Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321r6ta

Table 4. Hardware register map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0040h	Reserved Area (1 byte)				
0041h	TIMER B	TBCR2	Timer B Control Register 2	00h	R/W
0042h		TBCR1	Timer B Control Register 1	00h	R/W
0043h		TBCSR	Timer B Control/Status Register	xxxx x0xx b	R/W
0044h		TBIC1HR	Timer B Input Capture 1 High Register	xxh	Read Only
0045h		TBIC1LR	Timer B Input Capture 1 Low Register	xxh	Read Only
0046h		TBOC1HR	Timer B Output Compare 1 High Register	80h	R/W
0047h		TBOC1LR	Timer B Output Compare 1 Low Register	00h	R/W
0048h		TBCHR	Timer B Counter High Register	FFh	Read Only
0049h		TBCLR	Timer B Counter Low Register	FCh	Read Only
004Ah		TBACHR	Timer B Alternate Counter High Register	FFh	Read Only
004Bh		TBACLR	Timer B Alternate Counter Low Register	FCh	Read Only
004Ch		TBIC2HR	Timer B Input Capture 2 High Register	xxh	Read Only
004Dh		TBIC2LR	Timer B Input Capture 2 Low Register	xxh	Read Only
004Eh		TBOC2HR	Timer B Output Compare 2 High Register	80h	R/W
004Fh		TBOC2LR	Timer B Output Compare 2 Low Register	00h	R/W
0050h	SCI	SCISR	SCI Status Register	C0h	Read Only
0051h		SCIDR	SCI Data Register	xxh	R/W
0052h		SCIBRR	SCI Baud Rate Register	00h	R/W
0053h		SCICR1	SCI Control Register 1	x000 0000b	R/W
0054h		SCICR2	SCI Control Register 2	00h	R/W
0055h		SCIERPR	SCI Extended Receive Prescaler Register	00h	R/W
0056h			Reserved area	---	
0057h		SCIETPR	SCI Extended Transmit Prescaler Register	00h	R/W
0058h 006Fh	Reserved Area (24 Bytes)				
0070h	ADC	ADCCSR	Control/Status Register	00h	R/W
0071h		ADCDRH	Data High Register	00h	Read Only
0072h		ADCDRL	Data Low Register	00h	Read Only
0073h	PWM ART	PWMDCR3	PWM AR Timer Duty Cycle Register 3	00h	R/W
0074h		PWMDCR2	PWM AR Timer Duty Cycle Register 2	00h	R/W
0075h		PWMDCR1	PWM AR Timer Duty Cycle Register 1	00h	R/W
0076h		PWMDCR0	PWM AR Timer Duty Cycle Register 0	00h	R/W
0077h		PWMCR	PWM AR Timer Control Register	00h	R/W
0078h		ARTCSR	Auto-Reload Timer Control/Status Register	00h	R/W
0079h		ARTCAR	Auto-Reload Timer Counter Access Register	00h	R/W
007Ah		ARTARR	Auto-Reload Timer Auto-Reload Register	00h	R/W
007Bh		ARTICCSR	AR Timer Input Capture Control/Status Reg.	00h	R/W
007Ch		ARTICR1	AR Timer Input Capture Register 1	00h	Read Only
007Dh		ARTICR2	AR Timer Input Capture Register 1	00h	Read Only
007Eh 007Fh	Reserved Area (2 bytes)				

1. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.
2. The bits associated with unavailable pins must always keep their reset value.

Note: Legend: x = undefined, R/W = read/write

4 Flash program memory

4.1 Introduction

The ST7 dual voltage High Density Flash (HDFlash) is a non-volatile memory that can be electrically erased as a single block or by individual sectors and programmed on a byte-by-byte basis using an external V_{PP} supply.

The HDFlash devices can be programmed and erased off-board (plugged in a programming tool) or on-board using ICP (in-circuit programming) or IAP (in-application programming).

The array matrix organization allows each sector to be erased and reprogrammed without affecting other sectors.

4.2 Main features

- 3 Flash programming modes:
  - Insertion in a programming tool. In this mode, all sectors including option bytes can be programmed or erased.
  - ICP (in-circuit programming). In this mode, all sectors including option bytes can be programmed or erased without removing the device from the application board.
  - IAP (in-application programming). In this mode, all sectors except Sector 0 can be programmed or erased without removing the device from the application board and while the application is running.
- ICT (in-circuit testing) for downloading and executing user application test patterns in RAM
- Readout protection
- Register Access Security System (RASS) to prevent accidental programming or erasing

4.3 Structure

The Flash memory is organized in sectors and can be used for both code and data storage.

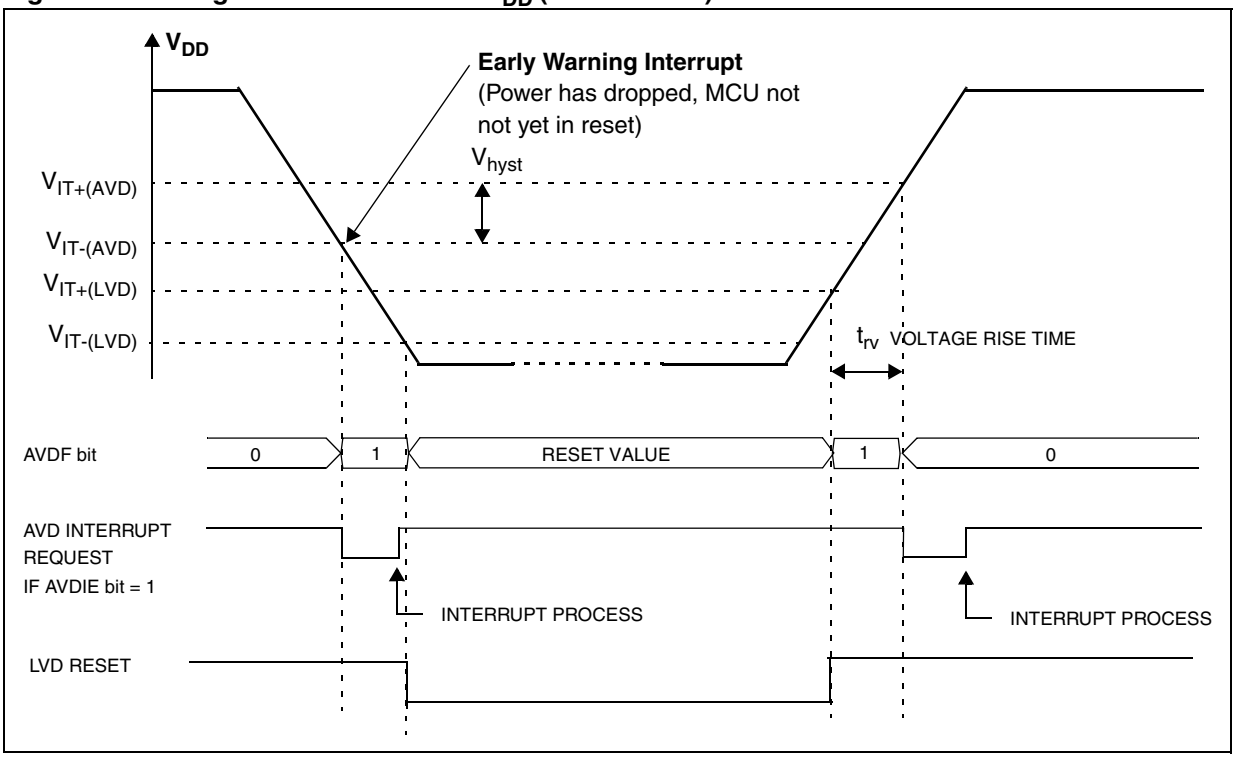
Depending on the overall Flash memory size in the microcontroller device, there are up to three user sectors (see *Table 5*). Each of these sectors can be erased independently to avoid unnecessary erasing of the whole Flash memory when only a partial erasing is required.

Table 5. Sectors available in Flash devices

Flash size (bytes)	Available sectors
4K	Sector 0
8K	Sectors 0, 1
> 8K	Sectors 0, 1, 2

The first two sectors have a fixed size of 4 Kbytes (see *Figure 5*). They are mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

Figure 15. Using the AVD to monitor V_{DD} (AVDS bit = 0)



Monitoring a voltage on the EVD pin

This mode is selected by setting the AVDS bit in the SICSR register.

The AVD circuitry can generate an interrupt when the AVDIE bit of the SICSR register is set. This interrupt is generated on the rising and falling edges of the comparator output. This means it is generated when either one of these two events occur:

- V_{EVD} rises up to $V_{IT+(EVD)}$
- V_{EVD} falls down to $V_{IT-(EVD)}$

The EVD function is illustrated in *Figure 16*.

For more details, refer to *Section 19: Electrical characteristics*.

Figure 29. I/O port general block diagram

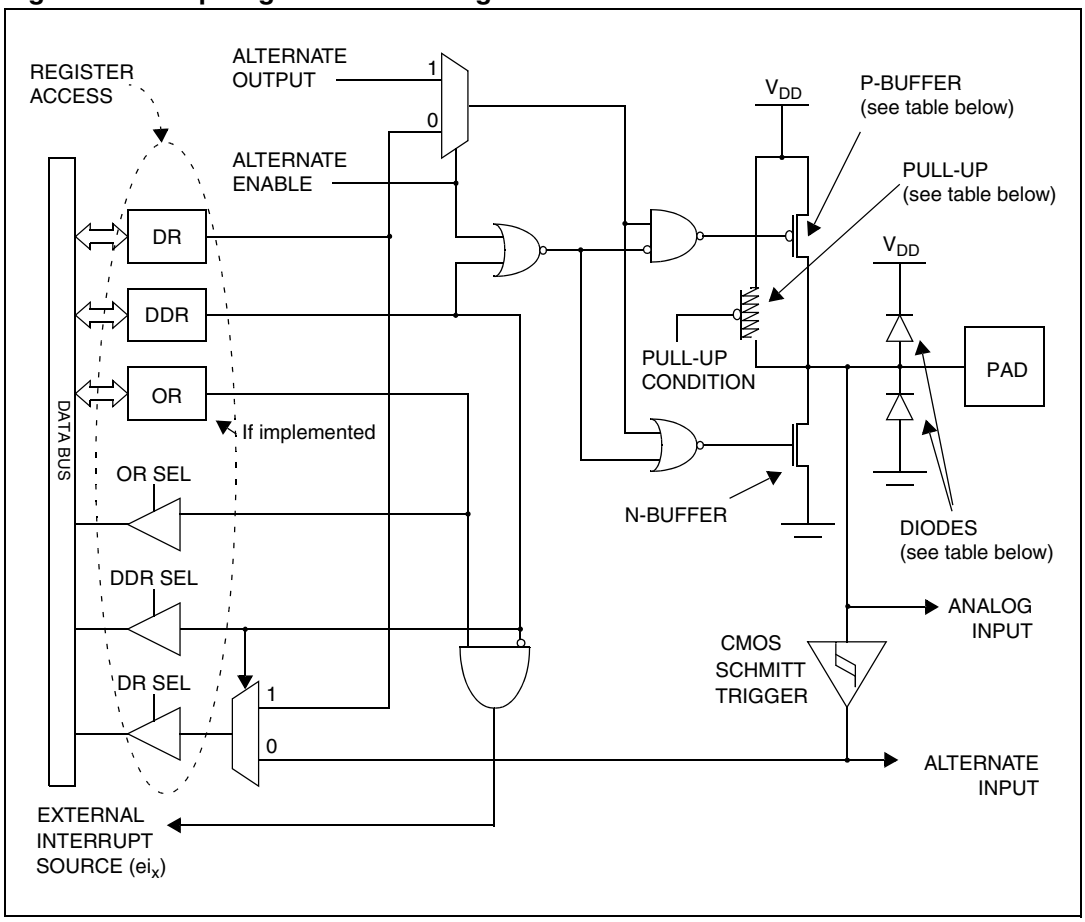


Table 29. I/O port mode options

Configuration mode		Pull-up	P-buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without Interrupt	Off	Off	On	On
	Pull-up with/without Interrupt	On			
Output	Push-pull	Off	On		
	Open-drain (logic level)		Off		
	True open-drain	NI	NI	NI(1)	

1. The diode to V_{DD} is not implemented in the true open-drain pads. A local protection between the pad and V_{SS} is implemented to protect the device against positive stress.

Legend:

Off - Implemented not activated

On - Implemented and activated

NI - Not implemented

Table 31. I/O port configuration (continued)

Port	Pin name	Input (DDR = 0)		Output (DDR = 1)	
		OR = 0	OR = 1	OR = 0	OR = 1
Port B	PB7, PB3	floating	floating interrupt	open-drain	push-pull
	PB6:5, PB4, PB2:0	floating	pull-up interrupt	open-drain	push-pull
Port C	PC7:0	floating	pull-up	open-drain	push-pull
Port D	PD7:0	floating	pull-up	open-drain	push-pull
Port E	PE7:3, PE1:0	floating	pull-up	open drain	push-pull
	PE2 (Flash devices)	pull-up input only			
	PE2 (ROM devices)	floating		open drain	push-pull
Port F	PF7:3	floating	pull-up	open-drain	push-pull
	PF2	floating	floating interrupt	open-drain	push-pull
	PF1:0	floating	pull-up interrupt	open-drain	push-pull

9.4 Low power modes

Table 32. Effect of low power modes on I/O ports

Mode	Effect
Wait	No effect on I/O ports. External interrupts cause the device to exit from Wait mode.
Halt	No effect on I/O ports. External interrupts cause the device to exit from Halt mode.

9.5 Interrupts

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers and the interrupt mask in the CC register is not active (RIM instruction).

Table 33. I/O port interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
External interrupt on selected external event	-	DDRx, ORx	Yes	Yes

Table 34. I/O port register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Reset value of all I/O port registers		0	0	0	0	0	0	0	0
0000h	PADR	MSB							LSB
0001h	PADDR								
0002h	PAOR								

Table 34. I/O port register map and reset values (continued)

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Reset value of all I/O port registers		0	0	0	0	0	0	0	0
0003h	PBDR	MSB							LSB
0004h	PBDDR								
0005h	PBOR								
0006h	PCDR	MSB							LSB
0007h	PCDDR								
0008h	PCOR								
0009h	PDDR	MSB							LSB
000Ah	PDDDR								
000Bh	PDOR								
000Ch	PEDR	MSB							LSB
000Dh	PEDDR								
000Eh	PEOR								
000Fh	PFDR	MSB							LSB
0010h	PFDDR								
0011h	PFOR								

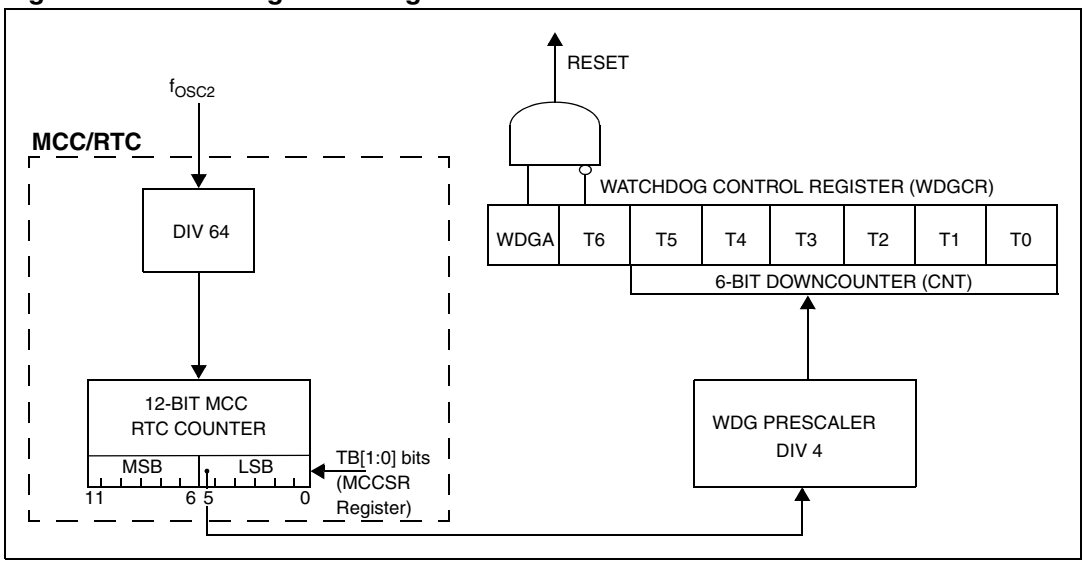
Related documentation

SPI Communication between ST7 and EEPROM (AN 970)

S/W implementation of I2C bus master (AN1045)

Software LCD driver (AN1048)

Figure 31. Watchdog block diagram



10.4 How to program the watchdog timeout

Figure 32 shows the linear relationship between the 6-bit value to be loaded in the Watchdog Counter (CNT) and the resulting timeout duration in milliseconds. This can be used for a quick calculation without taking the timing variations into account. If more precision is needed, use the formulae in *Figure 33*.

Caution: When writing to the WDGCR register, always write 1 in the T6 bit to avoid generating an immediate reset.

Figure 32. Approximate timeout duration

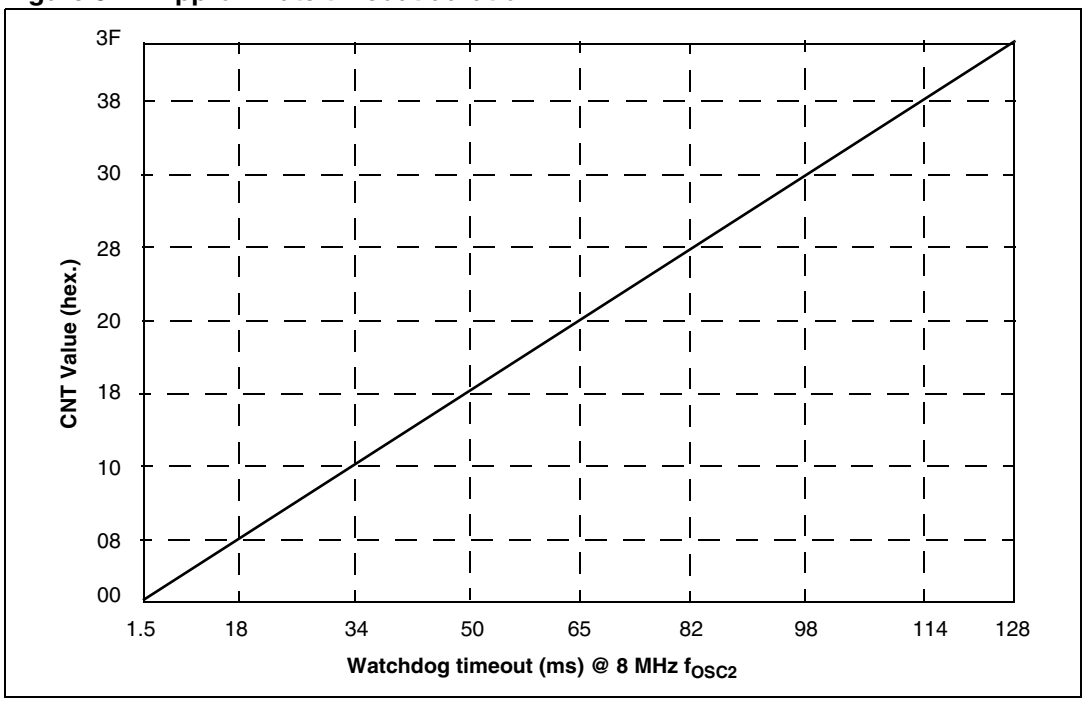
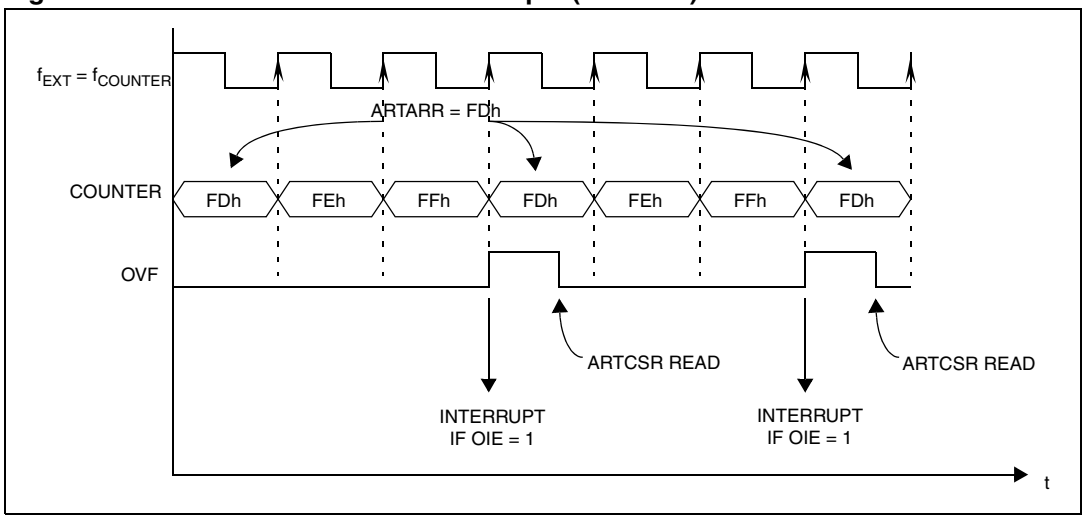


Figure 39. External event detector example (3 counts)



12.2.8 Input capture function

This mode allows the measurement of external signal pulse widths through ARTICRx registers.

Each input capture can generate an interrupt independently on a selected input signal transition. This event is flagged by a set of the corresponding CFx bits of the Input Capture Control/Status register (ARTICCSR).

These input capture interrupts are enabled through the CIEx bits of the ARTICCSR register.

The active transition (falling or rising edge) is software programmable through the CSx bits of the ARTICCSR register.

The read only input capture registers (ARTICRx) are used to latch the auto-reload counter value when a transition is detected on the ARTICx pin (CFx bit set in ARTICCSR register). After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

Note: After a capture detection, data transfer in the ARTICRx register is inhibited until it is read (clearing the CFx bit).

The timer interrupt remains pending while the CFx flag is set when the interrupt is enabled (CIEx bit set). This means that the ARTICRx register has to be read at each capture event to clear the CFx flag.

The timing resolution is given by auto-reload counter cycle time ($1/f_{COUNTER}$).

Note: During Halt mode, if both the input capture and the external clock are enabled, the ARTICRx register value is not guaranteed if the input capture pin and the external clock change simultaneously.

12.3.7 Input capture registers (ARTICRx)

ARTICRx Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
IC[7:0]							

RO

Table 54. ARTICRx register description

Bit	Name	Function
7:0	IC[7:0]	*Input Capture Data* These read only bits are set and cleared by hardware. An ARTICRx register contains the 8-bit auto-reload counter value transferred by the input capture channel x event.

Table 55. PWM auto-reload timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0073h	PWMDCR3 Reset value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0074h	PWMDCR2 Reset value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0075h	PWMDCR1 Reset value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0076h	PWMDCR0 Reset value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0077h	PWMCR Reset value	OE3 0	OE2 0	OE1 0	OE0 0	OP3 0	OP2 0	OP1 0	OP0 0
0078h	ARTCSR Reset value	EXCL 0	CC2 0	CC1 0	CC0 0	TCE 0	FCRL 0	RIE 0	OVF 0
0079h	ARTCAR Reset value	CA7 0	CA6 0	CA5 0	CA4 0	CA3 0	CA2 0	CA1 0	CA0 0
007Ah	ARTARR Reset value	AR7 0	AR6 0	AR5 0	AR4 0	AR3 0	AR2 0	AR1 0	AR0 0
007Bh	ARTICCSR Reset value	0	0	CS2 0	CS1 0	CIE2 0	CIE1 0	CF2 0	CF1 0
007Ch	ARTICR1 Reset value	IC7 0	IC6 0	IC5 0	IC4 0	IC3 0	IC2 0	IC1 0	IC0 0
007Dh	ARTICR2 Reset value	IC7 0	IC6 0	IC5 0	IC4 0	IC3 0	IC2 0	IC1 0	IC0 0

13.3 Functional description

13.3.1 Counter

The main block of the Programmable Timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high and low.

Counter Register (CR)

- Counter High Register (CHR) is the most significant byte (MS Byte)
- Counter Low Register (CLR) is the least significant byte (LS Byte)

Alternate Counter Register (ACR)

- Alternate Counter High Register (ACHR) is the most significant byte (MS Byte)
- Alternate Counter Low Register (ACLR) is the least significant byte (LS Byte)

These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register (SR) (see note at the end of paragraph entitled *16-bit read sequence*).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.

Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in One Pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in *Table 61: Timer clock selection*. The value in the counter register repeats every 131072, 262144 or 524288 CPU clock cycles depending on the CC[1:0] bits.

The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

5 *The alternate inputs (ICAP1 and ICAP2) are always directly connected to the timer. So any transitions on these pins activates the input capture function.*

6 *Moreover if one of the ICAPi pins is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.*

7 *This can be avoided if the input capture function i is disabled by reading the ICiHR (see note 1).*

8 *The TOF bit can be used with interrupt generation in order to measure events that go beyond the timer range (FFFFh).*

Figure 46. Input capture block diagram

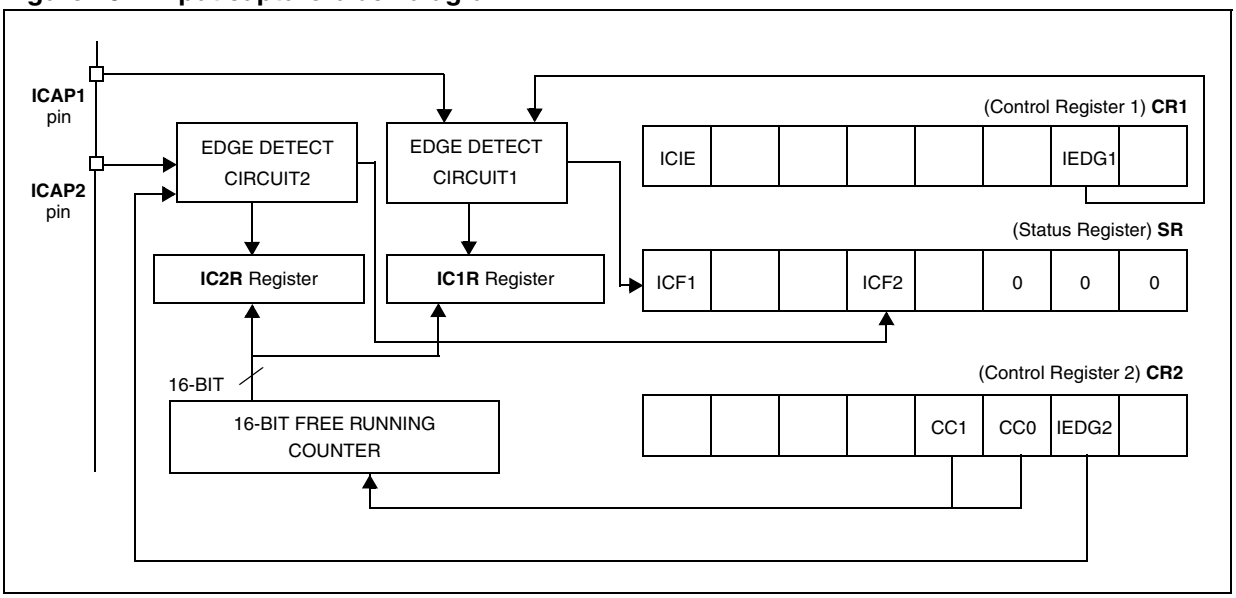
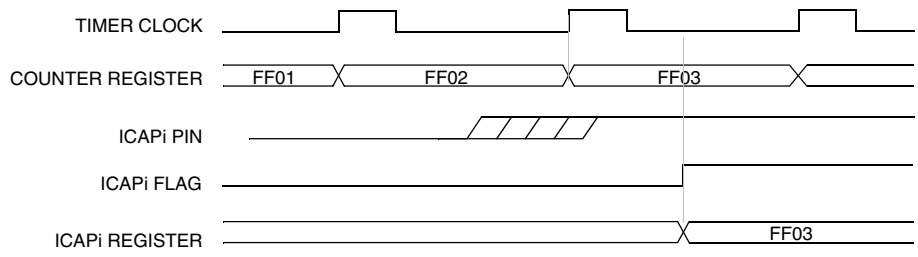
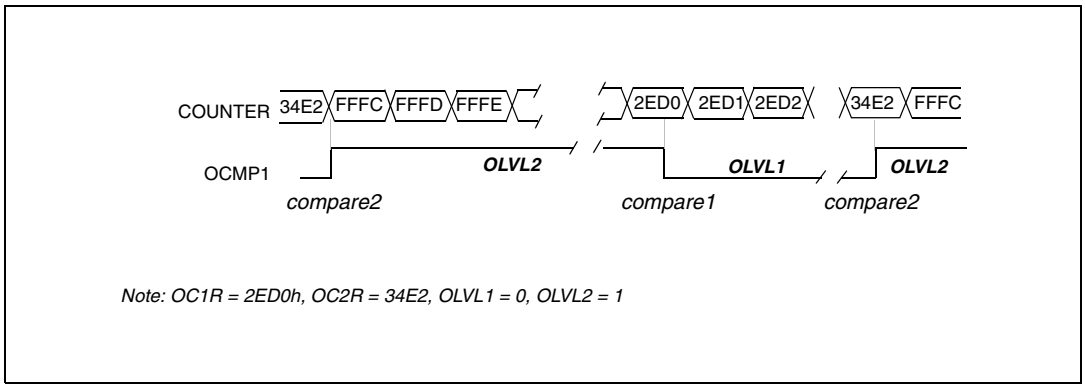


Figure 47. Input capture timing diagram



Note: The rising edge is the active edge.

Figure 53. Pulse width modulation mode timing example with 2 output compare functions



Note: On timers with only one Output Compare register, a fixed frequency PWM signal can be generated using the output compare and the counter overflow to define the pulse length.

13.3.7 Pulse width modulation mode

Pulse Width Modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so this functionality cannot be used when PWM mode is activated.

In PWM mode, double buffering is implemented on the output compare registers. Any new values written in the OC1R and OC2R registers are taken into account only at the end of the PWM period (OC2) to avoid spikes on the PWM output pin (OCMP1).

Procedure

To use pulse width modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal using the appropriate formula below according to the timer clock source used.
2. Load the OC1R register with the value corresponding to the period of the pulse if OLVL1 = 0 and OLVL2 = 1 using the appropriate formula below according to the timer clock source used.
3. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC1R register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC[1:0]) (see *Table 61: Timer clock selection*).

Table 60. CR2 register description (continued)

Bit	Name	Function
5	OPM	*One Pulse Mode* 0: One Pulse Mode is not active. 1: One Pulse Mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.
4	PWM	*Pulse Width Modulation* 0: PWM mode is not active. 1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.
3:2	CC[1:0]	*Clock Control* The timer clock mode depends on these bits (see *Table 61*).
1	IEDG2	*Input Edge 2* This bit determines which type of level transition on the ICAP2 pin will trigger the capture. 0: A falling edge triggers the capture. 1: A rising edge triggers the capture.
0	EXEDG	*External Clock Edge* This bit determines which type of level transition on the external clock pin EXTCLK will trigger the counter register. 0: A falling edge triggers the counter register. 1: A rising edge triggers the counter register.

Table 61. Timer clock selection

Timer clock	CC1	CC0
f_{CPU} / 4	0	0
f_{CPU} / 2	0	1
f_{CPU} / 8	1	0
External clock (where available)(1)	1	1

1. If the external clock pin is not available, programming the external clock configuration stops the counter.

13.7.3 Control/status register (CSR)

CSR Reset value: xxxx x0xx (xxh)

7	6	5	4	3	2	1	0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	Reserved	
RO	RO	RO	RO	RO	RW	-	

Table 62. CSR register description

Bit	Name	Function
7	ICF1	*Input Capture Flag 1* 0: No input capture (reset value) 1: An input capture has occurred on the ICAP1 pin or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.
6	OCF1	*Output Compare Flag 1* 0: No match (reset value) 1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.
5	TOF	*Timer Overflow Flag* 0: No timer overflow (reset value) 1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register. *Note: Reading or writing the ACLR register does not clear TOF.*
4	ICF2	*Input Capture Flag 2* 0: No input capture (reset value). 1: An input capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.
3	OCF2	*Output Compare Flag 2* 0: No match (reset value) 1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.
2	TIMD	*Timer disable* This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed, or the counter reset, while it is disabled. 0: Timer enabled 1: Timer prescaler, counter and outputs disabled
1:0	-	Reserved, must be kept cleared

13.7.4 Input capture 1 high register (IC1HR)

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).

IC1HR — Reset value: Undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

13.7.13 Alternate counter low register (ACLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

ACLR — Reset value: 1111 1100 (FCh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

13.7.14 Input capture 2 high register (IC2HR)

This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).

IC2HR — Reset value: Undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

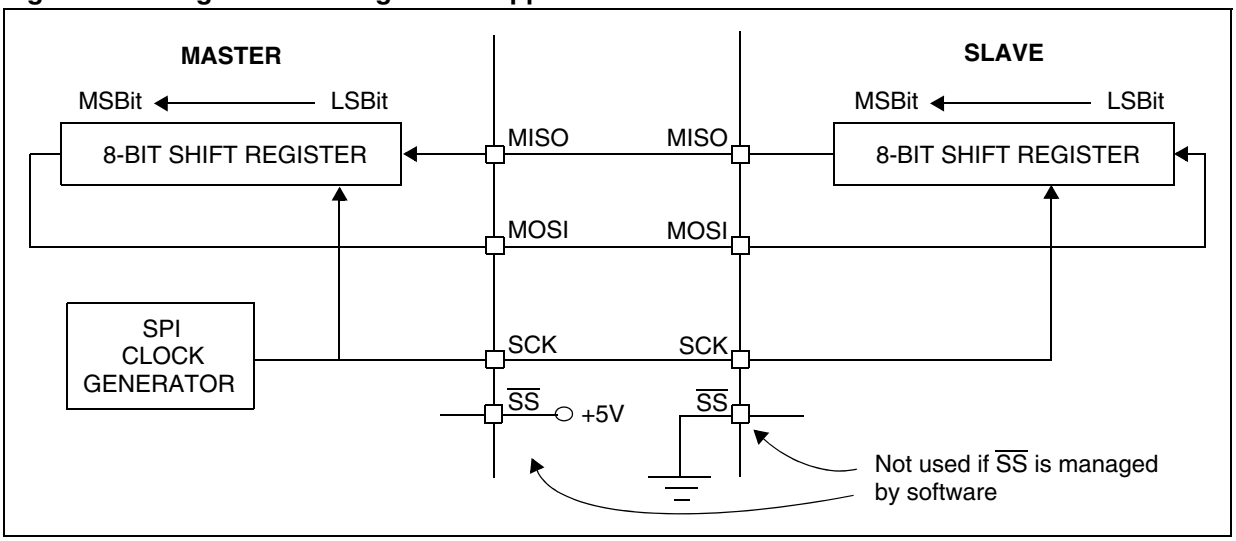
13.7.15 Input capture 2 low register (IC2LR)

This is an 8-bit read only register that contains the low part of the counter value (transferred by the Input Capture 2 event).

IC2LR — Reset value: Undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Figure 56. Single master/single slave application



14.3.2 Slave select management

As an alternative to using the $\overline{SS}$ pin to control the Slave Select signal, the application can choose to manage the Slave Select signal by software. This is configured by the SSM bit in the SPICSR register (see *Figure 58*)

In software management, the external $\overline{SS}$ pin is free for other application uses and the internal $\overline{SS}$ signal level is driven by writing to the SSI bit in the SPICSR register.

In Master mode

- $\overline{SS}$ internal must be held high continuously

In Slave mode

There are two cases depending on the data/clock timing relationship (see *Figure 57*):

If CPHA = 1 (data latched on 2nd clock edge):

- $\overline{SS}$ internal must be held low during the entire transmission. This implies that in single slave applications the $\overline{SS}$ pin either can be tied to V_{SS}, or made free for standard I/O by managing the $\overline{SS}$ function by software (SSM = 1 and SSI = 0 in the in the SPICSR register)

If CPHA = 0 (data latched on 1st clock edge):

- $\overline{SS}$ internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If $\overline{SS}$ is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see *Write collision error (WCOL) on page 128*).

14.3.4 Master mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set
2. A read to the SPIDR register.

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

14.3.5 Slave mode operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:
   a) Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see *Figure 59*).

Note: The slave must have the same CPOL and CPHA settings as the master.

   b) Manage the $\overline{SS}$ pin as described in *Slave select management on page 123* and *Figure 57*. If CPHA = 1, $\overline{SS}$ must be held low continuously. If CPHA = 0, $\overline{SS}$ must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

14.3.6 Slave mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

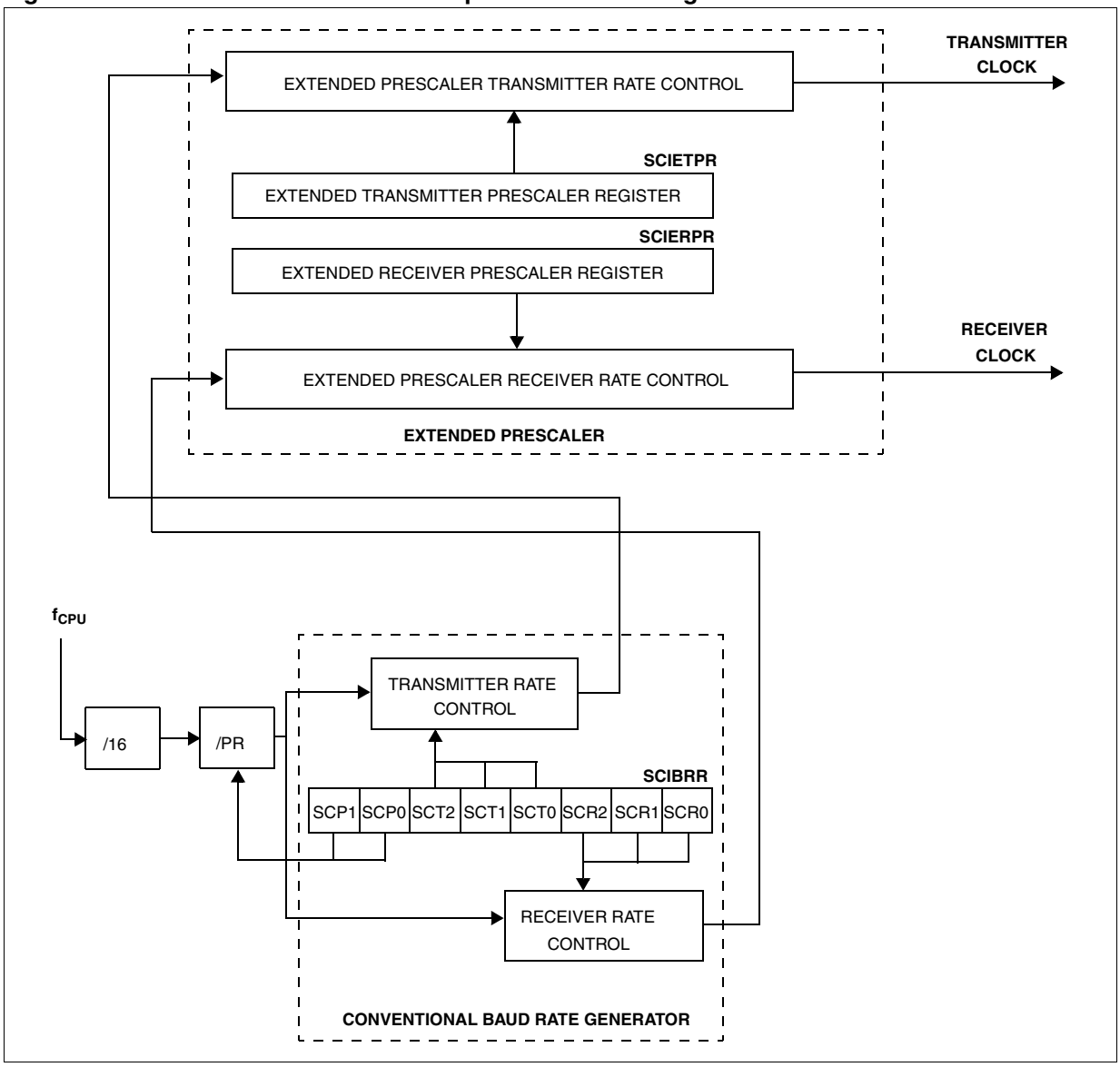
- The SPIF bit is set by hardware.
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set
2. A write or a read to the SPIDR register

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Figure 64. SCI baud rate and extended prescaler block diagram



Framing error

A framing error is detected when:

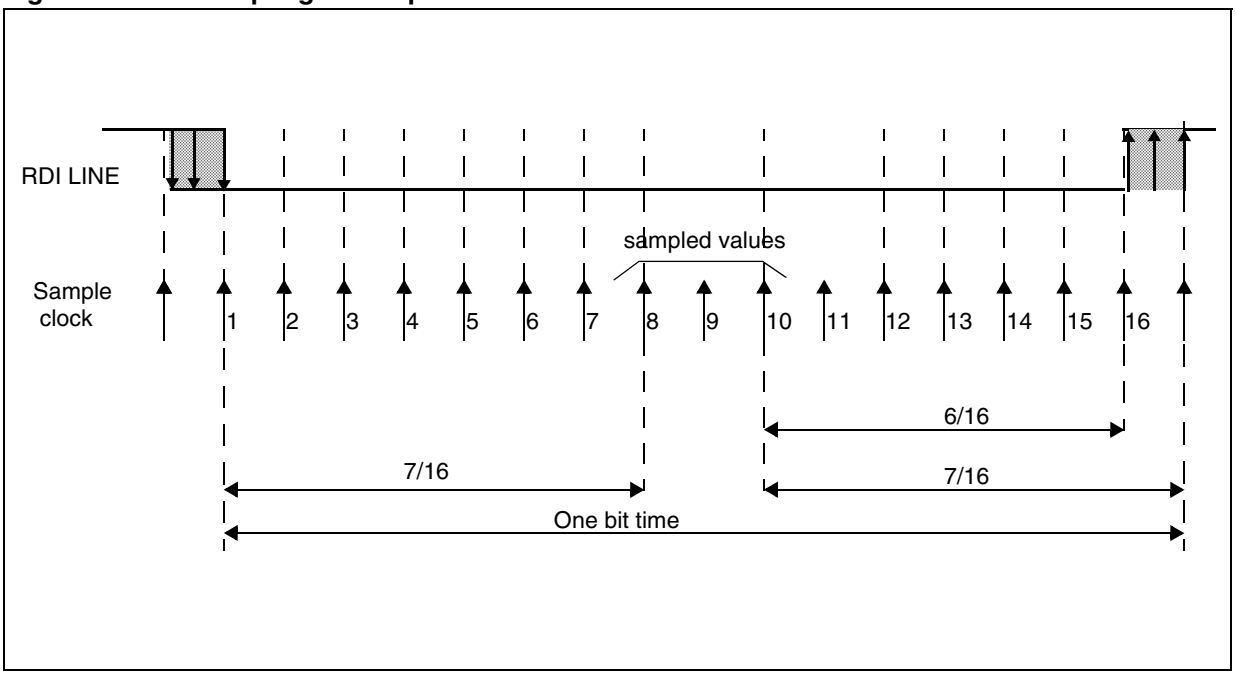
- The stop bit is not recognized on reception at the expected time, following either a de-synchronization or excessive noise.
- A break is received.

When the framing error is detected:

- The FE bit is set by hardware.
- Data is transferred from the Shift register to the SCIDR register.
- No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The FE bit is reset by a SCISR register read operation followed by a SCIDR register read operation.

Figure 65. Bit sampling in reception mode



15.5 Low power modes

Table 71. Effect of low power modes on SCI

Mode	Effect
Wait	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
Halt	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

15.6 Interrupts

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 72. SCI interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE	Yes	No
Received Data Ready to be Read	RDRF	RIE	Yes	No
Overrun Error Detected	OR		Yes	No

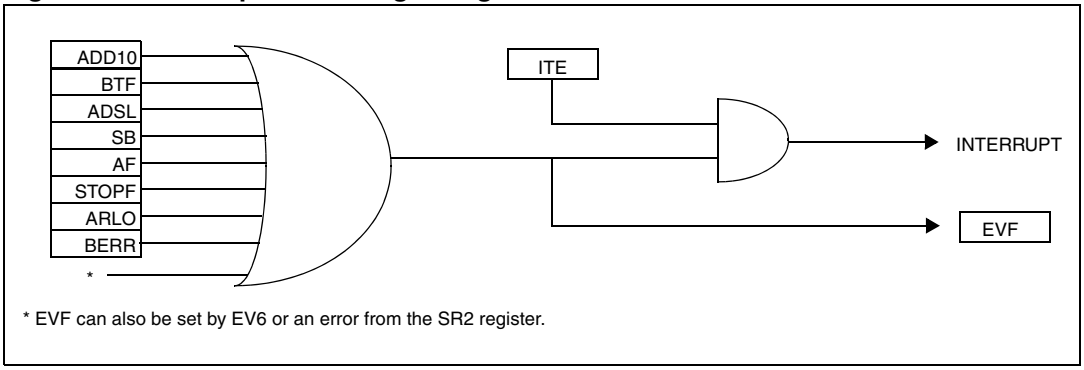
16.5 Low power modes

Table 81. Effect of low power modes on I^2C

Mode	Effect
Wait	No effect on I^2C interface. I^2C interrupts cause the device to exit from Wait mode.
Halt	I^2C registers are frozen. In Halt mode, the I^2C interface is inactive and does not acknowledge data on the bus. The I^2C interface resumes operation when the MCU is woken up by an interrupt with “exit from Halt mode” capability.

16.6 Interrupts

Figure 69. Interrupt control logic diagram



* EVF can also be set by EV6 or an error from the SR2 register.

Table 82. I^2C interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
10-bit Address Sent Event (Master mode)	ADD10	ITE	Yes	No
End of Byte Transfer Event	BTF			
Address Matched Event (Slave mode)	ADSEL			
Start Bit Generation Event (Master mode)	SB			
Acknowledge Failure Event	AF			
Stop Detection Event (Slave mode)	STOPF			
Arbitration Lost Event (Multimaster configuration)	ARLO			
Bus Error Event	BERR			

Note: The I^2C interrupt events are connected to the same interrupt vector (see Interrupts chapter). They generate an interrupt if the corresponding Enable Control bit is set and the I-bit in the CC register is reset (RIM instruction).