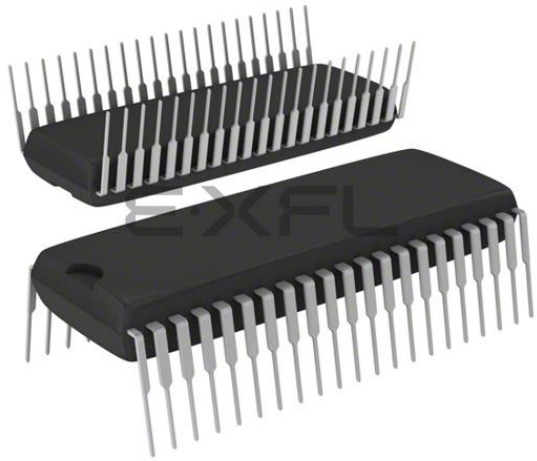


Welcome to E-XFL.COM

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	SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f324bj4b6

Figure 4. 32-pin LQFP package pinout

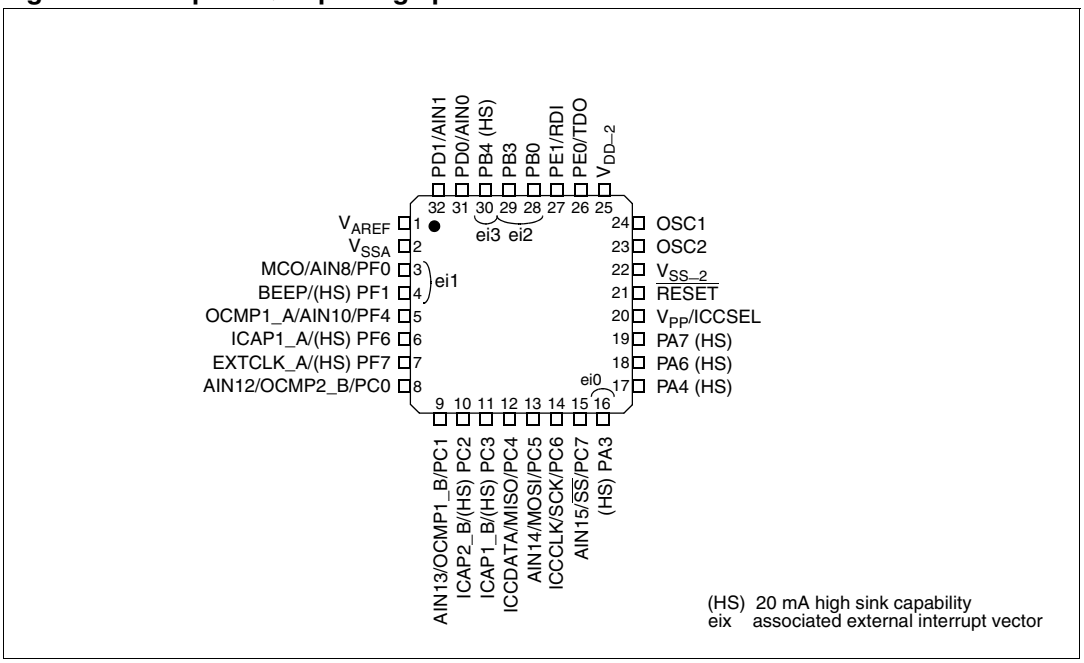
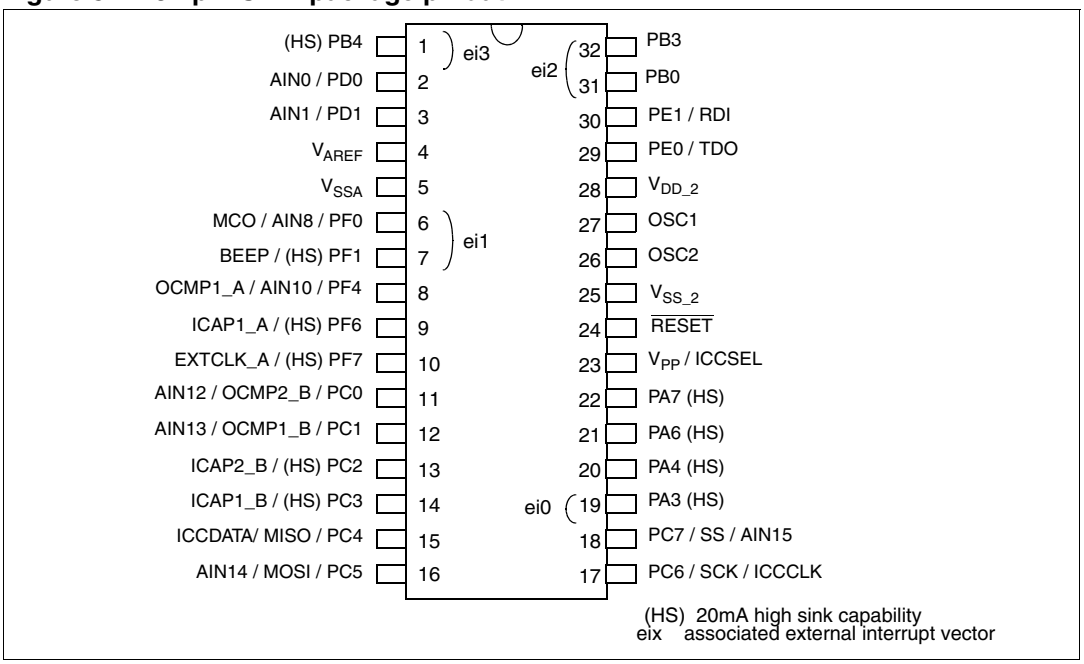


Figure 5. 32-pin SDIP package pinout



See *Section 12: Electrical characteristics on page 141* for external pin connection guidelines.

Refer to *Section 9: I/O ports on page 58* for more details on the software configuration of the I/O ports.

The reset configuration of each pin is shown in bold. This configuration is valid as long as the device is in reset state.

3 Register and memory map

As shown in *Figure 6*, the MCU is capable of addressing 64 Kbytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, up to 1024 bytes of RAM and up to 32 Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

The highest address bytes contain the user reset and interrupt vectors.

Caution: Never access memory locations marked as 'Reserved'. Accessing a reserved area can have unpredictable effects on the device.

Figure 6. Memory map

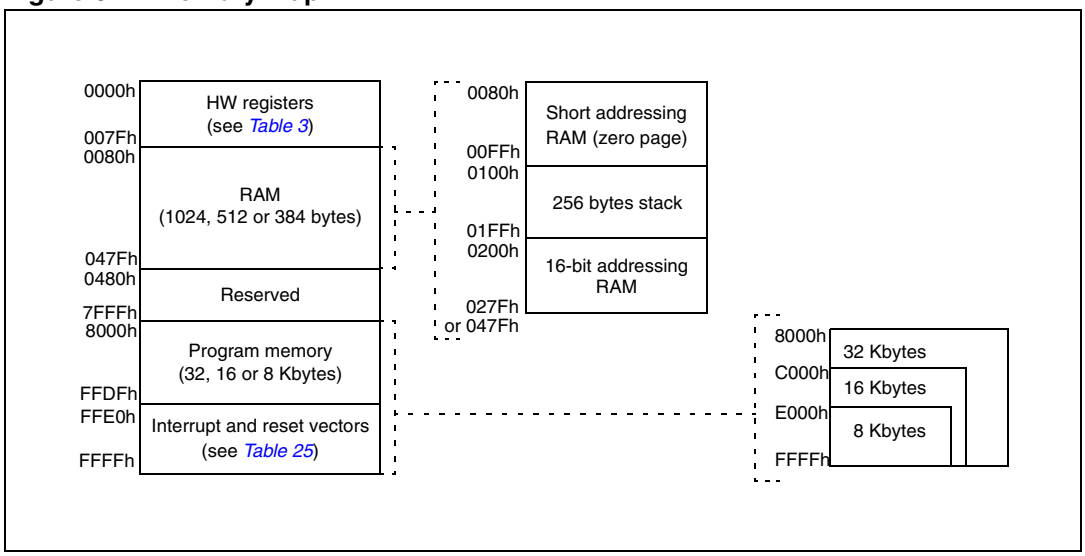


Table 3. Hardware register map

Address	Block	Register label	Register name	Reset status	Remarks
0000h 0001h 0002h	Port A(1)	PADR PADDR PAOR	Port A data register Port A data direction register Port A option register	00h(2) 00h 00h	R/W R/W R/W
0003h 0004h 0005h	Port B(1)	PBDR PBDDR PBOR	Port B data register Port B data direction register Port B option register	00h(2) 00h 00h	R/W R/W R/W
0006h 0007h 0008h	Port C	PCDR PCDDR PCOR	Port C data register Port C data direction register Port C option register	00h(2) 00h 00h	R/W R/W R/W
0009h 000Ah 000Bh	Port D(1)	PDADR PDDDR PDOR	Port D data register Port D data direction register Port D option register	00h(2) 00h 00h	R/W R/W R/W
000Ch 000Dh 000Eh	Port E(1)	PEDR PEDDR PEOR	Port E data register Port E data direction register Port E option register	00h(2) 00h 00h	R/W R/W(1) R/W(1)

Table 3. Hardware register map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
000Fh 0010h 0011h	Port F(1)	PFDR PFDDR PFOR	Port F data register Port F data direction register Port F option register	00h(2) 00h 00h	R/W R/W R/W
0012h to 0020h	Reserved area (15 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	SI	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)				
0031h 0032h 0033h 0034h 0035h 0036h 0037h 0038h 0039h 003Ah 003Bh 003Ch 003Dh 003Eh 003Fh	Timer A	TACR2 TACR1 TACSR TAIC1HR TAIC1LR TAOC1HR TAOC1LR TACHR TACLR TAACHR TAACLR TAIC2HR TAIC2LR TAOC2HR TAOC2LR	Timer A control register 2 Timer A control register 1 Timer A control/status register Timer A input capture 1 high register Timer A input capture 1 low register Timer A output compare 1 high register Timer A output compare 1 low register Timer A counter high register Timer A counter low register Timer A alternate counter high register Timer A alternate counter low register Timer A input capture 2 high register Timer A input capture 2 low register Timer A output compare 2 high register Timer A output compare 2 low register	00h 00h xxxx x0xxb xxh xxh 80h 00h FFh FCh FFh FCh xxh xxh 80h 00h	R/W R/W R/W Read only Read only R/W R/W Read only Read only Read only Read only Read only Read only R/W R/W
0040h	Reserved area (1 byte)				

7 Interrupts

7.1 Introduction

The ST7 enhanced interrupt management provides the following features:

- Hardware interrupts
- Software interrupt (TRAP)
- Nested or concurrent interrupt management with flexible interrupt priority and level management:
  - up to 4 software programmable nesting levels
  - up to 16 interrupt vectors fixed by hardware
  - 2 non-maskable events: reset, TRAP

This interrupt management is based on:

- Bit 5 and bit 3 of the CPU CC register (I1:0)
- Interrupt software priority registers (ISPRx)
- Fixed interrupt vector addresses located at the high addresses of the memory map (FFE0h to FFFFh) sorted by hardware priority order

This enhanced interrupt controller guarantees full upward compatibility with the standard (not nested) ST7 interrupt controller.

7.2 Masking and processing flow

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see *Table 14*). The processing flow is shown in *Figure 18.*

When an interrupt request has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to *Table 25: Interrupt mapping* for vector addresses).

The interrupt service routine should end with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I1 and I0 bits will be restored from the stack and the program in the previous level will resume.

peripheral control register. The general sequence for clearing an interrupt is based on an access to the status register followed by a read or write to an associated register.

Note: The clearing sequence resets the internal latch. A pending interrupt (that is, waiting to be serviced) is therefore lost if the clear sequence is executed.

7.3 Interrupts and low power modes

All interrupts allow the processor to exit the Wait low power mode. On the contrary, only external and other specified interrupts allow the processor to exit from the Halt modes (see column Exit from Halt in *Table 25: Interrupt mapping*). When several pending interrupts are present while exiting Halt mode, the first one serviced can only be an interrupt with Exit from Halt mode capability and it is selected through the same decision process shown in *Figure 19*.

Note: If an interrupt, that is not able to exit from Halt mode, is pending with the highest priority when exiting Halt mode, this interrupt is serviced after the first one serviced.

7.4 Concurrent and nested management

Figure 20 and *Figure 21* show two different interrupt management modes. The first is called concurrent mode and does not allow an interrupt to be interrupted, unlike the nested mode in *Figure 21*. The interrupt hardware priority is given in order from the lowest to the highest as follows: MAIN, IT4, IT3, IT2, IT1, IT0. Software priority is given for each interrupt.

Warning: A stack overflow may occur without notifying the software of the failure.

Figure 20. Concurrent interrupt management

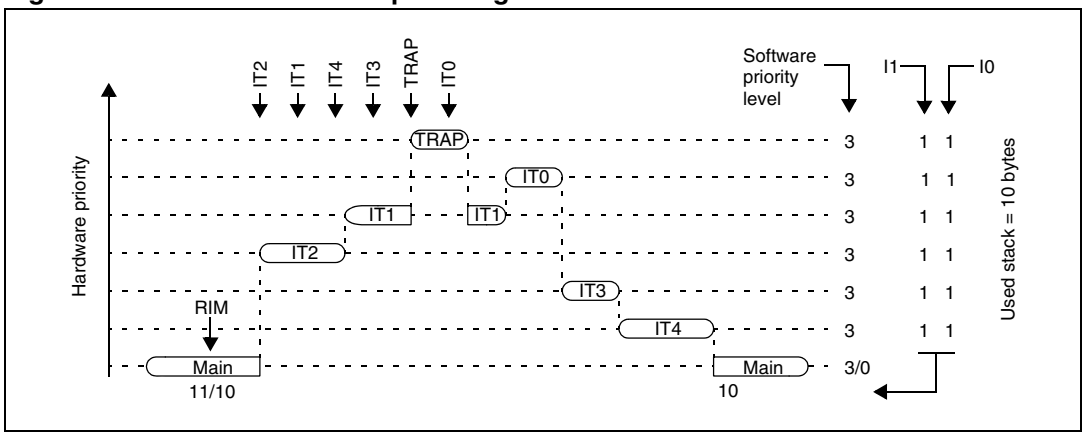
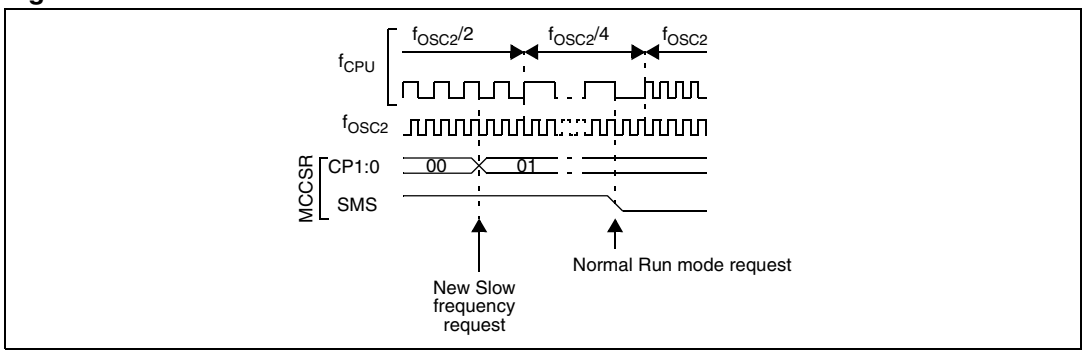


Table 18. Dedicated interrupt instruction set(1) (continued)

Instruction	New description	Function/example	I1	H	I0	N	Z	C
POP CC	POP CC from the Stack	Mem => CC	I1	H	I0	N	Z	C
RIM	Enable interrupt (level 0 set)	Load 10 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load 11 in I1:0 of CC	1		1			
TRAP	Software TRAP	Software NMI	1		1			
WFI	Wait for interrupt		1		0			

1. During the execution of an interrupt routine, the HALT, POP CC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

Figure 24. Slow mode clock transitions



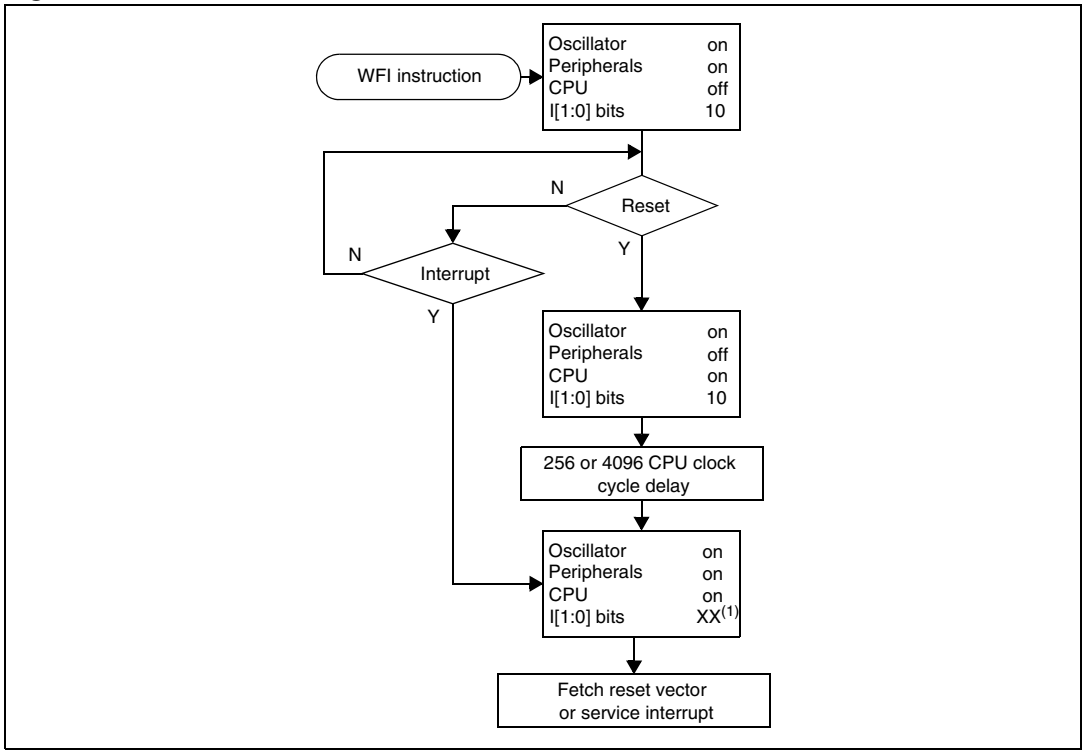
8.3 Wait mode

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 *Figure 25*.

Figure 25. Wait mode flowchart



1. 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.

9 I/O ports

9.1 Introduction

The I/O ports offer different functional modes:

- transfer of data through digital inputs and outputs,

and for specific pins:

- external interrupt generation,
- alternate signal input/output for the on-chip peripherals.

An I/O port contains up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

9.2 Functional description

Each port has two main registers:

- Data Register (DR)
- Data Direction Register (DDR)

and one optional register:

- Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR and OR registers: bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, (for specific ports which do not provide this register refer to *Section 9.3: I/O port implementation on page 62*). The generic I/O block diagram is shown in *Figure 30.*

9.2.1 Input modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Note: 1 Writing the DR register modifies the latch value but does not affect the pin status.

2 When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

3 Do not use read/modify/write instructions (BSET or BRES) to modify the DR register as this might corrupt the DR content for I/Os configured as input.

Table 39. MCCSR register description (continued)

Bit	Name	Function
6:5	CP[1:0]	CPU Clock Prescaler These bits select the CPU clock prescaler which is applied in different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software: 00: f_{CPU} in Slow mode = $f_{OSC2}/2$ 01: f_{CPU} in Slow mode = $f_{OSC2}/4$ 10: f_{CPU} in Slow mode = $f_{OSC2}/8$ 11: f_{CPU} in Slow mode = $f_{OSC2}/16$
4	SMS	Slow Mode Select This bit is set and cleared by software. 0: Normal mode. $f_{CPU} = f_{OSC2}$. 1: Slow mode. f_{CPU} is given by CP1, CP0. See *Section 8.2: Slow mode* and *Section 10.2: Main clock controller with real-time clock and beeper (MCC/RTC)* for more details.
3:2	TB[1:0]	Time Base control These bits select the programmable divider time base. They are set and cleared by software (see *Table 40*). A modification of the time base is taken into account at the end of the current period (previously set) to avoid an unwanted time shift. This allows to use this time base as a real-time clock.
1	OIE	Oscillator interrupt Enable This bit set and cleared by software. 0: Oscillator interrupt disabled 1: Oscillator interrupt enabled This interrupt can be used to exit from Active-halt mode. When this bit is set, calling the ST7 software HALT instruction enters the Active-halt power saving mode.
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	4 ms	2 ms	0	0
32000	8 ms	4 ms	0	1
80000	20 ms	10 ms	1	0
200000	50 ms	25 ms	1	1

The OC1R register value required for a specific timing application can be calculated using the following formula:

$$OCiR\ value = \frac{t * f_{CPU}}{PRESC} - 5$$

Where:

t = Pulse period (in seconds)
f_{CPU} = CPU clock frequency (in hertz)
PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits; see *Table 50*)

If the timer clock is an external clock the formula is:

$$OCiR = t * f_{EXT} - 5$$

Where:

t = Pulse period (in seconds)
f_{EXT} = External timer clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see *Figure 47*).

Note: 1 The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.

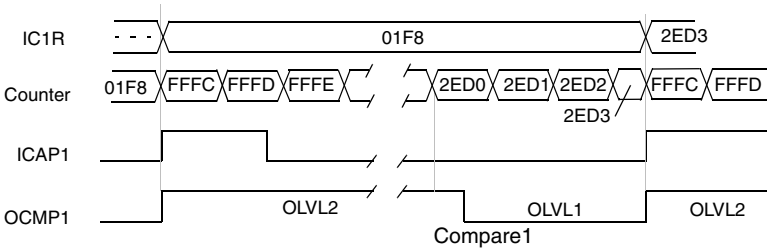
2 When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

3 If OLVL1 = OLVL2 a continuous signal will be seen on the OCMP1 pin.

4 The ICAP1 pin can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.

5 When one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

Figure 47. One Pulse mode timing example(1)



1. IEDG1 = 1, OC1R = 2ED0h, OLVL1 = 0, OLVL2 = 1

Input capture 1 low register (IC1LR)

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

IC1LR — Reset value: undefined

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

Output compare 1 high register (OC1HR)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

OC1HR — Reset value: 1000 0000 (80h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output compare 1 low register (OC1LR)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

OC1LR — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output compare 2 high register (OC2HR)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

OC2HR — Reset value: 1000 0000 (80h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output compare 2 low register (OC2LR)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

OC2LR — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Counter high register (CHR)

This is an 8-bit register that contains the high part of the counter value.

CHR — Reset value: 1111 1111 (FFh)

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

Counter low register (CLR)

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 accessing the CSR register clears the TOF bit.

CLR — Reset value: 1111 1100 (FCh)

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

Alternate counter high register (ACHR)

This is an 8-bit register that contains the high part of the counter value.

ACHR — Reset value: 1111 1111 (FFh)

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

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 58*).

As long as the SBK bit is set, the SCI send break frames to the TDO pin. After clearing this bit by software the SCI insert a logic 1 bit at the end of the last break frame to guarantee the recognition of the start bit of the next frame.

Idle characters

Setting the TE bit drives the SCI to send an idle frame before the first data frame.

Clearing and then setting the TE bit during a transmission sends an idle frame after the current word.

Note: Resetting and setting the TE bit causes the data in the TDR register to be lost. Therefore, the best time to toggle the TE bit is when the TDRE bit is set, that is, before writing the next byte in the SCIDR.

Receiver

The SCI can receive data words of either 8 or 9 bits. When the M bit is set, word length is 9 bits and the MSB is stored in the R8 bit in the SCICR1 register.

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, the SCIDR register consists or a buffer (RDR) between the internal bus and the received shift register (see *Figure 57*).

Procedure

1. Select the M bit to define the word length.
2. Select the desired baud rate using the SCIBRR and the SCIERPR registers.
3. Set the RE bit, this enables the receiver which begins searching for a start bit.

When a character is received:

- The RDRF bit is set. It indicates that the content of the shift register is transferred to the RDR.
- An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.
- The error flags can be set if a frame error, noise or an overrun error has been detected during reception.

Clearing the RDRF bit is performed by the following software sequence done by:

1. An access to the SCISR register

2. A read to the SCIDR register.

The RDRF bit must be cleared before the end of the reception of the next character to avoid an overrun error.

Break character

When a break character is received, the SCI handles it as a framing error.

Idle character

When a idle frame is detected, there is the same procedure as a data received character plus an interrupt if the ILIE bit is set and the I bit is cleared in the CCR register.

Table 69. SCI register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0050h	SCISR Reset value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
0051h	SCIDR Reset value	MSB x	x	x	x	x	x	x	LSB x
0052h	SCIBRR Reset value	SCP1 0	SCP0 0	SCT2 0	SCT1 0	SCT0 0	SCR2 0	SCR1 0	SCR0 0
0053h	SCICR1 Reset value	R8 x	T8 0	SCID 0	M 0	WAKE 0	PCE 0	PS 0	PIE 0
0054h	SCICR2 Reset value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0
0055h	SCIERPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	SCIPETPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0

12.4 LVD/AVD characteristics

12.4.1 Operating conditions with LVD

Subject to general operating conditions for T_A.

Table 87. Operating conditions with LVD

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)	VD level = high in option byte	4.0(1)	4.2	4.5	V
		VD level = med. in option byte(2)	3.55(1)	3.75	4.0(1)	
		VD level = low in option byte(2)	2.95(1)	3.15	3.35(1)	
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)	VD level = high in option byte	3.8	4.0	4.25(1)	
		VD level = med. in option byte(2)	3.35(1)	3.55	3.75(1)	
		VD level = low in option byte(2)	2.8(1)	3.0	3.15(1)	
$V_{hys(LVD)}$	LVD voltage threshold hysteresis(1)	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$	150	200	250	mV
Vt_{POR}	V_{DD} rise time(1)	Flash devices	6µs/V		100ms/V	
		8/16 Kbyte ROM devices			20ms/V	
		32 Kbyte ROM devices			∝ ms/V	
$t_{g(VDD)}$	Filtered glitch delay on V_{DD}(1)	Not detected by the LVD			40	ns

1. Data based on characterization results, tested in production for ROM devices only.
2. If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range.

12.4.2 Auxiliary voltage detector (AVD) thresholds

Subject to general operating conditions for T_A.

Table 88. AVD thresholds

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(AVD)}$	1 ⇒0 AVDF flag toggle threshold (V_{DD} rise)	VD level = high in option byte	4.4(1)	4.6	4.9	V
		VD level = med. in option byte	3.95(1)	4.15	4.4(1)	
		VD level = low in option byte	3.4(1)	3.6	3.8(1)	
$V_{IT-(AVD)}$	0 ⇒1 AVDF flag toggle threshold (V_{DD} fall)	VD level = high in option byte	4.2	4.4	4.65(1)	
		VD level = med. in option byte	3.75(1)	4.0	4.2(1)	
		VD level = low in option byte	3.2(1)	3.4	3.6(1)	
$V_{hys(AVD)}$	AVD voltage threshold hysteresis	$V_{IT+(AVD)}$-$V_{IT-(AVD)}$		200		mV
ΔV_{IT-}	Voltage drop between AVD flag set and LVD reset activated	$V_{IT-(AVD)}$-$V_{IT-(LVD)}$		450		

1. Data based on characterization results, tested in production for ROM devices only.

12.5 Supply current characteristics

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To obtain the total device consumption, the two current values must be added (except for Halt mode for which the clock is stopped).

12.5.1 ROM current consumption

Table 89. ROM current consumption

Symbol	Parameter	Conditions	32 Kbyte ROM devices		16/8 Kbyte ROM devices		Unit
			Typ	Max(1)	Typ	Max(1)	
I_{DD}	Supply current in Run mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 1 MHz f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz	0.55 1.10 2.20 4.38	0.87 1.75 3.5 7.0	0.46 0.93 1.9 3.7	0.69 1.4 2.7 5.5	mA
	Supply current in Slow mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 62.5 kHz f_{OSC} = 4 MHz, f_{CPU} = 125 kHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz	53 100 194 380	87 175 350 700	30 70 150 310	60 120 250 500	µA
	Supply current in Wait mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 1 MHz f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz	0.31 0.61 1.22 2.44	0.5 1.0 2.0 4.0	0.22 0.45 0.91 1.82	0.37 0.75 1.5 3	mA
	Supply current in Slow Wait mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 62.5 kHz f_{OSC} = 4 MHz, f_{CPU} = 125 kHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz	36 69 133 260	63 125 250 500	20 40 90 190	40 90 180 350	µA
	Supply current in Halt mode(3)	-40 °C ≤ T_A ≤ +85 °C	<1	10	<1	10	
		-40 °C ≤ T_A ≤ +125 °C	<1	50	<1	50	
	Supply current in Active-halt mode(4)	f_{OSC} = 2 MHz f_{OSC} = 4 MHz f_{OSC} = 8 MHz f_{OSC} = 16 MHz	15 28 55 107	20 38 75 200	11 22 43 85	15 30 60 150	

1. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
2. Measurements are done in the following conditions:
   - Program executed from RAM, CPU running with RAM access. The increase in consumption when executing from Flash is 50%.
   - All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load)
   - All peripherals in reset state
   - LVD disabled.
   - Clock input (OSC1) driven by external square wave
   - In Slow and Slow Wait modes, f_{CPU} is based on f_{OSC} divided by 32

   To obtain the total current consumption of the device, add the clock source (*Section 12.6.3*) and the peripheral power consumption (*Section 12.5.4*).
3. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or V_{SS} (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
4. Data based on characterization results, not tested in production. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or V_{SS} (no load); clock input (OSC1) driven by external square wave, LVD disabled. To obtain the total current consumption of the device, add the clock source consumption (*Section 12.6.3*).

Figure 67. Typical application with a crystal or ceramic resonator (32 Kbyte Flash and ROM devices)

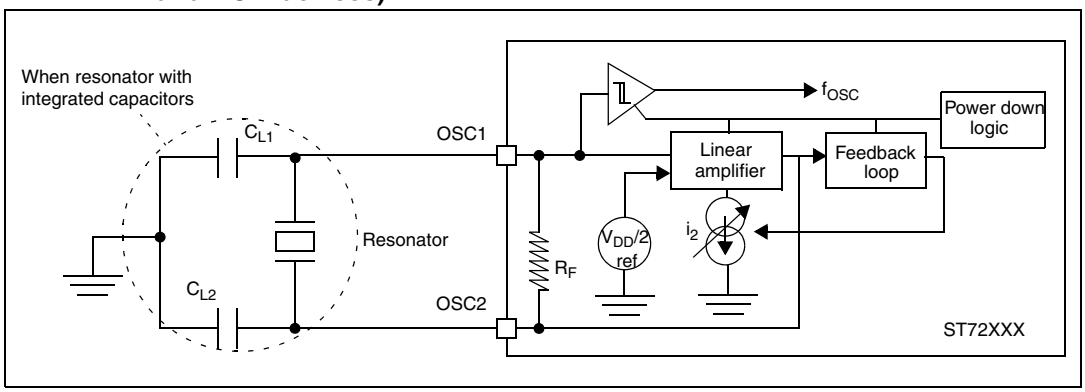


Table 97. OSCRANGE selection for typical resonators

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	CSTCE8M00G52A-R0	HS mode
	16	CSTCE16M0V51A-R0	HS mode

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

12.6.4 RC oscillators

Table 98. RC oscillators

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{OSC\ (RCINT)}$	Internal RC oscillator frequency (see *Figure 68)*	T_A = 25 °C, V_{DD} = 5 V	2	3.5	5.6	MHz

Figure 68. Typical $f_{OSC(RCINT)}$ vs T_A

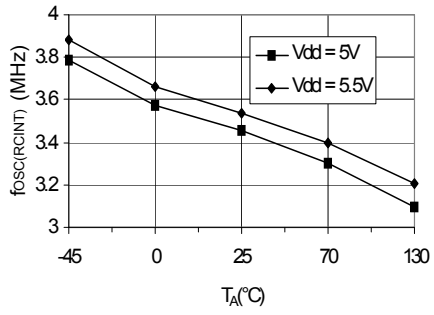


Figure 72. Typical V_{OL} at V_{DD} = 5 V (standard ports)

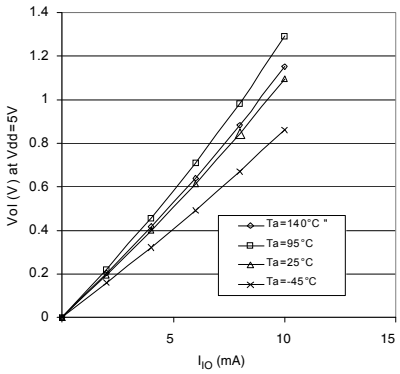


Figure 73. Typical V_{OL} at V_{DD} = 5 V (high-sink ports)

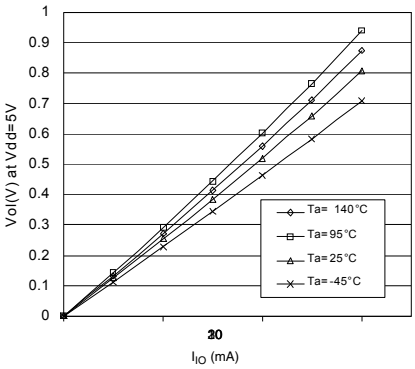
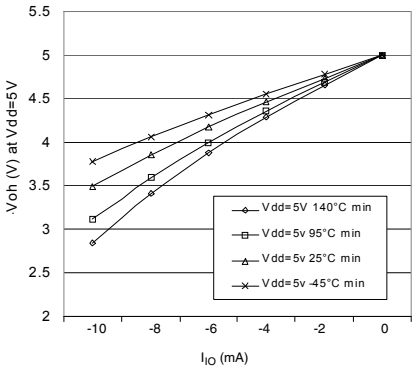


Figure 74. Typical V_{OH} at V_{DD} = 5 V



$\overline{\text{RESET}}$ pin protection when LVD is enabled

When the LVD is enabled, it is recommended to protect the $\overline{\text{RESET}}$ pin as shown in *Figure 78* and follow these guidelines:

1. The reset network protects the device against parasitic resets.
2. The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
3. Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in *Section 12.10.1*. Otherwise the reset will not be taken into account internally.
4. Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin (by an external pull-up for example) is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in *Section 12.2.2 on page 143*.
5. When the LVD is enabled, it is mandatory not to connect a pull-up resistor. A 10nF pull-down capacitor is recommended to filter noise on the reset line.
6. In case a capacitive power supply is used, it is recommended to connect a 1M ohm pull-down resistor to the $\overline{\text{RESET}}$ pin to discharge any residual voltage induced by this capacitive power supply (this will add 5 µA to the power consumption of the MCU).

Tips when using the LVD:

- Check that all recommendations related to reset circuit have been applied (see section above)
- Check that the power supply is properly decoupled (100 nF + 10 µF close to the MCU). Refer to AN1709. If this cannot be done, it is recommended to put a 100 nF + 1 M Ohm pull-down on the $\overline{\text{RESET}}$ pin.
- The capacitors connected on the $\overline{\text{RESET}}$ pin and also the power supply are key to avoiding any start-up marginality. In most cases, steps 1 and 2 above are sufficient for a robust solution. Otherwise: Replace 10 nF pull-down on the $\overline{\text{RESET}}$ pin with a 5 µF to 20 µF capacitor.

Figure 78. $\overline{\text{RESET}}$ pin protection when LVD is enabled

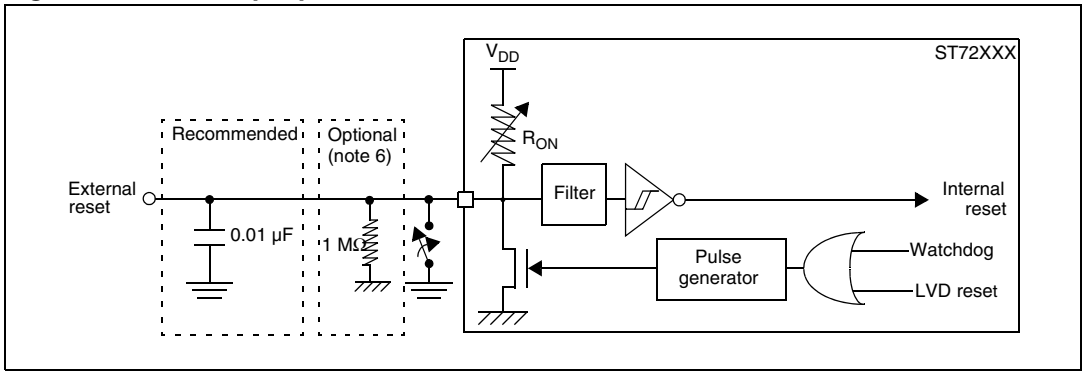


Table 121. Option byte 1 bit description (continued)

Bit	Name	Function
OPT0	PLL OFF	PLL activation This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL must not be used with the internal RC oscillator. The PLL is guaranteed only with an input frequency between 2 and 4 MHz. 0: PLL x2 enabled 1: PLL x2 disabled **Caution**: The PLL can be enabled only if the “OSCRANGE” (OPT3:1) bits are configured to “MP - 2~4 MHz”. Otherwise, the device functionality is not guaranteed.

Table 122. Package selection (OPT7)

Version	Selected package	PKG1
J	LQFP44/SDIP42	1
K	LQFP32/SDIP32	0

14.2 ROM devices

14.2.1 Transfer of customer code

Customer code is made up of the ROM/FASTROM contents and the list of the selected options (if any). The ROM/FASTROM contents are to be sent with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh. Complete the appended *ST72324Bxx MICROCONTROLLER OPTION LIST on page 182* to communicate the selected options to STMicroelectronics.

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred.

Figure 93: ST72324Bxx ordering information scheme on page 178 serves as a guide for ordering. The STMicroelectronics sales organization will be pleased to provide detailed information on contractual points.

Caution: The readout protection binary value is inverted between ROM and Flash products. The option byte checksum differs between ROM and Flash.