mode



For example, to generate a positive pulse on OC1 with a length of tPULSE and after a delay of tDELAY as soon as a rising edge is detected on the TI2 input pin, using TI2FP2 as trigger 1:

- Map TI2FP2 on TI2 by writing IC2S = 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: Write 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 tPULSE is defined by the difference between the auto-reload value and the compare value (TIMx_ARR - TIMx_CCR1).

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, enable PWM mode 2 by writing OC1M = 111 in the TIMx_CCMR1 register. Optionally, enable the buffer registers by writing OC1PE = 1 in the TIMx_CCMR1 register and ARPE in the TIMx_CR1 register. In this case, also 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 the example, the DIR and CMS bits in the TIMx_CR1 register should be low.

Since only one pulse is desired, software should set the OPM bit in the TIMx_CR1 register to stop the counter at the next update event (when the counter rolls over from the auto-reload value back to 0).

A special case: OCy fast enable

In one-pulse mode, the edge detection on the TIy input sets the CEN bit, which enables the counter. Then the comparison between the counter and the compare value toggles the output. However, several clock cycles are needed for this operation, and it limits the minimum delay (tDELAY min) achievable.

To output a waveform with the minimum delay, set the OCyFE bit in the TIMx_CCMR1 register. Then OCyREF (and OCy) is forced in response to the stimulus, without taking the comparison into account. Its new level is the same as if a compare match had occurred. OCyFE acts only if the channel is configured in PWM mode 1 or 2.

10.1.11 Encoder interface mode

To select encoder interface mode, write SMS = 001 in the TIMx_SMCR register to count only TI2 edges, SMS = 010 to count only TI1 edges, and SMS = 011 to count both TI1 and TI2 edges.

Select the TI1 and TI2 polarity by programming the CC1P and CC2P bits in the TIMx_CCER register. If needed, program the input filter as well.

The two inputs TI1 and TI2 are used to interface to an incremental encoder (see *Table 33*). Assuming that it is enabled, (the CEN bit in the TIMx_CR1 register written to 1) 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.) 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, and hardware modifies the DIR bit in the TIMx_CR1 register accordingly. The DIR bit is calculated at each transition on any input (TI1 or TI2), whether 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 TIMx_ARR or TIMx_ARR down to 0 depending on the direction), so TIMx_ARR must be configured before starting. In the same way, the capture, compare, prescaler, and trigger output features continue to work as normal.

In this mode the counter is modified automatically following the speed and the direction of the incremental encoder, and therefore its contents always represent the encoder's position. The count direction corresponds to the rotation direction of the connected sensor. *Table 33* summarizes the possible combinations, assuming TI1 and TI2 do not switch at the same time.

Table 33. 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 37 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 37. Example of counter operation in encoder interface mode

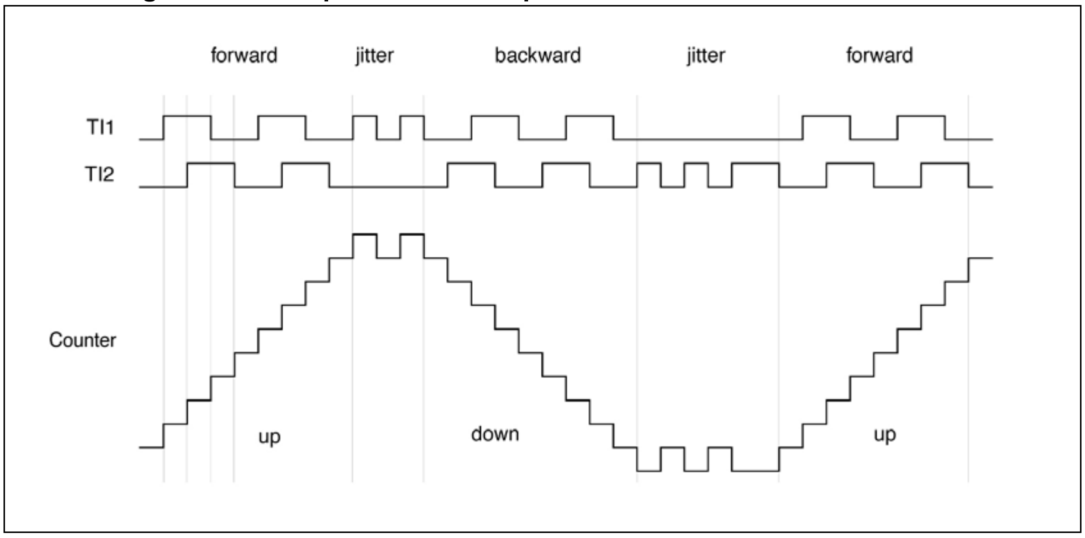
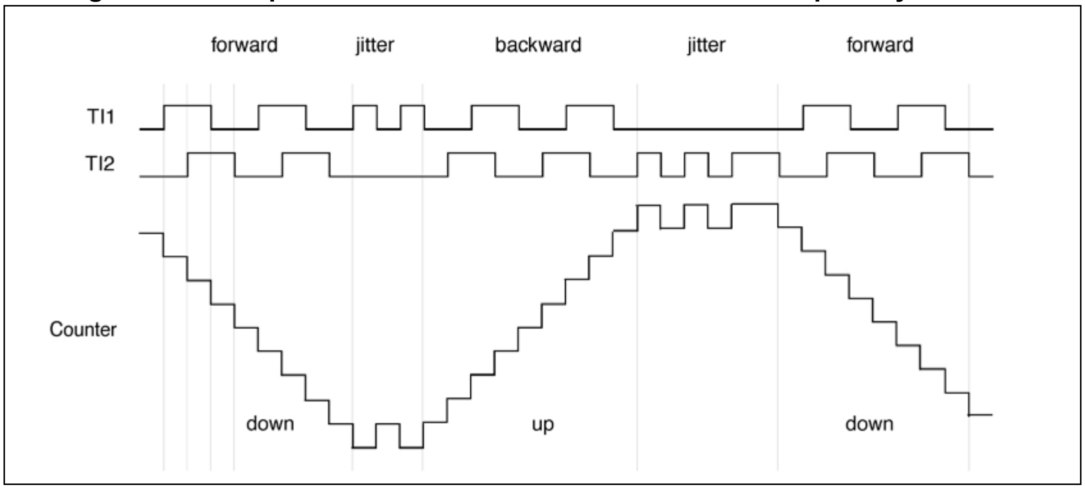


Figure 38 gives an example of counter behavior when IC1FP1 polarity is inverted (same configuration as above except CC1P = 1).

Figure 38. Example of encoder interface mode with IC1FP1 polarity inverted



The timer configured in encoder interface mode provides information on a sensor's current position. To obtain dynamic information (speed, acceleration/deceleration), measure the period between two encoder events using a second timer configured in capture mode. The output of the encoder that 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. Do this by latching the counter value into a third input capture register. (In this case the capture signal must be periodic and can be generated by another timer).

10.1.12 Timer input XOR function

The TI1S bit in the TIM1_CR2 register allows the input filter of channel 1 to be connected to the output of a XOR gate that combines the three input pins TIMxC2 to TIMxC4.

The XOR output can be used with all the timer input functions such as trigger or input capture. It is especially useful to interface to Hall effect sensors.

10.1.13 Timers and external trigger synchronization

The timers can be synchronized with an external trigger in several modes: Reset mode, Gated mode, and Trigger mode.

Slave mode: Reset mode

Reset mode reinitializes the counter and its prescaler in response to an event on a trigger input. Moreover, if the URS bit in the TIMx_CR1 register is low, an update event is generated. Then all the buffered registers (TIMx_ARR, TIMx_CCRy) are updated.

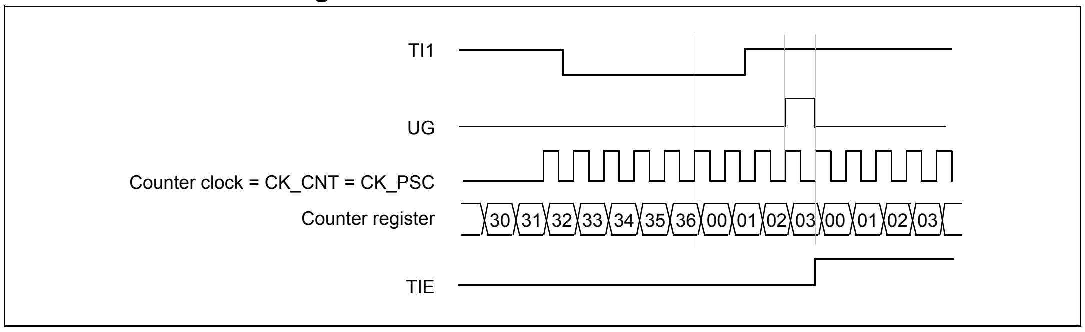
In the following example, the up-counter is cleared in response to a rising edge on the TI1 input:

- Configure the channel 1 to detect rising edges on TI1: Configure the input filter duration. In this example, no filter is required so IC1F = 0000. The capture prescaler is not used for triggering, so it is not configured. The CC1S bits select the input capture source only, CC1S = 01 in the TIMx_CCMR1 register. Write CC1P = 0 in the TIMx_CCER register to validate the polarity, and detect rising edges only.
- Configure the timer in Reset mode by writing SMS = 100 in the TIMx_SMCR register. Select TI1 as the input source by writing TS = 101 in the TIMx_SMCR register.
- Start the counter by writing CEN = 1 in the TIMx_CR1 register.

The counter starts counting on the internal clock, then behaves normally until the TI1 rising edge. When TI1 rises, the counter is cleared and restarts from 0. In the meantime, the trigger flag is set (the TIE bit in the TIMx_SR register) and an interrupt request can be sent if enabled (depending on the TIE bit in the TIMx_IER register).

Figure 39 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 the TI1 input.

Figure 39. Control circuit in Reset mode



Slave mode: Gated mode

In Gated mode the counter is enabled depending on the level of a selected input.

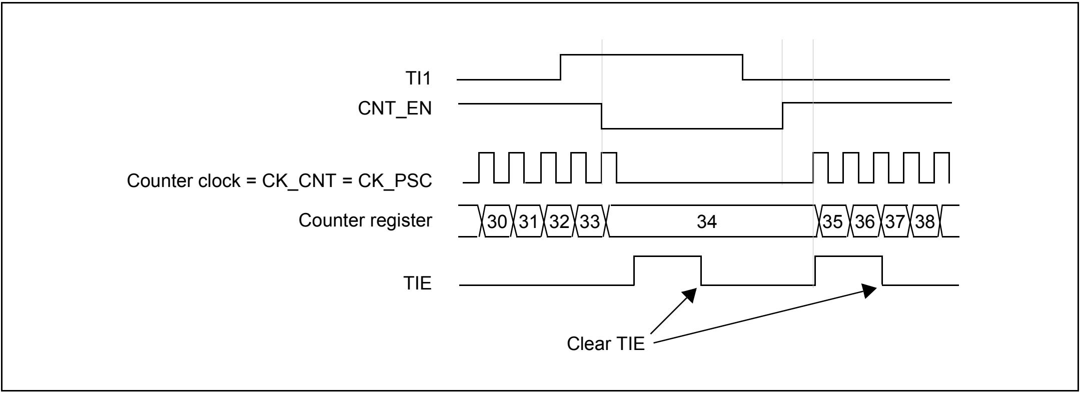
In the following example, the up-counter counts only when the TI1 input is low:

- Configure channel 1 to detect low levels on TI1 Configure the input filter duration. In this example, no filter is required, so IC1F = 0000. The capture prescaler is not used for triggering, so it is not configured. The CC1S bits select the input capture source only, CC1S = 01 in the TIMx_CCMR1 register. Write CC1P = 1 in the TIMx_CCER register to validate the polarity (and detect low level only).
- Configure the timer in Gated mode by writing SMS = 101 in the TIMx_SMCR register. Select TI1 as the input source by writing TS = 101 in the TIMx_SMCR register.
- Enable the counter by writing CEN = 1 in the TIMx_CR1 register. In Gated mode, the counter does not start if CEN = 0, regardless of 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 TIE flag in the TIMx_SR register is set when the counter starts and when it stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on the TI1 input.

Figure 40. Control circuit in Gated mode



Slave mode: Trigger mode

In Trigger mode the counter starts in response to an event on a selected input.

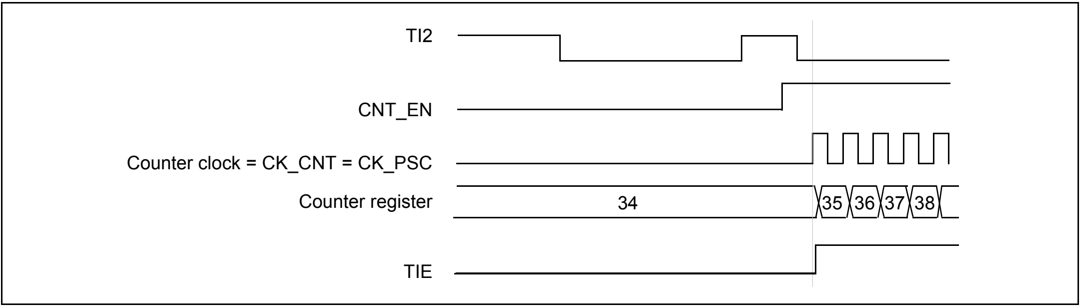
In the following example, the up-counter starts in response to a rising edge on the TI2 input:

- Configure channel 2 to detect rising edges on TI2 Configure the input filter duration. In this example, no filter is required so IC2F = 0000. The capture prescaler is not used for triggering, so it is not configured. The CC2S bits select the input capture source only, CC2S = 01 in the TIMx_CCMR1 register. Write CC2P = 0 in the TIMx_CCER register to validate the polarity and detect high level only.
- Configure the timer in Trigger mode by writing SMS = 110 in the TIMx_SMCR register. Select TI2 as the input source by writing TS = 110 in the TIMx_SMCR register.

When a rising edge occurs on TI2, the counter starts counting on the internal clock and the TIE 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 the TI2 input.

Figure 41. Control circuit in Trigger mode



Slave mode: External clock mode 2 + Trigger mode

External clock mode 2 can be used in combination with 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 not recommended to select ETR as TRGI through the TS bits of TIMx_SMCR register.

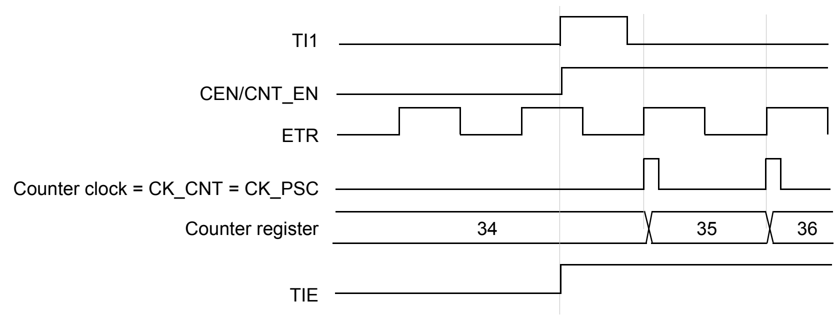
In the following example, the up-counter is incremented at each rising edge of the ETR signal as soon as a rising edge of TI1 occurs:

- 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 and does not need to be configured.
  - CC1S = 01in the TIMx_CCMR1 register to select only the input capture source.
  - CC1P = 0 in the TIMx_CCER register to validate the polarity (and detect rising edge only).
- Configure the timer in Trigger mode by writing SMS = 110 in the TIMx_SMCR register. Select TI1 as the input source by writing TS = 101 in the TIMx_SMCR register.

A rising edge on TI1 enables the counter and sets the TIE 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 42. Control circuit in External clock mode 2 + Trigger mode



10.1.14 Timer synchronization

The two timers can be linked together internally for timer synchronization or chaining. A timer configured in Master mode can reset, start, stop or clock the counter of the other timer configured in Slave mode.

Figure 43 presents an overview of the trigger selection and the master mode selection blocks.

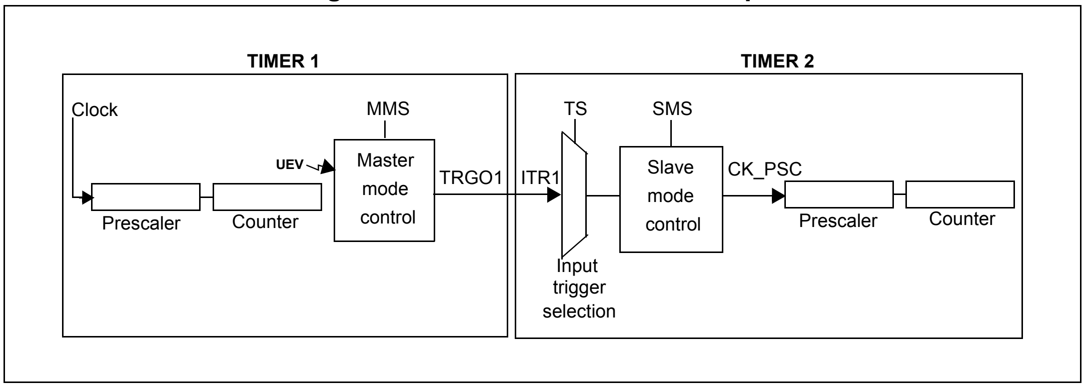
Using one timer as prescaler for the other timer

For example, to configure Timer 1 to act as a prescaler for Timer 2 (see *Figure 43*):

- Configure Timer 1 in Master mode so that it outputs a periodic trigger signal on each update event. Writing MMS = 010 in the TIM1_CR2 register causes a rising edge to be output on TRGO each time an update event is generated.
- To connect the TRGO output of Timer 1 to Timer 2, configure Timer 2 in slave mode using ITR0 as an internal trigger. Select this through the TS bits in the TIM2_SMCR register (writing TS = 000).
- 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 corresponds to the Timer 1 counter overflow).
- Finally both timers must be enabled by setting their respective CEN bits (TIMx_CR1 register).

Note: If OCy is selected on Timer 1 as trigger output (MMS = 1xx), its rising edge is used to clock the counter of Timer 2.

Figure 43. Master/slave timer example



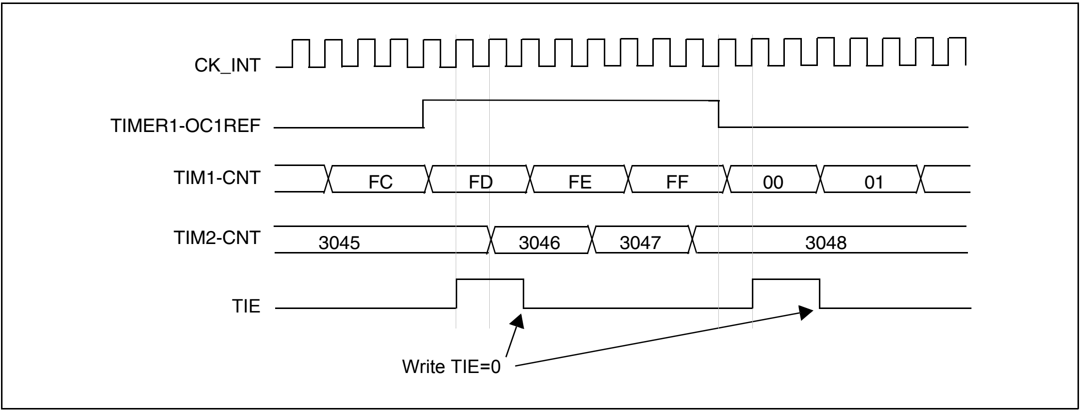
Using one timer to enable the other timer

In this example, the enable of Timer 2 is controlled with the output compare 1 of Timer 1. Refer to *Figure 43* for connections. 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 (fCK_CNT = fCK_INT /3).

- Configure Timer 1 in master mode to send its Output Compare Reference (OC1REF) 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 the 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 (TIM1_CR1 register).

Note: The counter 2 clock is not synchronized with counter 1, this mode only affects the Timer 2 counter enable signal.

Figure 44. Gating Timer 2 with OC1REF of Timer 1

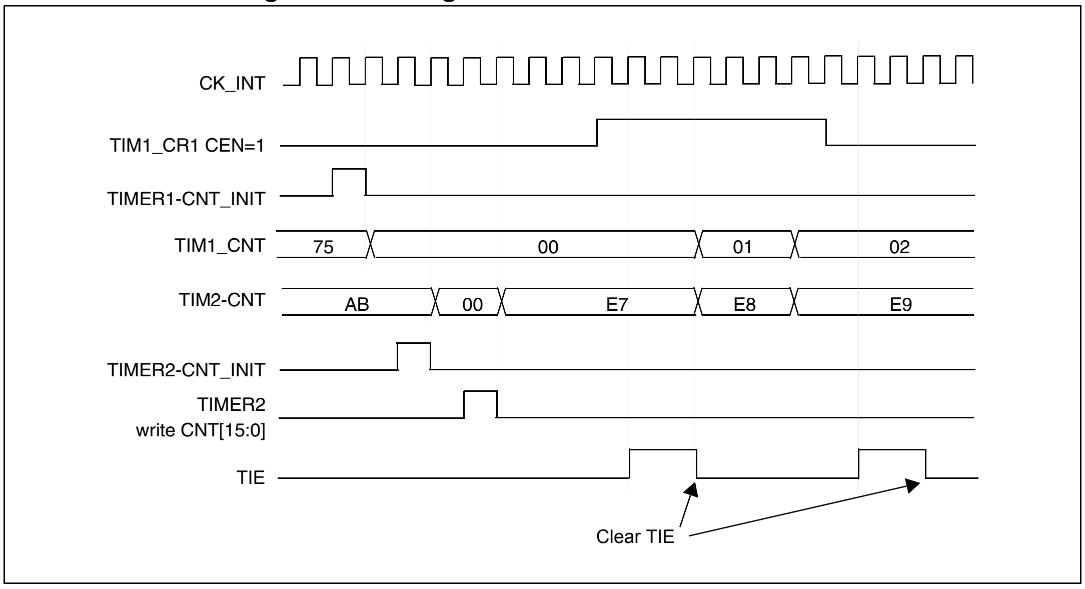


In the example in *Figure 44*, the Timer 2 counter and prescaler are not initialized before being started. So they start counting from their current value. It is possible to start from a given value by resetting both timers before starting Timer 1, then writing the desired value in the timer counters. The timers can easily be reset by software using the UG bit in the TIMx_EGR registers.

The next example, synchronizes 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 in master mode to send its Output Compare Reference (OC1REF) 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 the TIM2_SMCR register).
- Reset Timer 1 by writing 1 in the UG bit (TIM1_EGR register).
- Reset Timer 2 by writing 1 in the UG bit (TIM2_EGR register).
- Initialize Timer 2 to 0xE7 by writing 0xE7 in the Timer 2 counter (TIM2_CNTL).
- 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).

Figure 45. Gating Timer 2 with enable of Timer 1



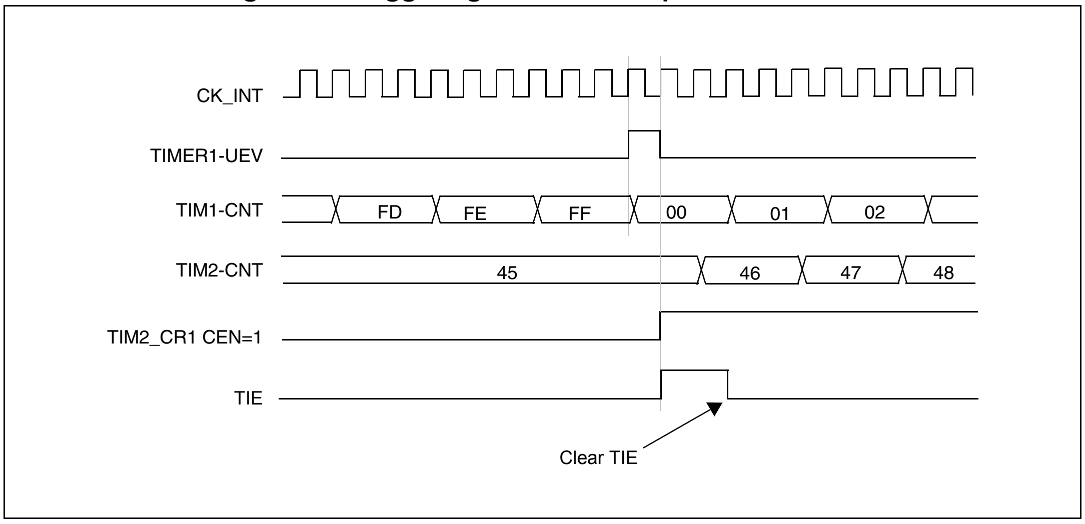
Using one timer to start the other timer

In this example, the enable of Timer 2 is set with the update event of Timer 1. Refer to *Figure 43* for connections. Timer 2 starts counting from its current value (which can be non-zero) on the divided internal clock as soon as Timer 1 generates the update event.

When Timer 2 receives the trigger signal its CEN bit is automatically set and the counter counts until 0 is written to the CEN bit in the TIM2_CR1 register. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_INT (fCK_CNT = fCK_INT/3).

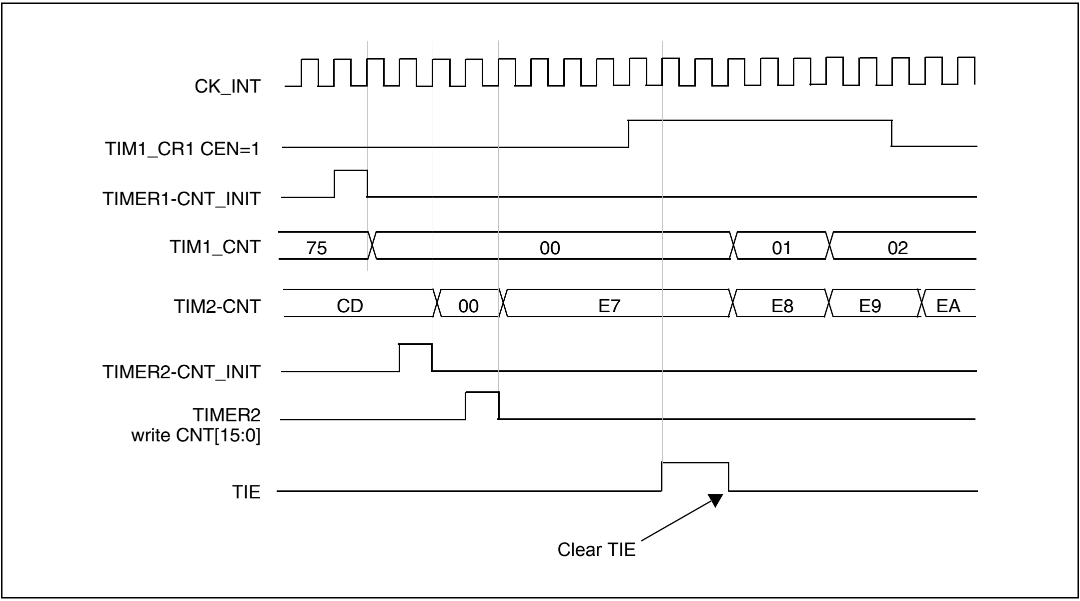
- Configure Timer 1 in master mode to send its update event as trigger output (MMS = 010 in the TIM1_CR2 register).
- Configure the Timer 1 period (TIM1_ARR register).
- Configure Timer 2 to get the input trigger from Timer 1 (TS = 000 in the TIM2_SMCR register).
- Configure Timer 2 in trigger mode (SMS = 110 in the TIM2_SMCR register).
- Start Timer 1: Write 1 in the CEN bit (TIM1_CR1 register).

Figure 46. Triggering timer 2 with update of Timer 1



As in the previous example, both counters can be initialized before starting counting. *Figure 45* shows the behavior with the same configuration shown in *Figure 46*, but in trigger mode instead of gated mode (SMS = 110 in the TIM2_SMCR register).

Figure 47. Triggering Timer 2 with enable of Timer 1



Starting both timers synchronously in response to an external trigger

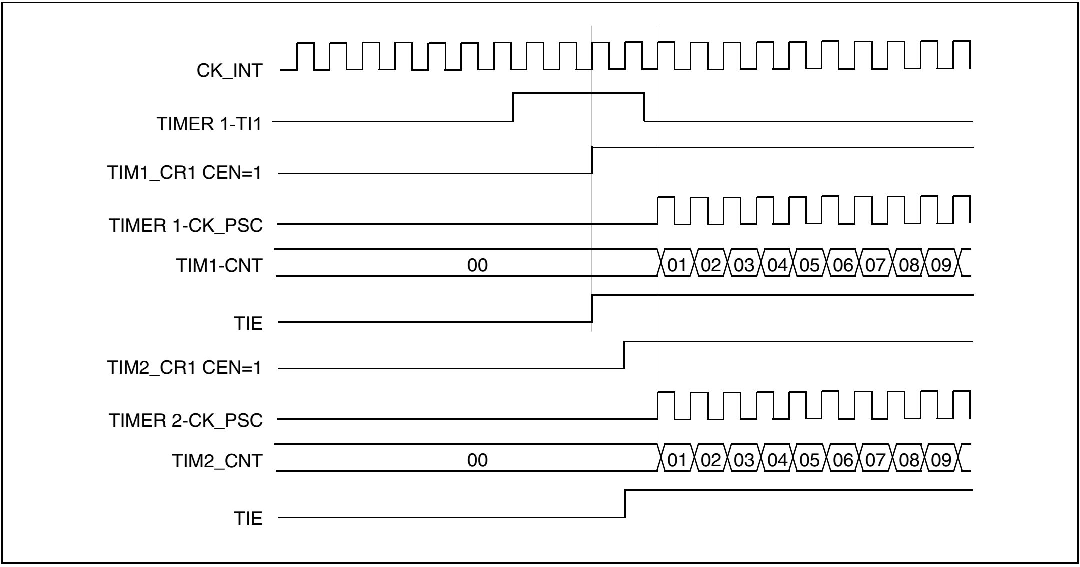
This example, sets the enable of Timer 1 when its TI1 input rises, and the enable of Timer 2 with the enable of Timer 1. Refer to Figure 43 for connections. To ensure the counters are aligned, Timer 1 must be configured in master/slave mode (slave with respect to TI1, master with respect to Timer 2):

- Configure Timer 1 in 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 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 the TIM2_SMCR register).

When a rising edge occurs on TI1 (Timer 1), both counters start counting synchronously on the internal clock and both timers' TIE flags are set.

Note: *In this example both timers are initialized before starting by setting their respective UG bits. Both counters starts from 0, but an offset can be inserted between them by writing any of the counter registers (TIMx_CNT). The master/slave mode inserts a delay between CNT_EN and CK_PSC on Timer 1.*

Figure 48. Triggering Timers 1 and 2 with Timer 1 TI1 input



10.1.15 Timer signal descriptions

Table 34. Timer signal descriptions

Signal	Internal/external	Description
CK_INT	Internal	Internal clock source: connects to STM32W108 peripheral clock (PCLK) in internal clock mode.
CK_PSC	Internal	Input to the clock prescaler.
ETR	Internal	External trigger input (used in external timer mode 2): a clock selected by EXTRIGSEL in TIMx_OR.
ETRF	Internal	External trigger: ETRP after filtering.
ETRP	Internal	External trigger: ETR after polarity selection, edge detection and prescaling.
ICy	External	Input capture or clock: TIy after filtering and edge detection.
ICyPS	Internal	Input capture signal after filtering, edge detection and prescaling: input to the capture register.
ITR0	Internal	Internal trigger input: connected to the other timer's output, TRGO.
OCy	External	Output compare: TIMxCy when used as an output. Same as OCyREF but includes possible polarity inversion.
OCyREF	Internal	Output compare reference: always active high, but may be inverted to produce OCy.
PCLK	External	Peripheral clock connects to CK_INT and used to clock input filtering. Its frequency is 12 MHz if using the 24 MHz HSE OSC and 6 Mhz if using the 12 MHz HSI RC oscillator.
TIy	Internal	Timer input: TIMxCy when used as a timer input.
TIyFPy	Internal	Timer input after filtering and polarity selection.
TIMxCy	Internal	Timer channel at a GPIO pin: can be a capture input (ICy) or a compare output (OCy).
TIMxCLK	External	Clock input (if selected) to the external trigger signal (ETR).
TIMxMSK	External	Clock mask (if enabled) AND'ed with the other timer's TIMxCLK signal.
TRGI	Internal	Trigger input for slave mode controller.

10.2 Interrupts

Several kinds of timer events can generate a timer interrupt, and each has a status flag in the TIMx_SR register to identify the reason(s) for the interrupt:

- TIE - set by a rising edge on an external trigger, either edge in gated mode
- CCyIF - set by a channel y input capture or output compare event
- UIF - set by an update event

Clear bits in TIMx_SR by writing a 1 to their bit position. When a channel is in capture mode, reading the TIMx_CCRy register will also clear the CCyIF bit.

The TIMx_IER register controls whether or not the TIMx_SR bits actually request an ARM® Cortex-M3 timer interrupt. Only the events whose bits are set to 1 in TIMx_IER can do so.

If an input capture or output compare event occurs and its CCyIM is already set, the corresponding capture/compare missed flag is set in the TIMx_MISSR register. Clear a bit in the TIMx_MISSR register by writing a 1 to it.

10.3 General-purpose timers 1 and 2 registers

10.3.1 Timer *x* interrupt and status register (TIMx_ISR)

Address offset: 0xA800 (TIM1) and 0xA804 (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			RSVD[3:0]				Reserved		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r	r	r	r			rw		rw	rw	rw	rw	rw

Bits 31:13]	Reserved, must be kept at reset value
Bits 12:9]	RSVD[3:0]: May change during normal operation
Bits 8:7]	Reserved, must be kept at reset value
Bit 6	TIF: Trigger interrupt
Bit 5	Reserved, must be kept at reset value
Bit 4	CC4IF: Capture or compare 4 interrupt pending
Bit 3	CC3IF: Capture or compare 3 interrupt pending
Bit 2	CC2IF: Capture or compare 2 interrupt pending
Bit 1	CC1IF: Capture or compare 1 interrupt pending
Bit 0	UIF: Update interrupt pending

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

10.3.4 Timer *x* control register 1 (TIMx_CR1)

Address offset: 0xE000 (TIM1) and 0xF000 (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								ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bit 7 ARPE: Auto-Reload Preload Enable
0: TIMx_ARR register is not buffered
1: TIMx_ARR register is buffered

Bits 6:5] CMS[1:0]: 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 configured output channels (CCyS=00 in TIMx_CCMRy 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 configured output channels (CCyS=00 in TIMx_CCMRy register) are set only when the counter is counting up.
11: Center-aligned mode 3. The counter counts up and down alternatively.
Output compare interrupt flags of configured output channels (CCyS=00 in TIMx_CCMRy register) are set both when the counter is counting up or down.
Note: Software may not switch from edge-aligned mode to center-aligned mode when the counter is enabled (CEN=1).

Bit 4 DIR: Direction
0: Counter used as up-counter
1: Counter used as down-counter

Bit 3 OPM: One Pulse Mode
0: Counter does not stop counting at the next update event.
1: Counter stops counting at the next update event (and clears the bit CEN).

Bit 2 URS: Update Request Source
0: When enabled, update interrupt requests are sent as soon as registers are updated (counter overflow/underflow, setting the UG bit, or update generation through the slave mode controller).
1: When enabled, update interrupt requests are sent only when the counter reaches overflow or underflow.

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

10.3.6 Timer *x* slave mode control register (TIMx_SMCR)

Address offset: 0xE008 (TIM1) and 0xF008 (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**
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bit 15 ETP: External Trigger Polarity
This bit selects whether ETR or the inverse of ETR is used for trigger operations.
0: ETR is non-inverted, active at a high level or rising edge
1: ETR is inverted, active at a low level or falling edge

Bit 14 ECE: 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 clocked by any active edge on the ETRF signal.
Note: Setting the ECE bit has the same effect as selecting external clock mode 1 with TRGI connected to ETRF (SMS=111 and TS=111). It is possible to use this mode simultaneously with the following slave modes: reset mode, gated mode and trigger mode. TRGI must not be connected to ETRF in this case (the TS bits must not be 111). If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input will be ETRF.

Bits 13:12 ETPS[1:0]: External Trigger Prescaler
External trigger signal ETRP frequency must be at most 1/4 of CK frequency. A prescaler can be enabled to reduce ETRP frequency. It is useful with fast external clocks.
00: ETRP prescaler off
01: Divide ETRP frequency by 2
10: Divide ETRP frequency by 4
11: Divide ETRP frequency by 8

Bits 11:8 ETF[3:0]: External Trigger Filter

This defines the frequency used to sample the ETRP signal, $f_{Sampling}$, and the length of the digital filter applied to ETRP. The digital filter is made of an event counter in which N events are needed to validate a transition on the output:

0000: $f_{Sampling}$ = PCLK, no filtering
0001: $f_{Sampling}$ = PCLK, N=2
0010: $f_{Sampling}$ = PCLK, N=4
0011: $f_{Sampling}$ = PCLK, N=8
0100: $f_{Sampling}$ = PCLK/2, N=
0101: $f_{Sampling}$ = PCLK/2, N=8
0110: $f_{Sampling}$ = PCLK/4, N=6
0111: $f_{Sampling}$ = PCLK/4, N=8
1111: $f_{Sampling}$ = PCLK/32, N=8
1110: $f_{Sampling}$ = PCLK/32, N=6
1101: $f_{Sampling}$ = PCLK/32, N=5
1100: $f_{Sampling}$ = PCLK/16, N=8
1011: $f_{Sampling}$ = PCLK/16, N=6
1010: $f_{Sampling}$ = PCLK/16, N=5
1001: $f_{Sampling}$ = PCLK/8, N=8
1000: $f_{Sampling}$ = PCLK/8, N=6

Note: PCLK is 12 MHz when the STM32W108 is using the 24 MHz HSE OSC, and 6 MHz if using the 12 MHz HSI RC oscillator.

Bit 7 MSM: Master/Slave Mode

0: No action

1: The effect of an event on the trigger input (TRGI) is delayed to allow exact synchronization between the current timer and the slave (through TRGO). It is useful for synchronizing timers on a single external event.

Bits 6:4 TS[2:0]: Trigger Selection

This bit field selects the trigger input used to synchronize the counter.

000: Internal Trigger 0 (ITR0)
100: TI1 Edge Detector (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 (when SMS=000) to avoid detecting spurious edges during the transition.

Bit 3 Reserved, must be kept at reset value

Bits 2:0 SMS[2:0]: 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.

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 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 signal (TRGI) >reinitializes the counter and generates an update of the registers.

101: Gated Mode. The counter clock is enabled when the trigger signal (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both starting and stopping the counter are controlled.

110: Trigger Mode. The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only starting the counter is controlled.

111: External Clock Mode 1. Rising edges of the selected trigger (TRGI) clock the counter.

Note: Gated mode must not be used if TI1F_ED is selected as the trigger input (TS=100). TI1F_ED outputs 1 pulse for each transition on TI1F, whereas gated mode checks the level of the trigger signal.

10.3.7 Timer *x* event generation register (TIMx_EGR)

Address offset: 0xE014 (TIM1) and 0xF014 (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									TG	Reserved	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

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

Bit 6 TG: Trigger Generation
0: Does nothing
1: Sets the TIM_TIF flag in the TIMx_SR register

Bit 5 Reserved, must be kept at reset value

Bit 4 CC4G: Capture/Compare 4 Generation
0: Does nothing
1: If CC4 configured as output channel, the CC4IF flag is set. If CC4 configured as input channel, the CC4IF flag is set. The CC4IM flag is set if the CC4IF flag was already high. The current value of the counter is captured in TIMx_CCR4 register.

Bit 3 CC3G: Capture/Compare 3 Generation
0: Does nothing
1: If CC3 configured as output channel, the CC3IF flag is set. If CC3 configured as input channel, the CC3IF flag is set. The CC3IM flag is set if the CC3IF flag was already high. The current value of the counter is captured in TIMx_CCR3 register.

Bit 2 CC2G: Capture/Compare 2 Generation
0: Does nothing
1: If CC2 configured as output channel, the CC2IF flag is set. If CC2 configured as input channel, the CC2IF flag is set. The CC2IM flag is set if the CC2IF flag was already high. The current value of the counter is captured in TIMx_CCR2 register.

Bit 1 CC1G: Capture/Compare 1 Generation
0: Does nothing
1: If CC1 configured as output channel, the CC1IF flag is set. If CC1 configured as input channel, the CC1IF flag is set. The CC1IM flag is set if the CC1IF flag was already high. The current value of the counter is captured in TIMx_CCR1 register.

Bit 0 UG: Update Generation
0: Does nothing
1: Re-initializes the counter and generates an update of the registers. This also clears the prescaler counter but the prescaler ratio is not affected. The counter is cleared if center-aligned mode is selected or if DIR=0 (up-counting), otherwise it takes the auto-reload value (TIM1_ARR) if DIR=1 (down-counting).

10.3.8 Timer x capture/compare mode register 1 (TIMx_CCMR1)

Address offset: 0xE018 (TIM1) and 0xF018 (TIM2)
Reset value: 0x0000 0000

The timer channels can be programmed as inputs (capture mode) or outputs (compare mode). The direction of a channel “y” is defined by configuring the corresponding CCyS bits in this register. All other bits have different functions in input and in output mode. For a given bit:

- OCxy describes its function when the channel is configured as an output (CCyS = 0)
- ICxy describes its function when the channel is configured as an input (CCyS > 0)

In short, the same bit can have a different meaning for the input stage and for the output stage. Care should be taken.

Output compare mode

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	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Reserved	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

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

Bits 14:12 OC2M[2:0]: Output Compare 2 Mode

Defines the behavior of the output reference signal OC2REF from which OC2 derives. OC2REF is active high whereas OC2"s active level depends on the CC2P bit.
000: Frozen - The comparison between the output compare register TIMx_CCR2 and the counter TIMx_CNT has no effect on the outputs.
001: Set OC2REF to active on match. The OC2REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 2 (TIMx_CCR2).
010: Set OC2REF to inactive on match. OC2REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 2 (TIMx_CCR2).
011: Toggle - OC2REF toggles when TIMx_CNT = TIMx_CCR2
100: Force OC2REF inactive
101: Force OC2REF active
110: PWM mode 1 - In up-counting, OC2REF is active as long as TIMx_CNT < TIMx_CCR2, otherwise OC2REF is inactive. In down-counting, OC2REF is inactive if TIMx_CNT > TIMx_CCR2, otherwise OC2REF is active.
111: PWM mode 2 - In up-counting, OC2REF is inactive if TIMx_CNT < TIMx_CCR2, otherwise OC2REF is active. In down-counting, OC2REF is active if TIMx_CNT > TIMx_CCR2, otherwise it is inactive.
Note: In PWM mode 1 or 2, the OC2REF level changes only when the result of the comparison changes or when the output compare mode switches from “frozen” mode to “PWM” mode.

Bit 11	OC2PE: Output Compare 2 Preload Enable 0: Buffer register for TIMx_CCR2 is disabled. TIMx_CCR2 can be written at anytime, the new value is used by the shadow register immediately. 1: Buffer register for TIMx_CCR2 is enabled. Read/write operations access the buffer register. TIMx_CCR2 buffer value is loaded in the shadow register at each update event. *Note: The PWM mode can be used without enabling the buffer register only in one pulse mode (OPM bit set in the TIMx_CR2 register), otherwise the behavior is undefined.*
Bit 10	OC2FE: Output Compare 2 Fast Enable This bit speeds the effect of an event on the trigger in input on the OC2 output. 0: OC2 behaves normally depending on the counter and CCR2 values even when the trigger is ON. The minimum delay to activate OC2 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 the OC2 output. OC2 is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate OC2 output is reduced to 3 clock cycles. OC2FE acts only if the channel is configured in PWM 1 or PWM 2 mode.
Bits 9:8	CC2S[1:0]: Capture / Compare 1 Selection This configures the channel as an output or an input. If an input, it selects the input source. 00: Channel is an output 01: Channel is an input and is mapped to TI2 10: Channel is an input and is mapped to TI1 11: Channel is an input and is mapped to TRGI. This mode requires an internal trigger input selected by the TS bit in the TIMx_SMCR register. *Note: CC2S may be written only when the channel is off (CC2E = 0 in the TIMx_CCER register).*
Bit 7	Reserved, must be kept at reset value
Bits 6:4	OC1M[2:0]: Output Compare 1 Mode See OC2M description above
Bit 3	OC1PE: Output Compare 1 Preload Enable See OC2PE description above
Bit 2	OC1FE: Output Compare 1 Fast Enable See OC2FE description above
Bits 1:0	CC1S[1:0]: Capture / Compare 1 Selection This configures the channel as an output or an input. If an input, it selects the input source. 00: Channel is an output 01: Channel is an input and is mapped to TI1 10: Channel is an input and is mapped to TI2 11: Channel is an input and is mapped to TRGI. This requires an internal trigger input selected by the TS bit in the TIMx_SMCR register. *Note: CC1S may be written only when the channel is off (CC1E = 0 in the TIMx_CCER register).*

Input capture mode

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**
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

Bits 31:16 Reserved, must be kept at reset value

Bits 15:12 IC2F[3:0]: Input Capture 1 Filter

This defines the frequency used to sample the TI2 input, Fsampling, and the length of the digital filter applied to TI2. The digital filter requires N consecutive samples in the same state before being output.
0000: Fsampling=PCLK, no filtering
0001: Fsampling=PCLK, N=2
0010: Fsampling=PCLK, N=4
0011: Fsampling=PCLK, N=8
0100: Fsampling=PCLK/2, N=6
0101: Fsampling=PCLK/2, N=8
0110: Fsampling=PCLK/4, N=6
0111: Fsampling=PCLK/4, N=8
1000: Fsampling=PCLK/8, N=6
1001: Fsampling=PCLK/8, N=8
1010: Fsampling=PCLK/16, N=5
1011: Fsampling=PCLK/16, N=6
1100: Fsampling=PCLK/16, N=8
1101: Fsampling=PCLK/32, N=5
1110: Fsampling=PCLK/32, N=6
1111: Fsampling=PCLK/32, N=8
Note: PCLK is 12 MHz when using the 24 MHz HSE OSC, and 6 MHz using the 12 MHz HSI RC oscillator.

Bits 11:10 IC2PSC[1:0]: Input Capture 1 Prescaler

00: No prescaling, capture each time an edge is detected on the capture input
01: Capture once every 2 events
10: Capture once every 4 events
11: Capture once every 6 events

Bits 9:8 CC2S[1:0]: Capture / Compare 1 Selection

This configures the channel as an output or an input. If an input, it selects the input source.
00: Channel is an output
01: Channel is an input and is mapped to TI2
10: Channel is an input and is mapped to TI1
11: Channel is an input and is mapped to TRGI. This mode requires an internal trigger input selected by the TS bit in the TIMx_SMCR register.
Note: CC2S may be written only when the channel is off (CC2E = 0 in the TIMx_CCER register).

Bits 7:4	IC1F[3:0]: Input Capture 1 Filter See IC2F description above
Bits 3:2	IC1PSC[1:0]: Input Capture 1 Prescaler See IC2PSC description above
Bits 1:0	CC1S[1:0]: Capture / Compare 1 Selection This configures the channel as an output or an input. If an input, it selects the input source. 00: Channel is an output 01: Channel is an input and is mapped to TI1 10: Channel is an input and is mapped to TI2 11: Channel is an input and is mapped to TRGI. This requires an internal trigger input selected by the TS bit in the TIMx_SMCR register. *Note: CC1S may be written only when the channel is off (CC1E = 0 in the TIMx_CCER register).*

10.3.9 Timer x capture/compare mode register 2 (TIMx_CCMR2)

Address offset: 0xE01C (TIM1) and 0xF01C (TIM2)
Reset value: 0x0000 0000

The timer channels can be programmed as inputs (capture mode) or outputs (compare mode). The direction of a channel “y” is defined by configuring the corresponding CCyS bits in this register. All other bits have different functions in input and in output mode. For a given bit:

- OCxy describes its function when the channel is configured as an output (CCyS = 0)
- ICxy describes its function when the channel is configured as an input (CCyS > 0)

In short, the same bit can have a different meaning for the input stage and for the output stage. Care should be taken.

Output compare mode

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	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		Reserved	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

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

Bits 14:12 OC4M[2:0]: Output Compare 4 Mode

Define the behavior of the output reference signal OC4REF from which OC4 derives. OC4REF is active high whereas OC4’s active level depends on the CC4P bit.

000: Frozen - The comparison between the output compare register TIMx_CCR4 and the counter TIMx_CNT has no effect on the outputs.

001: Set OC4REF to active on match. The OC4REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 4 (TIMx_CCR4).

010: Set OC4REF to inactive on match. OC4REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 4 (TIMx_CCR4).

011: Toggle - OC4REF toggles when TIMx_CNT = TIMx_CCR4

100: Force OC4REF inactive

101: Force OC4REF active

110: PWM mode 1 - In up-counting, OC4REF is active as long as TIMx_CNT < TIMx_CCR4, otherwise OC4REF is inactive. In down-counting, OC4REF is inactive if TIMx_CNT > TIMx_CCR4, otherwise OC4REF is active.

111: PWM mode 2 - In up-counting, OC4REF is inactive if TIMx_CNT < TIMx_CCR4, otherwise OC4REF is active. In down-counting, OC4REF is active if TIMx_CNT > TIMx_CCR4, otherwise it is inactive.

Note: In PWM mode 1 or 2, the OC4REF level changes only when the result of the comparison changes or when the output compare mode switches from “frozen” mode to “PWM” mode.

Bit 11 OC4PE: Output Compare 4 Preload Enable

0: Buffer register for TIMx_CCR4 is disabled. TIMx_CCR4 can be written at anytime, the new value is used by the shadow register immediately.

1: Buffer register for TIMx_CCR4 is enabled. Read/write operations access the buffer register. TIMx_CCR4 buffer value is loaded in the shadow register at each update event.

Note: The PWM mode can be used without enabling the buffer register only in one pulse mode (OPM bit set in the TIMx_CR2 register), otherwise the behavior is undefined.

Bit 10 OC4FE: Output Compare 4 Fast Enable

This bit speeds the effect of an event on the trigger in input on the OC4 output.

0: OC4 behaves normally depending on the counter and CCR4 values even when the trigger is ON. The minimum delay to activate OC4 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 the OC4 output. OC4 is set to the compare level independently from the result of the comparison. Delay to sample the trigger input and to activate OC4 output is reduced to 3 clock cycles. OC4FE acts only if the channel is configured in PWM 1 or PWM 2 mode.

Bits 9:8 CC4S[1:0]: Capture / Compare 1 Selection

This configures the channel as an output or an input. If an input, it selects the input source.

00: Channel is an output

01: Channel is an input and is mapped to TI4

10: Channel is an input and is mapped to TI3

11: Channel is an input and is mapped to TRGI. This mode requires an internal trigger input selected by the TS bit in the TIMx_SMCR register.

Note: CC2S may be written only when the channel is off (CC2E = 0 in the TIMx_CCER register).

Bit 7 Reserved, must be kept at reset value

Bits 6:4 OC3M[2:0]: Output Compare 1 Mode

See OC4M description above

Bit 3 OC3PE: Output Compare 3 Preload Enable

See OC4PE description above

Bit 2 OC3FE: Output Compare 3 Fast Enable

See OC4FE description above

Bits 1:0 CC3S[1:0]: Capture / Compare 3 Selection

This configures the channel as an output or an input. If an input, it selects the input source.

00: Channel is an output

01: Channel is an input and is mapped to TI3

10: Channel is an input and is mapped to TI4

11: Channel is an input and is mapped to TRGI. This requires an internal trigger input selected by the TS bit in the TIMx_SMCR register.

Note: CC3S may be written only when the channel is off (CC3E = 0 in the TIMx_CCER register).

Input capture mode

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**
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:12 IC4F[3:0]: Input Capture 1 Filter

This defines the frequency used to sample the TI4 input, $f_{Sampling}$, and the length of the digital filter applied to TI4. The digital filter requires N consecutive samples in the same state before being output.

0000: $f_{Sampling}$ = PCLK, no filtering
0001: $f_{Sampling}$ = PCLK, N=2
0010: $f_{Sampling}$ = PCLK, N=4
0011: $f_{Sampling}$ = PCLK, N=8
0100: $f_{Sampling}$ = PCLK/2, N=6
0101: $f_{Sampling}$ = PCLK/2, N=8
0110: $f_{Sampling}$ = PCLK/4, N=6
0111: $f_{Sampling}$ = PCLK/4, N=8
1000: $f_{Sampling}$ = PCLK/8, N=6
1001: $f_{Sampling}$ = PCLK/8, N=8
1010: $f_{Sampling}$ = PCLK/16, N=5
1011: $f_{Sampling}$ = PCLK/16, N=6
1100: $f_{Sampling}$ = PCLK/16, N=8.
1101: $f_{Sampling}$ = PCLK/32, N=5
1110: $f_{Sampling}$ = PCLK/32, N=6
1111: $f_{Sampling}$ = PCLK/32, N=8

Note: PCLK is 12 MHz when using the 24 MHz HSE OSC, and 6 MHz using the 12 MHz HSI RC oscillator.

Bits 11:10 IC4PSC[1:0]: Input Capture 1 Prescaler

00: No prescaling, capture each time an edge is detected on the capture input
01: Capture once every 2 events
10: Capture once every 4 events
11: Capture once every 6 events

Bits 9:8 CC4S[1:0]: Capture / Compare 1 Selection

This configures the channel as an output or an input. If an input, it selects the input source.
00: Channel is an output
01: Channel is an input and is mapped to TI4
10: Channel is an input and is mapped to TI3
11: Channel is an input and is mapped to TRGI. This mode requires an internal trigger input selected by the TS bit in the TIMx_SMCR register.

Note: CC2S may be written only when the channel is off (CC2E = 0 in the TIMx_CCER register).

Bits 7:4 IC3F[3:0]: Input Capture 1 Filter

See IC4F description above

Bits 3:2 IC3PSC[1:0]: Input Capture 1 Prescaler

See IC4PSC description above

Bits 1:0 CC3S[1:0]: Capture / Compare 3 Selection

This configures the channel as an output or an input. If an input, it selects the input source.

00: Channel is an output

01: Channel is an input and is mapped to TI3

10: Channel is an input and is mapped to TI4

11: Channel is an input and is mapped to TRGI. This requires an internal trigger input selected by the TS bit in the TIMx_SMCR register.

Note: CC3S may be written only when the channel is off (CC3E = 0 in the TIMx_CCER register).

10.3.10 Timer *x* capture/compare enable register (TIMx_CCER)

Address offset: 0xE020 (TIM1) and 0xF020 (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		CC4P	CC4E	Reserved		CC3P	CC3E	Reserved		CC2P	CC2E	Reserved		CC1P	CC1E
		rw	rw			rw	rw			rw	rw			rw	rw

Bits 31:14 Reserved, must be kept at reset value

Bit 13 CC4P: Capture/Compare 4 output Polarity
If CC4 is configured as an output channel:
0: OC4 is active high
1: OC4 is active low.

If CC4 configured as an input channel:
0: IC4 is not inverted. Capture occurs on a rising edge of IC4. When used as an external trigger, IC4 is not inverted.
0: IC4 is inverted. Capture occurs on a falling edge of IC4. When used as an external trigger, IC4 is inverted.
1: Capture is enabled

Bit 12 CC4E: Capture/Compare 4 output Enable
If CC4 is configured as an output channel:
0: OC4 is disabled
1: OC4 is enabled

If CC4 configured as an input channel:
0: Capture is disabled
1: Capture is enabled

Bits 11:10 Reserved, must be kept at reset value

Bit 9 CC3P: Refer to the CC4P description above

Bit 8 CC3E: Refer to the CC4E description above

Bits 7:6 Reserved, must be kept at reset value

Bit 5 CC2P: Refer to the CC4P description above

Bit 4 CC2E: Refer to the CC4E description above

Bits 3:2 Reserved, must be kept at reset value

Bit 1 CC1P: Refer to the CC4P description above

Bit 0 CC1E: Refer to the CC4E description above

10.3.11 Timer *x* counter register (TIMx_CNT)

Address offset: 0xE024 (TIM1) and 0xF024 (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**
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 CNT[15:0]: Counter value

10.3.12 Timer *x* prescaler register (TIMx_PSC)

Address offset: 0xE028 (TIM1) and 0xF028 (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												PSC[3:0]			
												rw	rw	rw	rw

Bits 31:4 Reserved, must be kept at reset value

Bits 3:0 PSC[3:0]: Prescaler value

The prescaler divides the internal timer clock frequency. The counter clock frequency CK_CNT is equal to fCK_PSC / (2 ^ PSC[3:0]). Clock division factors can range from 1 through 32768. The division factor is loaded into the shadow prescaler register at each update event (including when the counter is cleared through UG bit of TIM1_EGR register or through the trigger controller when configured in reset mode).

10.3.13 Timer *x* auto-reload register (TIMx_ARR)

Address offset: 0xE02C (TIM1) and 0xF02C (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
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16] Reserved, must be kept at reset value

Bits 15:0 ARR[15:0]: Auto-reload value

ARR[15:0] is the value to be loaded in the shadow auto-reload register.

The auto-reload register is buffered. Writing or reading the auto-reload register accesses the buffer register. The content of the buffer register is transferred in the shadow register permanently or at each update event UEV, depending on the auto-reload buffer enable bit (ARPE) in TIMx_CR1 register. The update event is sent when the counter reaches the overflow point (or underflow point when down-counting) and if the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software. The counter is blocked while the auto-reload value is 0.

10.3.14 Timer *x* capture/compare 1 register (TIMx_CCR1)

Address offset: 0xE034 (TIM1) and 0xF034 (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
CCR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 CCR[15:0]: Capture/compare value

If the CC1 channel is configured as an output (CC1S = 0):

CCR1 is the buffer value to be loaded in the actual capture/compare 1 register. It is loaded permanently if the preload feature is not selected in the TIM1_CCMR1 register (bit OC1PE). Otherwise the buffer value is copied to the shadow capture/compare 1 register when an update event occurs. The active capture/compare register contains the value to be compared to the counter TIM1_CNT and signaled on the OC1 output. If the CC1 channel is configured as an input (CC1S is not 0): CCR1 is the counter value transferred by the last input capture 1 event (IC1).

10.3.15 Timer *x* capture/compare 2 register (TIMx_CCR2)

Address offset: 0xE038 (TIM1) and 0xF038 (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**
CCR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	Reserved, must be kept at reset value
Bits 15:0	See description in the TIMx_CCR1 register

10.3.16 Timer *x* capture/compare 3 register (TIMx_CCR3)

Address offset: 0xE03C (TIM1) and 0xF03C (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**
CCR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	Reserved, must be kept at reset value
Bits 15:0	See description in the TIMx_CCR1 register

10.3.17 Timer *x* capture/compare 4 register (TIMx_CCR4)

Address offset: 0xE040 (TIM1) and 0xF040 (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
CCR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 See description in the TIMx_CCR1 register

10.3.18 Timer 1 option register (TIM1_OR)

Address offset: 0xE050
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												OR RSVD	CLK MSKEN	EXTRIGSEL[1:0]	
												rw	rw	rw	rw

Bits 31:4 Reserved, must be kept at reset value

Bit 3 ORRSVD

Reserved: this bit must always be set to 0

Bit 2 CLKMSKEN

Enables TIM1MSK when TIM1CLK is selected as the external trigger: 0 = TIM1MSK not used, 1 = TIM1CLK is ANDed with the TIM1MSK input.

Bits 1:0 EXTRIGSEL[1:0]:

Selects the external trigger used in external clock mode 2: 0 = PCLK, 1 = calibrated 1 kHz clock, 2 = 32 kHz reference clock (if available), 3 = TIM1CLK pin.

10.3.19 Timer 2 option register (TIM2_OR)

Address offset: 0xF050
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								RE MAPC4	RE MAPC3	RE MAPC2	RE MAPC1	OR RSVD	CLK MSKEN	EXTRIGSEL[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

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

Bit 7 REMAPC4
Selects the GPIO used for TIM2_CH4: 0 = PA2, 1 = PB4

Bit 6 REMAPC3
Selects the GPIO used for TIM2_CH3: 0 = PA1, 1 = PB3

Bit 5 REMAPC2
Selects the GPIO used for TIM2_CH2: 0 = PA3, 1 = PB2

Bit 4 REMAPC1
Selects the GPIO used for TIM2_CH1: 0 = PA0, 1 = PB1

Bit 3 ORRSVD
Reserved: this bit must always be set to 0

Bit 2 CLKMSKEN
Enables TIM2MSK when TIM2CLK is selected as the external trigger: 0 = TIM2MSK not used, 1 = TIM2CLK is ANDed with the TIM2MSK input.

Bits 1:0 EXTRIGSEL[1:0]:
Selects the external trigger used in external clock mode 2: 0 = PCLK, 1 = calibrated 1 kHz clock, 2 = 32 kHz reference clock (if available), 3 = TIM2CLK pin.

10.3.20 General-purpose timers 1 and 2 (TIM1/TIM2) register map

Table 35 gives the TIM1/TIM2 register map and reset values.

Table 35. TIM1/TIM2 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA800	TIM1_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSVD[3:0]				Res.	Res.	TIP	Res.	CC4IP	CC3IP	CC2IP	CC1IP	UIP
	Reset value																				0	0	0	0			0		0	0	0	0	0
0xA804-0xA814	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA818	TIM1_MISSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4IM	CC3IM	CC2IM	CC1IM	Res.	Res.	RSVD[6:0]						
	Reset value																				0	0	0	0			0	0	0	0	0	0	0
0xA81C-0xA83C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA840	TIM1_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																										0		0	0	0	0	0
0xA844-0xDFFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xE000	TIM1_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																									0	0	0	0	0	0	0	0
0xE004	TIM1_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			Res.	Res.	Res.	Res.
	Reset value																									0	0	0	0				
0xE008	TIM1_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0xE014	TIM1_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0xE018	TIM1_CCMR1 Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIM1_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F [3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE01C	TIM1_CCMR2 Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		Res.	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value																		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIM1_CCMR2 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F [3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F [3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE020	TIM1_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4P	CC4E	Res.	Res.	CC3P	CC3E	Res.	Res.	CC2P	CC2E	Res.	Res.	CC1P	CC1E
	Reset value																			0	0			0	0			0	0			0	0

Table 35. TIM1/TIM2 register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xE024	TIM1_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE028	TIM1_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE02C	TIM1_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE030	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xE034	TIM1_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE038	TIM1_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE03C	TIM1_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE040	TIM1_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE044-0xE04C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xE050	TIM1_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ORRSVD	CLKMSKEN	EXTRIGSEL[1:0]	
	Reset value																													0	0	0	0
0xA804	TIM2_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSVD[3:0]				Res.	Res.	TIP	Res.	CC4IP	CC3IP	CC2IP	CC1IP	UIP
	Reset value																				0	0	0	0			0		0	0	0	0	0
0xA808-0xA810	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA81C	TIM2_MISSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4IM	CC3IM	CC2IM	CC1IM	Res.	Res.	RSVD[6:0]						
	Reset value																				0	0	0	0			0	0	0	0	0	0	0
0xA820-0xA840	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA844	TIM2_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																										0		0	0	0	0	0
0xA848-0xEFFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xF000	TIM2_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																									0	0	0	0	0	0	0	0

Table 35. TIM1/TIM2 register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xF004	TIM2_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			Res.	Res.	Res.	Res.
	Reset value																									0	0	0	0				
0xF008	TIM2_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0xF014	TIM2_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0xF018	TIM2_CCMR1 Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIM2_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F [3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF01C	TIM2_CCMR2 Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		Res.	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value																		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIM2_CCMR2 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F [3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F [3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF020	TIM2_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4P	CC4E	Res.	Res.	CC3P	CC3E	Res.	Res.	CC2P	CC2E	Res.	Res.	CC1P	CC1E
	Reset value																			0	0			0	0			0	0			0	0
0xF024	TIM2_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF028	TIM2_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF02C	TIM2_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF030	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xF034	TIM2_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF038	TIM2_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF03C	TIM2_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF040	TIM2_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 35. TIM1/TIM2 register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xF044-0xF04C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xF050	TIM2_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REMAPC4	REMAPC3	REMAPC2	REMAPC1	ORRSVD	CLKMSKEN	EXTRIGSEL[1:0]	
	Reset value																									0	0	0	0	0	0	0	0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

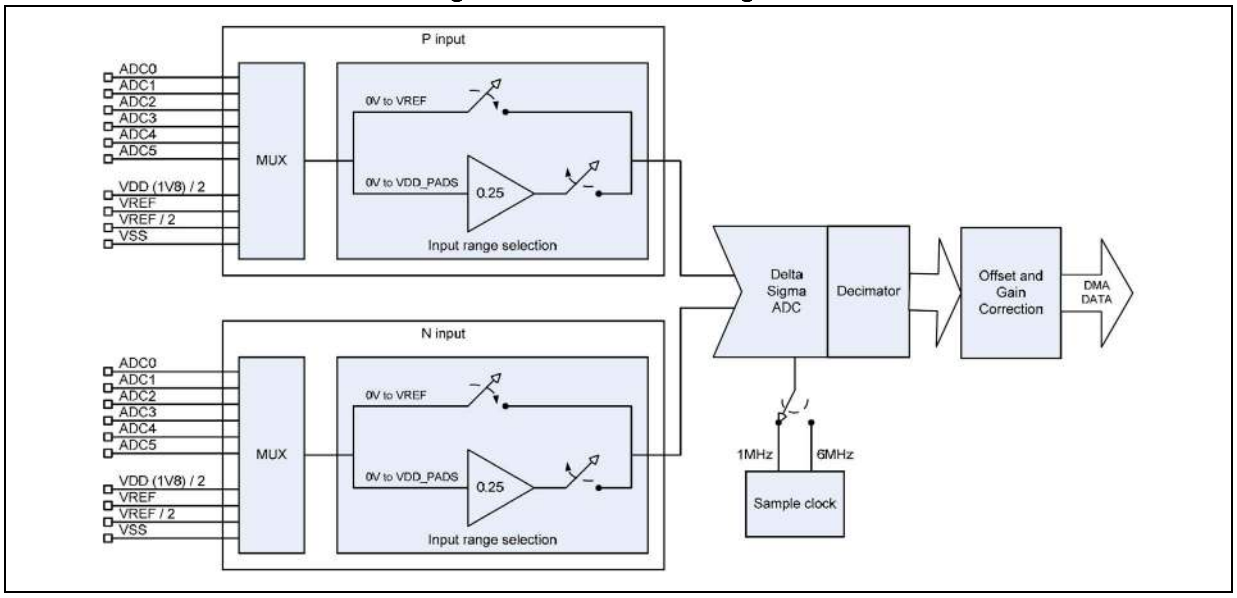
11 Analog-to-digital converter

The STM32W108 analog-to-digital converter (ADC) is a first-order sigma-delta converter with the following features:

- Resolution of up to 12 bits
- Sample times as fast as 5.33 µs (188 kHz)
- Differential and single-ended conversions from six external and four internal sources
- Two voltage ranges (differential): -VREF to +VREF, and –VDD_PADS to +VDD_PADS
- Choice of internal or external VREF: internal VREF may be output
- Digital offset and gain correction
- Dedicated DMA channel with one-shot and continuous operating modes

Figure 49 shows the basic ADC structure.

Figure 49. ADC block diagram



While the ADC Module supports both single-ended and differential inputs, the ADC input stage always operates in differential mode. Single-ended conversions are performed by connecting one of the differential inputs to VREF/2 while fully differential operation uses two external inputs.

Note: In high voltage mode, input buffers (with 0.25 gain only) may experience long term drift of its input offset voltage that affects ADC accuracy. In these cases, only the 1.2V input range mode of the ADC should be used. If measurement of signals >1.2V is required, then external attenuation should be added.

11.1 Functional description

11.1.1 Setup and configuration

To use the ADC follow this procedure, described in more detail in the next sections:

- Configure any GPIO pins to be used by the ADC in analog mode.
- Configure the voltage reference (internal or external).
- Set the offset and gain values.
- Reset the ADC DMA, define the DMA buffer, and start the DMA in the proper transfer mode.
- If interrupts will be used, configure the primary ADC interrupt and specific mask bits.
- Write the ADC configuration register to define the inputs, voltage range, sample time, and start the conversions.

11.1.2 GPIO usage

A GPIO pin used by the ADC as an input or voltage reference must be configured in analog mode by writing 0 to its 4-bit field in the proper GPIOx_CRH/L register. Note that a GPIO pin in analog mode cannot be used for any digital functions, and software always reads it as 1.

Table 36. ADC GPIO pin usage

Analog Signal	GPIO	Configuration control
ADC0 input	PB5	GPIOB_CRH[7:4]
ADC1 input	PB6	GPIOB_CRH[11:8]
ADC2 input	PB7	GPIOB_CRH[15:12]
ADC3 input	PC1	GPIOC_CRH[7:4]
ADC4 input	PA4	GPIOA_CRH[3:0]
ADC5 input	PA5	GPIOA_CRH[7:4]
VREF input or output	PB0	GPIOB_CRH[3:0]

See *Section 8: General-purpose input/output on page 89* for more information about how to configure the GPIO pins.

11.1.3 Voltage reference

The ADC voltage reference (VREF), may be internally generated or externally sourced from PB0. If internally generated, it may optionally be output on PB0.

To use an external reference, an ST system function must be called after reset and after waking from deep sleep. PB0 must also be configured in analog mode using GPIOB_CRH[3:0]. See the STM32W108 HAL documentation for more information on the system functions required to use an external reference.

11.1.4 Offset/gain correction

When a conversion is complete, the 16-bit converted data is processed by offset/gain correction logic:

- The basic ADC conversion result is added to the 16-bit signed (two's complement) value of the ADC offset register (ADC_OFFSETR).
- The offset-corrected data is multiplied by the 16-bit ADC gain register, ADC_GAINR, to produce a 16-bit signed result. If the product is greater than 0x7FFF (32767), or less than 0x8000 (-32768), it is limited to that value and the SAT bit is set in the ADC_ISR register.

ADC_GAINR is an unsigned scaled 16-bit value: GAIN[15] is the integer part of the gain factor and GAIN[14:0] is the fractional part. As a result, ADC_GAINR values can represent gain factors from 0 through ($2 – 2^{-15}$).

Reset initializes the offset to zero (ADC_OFFSETR = 0) and gain factor to one (ADC_GAINR = 0x8000).

11.1.5 DMA

The ADC DMA channel writes converted data, which incorporates the offset/gain correction, into a DMA buffer in RAM.

The ADC DMA buffer is defined by two registers:

- ADC_DMAMSAR is the start address of the buffer and must be even.
- ADC_DMANDTR specifies the size of the buffer in 16-bit samples, or half its length in bytes.

To prepare the DMA channel for operation, reset it by writing the RST bit in the ADC_DMACR register, then start the DMA in either linear or auto wrap mode by setting the LOAD bit in the ADC_DMACR register. The AUTOWRAP bit in the ADC_DMACR register selects the DMA mode: 0 for linear mode, 1 for auto wrap mode.

- In linear mode the DMA writes to the buffer until the number of samples given by ADC_DMANDTR has been output. Then the DMA stops and sets the DMABF bit in the ADC_ISR register. If another ADC conversion completes before the DMA is reset or the ADC is disabled, the DMAOVF bit in the ADC_ISR register is set.
- In auto wrap mode the DMA writes to the buffer until it reaches the end, then resets its pointer to the start of the buffer and continues writing samples. The DMA transfers continue until the ADC is disabled or the DMA is reset.

When the DMA fills the lower and upper halves of the buffer, it sets the DMABHF and DMABF bits, respectively, in the ADC_ISR register. The current location to which the DMA is writing can also be determined by reading the ADC_DMAMNAR register.

11.1.6 ADC configuration register

The ADC configuration register (ADC_CR) sets up most of the ADC operating parameters.

Input

The analog input of the ADC can be chosen from various sources. The analog input is configured with the CHSELP[3:0] and CHSELN[3:0] bits within the ADC_CR register. *Table 37* shows the possible input selections.

Table 37. ADC inputs

CHSELP[3:0]/ CHSELN[3:0] (1)	Analog source at ADC	GPIO pin	Purpose
0	ADC0	PB5	
1	ADC1	PB6	
2	ADC2	PB7	
3	ADC3	PC0	
4	ADC4	PA4	
5	ADC5	PA5	
6	No connection		
7	No connection		
8	GND	Internal connection	Calibration
9	VREF/2	Internal connection	Calibration
10	VREF	Internal connection	Calibration
11	1V8 VREG/2	Internal connection	Supply monitoring and calibration
12	No connection		
13	No connection		
14	No connection		
15	No connection		

1. Denotes bits CHSELP[3:0] or CHSELN[3:0] in register ADC_CR.

Table 38 shows the typical configurations of ADC inputs.

Table 38. Typical ADC input configurations

ADC P input	ADC N input	CHSELP[3:0]	CHSELN[3:0]	Purpose
ADC0	VREF/2	0	9	Single-ended
ADC1	VREF/2	1	9	Single-ended
ADC2	VREF/2	2	9	Single-ended
ADC3	VREF/2	3	9	Single-ended
ADC4	VREF/2	4	9	Single-ended
ADC5	VREF/2	5	9	Single-ended
ADC1	ADC0	1	0	Differential
ADC3	ADC2	3	2	Differential
ADC5	ADC4	5	4	Differential
GND	VREF/2	8	9	Calibration
VREF	VREF/2	10	9	Calibration
VDD_PADSA/2	VREF/2	11	9	Calibration

Input range

ADC inputs can be routed through input buffers to expand the input voltage range. The input buffers have a fixed 0.25 gain and the converted data is scaled by that factor.

With the input buffers disabled the single-ended input range is 0 to VREF and the differential input range is -VREF to +VREF. With the input buffers enabled the single-ended range is 0 to VDD_PADS and the differential range is -VDD_PADS to +VDD_PADS.

The input buffers are enabled for the ADC P and N inputs by setting the HVSELP and HVSELN bits respectively, in the ADC_CR register. The ADC accuracy is reduced when the input buffer is selected.

Sample time

ADC sample time is programmed by selecting the sampling clock and the clocks per sample.

- The sampling clock may be either 1 MHz or 6 MHz. If the CLK bit in the ADC_CR register is clear, the 6 MHz clock is used; if it is set, the 1 MHz clock is selected. The 6 MHz sample clock offers faster conversion times but the ADC resolution is lower than that achieved with the 1 MHz clock.
- The number of clocks per sample is determined by the SMP[2:0] bits in the ADC_CR register. SMP[2:0] values select from 32 to 4096 sampling clocks in powers of two. Longer sample times produce more significant bits. Regardless of the sample time, converted samples are always 16-bits in size with the significant bits left-aligned within the value.

Table 39 shows the options for ADC sample times and the significant bits in the conversion results.

Table 39. ADC sample times

SMP[2:0]	Sample clocks	Sample time (µs)		Sample frequency (kHz)		Significant bits
		1 MHz clock	6 MHz clock	1 MHz clock	6 MHz clock	
0	32	32	5.33	31.3	188	5
1	64	64	10.7	15.6	93.8	6
2	128	128	21.3	7.81	46.9	7
3	256	256	42.7	3.91	23.4	8
4	512	512	85.3	1.95	11.7	9
5	1024	1024	170	0.977	5.86	10
6	2048	2048	341	0.488	2.93	11
7	4096	4096	682	0.244	1.47	12

Note: ADC sample timing is the same whether the STM32W108 is using the 24 MHz HSE OSC or the 12 MHz HSI RC oscillator. This facilitates using the ADC soon after the CPU wakes from deep sleep, before switching to the crystal oscillator.

11.1.7 Operation

Setting the ADC_EN bit in the ADC_CR register enables the ADC; once enabled, it performs conversions continuously until it is disabled. If the ADC had previously been disabled, a 21 µs analog startup delay is imposed before the ADC starts conversions. The delay timing is performed in hardware and is simply added to the time until the first conversion result is output.

When the ADC is first enabled, and or if any change is made to ADC_CR after it is enabled, the time until a result is output is double the normal sample time. This is because the ADC's internal design requires it to discard the first conversion after startup or a configuration change. This is done automatically and is hidden from software except for the longer timing. Switching the processor clock between the RC and crystal oscillator also causes the ADC to go through this startup cycle. If the ADC was newly enabled, the analog delay time is added to the doubled sample time.

If the DMA is running when ADC_CR is modified, the DMA does not stop, so the DMA buffer may contain conversion results from both the old and new configurations.

The following procedure illustrates a simple polled method of using the ADC. After completing the procedure, the latest conversion results is available in the location written to by the DMA. This assumes that any GPIOs and the voltage reference have already been configured.

1. Allocate a 16-bit signed variable, for example analogData, to receive the ADC output. (Make sure that analogData is half-word aligned – that is, at an even address.)
2. Disable all ADC interrupts – write 0 to ADC_IER.
3. Set up the DMA to output conversion results to the variable, analogData.
   Reset the DMA – set the RST bit in ADC_DMACR.
   Define a one sample buffer – write analogData's address to ADC_DMAMSAR, set ADC_DMANDTR to 1.
4. Write the desired offset and gain correction values to the ADC_OFFSETR and ADC_GAINR registers.
5. Start the ADC and the DMA.
   Write the desired conversion configuration, with the ADC_EN bit set, to ADC_CR.
   Clear the ADC buffer full flag – write DMABF to ADC_ISR.
   Start the DMA in auto wrap mode – set the AUTOWRAP and LOAD bits in ADC_DMACR.
6. Wait until the DMABF bit is set in ADC_ISR, then read the result from analogData.

To convert multiple inputs using this approach, repeat Steps 4 through 6, loading the desired input configurations to ADC_CR in Step 5. If the inputs can use the same offset/gain correction, just repeat Steps 5 and 6.

11.1.8 Calibration

Sampling of internal connections GND, VREF/2, and VREF allow for offset and gain calibration of the ADC in applications where absolute accuracy is important. Offset error is calculated from the minimum input and gain error is calculated from the full scale input range. Correction using VREF is recommended because VREF is calibrated by the ST software against VDD_PADSA. The VDD_PADSA regulator is trimmed to 1.80 V ± 50 mV. If better absolute accuracy is required, the ADC can be configured to use an external reference. The ADC calibrates as a single-ended measurement. Differential signals require correction of both their inputs.

Table 40 shows the equations used to calculate the gain and offset correction values.

Table 40. ADC gain and offset correction equations

Calibration	Correction value
Gain, buffer disabled	$0 \times 8000 \times \frac{(N_{VREF} - N_{GND})}{0 \times 4000}$
Gain, buffer enabled	$0 \times 8000 \times \frac{(N_{VREF} - N_{VREF/2})}{0 \times 2000} \times \frac{1}{4}$
Offset, buffer disabled (after applying gain correction)	$\frac{1}{2} \times (N_{GND} - 0xE000)$
Offset, buffer enabled (after applying gain correction)	$\frac{1}{2} \times (N_{VREF/2} - 0xE800)$

Equation notes

- All N are 16-bit two's complement numbers.
- N_{GND} is a sampling of ground. Due to the ADC's internal design, VGND does not yield the minimum two's complement value 0x8000 as the conversion result. Instead, VGND yields a two's complement value close to 0xE000 when the input buffer is not selected. VGND cannot be measured when the input buffer is enabled because it is outside the buffer's input range.
- N_{VREF} is a sampling of VREF. Due to the ADC's internal design, VREF does not yield the maximum positive two's complement 0x7FFF as the conversion result. Instead, VREF yields a two's complement value close to 0x2000 when the input buffer is not selected and yields a two's complement value close to 0xF000 when the input buffer is selected.
- $N_{VREF/2}$ is a sampling of VREF/2. VREF/2 yields a two's complement value close to 0x0000 when the input buffer is not selected, and yields a two's complement value close to 0xE800 when the input buffer is selected.
- Offset correction is affected by the gain correction value. Offset correction is calculated after gain correction has been applied.

11.2 Interrupts

Four kinds of ADC events can generate an ADC interrupt, and each has a bit flag in the ADC_ISR register to identify the reason(s) for the interrupt:

- DMAOVF – an ADC conversion result was ready but the DMA was disabled (DMA buffer overflow).
- SAT – the gain correction multiplication exceeded the limits for a signed 16-bit number (gain saturation).
- DMABF – the DMA wrote to the last location in the buffer (DMA buffer full).
- DMABHF – the DMA wrote to the last location of the first half of the DMA buffer (DMA buffer half full).

Bits in ADC_ISR may be cleared by writing a 1 to their position.

The ADC_IER register controls whether or not ADC_ISR bits actually request the ARM® Cortex-M3 ADC interrupt; only the events whose bits are 1 in ADC_IER can do so.

For non-interrupt (polled) ADC operation set ADC_IER to zero, and read the bit flags in ADC_ISR to determine the ADC status.

Note: When making changes to the ADC configuration it is best to disable the DMA beforehand. If this isn’t done it can be difficult to determine at which point the sample data in the DMA buffer switch from the old configuration to the new configuration. However, since the ADC will be left running, if it completes a conversion after the DMA is disabled, the DMAOVF flag will be set. To prevent these unwanted DMA buffer overflow indications, clear the DMAOVF flag immediately after enabling the DMA, preferably with interrupts off. Disabling the ADC in addition to the DMA is often undesirable because of the additional analog startup time when it is re-enabled.

11.3 Analog-to-digital converter (ADC) registers

11.3.1 ADC interrupt status register (ADC_ISR)

Address offset: 0xA810
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											DMA OVF	SAT	DMA BF	DMA BHF	Reserved
											rw	rw	rw	rw	

Bits 31:5] Reserved, must be kept at reset value

Bit 4 DMAOVF: DMA buffer overflow interrupt pending

Bit 3 SAT: Gain correction saturation interrupt pending

Bit 2 DMABF: DMA buffer full interrupt pending

Bit 1 DMABHF: DMA buffer half full interrupt pending

Bit 0 Reserved: this bit should always be set to 1

11.3.2 ADC interrupt enable register (ADC_IER)

Address offset: 0xA850
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											DMA OVFIE	SATIE	DMA BFIE	DMA BHFIE	Reserved
											rw	rw	rw	rw	

Bits 31:5 Reserved, must be kept at reset value

Bit 4 DMAOVFIE: DMA buffer overflow interrupt enable

Bit 3 SATIE: Gain correction saturation interrupt enable

Bit 2 DMABFIE: DMA buffer full interrupt enable

Bit 1 DMABHFIE: DMA buffer half full interrupt enable

Bit 0 Reserved: this bit must always be set to 0

11.3.3 ADC control register (ADC_CR)

Address offset: 0xD004
Reset value: 0x0000 1800

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
SMP[2:0]			HVSELP	HVSELN	CHSELP[3:0]				CHSELN[3:0]				CLK	Reserved	ADON
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:13 SMP[2:0]: ADC sample time in clocks and the equivalent significant bits in the conversion
0: 32 clocks (5 bits)
1: 64 clocks (6 bits)
2: 128 clocks (7 bits)
3: 256 clocks (8 bits)
4: 512 clocks (9 bits)
5: 1024 clocks (10 bits)
6: 2048 clocks (11 bits)
7: 4096 clocks (12 bits)

Bit 12 HVSELP: Select voltage range for the P input channel
0: Low voltage range (input buffer disabled)
1: High voltage range (input buffer enabled)

Bit 11 HVSELN: Select voltage range for the N input channel
0: Low voltage range (input buffer disabled).
1: High voltage range (input buffer enabled).

Bits 10:7 CHSELP[3:0]: Input selection for the P channel
0x0: PB5 pin
0x1: PB6 pin
0x2: PB7 pin
0x3: PC1 pin
0x4: PA4 pin
0x5: PA5 pin
0x8: GND (0V) (not for high voltage range)
0x9: VREF/2 (0.6V)
0xA: VREF (1.2V)
0xB: VREG/2 (0.9V) (not for high voltage range)
0x6, 0x7, 0xC-0xF: Reserved, must be kept at reset value

Bits 6:3 CHSELN[3:0]: Input selection for the N channel
Refer to CHSELP[3:0] above for choices

Bit 2 CLK: Select ADC clock:
0: 6 MHz1: 1 MHz

Bit 1 Reserved: This bit must always be set to 0

Bit 0 ADON: A/D converter on/off

This bit is set and cleared by software. Write 1 to enable continuous conversions and write 0 to stop. When the ADC is started, the first conversion takes twice the usual number of clocks plus 21 microseconds. If anything in this register is modified while the ADC is running, the next conversion takes twice the usual number of clocks.

11.3.4 ADC offset register (ADC_OFFSETR)

Address offset: 0xD008
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**
OFFSET[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 OFFSET[15:0]:

16-bit signed offset added to the basic ADC conversion result before gain correction is applied.

11.3.5 ADC gain register (ADC_GAINR)

Address offset: 0xD00C
Reset value: 0x0000 8000

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**
GAIN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 GAIN[15:0]:

Gain factor that is multiplied by the offset-corrected ADC result to produce the output value. The gain is a 16-bit unsigned scaled integer value with a binary decimal point between bits 15 and 14. It can represent values from 0 to (almost) 2. The reset value is a gain factor of 1.

11.3.6 ADC DMA control register (ADC_DMACR)

Address offset: 0xD010
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											RST	Reserved		AUTO WRAP	LOAD
											w			rw	rw

Bits 31:5 Reserved, must be kept at reset value

Bit 4 RST:
Write 1 to reset the ADC DMA. This bit auto-clears.

Bits 3:2 Reserved, must be kept at reset value

Bit 1 AUTOWRAP: Selects DMA mode
0: Linear mode, the DMA stops when the buffer is full.
1: Auto-wrap mode, the DMA output wraps back to the start when the buffer is full.

Bit 0 LOAD: Loads the DMA buffer
Write 1 to start DMA (writing 0 has no effect). Cleared when DMA starts or is reset.

11.3.7 ADC DMA status register (ADC_DMASR)

Address offset: 0xD014
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														OVF	ACT
														r	r

Bits 31:2 Reserved, must be kept at reset value

Bit 1 OVF: DMA overflow
Occurs when an ADC result is ready and the DMA is not active. Cleared by DMA reset.

Bit 0 ACT: DMA status
Reads 1 if DMA is active.

11.3.8 ADC DMA memory start address register (ADC_DMAMSAR)

Address offset: 0xD018
Reset value: 0x2000 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			MSA[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]	MSA[12:0]: Memory start address

11.3.9 ADC DMA number of data to transfer register (ADC_DMANDTR)

Address offset: 0xD01C
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			NDT[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	NDT[12:0]: Number of data to transfer This is the number of 16-bit ADC conversion results the buffer can hold, not its length in bytes. (The length in bytes is twice this value).

11.3.10 ADC DMA memory next address register (ADC_DMAMNAR)

Address offset: 0xD020
Reset value: 0x2000 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		MNA[12:0]													Reserved
		r	r	r	r	r	r	r	r	r	r	r	r	r	

Bits 31:14 Reserved, must be kept at reset value

Bits 13:1 MNA[12:0]: Memory next address
The location that is written next by the DMA

Bit 0 Reserved, must be kept at reset value

11.3.11 ADC DMA count number of data transferred register (ADC_DMACNDTR)

Address offset: 0xD024
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			CNDT[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

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

Bits 12:0 CNDT[12:0]:
Count the number of DMA transferred data: the number of 16-bit conversion results that have been written to the buffer.

11.3.12 Analog-to-digital converter (ADC) register map

Table 41 gives the ADC register map and reset values.

Table 41. ADC register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA810	ADC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAOVF	SAT	DMABF	DMABHF	Res.
	Reset value																												0	0	0	0	
0xA814-0xA84C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA850	ADC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAOVFIE	SATIE	DMABFIE	DMABHFIE	Res.
	Reset value																												0	0	0	0	
0xA854-0xD000	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xD004	ADC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMP[2:0]			HVSELP	HVSELN	CHSELP[3:0]				CHSELN[3:0]				CLK	Res.	ADON
	Reset value																	0	0	0	1	1	0	0	0	0	0	0	0	0	0		0
0xD008	ADC_OFFSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD00C	ADC_GAINR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GAIN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD010	ADC_DMACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RST	Res.	Res.	AUTOWRAP	LOAD
	Reset value																												0			0	0
0xD014	ADC_DMASR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVF	ACT
	Reset value																															0	0
0xD018	ADC_DMAMSAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSA[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xD01C	ADC_DMANDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0xD020	ADC_DMAMNAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MNA[12:0]													Res.
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	
0xD024	ADC_DMACNDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNDT[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0

Refer to *Figure 3: STM32W108 memory mapping* and *Table 3: STM32W108xx peripheral register boundary addresses* for the register boundary addresses of the peripherals available in all STM32W108xx devices.

12 Interrupts

The interrupt system of the STM32W108 is composed of two parts:

- A standard ARM® Cortex-M3 Nested Vectored Interrupt Controller (NVIC) that provides top level interrupts
- An Event Manager (EM) that provides second level interrupts.

The NVIC and EM provide a simple hierarchy. All second level interrupts from the EM feed into top level interrupts in the NVIC. This two level hierarchy allows for both fine granular control of interrupt sources and coarse granular control over all peripherals, while allowing the peripherals to have their own interrupt vector.

In practice, top level peripheral interrupts are only used to enable or disable interrupts for an entire peripheral. Second level interrupts originate from hardware sources, and therefore are the main focus of applications using interrupts.

12.1 Nested vectored interrupt controller (NVIC)

The ARM® Cortex-M3 Nested Vectored Interrupt Controller (NVIC) facilitates low-latency exception and interrupt handling. The NVIC and the processor core interface are closely coupled, which enables low-latency interrupt processing and efficient processing of late arriving interrupts. The NVIC also maintains knowledge of the stacked (nested) interrupts to enable tail-chaining of interrupts.

The ARM® Cortex-M3 NVIC contains 10 standard interrupts that are related to chip and CPU operation and management. In addition to the 10 standard interrupts, it contains 17 individually vectored peripheral interrupts specific to the STM32W108.

The NVIC defines a list of exceptions (see *Table 42*). These exceptions include not only traditional peripheral interrupts, but also more specialized events such as faults and CPU reset. In the ARM® Cortex-M3 NVIC, a CPU reset event is considered an exception of the highest priority, and the stack pointer is loaded from the first position in the NVIC exception table. The NVIC exception table defines all exceptions and their position, including peripheral interrupts. The position of each exception is important since it directly translates to the location of a 32-bit interrupt vector for each interrupt, and defines the hardware priority of exceptions. Each exception in the table is a 32-bit address that is loaded into the program counter when that exception occurs. Exceptions 0 (stack pointer) through 15 (SysTick) are part of the standard ARM® Cortex-M3 NVIC, while exceptions 16 (Timer 1) through 32 (Debug) are the peripheral interrupts specific to the STM32W108 peripherals.

Table 42. NVIC exception table

Exception	Position	Description
-	0	Stack top is loaded from first entry of vector table on reset.
Reset	1	Invoked on power up and warm reset. On first instruction, drops to lowest priority (Thread mode). Asynchronous.
NMI	2	Cannot be stopped or preempted by any exception but reset. Asynchronous.
Hard Fault	3	All classes of fault, when the fault cannot activate because of priority or the Configurable Fault handler has been disabled. Synchronous.

Table 42. NVIC exception table (continued)

Exception	Position	Description
Memory Fault	4	MPU mismatch, including access violation and no match. Synchronous.
Bus Fault	5	Pre-fetch, memory access, and other address/memory-related faults. Synchronous when precise and asynchronous when imprecise.
Usage Fault	6	Usage fault, such as 'undefined instruction executed' or 'illegal state transition attempt'. Synchronous.
-	7-10	Reserved, must be kept at reset value
SVCall	11	System service call with SVC instruction. Synchronous.
Debug Monitor	12	Debug monitor, when not halting. Synchronous, but only active when enabled. It does not activate if lower priority than the current activation.
-	13	Reserved, must be kept at reset value
PendSV	14	Pendable request for system service. Asynchronous and only pended by software.
SysTick	15	System tick timer has fired. Asynchronous.
Timer 1	16	Timer 1 peripheral interrupt.
Timer 2	17	Timer 2 peripheral interrupt.
Management	18	Management peripheral interrupt.
Baseband	19	Baseband peripheral interrupt.
Sleep Timer	20	Sleep Timer peripheral interrupt.
Serial Controller 1	21	Serial Controller 1 peripheral interrupt.
Serial Controller 2	22	Serial Controller 2 peripheral interrupt.
Security	23	Security peripheral interrupt.
MAC Timer	24	MAC Timer peripheral interrupt.
MAC Transmit	25	MAC Transmit peripheral interrupt.
MAC Receive	26	MAC Receive peripheral interrupt.
ADC	27	ADC peripheral interrupt.
IRQA	28	IRQA peripheral interrupt.
IRQB	29	IRQB peripheral interrupt.
IRQC	30	IRQC peripheral interrupt.
IRQD	31	IRQD peripheral interrupt.
Debug	32	Debug peripheral interrupt.

The NVIC also contains a software-configurable interrupt prioritization mechanism. The Reset, NMI, and Hard Fault exceptions, in that order, are always the highest priority, and are not software-configurable. All other exceptions can be assigned a 5-bit priority number, with low values representing higher priority. If any exceptions have the same software-configurable priority, then the NVIC uses the hardware-defined priority. The hardware-defined priority number is the same as the position of the exception in the exception table. For example, if IRQA and IRQB both fire at the same time and have the same software-defined priority, the NVIC handles IRQA, with priority number 28, first because it has a higher hardware priority than IRQB with priority number 29.

For further information on the NVIC and Cortex-M3 exceptions, refer to the ARM® Cortex-M3 Technical Reference Manual and the ARM ARMv7-M Architecture Reference Manual.

12.2 Management interrupt registers

12.2.1 Management interrupt source register (MGMT_ISR)

Address offset: 0x4000 A018
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														HSE COMPH LIF	HSE COMPL LIF
														rw	rw

Bits 31:2	Reserved, must be kept at reset value
Bit 1	HSECOMPHLIF: OSC24M_HI interrupt
Bit 0	HSECOMPLLIF: OSC24M_LO interrupt

12.2.2 Management interrupt mask register (MGMT_IER)

Address offset: 0x4000 A058
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														HSE COMPH LIE	HSE COMPL LIE
														rw	rw

Bits 31:2	Reserved, must be kept at reset value
Bit 1	HSECOMPHLIE: OSC24M_HI mask
Bit 0	HSECOMPLLIE: OSC24M_LO mask

12.2.3 Management interrupt (MGMT) register map

Table 43 gives the ADC register map and reset values.

Table 43. MGMT register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA018	MGMT_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSECOMPHLIF	HSECOMPLLIF
	Reset value																															0	0
0xA01C-0xA054	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA058	MGMT_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSECOMPHLIE	HSECOMPLLIE
	Reset value																															0	0

Refer to *Figure 3: STM32W108 memory mapping* for the register boundary addresses.

13 Debug support

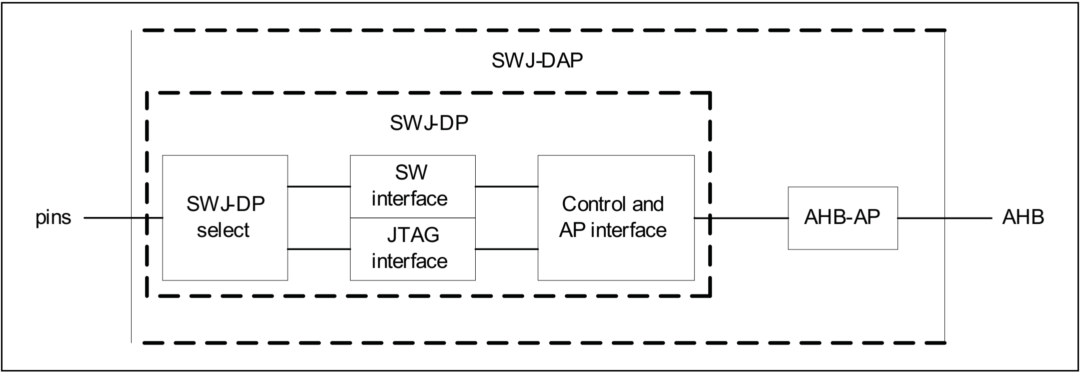
The STM32W108 includes a standard Serial Wire and JTAG (SWJ) Interface. The SWJ is the primary debug and programming interface of the STM32W108. The SWJ gives debug tools access to the internal buses of the STM32W108, and allows for non-intrusive memory and register access as well as CPU halt-step style debugging. Therefore, any design implementing the STM32W108 should make the SWJ signals readily available.

Serial Wire is an ARM® standard, bi-directional, two-wire protocol designed to replace JTAG, and provides all the normal JTAG debug and test functionality. JTAG is a standard five-wire protocol providing debug and test functionality. In addition, the two Serial Wire signals (SWDIO and SWCLK) are overlaid on two of the JTAG signals (JTMS and JTCK). This keeps the design compact and allows debug tools to switch between Serial Wire and JTAG as needed, without changing pin connections.

While Serial Wire and JTAG offer the same debug and test functionality, ST recommends Serial Wire. Serial Wire uses only two pins instead of five, and offers a simple communication protocol, high performance data rates, low power, built-in error detection, and protection from glitches.

The ARM® CoreSight Debug Access Port (DAP) comprises the Serial Wire and JTAG Interface (SWJ).The DAP includes two primary components: a debug port (the SWJ-DP) and an access port (the AHB-AP). The SWJ-DP provides external debug access, while the AHB-AP provides internal bus access. An external debug tool connected to the STM32W108's debug pins communicates with the SWJ-DP. The SWJ-DP then communicates with the AHB-AP. Finally, the AHB-AP communicates on the internal bus.

Figure 50. SWJ block diagram



Serial Wire and JTAG share five pins:

- JRST
- JTDO
- JTDI
- SWDIO/JTMS
- SWCLK/JTCK

Since these pins can be repurposed, refer to *Section 3: Pinout and pin description on page 19* and *Section 8: General-purpose input/output on page 89* for complete pin descriptions and configurations.

13.1 STM32W108 JTAG TAP connection

The STM32W108 MCU integrates two serially-connected JTAG TAPs in the following order; the TMC TAP dedicated for Test (IR is 4-bit wide) and the Cortex™-M3 TAP (IR is 4-bit wide).

To access the TAP of the Cortex-M3 for debug purposes:

1. First, it is necessary to shift the BYPASS instruction of the TMC TAP.
2. Then, for each IR shift, the scan chain contains 8 bits (= 4 + 4) and the unused TAP instruction must be shifted in using the BYPASS instruction.
3. For each data shift, the unused TAP, which is in BYPASS mode, adds 1 extra data bit in the data scan chain.

Note: ***Important****: Once Serial-Wire is selected using the dedicated ARM JTAG sequence, the TMC TAP is automatically disabled (JTMS forced high).*

14 Electrical characteristics

14.1 Parameter conditions

Unless otherwise specified, all voltages are referenced to V_{SS}.

14.1.1 Minimum and maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at T_A = 25 °C and T_A = $T_A$max (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean±3Σ).

14.1.2 Typical values

Unless otherwise specified, typical data are based on T_A = 25 °C, V_{DD} = 3.3 V (for the 2 V ≤ V_{DD} ≤ 3.6 V voltage range). They are given only as design guidelines and are not tested.

Typical ADC accuracy values are determined by characterization of a batch of samples from a standard diffusion lot over the full temperature range, where 95% of the devices have an error less than or equal to the value indicated (mean±2Σ).

14.1.3 Typical curves

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

14.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 51*.

14.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 52*.

Figure 51. Pin loading conditions

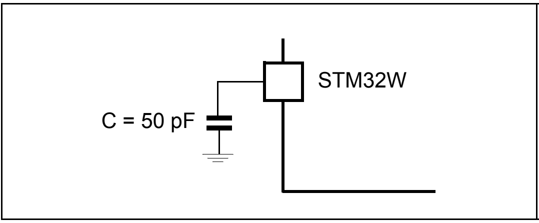
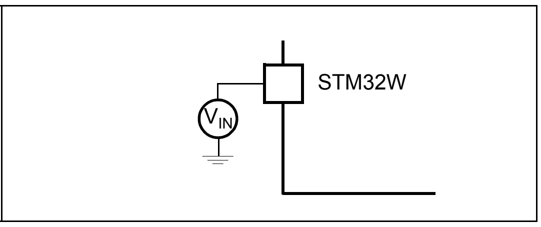


Figure 52. Pin input voltage



14.2 Absolute maximum ratings

Stresses above the absolute maximum ratings listed in *Table 44: Voltage characteristics*, *Table 45: Current characteristics*, and *Table 46: Thermal characteristics* may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 44. Voltage characteristics

Ratings	Min.	Max.	Unit
Regulator input voltage (VDD_PADS)	-0.3	+3.6	V
Analog, Memory and Core voltage (VDD_24MHZ, VDD_VCO, VDD_RF, VDD_IF, VDD_PADSA, VDD_MEM, VDD_PRE, VDD_SYNTH, VDD_CORE)	-0.3	+2.0	V
Voltage on RF_P,N; RF_TX_ALT_P,N	-0.3	+3.6	V
RF Input Power (for max level for correct packet reception see *Table 65: Receive characteristics*) RX signal into a lossless balun	–	+15	dBm
Voltage on any GPIO (PA[7:0], PB[7:0], PC[7:0]), SWCLK, NRST, VREG_OUT	-0.3	VDD_PADS +0.3	V
Voltage on BIAS_R, OSC_OUT, OSC_IN	-0.3	VDD_PADSA +0.3	V

Table 45. Current characteristics

Symbol	Ratings	Max.	Unit
I_{VDD}	Total current into V_{DD}/V_{DDA} power lines (source)	150	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink)	150	
I_{IO}	Output current sunk by any I/O and control pin	25	
	Output current source by any I/Os and control pin	– 25	
$I_{INJ(PIN)}$	Injected current on NRST pin	± 5	
	Injected current on HSE OSC_IN and LSE OSC_IN pins	± 5	
	Injected current on any other pin	± 5	
$\Sigma I_{INJ(PIN)}$	Total injected current (sum of all I/O and control pins)	± 25	

Table 46. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	–40 to +140	°C
T_J	Maximum junction temperature	150	°C

14.3 Operating conditions

14.3.1 General operating conditions

Table 47. General operating conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
–	Regulator input voltage (VDD_PADS)	2.1	–	3.6	V
–	Analog and memory input voltage (VDD_24MHZ, VDD_VCO, VDD_RF, VDD_IF, VDD_PADSA, VDD_MEM, VDD_PRE, and VDD_SYNTH)	1.7	1.8	1.9	V
–	Core input voltage (VDD_CORE)	1.18	1.25	1.32	V
T_{OPER}	Operating temperature range	-40	–	+85	°C

14.3.2 Operating conditions at power-up

Power-on resets (POR HV and POR LV)

The STM32W108 measures the voltage levels supplied to the three power domains. If a supply voltage drops below a low threshold, then a reset is applied. The reset is released if the supply voltage rises above a high threshold. There are three detection circuits for power on reset as follows:

- POR HV monitors the always on domain supply voltage. Thresholds are given in *Table 48*.
- POR LVcore monitors the core domain supply voltage. Thresholds are given in *Table 49*.
- POR LVmem monitors the memory supply voltage. Thresholds are given in *Table 50*.

Table 48. POR HV thresholds

Parameter	Test conditions	Min	Typ	Max	Unit
Always-on domain release		1.0	1.2	1.4	V
Always-on domain assert		0.5	0.6	0.7	V
Supply rise time	From 0.5 V to 1.7 V	–	–	250	µs

Table 49. POR LVcore thresholds

Parameter	Test conditions	Min	Typ	Max	Unit
1.25 V domain release		0.9	1.0	1.1	V
1.25 V domain assert		0.8	0.9	1.0	V

Table 50. POR LVmem thresholds

Parameter	Test conditions	Min	Typ	Max	Unit
1.8 V domain release		1.35	1.5	1.65	V
1.8 V domain assert		1.26	1.4	1.54	V

The POR LVcore and POR LVmem reset sources are merged to provide a single reset source, POR LV, to the Reset Generation module, since the detection of either event needs to reset the same system modules.

NRST pin

A single active low pin, NRST, is provided to reset the system. This pin has a Schmitt triggered input.

To afford good noise immunity and resistance to switch bounce, the pin is filtered with the Reset Filter module and generates the reset source RSTB to the Reset Generation module.

Table 51. Reset filter specification for RSTB

Parameter	Min	Typ	Max	Unit
Reset filter time constant	2.1	12.0	16.0	µs
Reset pulse width to guarantee a reset	26.0	–	–	µs
Reset pulse width guaranteed not to cause a reset	0	–	1.0	µs

14.3.3 Absolute maximum ratings (electrical sensitivity)

Based on three different tests (ESD, LU) using specific measurement methods, the device is stressed in order to determine its performance in terms of electrical sensitivity.

Electrostatic discharge (ESD)

Electrostatic discharges (a positive then a negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends on the number of supply pins in the device (3 parts × (n+1) supply pins). This test conforms to the JESD22-A114/C101 standard.

Table 52. ESD absolute maximum ratings

Symbol	Ratings	Conditions	Class	Maximum value(1)	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	T_A = +25 °C in compliance with JESD22-A114	2	±2000	V
$V_{ESD(CDM)}$	Electrostatic discharge voltage (charge device model) for non-RF pins	T_A = +25 °C in compliance with JESD22-A114	II	±400	V
	Electrostatic discharge voltage (charge device model) for RF pins			±225	
MSL	Moisture sensitivity level	–	–	MSL3	–

1. Based on characterization results, not tested in production.

Static latch-up

Two complementary static tests are required on six parts to assess the latch-up performance:

- A supply overvoltage is applied to each power supply pin
- A current injection is applied to each input, output and configurable I/O pin

These tests are compliant with EIA/JESD 78A IC latch-up standard.

Table 53. Electrical sensitivities

Symbol	Parameter	Conditions	Class
LU	Static latch-up class	T_A = +105 °C conforming to JESD78A	II level A

14.4 ADC characteristics

Table 54 describes the key ADC parameters measured at 25°C and VDD_PADS at 3.0 V, for a sampling clock of 1 MHz. HVSELP and HVSELN are programmed to 0 to disable the input buffer. The single-ended measurements were done at f_{input} = 7.7% $f_{Nyquist}$; 0 dBFS level (where full-scale is a 1.2 V p-p swing). The differential measurements were done at f_{input} = 7.7% $f_{Nyquist}$; -6 dBFS level (where full-scale is a 2.4 V p-p swing).

Table 54. ADC module key parameters for 1 MHz sampling(1)

Parameter	Performance							
SMP[2:0]	0	1	2	3	4	5	6	7
Conversion Time (µs)	32	64	128	256	512	1024	2048	4096
Nyquist Freq (kHz)	15.6	7.81	3.91	1.95	0.977	0.488	0.244	0.122
3 dB Cut-off (kHz)	9.43	4.71	2.36	1.18	0.589	0.295	0.147	0.0737
INL (codes peak)	0.083	0.092	0.163	0.306	0.624	1.229	2.451	4.926
INL (codes RMS)	0.047	0.051	0.093	0.176	0.362	0.719	1.435	2.848
DNL (codes peak)	0.028	0.035	0.038	0.044	0.074	0.113	0.184	0.333
DNL (codes RMS)	0.008	0.009	0.011	0.014	0.019	0.029	0.048	0.079
ENOB (from single-cycle test)	5.6	7.0	8.6	10.1	11.5	12.6	13.0	13.2
SNR (dB)								
Single-Ended	35	44	53	62	70	75	77	77
Differential	35	44	53	62	71	77	79	80
SINAD (dB)								
Single-Ended	35	44	53	61	67	69	70	70
Differential	35	44	53	62	70	75	76	76
SDFR (dB)								
Single-Ended	59	68	72	72	72	72	72	73
Differential	60	69	77	80	81	81	81	81
THD (dB)								
Single-Ended	-45	-54	-62	-67	-69	-69	-69	-69
Differential	-45	-54	-63	-71	-75	-76	-76	-76
ENOB (from SNR)								
Single-Ended	5.6	7.1	8.6	10.0	11.3	12.2	12.4	12.5
Differential	5.6	7.1	8.6	10.1	11.4	12.5	12.9	12.9
ENOB (from SINAD)								
Single-Ended	5.5	7.0	8.5	9.9	10.9	11.2	11.3	11.3
Differential	5.6	7.0	8.5	10.0	11.3	12.1	12.3	12.4
Equivalent ADC Bits	7 [15:9]	8 [15:8]	9 [15:7]	10 [15:6]	11 [15:5]	12 [15:4]	13 [15:3]	14 [15:2]

1. INL and DNL are referenced to a LSB of the Equivalent ADC Bits shown in the last row of *Table 54*. ENOB (effective number of bits) can be calculated from either SNR (signal to non-harmonic noise ratio) or SINAD (signal-to-noise and distortion ratio).

Table 55 describes the key ADC parameters measured at 25°C and VDD_PADS at 3.0 V, for a sampling rate of 6 MHz. HVSELP and HVSELN are programmed to 0 to disable the input buffer. The single-ended measurements were done at f_{input} = 7.7% $f_{Nyquist}$; 0 dBFS level (where full-scale is a 1.2 V p-p swing). The differential measurements were done at f_{input} = 7.7% $f_{Nyquist}$; -6 dBFS level (where full-scale is a 2.4 V p-p swing) and a common mode voltage of 0.6 V.

Table 55. ADC module key parameters for input buffer disabled and 6 MHz sampling(1)

Parameter	Performance							
SMP[2:0]	0	1	2	3	4	5	6	7
Conversion Time (µs)	5.33	10.7	21.3	42.7	85.3	171	341	683
Nyquist Freq (kHz)	93.8	46.9	23.4	11.7	5.86	2.93	1.47	0.732
3 dB Cut-off (kHz)	56.6	28.3	14.1	7.07	3.54	1.77	0.884	0.442
INL (codes peak)	0.084	0.084	0.15	0.274	0.518	1.057	2.106	4.174
INL (codes RMS)	0.046	0.044	0.076	0.147	0.292	0.58	1.14	2.352
DNL (codes peak)	0.026	0.023	0.044	0.052	0.096	0.119	0.196	0.371
DNL (codes RMS)	0.007	0.009	0.013	0.015	0.024	0.03	0.05	0.082
ENOB (from single-cycle test)	5.6	7.0	8.5	10.0	11.4	12.6	13.1	13.2
SNR (dB) Single-Ended Differential	35 35	44 44	53 53	62 62	70 71	75 77	76 79	77 80
SINAD (dB) Single-Ended Differential	35 35	44 44	53 53	62 62	68 70	71 75	71 77	71 77
SDFR (dB) Single-Ended Differential	60 60	68 69	75 77	75 80	75 80	75 80	75 80	75 80
THD (dB) Single-Ended Differential	-45 -45	-54 -54	-63 -63	-68 -71	-70 -76	-70 -77	-70 -78	-70 -78
ENOB (from SNR) Single-Ended Differential	5.6 5.6	7.1 7.1	8.6 8.6	10.0 10.1	11.4 11.5	12.1 12.5	12.4 12.9	12.5 13.0
ENOB (from SINAD) Single-Ended Differential	5.5 5.6	7.0 7.1	8.5 8.6	9.9 10.1	11.0 11.4	11.4 12.4	11.5 12.8	11.5 13.0
Equivalent ADC Bits	5 [15:11]	6 [15:10]	7 [15:9]	8 [15:8]	9 [15:7]	10 [15:6]	11 [15:5]	12 [15:4]

1. *INL and DNL are referenced to a LSB of the Equivalent ADC Bits shown in the last row of Table 55. ENOB (effective number of bits) can be calculated from either SNR (signal to non-harmonic noise ratio) or SINAD (signal-to-noise and distortion ratio).*

Table 56 describes the key ADC parameters measured at 25°C and VDD_PADS at 3.0 V, for a sampling rate of 6 MHz. HVSELP and HVSELN are programmed to 1 to enable the input buffer. The single-ended measurements were done at f_{input} = 7.7% $f_{Nyquist}$, level = 1.2 V p-p swing centered on 1.5 V. The differential measurements were done at f_{input} = 7.7% $f_{Nyquist}$, level = 1.2 V p-p swing and a common mode voltage of 1.5 V.

Table 56. ADC module key parameters for input buffer enabled and 6MHz sampling(1)

Parameter	Performance							
SMP[2:0]	0	1	2	3	4	5	6	7
Conversion Time (µs)	32	64	128	256	512	1024	2048	4096
Nyquist Freq (kHz)	93.8	46.9	23.4	11.7	5.86	2.93	1.47	0.732
3 dB Cut-off (kHz)	56.6	28.3	14.1	7.07	3.54	1.77k	0.884	0.442
INL (codes peak)	0.055	0.032	0.038	0.07	0.123	0.261	0.522	1.028
INL (codes RMS)	0.028	0.017	0.02	0.04	0.077	0.167	0.326	0.65
DNL (codes peak)	0.028	0.017	0.02	0.04	0.077	0.167	0.326	0.65
DNL (codes RMS)	0.01	0.006	0.006	0.007	0.008	0.013	0.023	0.038
ENOB (from single-cycle test)	3.6	5.0	6.6	8.1	9.5	10.7	11.3	11.6
SNR (dB) Single-Ended Differential	 23 23	 32 32	 41 41	 50 50	 59 59	 65 66	 67 69	 68 71
SINAD (dB) Single-Ended Differential	 23 23	 32 32	 41 41	 50 50	 58 59	 64 66	 66 69	 66 71
SDFR (dB) Single-Ended Differential	 48 48	 56 57	 65 65	 72 74	 72 82	 72 88	 73 88	 73 88
THD (dB) Single-Ended Differential	 -33 -33	 -42 -42	 -51 -51	 -59 -60	 -66 -69	 -68 -76	 -68 -80	 -68 -82
ENOB (from SNR) Single-Ended Differential	 3.6 3.6	 5.1 5.1	 6.6 6.6	 8.1 8.1	 9.5 9.5	 10.5 10.7	 10.9 11.3	 11 11.5
ENOB (from SINAD) Single-Ended Differential	 3.6 3.6	 5.0 5.1	 6.5 6.6	 8.0 8.0	 9.4 9.5	 10.3 10.6	 10.7 11.3	 10.7 11.4
Equivalent ADC Bits	7 [15:9]	8 [15:8]	9 [15:7]	10 [15:6]	11 [15:5]	12 [15:4]	13 [15:3]	14 [15:2]

1. *INL and DNL are referenced to a LSB of the Equivalent ADC Bits shown in the last row of Table 56. ENOB (effective number of bits) can be calculated from either SNR (signal to non-harmonic noise ratio) or SINAD (signal-to-noise and distortion ratio).*

Table 57 lists other specifications for the ADC not covered in *Table 54*, *Table 55*, and *Table 56*.

Table 57. ADC characteristics

Parameter	Min.	Typ.	Max.	Units
VREF	1.17	1.2	1.35	V
VREF output current	–	–	1	mA
VREF load capacitance	–	–	10	nF
External VREF voltage range	1.1	1.2	1.3	V
External VREF input impedance	1	–	–	MΩ
Minimum input voltage Input buffer disabled Input buffer enabled	 0 0.1	 – –	 – –	V
Maximum input voltage Input buffer disabled Input buffer enabled	 – –	 – –	 VREF VDD_PADS - 0.1	V
Single-ended signal range Input buffer disabled Input buffer enabled	 0 0.1	 – –	 VREF VDD_PADS – 0.1	V
Differential signal range Input buffer disabled Input buffer enabled	 -VREF -VDD_PADS + 0.1	 – –	 +VREF +VDD_PADS - 0.1	V
Common mode range Input buffer disabled Input buffer enabled	 0	 VDD_PADS/2	 VREF	V
Input referred ADC offset	-10	–	10	mV
Input Impedance 1 MHz sample clock 6 MHz sample clock Not sampling	 1 0.5 10	 – – –	 – – –	MΩ

Note: *The signal-ended ADC measurements are limited in their range and only guaranteed for accuracy within the limits shown in this table. The ADC's internal design allows for measurements outside of this range (±200 mV) when the input buffer is disabled, but the accuracy of such measurements is not guaranteed. The maximum input voltage is of more interest to the differential sampling where a differential measurement might be small, but a common mode can push the actual input voltage on one of the signals towards the upper voltage limit.*

14.5 Clock frequencies

14.5.1 High frequency internal clock characteristics

Table 58. 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		%

14.5.2 High frequency external clock characteristics

Table 59. 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

14.5.3 Low frequency internal clock characteristics

Table 60. Low-frequency RC oscillator characteristics

Parameter	Test conditions	Min.	Typ.	Max.	Unit
Nominal Frequency	After trimming	9	10	11	kHz
Analog trim step size	–	–	1	–	kHz
Supply dependence	For a voltage drop from 3.6 V to 3.1 V or 2.6 V to 2.1 V (without re-calibration)	–	–	1	%
Frequency dependence	Frequency variation with temperature for a change from -40 oC to +85oC (without re-calibration)	–	2	–	%

14.5.4 Low frequency external clock characteristics

Table 61. Low-frequency crystal oscillator characteristics

Parameter	Test conditions	Min.	Typ.	Max.	Unit
Frequency	–	–	32.768	–	kHz
Accuracy	Initial, temperature, and ageing	-100	–	+100	ppm
Load cap xin	–	–	27	–	pF
Load cap xout	–	–	18	–	pF
Crystal ESR	–	–	–	100	kΩ
Start-up time	–	–	–	2	s
Current consumption	At 25°C, VDD_PADS = 3.0 V	–	–	0.5	µA

14.6 DC electrical characteristics

Table 62. DC electrical characteristics

Parameter	Conditions	Min.	Typ.	Max.	Unit
Regulator input voltage (VDD_PADS)		2.1	–	3.6	V
Power supply range (VDD_MEM)	Regulator output or external input	1.7	1.8	1.9	V
Power supply range (VDD_CORE)	Regulator output	1.18	1.25	1.32	V
Deep sleep current					
Quiescent current, internal RC oscillator disabled	-40°C, VDD_PADS = 3.6 V	–	0.4	–	µA
	+25°C, VDD_PADS = 3.6 V	–	0.4	–	µA
	+85°C, VDD_PADS = 3.6 V	–	0.7	–	µA
Quiescent current, including internal RC oscillator	-40°C, VDD_PADS = 3.6 V	–	0.7	–	µA
	+25°C, VDD_PADS = 3.6 V	–	0.7	–	µA
	+85°C, VDD_PADS = 3.6 V	–	1.1	–	µA
Quiescent current, including 32.768 kHz oscillator	-40°C, VDD_PADS = 3.6 V	–	0.8	–	µA
	+25°C, VDD_PADS = 3.6 V	–	1.0	–	µA
	+85°C, VDD_PADS = 3.6 V	–	1.5	–	µA
Quiescent current, including internal RC oscillator and 32.768 kHz oscillator	-40°C, VDD_PADS = 3.6 V	–	1.1	–	µA
	+25°C, VDD_PADS = 3.6 V	–	1.3	–	µA
	+85°C, VDD_PADS = 3.6 V	–	1.8	–	µA
Simulated deep sleep (debug mode) current	With no debugger activity	–	300	–	µA
Reset current					
Quiescent current, NRST asserted	Typ at 25°C/3 V Max at 85°C/3.6 V	–	1.2	2.0	mA
Processor and peripheral currents					
ARM® Cortex-M3, RAM, and Flash memory	25 °C, 1.8 V memory and 1.25 V core ARM® Cortex-M3 running at 12 MHz from crystal oscillator Radio and all peripherals off	–	6.0	–	mA
ARM® Cortex-M3, RAM, and Flash memory	25 °C, 1.8 V memory and 1.25 V core ARM® Cortex-M3 running at 24 MHz from crystal oscillator Radio and all peripherals off	–	7.5	–	mA

Table 62. DC electrical characteristics (continued)

Parameter	Conditions	Min.	Typ.	Max.	Unit
ARM® Cortex-M3, RAM, and Flash memory sleep current	25 °C, 1.8 V memory and 1.25 V core ARM® Cortex-M3 clocked at 12 MHz from the crystal oscillator Radio and all peripherals off	–	3.0	–	mA
ARM® Cortex-M3, RAM, and Flash memory sleep current	25 °C, 1.8 V memory and 1.25 V core ARM® Cortex-M3 clocked at 6 MHz from the high frequency RC oscillator Radio and all peripherals off	–	2.0	–	mA
Serial controller current	For each controller at maximum data rate	–	0.2	–	mA
General purpose timer current	For each timer at maximum clock rate	–	0.25	–	mA
General purpose ADC current	At maximum sample rate, DMA enabled	–	1.1	–	mA
Rx current					
Radio receiver, MAC, and baseband	ARM® Cortex-M3 sleeping	–	22.0	–	mA
Total RX current (= $I_{\text{Radio receiver, MAC and baseband, CPU}}$ + $I_{\text{RAM, and Flash memory}}$)	VDD_PADS = 3.0 V, 25 °C, ARM® Cortex-M3 running at 12 MHz	–	25.0	–	mA
	VDD_PADS = 3.0 V, 25 °C, ARM® Cortex-M3 running at 24 MHz	–	26.5	–	mA
Boost mode total RX current (= $I_{\text{Radio receiver, MAC and baseband, CPU}}$+ $I_{\text{RAM, and Flash memory}}$)	VDD_PADS = 3.0 V, 25 °C, ARM® Cortex-M3 running at 12 MHz	–	27.0	–	mA
	VDD_PADS = 3.0 V, 2 5°C, ARM® Cortex-M3 running at 24 MHz	–	28.5	–	mA
Tx current					
Radio transmitter, MAC, and baseband	25 °C and 1.8 V core; max. power out (+3 dBm typical) ARM® Cortex-M3 sleeping	–	26.0	–	mA

Table 62. DC electrical characteristics (continued)

Parameter	Conditions	Min.	Typ.	Max.	Unit
Total Tx current (= $I_{\text{Radio transmitter, MAC and baseband, CPU}} + I_{\text{RAM, and Flash memory}}$)	VDD_PADS = 3.0 V, 25 °C; maximum power setting (+7 dBm); ARM® Cortex-M3 running at 12 MHz	–	42.0	–	mA
	VDD_PADS = 3.0 V, 25 °C; +3 dBm power setting; ARM® Cortex-M3 running at 12 MHz	–	29.5	–	mA
	VDD_PADS = 3.0 V, 25 °C; 0dBm power setting; ARM® Cortex-M3 running at 12 MHz	–	27.0	–	mA
	VDD_PADS = 3.0 V, 25 °C; minimum power setting; ARM® Cortex-M3 running at 12 MHz	–	21.0	–	mA
	VDD_PADS = 3.0 V, 25 °C; maximum power setting (+7 dBm); ARM® Cortex-M3 running at 24 MHz	–	43.5	–	mA
	VDD_PADS = 3.0 V, 25 °C; +3 dBm power setting; ARM® Cortex-M3 running at 24 MHz	–	31.0	–	mA
	VDD_PADS = 3.0 V, 25 °C; 0dBm power setting; ARM® Cortex-M3 running at 24 MHz	–	28.5	–	mA
	VDD_PADS = 3.0 V, 25 °C; minimum power setting; ARM® Cortex-M3 running at 24 MHz	–	22.5	–	mA

Figure 53 shows the variation of current in Transmit mode (with the ARM$^{®}$ Cortex-M3 running at 12 MHz).

Figure 53. Transmit power consumption

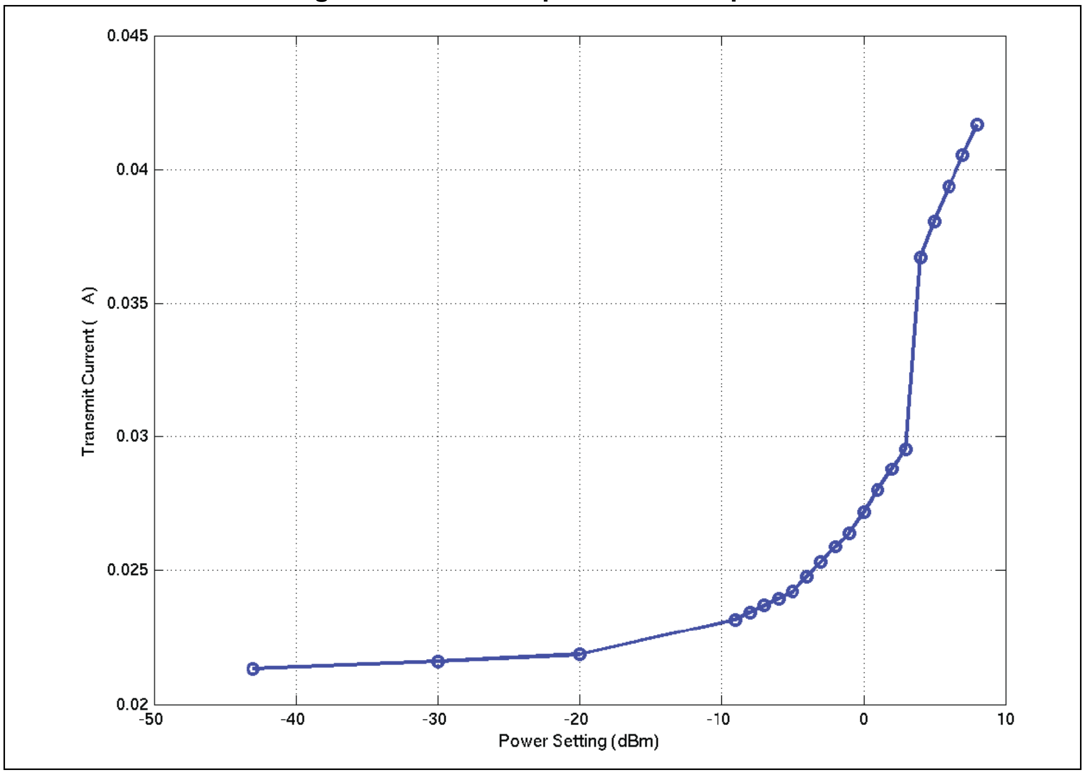
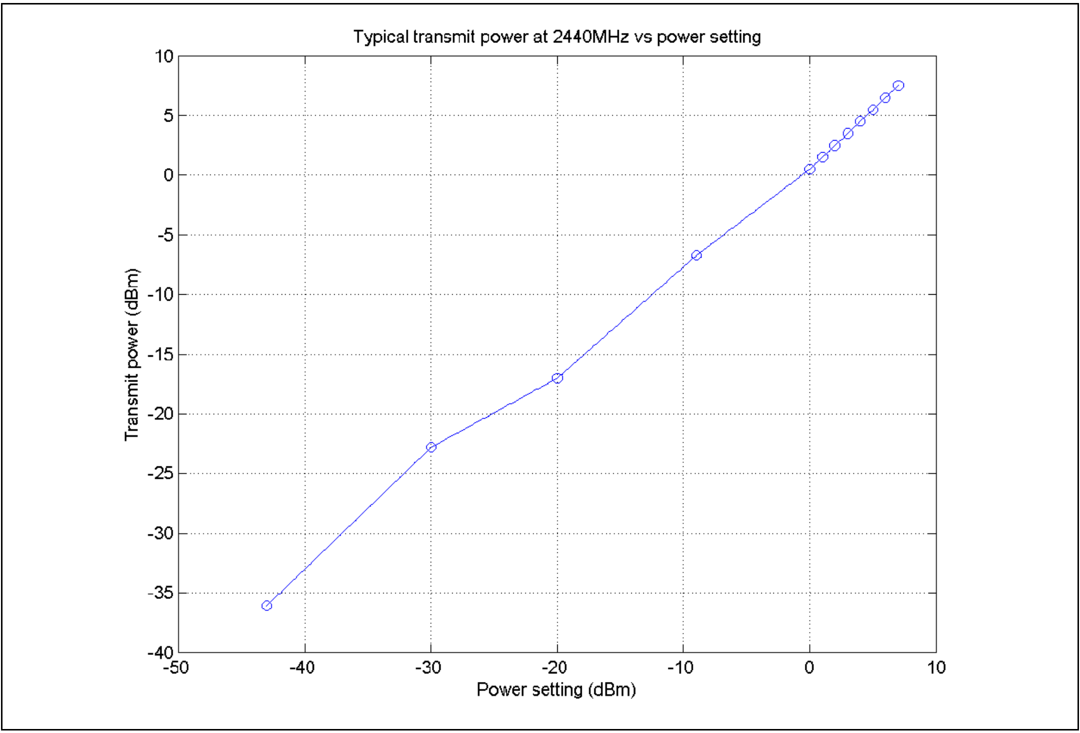


Figure 54 shows typical output power against power setting on the ST reference design.

Figure 54. Transmit output power



14.7 Digital I/O specifications

Table 63 lists the digital I/O specifications for the STM32W. The digital I/O power (named VDD_PADS) comes from three dedicated pins (Pins 23, 28 and 37). The voltage applied to these pins sets the I/O voltage.

Table 63. Digital I/O characteristics

Parameter	Conditions	Min.	Typ.	Max.	Unit
Voltage supply (Regulator Input)	VDD_PADS	2.1	–	3.6	V
Low Schmitt switching threshold	V_{SWIL} Schmitt input threshold going from high to low	0.42 x VDD_PAD S	–	0.50 x VDD_PAD S	V
High Schmitt switching threshold	V_{SWIH} Schmitt input threshold going from low to high	0.62 x VDD_PAD S	–	0.80 x VDD_PAD S	V
Input current for logic 0	I_{IL}	–	–	-0.5	µA
Input current for logic 1	I_{IH}	–	–	+0.5	µA
Input pull-up resistor value	R_{IPU}	24	29	34	kΩ
Input pull-down resistor value	R_{IPD}	24	29	34	kΩ
Output voltage for logic 0	V_{OL} (I_{OL} = 4 mA for standard pads, 8 mA for high current pads)	0	–	0.18 x VDD_PAD S	V
Output voltage for logic 1	V_{OH} (I_{OH} = 4 mA for standard pads, 8 mA for high current pads)	0.82 x VDD_PAD S	–	VDD_PAD S	V
Output source current (standard current pad)	I_{OHS}	–	–	4	mA
Output sink current (standard current pad)	I_{OLS}	–	–	4	mA
Output source current high current pad: PA6, PA7, PB6, PB7, PC0	I_{OHH}	–	–	8	mA
Output sink current high current pad: PA6, PA7, PB6, PB7, PC0	I_{OLH}	–	–	8	mA
Total output current (for I/O Pads)	I_{OH} + I_{OL}	–	–	40	mA
Input voltage threshold for OSC32_OUT		0.2 x VDD_PAD S	–	0.8 x VDD_PAD S	V
Input voltage threshold for OSC_OUT		0.2 x VDD_PAD SA	–	0.8 x VDD_PAD SA	V

14.8 Non-RF system electrical characteristics

Table 64 lists the non-RF system level characteristics for the STM32W.

Table 64. Non-RF system electrical characteristics

Parameter	Conditions	Min.	Typ.	Max.	Unit
System wakeup time from deep sleep	From wakeup event to first ARM® Cortex-M3 instruction running from 6MHz internal RC clock Includes supply ramp time and oscillator startup time	–	110	–	µs
Shutdown time going into deep sleep	From last ARM® Cortex-M3 instruction to deep sleep mode	–	5	–	µs

14.9 RF electrical characteristics

14.9.1 Receive

Table 65 lists the key parameters of the integrated IEEE 802.15.4 receiver on the STM32W.

Note: Receive measurements were collected with ST's STM32W Ceramic Balun Reference Design (Version A0) at 2440 MHz. The Typical number indicates one standard deviation above the mean, measured at room temperature (25°C). The Min and Max numbers were measured over process corners at room temperature

Table 65. Receive characteristics

Parameter	Conditions	Min.	Typ.	Max.	Unit
Frequency range		2400	–	2500	MHz
Sensitivity (boost mode)	1% PER, 20 byte packet defined by IEEE 802.15.4-2003	–	-102	-96	dBm
Sensitivity	1% PER, 20 byte packet defined by IEEE 802.15.4-2003	–	-100	-94	dBm
High-side adjacent channel rejection	IEEE 802.15.4 signal at -82 dBm	–	35	–	dB
Low-side adjacent channel rejection	IEEE 802.15.4 signal at -82 dBm	–	35	–	dB
2nd high-side adjacent channel rejection	IEEE 802.15.4 signal at -82 dBm	–	46	–	dB
2nd low-side adjacent channel rejection	IEEE 802.15.4 signal at -82 dBm	–	46	–	dB
Channel rejection for all other channels	IEEE 802.15.4 signal at -82 dBm	–	40	–	dB
802.11g rejection centered at +12 MHz or -13 MHz	IEEE 802.15.4 signal at -82 dBm	–	36	–	dB

Table 65. Receive characteristics (continued)

Parameter	Conditions	Min.	Typ.	Max.	Unit
Maximum input signal level for correct operation		0	–	–	dBm
Co-channel rejection	IEEE 802.15.4 signal at -82 dBm	–	-6	–	dBc
Relative frequency error (2x40 ppm required by IEEE 802.15.4)		-120	–	+120	ppm
Relative timing error (2x40 ppm required by IEEE 802.15.4)		-120	–	+120	ppm
Linear RSSI range	As defined by IEEE 802.15.4	40	–	–	dB
RSSI Range		-90	–	-40	dBm

14.9.2 Transmit

Table 66 lists the key parameters of the integrated IEEE 802.15.4 transmitter on the STM32W.

Note: Transmit measurements were collected with ST's STM32W Ceramic Balun Reference Design (Version A0) at 2440 MHz. The Typical number indicates one standard deviation above the mean, measured at room temperature (25°C). The Min and Max numbers were measured over process corners at room temperature

Table 66. Transmit characteristics

Parameter	Conditions	Min.	Typ.	Max.	Unit
Maximum output power (boost mode)	At highest power setting	–	8	–	dBm
Maximum output power	At highest power setting	1	5	–	dBm
Minimum output power	At lowest power setting	–	-55	–	dBm
Error vector magnitude	As defined by IEEE 802.15.4, which sets a 35% maximum	–	5	15	%
Carrier frequency error		-40	–	+40	ppm
PSD mask relative	3.5 MHz away	-20	–	–	dB
PSD mask absolute	3.5 MHz away	-30	–	–	dBm

14.9.3 Synthesizer

Table 67 lists the key parameters of the integrated synthesizer on the STM32W.

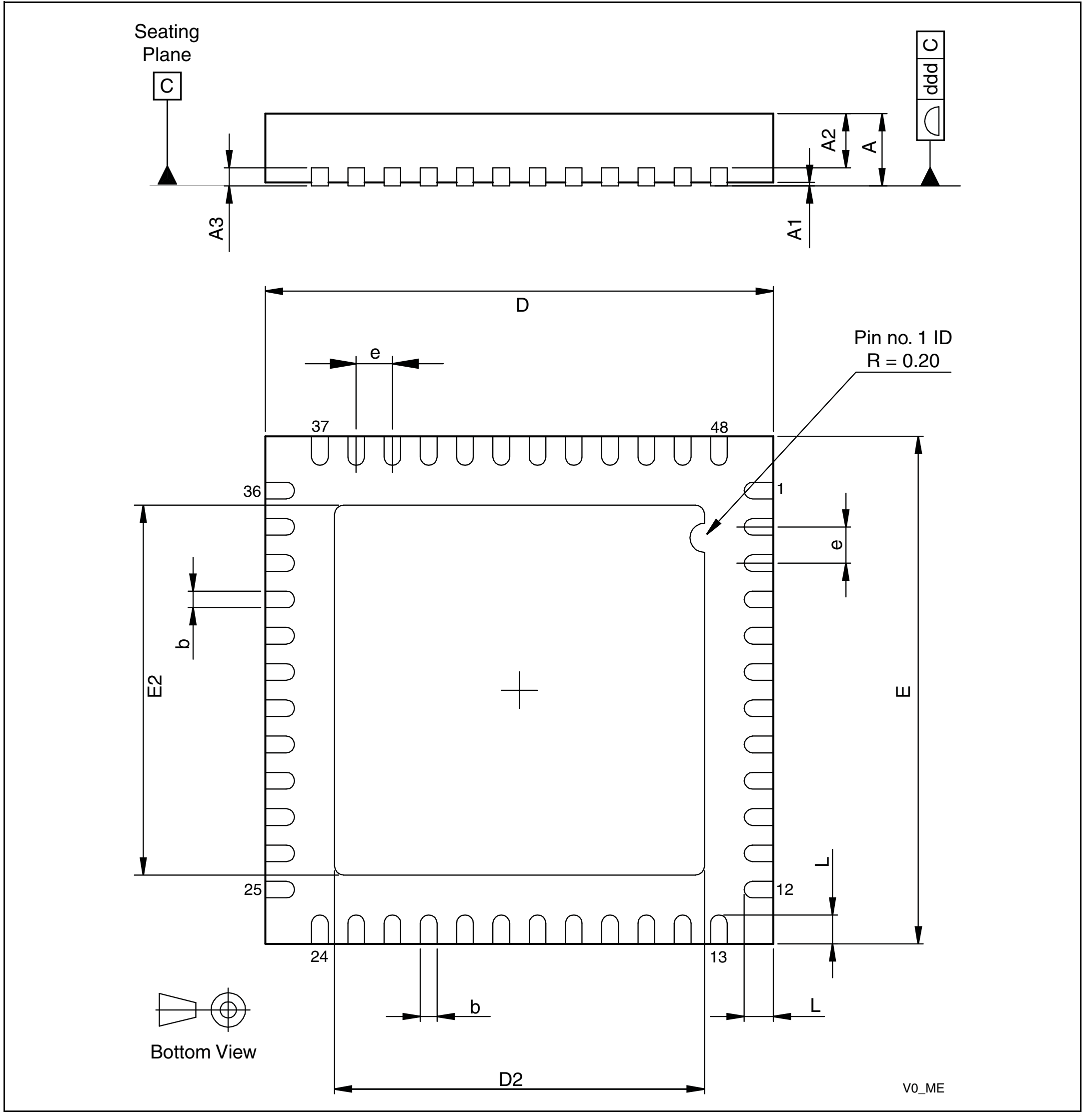
Table 67. Synthesizer characteristics

Parameter	Conditions	Min.	Typ.	Max.	Unit
Frequency range		2400	–	2500	MHz
Frequency resolution		–	11.7	–	kHz
Lock time	From off, with correct VCO DAC setting	–	–	100	µs
Relock time	Channel change or RX/TX turnaround (IEEE 802.15.4 defines 192 µs turnaround time)	–	–	100	µs
Phase noise at 100 kHz offset		–	-71	–	dBc/Hz
Phase noise at 1 MHz offset		–	-91	–	dBc/Hz
Phase noise at 4 MHz offset		–	-103	–	dBc/Hz
Phase noise at 10 MHz offset		–	-111	–	dBc/Hz

15 Package characteristics

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: *www.st.com*. ECOPACK® is an ST trademark.

Figure 55. VFQFPN48 7x7mm package outline



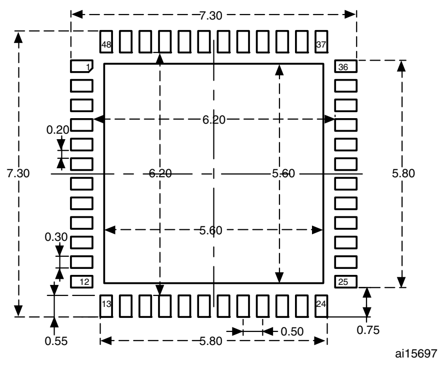
1. Drawing is not to scale.

Table 68. VFQFPN48 7x7mm package mechanical data

Symbol	Millimeters			Inches[1]		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	0.800	0.900	1.000	0.0315	0.0354	0.0394
A1		0.020	0.050		0.0008	0.0020
A2		0.650	1.000		0.0256	0.0394
A3		0.250			0.0098	
b	0.180	0.230	0.300	0.0071	0.0091	0.0118
D	6.850	7.000	7.150	0.2697	0.2756	0.2815
D2	2.250	4.700	5.250	0.0886	0.1850	0.2067
E	6.850	7.000	7.150	0.2697	0.2756	0.2815
E2	2.250	4.700	5.250	0.0886	0.1850	0.2067
e	0.450	0.500	0.550	0.0177	0.0197	0.0217
L	0.300	0.400	0.500	0.0118	0.0157	0.0197
ddd			0.080			0.0031

1. Values in inches are converted from mm and rounded to 4 decimal digits.

Figure 56. VFQFPN48 7x7mm recommended footprint (dimensions in mm)



1. Drawing is not to scale.

16 Ordering information scheme

Example: STM32 W 108 C 8 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

C = 48 pins

Code size

8 = 64 Kbytes 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)]

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.

For a list of available options (speed, package, etc.) or for further information on any aspect of this device, please contact your nearest ST sales office.

17 Revision history

Table 69. Document revision history

Date	Revision	Changes
05-May-2011	1	Initial revision.
28-Jul-2011	2	Modified first page (Exceptional RF performance) Modified *Section 9.12.8: Serial controller transmit DMA end address channel B register (SCx_DMATXENDADDBR) on page 145* In *Section 12.3: Nested vectored interrupt controller (NVIC) interrupts on page 210*, address offsets replaced with addresses One temperature range added
3-Sep-2012	3	**TEERMINOLOGY-RELATED CHANGES** Renamed following registers (and their constituent bits): INT_SCxFLAG to SCx_ISR (updated bit descriptions) INT_SCxCFG to SCxIER (updated bit descriptions) SCx_INTMODE to SCx_ICR (updated bit descriptions) SCx_MODE to SCx_CR (updated bit descriptions) SCx_DATA to SCx_DR (updated bit descriptions) SCx_RATELIN to SCx_CRR1 (updated bit descriptions) SCx_RATEEXP to SCx_CRR2 (updated bit descriptions) SCx_SPISTAT to SCx_SPISR (updated bit descriptions) SCx_SPICFG to SCx_SPICR (updated bit descriptions) SCx_TWISTAT to SCx_I2CSR (updated bit descriptions) SCx_TWICTRL1 to SCx_I2CCR1 (updated bit descriptions) SCx_TWICTRL2 to SCx_I2CCR2 (updated bit descriptions) SC1_UARTSTAT to SC1_UARTSR (updated bit descriptions) SC1_UARTCFG to SC1_UARTCR (updated bit descriptions) SC1_UARTPER to SC1_UARTBRR1 SC1_UARTFRAC to SC1_UARTBRR2 (updated bit descriptions) SCx_DMASTAT to SCx_DMASR (updated bit descriptions) SCx_DMACTRL to SCx_DMACR (updated bit descriptions) SCx_TXBEGA to SCx_DMATXBEGADDAR (updated bit descriptions) SCx_TXBEGB to SCx_DMATXBEGADDBR (updated bit descriptions) SCx_TXENDA to SCx_DMATXENDADDAR (updated bit descriptions) SCx_TXENDB to SCx_DMATXENDADDBR (updated bit descriptions) SCx_TXCNT to SCX_DMATXCNTR (updated bit descriptions) SCx_RXBEGA to SCx_DMARXBEGADDAR (updated bit descriptions) SCx_RXBEGAB to SCx_DMARXBEGADDBR (updated bit descriptions) SCx_RXENDA to SCx_DMARXENDADDAR (updated bit descriptions) SCx_RXENDB to SCx_DMARXENDADDBR (updated bit descriptions) SCx_RXCNTA to SCx_DMARXCNTAR (updated bit descriptions) SCx_RXCNTB to SCx_DMARXCNTBR (updated bit descriptions)

Table 69. Document revision history (continued)

Date	Revision	Changes
3-Sep-2012	3 (continued)	SCx_RXCNTSAVED to SCx_DMARXCNTSAVEDR (updated bit descriptions) SCx_RXERRA to SCx_DMARXERRAR (updated bit descriptions) SCx_RXERRB to SCx_DMARXERRBR (updated bit descriptions) SLEEPTMR_CFG to SLPTMR_CR (updated bit descriptions) SLEEPTMR_CNTH to SLPTMR_CNTH SLEEPTMR_CNTL to SLPTMR_CNTL SLEEPTMR_CMPAH to SLPTMR_CMPAH (updated bit descriptions) SLEEPTMR_CMPAL to SLPTMR_CMPAL (updated bit descriptions) SLEEPTMR_CMPBH to SLPTMR_CMPBH (updated bit descriptions) SLEEPTMR_CMPBL to SLPTMR_CMPBL (updated bit descriptions) NT_SLEEPTMRFLAG to SLPTMR_ISR (updated bit description) INT_SLEEPTMRCFG to SLPTMR_IER (updated bit description) SLEEPTMR_CLKEN to CLK_SLEEPCR (updated bit descriptions) INT_TIMxCFG to TIMx_IER INT_TIMxFLAG to TIMx_ISR INT_TIMxMISS to TIMx_MISSR GPIO_PxCFGL to GPIOx_CRL (updated bit descriptions) GPIO_PxCFGH to GPIOx_CRH (updated bit descriptions) GPIO_PxIN to GPIOx_IDR (updated bit descriptions) GPIO_PxOUT to GPIOx_ODR (updated bit descriptions) GPIO_PxCLR to GPIOx_BRR (updated bit descriptions) GPIO_PxSET to GPIOx_BSR (updated bit descriptions) GPIO_PxWAKE to PWR_WAKEPxR (updated bit descriptions) GPIO_WAKEFILT to PWR_WAKEFILTR (updated bit descriptions) GPIO_IRQxSEL to EXTIx_CR (updated bit descriptions) GPIO_INTCFGx to EXTIx_TSR (updated bit descriptions) INT_GPIOFLAG to EXTI_PR (updated bit descriptions) GPIO_DBGCFG to GPIO_DBGCR (updated bit descriptions) GPIO_DBGSTAT to GPIO_DBGSR (updated bit descriptions) ADC_CFG to ADC_CR (updated bit descriptions) ADC_OFFSET to ADC_OFFSETR ADC_GAIN to ADC_GAINR ADC_DMACFG to ADC_DMACR (updated bit descriptions) ADC_DMASTAT to ADC_DMASR ADC_DMABEG to ADC_DMAMSAR (updated bit descriptions) ADC_DMASIZE to ADC_DMANDTR (updated bit descriptions) ADC_DMACUR to ADC_DMAMNAR (updated bit descriptions) ADC_DMACNT to ADC_DMACNDTR (updated bit descriptions) INT_ADCFLAG to ADC_ISR INT_ADCCFG to ADC_IER RESET_EVENT to RST_SR OSC24M_CTRL to CLK_HSECR2 (updated bit descriptions) CPU_CLK_SEL to CLK_CPU_CR WDOG_CFG to WDG_CR WDOG_CTRL to WDG_KR WDOG_RESTART to WDG_KICKSR (added bit description)

Table 69. Document revision history (continued)

Date	Revision	Changes
3-Sep-2012	3 (continued)	Renamed the constituent bits of the following registers throughout document: TIMx_CR1 TIMx_CR2 TIMx_SMCR TIMx_EGR TIMx_CCMR1 TIMx_CCMR2 TIMx_CCER TIMx_CNT TIMx_PSC TIMx_ARR TIMx_CCR1 TIMx_CCR2 TIMx_CCR3 TIMx_CCR4 TIM1_OR TIM2_OR Renamed the following terms: OSC32A to OSC32_OUT, OSC32B to OSC32_IN, OSC32K to LSE OSC, CLK32K to LSE, OSCRC to LSI10K, CLK1K to LSI1K, OSCA to OSC_OUT, OSCB to OSC_IN, OSC24M to HSE OSC, OSCHF to HSI, 24 MHz XTAL to 24 MHz HSE OSC, 12 MHz RC to 12 MHz HSI RC, 10 kHz RC to 10 kHz LSI RC, 32 kHz XTAL to 32 kHz HSE OSC. Updated terminology of *Figure 4: System module block diagram* and *Figure 5: Clocks block diagram*. *Section 6.4.5*: Replaced: ENABLE bit with EN, SLEEP_CONFIG with SLPTMR_CR, COMP_A_H with CMPAH, COMP_A_L with CMPAL, COMP_B_H with CMPBH, COMP_B_L with CMPBL. *Figure 8*: Replaced SCx_I2CSTAT, SCx_I2CCTRL1, and SCx_I2CCTRL2 with SCx_I2CSR, SCx_I2CCR1, and SCx_I2CCR2 respectively. *Section 10*: replaced TMRx with TIMx, TIM_USR with USR, CCR1L with CCR1[15:0], CCR1H with CCR1[31:16], CNT with TIMx_CNT, CNT with CNT[15:0], CCR1 with TIMx_CCR1, TIMx_CC1R with TIMx_CCR1, TIM_CCRx with TIMx_CCRy, TIM_CMS with CMS[1:0], OCxREF with OCyREF, TIM_CCyIF with CCyIF, TIM_SMCR with TIMx_SMCR, TIM_OC* with OCxy, and TIM_IC* with ICxy. **GENERIC CHANGES** Updated “reserved” bit descriptions throughout document. Removed all references to NVIC. Removed all references to the INT_CFGSET register, INT_CFGCLR register, and all registers originally in *Section 12: Interrupts*. Removed all non-defined asterisks. **SPECIFIC CHANGES** *Section 1.2.1*: updated selectable voltage ranges of ADC. Added *Section 4.1: Memory organization and memory map*. *Section 4.2: Flash memory*: replaced 1000 with 10k write/erase cycles.

Table 69. Document revision history (continued)

Date	Revision	Changes
3-Sep-2012	3 (continued)	Added *Section 4.3.3* and *Section 4.3.4*. *Section 6.2.4*: added *Reset (RST) register map*. *Section 6.3: Clocks*: added note beneath bullets points (before *Figure 5*). *Section 6.3.6*: added *Low-speed internal 10 KHz clock (LSI10K) control register (CLK_LSI10KCR)*, *Low-speed internal 1 KHz clock control register (CLK_LSI1KCR)*, *High-speed external clock control register 1 (CLK_HSECR1)*, *High-speed internal clock control register (CLK_HSICR)*, *High-speed external clock comparator register (CLK_HSECOMPR)*, *Clock period control register (CLK_PERIODCR)*, *Clock period status register (CLK_PERIODSR)*, *Clock dither control register (CLK_DITHERCR)*, *High-speed external clock control register 2 (CLK_HSECR2)*, *CPU clock control register (CLK_CPUCR)*, *Clock switching (CLK) register map*. Added *Section 6.4.5*. *Section 6.4.5*: Added *Sleep timer force interrupt register (SLPTMR_IFR)*, *MACTimer counter register (MACTMR_CNTR)*, *MACTimer counter register (MACTMR_CR)* and *MAC timer (MACTMR)/Watchdog (WDG)/Sleeptimer(SLPTMR) register map*. *Section 6.5.1*: updated last bullet point. *Section 6.5.2*: updated bullet points. Added *Section 6.5.5: Power management registers*. *Section 8.1.1*: updated paragraph concerning configuration registers. Added *Table 8.5.10: PC TRACE or debug select register (GPIO_PCTRACECR)*. Added *Table 8.5.13: General-purpose input/output (GPIO) register map*. *Section 9.4.1*: removed “see table 86”. *Table 21: SPI master mode formats*, *Table 23: SPI slave mode formats*, and *Figure 26: I2C master frame segments*: updated heading lines and removed footnote. Removed *Section 9.10: SPI slave mode registers.* Added *Section 9.12.17: Serial interface (SC1/SC2) register map*. *Section 10.1.1*: updated *Prescaler* section. *Section 10.3.8* and *Section 10.3.9*: separated register descriptions into “output compare mode” and “input capture mode”. Added *Section 10.3.20: General-purpose timers 1 and 2 (TIM1/TIM2) register map*. *Section 11.1.8*: updated information regarding the VDD_PADSA regulator. *Section 11.3.9* and *Section 11.3.11*: Bit ranges changed to [12:0]. *Section 11.3.10*: Bit ranges changed to [13:1]. Added *Section 11.3.12: Analog-to-digital converter (ADC) register map*. *Section 12: Interrupts*: updated section and removed register descriptions. Added *Section 12.2: Management interrupt registers*.

Table 69. Document revision history (continued)

Date	Revision	Changes
3-Sep-2012	3 (continued)	Added *Figure 56: VFQFPN48 7x7mm recommended footprint (dimensions in mm)*
23-Sep-2013	4	Replaced GPIOx_WAKER by PWR_WAKEPxR, GPIO_WAKEFR by PWR_WAKEFILTR, GPIO_WAKE by GPIO_SEL, WAKE_SEL by ETXTIx_CR and SEL_GPIO by GPIO_SEL in *Section 8: General-purpose input/output*. Changed VREF max. value from 1.23 to 1.35 in *Table 57: ADC characteristics*.

Index