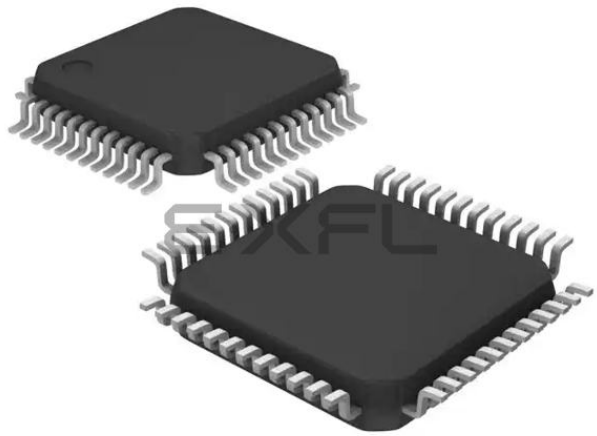


Welcome to E-XFL.COM

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, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f345c4t6tr

Figure 3. LQFP44 package pinout

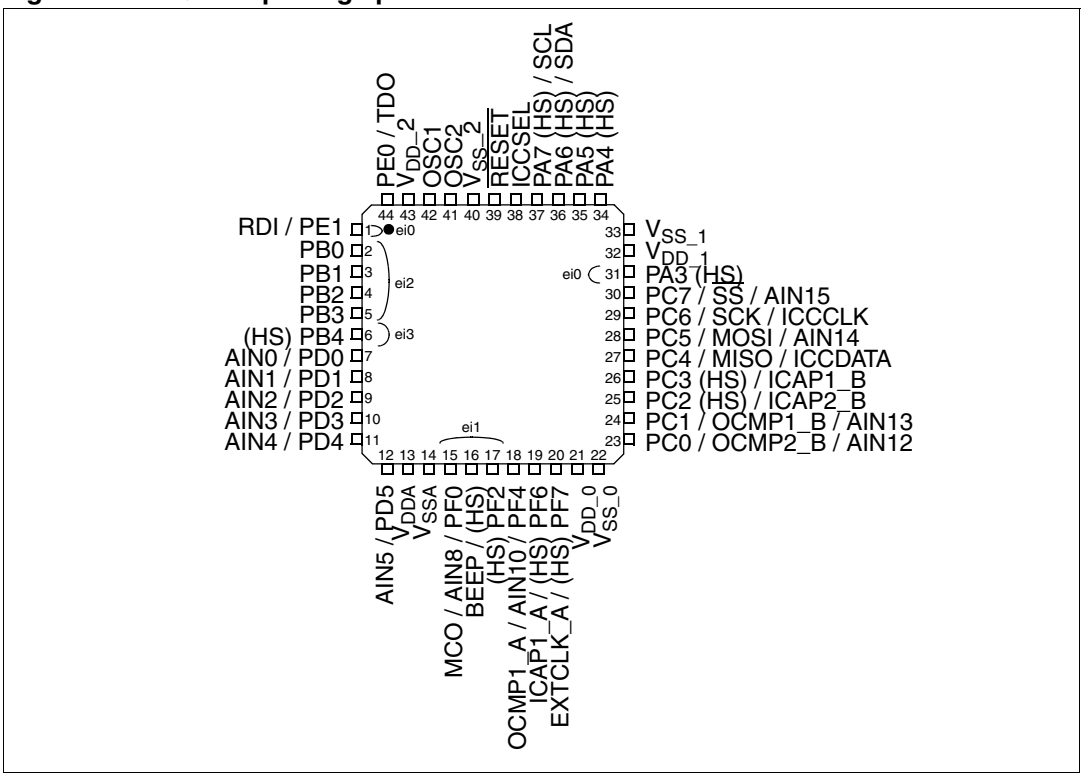


Table 4. Hardware register map (continued)

Address	Block	Register label	Register name	Reset status (1)	Remarks (2)
0031h	TIMER A	TACR2	Timer A Control Register 2	00h	R/W
0032h		TACR1	Timer A Control Register 1	00h	R/W
0033h		TACSR	Timer A Control/Status Register	xxh	R/W
0034h		TAIC1HR	Timer A Input Capture 1 High Register	xxh	Read Only
0035h		TAIC1LR	Timer A Input Capture 1 Low Register	xxh	Read Only
0036h		TAOC1HR	Timer A Output Compare 1 High Register	80h	R/W
0037h		TAOC1LR	Timer A Output Compare 1 Low Register	00h	R/W
0038h		TACHR	Timer A Counter High Register	FFh	Read Only
0039h		TACLR	Timer A Counter Low Register	FCh	Read Only
003Ah		TAACHR	Timer A Alternate Counter High Register	FFh	Read Only
003Bh		TAACLR	Timer A Alternate Counter Low Register	FCh	Read Only
003Ch		TAIC2HR	Timer A Input Capture 2 High Register	xxh	Read Only
003Dh		TAIC2LR	Timer A Input Capture 2 Low Register	xxh	Read Only
003Eh		TAOC2HR	Timer A Output Compare 2 High Register	80h	R/W
003Fh		TAOC2LR	Timer A Output Compare 2 Low Register	00h	R/W
0040h	Reserved area (1 Byte)				
0041h	TIMER B	TBCR2	Timer B Control Register 2	00h	R/W
0042h		TBCR1	Timer B Control Register 1	00h	R/W
0043h		TBCSR	Timer B Control/Status Register	xxh	R/W
0044h		TBIC1HR	Timer B Input Capture 1 High Register	xxh	Read Only
0045h		TBIC1LR	Timer B Input Capture 1 Low Register	xxh	Read Only
0046h		TBOC1HR	Timer B Output Compare 1 High Register	80h	R/W
0047h		TBOC1LR	Timer B Output Compare 1 Low Register	00h	R/W
0048h		TBCHR	Timer B Counter High Register	FFh	Read Only
0049h		TBCLR	Timer B Counter Low Register	FCh	Read Only
004Ah		TBACHR	Timer B Alternate Counter High Register	FFh	Read Only
004Bh		TBACLR	Timer B Alternate Counter Low Register	FCh	Read Only
004Ch		TBIC2HR	Timer B Input Capture 2 High Register	xxh	Read Only
004Dh		TBIC2LR	Timer B Input Capture 2 Low Register	xxh	Read Only
004Eh		TBOC2HR	Timer B Output Compare 2 High Register	80h	R/W
004Fh		TBOC2LR	Timer B Output Compare 2 Low Register	00h	R/W
0050h	SCI	SCISR	SCI Status Register	C0h	Read Only
0051h		SCIDR	SCI Data Register	xxh	R/W
0052h		SCIBRR	SCI Baud Rate Register	00h	R/W
0053h		SCICR1	SCI Control Register 1	x000 0000b	R/W
0054h		SCICR2	SCI Control Register 2	00h	R/W
0055h			Reserved area	--	
0056h		SCIERPR	SCI Extended Receive Prescaler Register	00h	R/W
0057h		SCIETPR	SCI Extended Transmit Prescaler Register	00h	R/W
0058h	I^2C	I2CCR	I^2C Control Register	00h	R/W
0059h		I2CSR1	I^2C Status Register 1	00h	Read Only
005Ah		I2CSR2	I^2C Status Register 2	00h	Read Only
005Bh		I2CCCR	I^2C Clock Control Register	00h	R/W
005Ch		I2COAR1	I^2C Own Address Register 1	00h	R/W
005Dh		I2COAR2	I^2C Own Address Register2	40h	R/W
005Eh		I2CDR	I^2C Data Register	00h	R/W
005Fh	Reserved area (1 byte)				

4.3.1 In-circuit programming (ICP)

ICP uses a protocol called ICC (in-circuit communication) which allows an ST7 plugged on a printed circuit board (PCB) to communicate with an external programming device connected via cable. ICP is performed in three steps:

Switch the ST7 to ICC mode (in-circuit communications). This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the $\overline{\text{RESET}}$ pin is pulled low. When the ST7 enters ICC mode, it fetches a specific reset vector which points to the ST7 System Memory containing the ICC protocol routine. This routine enables the ST7 to receive bytes from the ICC interface.

- Download ICP Driver code in RAM from the ICCDATA pin
- Execute ICP Driver code in RAM to program the Flash memory

Depending on the ICP Driver code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection of the serial communication interface for downloading).

4.3.2 In-application programming (IAP)

This mode uses an IAP Driver program previously programmed in Sector 0 by the user (in ICP mode).

This mode is fully controlled by user software. This allows it to be adapted to the user application (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored etc.)
IAP mode can be used to program any memory areas except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

4.4 ICC interface

ICP needs a minimum of 4 and up to 7 pins to be connected to the programming tool. These pins are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- ICCSEL: ICC selection
- OSC1: main clock input for external source (not required on devices without OSC1/OSC2 pins)
- V_{DD}: application board power supply (optional, see Note 3)

Note: 1 If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.

2 During the ICP session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5 mA at high level (push pull output or pull-up resistor<1,000). A schottky diode can be used

Read operation (E2LAT = 0)

The EEPROM can be read as a normal ROM location when the E2LAT bit of the EECSR register is cleared.

On this device, Data EEPROM can also be used to execute machine code. Take care not to write to the Data EEPROM while executing from it. This would result in an unexpected code being executed.

Write operation (E2LAT = 1)

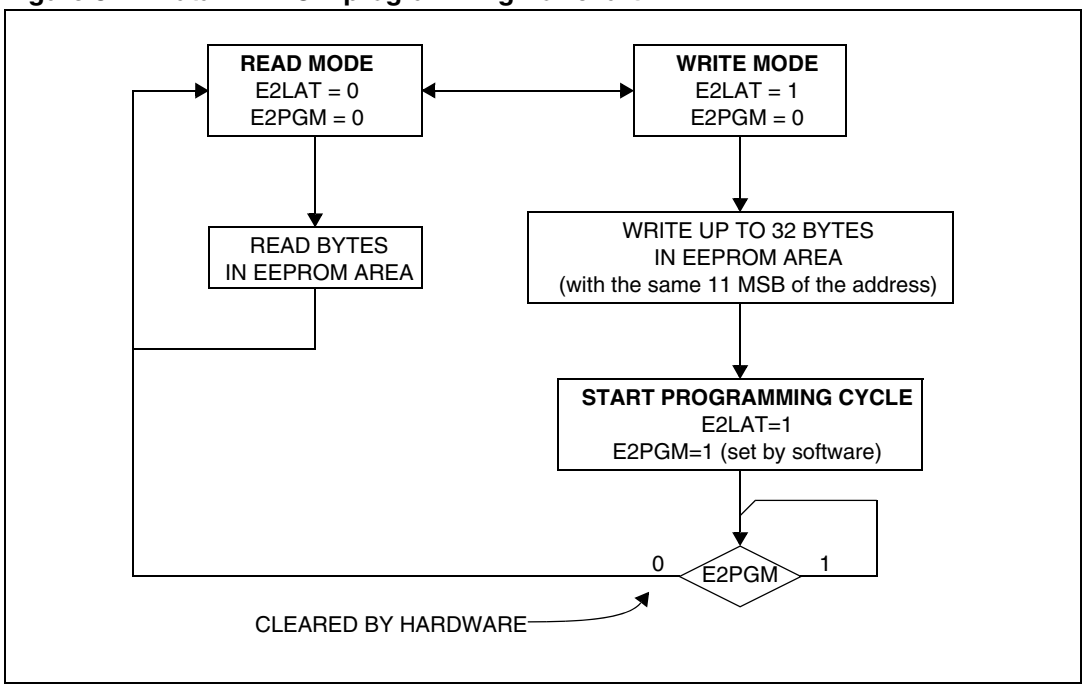
To access the write mode, the E2LAT bit has to be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 32 data latches according to its address.

When E2PGM bit is set by the software, all the previous bytes written in the data latches (up to 32) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must take care that all the bytes written between two programming sequences have the same high address: only the five Least Significant Bits of the address can change.

The programming cycle is fully completed when the E2PGM bit is cleared.

Note: Care should be taken during the programming cycle. Writing to the same memory location will over-program the memory (logical AND between the two write access data results) because the data latches are only cleared at the end of the programming cycle and by the falling edge of the E2LAT bit.
It is not possible to read the latched data.
This note is illustrated by the Figure 10.

Figure 8. Data EEPROM programming flowchart



5.7 Register description

5.7.1 EEPROM control/status register (EECSR)

Reset value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	E2LAT	E2PGM
Read/Write							

Bits 7:2 = Reserved, forced by hardware to 0.

Bit 1 = **E2LAT** *Latch Access Transfer*

This bit is set by software. It is cleared by hardware at the end of the programming cycle. It can only be cleared by software if the E2PGM bit is cleared.
0: Read mode
1: Write mode

Bit 0 = **E2PGM** *Programming control and status*

This bit is set by software to begin the programming cycle. At the end of the programming cycle, this bit is cleared by hardware.
0: Programming finished or not yet started
1: Programming cycle is in progress

Note: If the E2PGM bit is cleared during the programming cycle, the memory data is not guaranteed.

Table 5. Data EEPROM register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0020h	**EECSR** Reset value	0	0	0	0	0	0	E2LAT 0	E2PGM 0

Application notes

The LVDRF flag is not cleared when another reset type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure.
In this case, a watchdog reset can be detected by software while an external reset can not.

Caution: When the LVD is not activated with the associated option byte, the WDGRF flag can not be used in the application.

8.5.2 Interrupt software priority registers (ISPRX)

Reset value: 1111 1111 (FFh)

	7							0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
ISPR3	1	1	1	1	I1_13	I0_13	I1_12	I0_12
Read/Write (bits 7:4 of **ISPR3** are read only)								

These four registers contain the interrupt software priority of each interrupt vector.

- Each interrupt vector (except reset and TRAP) has corresponding bits in these registers where its own software priority is stored. This correspondence is shown in the following table.

Table 15. Interrupt vector addresses

Vector address	ISPRx bits
FFFBh-FFFAh	I1_0 and I0_0 bits*
FFF9h-FFF8h	I1_1 and I0_1 bits
...	...
FFE1h-FFE0h	I1_13 and I0_13 bits

- Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.
- Level 0 can not be written (I1_x=1, I0_x=0). In this case, the previously stored value is kept. (example: previous = CFh, write = 64h, result = 44h)

The reset, and TRAP vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

Caution: If the I1_x and I0_x bits are modified while the interrupt x is executed, the following behavior has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

Table 16. Dedicated interrupt instruction set

Instruction	New description	Function/example	I1	H	I0	N	Z	C
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC	I1	H	I0	N	Z	C
JRM	Jump if I1:0=11 (level 3)	I1:0=11 ?						
JRNM	Jump if I1:0<>11	I1:0<>11 ?						
POP CC	Pop CC from the stack	Mem => CC	I1	H	I0	N	Z	C

8.6 External interrupts

8.6.1 I/O port interrupt sensitivity

The external interrupt sensitivity is controlled by the IPA, IPB and ISxx bits of the EICR register (*Figure 24*). This control allows to have up to 4 fully independent external interrupt source sensitivities.

Each external interrupt source can be generated on four (or five) different events on the pin:

- Falling edge
- Rising edge
- Falling and rising edge
- Falling edge and low level
- Rising edge and high level (only for ei0 and ei2)

To guarantee correct functionality, the sensitivity bits in the EICR register can be modified only when the I1 and I0 bits of the CC register are both set to 1 (level 3). This means that interrupts must be disabled before changing sensitivity.

The pending interrupts are cleared by writing a different value in the ISx[1:0], IPA or IPB bits of the EICR.

Figure 24. External interrupt control bits

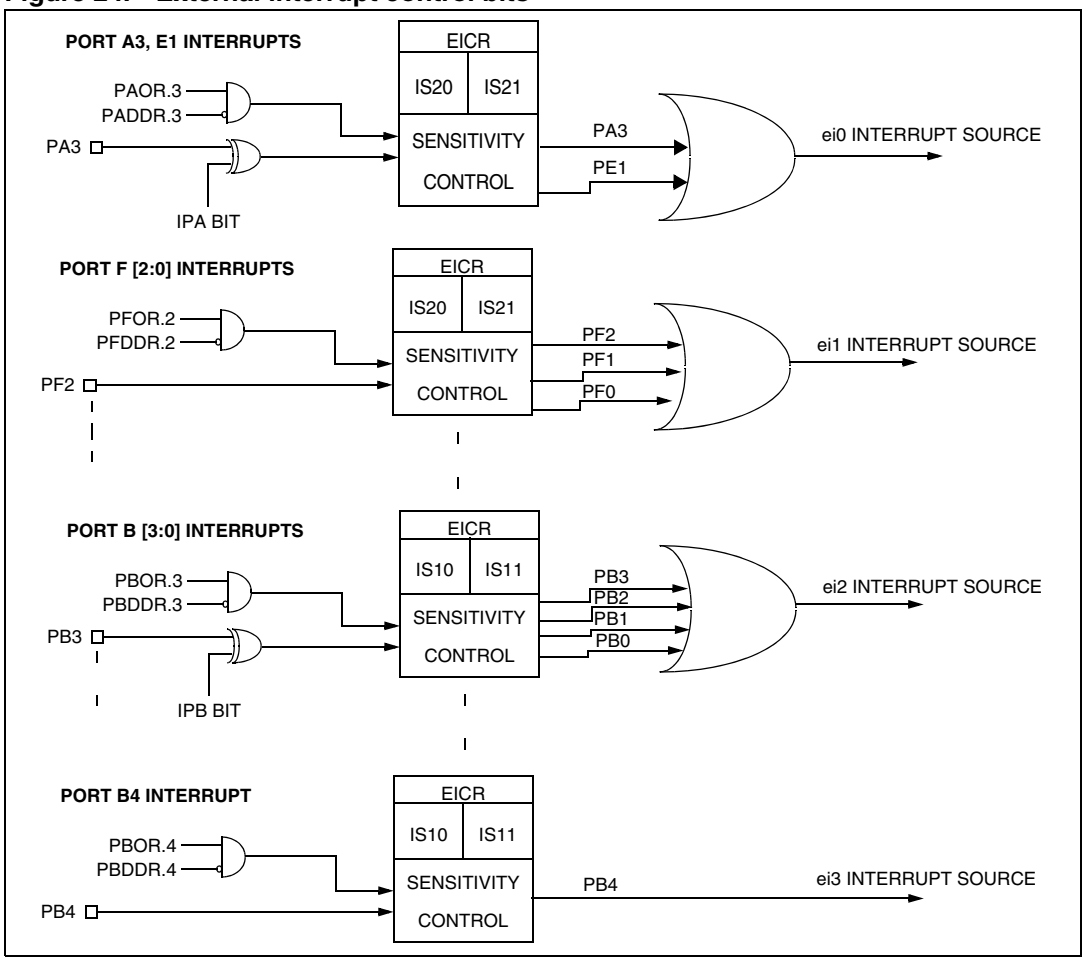
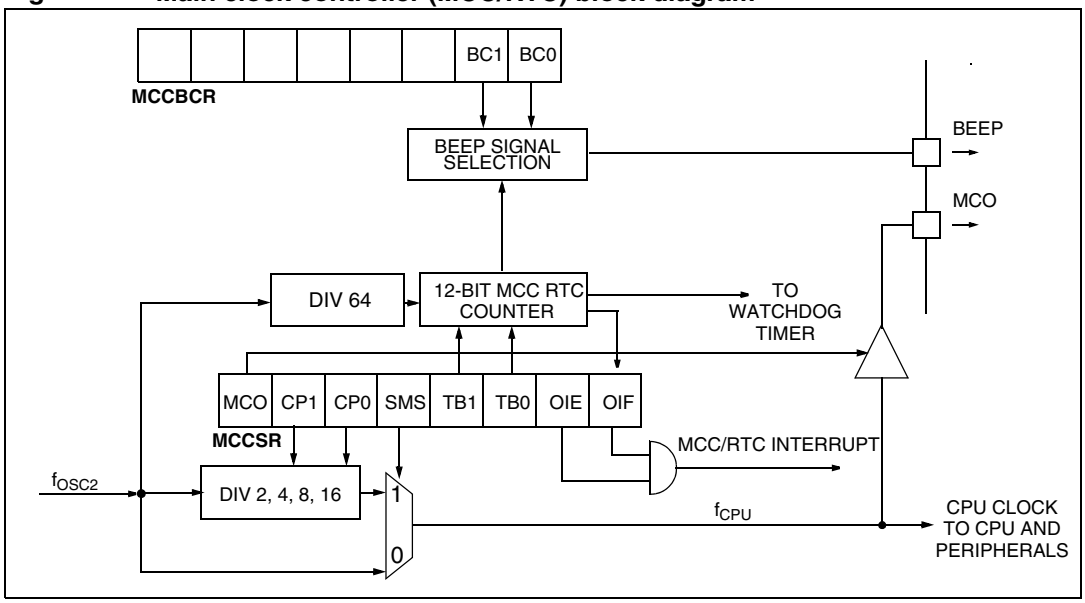


Figure 41. Main clock controller (MCC/RTC) block diagram



11.2.5 Low-power modes

Table 39. Mode description

Mode	Description
Wait	No effect on MCC/RTC peripheral. MCC/RTC interrupt cause the device to exit from Wait mode.
Active-halt	No effect on MCC/RTC counter (OIE bit is set), the registers are frozen. MCC/RTC interrupt cause the device to exit from Active-halt mode.
Halt	MCC/RTC counter and registers are frozen. MCC/RTC operation resumes when the MCU is woken up by an interrupt with “exit from Halt” capability.

11.2.6 Interrupts

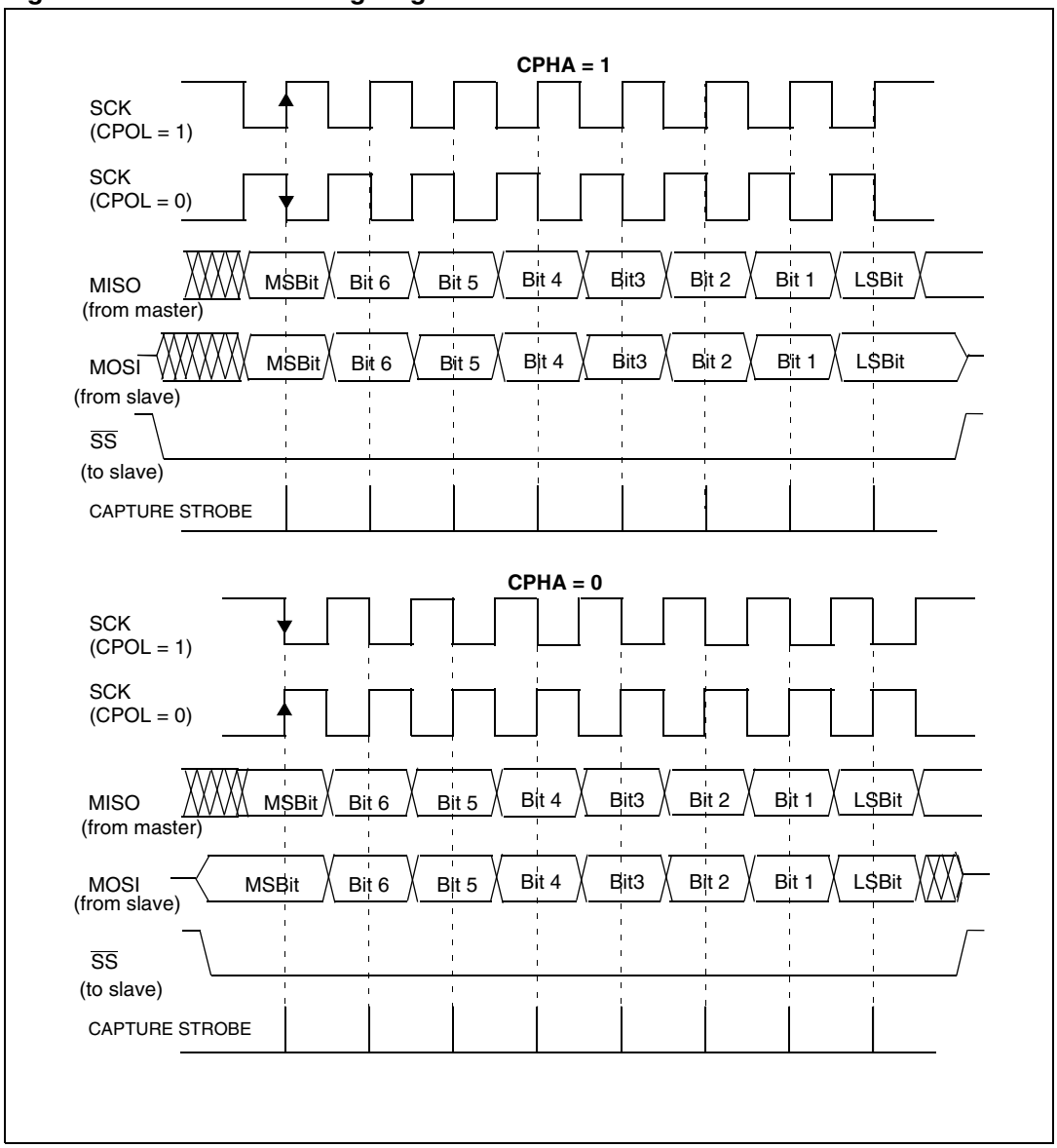
The MCC/RTC interrupt event generates an interrupt if the OIE bit of the MCCSR register is set and the interrupt mask in the CC register is not active (RIM instruction).

Table 40. Interrupt event

Interrupt event	Event flag	Enable Control bit	Exit from Wait	Exit from Halt
Time base overflow event	OIF	OIE	Yes	No (1)

1. The MCC/RTC interrupt wakes up the MCU from Active-halt mode, not from Halt mode.

Figure 60. Data clock timing diagram



1. This figure should not be used as a replacement for parametric information. Refer to *Section 13: Electrical characteristics*.

11.4.5 Error flags

Master mode fault (MODF)

Master mode fault occurs when the master device's $\overline{SS}$ pin is pulled low.

When a Master mode fault occurs:

- The MODF bit is set and an SPI interrupt request is generated if the SPIE bit is set.
- The SPE bit is reset. This blocks all output from the device and disables the SPI peripheral.
- The MSTR bit is reset, thus forcing the device into slave mode.

Framing error

A framing error is detected when:

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

– A break is received.

When the framing error is detected:

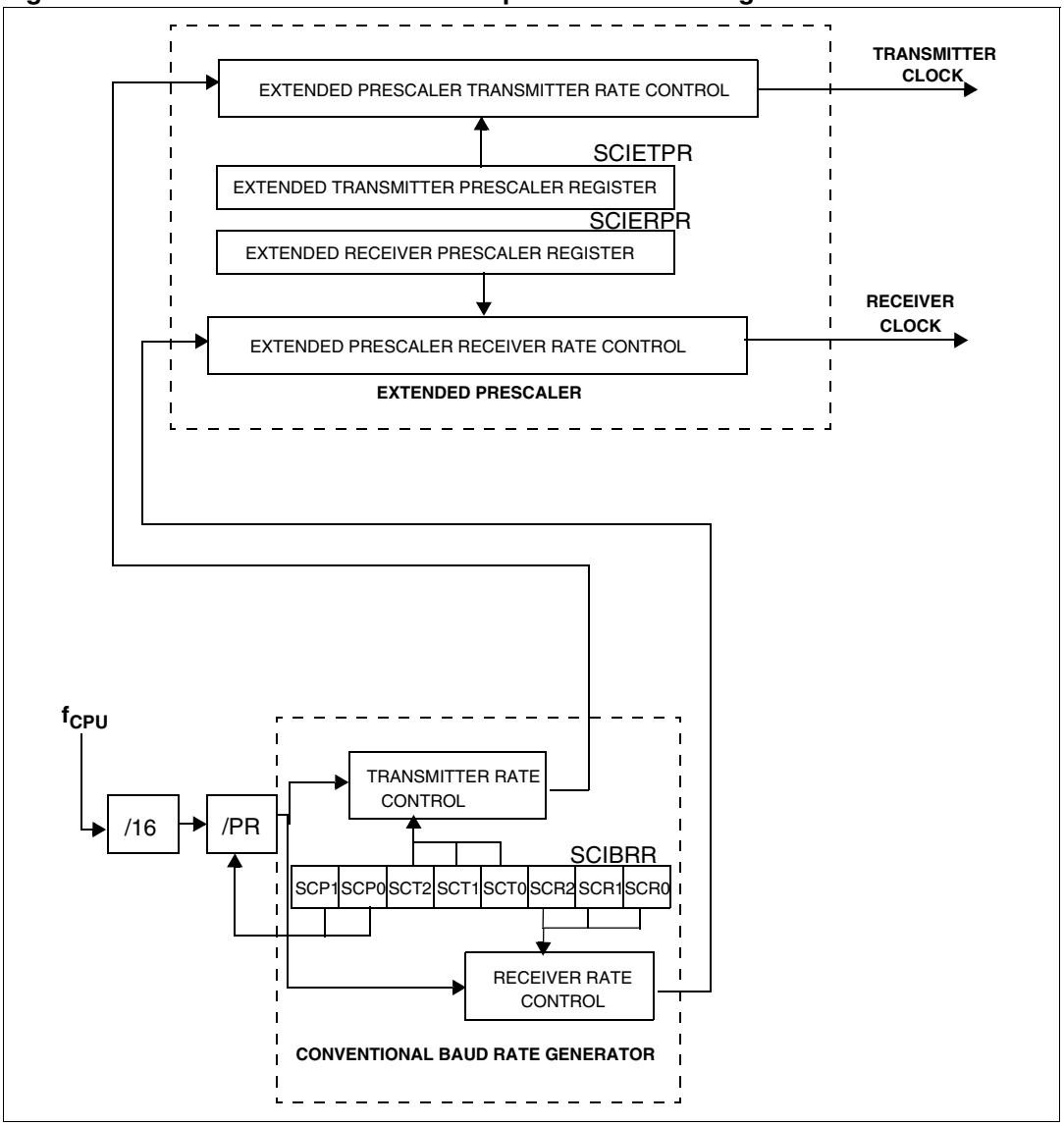
– the FE bit is set by hardware

– Data is transferred from the Shift register to the SCIDR register.

– No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

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

Figure 65. SCI baud rate and extended prescaler block diagram



Address matched: the interface generates in sequence the following:

- An Acknowledge pulse
- Depending on the LSB of the slave address sent by the master, slaves enter transmitter or receiver mode.
- Send an interrupt to the CPU after completion of the read/write operation after detecting the Stop/ Restart condition on the SDA line.

Note: The Status Register has to be read to clear the event flag associated with the interrupt.

An interrupt will be generated only if the interrupt enable bit is set in the Control Register.

Slaves 1 and 2 have a common interrupt and the Slave 3 has a separate interrupt.

At the end of write operation, I2C3S is temporarily disabled by hardware by setting BusyW bit in CR2. The byte count register, status register and current address register should be saved before resetting BusyW bit.

Slave reception (write operations)

Byte Write: The Slave address is followed by an 8-bit byte address. Upon receipt of this address, an acknowledge is generated, the address is moved into the current address register and the 8-bit data is clocked in. Once the data is shifted in, a DMA request is generated and the data is written in the RAM. The addressing device will terminate the write sequence with a stop condition. Refer to *Figure 76*.

Page Write: A page write is initiated in a similar way to a byte write, but the addressing device does not send a stop condition after the first data byte. The page length is programmed using bits 7:6 (PL[1:0]) in the Control Register1.

The current address register value is incremented by one every time a byte is written. When this address reaches the page boundary, the next byte will be written at the beginning of the same page. Refer to *Figure 77*.

Slave transmission (Read operations)

Current address read: The current address register maintains the last address accessed during the last read or write operation incremented by one.

During this operation the I2C slave reads the data pointed by the current address register. Refer to *Figure 78*.

Random read: Random read requires a dummy byte write sequence to load in the byte address. The addressing device then generates restart condition and resends the device address similar to current address read with the read/write bit high. Refer to *Figure 79*. Some types of I2C masters perform a dummy write with a stop condition and then a current address read.

In either case, the slave generates a DMA request, sends an acknowledge and serially clocks out the data.

When the memory address limit is reached, the current address will roll over and the random read will continue till the addressing master sends a stop condition.

Sequential read: Sequential reads are initiated by either a current address read or a random address read. After the addressing master receives the data byte, it responds with an acknowledge. As long as the slave receives an acknowledge, it continues to increment the current address register and clock out sequential data bytes.

Table 74. ADC register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0070h	**ADCCSR** Reset value	EOC 0	SPEED 0	ADON 0	0	CH3 0	CH2 0	CH1 0	CH0 0
0071h	**ADCDRH** Reset value	D9 0	D8 0	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0
0072h	**ADCDRL** Reset value	0	0	0	0	0	0	D1 0	D0 0

Indexed (No offset)

There is no offset (no extra byte after the opcode), and allows 00 - FF addressing space.

- Indexed (Short)

  The offset is a byte, thus requires only 1 byte after the opcode and allows 00 - 1FE addressing space.

- Indexed (Long)

  The offset is a word, thus allowing 64-Kbyte addressing space and requires 2 bytes after the opcode.

12.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two submodes:

Indirect (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64-Kbyte addressing space, and requires 1 byte after the opcode.

12.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two submodes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64-Kbyte addressing space, and requires 1 byte after the opcode.

Table 85. Current characteristics

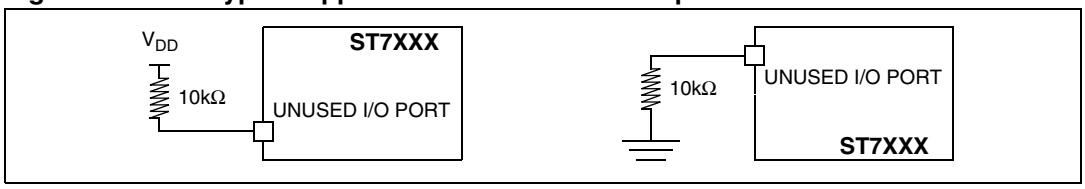
Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) (1)	75	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) (1)	150	
I_{IO}	Output current sunk by any standard I/O and control pin	20	
	Output current sunk by any high sink I/O pin	40	
	Output current source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

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 disturbs the analog performance of the device. In particular, it induces leakage currents throughout the device including the analog inputs. To avoid undesirable effects on the analog functions, care must be taken:
   - Analog input pins must have a negative injection less than 0.8 mA (assuming that the impedance of the analog voltage is lower than the specified limits)
   - Pure digital pins must have a negative injection less than 1.6 mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.
4. No negative current injection allowed on PB0 pin.
5. 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 characterisation with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

Table 86. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	–65 to +150	°C
T_J	Maximum junction temperature (see *Table 118*)		

Figure 98. Two typical applications with unused I/O pin



1. I/O can be left unconnected if it is configured as output (0 or 1) by the software. This has the advantage of greater EMC robustness and lower cost.

Caution: During normal operation the ICCCLK pin must be pulled-up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset.

Table 108. Output driving current (1)

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} (2)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see *Figure 101*)	V_{DD} = 5 V	I_{IO} = +5 mA		1.0	V
			I_{IO} = +2 mA		0.4	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see *Figure 104*)		I_{IO} = +20 mA		1.3	
			I_{IO} = +8 mA		0.75	
V_{OH} (3)	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 113*)		I_{IO} = -5 mA	V_{DD}-1.5		
			I_{IO} = -2 mA	V_{DD}-0.8		
V_{OL} (2)(4)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see *Figure 100*)	V_{DD} = 3.3 V	I_{IO} = +2 mA		0.7	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time		I_{IO} = +8 mA		0.5	
V_{OH} (3)(4)	Output high level voltage for an I/O pin when 4 pins are sourced at same time (*Figure 111*)		I_{IO} = -2 mA	V_{DD}-0.8		
V_{OL} (2)(4)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see *Figure 102*)	V_{DD} = 2.7 V	I_{IO} = +2 mA		0.9	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time		I_{IO} = +8 mA		0.6	
V_{OH} (3)(4)	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 110*)		I_{IO} = -2 mA	V_{DD}-0.9		

1. Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A unless otherwise specified.
2. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Table 85: Current characteristics* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
3. The I_{IO} current sourced must always respect the absolute maximum rating specified in *Table 85: Current characteristics* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}.
4. Not tested in production, based on characterization results.

15.3.4 Order codes for ST72F34x development tools

Table 125. Development tool order codes

MCU	Starter kit	Emulator	Programming tool	
			In-circuit debugger/ programmer	Dedicated programmer
ST72F344 ST72F345	ST72F34x-SK/RAIS (1)	ST7MDT40-EMU3 (2)	STX-RLINK (3) ST7-STICK (4)(5)	ST7SB20J/xx (4)(6) ST7SB40-QP48/xx (4)(7)

1. USB connection to PC
2. ST7MDT40-EMU3 order code is discontinued
3. RLink with ST7 tool set
4. Add suffix /EU, /UK or /US for the power supply for your region
5. Parallel port connection to PC
6. Only available for LQFP32 and LQFP44 packages
7. Only available for LQFP48 package

For additional ordering codes for spare parts and accessories, refer to the online product selector at www.st.com/mcu.