Welcome to E-XFL.COM

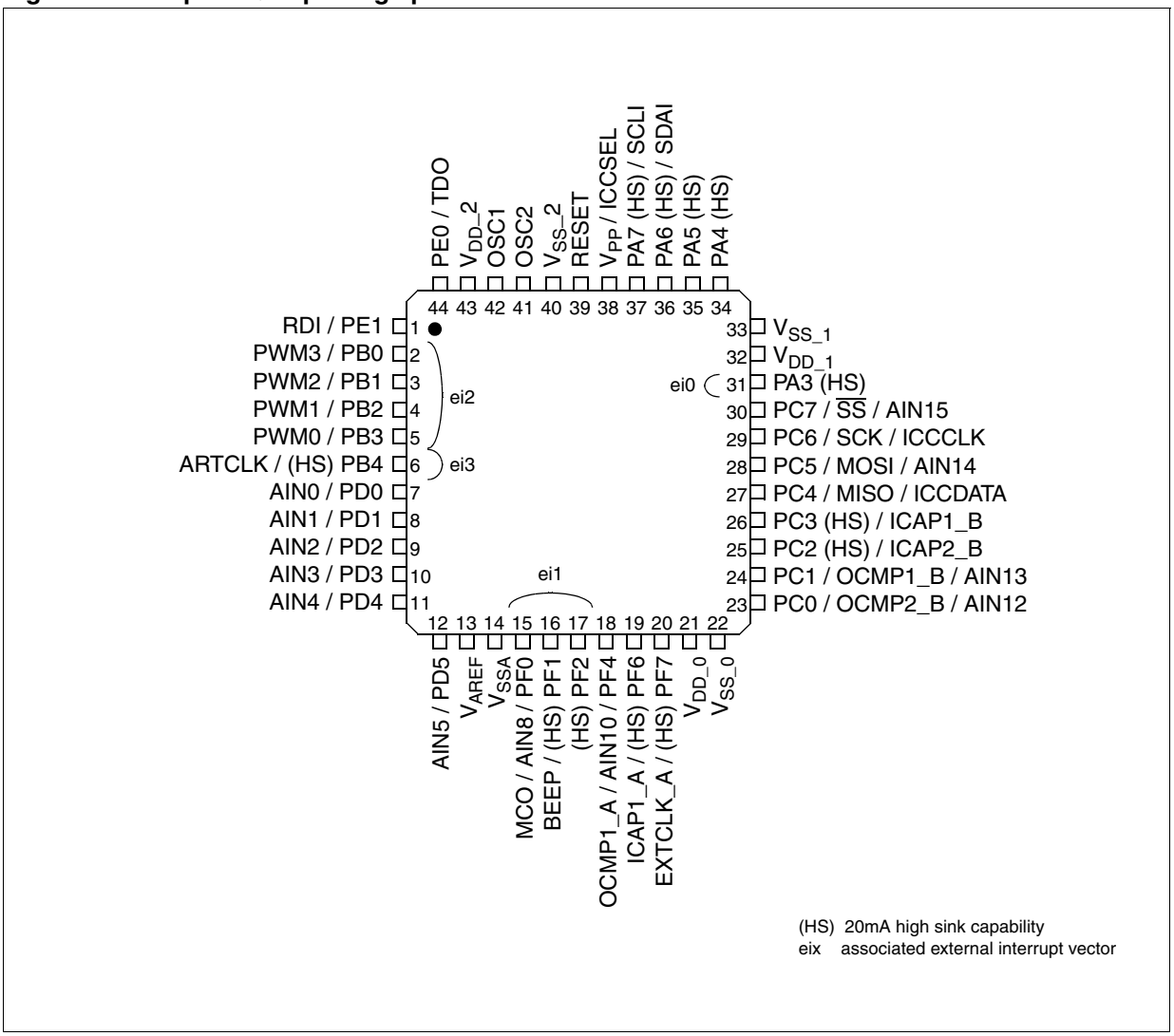
What is "Embedded - Microcontrollers"?

"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	32
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321j9ta

Figure 3. 44-pin LQFP package pinout



6.5.5 Internal watchdog RESET

The RESET sequence generated by an internal Watchdog counter overflow is shown in *Figure 13*.

Starting from the Watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Figure 13. RESET sequences

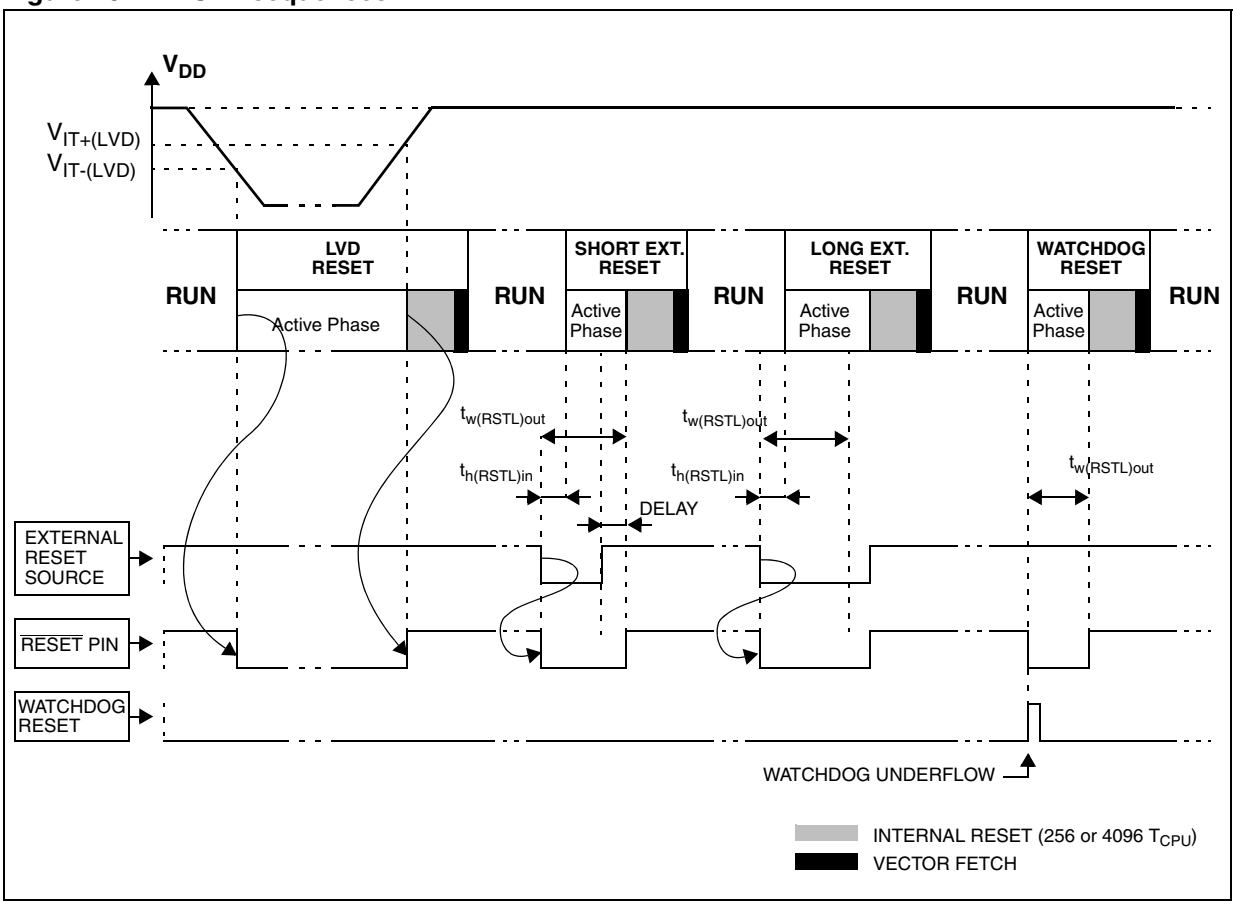
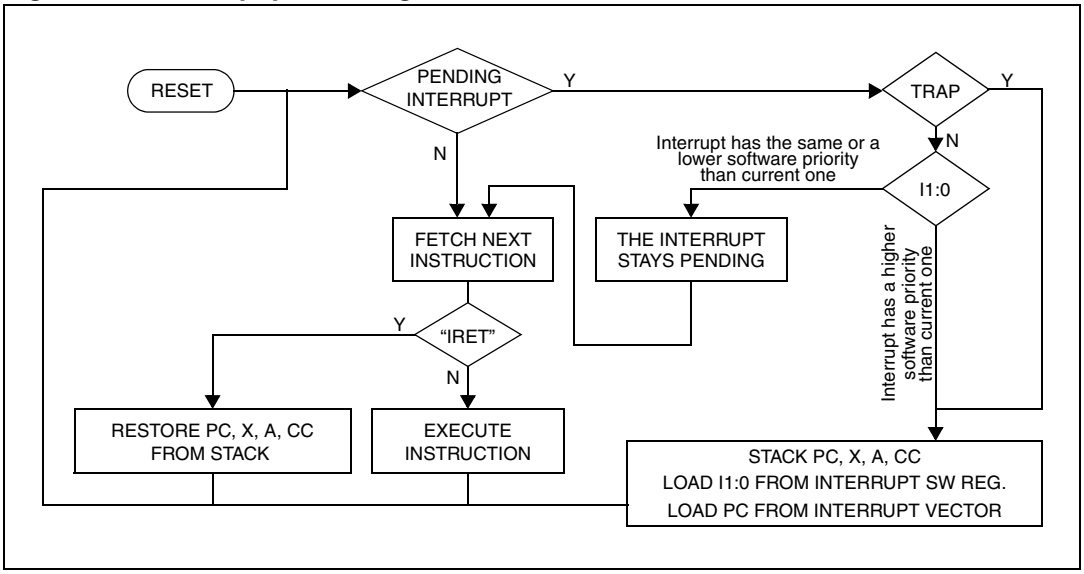


Table 15. Interrupt software priority levels

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low ↓ High	1	0
Level 1		0	1
Level 2		0	0
Level 3 (= interrupt disable)		1	1

Figure 17. Interrupt processing flowchart



Servicing pending interrupts

As several interrupts can be pending at the same time, the interrupt to be taken into account is determined by the following two-step process:

- the highest software priority interrupt is serviced,
- if several interrupts have the same software priority then the interrupt with the highest hardware priority is serviced first.

Figure 18 describes this decision process.

Figure 18. Priority decision process flowchart

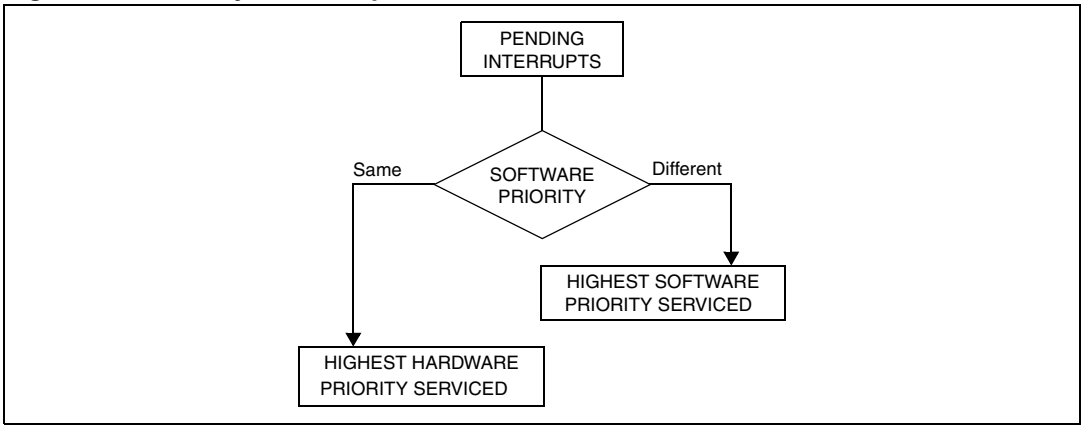
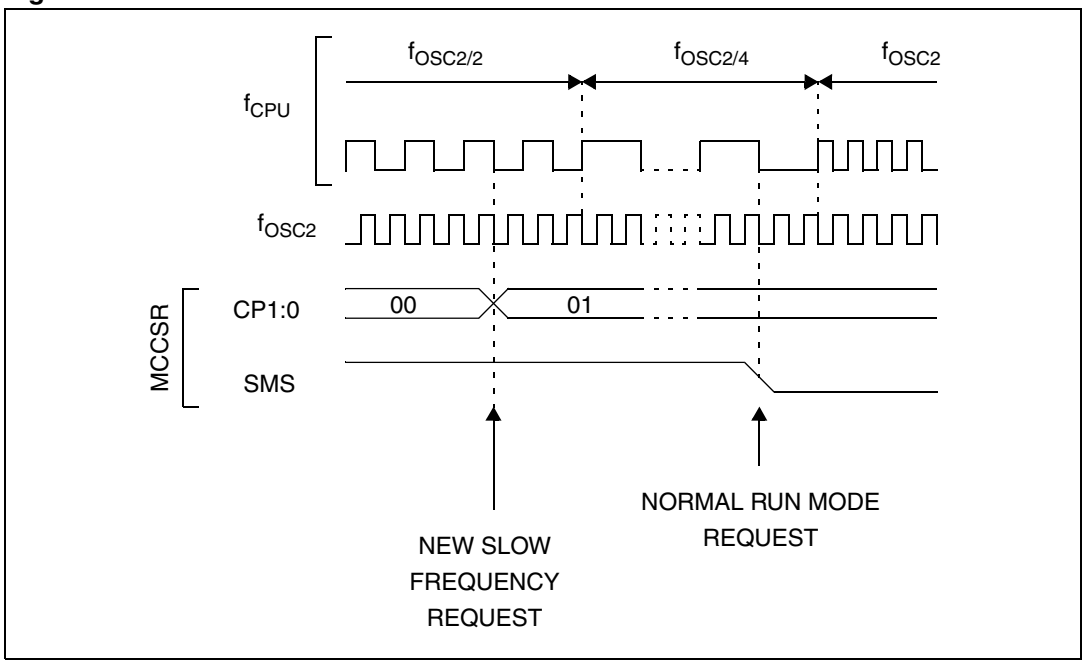


Figure 23. Slow mode clock transitions



8.3 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the 'WFI' instruction.

All peripherals remain active. During Wait mode, the I[1:0] bits of the CC register are forced to '10', to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in Wait mode until an interrupt or RESET occurs, whereupon the Program Counter branches to the starting address of the interrupt or Reset service routine.

The MCU will remain in Wait mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to the following *Figure 24*.

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).

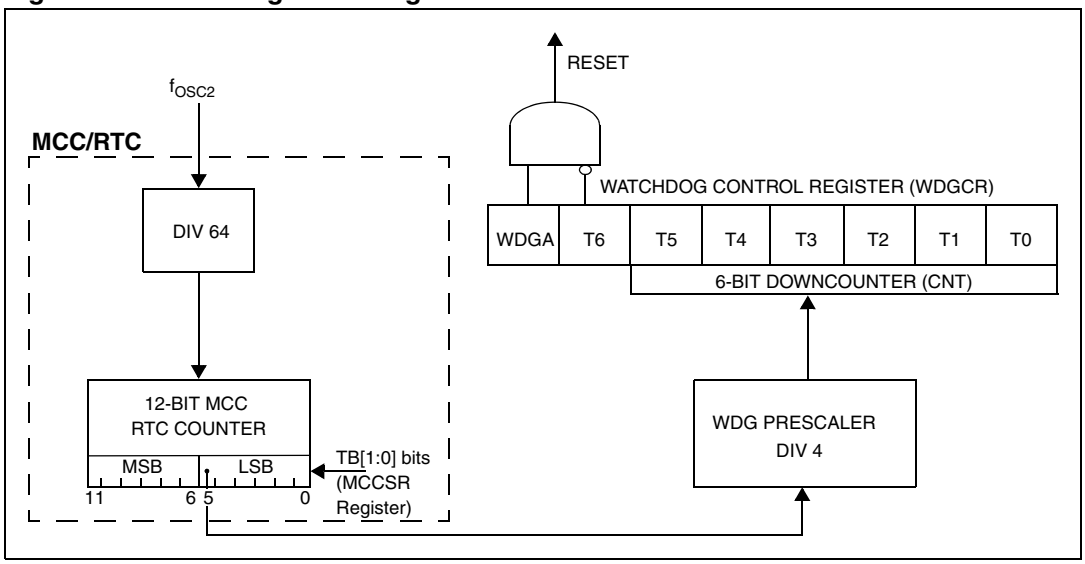
Table 33. I/O port interrupt control/wake-up capability

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								

Figure 31. Watchdog block diagram

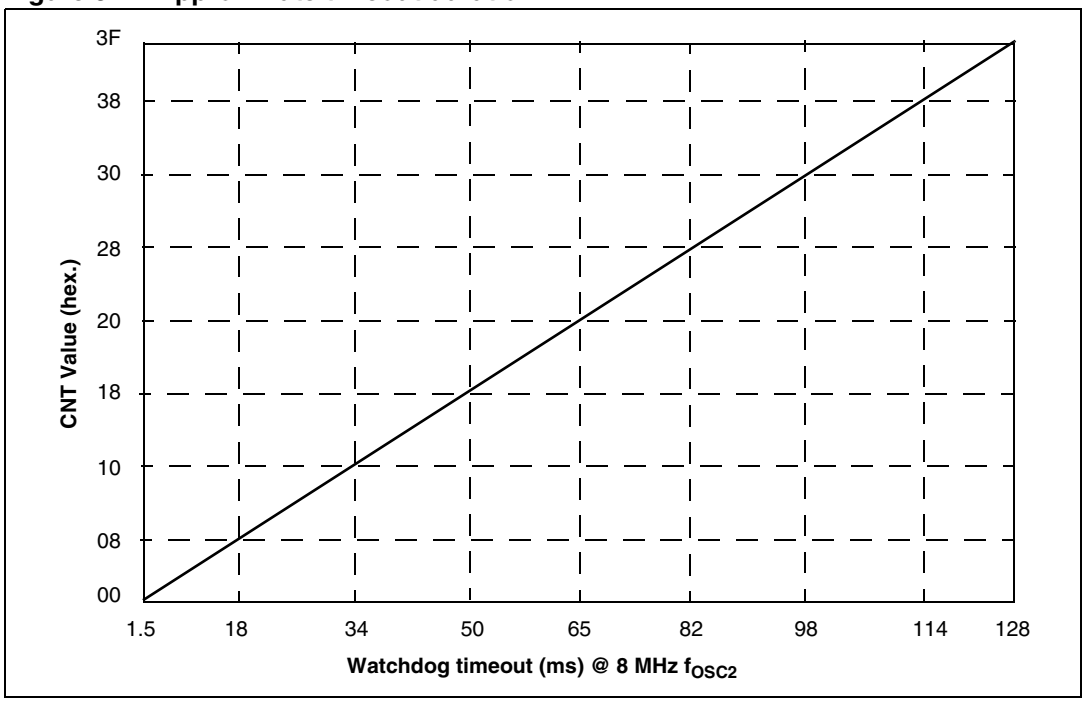


10.4 How to program the watchdog timeout

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



13 16-bit timer

13.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (input capture) or generation of up to two 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 and the CPU clock prescaler.

Some ST7 devices have two on-chip 16-bit timers. They are completely independent, and do not share any resources. They are synchronized after an MCU reset as long as the timer clock frequencies are not modified.

This description covers one or two 16-bit timers. In ST7 devices with two timers, register names are prefixed with TA (Timer A) or TB (Timer B).

13.2 Main features

- Programmable prescaler: f_{CPU} divided by 2, 4 or 8
- Overflow status flag and maskable interrupt
- External clock input (must be at least four times slower than the CPU clock speed) with the choice of active edge
- 1 or 2 Output Compare functions each with:
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- 1 or 2 Input Capture functions each with:
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse Width Modulation mode (PWM)
- One Pulse mode
- Reduced Power mode
- 5 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)[a]

The block diagram is shown in *Figure 41*.

Note: When reading an input signal on a non-bonded pin, the value will always be '1'.

a. Some timer pins may not be available (not bonded) in some ST7 devices. Refer to the device pinout description.

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	-	

14.8 SPI registers

14.8.1 Control register (SPICR)

SPICR Reset value: 0000 xxxx (0xh)

7	6	5	4	3	2	1	0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR[1:0]	
RW	RW	RW	RW	RW	RW	RW	

Table 66. SPICR register description

Bit	Name	Function
7	SPIE	*Serial Peripheral Interrupt Enable* This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SPI interrupt is generated whenever SPIF = 1, MODF = 1 or OVR = 1 in the SPICSR register.
6	SPE	*Serial Peripheral Output Enable* This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see *Master mode fault (MODF) on page 128*). The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins. 0: I/O pins free for general purpose I/O 1: SPI I/O pin alternate functions enabled
5	SPR2	*Divider Enable* This bit is set and cleared by software and is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to *Table 67*. 0: Divider by 2 enabled 1: Divider by 2 disabled *Note: This bit has no effect in slave mode.*
4	MSTR	*Master Mode* This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see *Master mode fault (MODF) on page 128*). 0: Slave mode 1: Master mode. The function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.
3	CPOL	*Clock Polarity* This bit is set and cleared by software. This bit determines the idle state of the serial Clock. The CPOL bit affects both the master and slave modes. 0: SCK pin has a low level idle state 1: SCK pin has a high level idle state *Note: If CPOL is changed at the communication byte boundaries, the SPI must be disabled by resetting the SPE bit.*
2	CPHA	*Clock Phase* This bit is set and cleared by software. 0: The first clock transition is the first data capture edge. 1: The second clock transition is the first capture edge. *Note: The slave must have the same CPOL and CPHA settings as the master.*

Table 68. SPICSR register description (continued)

Bit	Name	Function
4	MODF	*Mode Fault flag* This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see *Master mode fault (MODF) on page 128*). An SPI interrupt can be generated if SPIE = 1 in the SPICSR register. This bit is cleared by a software sequence (An access to the SPICR register while MODF = 1 followed by a write to the SPICR register). 0: No master mode fault detected 1: A fault in master mode has been detected
3	-	Reserved, must be kept cleared
2	SOD	*SPI Output Disable* This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode). 0: SPI output enabled (if SPE = 1) 1: SPI output disabled
1	SSM	*$\overline{SS}$ Management* This bit is set and cleared by software. When set, it disables the alternate function of the SPI $\overline{SS}$ pin and uses the SSI bit value instead. See *Slave select management on page 123*. 0: Hardware management ($\overline{SS}$ managed by external pin) 1: Software management (internal $\overline{SS}$ signal controlled by SSI bit. External $\overline{SS}$ pin free for general-purpose I/O)
0	SSI	*$\overline{SS}$ Internal Mode* This bit is set and cleared by software. It acts as a ‘chip select’ by controlling the level of the $\overline{SS}$ slave select signal when the SSM bit is set. 0: Slave selected 1: Slave deselected

14.8.3 Data I/O register (SPIDR)

SPIDR Reset value: Undefined

7	6	5	4	3	2	1	0
D[7:0]							
RW							

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, a write to this register will initiate transmission/reception of another byte.

Note: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

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

15.4.2 Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. When the M bit is set, word length is 9 bits and the 9th bit (the MSB) has to be stored in the T8 bit in the SCICR1 register.

Character transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see *Figure 62*).

Procedure

1. Select the M bit to define the word length.
2. Select the desired baud rate using the SCIBRR and the SCIETPR registers.
3. Set the TE bit to assign the TDO pin to the alternate function and to send an idle frame as first transmission.
4. Access the SCISR register and write the data to send in the SCIDR register (this sequence clears the TDRE bit). Repeat this sequence for each data to be transmitted.

Clearing the TDRE bit is always performed by the following software sequence:

1. An access to the SCISR register
2. A write to the SCIDR register

The TDRE bit is set by hardware and it indicates:

- The TDR register is empty.
- The data transfer is beginning.
- The next data can be written in the SCIDR register without overwriting the previous data.

This flag generates an interrupt if the TIE bit is set and the I bit is cleared in the CCR register.

When a transmission is taking place, a write instruction to the SCIDR register stores the data in the TDR register and which is copied in the shift register at the end of the current transmission.

When no transmission is taking place, a write instruction to the SCIDR register places the data directly in the shift register, the data transmission starts, and the TDRE bit is immediately set.

When a frame transmission is complete (after the stop bit) the TC bit is set and an interrupt is generated if the TCIE is set and the I bit is cleared in the CCR register.

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

1. An access to the SCISR register
2. A write to the SCIDR register

Note: The TDRE and TC bits are cleared by the same software sequence.

Break characters

Setting the SBK bit loads the shift register with a break character. The break frame length depends on the M bit (see *Figure 63*).

As long as the SBK bit is set, the SCI send break frames to the TDO pin. After clearing this

Conventional baud rate generation

The baud rate for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(16*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(16*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits)

TR = 1, 2, 4, 8, 16, 32, 64,128 (see SCT[2:0] bits)

RR = 1, 2, 4, 8, 16, 32, 64,128 (see SCR[2:0] bits)

All these bits are in the SCIBRR register.

Example: If f_{CPU} is 8 MHz (normal mode) and if PR = 13 and TR = RR = 1, the transmit and receive baud rates are 38400 baud.

Note: The baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.

Extended baud rate generation

The extended prescaler option provides a very fine tuning of the baud rate, using a 255 value prescaler, whereas the conventional baud rate generator retains industry standard software compatibility.

The extended baud rate generator block diagram is described in the *Figure 64*.

The output clock rate sent to the transmitter or to the receiver is the output from the 16 divider divided by a factor ranging from 1 to 255 set in the SCIERPR or the SCIETPR register.

Note: The extended prescaler is activated by setting the SCIETPR or SCIERPR register to a value other than zero. The baud rates are calculated as follows:

$$Tx = \frac{f_{CPU}}{16*ETPR*(PR*TR)} \qquad Rx = \frac{f_{CPU}}{16*ERPR*(PR*RR)}$$

with:

ETPR = 1,..,255 (see SCIETPR register)

ERPR = 1,..,255 (see SCIERPR register)

Receiver muting and wake-up feature

In multiprocessor configurations it is often desirable that only the intended message recipient should actively receive the full message contents, thus reducing redundant SCI service overhead for all non-addressed receivers.

The non-addressed devices may be placed in sleep mode by means of the muting function.

Setting the RWU bit by software puts the SCI in sleep mode:

- All the reception status bits cannot be set.
- All the receive interrupts are inhibited.

A muted receiver may be awakened by one of the following two ways:

- by Idle Line detection if the WAKE bit is reset
- by Address Mark detection if the WAKE bit is set

16 I²C bus interface (I2C)

16.1 Introduction

The I²C bus interface serves as an interface between the microcontroller and the serial I²C bus. It provides both multimaster and slave functions, and controls all I²C bus-specific sequencing, protocol, arbitration and timing. It supports fast I²C mode (400 kHz).

16.2 Main features

- Parallel-bus/I²C protocol converter
- Multimaster capability
- 7-bit/10-bit addressing
- SMBus V1.1 compliant
- Transmitter/Receiver flag
- End-of-byte transmission flag
- Transfer problem detection

16.2.1 I²C master features

- Clock generation
- I²C bus busy flag
- Arbitration Lost flag
- End of byte transmission flag
- Transmitter/Receiver flag
- Start bit detection flag
- Start and Stop generation

16.2.2 I²C slave features

- Stop bit detection
- I²C bus busy flag
- Detection of misplaced start or stop condition
- Programmable I²C address detection
- Transfer problem detection
- End-of-byte transmission flag
- Transmitter/Receiver flag

16.4 Functional description

Refer to the CR, SR1 and SR2 registers in *Section 16.7* for the bit definitions.

By default the I^2C interface operates in Slave mode (M/SL bit is cleared) except when it initiates a transmit or receive sequence.

First the interface frequency must be configured using the FRi bits in the OAR2 register.

16.4.1 Slave mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register; then it is compared with the address of the interface or the General Call address (if selected by software).

Note: In 10-bit addressing mode, the comparison includes the header sequence (11110xx0) and the two most significant bits of the address.

Header matched (10-bit mode only): The interface generates an acknowledge pulse if the ACK bit is set.

Address not matched: The interface ignores it and waits for another Start condition.

Address matched: The interface generates in sequence:

- an acknowledge pulse if the ACK bit is set
- EVF and ADSL bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register, **holding the SCL line low** (see *Figure 68: Transfer sequencing* EV1).

Next, in 7-bit mode read the DR register to determine from the least significant bit (Data Direction Bit) if the slave must enter Receiver or Transmitter mode.

In 10-bit mode, after receiving the address sequence the slave is always in receive mode. It will enter transmit mode on receiving a repeated Start condition followed by the header sequence with matching address bits and the least significant bit set (11110xx1).

Slave receiver

Following the address reception and after the SR1 register has been read, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- an acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see *Figure 68: Transfer sequencing* EV2).

Slave transmitter

Following the address reception and after SR1 register has been read, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see *Figure 68: Transfer sequencing* EV3).

When the acknowledge pulse is received:

- The EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Table 88. OAR1 register description

Bit	Name	Function	
		7-bit addressing mode	10-bit addressing mode
7:1	ADD[7:1]	*Interface address* These bits define the I²C bus address of the interface. They are not cleared when the interface is disabled (PE = 0).	Not applicable
0	ADD0	*Address direction bit* This bit is 'don't care', the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE = 0). Address 01h is always ignored.	
7:0	ADD[7:0]	Not applicable	*Interface address* These are the least significant bits of the I²C bus address of the interface. They are not cleared when the interface is disabled (PE = 0).

16.7.7 I²C own address register (OAR2)

OAR2 Reset value: 0100 0000 (40h)

7	6	5	4	3	2	1	0
FR[1:0]		Reserved			ADD[9:8]		Reserved
RW		-			RW		-

Table 89. OAR2 register description

Bit	Name	Function
7:6	FR[1:0]	*Frequency bits* These bits are set by software only when the interface is disabled (PE = 0). To configure the interface to I²C specified delays, select the value corresponding to the CPU frequency f_{CPU}. 00: f_{CPU} < 6 MHz 01: f_{CPU} = 6 to 8 MHz
5:3	-	Reserved
2:1	ADD[9:8]	*Interface address* These are the most significant bits of the I²C bus address of the interface (10-bit mode only). They are not cleared when the interface is disabled (PE = 0).
0	-	Reserved

19.4.3 On-chip peripherals

Measured on LQFP64 generic board T_A = 25°C, f_{CPU} = 4 MHz.

Table 113. On-chip peripherals current consumption

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(TIM)}$	16-bit timer supply current(1)	V_{DD} = 5.0V	50	µA
$I_{DD(ART)}$	ART PWM supply current(2)	V_{DD} = 5.0V	75	µA
$I_{DD(SPI)}$	SPI supply current(3)	V_{DD} = 5.0V	400	µA
$I_{DD(SCI)}$	SCI supply current(4)			
$I_{DD(I2C)}$	I2C supply current(5)	V_{DD} = 5.0V	175	µA
$I_{DD(ADC)}$	ADC supply current when converting(6)	V_{DD} = 5.0V	400	µA

1. Data based on a differential I_{DD} measurement between reset configuration (timer counter running at $f_{CPU}/4$) and timer counter stopped (only TIMD bit set). Data valid for one timer.
2. Data based on a differential I_{DD} measurement between reset configuration (timer stopped) and timer counter enabled (only TCE bit set).
3. Data based on a differential I_{DD} measurement between reset configuration (SPI disabled) and a permanent SPI master communication at maximum speed (data sent equal to 55h). This measurement includes the pad toggling consumption.
4. Data based on a differential I_{DD} measurement between SCI low power state (SCID = 1) and a permanent SCI data transmit sequence.
5. Data based on a differential I_{DD} measurement between reset configuration (I2C disabled) and a permanent I2C master communication at 100 kHz (data sent equal to 55h). This measurement includes the pad toggling consumption (27k ohm external pull-up on clock and data lines).
6. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions.

19.7 EMC (electromagnetic compatibility) characteristics

Susceptibilitytests are performed on a sample basis during product characterization.

19.7.1 Functional EMS (electromagnetic susceptibility)

Based on a simple running application on the product (toggling two LEDs through I/O ports), the product is stressed by two electromagnetic events until a failure occurs (indicated by the LEDs).

- **ESD**: Electrostatic discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- **FTB**: A burst of fast transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100pF capacitor until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed. The test results given in *Table 122* below are based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

EMC characterization and optimization are performed at component level with a typical application environment and simplified MCU software. It should be noted that good EMC performance is highly dependent on the user application and the software in particular.

Therefore it is recommended that the user applies EMC software optimization and prequalification tests in relation with the EMC level requested for his application.

Software recommendations

The software flowchart must include the management of runaway conditions such as:

- Corrupted program counter
- Unexpected reset
- Critical Data corruption (control registers...)

Prequalification trials

Most of the common failures (unexpected reset and program counter corruption) can be reproduced by manually forcing a low state on the $\overline{\text{RESET}}$ pin or the oscillator pins for 1 second.

To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behavior is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

19.8 I/O port pin characteristics

19.8.1 General characteristics

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Table 126. I/O port pin general characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage(1)	CMOS ports			$0.3xV_{DD}$	V
V_{IH}	Input high level voltage(1)		$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis(2)			0.7		
$I_{INJ(PIN)}$(3)	Injected current on PC6 pin (Flash devices only)	V_{DD} = 5V	0		+4	mA
	Injected current on an I/O pin				±4	
$\Sigma I_{INJ(PIN)}$(3)	Total injected current (sum of all I/O and control pins)				±25	
I_L	Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$			±1	µA
I_S	Static current consumption	Floating input mode(4)		400		
R_{PU}	Weak pull-up equivalent resistor(5)	$V_{IN} = V_{SS}$ V_{DD} = 5V	50	120	250	kΩ
C_{IO}	I/O pin capacitance			5		pF
$t_{f(IO)out}$	Output high to low level fall time(1)	C_L = 50pF Between 10% and 90%		25		ns
$t_{r(IO)out}$	Output low to high level rise time(1)			25		
$t_{w(IT)in}$	External interrupt pulse time(6)		1			t_{CPU}

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
3. When the current limitation is not possible, the V_{IN} maximum must be respected, otherwise refer to $I_{INJ(PIN)}$ specification. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$. Refer to *Section 19.2.2: Current characteristics* for more details.
4. Configuration not recommended. All unused pins must be kept at a fixed voltage. This can be done by using the output mode of the I/O, for example, and leaving the I/O unconnected on the board or by an external pull-up or pull-down resistor (see *Figure 82*). Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values.
5. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in *Figure 83*).
6. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

ST72321-Auto Microcontroller FASTROM/ROM Option List

(Last update August 2007)

Customer: . . . . . .
Address: . . . . . .
. . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference/ROM Code* : . . . . . .

*The ROM code name is assigned by STMicroelectronics.
ROM code must be sent in .S19 format. .Hex extension cannot be processed.

Device Type/Memory Size/Package (check only one option):

FASTROM DEVICE:	60K	48K	32K
LQFP44 10x10:	[ ] ST72P321(J9)T	[ ] ST72P321(J7)T	
LQFP64 10x10:	[ ] ST72P321(AR9)T	[ ] ST72P321(AR7)T	[ ] ST72P321(AR6)T
LQFP64 14x14:	[ ] ST72P321(R9)T	[ ] ST72P321(R7)T	[ ] ST72P321(R6)T

ROM DEVICE:	60K	48K	32K
LQFP44 10x10:	[ ] ST72321(J9)T	[ ] ST72321(J7)T	
LQFP64 10x10:	[ ] ST72321(AR9)T	[ ] ST72321(AR7)T	[ ] ST72321(AR6)T
LQFP64 14x14:	[ ] ST72321(R9)T	[ ] ST72321(R7)T	[ ] ST72321(R6)T

Conditioning for LQFP package (check only one option):

[ ] Tape & Reel [ ] Tray

Temperature range : [ ] A (-40°C to +85°C)
[ ] B (-40°C to +105°C)
[ ] C (-40°C to +125°C)

Special Marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ " (10 characters max)
Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Clock Source Selection: [ ] Resonator:
[ ] LP: Low power resonator (1 to 2 MHz)
[ ] MP: Medium power resonator (2 to 4 MHz)
[ ] MS: Medium speed resonator (4 to 8 MHz)
[ ] HS: High speed resonator (8 to 16 MHz)
[ ] Internal RC(1)
[ ] External Clock (sets MP Medium Power resonator in Option Byte)

PLL(2) [ ] Disabled [ ] Enabled

LVD Reset [ ] Disabled [ ] High threshold
[ ] Med.threshold [ ] Low threshold

Reset Delay [ ] 256 Cycles [ ] 4096 Cycles

Watchdog Selection [ ] Software Activation [ ] Hardware Activation

Watchdog Reset on Halt: [ ] Reset [ ] No Reset

Readout Protection(3): [ ] Disabled [ ] Enabled

Date. . . . . Signature . . . . . .

Note 1: Internal RC can only be used if LVD is enabled
Note 2: PLL must not be enabled if internal RC or External Clock is selected.
Note 3: Readout protection not supported if LVD is enabled.

Caution: The Readout Protection binary value is inverted between ROM and Flash products. The option byte checksum will differ between ROM and Flash.

Please download the latest version of this option list from www.st.com.