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/st72f321bj9tae

Table 2. Device pin description (continued)

Pin No.		Pin name	Type	Level		Port						Main function (after reset)	Alternate function
						Input				Output			
LQFP64	LQFP44			Input	Output	float	wpu	int	ana	OD	PP		
62	1	PE1/RDI	I/O	C_T		**X**	X			X	X	Port E1	SCI Receive Data In
63	-(1)	PE2	I/O	C_T			**X**			X(6)	X(6)	Port E2	
64	-(2)(1)	PE3	I/O	C_T		**X**	X			X	X	Port E3	

1. On the chip, each I/O port may have up to 8 pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.
2. On the chip, each I/O port may have up to eight pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.
3. It is mandatory to connect all available V_{DD} and V_{AREF} pins to the supply voltage and all V_{SS} and V_{SSA} pins to ground.
4. Not connected in 48 Kbyte and 64 Kbyte ROM devices
5. OSC1 and OSC2 pins connect a crystal/ceramic resonator or an external source to the on-chip oscillator; see *Section 6.4: Multi-oscillator (MO)* and *Section 19.5: Clock and timing characteristics* for more details.
6. Pull-up always activated on PE2; see limitation *Section 22.1.8: Pull-up always active on PE2*.

Legend / Abbreviations for *Table 2*:

Type: I = input, O = output, S = supply

Input level: A = dedicated analog input

In/Output level: C = CMOS $0.3V_{DD}/0.7V_{DD}$, C_T = CMOS $0.3V_{DD}/0.7V_{DD}$ with input trigger

Output level: HS = 20mA high sink (on N-buffer only)

Port and control configuration:

- Input: float = floating, wpu = weak pull-up, int = interrupt(a), ana = analog
- Output: OD = open-drain(b), PP = push-pull

a. In the interrupt input column, “eiX” defines the associated external interrupt vector. If the weak pull-up column (wpu) is merged with the interrupt column (int), then the I/O configuration is pull-up interrupt input, otherwise the configuration is floating interrupt input.

b. In the open-drain output column, “T” defines a true open-drain I/O (P-Buffer and protection diode to V_{DD} are not implemented). See *Section 9: I/O ports* and *Section 19.8: I/O port pin characteristics* for more details.

Table 3. Hardware register map (continued)

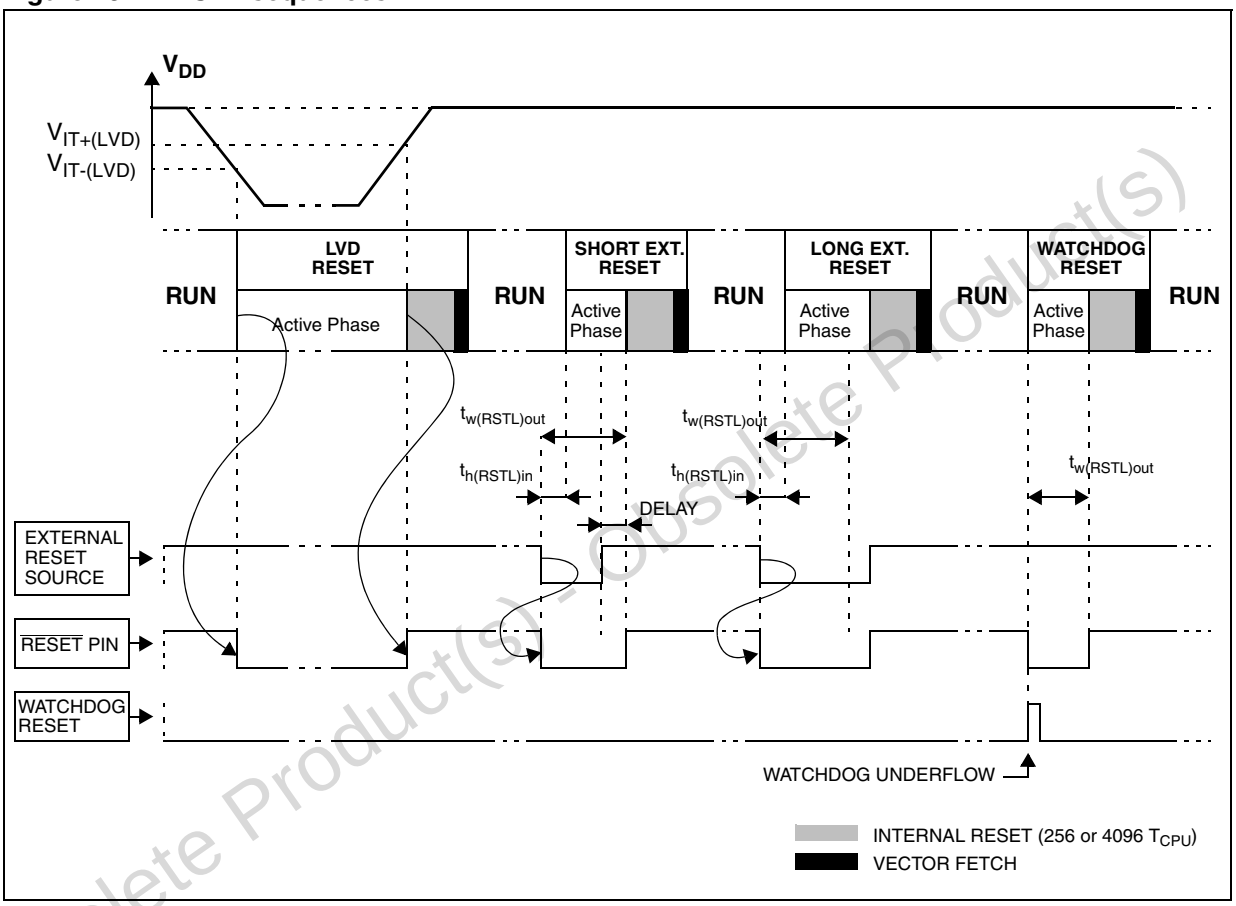
Address	Block	Register label	Register name	Reset status	Remarks
000Ch 000Dh 000Eh	Port E(2)	PEDR PEDDR PEOR	Port E Data Register Port E Data Direction Register Port E Option Register	00h(1) 00h 00h	R/W R/W(2) R/W(2)
000Fh 0010h 0011h	Port F(2)	PFDR PFDDR PFOR	Port F Data Register Port F Data Direction Register Port F Option Register	00h(1) 00h 00h	R/W R/W R/W
0012h to 0017h	Reserved area (6 bytes)				
0018h 0019h 001Ah 001Bh 001Ch 001Dh 001Eh	I^2C	I2CCR I2CSR1 I2CSR2 I2CCCR I2COAR1 I2COAR2 I2CDR	I^2C Control Register I^2C Status Register 1 I^2C Status Register 2 I^2C Clock Control Register I^2C Own Address Register 1 I^2C Own Address Register2 I^2C Data Register	00h 00h 00h 00h 00h 00h 00h	R/W Read only Read only R/W R/W R/W R/W
001Fh 0020h	Reserved area (2 bytes)				
0021h 0022h 0023h	SPI	SPIDR SPICR SPICSR	SPI Data I/O Register SPI Control Register SPI Control/Status Register	xxh 0xh 00h	R/W R/W R/W
0024h 0025h 0026h 0027h	ITC	ISPR0 ISPR1 ISPR2 ISPR3	Interrupt Software Priority Register 0 Interrupt Software Priority Register 1 Interrupt Software Priority Register 2 Interrupt Software Priority Register 3	FFh FFh FFh FFh	R/W R/W R/W R/W
0028h		EICR	External Interrupt Control Register	00h	R/W
0029h	FLASH	FCSR	Flash Control/Status Register	00h	R/W
002Ah	WATCHDOG	WDGCR	Watchdog Control Register	7Fh	R/W
002Bh		SICSR	System Integrity Control/Status Register	000x 000xb	R/W
002Ch 002Dh	MCC	MCCSR MCCBCR	Main Clock Control/Status Register Main Clock Controller/Beep Control Register	00h 00h	R/W R/W
002Eh to 0030h	Reserved area (3 bytes)				

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



8 Power saving modes

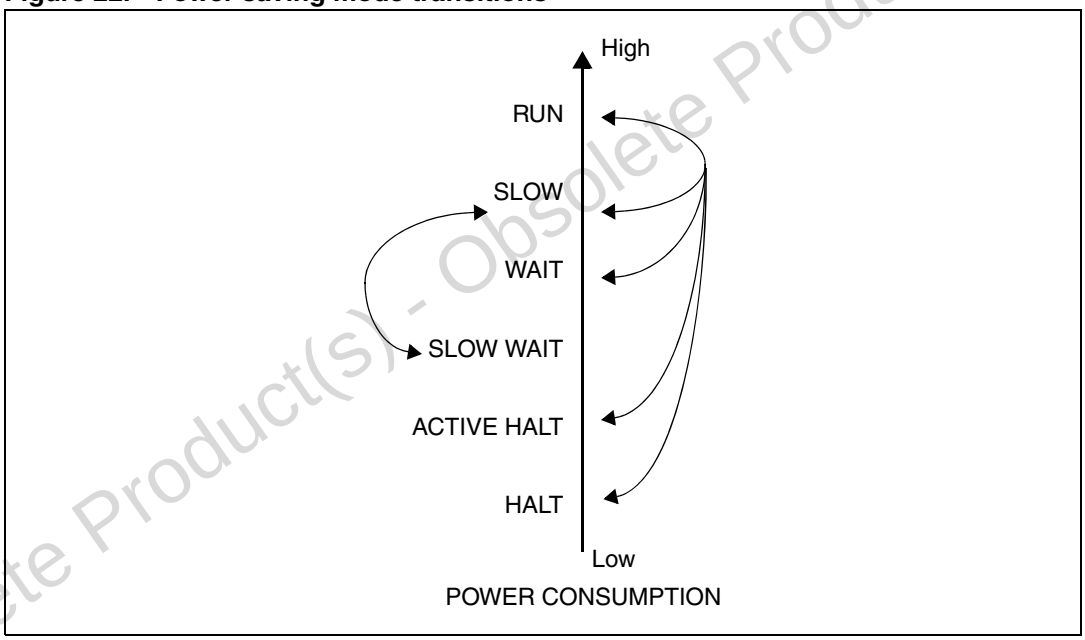
8.1 Introduction

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7 (see *Figure 22*): Slow, Wait (Slow Wait), Active Halt and Halt.

After a RESET the normal operating mode is selected by default (Run mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided or multiplied by 2 (f_{OSC2}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

Figure 22. Power saving mode transitions



8.2 Slow mode

This mode has two targets:

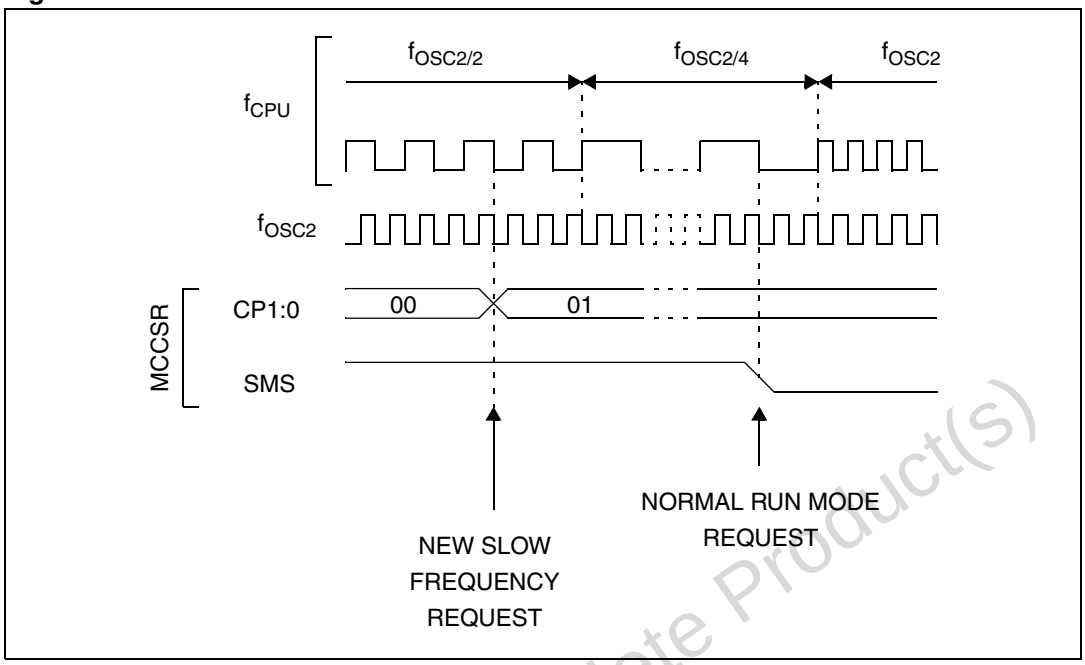
- To reduce power consumption by decreasing the internal clock in the device,
- To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

Slow mode is controlled by three bits in the MCCSR register: the SMS bit which enables or disables Slow mode and two CPx bits which select the internal slow frequency (f_{CPU}).

In this mode, the master clock frequency (f_{OSC2}) can be divided by 2, 4, 8 or 16. The CPU and peripherals are clocked at this lower frequency (f_{CPU}).

Note: Slow Wait mode is activated when entering the Wait mode while the device is already in Slow mode.

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.

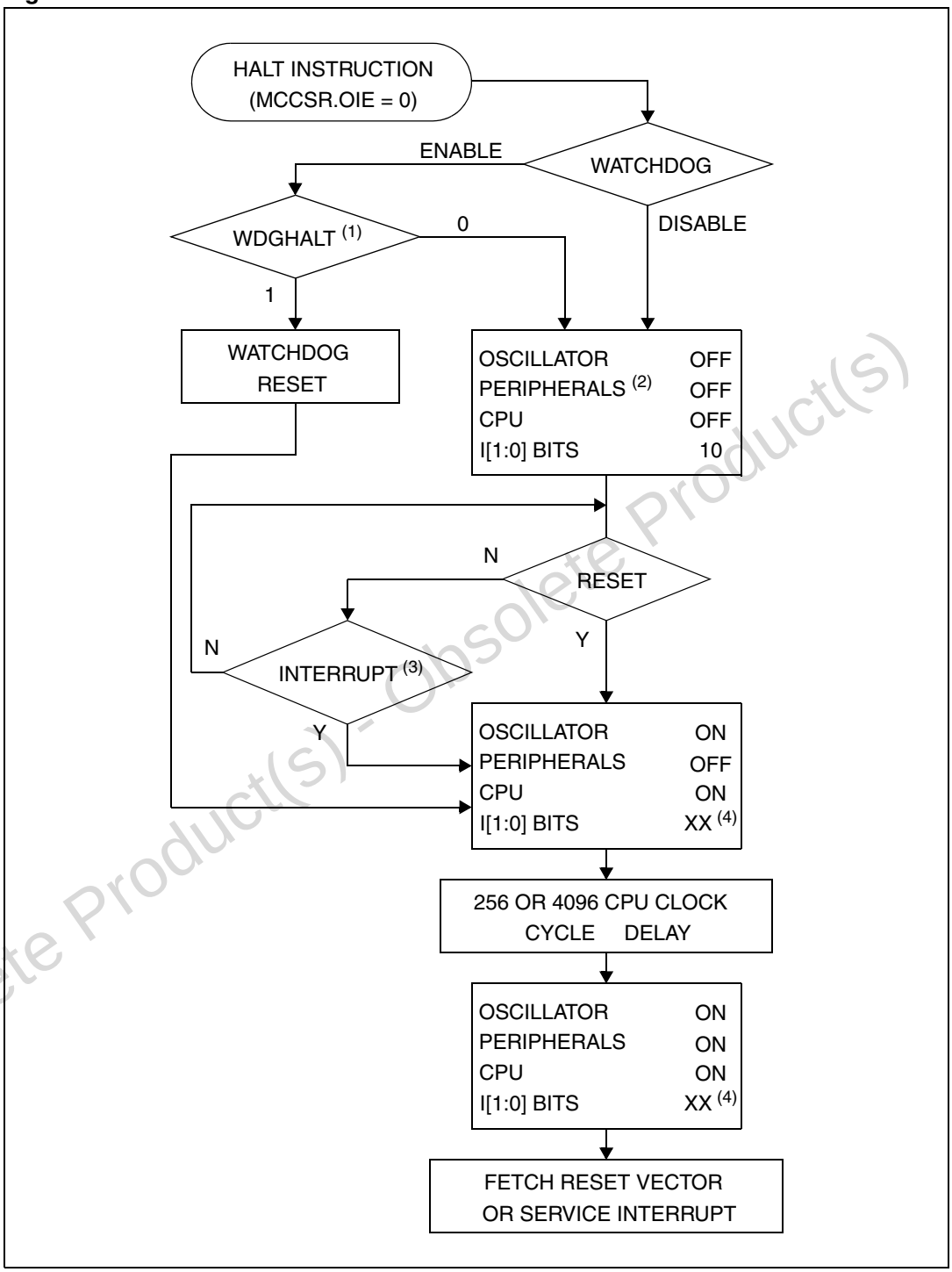
This power saving mode is selected by calling the 'WFI' instruction.

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*.

Figure 28. Halt mode flowchart



1. WDGHALT is an option bit. See *Section 21.1.1: Flash configuration on page 225* for more details.
2. Peripheral clocked with an external clock source can still be active.
3. Only some specific interrupts can exit the MCU from Halt mode (such as external interrupt). Refer to *Table 19: Interrupt mapping* for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

Table 39. MCCSR register description (continued)

Bit	Name	Function
0	OIF	*Oscillator interrupt flag* This bit is set by hardware and cleared by software reading the MCCSR register. It indicates when set that the main oscillator has reached the selected elapsed time (TB1:0). 0: Timeout not reached 1: Timeout reached **Caution**: The BRES and BSET instructions must not be used on the MCCSR register to avoid unintentionally clearing the OIF bit.

Table 40. Time base selection

Counter prescaler	Time base		TB1	TB0
	f_{OSC2} = 4 MHz	f_{OSC2} = 8 MHz		
16000	4ms	2ms	0	0
32000	8ms	4ms	0	1
80000	20ms	10ms	1	0
200000	50ms	25ms	1	1

11.8.2 MCC beep control register (MCCBCR)

MCCBCR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved						BC[1:0]	
-						RW	

Table 41. MCCBCR register description

Bit	Name	Function
7:2	-	Reserved, must be kept cleared.
1:0	BC[1:0]	*Beep control* These 2 bits select the PF1 pin beep capability (see *Table 42*).

Table 42. Beep frequency selection

BC1	BC0	Beep mode with f_{OSC2} = 8 MHz	
0	0	Off	
0	1	~2 kHz	Output Beep signal ~50% duty cycle
1	0	~1 kHz	
1	1	~500 Hz	

The beep output signal is available in Active Halt mode but has to be disabled to reduce consumption.

12.3.5 Duty cycle registers (PWMDCRx)

PWMDCRx Reset value: 0000 0000 (00h)

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

Table 51. PWMDCRx register description

Bit	Name	Function
7:0	DC[7:0]	*Duty Cycle Data* These bits are set and cleared by software.

A PWMDCRx register is associated with the OCRx register of each PWM channel to determine the second edge location of the PWM signal (the first edge location is common to all channels and given by the ARTARR register). These PWMDCR registers allow the duty cycle to be set independently for each PWM channel.

12.3.6 Input capture control / status register (ARTICCSR)

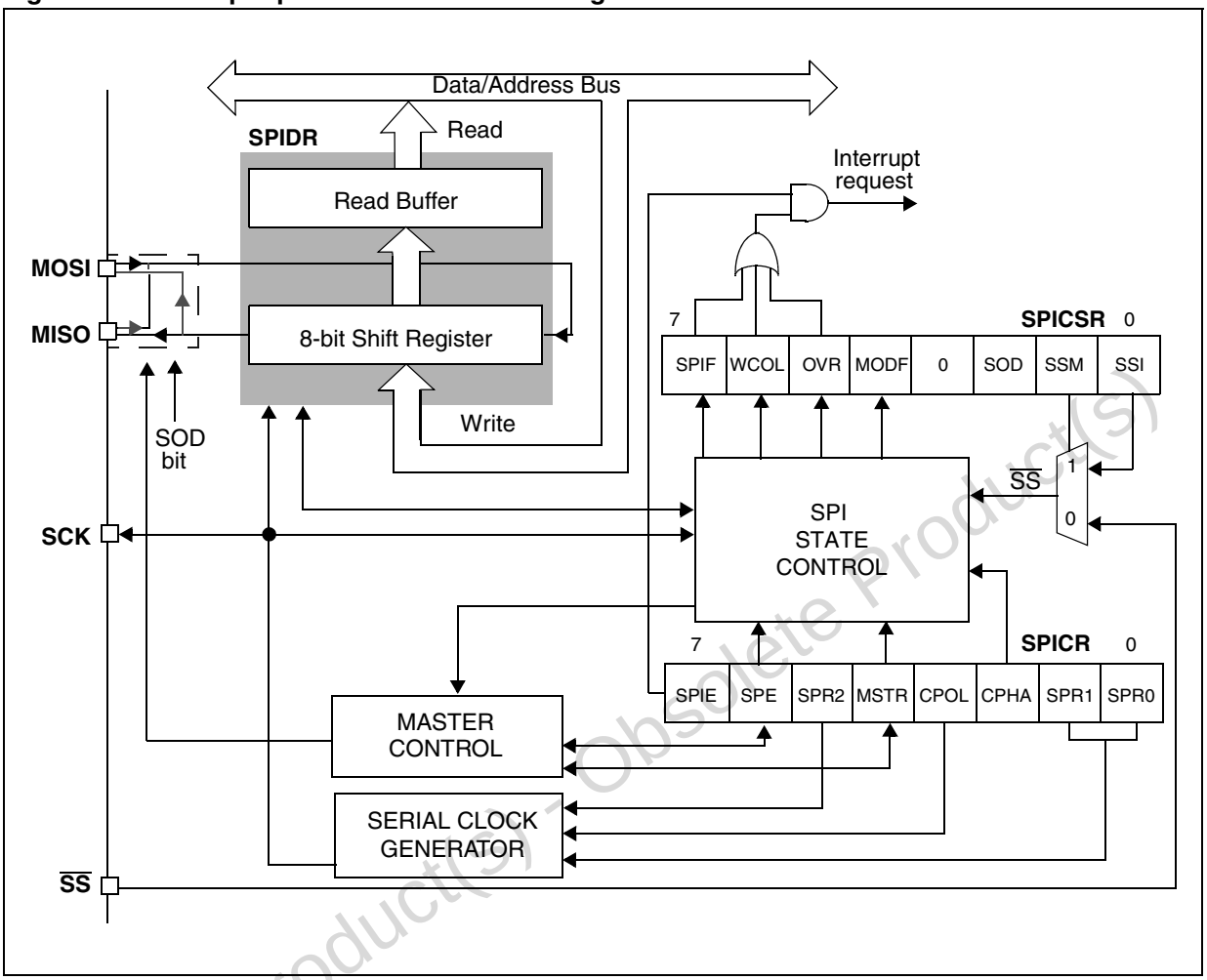
ARTICCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved		CS[2:1]		CIE[2:1]		CF[2:1]	
-		RW		RW		RW	

Table 52. ARTICCSR register description

Bit	Name	Function
7:6	-	Reserved, always read as 0.
5:4	CS[2:1]	*Capture Sensitivity* These bits are set and cleared by software. They determine the trigger event polarity on the corresponding input capture channel. 0: Falling edge triggers capture on channel x 1: Rising edge triggers capture on channel x
3:2	CIE[2:1]	*Capture Interrupt Enable* These bits are set and cleared by software. They enable or disable the Input capture channel interrupts independently. 0: Input capture channel x interrupt disabled 1: Input capture channel x interrupt enabled
1:0	CF[2:1]	*Capture Flag* These bits are set by hardware and cleared by software reading the corresponding ARTICRx register. Each CFx bit indicates that an input capture x has occurred. 0: No input capture on channel x 1: An input capture has occurred on channel x.

Figure 55. Serial peripheral interface block diagram



14.3.1 Functional description

A basic example of interconnections between a single master and a single slave is illustrated in *Figure 56*.

The MOSI pins are connected together and the MISO pins are connected together. In this way data is transferred serially between master and slave (most significant bit first).

The communication is always initiated by the master. When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex communication with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

To use a single data line, the MISO and MOSI pins must be connected at each node (in this case only simplex communication is possible).

Four possible data/clock timing relationships may be chosen (see *Figure 59*) but master and slave must be programmed with the same timing mode.

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

Error cases

- **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and BERR bits are set by hardware with an interrupt if ITE is set.
  Note that BERR will not be set if an error is detected during the first or second pulse of each 9-bit transaction:
  - ***Single Master Mode***
    If a Start or Stop is issued during the first or second pulse of a 9-bit transaction, the BERR flag will not be set and transfer will continue however the BUSY flag will be reset. To work around this, slave devices should issue a NACK when they receive a misplaced Start or Stop. The reception of a NACK or BUSY by the master in the middle of communication makes it possible to re-initiate transmission.
  - ***Multimaster Mode***
    Normally the BERR bit would be set whenever unauthorized transmission takes place while transfer is already in progress. However, an issue will arise if an external master generates an unauthorized Start or Stop while the I^2C master is on the first or second pulse of a 9-bit transaction. It is possible to work around this by polling the BUSY bit during I^2C master mode transmission. The resetting of the BUSY bit can then be handled in a similar manner as the BERR flag being set.
- **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the Start or Stop bit. The AF bit is cleared by reading the I2CSR2 register. However, if read before the completion of the transmission, the AF flag will be set again, thus possibly generating a new interrupt. Software must ensure either that the SCL line is back at 0 before reading the SR2 register, or be able to correctly handle a second interrupt during the 9th pulse of a transmitted byte.
- **ARLO:** Detection of an arbitration lost condition.
  In this case the ARLO bit is set by hardware (with an interrupt if the ITE bit is set and the interface goes automatically back to slave mode (the M/SL bit is cleared).

Note: In all these cases, the SCL line is not held low; however, the SDA line can remain low due to possible '0' bits transmitted last. It is then necessary to release both lines by software.

16.7 Register description

16.7.1 I²C control register (CR)

CR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved		PE	ENGC	START	ACK	STOP	ITE
-		RW	RW	RW	RW	RW	RW

Table 82. CR register description

Bit	Name	Function
7:6	-	Reserved. Forced to 0 by hardware.
5	PE	*Peripheral enable* This bit is set and cleared by software. 0: Peripheral disabled 1: Master/Slave capability *Notes:* *- When PE = 0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released while PE = 0* *- When PE = 1, the corresponding I/O pins are selected by hardware as alternate functions.* *To enable the I²C interface, write the CR register **TWICE** with PE = 1 as the first write only activates the interface (only PE is set).*
4	ENGC	*Enable General Call* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0). The 00h General Call address is acknowledged (01h ignored). 0: General Call disabled 1: General Call enabled *Note: In accordance with the I2C standard, when GCAL addressing is enabled, an I2C slave can only receive data. It will not transmit data to the master.*
3	START	*Generation of a Start condition* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0) or when the Start condition is sent (with interrupt generation if ITE = 1). **In Master mode** 0: No start generation 1: Repeated start generation **In Slave mode** 0: No start generation 1: Start generation when the bus is free
2	ACK	*Acknowledge enable* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0). 0: No acknowledge returned 1: Acknowledge returned after an address byte or a data byte is received

Table 84. SR2 register description (continued)

Bit	Name	Function
2	ARLO	*Arbitration lost* This bit is set by hardware when the interface loses the arbitration of the bus to another master. An interrupt is generated if ITE = 1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE = 0). After an ARLO event the interface switches back automatically to Slave mode (M/SL = 0). The SCL line is not held low while ARLO = 1. 0: No arbitration lost detected 1: Arbitration lost detected *Note: In a Multimaster environment, when the interface is configured in Master Receive mode it does not perform arbitration during the reception of the Acknowledge bit. Mishandling of the ARLO bit from the I2CSR2 register may occur when a second master simultaneously requests the same data from the same slave and the I^2C master does not acknowledge the data. The ARLO bit is then left at 0 instead of being set.*
1	BERR	*Bus error* This bit is set by hardware when the interface detects a misplaced Start or Stop condition. An interrupt is generated if ITE = 1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE = 0). The SCL line is not held low while BERR = 1. 0: No misplaced Start or Stop condition 1: Misplaced Start or Stop condition *Note: If a Bus Error occurs, a Stop or a repeated Start condition should be generated by the Master to re-synchronize communication, get the transmission acknowledged and the bus released for further communication.*
0	GCAL	*General Call (Slave mode)* This bit is set by hardware when a general call address is detected on the bus while ENGC = 1. It is cleared by hardware detecting a Stop condition (STOPF = 1) or when the interface is disabled (PE = 0). 0: No general call address detected on bus 1: General call address detected on bus

16.7.4 I^2C clock control register (CCR)

CCR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
FM/SM	CC[6:0]						
RW	RW						

Table 85. CCR register description

Bit	Name	Function
7	FM/SM	*Fast/Standard I^2C mode* This bit is set and cleared by software. It is not cleared when the interface is disabled (PE = 0). 0: Standard I^2C mode 1: Fast I^2C mode

Table 96. CPU addressing mode overview (continued)

Mode			Syntax	Destination	Pointer address (Hex.)	Pointer size (Hex.)	Length (bytes)
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC+/-127			+ 1
Relative	Indirect		jrne [$10]	PC+/-127	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

18.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Table 97. Inherent instructions

Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask (level 3)
RIM	Reset Interrupt Mask (level 0)
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

Figure 76. Application with a crystal or ceramic resonator for ROM (LQFP64 or any 48/60K ROM)

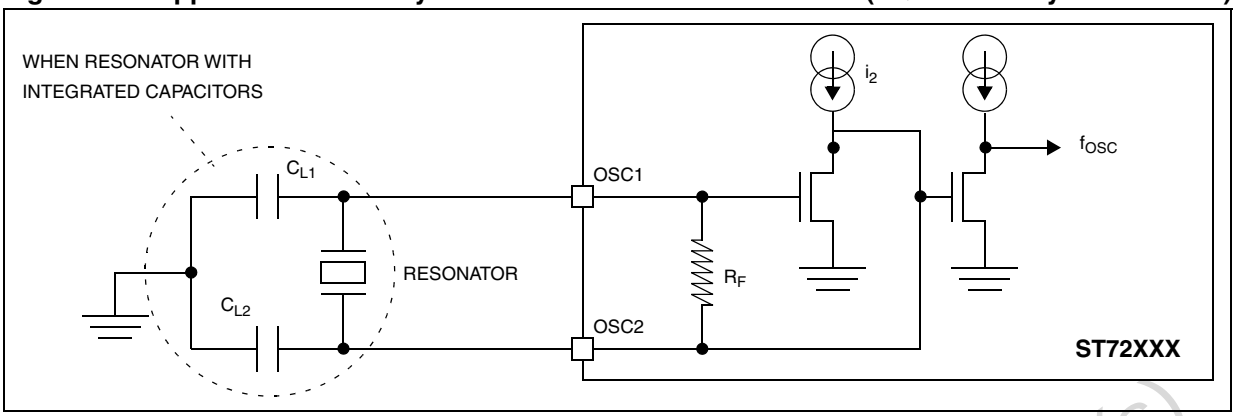


Table 116. OSCRANGE selection for typical resonators

Supplier	f_{OSC} (MHz)	Typical ceramic resonators[1]	
		Reference	Recommended OSCRANGE option bit configuration
Murata	2	CSTCC2M00G56A-R0	MP mode[2]
	4	CSTCR4M00G55B-R0	MS mode
	8	CSTCE8M00G55A-R0	HS mode
	16	CSTCE16M0G53A-R0	

1. Resonator characteristics given by the ceramic resonator manufacturer. For more information on these resonators, please consult www.murata.com.
2. LP mode is not recommended for 2 MHz resonator because the peak to peak amplitude is too small (> 0.8V).

19.5.4 RC oscillators

Table 117. RC oscillator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{OSC(RCINT)}$	Internal RC oscillator frequency (see *Figure 77*)	T_A = 25°C, V_{DD} = 5V	2	3.5	5.6	MHz

Figure 77. Typical $f_{OSC(RCINT)}$ versus T_A

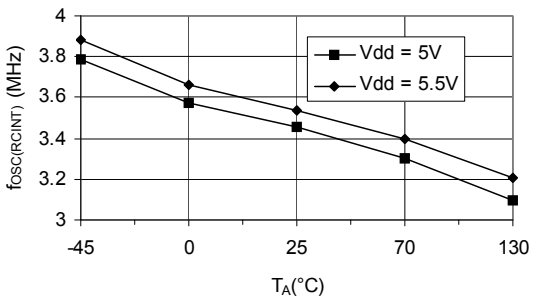
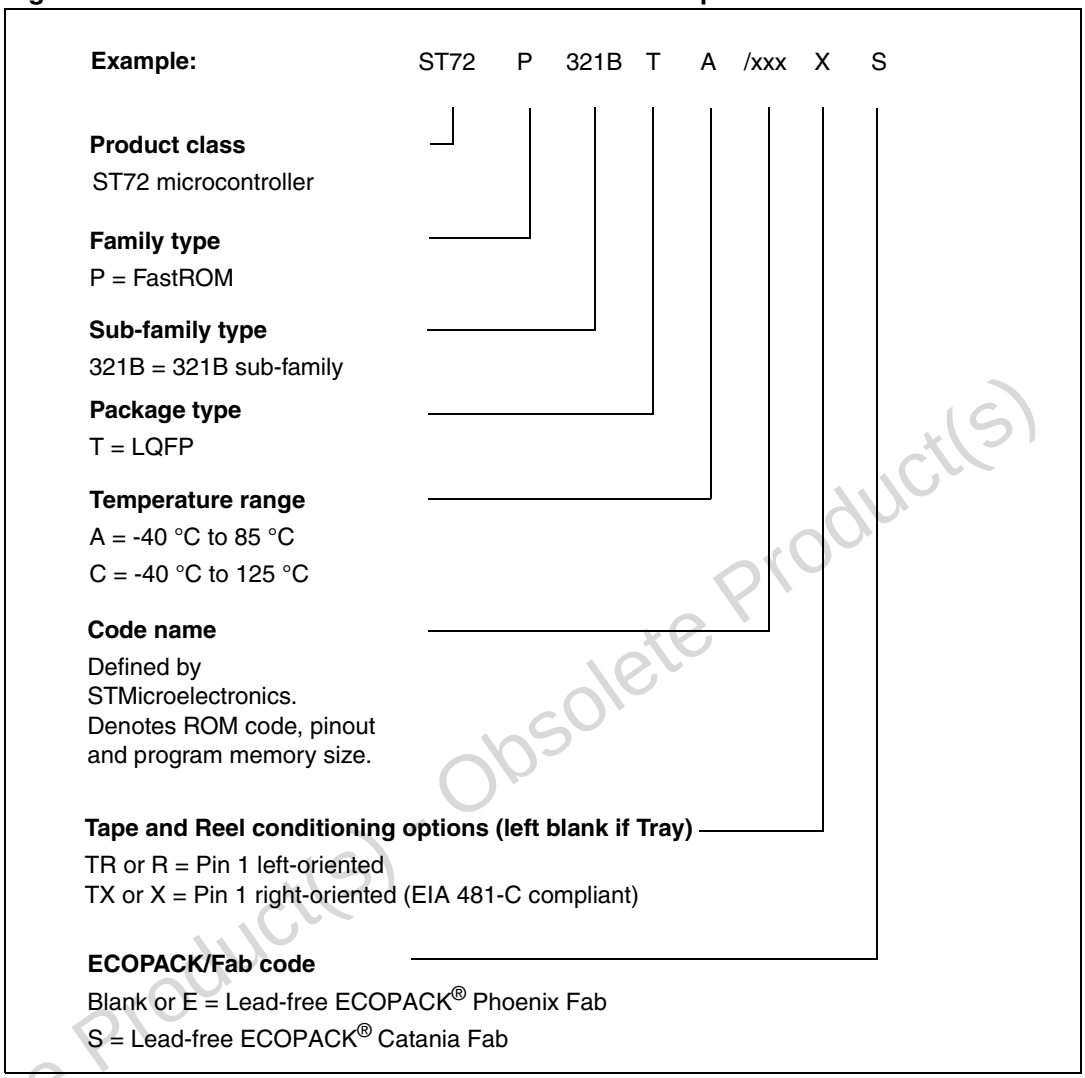


Figure 104. ST72P321Bxxx-Auto FastROM commercial product structure



```
.ext1_rt
; entry to interrupt routine
LD A,#00

LD sema,A

IRET
```

Case 2: Writing to PxOR or PxDDR with global interrupts disabled:

```
SIM
; set the interrupt mask
LD A,PFDR

AND A,#$02

LD  X,A
; store the level before writing to PxOR/PxDDR
LD A,#$90

LD PFDDR,A
; Write into PFDDR
LD A,#$ff

LD PFOR,A
          ; Write to PFOR
LD A,PFDR

AND A,#$02

LD Y,A
; store the level after writing to PxOR/PxDDR
LD A,X
; check for falling edge
cp A,#$02

jrne OUT

TNZ Y

jrne OUT

LD A,#$01

LD sema,A
; set the semaphore to '1' if edge is detected
RIM
 ; reset the interrupt mask
LD A,sema
; check the semaphore status
CP A,#$01
```