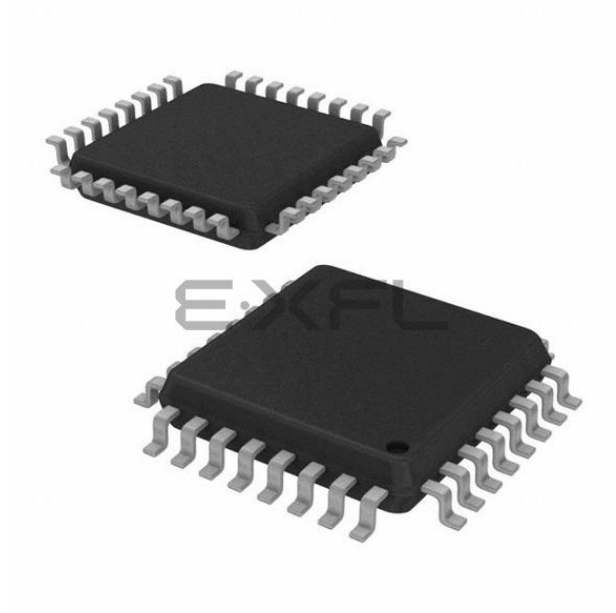


Welcome to E-XFL.COM

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	24
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 8x10b
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/st72f344k4t6tr

1 Description

The ST7234x devices are members of the ST7 microcontroller family. *Table 2* gives the available part numbers and details on the devices. All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

They feature single-voltage Flash memory with byte-by-byte in-circuit programming (ICP) and in-application programming (IAP) capabilities.

Under software control, all devices can be placed in Wait, Slow, Auto-wakeup from Halt, Active-halt or Halt mode, reducing the power consumption when the application is in idle or stand-by state.

The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

The devices feature an on-chip debug module (DM) to support in-circuit debugging (ICD). For a description of the DM registers, refer to the ST7 ICC Protocol Reference Manual.

Figure 1. General block diagram

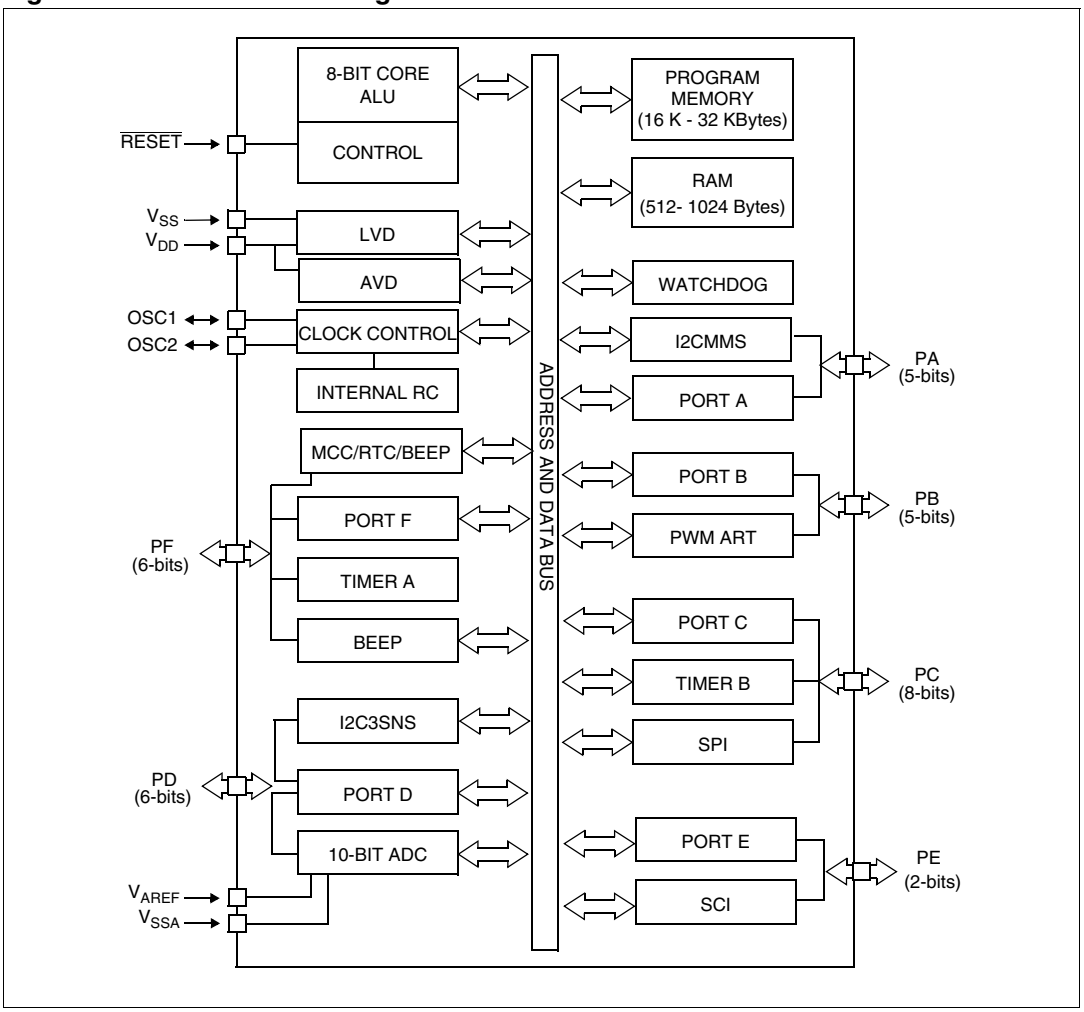


Table 4. Hardware register map

Address	Block	Register label	Register name	Reset status (1)	Remarks (2)
0000h 0001h 0002h	Port A (3)	PADR PADDR PAOR	Port A Data Register Port A Data Direction Register Port A Option Register	00h (4) 00h 00h	R/W R/W R/W
0003h 0004h 0005h	Port B (3)	PBDR PBDDR PBOR	Port B Data Register Port B Data Direction Register Port B Option Register	00h (4) 00h 00h	R/W R/W R/W
0006h 0007h 0008h	Port C (3)	PCDR PCDDR PCOR	Port C Data Register Port C Data Direction Register Port C Option Register	00h (4) 00h 00h	R/W R/W R/W
0009h 000Ah 000Bh	Port D (3)	PDADR PDDDR PDOR	Port D Data Register Port D Data Direction Register Port D Option Register	00h (4) 00h 00h	R/W R/W R/W
000Ch 000Dh 000Eh	Port E (3)	PEDR PEDDR PEOR	Port E Data Register Port E Data Direction Register Port E Option Register	00h (4) 00h 00h	R/W R/W R/W
000Fh 0010h 0011h	Port F (3)	PFDR PFDDR PFOR	Port F Data Register Port F Data Direction Register Port F Option Register	00h (4) 00h 00h	R/W R/W R/W
0012h to 0016h	Reserved area (5 bytes)				
0017h 0018h	RC	RCCRH RCCRL	RC oscillator Control Register High RC oscillator Control Register Low	FFh 03h	R/W R/W
0019h	Reserved area (1 byte)				
001Ah to 001Fh	DM (5)	Reserved area (6 bytes)			
00020h	EEPROM	EECSR	Data EEPROM Control/Status Register	00h	R/W
0021h 0022h 0023h	SPI	SPIDR SPICR SPICSR	SPI Data I/O Register SPI Control Register SPI Control Status Register	xxh 0xh 00h	R/W R/W R/W
0024h 0025h 0026h 0027h	ITC	ISPR0 ISPR1 ISPR2 ISPR3	Interrupt Software Priority Register 0 Interrupt Software Priority Register 1 Interrupt Software Priority Register 2 Interrupt Software Priority Register 3	FFh FFh FFh FFh	R/W R/W R/W R/W
0028h		EICR	External Interrupt Control Register	00h	R/W
00029h	Flash	FCSR	Flash Control/Status Register	00h	R/W
002Ah	WWDG	WDGCR	Watchdog Control Register	7Fh	R/W
002Bh	SI	SICSR	System Integrity Control/Status Register	000x 000xb	R/W
002Ch 002Dh	MCC	MCCSR MCCBCR	Main Clock Control/Status Register MCC Beep Control Register	00h 00h	R/W R/W
002Eh 002Fh	AWU	AWUCSR AWUPR	AWU Control/Status Register AWU Prescaler Register	00h FFh	R/W R/W
0030h	WWDG	WDGWR	Window Watchdog Control Register	7Fh	R/W

5 Data EEPROM

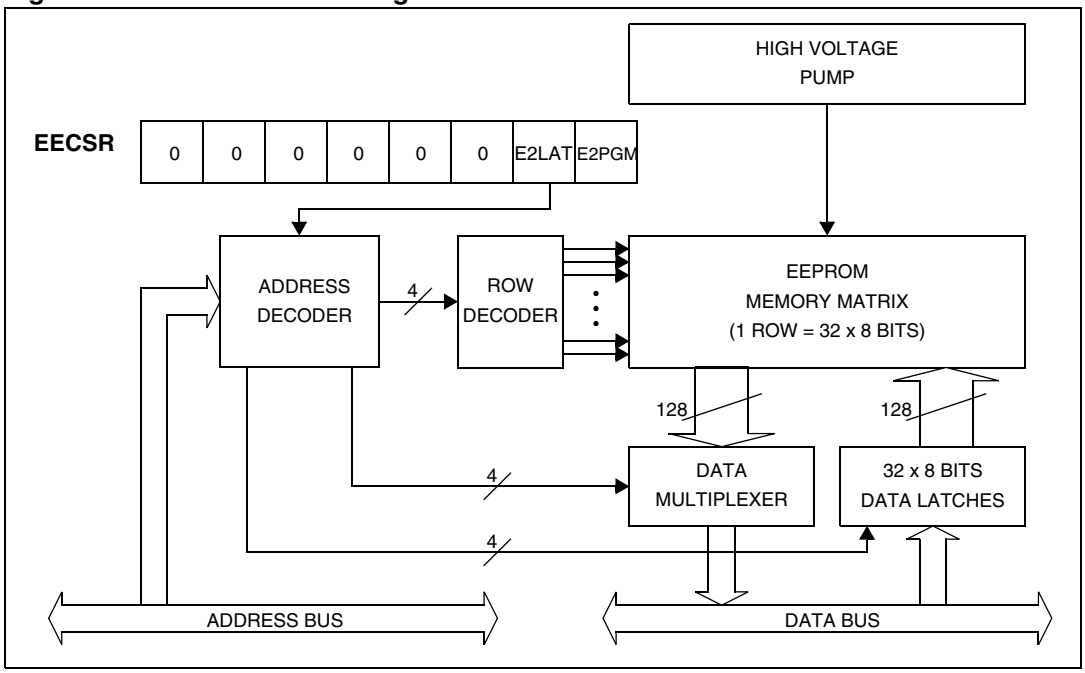
5.1 Introduction

The electrically erasable programmable read only memory can be used as a non-volatile backup for storing data. Using the EEPROM requires a basic access protocol described in this chapter.

5.2 Main features

- Up to 32 bytes programmed in the same cycle
- EEPROM mono-voltage (charge pump)
- Chained erase and programming cycles
- Internal control of the global programming cycle duration
- Wait mode management
- Readout protection

Figure 7. EEPROM block diagram



5.3 Memory access

The Data EEPROM memory read/write access modes are controlled by the E2LAT bit of the EEPROM Control/Status register (EECSR). The flowchart in *Figure 8* describes these different memory access modes.

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)

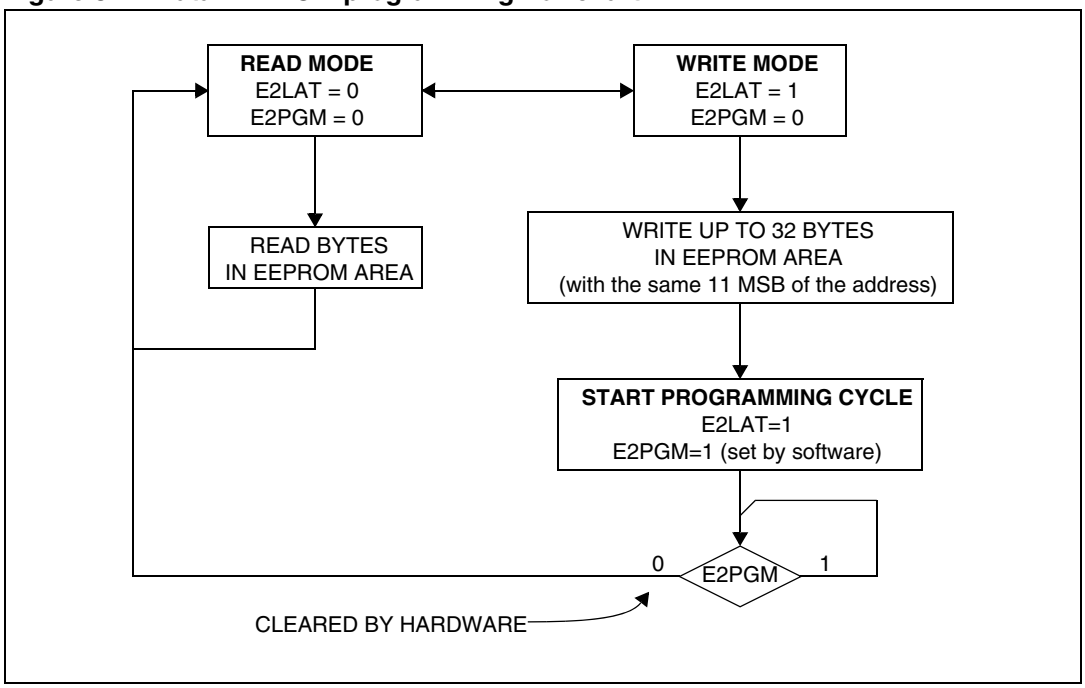
To access the write mode, the E2LAT bit has to be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 32 data latches according to its address.

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



Bit 2 = **N** *Negative*.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It's a copy of the result 7^{th} bit.

0: The result of the last operation is positive or null.
1: The result of the last operation is negative
(that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero*.

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.

Bit 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.

Interrupt management bits

Bits 5,3 = **I1, I0** *Interrupt*

The combination of the I1 and I0 bits gives the current interrupt software priority.

Table 6. Interrupt software priority

Priority	I1	I0
Level 0 (main)	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 (IxSPR). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See the interrupt management chapter for more details.

7.2 Phase locked loop

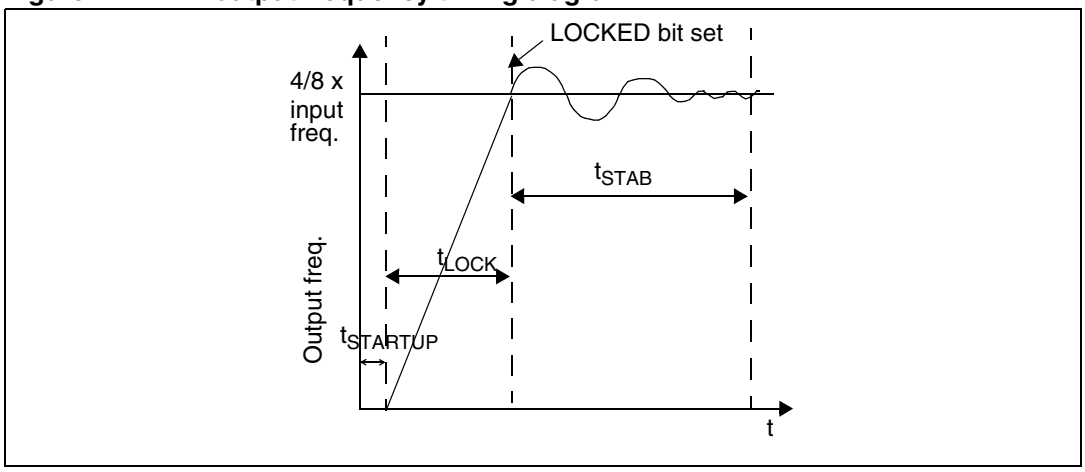
The PLL can be used to multiply a 1 MHz frequency from the RC oscillator or the external clock by 4 or 8 to obtain f_{OSC} of 4 or 8 MHz. The PLL is enabled and the multiplication factor of 4 or 8 is selected by 3 option bits. Refer to *Table 7* for the PLL configuration depending on the required frequency and the application voltage. Refer to *Section 15.1* for the option byte description.

Table 7. PLL configurations

Target ratio	V_{DD}	PLL ratio	DIV2
x4(1)	2.7 V - 3.65 V	x4	OFF
x4	3.3 V - 5.5 V	x8	ON
x8		x8	OFF

1. For a target ratio of x4 between 3.3 V - 3.65 V, this is the recommended configuration.

Figure 14. PLL output frequency timing diagram



When the PLL is started, after reset or wakeup from Halt mode or AWUFH mode, it outputs the clock after a delay of $t_{STARTUP}$

When the PLL output signal reaches the operating frequency, the LOCKED bit in the SICSCR register is set. Full PLL accuracy (ACC_{PLL}) is reached after a stabilization time of t_{STAB} (see *Figure 14*).

Refer to *Section 7.6.5: Register description* for a description of the LOCKED bit in the SICSR register.

Caution: The PLL is not recommended for applications where timing accuracy is required.

Caution: When the RC oscillator and the PLL are enabled, it is recommended to calibrate this clock through the RCCRH and RCCRL registers.

7.5.4 Internal low-voltage detector (LVD) reset

Two different reset sequences caused by the internal LVD circuitry can be distinguished:

- Power-on reset
- Voltage-drop reset

The device $\overline{\text{RESET}}$ pin acts as an output that is pulled low when $V_{DD}<V_{IT+}$ (rising edge) or $V_{DD}<V_{IT-}$ (falling edge) as shown in *Figure 17*.

The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

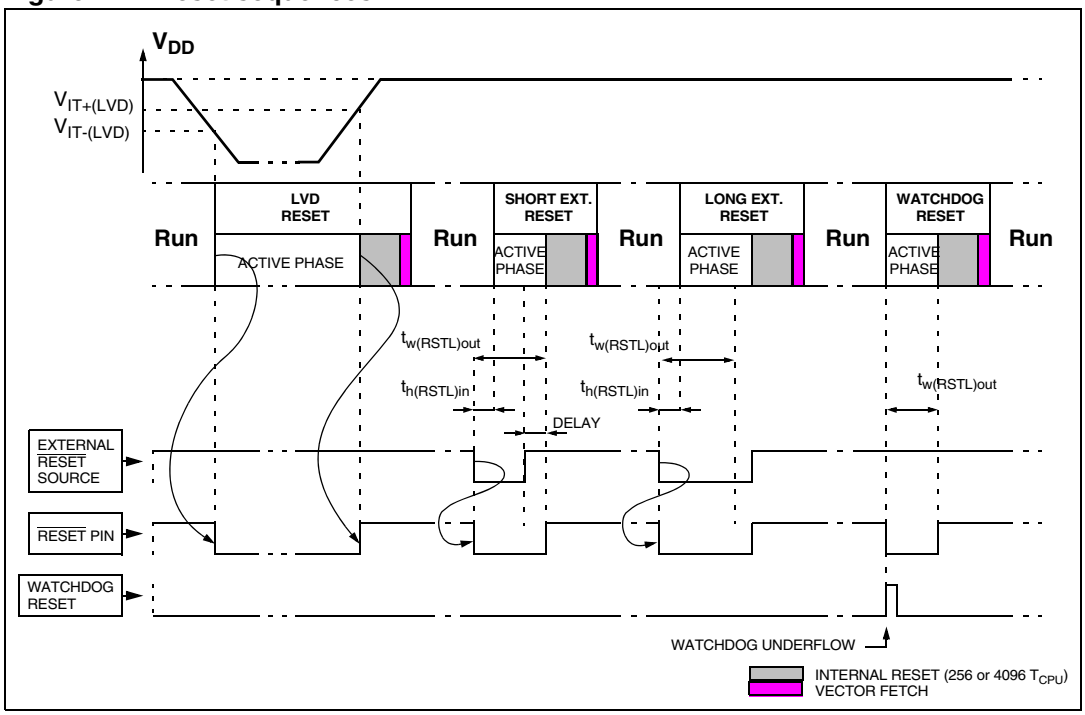
Note: It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from Reset, to ensure the application functions properly.

7.5.5 Internal watchdog reset

The reset sequence generated by a internal Watchdog counter overflow is shown in *Figure 17*.

Starting from the Watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Figure 17. Reset sequences



7.6 System integrity management (SI)

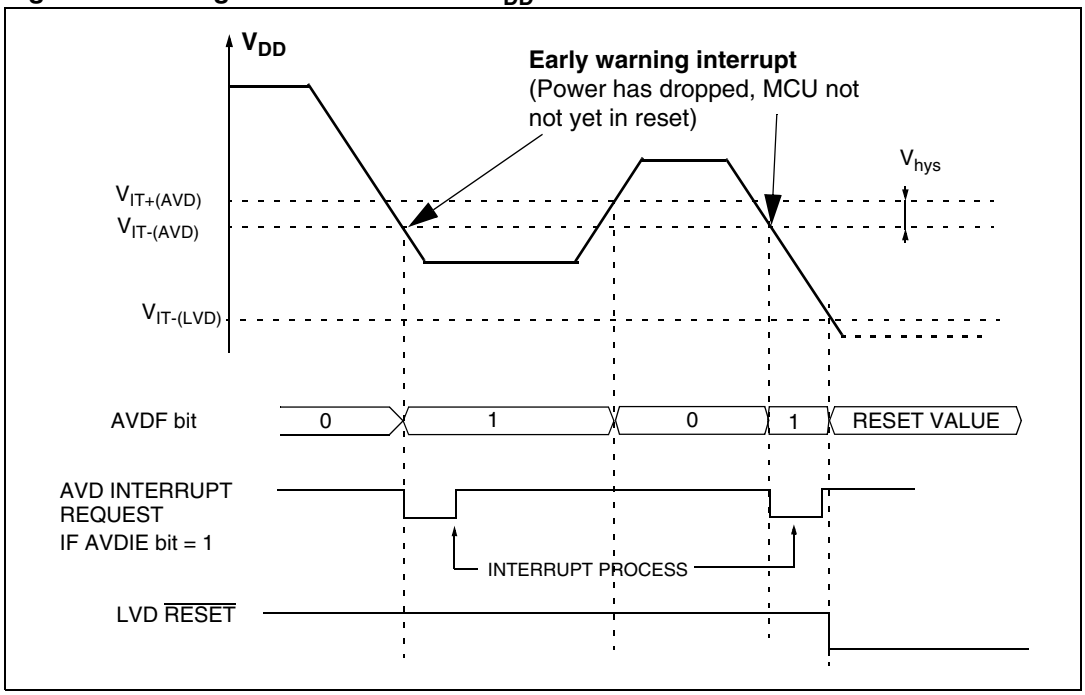
The system integrity management block contains the low-voltage detector (LVD) and auxiliary-voltage detector (AVD) functions. It is managed by the SICSR register.

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Section 12.2.1: Illegal opcode reset for further details.

In the case of a drop in voltage below $V_{IT-(AVD)}$, the AVDF flag is set and an interrupt request is issued.

If V_{DD} rises above the $V_{IT+(AVD)}$ threshold voltage the AVDF bit is cleared automatically by hardware. No interrupt is generated, and therefore software should poll the AVDF bit to detect when the voltage has risen, and resume normal processing.

Figure 19. Using the AVD to monitor V_{DD}



7.6.3 Low-power modes

Table 10. Low-power mode description

Mode	Description
Wait	No effect on SI. AVD interrupts cause the device to exit from Wait mode.
Halt	The SICSR register is frozen.

7.6.4 Interrupts

The AVD interrupt event generates an interrupt if the corresponding AVDIE bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 11. Interrupt event

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
AVD event	AVDF	AVDIE	Yes	No

When an interrupt request is not serviced immediately, it is latched and then processed when its software priority combined with the hardware priority becomes the highest one.

Note: 1 The hardware priority is exclusive while the software one is not. This allows the previous process to succeed with only one interrupt.

2 TLI, reset and TRAP can be considered as having the highest software priority in the decision process.

Different interrupt vector sources

Two interrupt source types are managed by the ST7 interrupt controller: the non-maskable type (reset, TRAP) and the maskable type (external or from internal peripherals).

Non-maskable sources

These sources are processed regardless of the state of the I1 and I0 bits of the CC register (see *Figure 20*). After stacking the PC, X, A and CC registers (except for reset), the corresponding vector is loaded in the PC register and the I1 and I0 bits of the CC are set to disable interrupts (level 3). These sources allow the processor to exit Halt mode.

- TRAP (non-maskable software interrupt)

This software interrupt is serviced when the TRAP instruction is executed. It will be serviced according to the flowchart in *Figure 20*.

- reset

The reset source has the highest priority in the ST7. This means that the first current routine has the highest software priority (level 3) and the highest hardware priority.
See the reset chapter for more details.

Maskable sources

Maskable interrupt vector sources can be serviced if the corresponding interrupt is enabled and if its own interrupt software priority (in ISPRx registers) is higher than the one currently being serviced (I1 and I0 in CC register). If any of these two conditions is false, the interrupt is latched and thus remains pending.

- External interrupts

  External interrupts allow the processor to exit from Halt low-power mode. External interrupt sensitivity is software selectable through the External Interrupt Control register (EICR).
  External interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.
  If several input pins of a group connected to the same interrupt line are selected simultaneously, these will be logically ORed.

- Peripheral interrupts

  Usually, the peripheral interrupts cause the MCU to exit from Halt mode except those mentioned in the “Interrupt Mapping” table. A peripheral interrupt occurs when a specific flag is set in the peripheral status registers and if the corresponding enable bit is set in the peripheral control register.
  The general sequence for clearing an interrupt is based on an access to the status register followed by a read or write to an associated register.

Note: The clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being serviced) will therefore be lost if the clear sequence is executed.

Figure 35. I/O port general block diagram

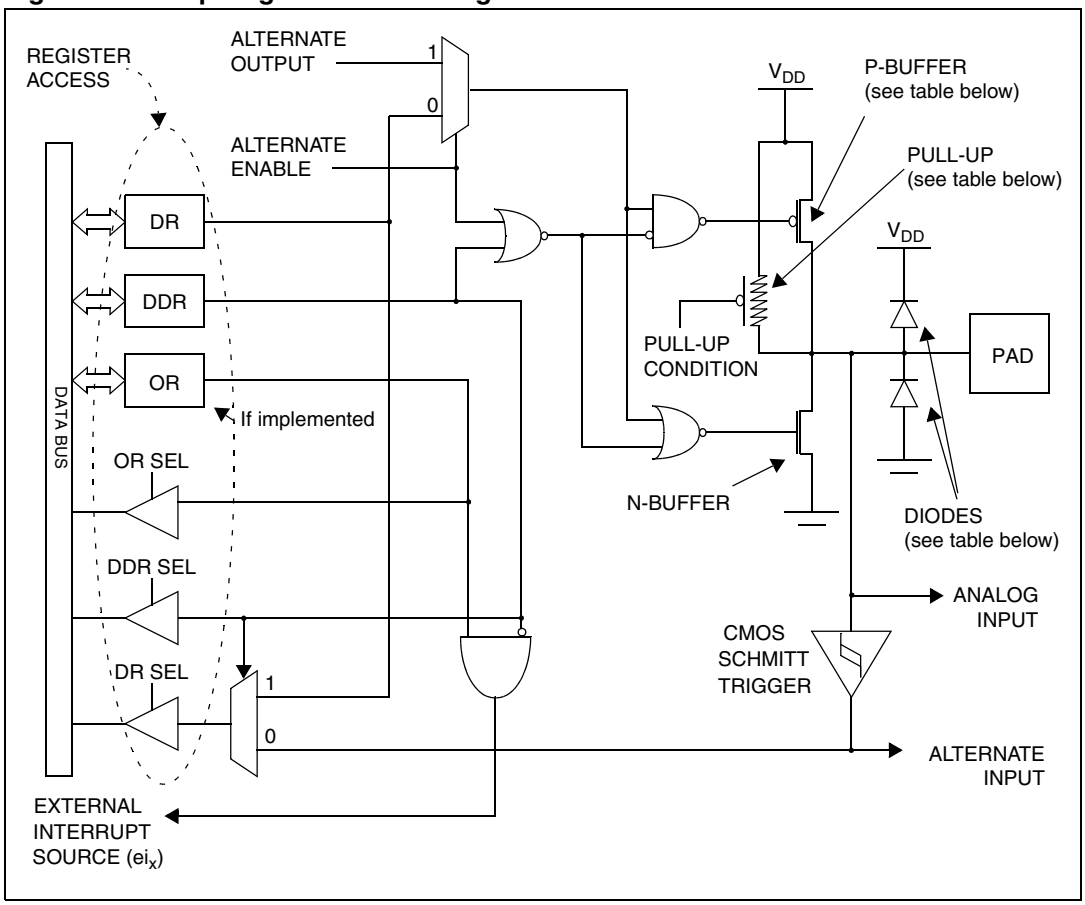


Table 27. I/O Port mode options(1)

Configuration mode		Pull-up	P-Buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without Interrupt	Off	Off	On	On
	Pull-up with/without Interrupt	On			
Output	Push-pull	Off	On		
	Open drain (logic level)		Off		
	True open drain	NI	NI	NI(2)	

1. NI = not implemented, Off = implemented not activated, On = implemented and activated.
2. The diode to VDD is not implemented in the true open drain pads. A local protection between the pad and VSS is implemented to protect the device against positive stress.

11.1.7 Hardware watchdog option

If hardware watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the WDGCR is not used. Refer to the Option Byte description.

11.1.8 Using Halt mode with the WDG (WDGHALT option)

The following recommendation applies if Halt mode is used when the watchdog is enabled.

- Before executing the HALT instruction, refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.

11.1.9 Interrupts

None.

11.1.10 Register description

Control register (WDGCR)

Reset value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0
Read/Write							

Bit 7 = **WDGA** *Activation bit.*

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled

Note: This bit is not used if the hardware watchdog option is enabled by option byte.

Bits 6:0 = **T[6:0]** *7-bit counter (MSB to LSB).*

These bits contain the value of the watchdog counter. It is decremented every 16384 f_{OSC2} cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

Window register (WDGWR)

Reset value: 0111 1111 (7Fh)

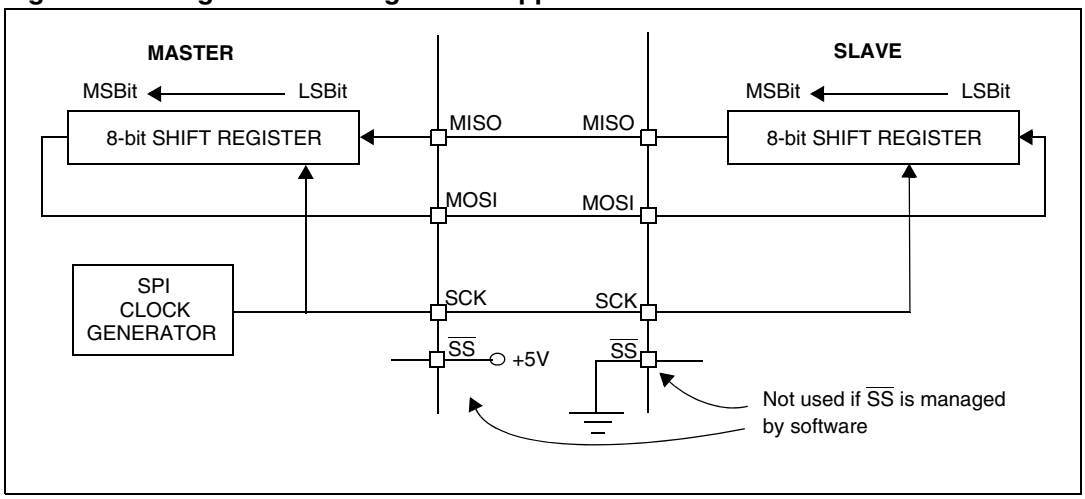
7							0
-	W6	W5	W4	W3	W2	W1	W0
Read/Write							

Bit 7 = Reserved

Bits 6:0 = **W[6:0]** *7-bit window value*

These bits contain the window value to be compared to the downcounter.

Figure 57. Single master/ single slave application



Slave select management

As an alternative to using the $\overline{SS}$ pin to control the Slave Select signal, the application can choose to manage the Slave Select signal by software. This is configured by the SSM bit in the SPICSR register (see *Figure 59*).

In software management, the external $\overline{SS}$ pin is free for other application uses and the internal $\overline{SS}$ signal level is driven by writing to the SSI bit in the SPICSR register.

- In Master mode:
  - $\overline{SS}$ internal must be held high continuously
- In Slave Mode:

  There are two cases depending on the data/clock timing relationship (see *Figure 58*):

  If CPHA = 1 (data latched on second clock edge):
  - $\overline{SS}$ internal must be held low during the entire transmission. This implies that in single slave applications the $\overline{SS}$ pin either can be tied to V_{SS}, or made free for standard I/O by managing the $\overline{SS}$ function by software (SSM = 1 and SSI = 0 in the in the SPICSR register)

  If CPHA = 0 (data latched on first clock edge):
  - $\overline{SS}$ internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If $\overline{SS}$ is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see *Write collision error (WCOL)*).

Table 63. SCI register map and reset values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0050h	SCISR Reset value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
0051h	SCIDR Reset value	MSB x	x	x	x	x	x	x	LSB x
0052h	SCIBRR Reset value	SCP1 0	SCP0 0	SCT2 0	SCT1 0	SCT0 0	SCR2 0	SCR1 0	SCR0 0
0053h	SCICR1 Reset value	R8 x	T8 0	SCID 0	M 0	WAKE 0	PCE 0	PS 0	PIE 0
0054h	SCICR2 Reset value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0
0056h	SCIERPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	SCIPETPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0

Figure 69). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.

- Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

  The SCL line is held low while BTF=1.

  0: Byte transfer not done
  1: Byte transfer succeeded

Bit 2 = **ADSL** *Address matched (Slave mode).*

This bit is set by hardware as soon as the received slave address matched with the OAR register content or a general call is recognized. An interrupt is generated if ITE=1. It is cleared by software reading SR1 register or by hardware when the interface is disabled (PE=0).

The SCL line is held low while ADSL=1.

0: Address mismatched or not received
1: Received address matched

Bit 1 = **M/SL** *Master/Slave.*

This bit is set by hardware as soon as the interface is in Master mode (writing START=1). It is cleared by hardware after detecting a Stop condition on the bus or a loss of arbitration (ARLO=1). It is also cleared when the interface is disabled (PE=0).

0: Slave mode
1: Master mode

Bit 0 = **SB** *Start bit (Master mode).*

This bit is set by hardware as soon as the Start condition is generated (following a write START=1). An interrupt is generated if ITE=1. It is cleared by software reading SR1 register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE=0).

0: No Start condition
1: Start condition generated

I²C status register 2 (SR2)

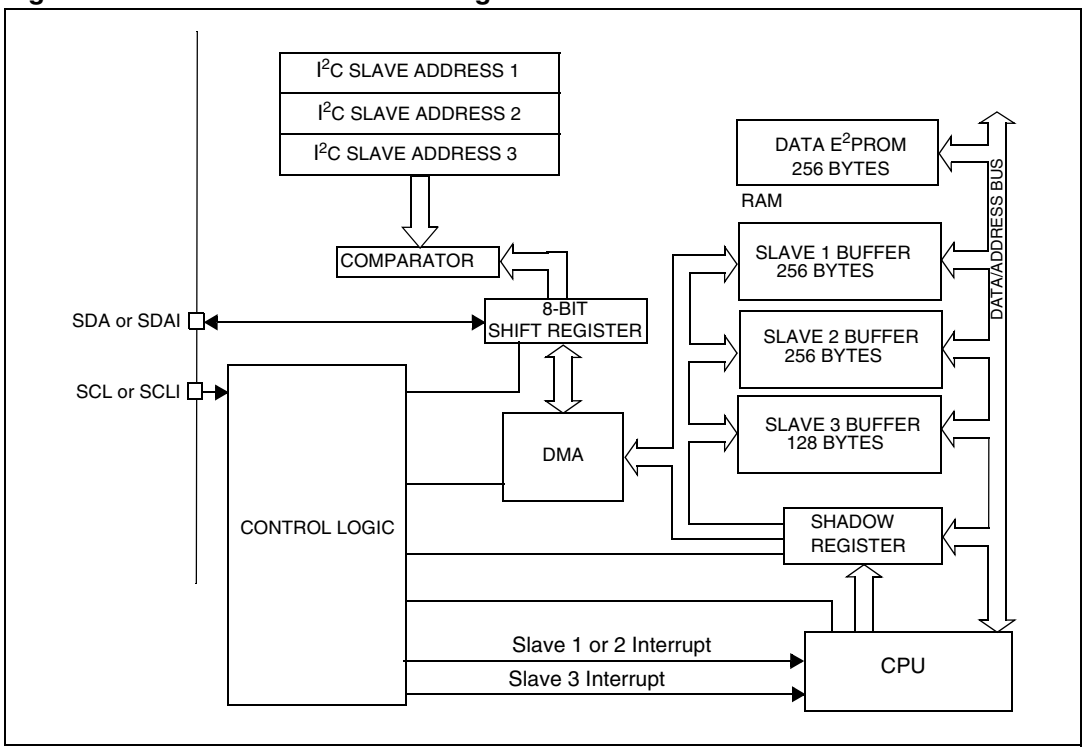
Reset value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	ARLO	BERR	GCAL
Read Only							

Bit 7:5 = Reserved.

Forced to 0 by hardware.

Figure 71. I²C3S interface block diagram



11.7.3 General description

In addition to receiving and transmitting data, I2C3S converts it from serial to parallel format and vice versa. The interrupts are enabled or disabled by software. The I2C3S is connected to the I²C bus by a data pin (SDA) and by a clock pin (SCL). It can be connected both with a standard I²C bus and a Fast I²C bus. The interface operates only in Slave mode as transmitter/receiver.

In order to fully emulate standard I²C EEPROM devices with highest transfer speed, the peripheral prevents I²C clock signal stretching and performs data transfer between the shift register and the RAM buffers using DMA.

Communication flow

A serial data transfer normally begins with a start condition and ends with a stop condition. Both start and stop conditions are generated by an external master. Refer to *Figure 67* for the standard protocol. The I2C3S is not a master and is not capable of generating a start/stop condition on the SDA line. The I2C3S is capable of recognizing 3 slave addresses which are user programmable. The three I²C slave addresses can be individually enabled/disabled by software.

Since the I2C3S interface always acts as a slave, it does not generate a clock. Data and addresses are transferred as 8-bit bytes, MSB first. The first byte following the start condition contains the slave address. A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter.

Bits 3:0 = **CH[3:0]** *Channel Selection*

These bits are set and cleared by software. They select the analog input to convert.

Table 73. Channel selection

Channel pin(1)	CH3	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
Reserved	0	1	1	0
Reserved	0	1	1	1
AIN8	1	0	0	0
Reserved	1	0	0	1
AIN10	1	0	1	0
Reserved	1	0	1	1
AIN12	1	1	0	0
AIN13	1	1	0	1
AIN14	1	1	1	0
AIN15	1	1	1	1

1. The number of channels is device dependent. Refer to the device pinout description.

Data register (ADCDRH)

Reset value: 0000 0000 (00h)

7							0
D9	D8	D7	D6	D5	D4	D3	D2
Read Only							

Bits 7:0 = **D[9:2]** *MSB of Converted Analog Value*

Data register (ADCDRL)

Reset value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	D1	D0
Read Only							

Bits7:2 = Reserved. Forced by hardware to 0.

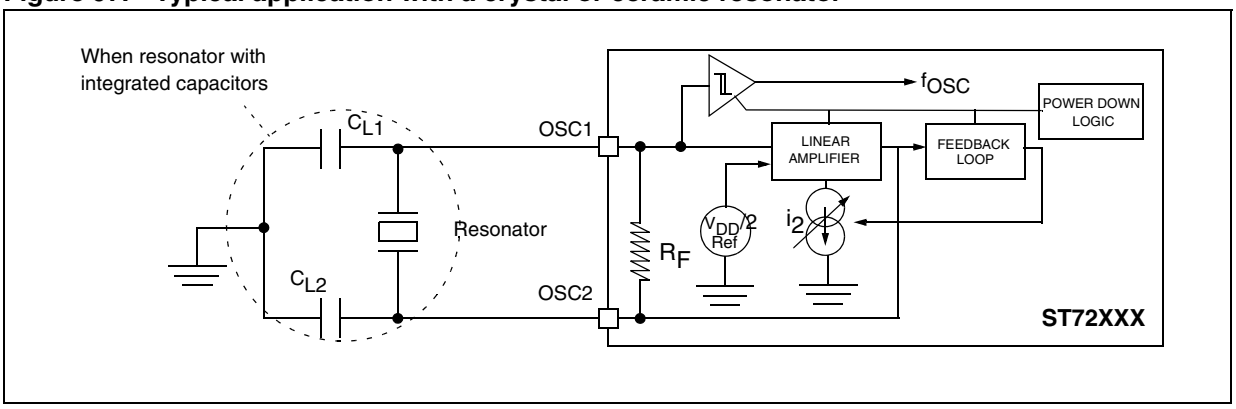
Bits 1:0 = **D[1:0]** *LSB of Converted Analog Value*

Table 99. Recommended load capacitance vs. equivalent serial resistance of ceramic resonator

Supplier	f_{OSC} [MHz]	Typical ceramic resonators (1)		CL1 (2) [pF]	CL2 (2) [pF]	RF [Ω]	Rd [Ω]	Supply voltage range (V)	Temperature range (°C)
		Type(3)	Reference						
Murata	2	SMD	CSTCC2M00G56Z-R0	(47)	(47)	Open	0	2.7 to 5.5	–40 °C to 85 °C
	4	SMD	CSTCR4M00G55Z-R0	(39)	(39)	Open	0		
		LEAD	CSTLS4M00G56Z-B0	(47)	(47)	Open	0		
	8	SMD	CSTCE8M00G52Z-R0	(10)	(10)	Open	0		
		LEAD	CSTLS8M00G53Z-B0	(15)	(15)	Open	0		
	16	SMD	CSTCE16M0V51Z-R0	(5)	(5)	Open	0	3.3 to 5.5	
		LEAD	CSTLS16M0X53Z-B0	(15)	(15)	6.8 k	0	3.4 to 5.5	

1. Resonator characteristics given by the ceramic resonator manufacturer. For more information on these resonators, please consult Murata's web site.
2. () means load capacitor built in resonator.
3. SMD = [-R0: Plastic tape package (Ø =180mm), -B0: Bulk]
LEAD = [-B0: Bulk].

Figure 97. Typical application with a crystal or ceramic resonator



13.8 EMC characteristics

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

13.8.1 Functional EMS (electromagnetic susceptibility)

Based on a simple running application on the product (toggling 2 LEDs through I/O ports), the product is stressed by two electromagnetic events until a failure occurs (indicated by the LEDs).

- **ESD**: Electro-Static 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 100 pF 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 are given in the table below 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 $\overline{\text{RESET}}$ 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).

Table 103. EMS test results

Symbol	Parameter	Conditions	Level/ Class
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD} = 5 V, T_A = +25 °C, f_{OSC} = 8 MHz conforms to IEC 1000-4-2	TBD
V_{FFTB}	Fast transient voltage burst limits to be applied through 100 pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD} = 5 V, T_A = +25 °C, f_{OSC} = 8 MHz conforms to IEC 1000-4-4	TBD

```
LD Y,A; store the level after writing to PxOR/PxDDR
LD A,X; check for falling edge
cp A,#02
jrne OUT
TNZ Y
jrne OUT
LD A,sema; check the semaphore status if edge is detected
CP A,#01
jrne OUT
call call_routine; call the interrupt routine
OUT:LD A,#00
LD sema,A
.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
```

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

```
SIM; set the interrupt mask
LD A,PFDR
AND A,#$02
LD  X,A; store the level before writing to PxOR/PxDDR
LD A,#$90
LD PFDDR,A; Write into PFDDR
LD A,#$ff
LD PFOR,A; Write to PFOR
LD A,PFDR
AND A,#$02
LD Y,A; store the level after writing to PxOR/PxDDR
LD A,X; check for falling edge
cp A,#$02
jrne OUT
TNZ Y
jrne OUT
LD A,#$01
LD sema,A; set the semaphore to '1' if edge is detected
RIM; reset the interrupt mask
LD A,sema; check the semaphore status
CP A,#$01
jrne OUT
call call_routine; call the interrupt routine
RIM
```

16.7.3 Workaround

The sequence to program the EEPROM data (refer to *Section 5.3*) must be executed within C000h-DFFFh area or from the RAM. It is as follows:

```
set E2LAT bit
write up to 32 bytes in E2PROM area
SIM ; to disable the interrupts
set E2PGM bit
wait for E2PGM=0
RIM ; to enable the interrupts
return to the program memory
```