

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	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 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/st72f324bj6b6

Contents

1	Description	14
2	Pin description	15
3	Register and memory map	20
4	Flash program memory	23
4.1	Introduction	23
4.2	Main features	23
4.3	Structure	23
4.3.1	Readout protection	24
4.4	ICC interface	25
4.5	ICP (in-circuit programming)	26
4.6	IAP (in-application programming)	26
4.7	Related documentation	26
4.7.1	Flash Control/Status Register (FCSR)	26
5	Central processing unit (CPU)	27
5.1	Introduction	27
5.2	Main features	27
5.3	CPU registers	27
5.3.1	Accumulator (A)	28
5.3.2	Index registers (X and Y)	28
5.3.3	Program counter (PC)	28
5.3.4	Condition Code register (CC)	28
5.3.5	Stack Pointer register (SP)	30
6	Supply, reset and clock management	31
6.1	Introduction	31
6.2	PLL (phase locked loop)	31
6.3	Multi-oscillator (MO)	32
6.3.1	External clock source	32
6.3.2	Crystal/ceramic oscillators	33

6.3.3	Internal RC oscillator	33
6.4	Reset sequence manager (RSM)	34
6.4.1	Asynchronous external RESET pin	34
6.5	System integrity management (SI)	36
6.5.1	LVD (low voltage detector)	36
6.5.2	AVD (auxiliary voltage detector)	37
6.5.3	Low power modes	38
6.5.4	Interrupts	38
6.6	SI registers	39
6.6.1	System integrity (SI) control/status register (SICSR)	39
7	Interrupts	41
7.1	Introduction	41
7.2	Masking and processing flow	41
7.2.1	Servicing pending interrupts	42
7.2.2	Different interrupt vector sources	43
7.2.3	Non-maskable sources	43
7.2.4	Maskable sources	43
7.3	Interrupts and low power modes	44
7.4	Concurrent and nested management	44
7.5	Interrupt registers	45
7.5.1	CPU CC register interrupt bits	45
7.5.2	Interrupt software priority registers (ISPRx)	46
7.6	External interrupts	48
7.6.1	I/O port interrupt sensitivity	48
7.6.2	External interrupt control register (EICR)	49
8	Power saving modes	52
8.1	Introduction	52
8.2	Slow mode	52
8.3	Wait mode	53
8.4	Active-halt and Halt modes	54
8.4.1	Active-halt mode	54
8.4.2	Halt mode	55
9	I/O ports	58

10.3.7	16-bit timer registers	89
10.4	Serial peripheral interface (SPI)	96
10.4.1	Introduction	96
10.4.2	Main features	96
10.4.3	General description	97
10.4.4	Clock phase and clock polarity	101
10.4.5	Error flags	102
10.4.6	Low power modes	104
10.4.7	Interrupts	104
10.4.8	SPI registers	104
10.5	Serial communications interface (SCI)	108
10.5.1	Introduction	108
10.5.2	Main features	108
10.5.3	General description	109
10.5.4	Functional description	111
10.5.5	Low power modes	120
10.5.6	Interrupts	120
10.5.7	SCI registers	120
10.6	10-bit A/D converter (ADC)	128
10.6.1	Introduction	128
10.6.2	Main features	128
10.6.3	Functional description	129
10.6.4	Low power modes	130
10.6.5	Interrupts	130
10.6.6	ADC registers	130
11	Instruction set	133
11.1	CPU addressing modes	133
11.1.1	Inherent	134
11.1.2	Immediate	134
11.1.3	Direct	135
11.1.4	Indexed (no offset, short, long)	135
11.1.5	Indirect (short, long)	135
11.1.6	Indirect indexed (short, long)	136
11.1.7	Relative mode (direct, indirect)	137
11.2	Instruction groups	137

12.10	Control pin characteristics	162
12.10.1	Asynchronous RESET pin	162
12.10.2	ICCSEL/VPP pin	165
12.11	Timer peripheral characteristics	165
12.11.1	16-bit timer	165
12.12	Communication interface characteristics	166
12.12.1	Serial peripheral interface (SPI)	166
12.13	10-bit ADC characteristics	168
12.13.1	Analog power supply and reference pins	170
12.13.2	General PCB design guidelines	170
12.13.3	ADC accuracy	171
13	Package characteristics	172
13.1	ECOPACK	172
13.2	Package mechanical data	172
13.2.1	LQFP44 package mechanical data	172
13.2.2	SDIP42 package mechanical data	173
13.2.3	LQFP32 package mechanical data	174
13.2.4	SDIP32 package mechanical data	175
13.3	Thermal characteristics	177
14	Device configuration and ordering information	178
14.1	Flash devices	179
14.1.1	Flash configuration	179
14.2	ROM devices	181
14.2.1	Transfer of customer code	181
14.3	Development tools	183
14.3.1	Introduction	183
14.3.2	Evaluation tools and starter kits	183
14.3.3	Development and debugging tools	183
14.3.4	Programming tools	183
14.3.5	Socket and emulator adapter information	184
14.4	ST7 Application notes	184
15	Known limitations	185
15.1	All Flash and ROM devices	185

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 4](#)). Each of these sectors can be erased independently to avoid unnecessary erasing of the whole Flash memory when only a partial erasing is required.

The first two sectors have a fixed size of 4 Kbytes (see [Figure 7](#)). 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).

Table 4. Sectors available in Flash devices

Flash size	Available sectors
4 Kbytes	Sector 0
8 Kbytes	Sectors 0, 1
>8 Kbytes	Sectors 0, 1, 2

Table 6. Arithmetic management bits (continued)

Bit	Name	Function
1	Z	Zero (Arithmetic Management bit) This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero. 0: The result of the last operation is different from zero. 1: The result of the last operation is zero. This bit is accessed by the JREQ and JRNE test instructions.
0	C	Carry/borrow This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation. 0: No overflow or underflow has occurred. 1: An overflow or underflow has occurred. This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the 'bit test and branch', shift and rotate instructions.

Table 7. Software interrupt bits

Bit	Name	Function
5	I1	Software Interrupt Priority 1 The combination of the I1 and I0 bits determines the current interrupt software priority (see Table 8).
3	I0	Software Interrupt Priority 0 The combination of the I1 and I0 bits determines the current interrupt software priority (see Table 8).

Table 8. Interrupt software priority selection

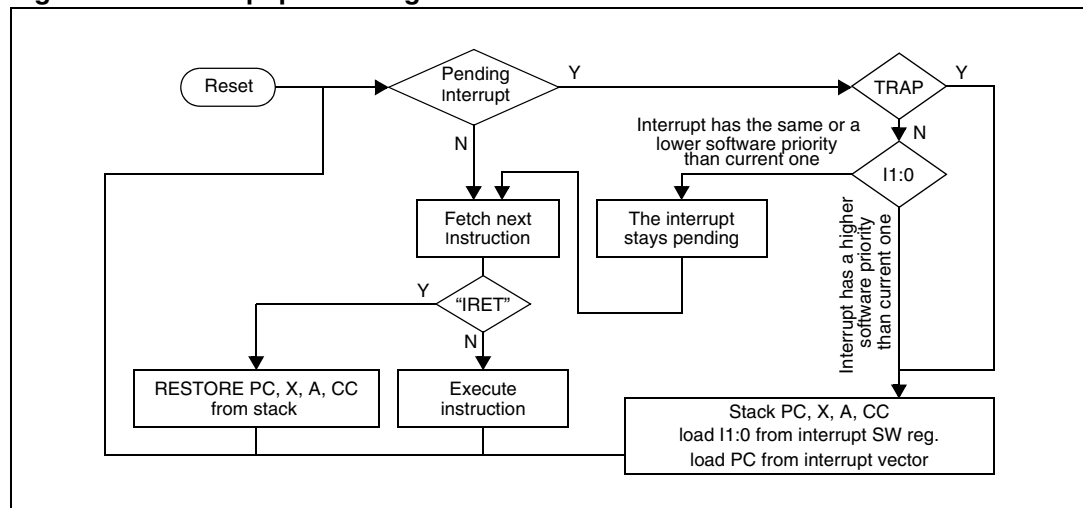
Interrupt software priority	Level	I1	I0
Level 0 (main)	<div style="text-align: center;"> Low ↓ High </div>	1	0
Level 1		0	1
Level 2		0	0
Level 3 (= interrupt disable)		1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (ISPRx). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See [Section 7: Interrupts on page 41](#) for more details.

Table 14. Interrupt software priority levels

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

Figure 18. Interrupt processing flowchart

7.2.1 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 19 describes this decision process.

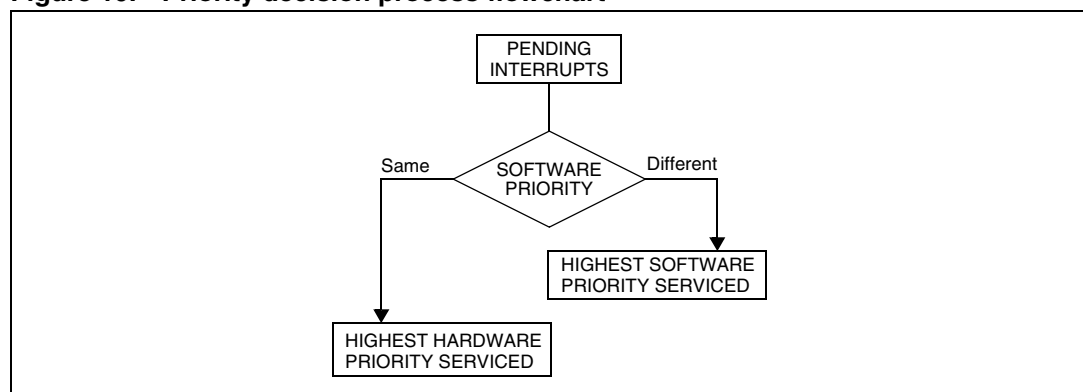
Figure 19. Priority decision process flowchart

Table 21. Interrupt sensitivity - ei3

IS11	IS10	External interrupt sensitivity
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Table 22. Interrupt sensitivity - ei0

IS21	IS20	External interrupt sensitivity	
		IPA bit = 0	IPA bit = 1
0	0	Falling edge and low level	Rising edge and high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

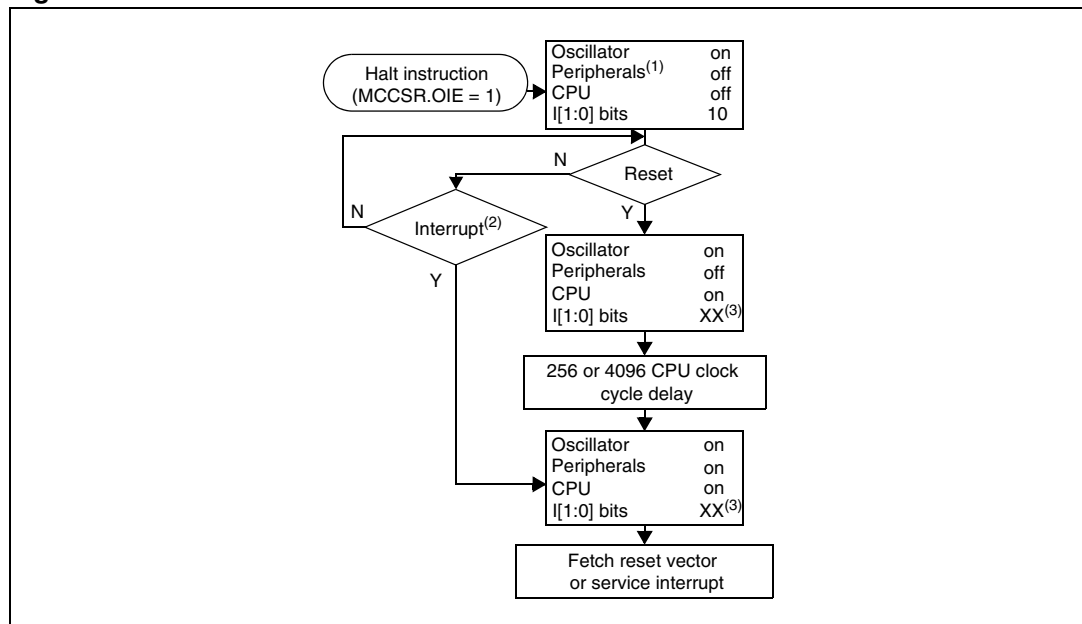
Table 23. Interrupt sensitivity - ei1

IS21	IS20	External interrupt sensitivity
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Table 24. Nested interrupts register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0024h	ISPR0 reset value	ei1		ei0		MCC + SI			
		I1_3 1	I0_3 1	I1_2 1	I0_2 1	I1_1 1	I0_1 1	1	1
0025h	ISPR1 reset value	SPI				ei3		ei2	
		I1_7 1	I0_7 1	I1_6 1	I0_6 1	I1_5 1	I0_5 1	I1_4 1	I0_4 1
0026h	ISPR2 reset value	AVD		SCI		Timer B		Timer A	
		I1_11 1	I0_11 1	I1_10 1	I0_10 1	I1_9 1	I0_9 1	I1_8 1	I0_8 1
0027h	ISPR3 reset value	1	1	1	1	I1_13 1	I0_13 1	I1_12 1	I0_12 1
0028h	EICR reset value	IS11 0	IS10 0	IPB 0	IS21 0	IS20 0	IPA 0	0	0

Figure 27. Active-halt mode flowchart



1. Peripheral clocked with an external clock source can still be active.
2. Only the MCC/RTC interrupt and some specific interrupts can exit the MCU from Active-halt mode (such as external interrupt). Refer to [Table 25: Interrupt mapping on page 51](#) for more details.
3. 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 restored when the CC register is popped.

8.4.2 Halt mode

The Halt mode is the lowest power consumption mode of the MCU. It is entered by executing the 'HALT' instruction when the OIE bit of the Main Clock Controller Status register (MCCSR) is cleared (see [Section 10.2: Main clock controller with real-time clock and beeper \(MCC/RTC\) on page 69](#) for more details on the MCCSR register).

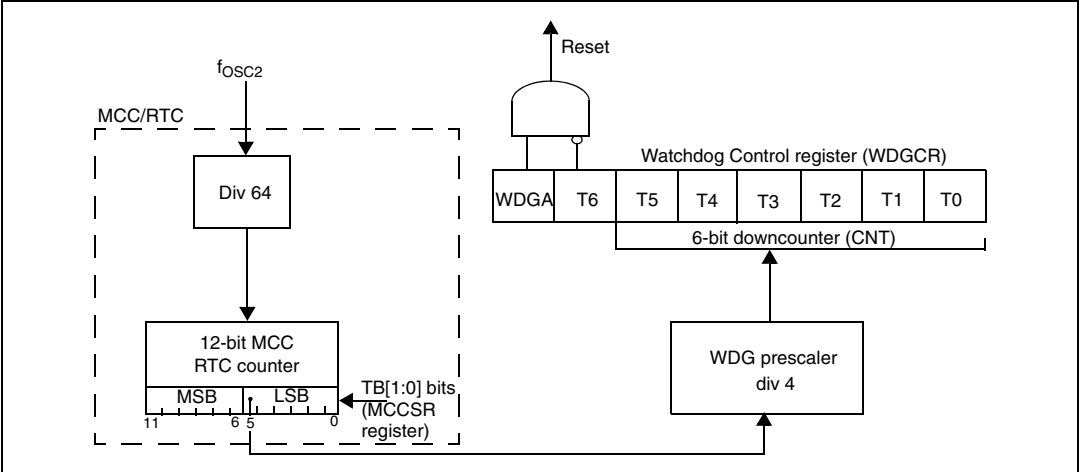
The MCU can exit Halt mode on reception of either a specific interrupt (see [Table 25: Interrupt mapping](#)) or a reset. When exiting Halt mode by means of a reset or an interrupt, the oscillator is immediately turned on and the 256 or 4096 CPU cycle delay is used to stabilize the oscillator. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see [Figure 29](#)).

When entering Halt mode, the I[1:0] bits in the CC register are forced to '10b' to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In Halt mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

The compatibility of Watchdog operation with Halt mode is configured by the "WDGHALT" option bit of the option byte. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog reset (see [Section 14.1 on page 179](#) for more details).

Figure 32. Watchdog block diagram



10.1.4 How to program the Watchdog timeout

Figure 33 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 34.

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

Figure 33. Approximate timeout duration

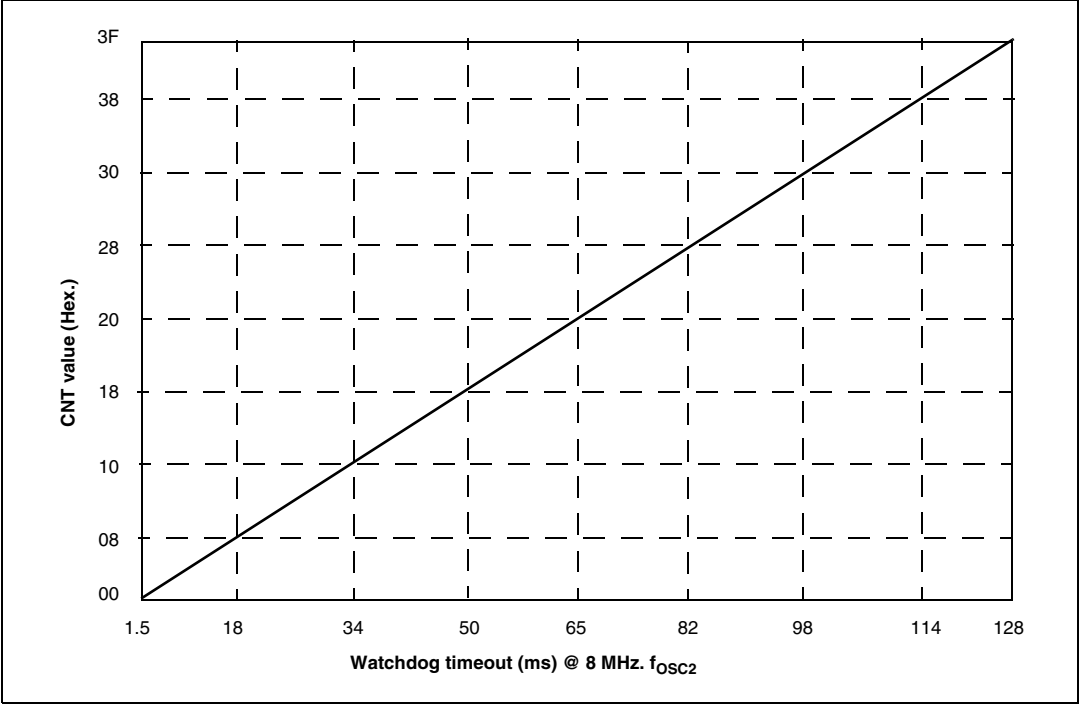


Table 39. MCCR register description (continued)

Bit	Name	Function
6:5	CP[1:0]	<p>CPU Clock Prescaler</p> <p>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:</p> <p>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 = $f_{OSC2}/16$</p>
4	SMS	<p>Slow Mode Select</p> <p>This bit is set and cleared by software.</p> <p>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.</p>
3:2	TB[1:0]	<p>Time Base control</p> <p>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.</p>
1	OIE	<p>Oscillator interrupt Enable</p> <p>This bit set and cleared by software.</p> <p>0: Oscillator interrupt disabled 1: Oscillator interrupt enabled</p> <p>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</p>
0	OIF	<p>Oscillator interrupt Flag</p> <p>This bit is set by hardware and cleared by software reading the MCCR register. It indicates when set that the main oscillator has reached the selected elapsed time (TB1:0).</p> <p>0: Timeout not reached 1: Timeout reached</p> <p>Caution: The BRES and BSET instructions must not be used on the MCCR register to avoid unintentionally clearing the OIF bit.</p>

Table 40. Time base selection

Counter prescaler	Time base		TB1	TB0
	$f_{OSC2} = 4 \text{ MHz}$	$f_{OSC2} = 8 \text{ MHz}$		
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:

$$\text{OC1R value} = \frac{t \cdot f_{\text{CPU}}}{\text{PRESC}} - 5$$

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:

$$\text{OC1R} = t \cdot f_{\text{EXT}} - 5$$

Where:

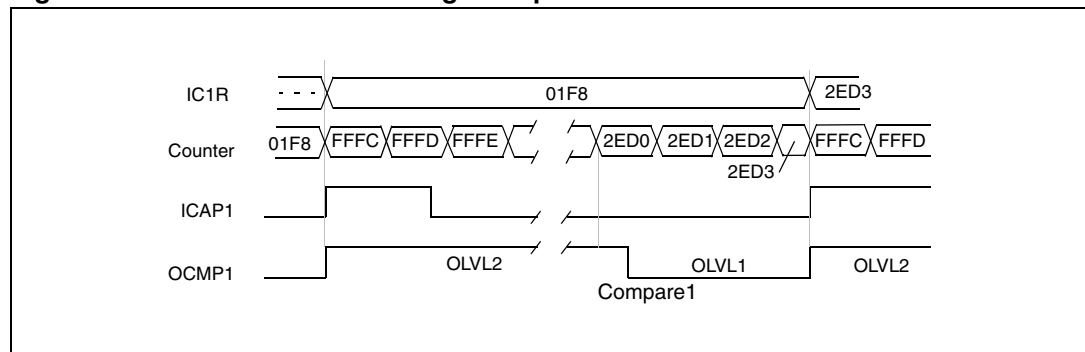
t = Pulse period (in seconds)

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

10.3.6 Summary of timer modes

Table 48. Summary of timer modes

Mode	Timer resources			
	Input Capture 1	Input Capture 2	Output Compare 1	Output Compare 2
Input Capture (1 and/or 2)	Yes	Yes	Yes	Yes
Output Compare (1 and/or 2)				
One Pulse mode	No	Not recommended ⁽¹⁾	No	Partially ⁽²⁾
PWM mode		Not recommended ⁽³⁾		No

1. See note 4 in [One Pulse mode on page 84](#).

2. See note 5 in [One Pulse mode on page 84](#).

3. See note 4 in [Pulse Width Modulation mode on page 86](#).

10.3.7 16-bit timer registers

Each timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

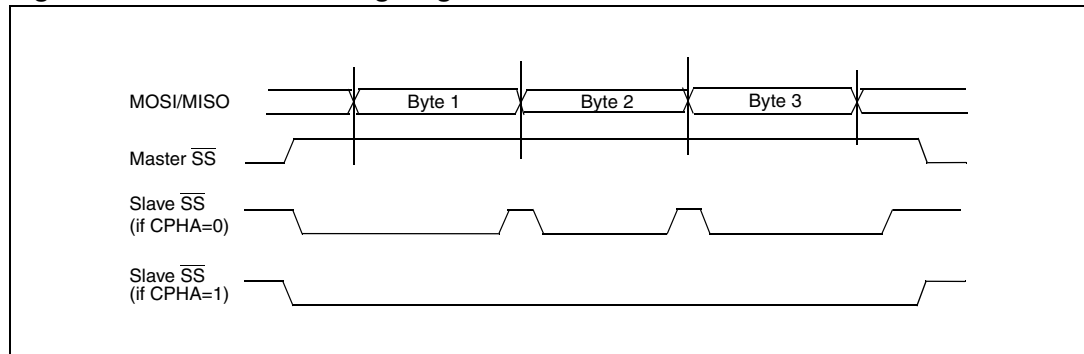
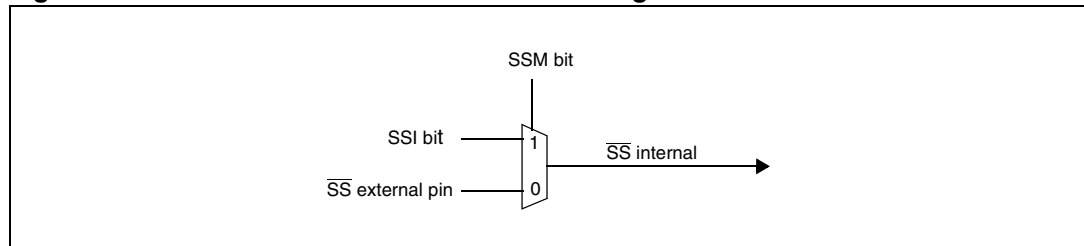
Control Register 1 (CR1)

CR1 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 49. CR1 register description

Bit	Name	Function
7	ICIE	Input Capture Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.
6	OCIE	Output Compare Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.
5	TOIE	Timer Overflow Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Figure 52. Generic \overline{SS} timing diagram**Figure 53. Hardware/software slave select management****Master mode operation**

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

Note: *The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).*

How to operate the SPI in master mode

To operate the SPI in master mode, perform the following steps in order:

1. Write to the SPICR register:
 - Select the clock frequency by configuring the SPR[2:0] bits.
 - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits.
Figure 54 shows the four possible configurations.
 - Note: The slave must have the same CPOL and CPHA settings as the master.*
2. Write to the SPICSR register:
 - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the \overline{SS} pin high for the complete byte transmit sequence.
3. Write to the SPICR register:
 - Set the MSTR and SPE bits.
 - Note: MSTR and SPE bits remain set only if \overline{SS} is high.*

Caution: If the SPICSR register is not written first, the SPICR register setting (MSTR bit) might not be taken into account.

The transmit sequence begins when software writes a byte in the SPIDR register.

10.4.6 Low power modes

Table 53. Effect of low power modes on SPI

Mode	Description
Wait	No effect on SPI. SPI interrupt events cause the device to exit from Wait mode.
Halt	SPI registers are frozen. In Halt mode, the SPI is inactive. SPI operation resumes when the MCU is woken up by an interrupt with Exit from Halt mode capability. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetching). If several data are received before the wakeup event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the device.

Using the SPI to wake up the MCU from Halt mode

In slave configuration, the SPI is able to wake up the ST7 device from Halt mode through a SPIF interrupt. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetch). If multiple data transfers have been performed before software clears the SPIF bit, then the OVR bit is set by hardware.

Note: When waking up from Halt mode, if the SPI remains in Slave mode, it is recommended to perform an extra communications cycle to bring the SPI from Halt mode state to normal state. If the SPI exits from Slave mode, it returns to normal state immediately.

Caution: The SPI can wake up the ST7 from Halt mode only if the Slave Select signal (external \overline{SS} pin or the SSI bit in the SPICSR register) is low when the ST7 enters Halt mode. Therefore, if Slave selection is configured as external (see [Slave select management on page 98](#)), make sure the master drives a low level on the SS pin when the slave enters Halt mode.

10.4.7 Interrupts

Table 54. SPI interrupt control/wakeup capability⁽¹⁾

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
SPI end of transfer event	SPIF	SPIE	Yes	Yes
Master mode fault event	MODF			No
Overrun error	OVR			

1. The SPI interrupt events are connected to the same interrupt vector (see [Section 7: Interrupts](#)). They generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

10.4.8 SPI registers

SPI 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]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- Parity control
 - Transmits parity bit
 - Checks parity of received data byte
- Reduced power consumption mode

10.5.3 General description

The interface is externally connected to another device by two pins (see [Figure 58](#)):

- TDO: Transmit Data Output. When the transmitter and the receiver are disabled, the output pin returns to its I/O port configuration. When the transmitter and/or the receiver are enabled and nothing is to be transmitted, the TDO pin is at high level.
- RDI: Receive Data Input is the serial data input. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

Through these pins, serial data is transmitted and received as frames comprising:

- an Idle Line prior to transmission or reception
- a start bit
- a data word (8 or 9 bits) least significant bit first
- a Stop bit indicating that the frame is complete

This interface uses two types of baud rate generator:

- a conventional type for commonly-used baud rates
- an extended type with a prescaler offering a very wide range of baud rates even with non-standard oscillator frequencies

10.6 10-bit A/D converter (ADC)

10.6.1 Introduction

The on-chip analog-to-digital converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

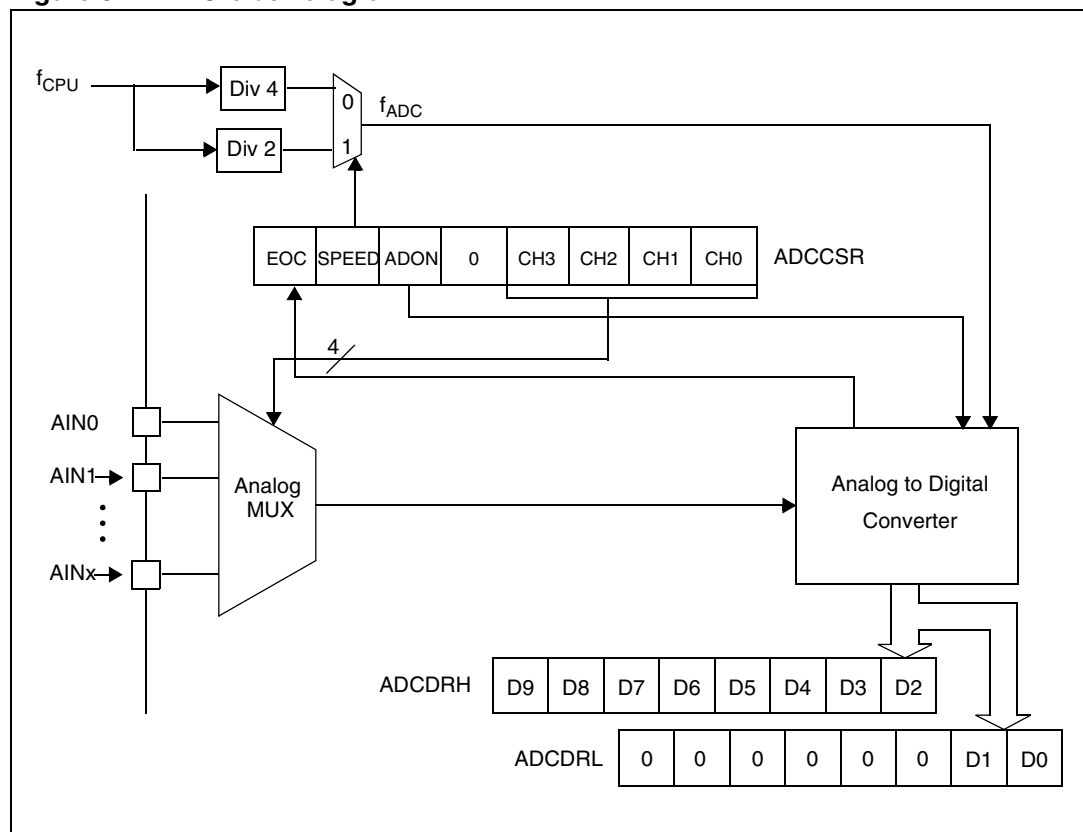
The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

10.6.2 Main features

- 10-bit conversion
- Up to 16 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in [Figure 61](#).

Figure 61. ADC block diagram



12.2.2 Current characteristics

Table 84. Current characteristics

Symbol	Ratings	Max value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) ⁽¹⁾	32-pin devices	75
		44-pin devices	150
I_{VSS}	Total current out of V_{SS} ground lines (sink) ⁽¹⁾	32-pin devices	75
		44-pin devices	150
I_{IO}	Output current sunk by any standard I/O and control pin	20	mA
	Output current sunk by any high sink I/O pin	40	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}^{(2)(3)}$	Injected current on V_{PP} pin	± 5	
	Injected current on $RESET$ pin	± 5	
	Injected current on OSC1 and OSC2 pins	± 5	
	Injected current on ROM and 32 Kbyte Flash devices PB0 pin	± 5	
	Injected current on 8/16 Kbyte Flash devices PB0 pin	+ 5	
	Injected current on any other pin ⁽⁴⁾⁽⁵⁾	± 5	
$\Sigma I_{INJ(PIN)}^{(2)}$	Total injected current (sum of all I/O and control pins) ⁽⁴⁾	± 25	

1. All power (V_{DD}) and ground (V_{SS}) lines must always be connected to the external supply.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding V_{IN} maximum must always be respected.
3. Negative injection degrades the analog performance of the device. See note in [Section 12.13.3: ADC accuracy on page 171](#). If the current injection limits given in [Table 106: General characteristics on page 158](#) are exceeded, general device malfunction may result.
4. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterization with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.
5. True open drain I/O port pins do not accept positive injection.

12.2.3 Thermal characteristics

Table 85. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see Section 13.3: Thermal characteristics)		

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
V_{tPOR}	V_{DD} rise time ⁽¹⁾	Flash devices	6 μ s/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 \Rightarrow 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 \Rightarrow 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.

Table 123. STMicroelectronics development tools

Supported products	Emulation				Programming
	ST7 DVP3 series		ST7 EMU3 series		ICC socket board
	Emulator	Connection kit	Emulator	Active probe and TEB	
ST72324BJ, ST72F324BJ	ST7MDT20-DVP3	ST7MDT20-T44/DVP	ST7MDT20J-EMU3	ST7MDT20J-TEB	ST7SB20J/xx ⁽¹⁾
ST72324BK, ST72F324BK		ST7MDT20-T32/DVP			

1. Add suffix /EU, /UK, /US for the power supply of your region.

14.3.5 Socket and emulator adapter information

For information on the type of socket that is supplied with the emulator, refer to the suggested list of sockets in [Table 124](#).

Note: *Before designing the board layout, it is recommended to check the overall dimensions of the socket as they may be greater than the dimensions of the device.*

For footprint and other mechanical information about these sockets and adapters, refer to the manufacturer's datasheet (www.yamaichi.de for LQFP44 10x10 and www.ironwoodelectronics.com for LQFP32 7x7).

Table 124. Suggested list of socket types

Device	Socket (supplied with ST7MDT20J-EMU3)	Emulator adapter (supplied with ST7MDT20J-EMU3)
LQFP32 7X7	IRONWOOD SF-QFE32SA-L-01	IRONWOOD SK-UGA06/32A-01
LQFP44 10X10	YAMAICHI IC149-044-*52-*5	YAMAICHI ICP-044-5

14.4 ST7 Application notes

All relevant ST7 application notes can be found on www.st.com.