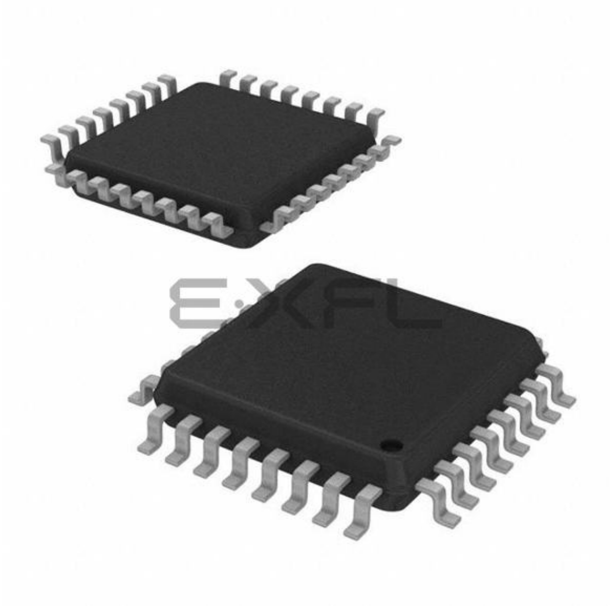


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
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	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f324bk4t6tr

Table 49. CR1 register description 89
Table 50. CR2 register description 91
Table 51. CSR register description 92
Table 52. 16-bit timer register map and reset values 96
Table 53. Effect of low power modes on SPI 105
Table 54. SPI interrupt control/wake-up capability 105
Table 55. SPICR register description 106
Table 56. SPI master mode SCK frequency 107
Table 57. SPICSR register description 108
Table 58. SPI register map and reset values 109
Table 59. Frame formats 119
Table 60. Effect of low power modes on SCI 122
Table 61. SCI interrupt control/wake-up capability 122
Table 62. SCISR register description 123
Table 63. SCICR1 register description 124
Table 64. SCICR2 register description 125
Table 65. SCIBRR register description 127
Table 66. SCIERPR register description 128
Table 67. SCIETPR register description 129
Table 68. Baud rate selection 129
Table 69. SCI register map and reset values 130
Table 70. Effect of low power modes on ADC 133
Table 71. ADCCSR register description 133
Table 72. ADCDRH register description 134
Table 73. ADCDRL register description 135
Table 74. ADC register map and reset values 135
Table 75. Addressing mode groups 136
Table 76. CPU addressing mode overview 136
Table 77. Inherent instructions 137
Table 78. Immediate instructions 138
Table 79. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes 139
Table 80. Relative direct and indirect instructions and functions 140
Table 81. Instruction groups 140
Table 82. Instruction set overview 143
Table 83. Voltage characteristics 146
Table 84. Current characteristics 147
Table 85. Thermal characteristics. 147
Table 86. Operating conditions. 148
Table 87. Operating conditions with LVD 149
Table 88. AVD thresholds. 149
Table 89. ROM current consumption 150
Table 90. Flash current consumption 151
Table 91. Oscillators, PLL and LVD current consumption 152
Table 92. On-chip peripherals current consumption. 152
Table 93. General timings. 153
Table 94. External clock source 153
Table 95. Crystal and ceramic resonator oscillators. 154
Table 96. OSCRANGE selection for typical resonators 155
Table 97. RC oscillators 155
Table 98. PLL characteristics 156
Table 99. RAM and hardware registers 156
Table 100. Dual voltage HDFlash memory. 157

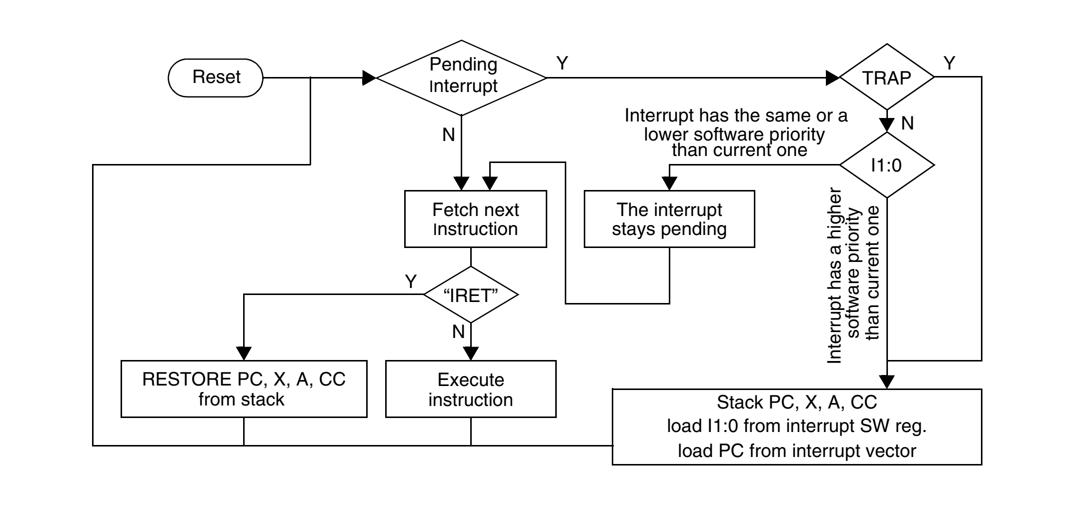
List of figures

Figure 1. Device block diagram 14
Figure 2. 44-pin LQFP package pinout 15
Figure 3. 32-pin LQFP package pinout 15
Figure 4. Memory map. 19
Figure 5. Memory map and sector address 23
Figure 6. Typical ICC interface 24
Figure 7. CPU registers 26
Figure 8. Stack manipulation example. 30
Figure 9. PLL block diagram 31
Figure 10. Clock, reset and supply block diagram 32
Figure 11. Reset sequence phases 34
Figure 12. Reset block diagram 35
Figure 13. RESET sequences 36
Figure 14. Low voltage detector vs reset 37
Figure 15. Using the AVD to monitor V_{DD} 38
Figure 16. Interrupt processing flowchart 42
Figure 17. Priority decision process flowchart 42
Figure 18. Concurrent interrupt management 44
Figure 19. Nested interrupt management 45
Figure 20. External interrupt control bits 48
Figure 21. Power saving mode transitions. 52
Figure 22. Slow mode clock transitions 53
Figure 23. Wait mode flowchart 53
Figure 24. Active Halt timing overview. 54
Figure 25. Active Halt mode flowchart 55
Figure 26. HALT timing overview. 56
Figure 27. Halt mode flowchart 56
Figure 28. I/O port general block diagram 60
Figure 29. Interrupt I/O port state transitions 62
Figure 30. Watchdog block diagram 66
Figure 31. Approximate timeout duration. 66
Figure 32. Exact timeout duration (t_{min} and t_{max}). 67
Figure 33. Main clock controller (MCC/RTC) block diagram 70
Figure 34. Timer block diagram 76
Figure 35. 16-bit read sequence 77
Figure 36. Counter timing diagram, internal clock divided by 2 78
Figure 37. Counter timing diagram, internal clock divided by 4 78
Figure 38. Counter timing diagram, internal clock divided by 8 78
Figure 39. Input capture block diagram 80
Figure 40. Input capture timing diagram 80
Figure 41. Output compare block diagram 83
Figure 42. Output compare timing diagram, $f_{TIMER} = f_{CPU}/2$ 83
Figure 43. Output compare timing diagram, $f_{TIMER} = f_{CPU}/4$ 83
Figure 44. One pulse mode cycle 84
Figure 45. One Pulse mode timing example(1) 85
Figure 46. Pulse width modulation mode timing example with two output compare functions(1)(2) . . 86
Figure 47. Pulse width modulation cycle 87
Figure 48. Serial peripheral interface block diagram 98

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 16. Interrupt processing flowchart


Reset
Pending Interrupt
Y
N
TRAP
Y
N
Interrupt has the same or a lower software priority than current one
I1:0
Fetch next instruction
The interrupt stays pending
Interrupt has a higher software priority than current one
Y
"IRET"
N
RESTORE PC, X, A, CC from stack
Execute instruction
Stack PC, X, A, CC
load I1:0 from interrupt SW reg.
load PC from interrupt vector

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 17 describes this decision process.

Figure 17. Priority decision process flowchart

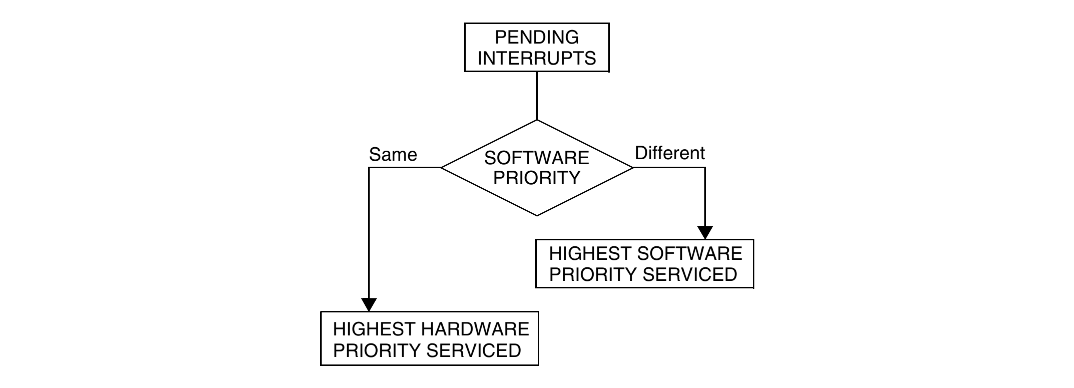
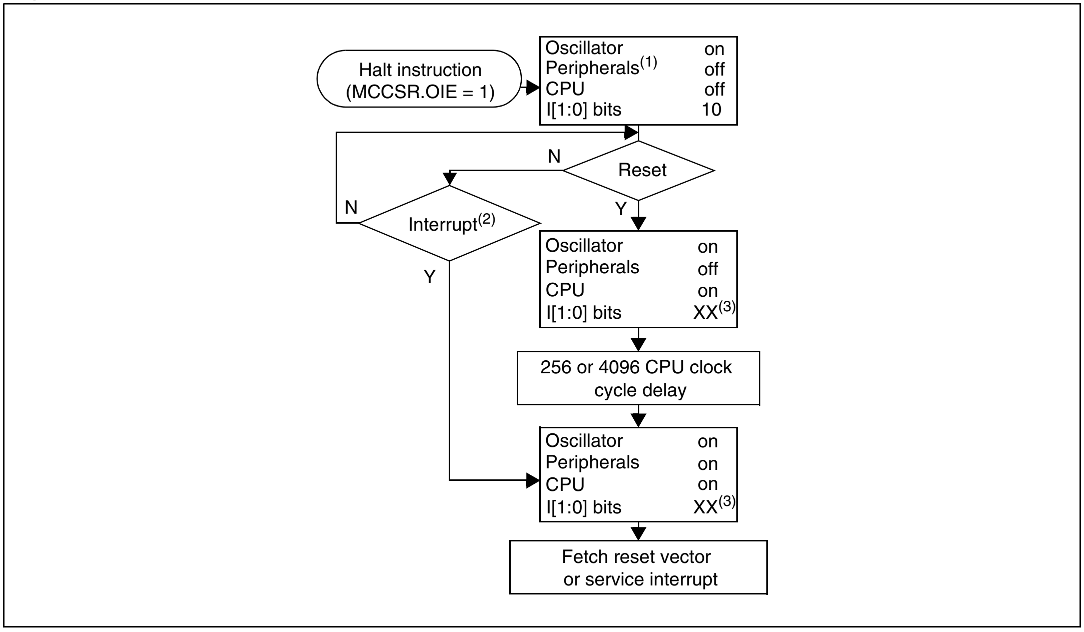

PENDING INTERRUPTS
Same
SOFTWARE PRIORITY
Different
HIGHEST SOFTWARE PRIORITY SERVICED
HIGHEST HARDWARE PRIORITY SERVICED

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

Instruction	New description	Function/example	I1	H	I0	N	Z	C
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
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 25. 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 27*).

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.

10.3 16-bit timer

10.3.1 Introduction

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

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (input capture) or generation of up to two output waveforms (output compare and PWM).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

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

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

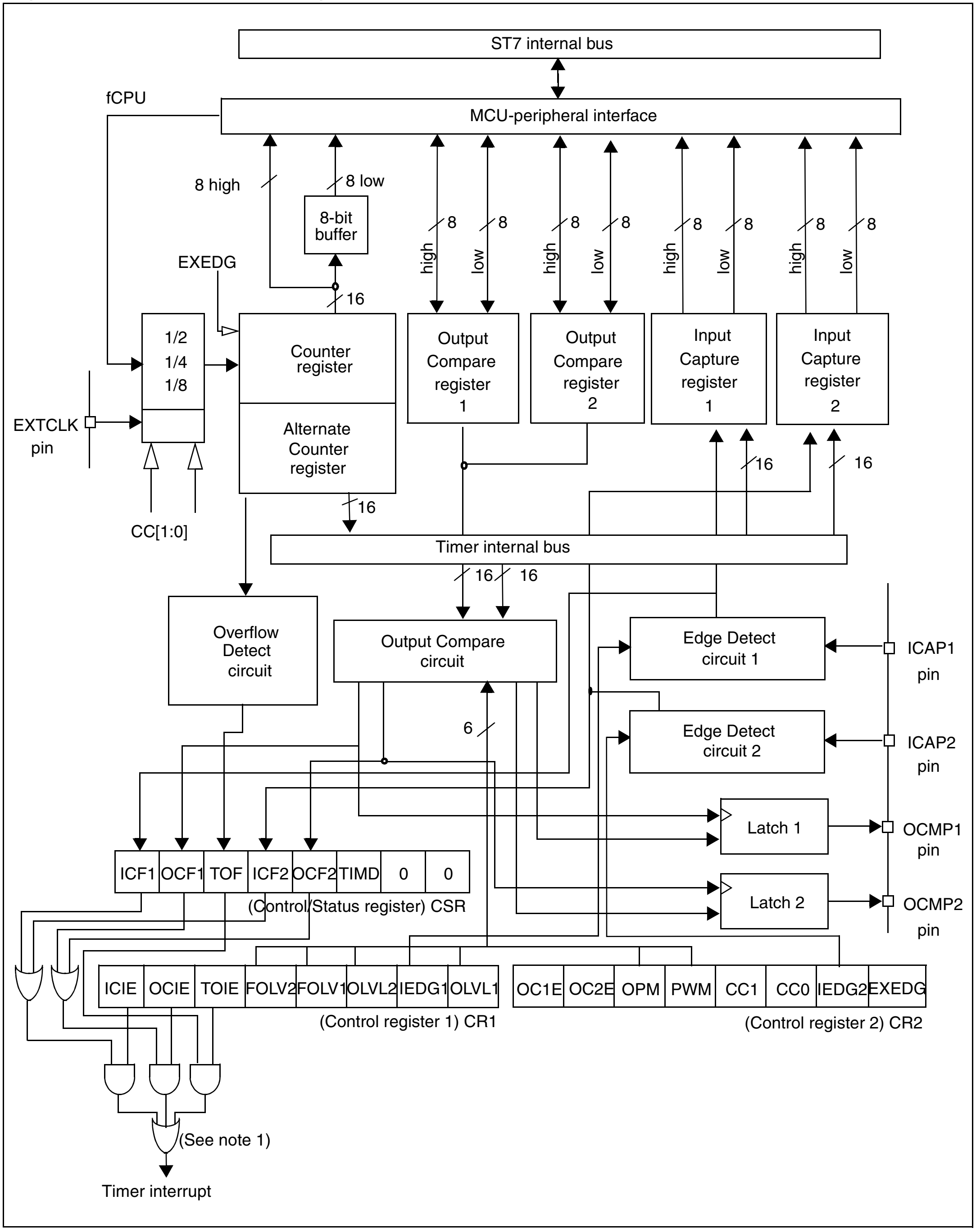
10.3.2 Main features

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

The timer block diagram is shown in *Figure 34*.

[c] Some timer pins may not be available (not bonded) in some ST7 devices. Refer to *Section 2: Pin description*. When reading an input signal on a non-bonded pin, the value will always be ‘1’.

Figure 34. Timer block diagram



1. If IC, OC and TO interrupt requests have separate vectors then the last OR is not present (see *Table 25: Interrupt mapping on page 51*).

Input capture

In this section, the index, *i*, may be 1 or 2 because there are two input capture functions in the 16-bit timer.

The two 16-bit input capture registers (IC1R/IC2R) are used to latch the value of the free running counter after a transition is detected on the ICAP*i* pin (see *Figure 40*).

Table 44. Input capture byte distribution

Register	MS byte	LS byte
ICiR	IC*i*HR	IC*i*LR

The ICiR registers are read-only registers.

The active transition is software programmable through the IEDG*i* bit of Control Registers (CR*i*).

Timing resolution is one count of the free running counter: (f_{CPU}/CC[1:0]).

Procedure

To use the input capture function select the following in the CR2 register:

- Select the timer clock (CC[1:0]) (see *Table 50*).
- Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

Select the following in the CR1 register:

- Set the ICIE bit to generate an interrupt after an input capture coming from either the ICAP1 pin or the ICAP2 pin
- Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

When an input capture occurs:

- ICF*i* bit is set.
- The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see *Figure 40*).
- A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set
2. An access (read or write) to the IC*i*LR register

One Pulse mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

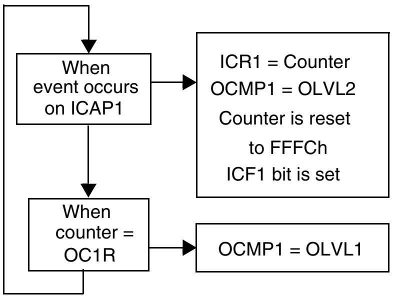
The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure

To use One Pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula below).
2. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
   - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
3. Select the following in the CR2 register:
   - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
   - Set the OPM bit.
   - Select the timer clock CC[1:0] (see *Table 50*).

Figure 44. One pulse mode cycle



Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.

Table 55. SPICR register description (continued)

Bit	Name	Function
2	CPHA	Clock Phase This bit is set and cleared by software. 0: The first clock transition is the first data capture edge. 1: The second clock transition is the first capture edge. *Note: The slave must have the same CPOL and CPHA settings as the master.*
1:0	SPR[1:0]	Serial clock frequency These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode (see *Table 56*). *Note: These 2 bits have no effect in slave mode.*

Table 56. SPI master mode SCK frequency

Serial clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

SPI Control/Status Register (SPICSR)

SPICSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SPIF	WCOL	OVR	MODF	Reserved	SOD	SSM	SSI
RO	RO	RO	RO	-	R/W	R/W	R/W

10.5 Serial communications interface (SCI)

10.5.1 Introduction

The serial communications interface (SCI) offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The SCI offers a very wide range of baud rates using two baud rate generator systems.

10.5.2 Main features

- Full duplex, asynchronous communications
- NRZ standard format (mark/space)
- Dual baud rate generator systems
- Independently programmable transmit and receive baud rates up to 500K baud.
- Programmable data word length (8 or 9 bits)
- Receive buffer full, Transmit buffer empty and End of Transmission flags
- 2 receiver wake-up modes
  - Address bit (MSB)
  - Idle line
- Muting function for multiprocessor configurations
- Separate enable bits for Transmitter and Receiver
- 4 error detection flags
  - Overrun error
  - Noise error
  - Frame error
  - Parity error
- 5 interrupt sources with flags
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle line received
  - Overrun error detected
- Parity control
  - Transmits parity bit
  - Checks parity of received data byte
- Reduced power consumption mode

Receiver muting and wake-up feature

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

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

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

All the reception status bits cannot be set.

All the receive interrupts are inhibited.

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

- by Idle Line detection if the Wake bit is reset,
- by Address Mark detection if the Wake bit is set.

A receiver wakes up by Idle Line detection when the Receive line has recognized an Idle Frame. Then the RWU bit is reset by hardware but the Idle bit is not set.

A receiver wakes up by Address Mark detection when it received a ‘1’ as the most significant bit of a word, thus indicating that the message is an address. The reception of this particular word wakes up the receiver, resets the RWU bit and sets the RDRF bit, which allows the receiver to receive this word normally and to use it as an address word.

Caution: In Mute mode, do not write to the SCICR2 register. If the SCI is in Mute mode during the read operation (RWU = 1) and an address mark wake-up event occurs (RWU is reset) before the write operation, the RWU bit will be set again by this write operation. Consequently the address byte is lost and the SCI is not woken up from Mute mode.

Parity control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in *Table 59*.

Table 59. Frame formats(1)(2)

M bit	PCE bit	SCI frame
0	0	I SB I 8 bit data I STB I
0	1	I SB I 7-bit data I PB I STB I
1	0	I SB I 9-bit data I STB I
1	1	I SB I 8-bit data PB I STB I

1. SB = Start bit, STB = Stop bit, and PB = Parity bit.
2. In case of wake-up by an address mark, the MSB bit of the data is taken into account and not the Parity bit.

Table 62. SCISR register description

Bit	Name	Function
7	TDRE	Transmit Data Register Empty This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE bit = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Data is not transferred to the shift register. 1: Data is transferred to the shift register. *Note: Data will not be transferred to the shift register unless the TDRE bit is cleared.*
6	TC	Transmission Complete This bit is set by hardware when transmission of a frame containing data is complete. An interrupt is generated if TCIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Transmission is not complete 1: Transmission is complete *Note: TC is not set after the transmission of a Preamble or a Break.*
5	RDRF	Received Data Ready Flag This bit is set by hardware when the content of the RDR register has been transferred to the SCIDR register. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: Data is not received 1: Received data is ready to be read
4	IDLE	Idle line detect This bit is set by hardware when a Idle Line is detected. An interrupt is generated if the ILIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No idle line is detected 1: Idle line is detected *Note: The IDLE bit is not reset until the RDRF bit has itself been set (that is, a new idle line occurs).*
3	OR	Overrun error This bit is set by hardware when the word currently being received in the shift register is ready to be transferred into the RDR register while RDRF = 1. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No overrun error 1: Overrun error is detected *Note: When this bit is set RDR register content is not lost but the shift register is overwritten.*
2	NF	Noise Flag This bit is set by hardware when noise is detected on a received frame. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No noise is detected 1: Noise is detected *Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt.*

Table 64. SCICR2 register description (continued)

Bit	Name	Function
5	RIE	Receiver interrupt Enable This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever OR = 1 or RDRF = 1 in the SCISR register.
4	ILIE	Idle Line Interrupt Enable This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever IDLE = 1 in the SCISR register.
3	TE	Transmitter Enable This bit enables the transmitter. It is set and cleared by software. 0: Transmitter is disabled 1: Transmitter is enabled *Notes:* *- During transmission, a ‘0’ pulse on the TE bit (‘0’ followed by ‘1’) sends a preamble (Idle line) after the current word.* *- When TE is set there is a 1 bit-time delay before the transmission starts.* **Caution:** The TDO pin is free for general purpose I/O only when the TE and RE bits are both cleared (or if TE is never set).
2	RE	Receiver Enable This bit enables the receiver. It is set and cleared by software. 0: Receiver is disabled 1: Receiver is enabled and begins searching for a start bit *Note: Before selecting Mute mode (setting the RWU bit), the SCI must first receive some data, otherwise it cannot function in Mute mode with Wake-Up by Idle line detection.*
1	RWU	Receiver Wake-Up This bit determines if the SCI is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wake-up sequence is recognized. 0: Receiver in Active mode 1: Receiver in Mute mode
0	SBK	Send Break This bit set is used to send break characters. It is set and cleared by software. 0: No break character is transmitted. 1: Break characters are transmitted. *Note: If the SBK bit is set to ‘1’ and then to ‘0’, the transmitter will send a Break word at the end of the current word.*

Note: To reduce disturbance to the RC oscillator, it is recommended to place decoupling capacitors between V_{DD} and V_{SS} as shown in Figure 85 on page 173.

12.6.5 PLL characteristics

Table 98. PLL characteristics

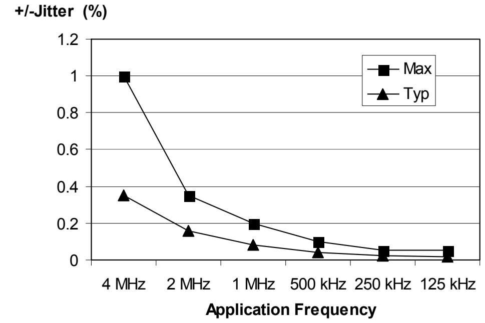
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	PLL input frequency range		2		4	MHz
$\Delta f_{CPU}/f_{CPU}$	Instantaneous PLL jitter[1]	f_{OSC} = 4 MHz		0.7	2	%

[1] Data characterized but not tested

The user must take the PLL jitter into account in the application (for example in serial communication or sampling of high frequency signals). The PLL jitter is a periodic effect, which is integrated over several CPU cycles. Therefore the longer the period of the application signal, the less it will be impacted by the PLL jitter.

Figure 67 shows the PLL jitter integrated on application signals in the range 125 kHz to 2 MHz. At frequencies of less than 125 kHz, the jitter is negligible.

Figure 67. Integrated PLL jitter vs signal frequency[1]



[1] Measurement conditions: f_{CPU} = 8 MHz

12.7 Memory characteristics

12.7.1 RAM and hardware registers

Table 99. RAM and hardware registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode[1]	Halt mode (or reset)	1.6			V

[1] Minimum V_{DD} supply voltage without losing data stored in RAM (in Halt mode or under reset) or in hardware registers (only in Halt mode). Not tested in production.

12.7.2 Flash memory

Table 100. Dual voltage HDFlash memory

Symbol	Parameter	Conditions	Min[1]	Typ	Max[1]	Unit
f_{CPU}	Operating frequency	Read mode	0		8	MHz
		Write/Erase mode	1		8	
V_{PP}	Programming voltage[2]	$4.5V \le V_{DD} \le 5.5V$	11.4		12.6	V
I_{DD}	Supply current[3]	Write/Erase		<10		µA
I_{PP}	V_{PP} current[3]	Read (V_{PP} = 12V)			200	µA
		Write/Erase			30	mA
t_{VPP}	Internal V_{PP} stabilization time			10		µs
t_{RET}	Data retention	T_A = 55°C	20			years
N_{RW}	Write/Erase cycles	T_A = 85°C	100			cycles
T_{PROG} T_{ERASE}	Programming or erasing temperature range		-40	25	85	°C

1. Data based on characterization results, not tested in production.
2. V_{PP} must be applied only during the programming or erasing operation and not permanently for reliability reasons.
3. Data based on simulation results, not tested in production.

12.8 EMC characteristics

Susceptibility tests are performed on a sample basis during product characterization.

12.8.1 Functional electromagnetic susceptibility (EMS)

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

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

A device reset allows normal operations to be resumed. The test results given in *Table 101 on page 159* are based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

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

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

Software recommendations

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

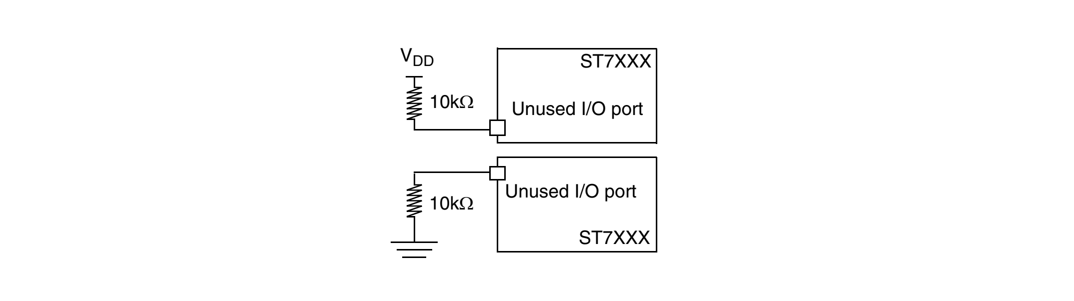
- corrupted program counter
- unexpected reset
- critical data corruption (control registers...)

Prequalification trials

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

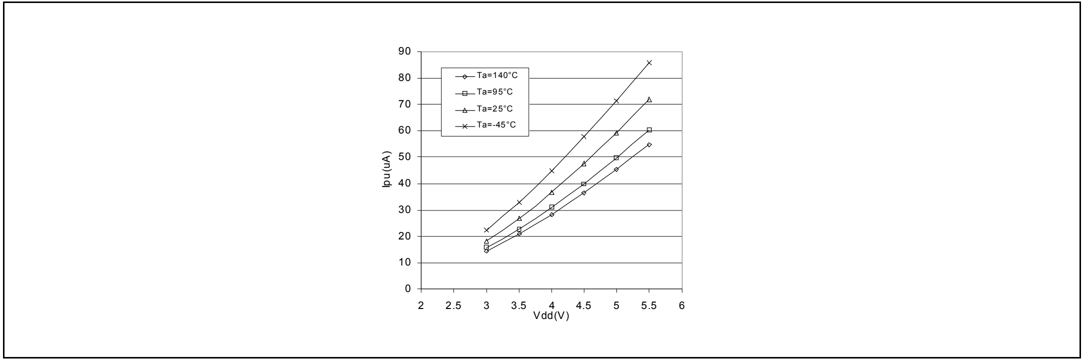
To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behavior is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

Figure 68. Unused I/O pins configured as input[1]



[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.

Figure 69. Typical I_{PU} vs. V_{DD} with $V_{IN} = V_{SS}$



12.9.2 Output driving current

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

Table 106. Output driving current

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL}[1]	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see *Figure 70*)	V_{DD} = 5V	I_{IO} = +5mA		1.2	V
			I_{IO} = +2mA		0.5	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see *Figure 71* and *Figure 73*)		I_{IO} = +20mA $T_A \leq 85°$ $T_A > 85°C$		1.3 1.5	
			I_{IO} = +8mA		0.6	
V_{OH}[2]	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 72* and *Figure 75*)		I_{IO} = -5mA, $T_A \leq 85°C$ $T_A > 85°C$	V_{DD} - 1.4 V_{DD} - 1.6		
			I_{IO} = -2mA	V_{DD} - 0.7		

[1] The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 12.2.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.

[2] The I_{IO} current sourced must always respect the absolute maximum rating specified in *Section 12.2.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins do not have V_{OH}.

Table 118. Option byte 1 bit description (continued)

Bit	Name	Function
OPT3:1	OSCRANGE[2:0]	Oscillator range When the resonator oscillator type is selected, these option bits select the resonator oscillator current source corresponding to the frequency range of the used resonator. When the external clock source is selected, these bits are set to medium power (2 ~ 4 MHz). 000: Typ. frequency range (LP) = 1 ~ 2 MHz 001: Typ. frequency range (MP) = 2 ~ 4 MHz 010: Typ. frequency range (MS) = 4 ~ 8 MHz 011: Typ. frequency range (HS) = 8 ~ 16 MHz
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 119. Package selection (OPT7)

Version	Selected package	PKG1
J	LQFP44	1
K	LQFP32	0

```
RIM ; reset the interrupt mask
LD A,sema ; check the semaphore status
CP A,#$01
jrne OUT
call call_routine ; call the interrupt routine
RIM
OUT:RIM
JP while_loop
.call_routine ; entry to call_routine
PUSH A
PUSH X
PUSH CC
.ext1_rt ; entry to interrupt routine
LD A,#$00
LD sema,A
IRET
```

15.1.3 Unexpected reset fetch

If an interrupt request occurs while a “POP CC” instruction is executed, the interrupt controller does not recognize the source of the interrupt and, by default, passes the reset vector address to the CPU.

Workaround

To solve this issue, a “POP CC” instruction must always be preceded by a “SIM” instruction.

15.1.4 Clearing active interrupts outside interrupt routine

When an active interrupt request occurs at the same time as the related flag is being cleared, an unwanted reset may occur.

Note: Clearing the related interrupt mask will not generate an unwanted reset.

Concurrent interrupt context

The symptom does not occur when the interrupts are handled normally, that is, when:

- The interrupt flag is cleared within its own interrupt routine
- The interrupt flag is cleared within any interrupt routine
- The interrupt flag is cleared in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

Perform SIM and RIM operation before and after resetting an active interrupt request. Example:

- – SIM
- – Reset interrupt flag
- – RIM