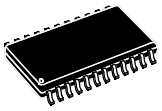
ST7260xx

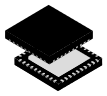
Low speed USB 8-bit MCU family with up to 8K Flash and serial communications interface

Features

- **Memories**
  - 4 or 8 Kbytes program memory: high density Flash (HDFlash), or FastROM with readout and write protection
  - In-application programming (IAP) and in-circuit programming (ICP)
  - 384 bytes RAM memory (128-byte stack)
- **Clock, reset and supply management**
  - Run, Wait, Slow and Halt CPU modes
  - 12 or 24 MHz oscillator
  - RAM Retention mode
  - Optional low voltage detector (LVD)
- **USB (Universal Serial Bus) interface**
  - DMA for low speed applications compliant with USB 1.5 Mbs (version 2.0) and HID specifications (version 1.0)
  - Integrated 3.3 V voltage regulator and transceivers
  - Supports USB DFU class specification
  - Suspend and Resume operations
  - 3 Endpoints with programmable In/Out configuration
- **Up to 19 I/O ports**
  - Up to 8 high sink I/Os (10 mA at 1.3 V)



SO24



QFN40 (6x6)

  - 2 very high sink true open drain I/Os (25 mA at 1.5 V)
  - Up to 8 lines with interrupt capability
- **2 timers**
  - Programmable Watchdog
  - 16-bit Timer with 2 Input Captures, 2 Output Compares, PWM output and clock input
- **Communications interface**
  - Asynchronous serial communications interface (SCI)
- **Instruction set**
  - 63 basic instructions
  - 17 main addressing modes
  - 8 x 8 unsigned multiply instruction
- **Development tools**
  - Versatile development tools including , software library, hardware emulator, programming boards, HID and DFU software layer

Table 1. Device summary

Features	ST7260K2	ST7260K1	ST7260E2	ST7260E1
Flash program memory - bytes	8 K	4 K	8 K	4 K
RAM (stack) - bytes	384 (128)			
Peripherals	Watchdog timer, 16-bit timer, USB, SCI			
Operating supply	4.0 V to 5.5 V			
CPU frequency	8 MHz (with 24 MHz oscillator) or 4 MHz (with 12 MHz oscillator)			
Operating temperature	0 °C to +70 °C			
Packages	QFN40 (6x6)		SO24	

Contents

1 Description 7

2 Block diagram 8

3 Pin description 9

4 Register & memory map 14

5 Flash program memory 17

5.1 Introduction 17

5.2 Main features 17

5.3 Structure 17

5.3.1 Readout protection 18

5.4 ICC interface 19

5.5 ICP (in-circuit programming) 20

5.6 IAP (in-application programming) 20

5.7 Related documentation 20

5.7.1 Flash control/status register (FCSR) 20

6 Central processing unit (CPU) 21

6.1 Introduction 21

6.2 Main features 21

6.3 CPU registers 21

6.3.1 Accumulator (A) 22

6.3.2 Index registers (X and Y) 22

6.3.3 Program counter (PC) 22

6.3.4 Condition code register (CC) 22

6.3.5 Stack pointer register (SP) 24

7 Reset and clock management 25

7.1 Reset 25

7.2 Low voltage detector (LVD) 25

7.2.1 Watchdog reset 25

7.2.2 External reset 25

7.3 Clock system 27
7.3.1 General description 27
7.3.2 External clock 28

8 Interrupts 30
8.0.1 Interrupt register (ITRFRE) 32

9 Power saving modes 33
9.1 Introduction 33
9.2 Halt mode 33
9.3 Slow mode 34
9.4 Wait mode 34

10 I/O ports 36
10.1 Introduction 36
10.2 Functional description 36
10.2.1 Port A 37
10.2.2 Port B 39
10.2.3 Port C 40
10.2.4 Register description 41
10.2.5 Data register (PxDR) 41
10.2.6 Data direction register (PxDDR) 42
10.2.7 Related documentation 42

11 Miscellaneous register 43

12 Watchdog timer (WDG) 44
12.1 Introduction 44
12.2 Main features 44
12.3 Functional description 44
12.3.1 Software watchdog option 45
12.3.2 Hardware watchdog option 45
12.3.3 Low power modes 46
12.3.4 Using Halt mode with the WDG (option) 46
12.3.5 Interrupts 46
12.3.6 Control register (WDGCR) 47

12.4 16-bit timer 48
12.4.1 Introduction 48
12.4.2 Main features 48
12.4.3 Functional description 49
12.4.4 Low power modes 62
12.4.5 Interrupts 62
12.4.6 Summary of timer modes 63
12.4.7 16-bit timer registers 63

13 Serial communications interface (SCI) 71
13.1 Introduction 71
13.2 Main features 71
13.2.1 General description 72
13.2.2 Functional description 73
13.2.3 Low power modes 81
13.2.4 Interrupts 81
13.3 Register description 82
13.3.1 Status register (SCISR) 82
13.3.2 Control register 1 (SCICR1) 84
13.3.3 Control register 2 (SCICR2) 85
13.3.4 Data register (SCIDR) 86
13.3.5 Baud rate register (SCIBRR) 87

14 USB interface (USB) 89
14.1 Introduction 89
14.2 Main features 89
14.3 Functional description 89
14.4 Register description 90
14.4.1 DMA address register (DMAR) 90
14.4.2 Interrupt/DMA register (IDR) 90
14.4.3 PID register (PIDR) 91
14.4.4 Interrupt status register (ISTR) 92
14.4.5 Interrupt mask register (IMR) 93
14.4.6 Control register (CTLR) 94
14.4.7 Device address register (DADDR) 95
14.4.8 Endpoint n register A (EPnRA) 95

14.4.9 Endpoint n register B (EPnRB) . . . 96
14.4.10 Endpoint 0 register B (EP0RB) . . . 97
14.5 Programming considerations . . . 97
14.5.1 Initializing the registers . . . 97
14.5.2 Initializing DMA buffers . . . 98
14.5.3 Endpoint initialization . . . 98
14.5.4 Interrupt handling . . . 98

15 Instruction set . . . 101
15.1 ST7 addressing modes . . . 101
15.1.1 Inherent . . . 102
15.1.2 Immediate . . . 103
15.1.3 Direct . . . 103
15.1.4 Indexed (no offset, short, long) . . . 103
15.1.5 Indirect (short, long) . . . 104
15.1.6 Indirect indexed (short, long) . . . 104
15.1.7 Relative mode (direct, indirect) . . . 105
15.2 Instruction groups . . . 105

16 Electrical characteristics . . . 109
16.1 Parameter conditions . . . 109
16.1.1 Minimum and maximum values . . . 109
16.1.2 Typical values . . . 109
16.1.3 Typical curves . . . 109
16.1.4 Loading capacitor . . . 109
16.1.5 Pin input voltage . . . 109
16.2 Absolute maximum ratings . . . 111
16.3 Operating conditions . . . 112
16.3.1 General operating conditions . . . 112
16.3.2 Operating conditions with low voltage detector (LVD) . . . 113
16.4 Supply current characteristics . . . 113
16.5 Clock and timing characteristics . . . 114
16.5.1 General timings . . . 114
16.5.2 Control timing characteristics . . . 114
16.5.3 External clock source . . . 116
16.6 Memory characteristics . . . 117

16.6.1 RAM and hardware registers . . . 117
16.6.2 Flash memory . . . 117
16.7 EMC characteristics . . . 118
16.7.1 Functional EMS (electromagnetic susceptibility) . . . 118
16.7.2 Designing hardened software to avoid noise problems . . . 118
16.7.3 Electro magnetic interference (EMI) . . . 119
16.7.4 Absolute maximum ratings (electrical sensitivity) . . . 119
16.7.5 Electrostatic discharge (ESD) . . . 119
16.8 I/O port pin characteristics . . . 121
16.8.1 General characteristics . . . 121
16.8.2 Output driving current . . . 123
16.9 Control pin characteristics . . . 125
16.9.1 Asynchronous $\overline{\text{RESET}}$ pin . . . 125
16.9.2 USB - universal bus interface . . . 127
16.9.3 SCI - serial communications interface . . . 128

17 Package characteristics . . . 129
17.1 Package mechanical data . . . 129
17.1.1 Thermal characteristics . . . 130

18 Device configuration and ordering information . . . 131
18.1 Option byte . . . 131
18.2 Device ordering information and transfer of customer code . . . 132
18.3 Development tools . . . 132

19 Known limitations . . . 135
19.1 PA2 limitation with OCMP1 enabled . . . 135
19.2 Unexpected reset fetch . . . 135
19.3 SCI wrong break duration . . . 135

20 Device marking . . . 137

21 Revision history . . . 138

1 Description

The ST7260xx devices are members of the ST7 microcontroller family designed for USB applications running from 4.0 to 5.5 V. Different package options offer up to 19 I/O pins.

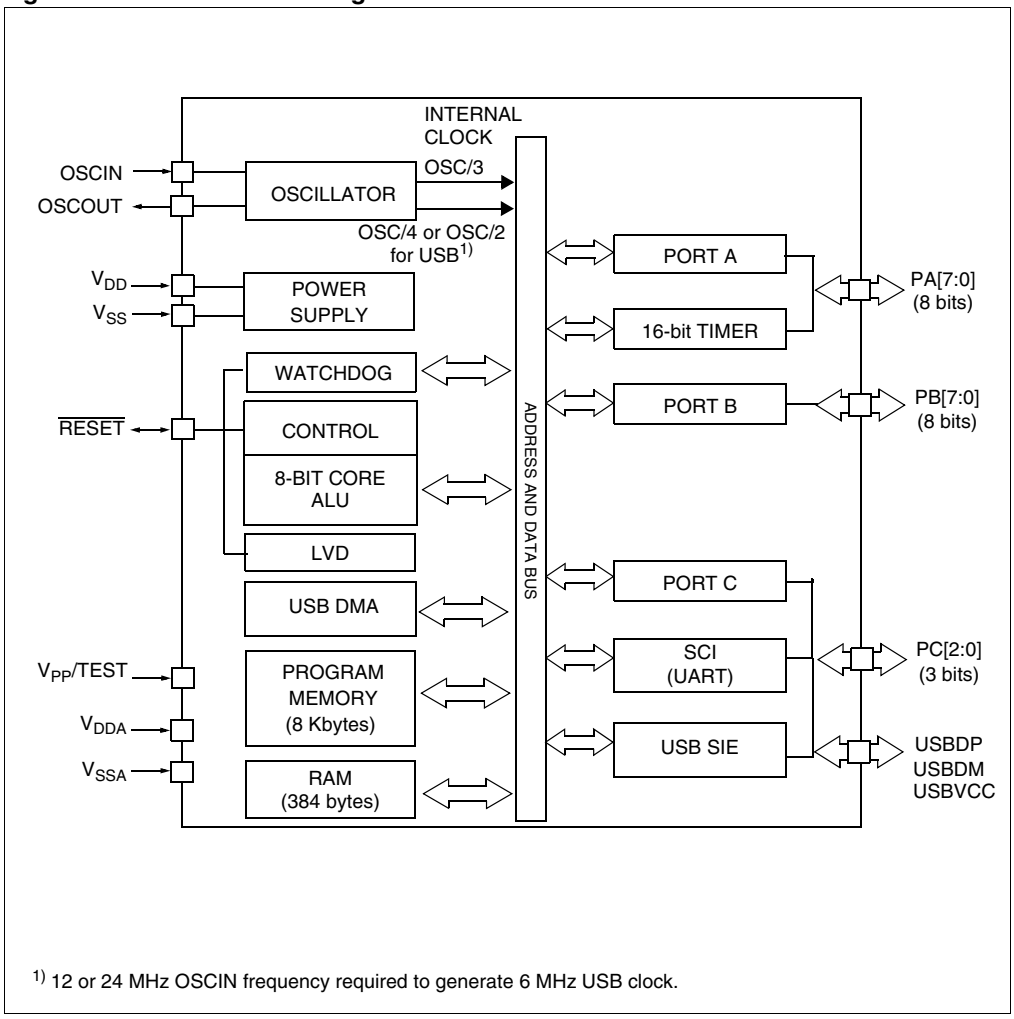
All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set and are available with Flash program memory. The ST7 family architecture offers both power and flexibility to software developers, enabling the design of highly efficient and compact application code.

The on-chip peripherals include a low speed USB interface and an asynchronous SCI interface. For power economy, the microcontroller can switch dynamically into, Slow, Wait, Active Halt or Halt mode when the application is in idle or stand-by state.

Typical applications include consumer, home, office and industrial products.

2 Block diagram

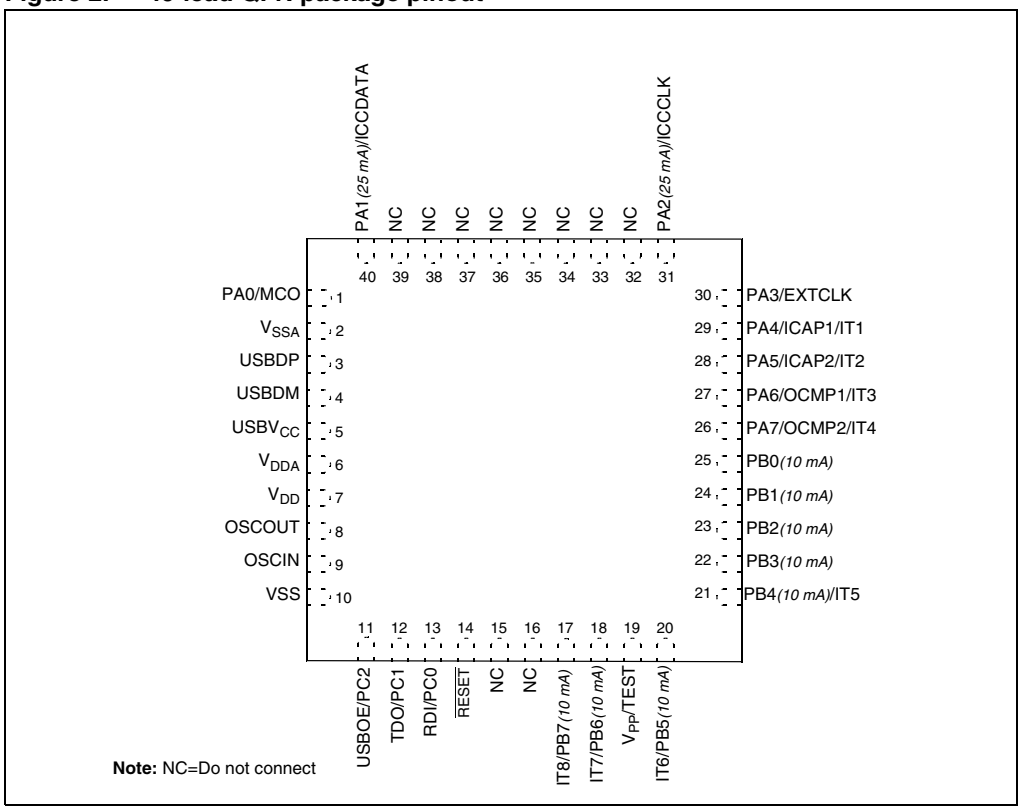
Figure 1. General block diagram



1) 12 or 24 MHz OSCIN frequency required to generate 6 MHz USB clock.

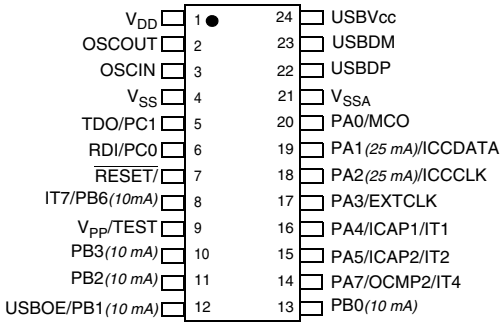
3 Pin description

Figure 2. 40-lead QFN package pinout



Note: NC=Do not connect

Figure 3. 24-pin SO package pinout



$\overline{\text{RESET}}$ (see Note 1): Bidirectional. This active low signal forces the initialization of the MCU. This event is the top priority non maskable interrupt. This pin is switched low when the Watchdog is triggered or the V_{DD} is low. It can be used to reset external peripherals.

OSCIN/OSCOUT: Input/Output Oscillator pin. These pins connect a parallel-resonant crystal, or an external source, to the on-chip oscillator.

V_{DD}/V_{SS} (see Note 2): Main power supply and ground voltages.

V_{DDA}/V_{SSA} (see Note 2): Power supply and ground voltages for analog peripherals.

Alternate functions: Several pins of the I/O ports assume software programmable alternate functions as shown in the pin description.

Note: 1 Note 1: Adding two 100 nF decoupling capacitors on the Reset pin (respectively connected to VDD and VSS) will significantly improve product electromagnetic susceptibility performance.

2 To enhance the reliability of operation, it is recommended that V_{DDA} and V_{DD} be connected together on the application board. This also applies to V_{SSA} and V_{SS}.

3 The USBOE alternate function is mapped on Port C2 in QFN40 devices. In SO24 devices it is mapped on Port B1.

4 The timer OCMP1 alternate function is mapped on Port A6 in QFN40 pin devices. In SO24 devices it is not available.

Legend / abbreviations for *Figure 2*, *Figure 3* and *Table 2*, *Table 3*:

Type: I = input, O = output, S = supply

In/Output level: CT = CMOS 0.3 V_{DD} / 0.7 V_{DD} with input trigger

Output level: 10 mA = 10 mA high sink (Fn N-buffer only)

25 mA = 25 mA very high sink (on N-buffer only)

Port and control configuration:

- Input: float = floating, wpu = weak pull-up, int = interrupt
- Output: OD = open drain, PP = push-pull, T = True open drain

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

Table 2. Device pin description (QFN40)

Pin n°	Pin name	Type	Level		Port / control					Main function (after reset)	Alternate function
			Input	Output	Input			Output			
					float	wpu	int	OD	PP		
1	PA0/MCO	I/O	CT				X		X	Port A0	Main Clock Output
2	V_{SSA}	S								Analog ground	
3	USBDP	I/O								USB bidirectional data (data +)	
4	USBDM	I/O								USB bidirectional data (data -)	
5	USBVCC	O								USB power supply	
6	V_{DDA}	S								Analog supply voltage	
7	V_{DD}	S								Power supply voltage (4V - 5.5V)	
8	OSCOUT	O								Oscillator output	
9	OSCIN	I								Oscillator input	
10	V_{SS}	S								Digital ground	
11	PC2/USBOE	I/O	CT			X			X	Port C2	USB Output Enable
12	PC1/TDO	I/O	CT			X			X	Port C1	SCI Transmit Data Output
13	PC0/RDI	I/O	CT			X			X	Port C0	SCI Receive Data Input
14	RESET	I/O				X		X		Reset	
15	NC	--								Not connected	
16	NC	--								Not connected	
17	PB7/IT8	I/O	CT	10 mA	X		X		X	Port B7	
18	PB6/IT7	I/O	CT	10 mA	X		X		X	Port B6	
19	V_{PP}/TEST	S								Programming supply	
20	PB5/IT6	I/O	CT	10 mA	X		X		X	Port B5	
21	PB4/IT5	I/O	CT	10 mA	X		X		X	Port B4	
22	PB3	I/O	CT	10 mA	X				X	Port B3	
23	PB2	I/O	CT	10 mA	X				X	Port B2	
24	PB1	I/O	CT	10 mA	X				X	Port B1	
25	PB0	I/O	CT	10 mA	X				X	Port B0	
26	PA7/OCMP2/IT4	I/O	CT			X	X		X	Port A7	Timer Output Compare 2
27	PA6/OCMP1/IT3	I/O	CT			X	X		X	Port A6	Timer Output Compare 1
28	PA5/ICAP2/IT2	I/O	CT			X	X		X	Port A5	Timer Input Capture 2
29	PA4/ICAP1/IT1	I/O	CT			X	X		X	Port A4	Timer Input Capture 1

Table 2. Device pin description (QFN40) (continued)

Pin n°	Pin name	Type	Level		Port / control					Main function (after reset)	Alternate function
			Input	Output	Input			Output			
					float	wpu	int	OD	PP		
30	PA3/EXTCLK	I/O	CT			X			X	Port A3	Timer External Clock
31	PA2/ICCCLK	I/O	C_T	25 mA	X			T		Port A2	ICC Clock
32	NC	--								Do not connect	
33	NC	--								Do not connect	
34	NC	--								Do not connect	
35	NC	--								Do not connect	
36	NC	--								Do not connect	
37	NC	--								Do not connect	
38	NC	--								Do not connect	
39	NC	--								Do not connect	
40	PA1/ICCDATA	I/O	CT	25 mA	X			T		Port A1	ICC Data

Table 3. Device pin description (SO24)

Pin n°	Pin name	Type	Level		Port / control					Main function (after reset)	Alternate function
			Input	Output	Input			Output			
					float	wpu	int	OD	PP		
1	V_{DD}	S								Power supply voltage (4 V - 5.5 V)	
2	OSCOUT	O								Oscillator output	
3	OSCIN	I								Oscillator input	
4	V_{SS}	S								Digital ground	
5	PC1/TDO	I/O	CT			X			X	Port C1	SCI Transmit Data Output
6	PC0/RDI	I/O	CT			X			X	Port C0	SCI Receive Data Input
7	RESET	I/O				X		X		Reset	
8	PB6/IT7	I/O	CT	10 mA	X		X		X	Port B6	
9	V_{PP}/TEST	S								Programming supply	
10	PB3	I/O	CT	10 mA	X				X	Port B3	
11	PB2	I/O	CT	10 mA	X				X	Port B2	
12	PB1/USBOE	I/O	CT	10 mA	X				X	Port B1	USB Output Enable
13	PB0	I/O	CT	10 mA	X				X	Port B0	
14	PA7/OCMP2/IT4	I/O	CT			X	X		X	Port A7	Timer Output Compare 2
15	PA5/ICAP2/IT2	I/O	CT			X	X		X	Port A5	Timer Input Capture 2
16	PA4/ICAP1/IT1	I/O	CT			X	X		X	Port A4	Timer Input Capture 1
17	PA3/EXTCLK	I/O	CT			X			X	Port A3	Timer External Clock
18	PA2/ICCCLK	I/O	C_T	25 mA	X			T		Port A2	ICC Clock
19	PA1/ICCDATA	I/O	CT	25 mA	X			T		Port A1	ICC Data
20	PA0/MCO	I/O	CT				X		X	Port A0	Main Clock Output
21	V_{SSA}	S								Analog ground	
22	USBDP	I/O								USB bidirectional data (data +)	
23	USBDM	I/O								USB bidirectional data (data -)	
24	USBVCC	O								USB power supply	

4 Register & memory map

As shown in *Figure 4*, the MCU is capable of addressing 8 Kbytes of memories and I/O registers.

The available memory locations consist of up to 384 bytes of RAM including 64 bytes of register locations, and up to 8 Kbytes of user program memory in which the upper 32 bytes are reserved for interrupt vectors. The RAM space includes up to 128 bytes for the stack from 0100h to 017Fh.

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

Note: ***Important****: memory locations noted "Reserved" must never be accessed. Accessing a reserved area can have unpredictable effects on the device.*

Figure 4. Memory map

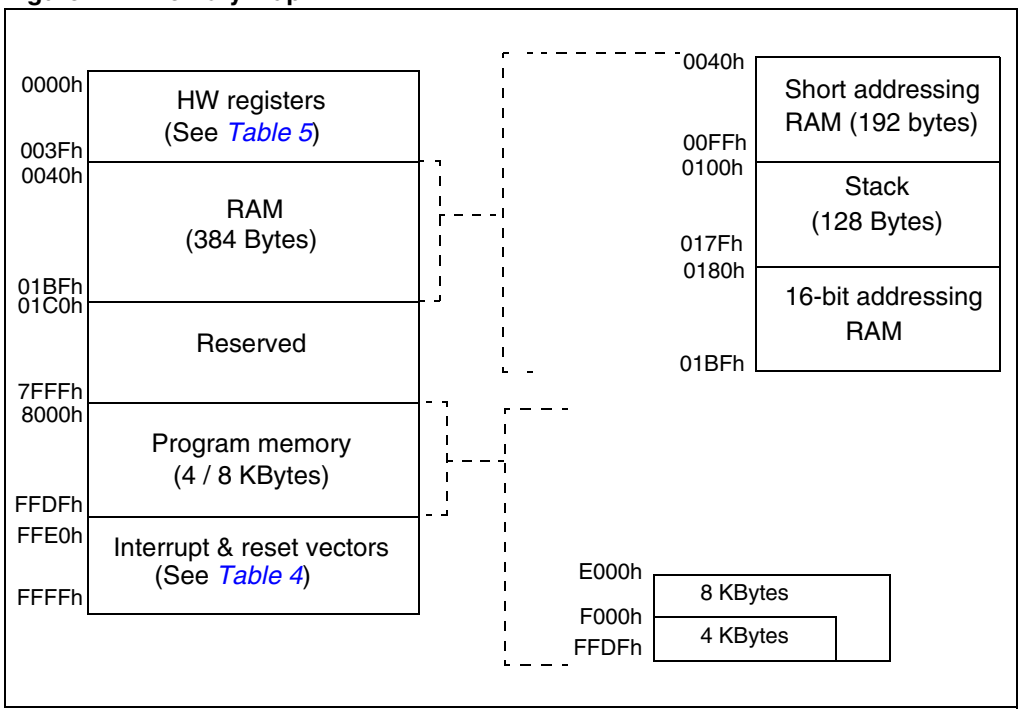


Table 4. Interrupt vector map

Vector address	Description	Masked by	Remarks	Exit from Halt mode
FFE0h-FFEDh	Reserved area			
FFEEh-FFEFh	USB interrupt vector	I- bit	Internal interrupt	No
FFF0h-FFF1h	SCI interrupt vector	I- bit	Internal interrupt	No
FFF2h-FFF3h	Reserved area			
FFF4h-FFF5h	TIMER interrupt vector	I- bit	Internal interrupt	No
FFF6h-FFF7h	IT1 to IT8 interrupt vector	I- bit	External interrupt	Yes
FFF8h-FFF9h	USB end suspend mode interrupt vector	I- bit	External interrupts	Yes
FFFAh-FFFBh	Flash start programming interrupt vector	I- bit	Internal interrupt	Yes
FFFCh-FFFDh	TRAP (software) interrupt vector	None	CPU interrupt	No
FFFEh-FFFFh	RESET vector	None		Yes

Table 5. Hardware register memory map

Address	Block	Register label	Register name	Reset status	Remarks
0000h 0001h	Port A	PADR PADDR	Port A Data Register Port A Data Direction Register	00h 00h	R/W R/W
0002h 0003h	Port B	PBDR PBDDR	Port B Data Register Port B Data Direction Register	00h 00h	R/W R/W
0004h 0005h	Port C	PCDR PCDDR	Port C Data Register Port C Data Direction Register	1111 x000b 1111 x000b	R/W R/W
0006h to 0007h	Reserved (2 bytes)				
0008h	ITC	ITIFRE	Interrupt Register	00h	R/W
0009h	MISC	MISCR	Miscellaneous Register	00h	R/W
000Ah to 000Bh	Reserved (2 bytes)				
000Ch	WDG	WDGCR	Watchdog Control Register	7Fh	R/W
000Dh to 0010h	Reserved (4 bytes)				

Table 5. Hardware register memory map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0011h	TIM	TCR2	Timer Control Register 2	00h	R/W
0012h		TCR1	Timer Control Register 1	00h	R/W
0013h		TCSR	Timer Control/Status Register	00h	R/W
0014h		TIC1HR	Timer Input Capture High Register 1	xxh	Read only
0015h		TIC1LR	Timer Input Capture Low Register 1	xxh	Read only
0016h		TOC1HR	Timer Output Compare High Register 1	80h	R/W
0017h		TOC1LR	Timer Output Compare Low Register 1	00h	R/W
0018h		TCHR	Timer Counter High Register	FFh	Read only
0019h		TCLR	Timer Counter Low Register	FCh	R/W
001Ah		TACHR	Timer Alternate Counter High Register	FFh	Read only
001Bh		TACLR	Timer Alternate Counter Low Register	FCh	R/W
001Ch		TIC2HR	Timer Input Capture High Register 2	xxh	Read only
001Dh		TIC2LR	Timer Input Capture Low Register 2	xxh	Read only
001Eh		TOC2HR	Timer Output Compare High Register 2	80h	R/W
001Fh		TOC2LR	Timer Output Compare Low Register 2	00h	R/W
0020h	SCI	SCISR	SCI Status Register	C0h	Read only
0021h		SCIDR	SCI Data Register	xxh	R/W
0022h		SCIBRR	SCI Baud Rate Register	00h	R/W
0023h		SCICR1	SCI Control Register 1	x000 0000b	R/W
0024h		SCICR2	SCI Control Register 2	00h	R/W
0025h	USB	USBPIDR	USB PID Register	x0h	Read only
0026h		USBDMAR	USB DMA address Register	xxh	R/W
0027h		USBIDR	USB Interrupt/DMA Register	x0h	R/W
0028h		USBISTR	USB Interrupt Status Register	00h	R/W
0029h		USBIMR	USB Interrupt Mask Register	00h	R/W
002Ah		USBCTLR	USB Control Register	06h	R/W
002Bh		USBDADDR	USB Device Address Register	00h	R/W
002Ch		USBEP0RA	USB Endpoint 0 Register A	0000 xxxxb	R/W
002Dh		USBEP0RB	USB Endpoint 0 Register B	80h	R/W
002Eh		USBEP1RA	USB Endpoint 1 Register A	0000 xxxxb	R/W
002Fh		USBEP1RB	USB Endpoint 1 Register B	0000 xxxxb	R/W
0030h		USBEP2RA	USB Endpoint 2 Register A	0000 xxxxb	R/W
0031h		USBEP2RB	USB Endpoint 2 Register B	0000 xxxxb	R/W
0032h to 0036h	Reserved (5 Bytes)				
0037h	Flash	FCSR	Flash Control /Status Register	00h	R/W
0038h to 003Fh	Reserved (8 bytes)				

5 Flash program memory

5.1 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.

5.2 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

5.3 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 6*). 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 5*). 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 6. Sectors available in Flash devices

Flash size (bytes)	Available sectors
4K	Sector 0
8K	Sectors 0, 1
>8K	Sectors 0, 1, 2

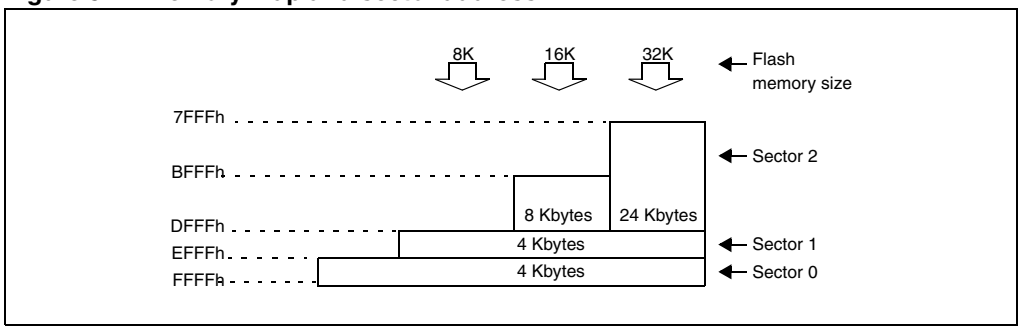
5.3.1 Readout protection

Readout protection, when selected, provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller.

In Flash devices, this protection is removed by reprogramming the option. In this case, the entire program memory is first automatically erased.

Readout protection is enabled and removed through the FMP_R bit in the option byte.

Figure 5. Memory map and sector address

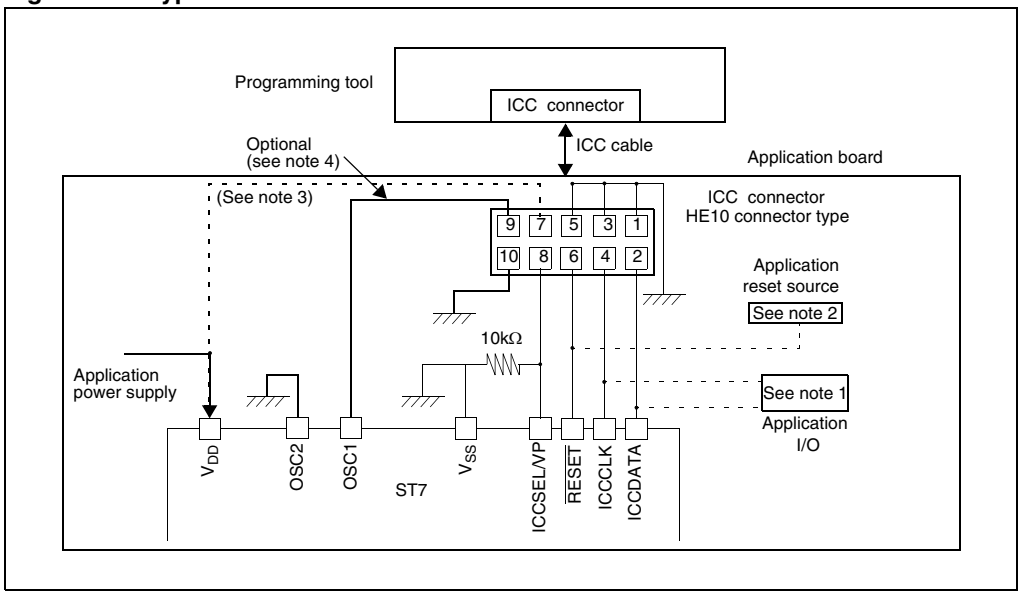


5.4 ICC interface

ICC needs a minimum of 4 and up to 6 pins to be connected to the programming tool (see *Figure 6*). These pins are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input/output serial data pin
- ICCSEL/V_{PP}: programming voltage
- OSC1 (or OSCIN): main clock input for external source (optional)
- V_{DD}: application board power supply (see *Figure 6*, Note 3).

Figure 6. Typical ICC interface



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 ICC 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 5mA at high level (PUSH-pull output or pull-up resistor <1K). A schottky diode can be used to isolate the application reset circuit in this case. When using a classical RC network with R>1K or a reset management IC with open drain output and pull-up resistor >1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.
3. The use of Pin 7 of the ICC connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.
4. Pin 9 has to be connected to the OSC1 (OSCIN) pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.

5.5 ICP (in-circuit programming)

To perform ICP the microcontroller must be switched to ICC (in-circuit communication) mode by an external controller or programming tool.

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

When using an STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user needs only to implement the ICP hardware interface on the application board (see *Figure 6*). For more details on the pin locations, refer to the device pinout description.

5.6 IAP (in-application programming)

This mode uses a BootLoader program previously stored in Sector 0 by the user (in ICP mode or by plugging the device in a programming tool).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (such as user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored). For example, it is possible to download code from the SCI, or USB interface and program it in the Flash. IAP mode can be used to program any of the Flash sectors except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

5.7 Related documentation

For details on Flash programming and ICC protocol, refer to the *ST7 Flash Programming Reference Manual* and to the *ST7 ICC Protocol Reference Manual*.

5.7.1 Flash control/status register (FCSR)

This register is reserved for use by programming tool software. It controls the Flash programming and erasing operations.

FCSR Reset value:0000 0000 (00h)

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

Table 7. Flash control/status register address and reset value

Address (Hex)	Register label	7	6	5	4	3	2	1	0
0037h	FCSR reset value	0	0	0	0	0	0	0	0

6 Central processing unit (CPU)

6.1 Introduction

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

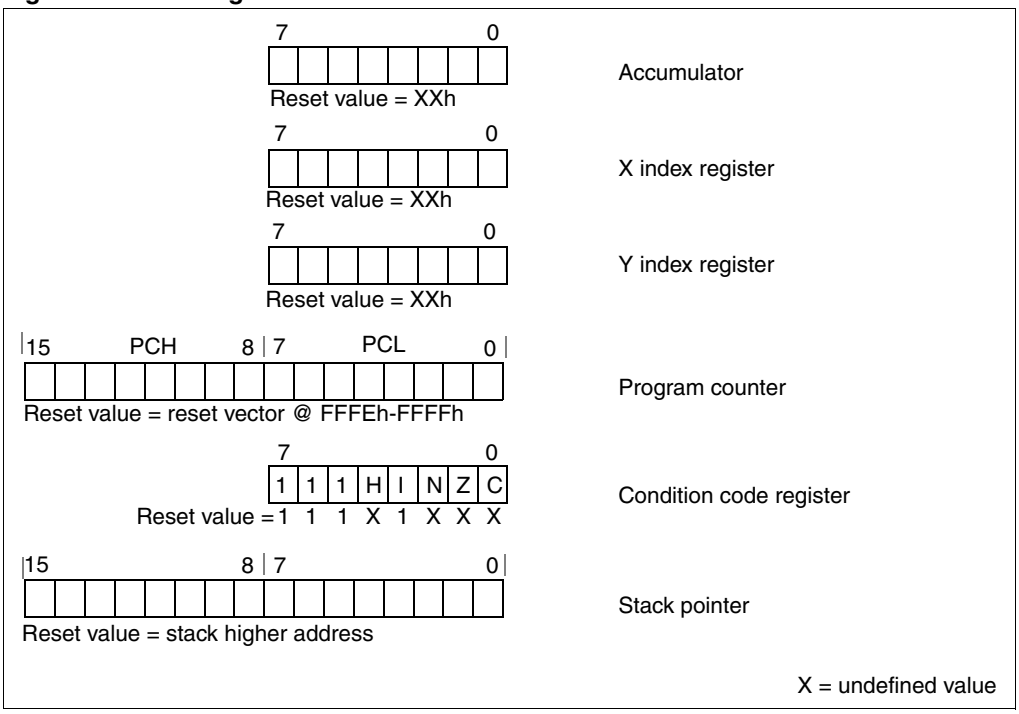
6.2 Main features

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

6.3 CPU registers

The six CPU registers shown in *Figure 7* are not present in the memory mapping and are accessed by specific instructions.

Figure 7. CPU registers



6.3.1 Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

6.3.2 Index registers (X and Y)

In indexed addressing modes, these 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

6.3.3 Program counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

6.3.4 Condition code register (CC)

The 8-bit Condition Code register contains the interrupt mask and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions. These bits can be individually tested and/or controlled by specific instructions.

CC Reset value: 111x1xxx

7	6	5	4	3	2	1	0
1	1	1	H	I	N	Z	C
			R/W	R/W	R/W	R/W	R/W

Table 8. CC register description

Bit	Name	Function
4	H	Half carry This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions. 0: No half carry has occurred. 1: A half carry has occurred. This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Table 8. CC register description

Bit	Name	Function
3	I	Interrupt mask This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software. 0: Interrupts are enabled. 1: Interrupts are disabled. This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions. **Note:** Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptible because the I bit is set by hardware at the start of the routine and reset by the IRET instruction at the end of the routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine
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 is a copy of the result 7th 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.
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.

6.3.5 Stack pointer register (SP)

SP — Reset value: 01 7Fh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	SP6	SP5	SP4	SP3	SP2	SP1	SP0
									R/W	R/W	R/W	R/W	R/W	R/W	R/W

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see *Figure 8*).

Since the stack is 128 bytes deep, the 9 most significant bits are forced by hardware. Following an MCU reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by an LD instruction.

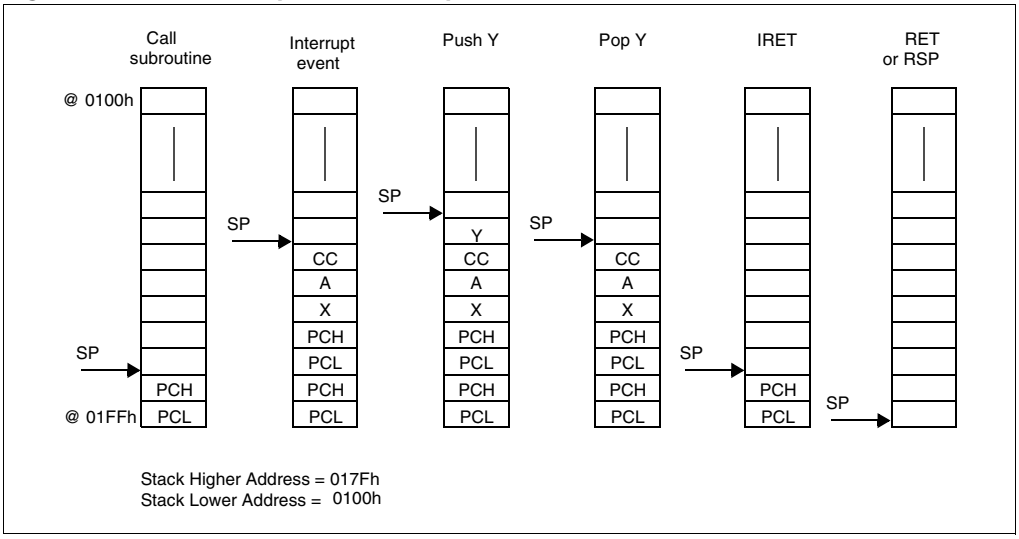
Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in *Figure 8*.

- When an interrupt is received, the SP is decremented and the context is pushed on the stack.
- On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 8. Stack manipulation example



7 Reset and clock management

7.1 Reset

The Reset procedure is used to provide an orderly software start-up or to exit low power modes.

Three reset modes are provided: a low voltage (LVD) reset, a watchdog reset and an external reset at the $\overline{\text{RESET}}$ pin.

A reset causes the reset vector to be fetched from addresses FFFEh and FFFFh in order to be loaded into the PC and with program execution starting from this point.

An internal circuitry provides a 4096 CPU clock cycle delay from the time that the oscillator becomes active.

Caution: When the ST7 is unprogrammed or fully erased, the Flash is blank and the RESET vector is not programmed. For this reason, it is recommended to keep the $\overline{\text{RESET}}$ pin in low state until programming mode is entered, in order to avoid unwanted behavior

7.2 Low voltage detector (LVD)

Low voltage reset circuitry generates a reset when V_{DD} is:

- below V_{IT+} when V_{DD} is rising,
- below V_{IT-} when V_{DD} is falling.

During low voltage reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

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.

7.2.1 Watchdog reset

When a watchdog reset occurs, the $\overline{\text{RESET}}$ pin is pulled low permitting the MCU to reset other devices in the same way as the low voltage reset (*Figure 9*).

7.2.2 External reset

The external reset is an active low input signal applied to the $\overline{\text{RESET}}$ pin of the MCU. As shown in *Figure 12*, the $\overline{\text{RESET}}$ signal must stay low for a minimum of one and a half CPU clock cycles.

An internal Schmitt trigger at the $\overline{\text{RESET}}$ pin is provided to improve noise immunity.

Figure 9. Low voltage detector functional diagram

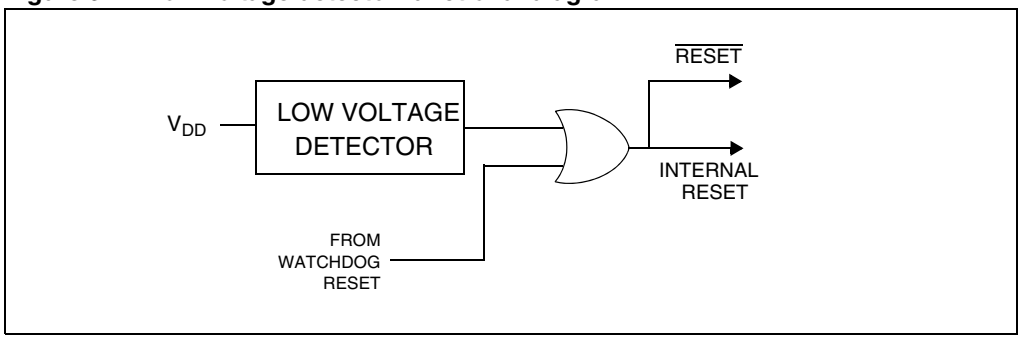
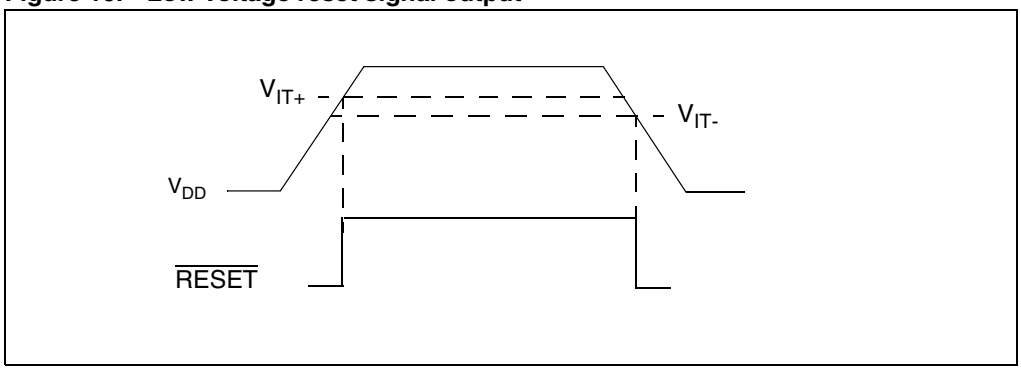


Figure 10. Low voltage reset signal output



Note: Hysteresis (V_{IT+}-V_{IT-}) = V_{hys}

Figure 11. Temporization timing diagram after an internal reset

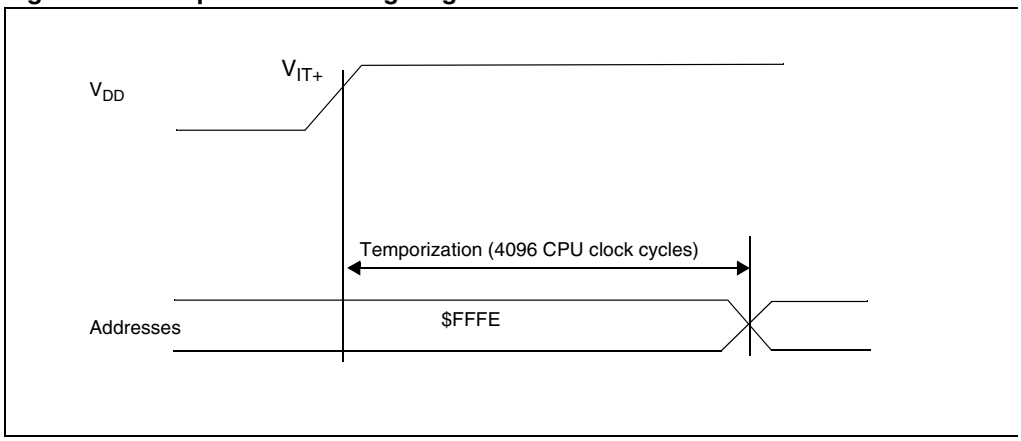
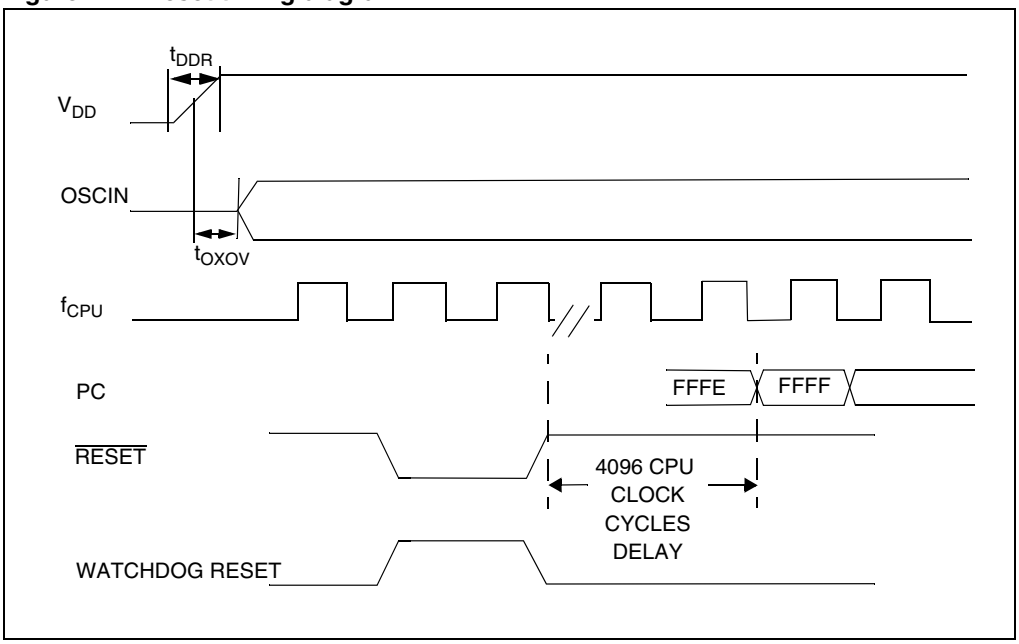


Figure 12. Reset timing diagram



Note: Refer to Electrical Characteristics for values of t_{DDR}, t_{oxov}, V_{IT+}, V_{IT-} and V_{hys}

7.3 Clock system

7.3.1 General description

The MCU accepts either a crystal or ceramic resonator, or an external clock signal to drive the internal oscillator. The internal clock (f_{CPU}) is derived from the external oscillator frequency (f_{OSC}), which is divided by 3 (and by 2 or 4 for USB, depending on the external clock used). The internal clock is further divided by 2 by setting the SMS bit in the Miscellaneous Register.
Using the OSC24/12 bit in the option byte, a 12 MHz or a 24 MHz external clock can be used to provide an internal frequency of either 2, 4 or 8 MHz while maintaining a 6 MHz for the USB (refer to *Figure 15*).

The internal clock signal (f_{CPU}) is also routed to the on-chip peripherals. The CPU clock signal consists of a square wave with a duty cycle of 50%.

The internal oscillator is designed to operate with an AT-cut parallel resonant quartz or ceramic resonator in the frequency range specified for f_{osc}. The circuit shown in *Figure 14* is recommended when using a crystal, and *Table 9* lists the recommended capacitance. The crystal and associated components should be mounted as close as possible to the input pins in order to minimize output distortion and start-up stabilisation time.

Table 9. Recommended values for 24 MHz crystal resonator

Symbol	Values		
R_{SMAX}(1)	20 Ω	25 Ω	70 Ω
C_{OSCIN}	56pF	47pF	22pF
C_{OSCOUT}	56pF	47pF	22pF
R_P	1-10 MΩ	1-10 MΩ	1-10 MΩ

1. R_{SMAX} is the equivalent serial resistor of the crystal (see crystal specification).

7.3.2 External clock

An external clock may be applied to the OSCIN input with the OSCOUT pin not connected, as shown on *Figure 13*. The t_{OXOV} specifications do not apply when using an external clock input. The equivalent specification of the external clock source should be used instead of t_{OXOV} (see *Section 16.5: Clock and timing characteristics*).

Figure 13. External clock source connections

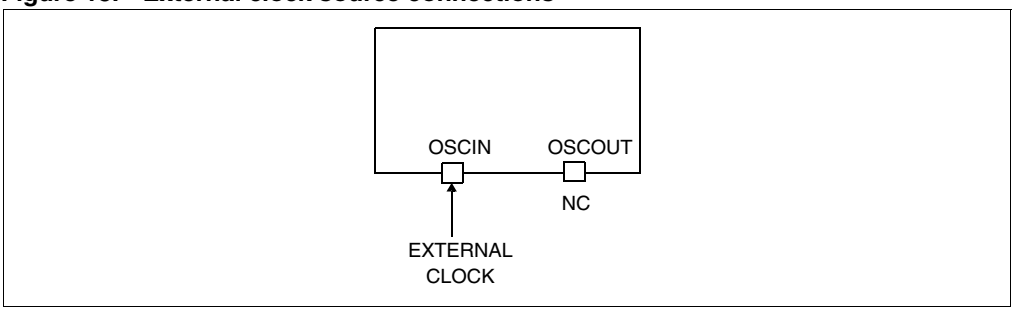


Figure 14. Crystal/ceramic resonator

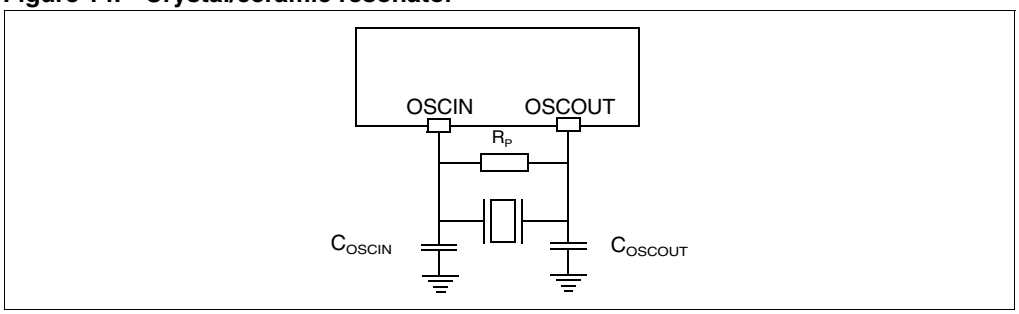
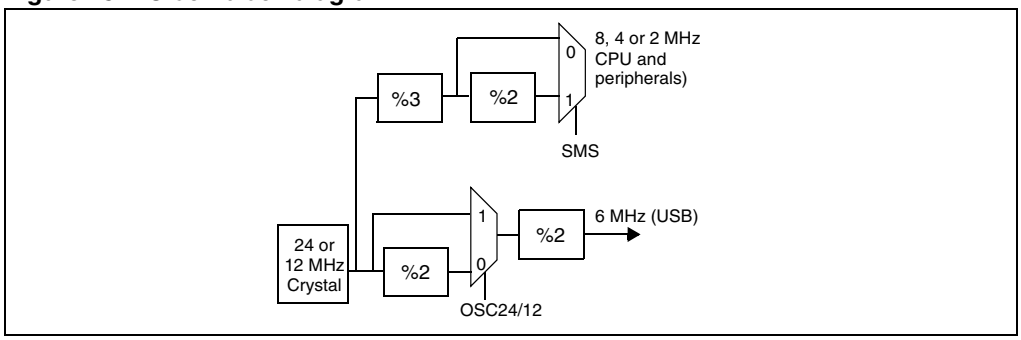


Figure 15. Clock block diagram



8 Interrupts

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in *Table 10: Interrupt mapping* and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in *Figure 16*.

The maskable interrupts must be enabled clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

When an interrupt has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- The I bit of the CC register is set to prevent additional interrupts.

The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to *Table 10: Interrupt mapping* for vector addresses).

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

Note: As a consequence of the IRET instruction, the I bit will be cleared and the main program will resume.

Priority management

By default, a servicing interrupt cannot be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case several interrupts are simultaneously pending, a hardware priority defines which one will be serviced first (see *Table 10: Interrupt mapping*).

Non-maskable software interrupts

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on *Figure 16*.

Interrupts and low power mode

All interrupts allow the processor to leave the Wait low power mode. Only external and specific mentioned interrupts allow the processor to leave the Halt low power mode (refer to the “Exit from HALT“ column in *Table 10: Interrupt mapping*).

External interrupts

The pins ITi/PAk and ITj/PBk (i=1,2; j= 5,6; k=4,5) can generate an interrupt when a rising edge occurs on this pin. Conversely, the ITl/PAn and ITm/PBn pins (l=3,4; m= 7,8; n=6,7) can generate an interrupt when a falling edge occurs on this pin.

Interrupt generation will occur if it is enabled with the ITiE bit (i=1 to 8) in the ITRFRE register and if the I bit of the CCR is reset.

Peripheral interrupts

Different peripheral interrupt flags in the status register are able to cause an interrupt when they are active if both:

- The I bit of the CC register is cleared.
- The corresponding enable bit is set in the control register.

If any of these two conditions is false, the interrupt is latched and thus remains pending.

Clearing an interrupt request is done by one of the two following operations:

- Writing “0” to the corresponding bit in the status register.
- Accessing the status register while the flag is set followed by a read or write of an associated register.

Note: 1 The clearing sequence resets the internal latch. A pending interrupt (i.e. waiting to be enabled) will therefore be lost if the clear sequence is executed.

2 All interrupts allow the processor to leave the Wait low power mode.

3 Exit from Halt mode may only be triggered by an External Interrupt on one of the ITi ports (PA4-PA7 and PB4-PB7), an end suspend mode Interrupt coming from USB peripheral, or a reset.

Figure 16. Interrupt processing flowchart

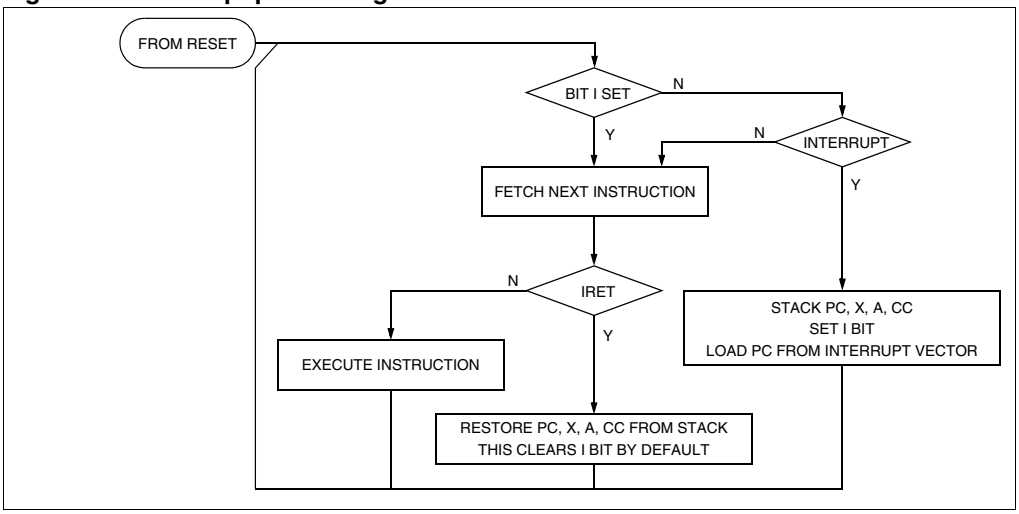


Table 10. Interrupt mapping

N°	Source block	Description	Register label	Priority order	Exit from Halt	Vector address
	RESET	Reset	N/A	Highest Priority	yes	FFFEh-FFFFh
	TRAP	Software Interrupt			no	FFFCh-FFFDh
	FLASH	Flash Start Programming Interrupt			yes	FFFAh-FFFBh
	USB	End Suspend Mode	ISTR		yes	FFF8h-FFF9h
1	ITi	External Interrupts	ITRFRE			FFF6h-FFF7h
2	TIMER	Timer Peripheral Interrupts	TIMSR		no	FFF4h-FFF5h
3	Reserved					FFF2h-FFF3h
4	SCI	SCI Peripheral Interrupts	SCISR	Lowest Priority		FFF0h-FFF1h
5	USB	USB Peripheral Interrupts	ISTR			FFEEh-FFEFh

8.0.1 Interrupt register (ITRFRE)

ITRFRE — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
IT8E	IT7E	IT6E	IT5E	IT4E	IT3E	IT2E	IT1E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11. ITRFRE register description

Bit	Name	Function
7:0	ITiE (i=1 to 8)	Interrupt enable control bits If an ITiE bit is set, the corresponding interrupt is generated when: – a rising edge occurs on the pin PA4/IT1 or PA5/IT2 or PB4/IT5 or PB5/IT6 or – a falling edge occurs on the pin PA6/IT3 or PA7/IT4 or PB6/IT7 or PB7/IT8 No interrupt is generated elsewhere..

Table 12. Interrupt register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0008h	ITRFRE reset value	IT8E 0	IT7E 0	IT6E 0	IT5E 0	IT4E 0	IT3E 0	IT2E 0	IT1E 0

9 Power saving modes

9.1 Introduction

To give a large measure of flexibility to the application in terms of power consumption, two main power saving modes are implemented in the ST7.

After a RESET, the normal operating mode is selected by default (RUN mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided by 3 (f_{CPU}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

9.2 Halt mode

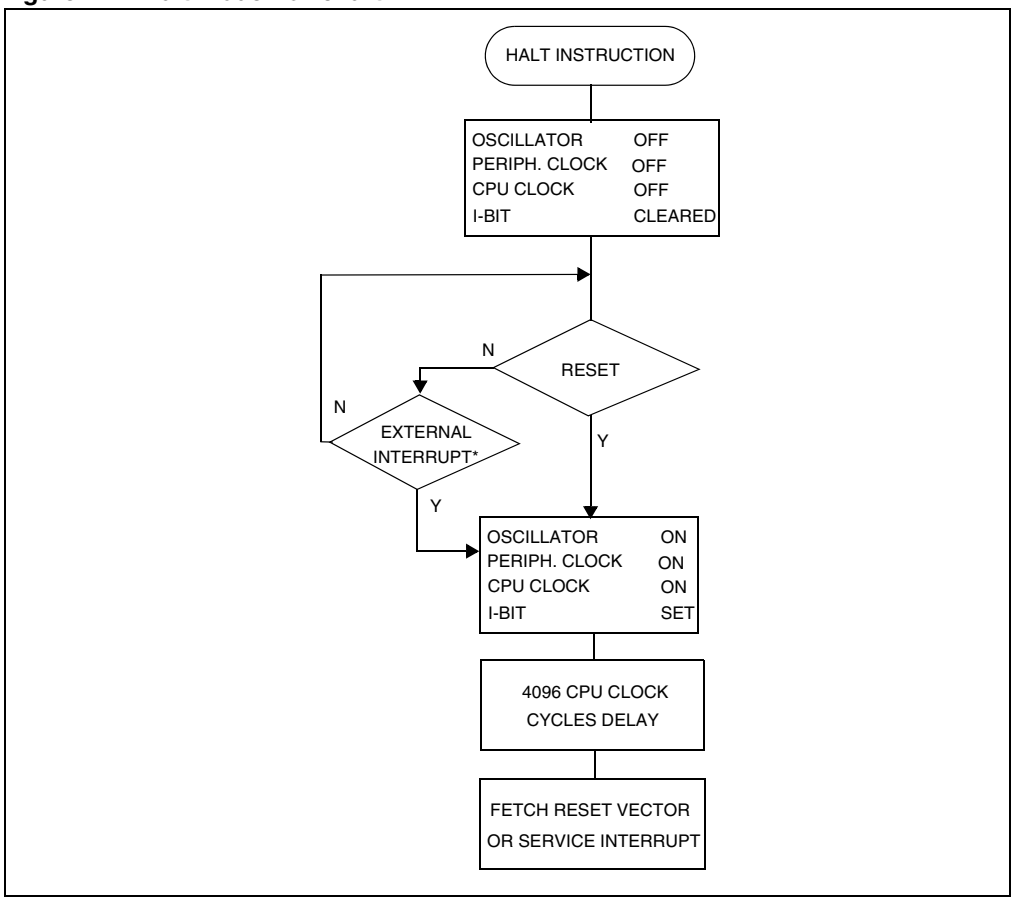
The MCU consumes the least amount of power in Halt mode. The Halt mode is entered by executing the HALT instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals.

When entering Halt mode, the I bit in the Condition Code Register is cleared. Thus, all external interrupts (ITi or USB end suspend mode) are allowed and if an interrupt occurs, the CPU clock becomes active.

The MCU can exit Halt mode on reception of either an external interrupt on ITi, an end suspend mode interrupt coming from USB peripheral, or a reset. The oscillator is then turned on and a stabilization time is provided before releasing CPU operation. The stabilization time is 4096 CPU clock cycles.

After the start up delay, the CPU continues operation by servicing the interrupt which wakes it up or by fetching the reset vector if a reset wakes it up.

Figure 17. Halt mode flowchart



Note: Before servicing an interrupt, the CC register is pushed on the stack. The I-Bit is set during the interrupt routine and cleared when the CC register is popped.

9.3 Slow mode

In Slow mode, the oscillator frequency can be divided by 2 as selected by the SMS bit in the Miscellaneous Register. The CPU and peripherals are clocked at this lower frequency. Slow mode is used to reduce power consumption, and enables the user to adapt the clock frequency to the available supply voltage.

9.4 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the “WFI” ST7 software instruction.

All peripherals remain active. During Wait mode, the I bit of the CC register is forced to 0 to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in Wait mode until an interrupt or Reset occurs, whereupon the Program Counter branches to the starting address of the interrupt or Reset service routine.

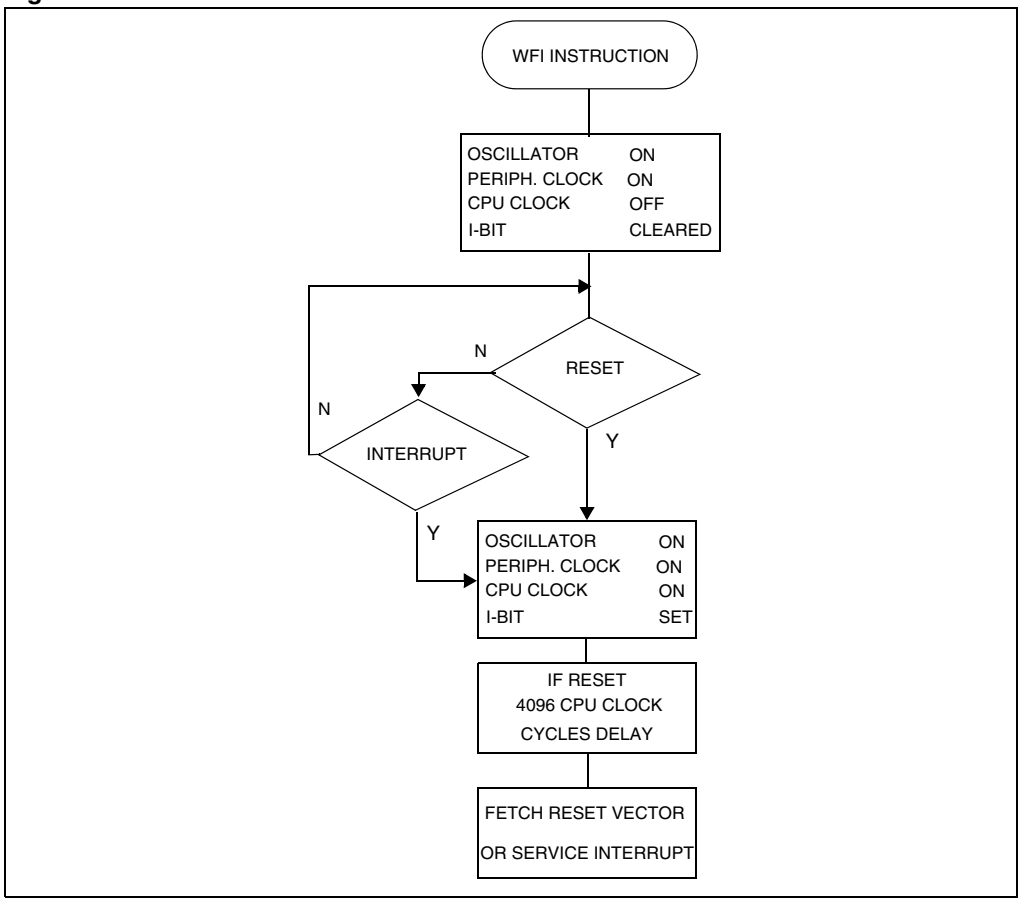
The MCU will remain in Wait mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to *Figure 18*.

Related documentation

- AN 980: ST7 keypad decoding techniques, implementing wake-up on keystroke
- AN1014: How to minimize the ST7 power consumption
- AN1605: Using an active RC to wakeup the ST7LITE0 from power saving mode

Figure 18. Wait mode flowchart



Note: Before servicing an interrupt, the CC register is pushed on the stack. The I-Bit is set during the interrupt routine and cleared when the CC register is popped.

10 I/O ports

10.1 Introduction

The I/O ports offer different functional modes:

- Transfer of data through digital inputs and outputs and for specific pins
- Alternate signal input/output for the on-chip peripherals
- External interrupt generation

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

10.2 Functional description

Each port is associated to 2 main registers:

- Data register (DR)
- Data direction register (DDR)

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

Table 13. I/O pin functions

DDR	Mode
0	Input
1	Output

Input modes

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

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

Note: 1 All the inputs are triggered by a Schmitt trigger.

2 When switching from input mode to output mode, the DR register should be written first to output the correct value as soon as the port is configured as an output.

Interrupt function

When an I/O is configured as an Input with Interrupt, an event on this I/O can generate an external Interrupt request to the CPU. The interrupt sensitivity is given independently according to the description mentioned in the ITRFRE interrupt register.

Each pin can independently generate an Interrupt request.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see Interrupts section). If more than one input pin is selected simultaneously as an interrupt source, this is logically ORed. For this reason if one of the interrupt pins is tied low, the other ones are masked.

Output mode

The pin is configured in output mode by setting the corresponding DDR register bit (see *Table 13*).

In this mode, writing “0” or “1” to the DR register applies this digital value to the I/O pin through the latch. Therefore, the previously saved value is restored when the DR register is read.

Note: The interrupt function is disabled in this mode.

Alternate function

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over standard I/O programming. When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin has to be configured in input mode. In this case, the pin’s state is also digitally readable by addressing the DR register.

Note: 1 Input pull-up configuration can cause an unexpected value at the input of the alternate peripheral input.

2 When the on-chip peripheral uses a pin as input and output, this pin must be configured as an input (DDR = 0).

Caution: The alternate function must not be activated as long as the pin is configured as an input with interrupt in order to avoid generating spurious interrupts.

10.2.1 Port A

Table 14. Port A0, A3, A4, A5, A6, A7 description

Port A	I/O		Alternate function	
	Input(1)	Output	Signal	Condition
PA0	with pull-up	push-pull	MCO (Main Clock Output)	MCO = 1 (MISCR)
PA3	with pull-up	push-pull	Timer EXTCLK	CC1 =1 CC0 = 1 (Timer CR2)
PA4	with pull-up	push-pull	Timer ICAP1	
			IT1 Schmitt triggered input	IT1E = 1 (ITIFRE)
PA5	with pull-up	push-pull	Timer ICAP2	
			IT2 Schmitt triggered input	IT2E = 1 (ITIFRE)
PA6(2)	with pull-up	push-pull	Timer OCMP1	OC1E = 1
			IT3 Schmitt triggered input	IT3E = 1 (ITIFRE)
PA7	with pull-up	push-pull	Timer OCMP2	OC2E = 1
			IT4 Schmitt triggered input	IT4E = 1 (ITIFRE)

1. Reset state
2. Not available on SO24

Figure 19. PA0, PA3, PA4, PA5, PA6, PA7 configuration

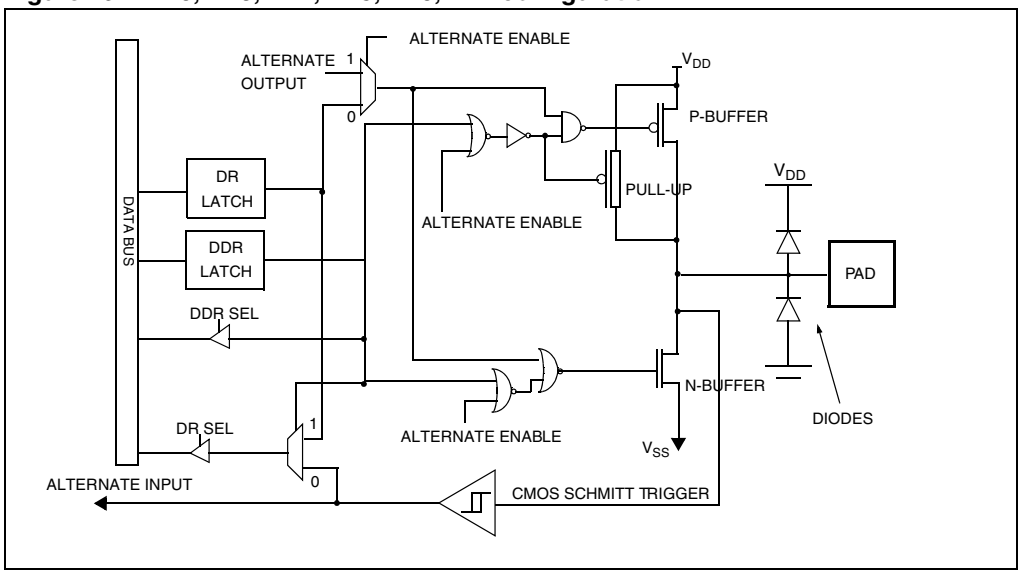
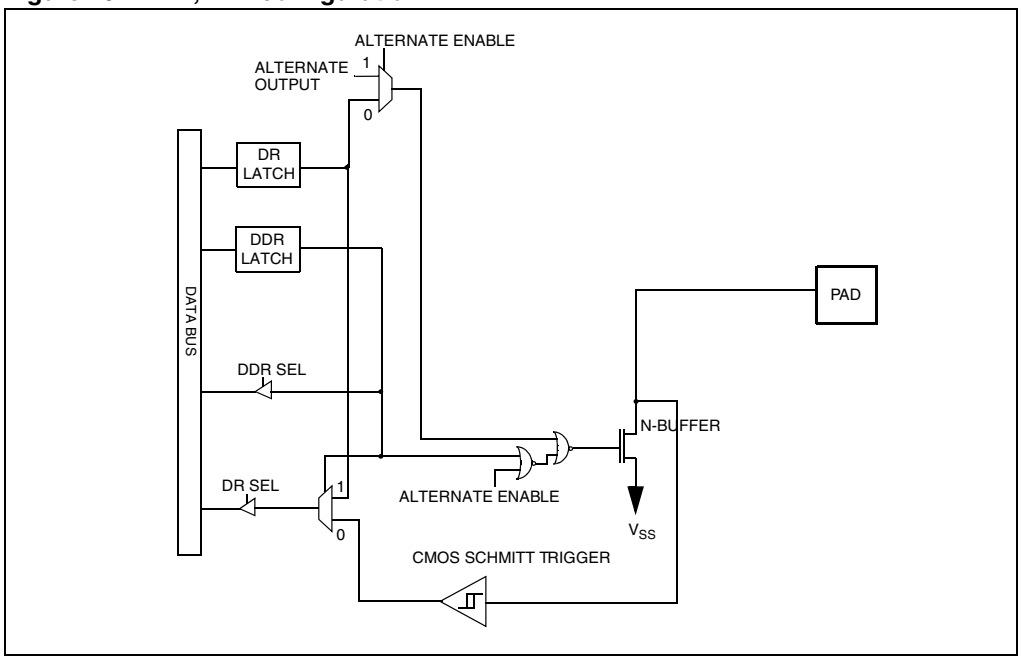


Table 15. PA1, PA2 description

Port A	I/O		Alternate function	
	Input(1)	Output	Signal	Condition
PA1	without pull-up	Very High Current open drain		
PA2	without pull-up	Very High Current open drain		

1. Reset state

Figure 20. PA1, PA2 configuration



10.2.2 Port B

Table 16. Port B description

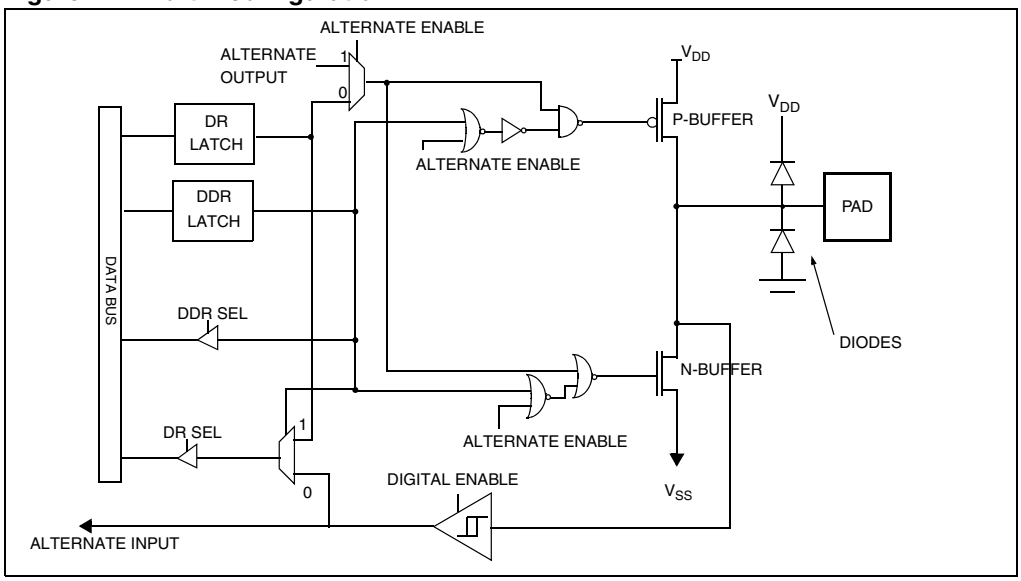
Port B	I/O		Alternate function	
	Input(1)	Output	Signal	Condition
PB0	without pull-up	push-pull		
PB1	without pull-up	push-pull		
			USBOE (USB output enable)(2)	USBOE =1 (MISCR)
PB2	without pull-up	push-pull		
PB3	without pull-up	push-pull		
PB4	without pull-up	push-pull		
			IT5 Schmitt triggered input	IT4E = 1 (ITIFRE)
PB5	without pull-up	push-pull		
			IT6 Schmitt triggered input	IT5E = 1 (ITIFRE)
PB6	without pull-up	push-pull		
			IT7 Schmitt triggered input	IT6E = 1 (ITIFRE)

Table 16. Port B description (continued)

Port B	I/O		Alternate function	
	Input(1)	Output	Signal	Condition
PB7	without pull-up	push-pull		
			IT8 Schmitt triggered input	IT7E = 1 (ITIFRE)

1. Reset state
2. On SO24 only

Figure 21. Port B configuration



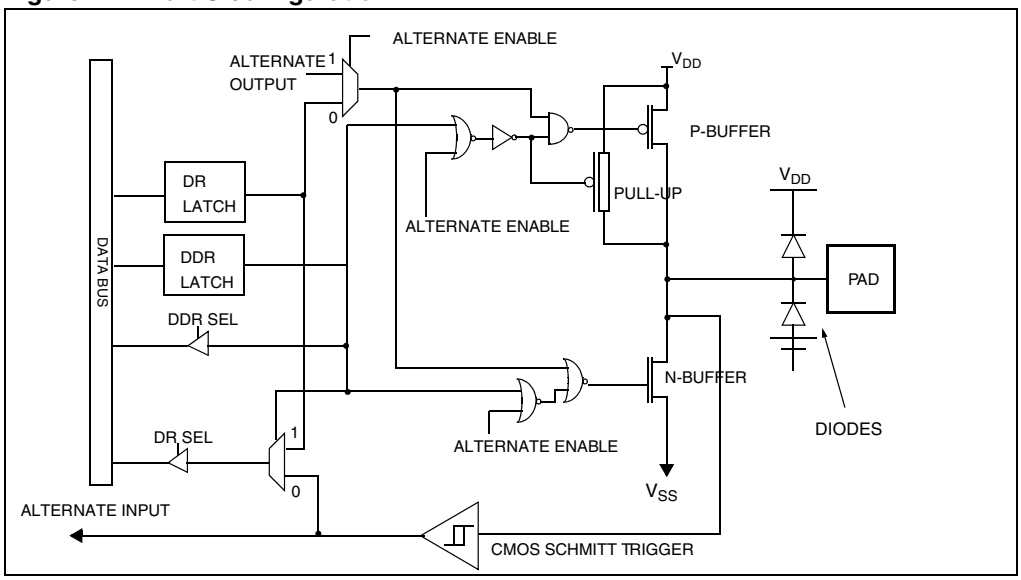
10.2.3 Port C

Table 17. Port C description

Port C	I/O		Alternate function	
	Input(1)	Output	Signal	Condition
PC0	with pull-up	push-pull	RDI (SCI input)	
PC1	with pull-up	push-pull	TDO (SCI output)	SCI enable
PC2(2)	with pull-up	push-pull	USBOE (USB output enable)	USBOE =1 (MISCR)

1. Reset state
2. Not available on SO24

Figure 22. Port C configuration



10.2.4 Register description

10.2.5 Data register (PxDR)

PADR Reset value: 0000 0000 (00h)
PBDR Reset value: 0000 0000 (00h)
PCDR Reset value: 1111 x000 (Fxh)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18. PxDR register description

Bit	Name	Function
7:0	D[7:0]	*Data bits* The DR register has a specific behavior according to the selected input/output configuration. Writing the DR register is always taken into account even if the pin is configured as an input. Reading the DR register returns either the DR register latch content (pin configured as output) or the digital value applied to the I/O pin (pin configured as input). – When using open-drain I/Os in output configuration, the value read in DR is the digital value applied to the I/O pin. – For Port C, unused bits (7-3) are not accessible

10.2.6 Data direction register (PxDDR)

PADDR Reset value: 0000 0000 (00h)
PBDDR Reset value: 0000 0000 (00h)
PCDDR Reset value: 1111 x000 (Fxh)

7	6	5	4	3	2	1	0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19. PxDDR register description

Bit	Name	Function
7:0	DD [7:0]	*Data Direction bits* The DDR register gives the input/output direction configuration of the pins. Each bit is set and cleared by software. 0: Input mode 1: Output mode For Port C, unused bits (7-3) are not accessible

Table 20. I/O port register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0000h	PADR reset value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0001h	PADDR reset value	DD7 0	DD6 0	DD5 0	DD4 0	DD3 0	DD2 0	DD1 0	DD0 0
0002h	PBDR reset value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0003h	PBDDR reset value	DD7 0	DD6 0	DD5 0	DD4 0	DD3 0	DD2 0	DD1 0	DD0 0
0004h	PCDR reset value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0005h	PCDDR reset value	DD7 0	DD6 0	DD5 0	DD4 0	DD3 0	DD2 0	DD1 0	DD0 0
0006h	Reserved								
0007h	Reserved								

10.2.7 Related documentation

- AN 970: SPI Communication between ST7 and EEPROM
- AN1048: Software LCD driver

11 Miscellaneous register

MISCR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
-	-	-	-	-	SMS	USBOE	MCO
					R/W	R/W	R/W

Table 21. MISCR register description

Bit	Name	Function
7:3		Reserved
2	SMS	*Slow mode select* This bit is set by software and only cleared by hardware after a reset. If this bit is set, it enables the use of an internal divide-by-2 clock divider (refer to *Figure 15 on page 29*). The SMS bit has no effect on the USB frequency. 0: Divide-by-2 disabled and CPU clock frequency is standard 1: Divide-by-2 enabled and CPU clock frequency is halved
1	USB OE	*USB enable* If this bit is set, the port PC2 (PB1 on SO24) outputs the USB output enable signal (at “1” when the ST7 USB is transmitting data). Unused bits 7-4 are set.
0	MCO	*Main clock out selection* This bit enables the MCO alternate function on the PA0 I/O port. It is set and cleared by software. 0: MCO alternate function disabled (I/O pin free for general-purpose I/O) 1: MCO alternate function enabled (f_{CPU} on I/O port)

Table 22. Miscellaneous register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0009h	MISCR reset value						SMS 0	USB OE 0	MCO 0

12 Watchdog timer (WDG)

12.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

12.2 Main features

- Programmable free-running counter (64 increments of 49,152 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero
- Optional reset on HALT instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte.

12.3 Functional description

The counter value stored in the CR register (bits T6:T0), is decremented every 49,152 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T6:T0) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for a period of t_{DOG} (see *Table 62: Control timings on page 114*).

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is free-running: it counts down even if the watchdog is disabled. The value to be stored in the CR register must be between FFh and C0h (see *Table 23: Watchdog timing (f_{CPU} = 8 MHz) on page 45*):

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T5:T0 bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Figure 23. Watchdog block diagram

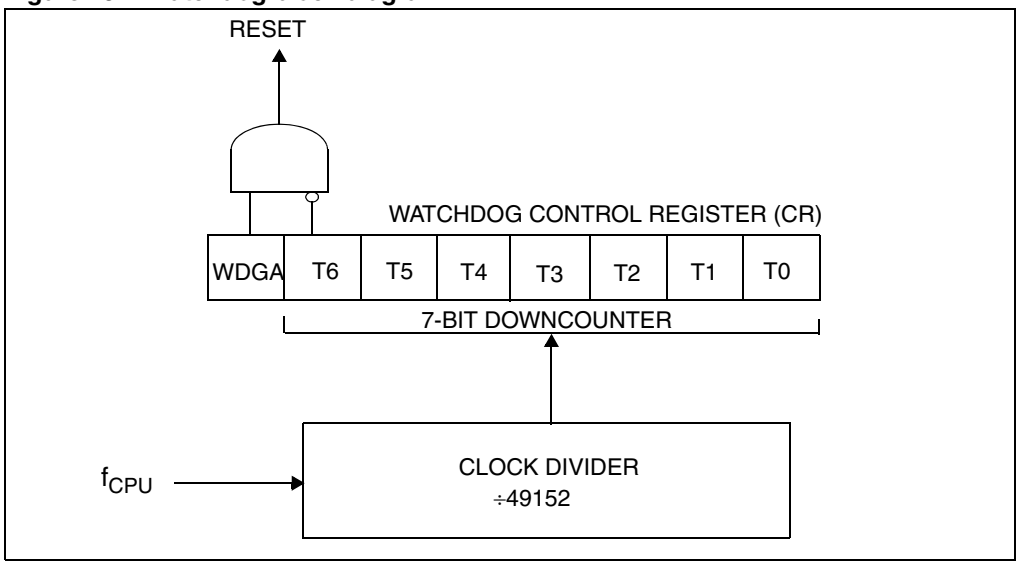


Table 23. Watchdog timing (f_{CPU} = 8 MHz)

	CR register initial value	WDG timeout period (ms)
Max	FFh	393.216
Min	C0h	6.144

Note: 1 Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

2 The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

12.3.1 Software watchdog option

If Software Watchdog is selected by option byte, the watchdog is disabled following a reset. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

12.3.2 Hardware watchdog option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the CR is not used.

12.3.3 Low power modes

WAIT Instruction

No effect on Watchdog.

HALT Instruction

If the Watchdog reset on HALT option is selected by option byte, a HALT instruction causes an immediate reset generation if the Watchdog is activated (WDGA bit is set).

12.3.4 Using Halt mode with the WDG (option)

If the Watchdog reset on HALT option is not selected by option byte, the Halt mode can be used when the watchdog is enabled.

In this case, the HALT instruction stops the oscillator. When the oscillator is stopped, the WDG stops counting and is no longer able to generate a reset until the microcontroller receives an external interrupt or a reset.

If an external interrupt is received, the WDG restarts counting after 4096 CPU clocks. If a reset is generated, the WDG is disabled (reset state).

Recommendations

- Make sure that an external event is available to wake up the microcontroller from Halt mode.
- Before executing the HALT instruction, refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.
- When using an external interrupt to wake up the microcontroller, reinitialize the corresponding I/O as “Input Pull-up with Interrupt” before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.
- For the same reason, reinitialize the level sensitiveness of each external interrupt as a precautionary measure.
- The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant with the value 0x8E.
- As the HALT instruction clears the I bit in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wake-up event (reset or external interrupt).

12.3.5 Interrupts

None.

12.3.6 Control register (WDGCR)

WDGCR Reset value: 0111 1111 (7Fh)

7	6	5	4	3	2	1	0
WDGA	T[6:0]						
R/W	R/W						

Table 24. WDGCR register description

Bit	Name	Function
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.*
6:0	T[6:0]	7-bit counter (MSB to LSB) These bits contain the value of the Watchdog counter. A reset is produced when it rolls over from 40h to 3Fh (T6 is cleared).

Table 25. Watchdog timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
000Ch	WDGCR reset value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

12.4 16-bit timer

12.4.1 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).

12.4.2 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)[a]

The timer block diagram is shown in *Figure 24*.

a. Some timer pins may not be available (not bonded) in some ST7 devices. Refer to *Section 3: Pin description*. When reading an input signal on a non-bonded pin, the value will always be '1'.

12.4.3 Functional description

Counter

The main block of the programmable timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high and low.

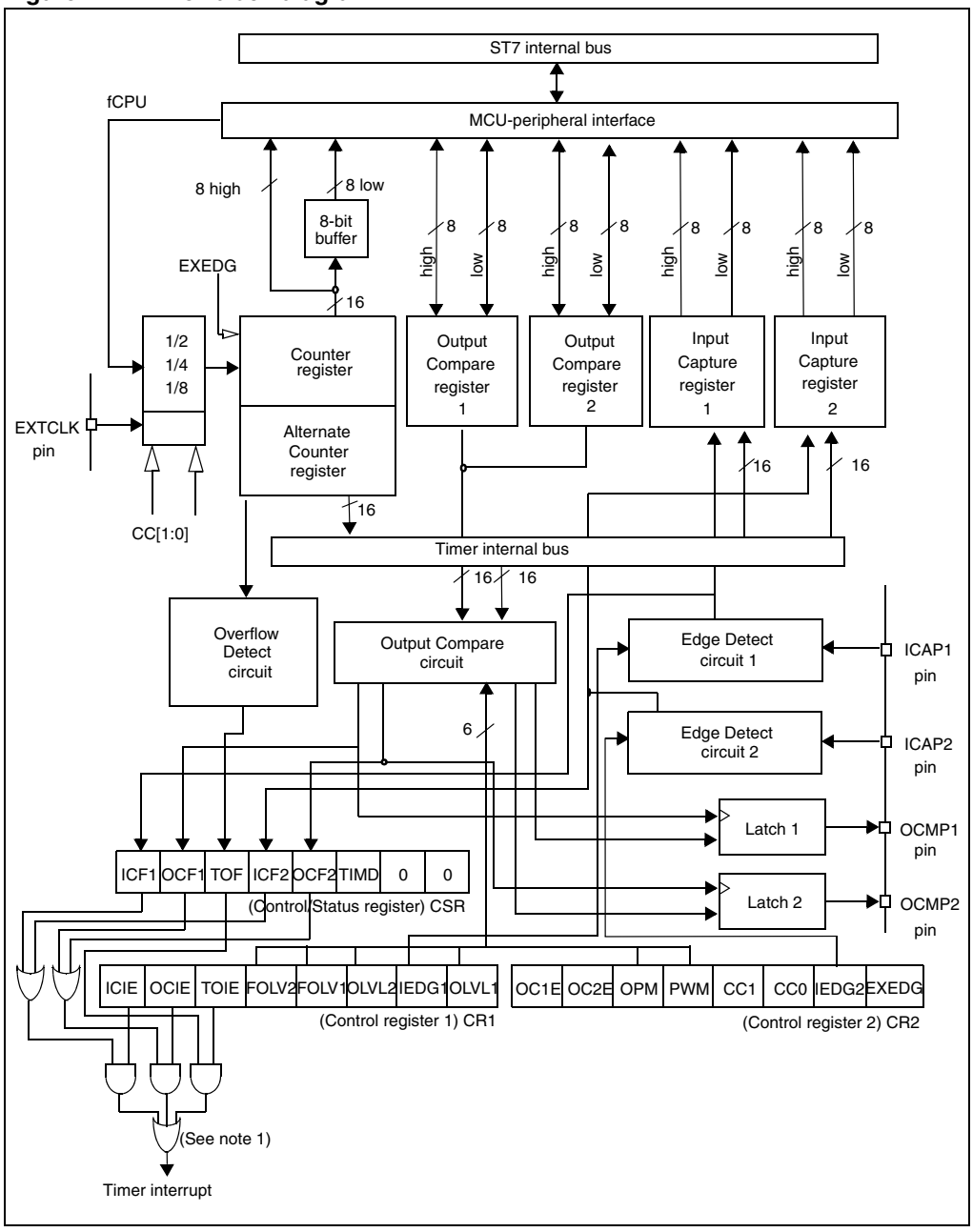
- Counter Register (CR)
  - Counter High Register (CHR) is the most significant byte (MSB)
  - Counter Low Register (CLR) is the least significant byte (LSB)
- Alternate Counter Register (ACR)
  - Alternate Counter High Register (ACHR) is the most significant byte (MSB)
  - Alternate Counter Low Register (ACLR) is the least significant byte (LSB)

These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (timer overflow flag), located in the Status register (SR) (see note at the end of paragraph entitled *16-bit read sequence)*.

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value. Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in one pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in *Table 32*. The value in the counter register repeats every 131072, 262144 or 524288 CPU clock cycles depending on the CC[1:0] bits. The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

Figure 24. Timer block diagram

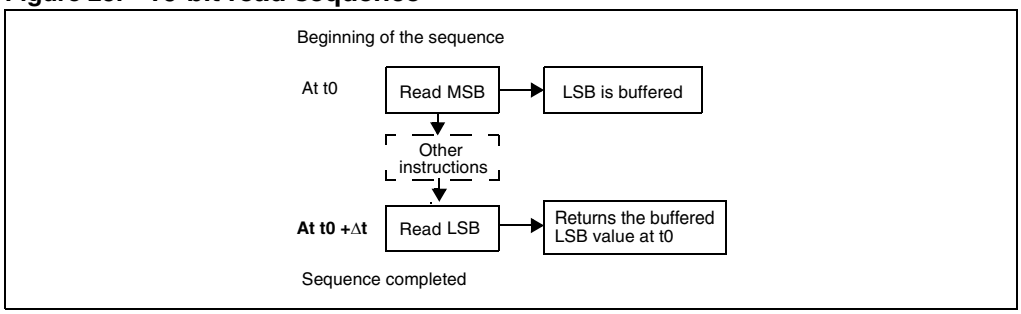


1. If IC, OC and TO interrupt requests have separate vectors then the last OR is not present (see *Table 10: Interrupt mapping on page 32*).

16-bit read sequence

The 16-bit read sequence (from either the Counter register or the Alternate Counter register) is illustrated in the following *Figure 25*.

Figure 25. 16-bit read sequence



The user must first read the MSB, afterwhich the LSB value is automatically buffered.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LSB of the count value at the time of the read.

Whatever the timer mode used (input capture, output compare, one pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFFFh to 0000h then:

- The TOF bit of the SR register is set.
- A timer interrupt is generated if:
  - TOIE bit of the CR1 register is set and
  - I bit of the CC register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done in two steps:

1. Reading the SR register while the TOF bit is set.
2. An access (read or write) to the CLR register.

Note: The TOF bit is not cleared by access to the ACLR register. The advantage of accessing the ACLR register rather than the CLR register is that it allows simultaneous use of the overflow function and reading the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by Wait mode.

In Halt mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a reset).

External clock

The external clock (where available) is selected if CC0 = 1 and CC1 = 1 in the CR2 register.

The status of the EXEDG bit in the CR2 register determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronized with the falling edge of the internal CPU clock.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

Figure 26. Counter timing diagram, internal clock divided by 2

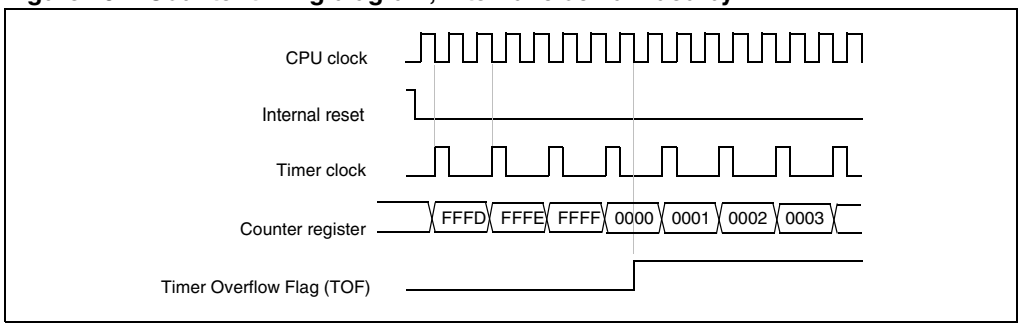


Figure 27. Counter timing diagram, internal clock divided by 4

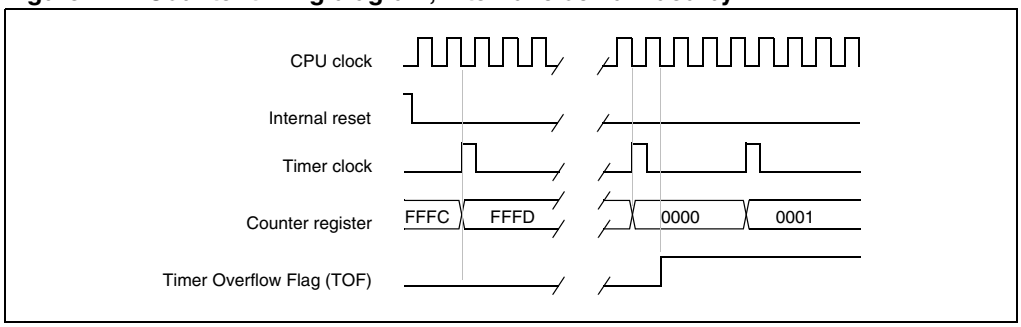
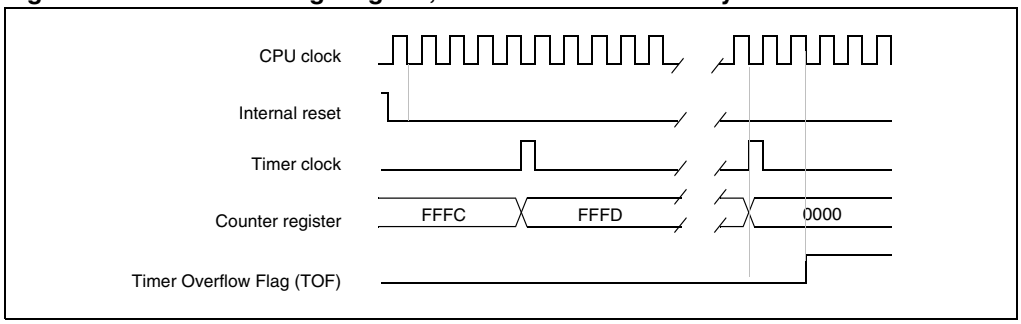


Figure 28. Counter timing diagram, internal clock divided by 8



Note: The MCU is in reset state when the internal reset signal is high, when it is low the MCU is running.

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 30*).

Table 26. 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 32*).
- 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 30*).
- 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

Note: 1 After reading the ICiHR register, transfer of input capture data is inhibited and ICFi will never be set until the ICiLR register is also read.

2 The ICiR register contains the free running counter value which corresponds to the most recent input capture.

3 The two input capture functions can be used together even if the timer also uses the two output compare functions.

4 In One pulse mode and PWM mode only Input Capture 2 can be used.

5 The alternate inputs (ICAP1 and ICAP2) are always directly connected to the timer. So any transitions on these pins activates the input capture function.
Moreover if one of the ICAPi pins is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.
This can be avoided if the input capture function i is disabled by reading the ICiHR (see note 1).

6 The TOF bit can be used with interrupt generation in order to measure events that go beyond the timer range (FFFFh).

Figure 29. Input capture block diagram

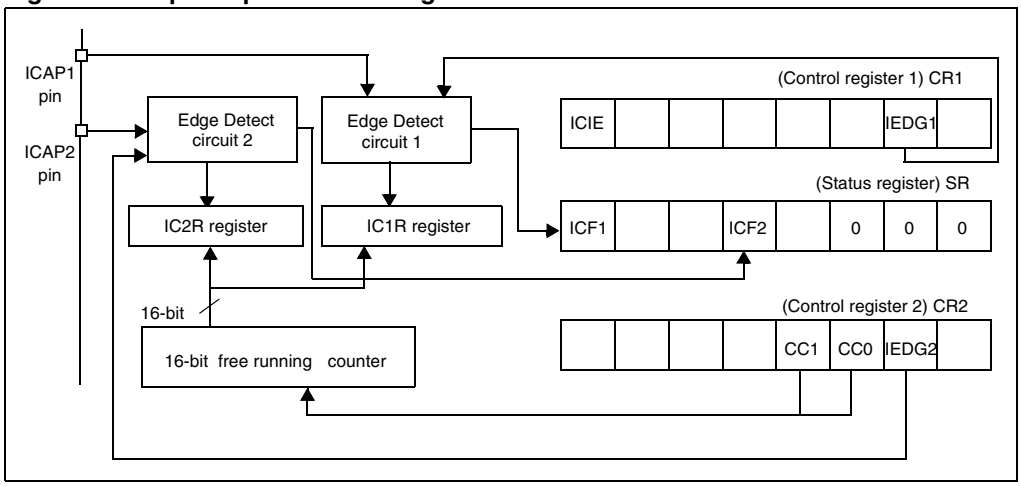
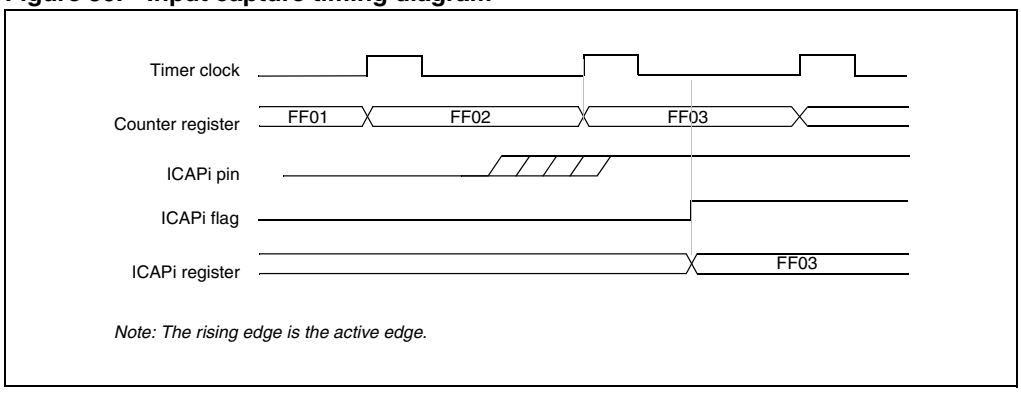


Figure 30. Input capture timing diagram



Note: The rising edge is the active edge.

Output compare

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

This function can be used to control an output waveform or indicate when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OC*i*E bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare register 1 (OC1R) and Output Compare register 2 (OC2R) contain the value to be compared to the counter register each timer clock cycle.

Table 27. Output compare byte distribution

Register	MS byte	LS byte
OCiR	OC*i*HR	OC*i*LR

These registers are readable and witable and are not affected by the timer hardware. A reset event changes the OC*i*R value to 8000h.

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

Procedure

To use the Output Compare function, select the following in the CR2 register:

- Set the OC*i*E bit if an output is needed then the OCMP*i* pin is dedicated to the output compare *i* signal.
- Select the timer clock (CC[1:0]) (see *Table 32*).

And select the following in the CR1 register:

- Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When a match is found between OCRi register and CR register:

- OCF*i* bit is set
- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset)
- A timer interrupt is generated if the OCIE bit is set in the CR1 register and the I bit is cleared in the CC register (CC).

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\Delta \text{ OC}i\text{R} = \frac{\Delta t * f_{CPU}}{\text{PRESC}}$$

Where:

Δt = Output compare period (in seconds)
f_{CPU} = CPU clock frequency (in hertz)
PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits; see *Table 32*)

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

$$\Delta OCiR = \Delta t * f_{EXT}$$

Where:

Δt = Output compare period (in seconds)
f_{EXT} = External timer clock frequency (in hertz)

Clearing the output compare interrupt request (that is, clearing the OCF*i* bit) is done by:

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

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).

Note: 1 After a processor write cycle to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

2 If the OCiE bit is not set, the OCMPi pin is a general I/O port and the OLVLi bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.

3 In both internal and external clock modes, OCFi and OCMPi are set while the counter value equals the OCiR register value (see Figure 32 on page 57 for an example with $f_{CPU}/2$ and Figure 33 on page 57 for an example with $f_{CPU}/4$). This behavior is the same in OPM or PWM mode.

4 The output compare functions can be used both for generating external events on the OCMPi pins even if the input capture mode is also used.

5 The value in the 16-bit OCiR register and the OLVi bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

Forced output compare capability

When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit = 1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

The FOLVL*i* bits have no effect in both one pulse mode and PWM mode.

Figure 31. Output compare block diagram

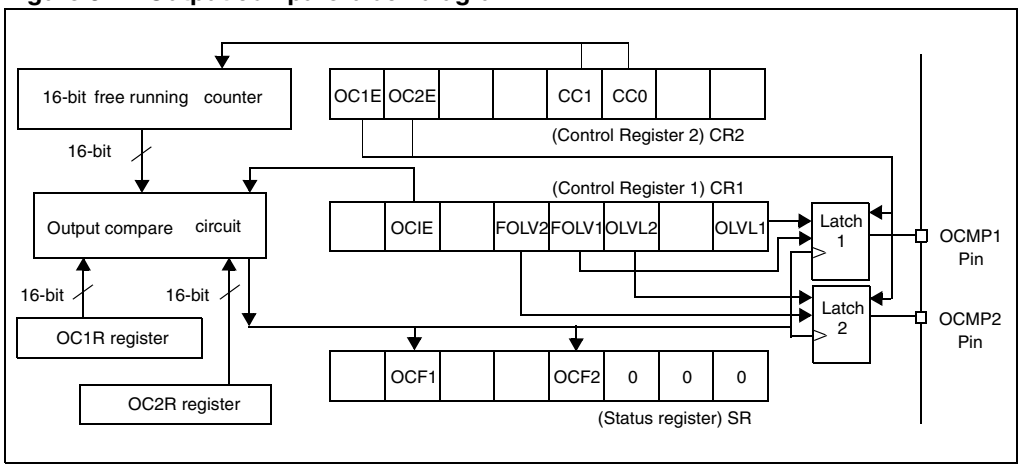


Figure 32. Output compare timing diagram, $f_{TIMER} = f_{CPU}/2$

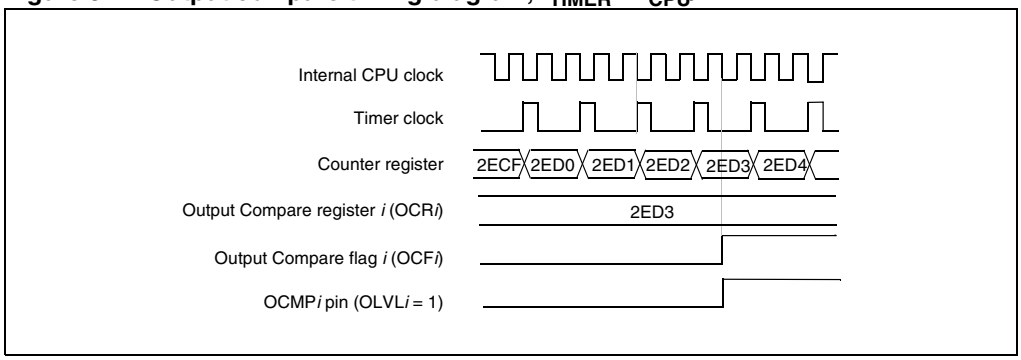
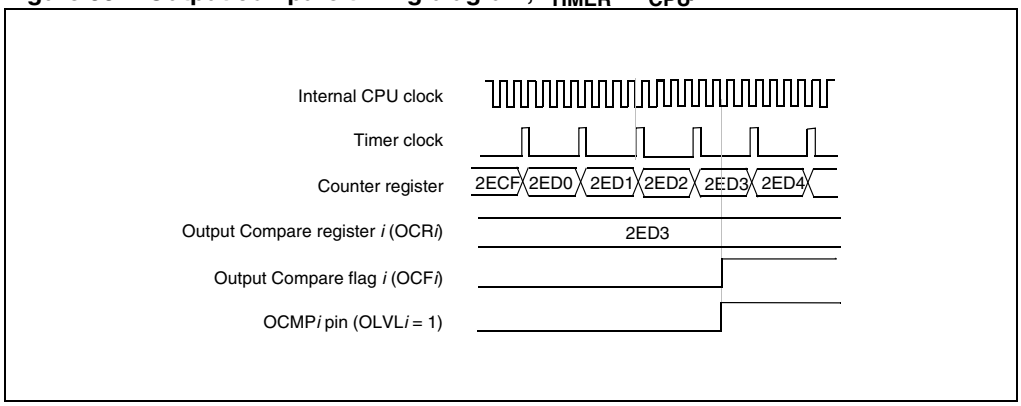


Figure 33. Output compare timing diagram, $f_{TIMER} = f_{CPU}/4$



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.

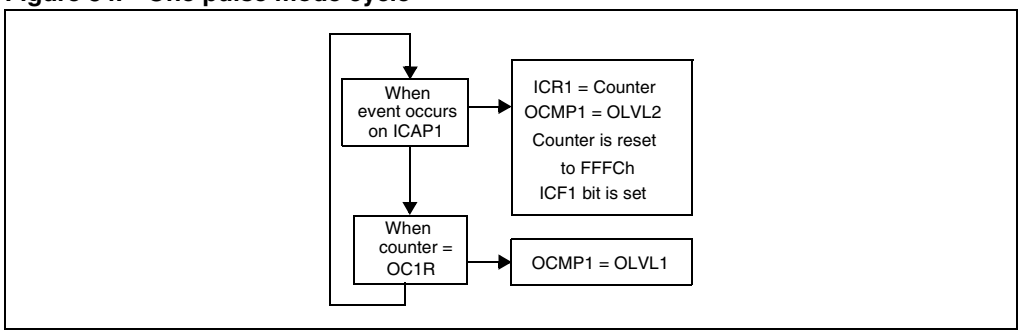
The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

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 32*).

Figure 34. 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.

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

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

Where:

t = Pulse period (in seconds)
f_{CPU} = CPU clock frequency (in hertz)
PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits; see *Table 32*)

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

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

Where:

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

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see *Figure 35*).

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

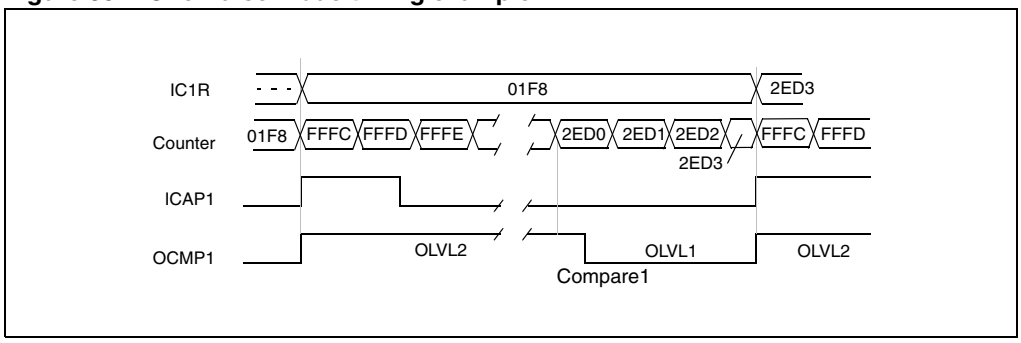
2 When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

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.

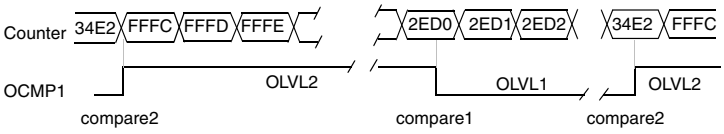
5 When one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

Figure 35. One Pulse mode timing example(1)



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

Figure 36. Pulse width modulation mode timing example with two output compare functions(1)(2)



1. OC1R = 2ED0h, OC2R = 34E2, OLVL1 = 0, OLVL2 = 1
2. On timers with only one Output Compare register, a fixed frequency PWM signal can be generated using the output compare and the counter overflow to define the pulse length.

Pulse width modulation mode

Pulse Width Modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so this functionality can not be used when PWM mode is activated.

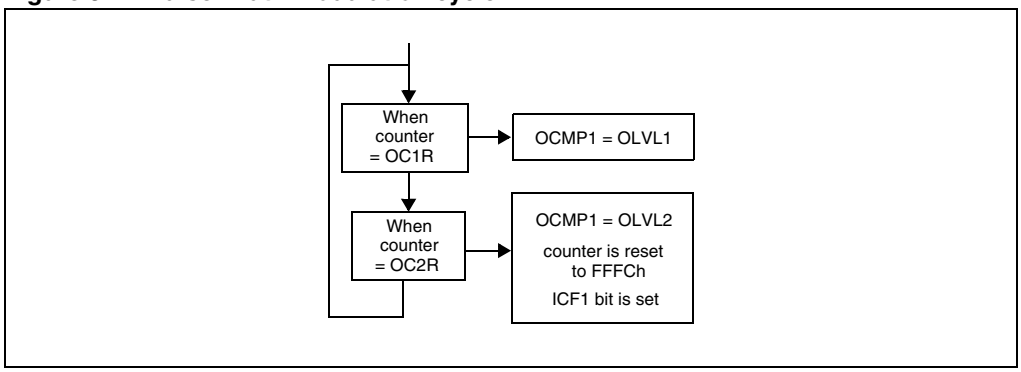
In PWM mode, double buffering is implemented on the output compare registers. Any new values written in the OC1R and OC2R registers are taken into account only at the end of the PWM period (OC2) to avoid spikes on the PWM output pin (OCMP1).

Procedure

To use Pulse Width Modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal using the formula below.
2. Load the OC1R register with the value corresponding to the period of the pulse if (OLVL1 = 0 and OLVL2 = 1) using the formula in the opposite column.
3. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC1R register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC[1:0]) (see *Table 32*).

Figure 37. Pulse width modulation cycle



If OLVL1 = 1 and OLVL2 = 0, the length of the positive pulse is the difference between the OC2R and OC1R registers.

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

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

$$OC\mathit{i}R \text{ value} = \frac{t * f_{CPU}}{PRESC} - 5$$

Where:

t = Signal or pulse period (in seconds)
f_{CPU} = CPU clock frequnency (in hertz)
PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits; see *Table 32*)

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

$$OC\mathit{i}R = t * f_{EXT} - 5$$

Where:

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

The Output Compare 2 event causes the counter to be initialized to FFFCh (see *Figure 36*).

Note: 1 After a write instruction to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

2 The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.

3 The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.

4 In PWM mode the ICAP1 pin can not be used to perform input capture because it is disconnected to the timer. 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 period and ICF1 can also generates interrupt if ICIE is set.

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

12.4.4 Low power modes

Table 28. Effect of low power modes on 16-bit timer

Mode	Description
Wait	No effect on 16-bit timer. Timer interrupts cause the device to exit from Wait mode.
Halt	16-bit timer registers are frozen. In Halt mode, the counter stops counting until Halt mode is exited. Counting resumes from the previous count when the MCU is woken up by an interrupt with Exit from Halt mode capability or from the counter reset value when the MCU is woken up by a reset. If an input capture event occurs on the ICAP*i* pin, the input capture detection circuitry is armed. Consequently, when the MCU is woken up by an interrupt with Exit from Halt mode capability, the ICF*i* bit is set, and the counter value present when exiting from Halt mode is captured into the IC*i*R register.

12.4.5 Interrupts

Table 29. 16-bit timer interrupt control/wake-up capability(1)

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Input Capture 1 event/counter reset in PWM mode	ICF1	ICIE	Yes	No
Input Capture 2 event	ICF2			
Output Compare 1 event (not available in PWM mode)	OCF1	OCIE		
Output Compare 2 event (not available in PWM mode)	OCF2			
Timer Overflow event	TOF	TOIE		

1. The 16-bit timer interrupt events are connected to the same interrupt vector (see *Section 8: Interrupts*). These events generate an interrupt if the corresponding Enable Control bit is set and the interrupt mask in the CC register is reset (RIM instruction).

12.4.6 Summary of timer modes

Table 30. 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 58*.
2. See note 5 in *One pulse mode on page 58*.
3. See note 4 in *Pulse width modulation mode on page 60*.

12.4.7 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 31. 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.

Table 31. CR1 register description (continued)

Bit	Name	Function
4	FOLV2	Forced Output compare 2 This bit is set and cleared by software. 0: No effect on the OCMP2 pin. 1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison.
3	FOLV1	Forced Output compare 1 This bit is set and cleared by software. 0: No effect on the OCMP1 pin. 1: Forces OLVL1 to be copied to the OCMP1 pin, if the OC1E bit is set and even if there is no successful comparison.
2	OLVL2	Output Level 2 This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse mode and Pulse Width modulation mode.
1	IEDG1	Input Edge 1 This bit determines which type of level transition on the ICAP1 pin will trigger the capture. 0: A falling edge triggers the capture. 1: A rising edge triggers the capture.
0	OLVL1	Output Level 1 The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

Control Register 2 (CR2)

CR2 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
OC1E	OC2E	OPM	PWM	CC[1:0]		IEDG2	EXEDG
R/W	R/W	R/W	R/W	R/W		R/W	R/W

Table 32. CR2 register description

Bit	Name	Function
7	OCIE	Output Compare 1 Pin Enable This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in Output Compare mode, both OLV1 and OLV2 in PWM and One-Pulse mode). Whatever the value of the OC1E bit, the Output Compare 1 function of the timer remains active. 0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O). 1: OCMP1 pin alternate function enabled.
6	OC2E	Output Compare 2 Pin Enable This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in Output Compare mode). Whatever the value of the OC2E bit, the Output Compare 2 function of the timer remains active. 0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O). 1: OCMP2 pin alternate function enabled.

Table 32. CR2 register description (continued)

Bit	Name	Function
5	OPM	One Pulse Mode 0: One Pulse mode is not active. 1: One Pulse mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.
4	PWM	Pulse Width Modulation 0: PWM mode is not active. 1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.
3:2	CC[1:0]	Clock Control The timer clock mode depends on these bits. 00: Timer clock = $f_{CPU}/4$ 01: Timer clock = $f_{CPU}/2$ 10: Timer clock = $f_{CPU}/8$ 11: Timer clock = external clock (where available) *Note: If the external clock pin is not available, programming the external clock configuration stops the counter.*
1	IEDG2	Input Edge 2 This bit determines which type of level transition on the ICAP2 pin will trigger the capture. 0: A falling edge triggers the capture. 1: A rising edge triggers the capture.
0	EXEDG	External Clock Edge This bit determines which type of level transition on the external clock pin EXTCLK will trigger the counter register. 0: A falling edge triggers the counter register. 1: A rising edge triggers the counter register.

Control/Status Register (CSR)

CSR Reset value: xxxx x0xx (xxh)

7	6	5	4	3	2	1	0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	Reserved	
RO	RO	RO	RO	RO	R/W	-	

Table 33. CSR register description

Bit	Name	Function
7	ICF1	Input Capture Flag 1 0: No Input Capture (reset value). 1: An Input Capture has occurred on the ICAP1 pin or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Table 33. CSR register description (continued)

Bit	Name	Function
6	OCF1	Output Compare Flag 1 0: No match (reset value). 1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.
5	TOF	Timer Overflow Flag 0: No timer overflow (reset value). 1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register. *Note: Reading or writing the ACLR register does not clear TOF.*
4	ICF2	Input Capture Flag 2 0: No input capture (reset value). 1: An Input Capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.
3	OCF2	Output Compare Flag 2 0: No match (reset value). 1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.
2	TIMD	Timer Disable This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed, or the counter reset, while it is disabled. 0: Timer enabled. 1: Timer prescaler, counter and outputs disabled.
1:0	-	Reserved, must be kept cleared.

Input capture 1 high register (IC1HR)

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

IC1HR — Reset value: undefined

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

Input capture 1 low register (IC1LR)

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

IC1LR — Reset value: undefined

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

Output compare 1 high register (OC1HR)

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

OC1HR — Reset value: 1000 0000 (80h)

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

Output compare 1 low register (OC1LR)

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

OC1LR — Reset value: 0000 0000 (00h)

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

Output compare 2 high register (OC2HR)

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

OC2HR — Reset value: 1000 0000 (80h)

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

Output compare 2 low register (OC2LR)

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

OC2LR Reset value: 0000 0000 (00h)

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

Counter high register (CHR)

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

CHR Reset value: 1111 1111 (FFh)

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

Counter low register (CLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.

CLR Reset value: 1111 1100 (FCh)

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

Alternate counter high register (ACHR)

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

ACHR Reset value: 1111 1111 (FFh)

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

Alternate counter low register (ACLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

ACLR — Reset value: 1111 1100 (FCh)

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

Input capture 2 high register (IC2HR)

This is an 8-bit register that contains the high part of the counter value (transferred by the Input Capture 2 event).

1C2HR — Reset value: undefined

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

Input capture 2 low register (IC2LR)

This is an 8-bit register that contains the low part of the counter value (transferred by the Input Capture 2 event).

1C2LR — Reset value: undefined

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

Table 34. 16-bit timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
11	CR1 Reset value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
12	CR2 Reset value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
13	CSR Reset value	ICF1 x	OCF1 x	TOF x	ICF2 x	OCF2 x	TIMD 0	- x	- x
14	IC1HR Reset value	MSB x	x	x	x	x	x	x	LSB x
15	IC1LR Reset value	MSB x	x	x	x	x	x	x	LSB x
16	OC1HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
17	OC1LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
18	CHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
19	CLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
1A	ACHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
1B	ACLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
1C	IC2HR Reset value	MSB x	x	x	x	x	x	x	LSB x
1D	IC2LR Reset value	MSB x	x	x	x	x	x	x	LSB x
1E	OC2HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
1F	OC2LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0

13 Serial communications interface (SCI)

13.1 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.

13.2 Main features

- Full duplex, asynchronous communications
- NRZ standard format (Mark/Space)
- Independently programmable transmit and receive baud rates up to 250K baud.
- Programmable data word length (8 or 9 bits)
- Receive buffer full, Transmit buffer empty and End of Transmission flags
- Two receiver wake-up modes:
  - Address bit (MSB)
  - Idle line
- Muting function for multiprocessor configurations
- Separate enable bits for Transmitter and Receiver
- Four error detection flags:
  - Overrun error
  - Noise error
  - Frame error
  - Parity error
- Six interrupt sources with flags:
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle line received
  - Overrun error detected
  - Parity error
- Parity control:
  - Transmits parity bit
  - Checks parity of received data byte
- Reduced power consumption mode

13.2.1 General description

The interface is externally connected to another device by two pins (see *Figure 39*):

- 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.

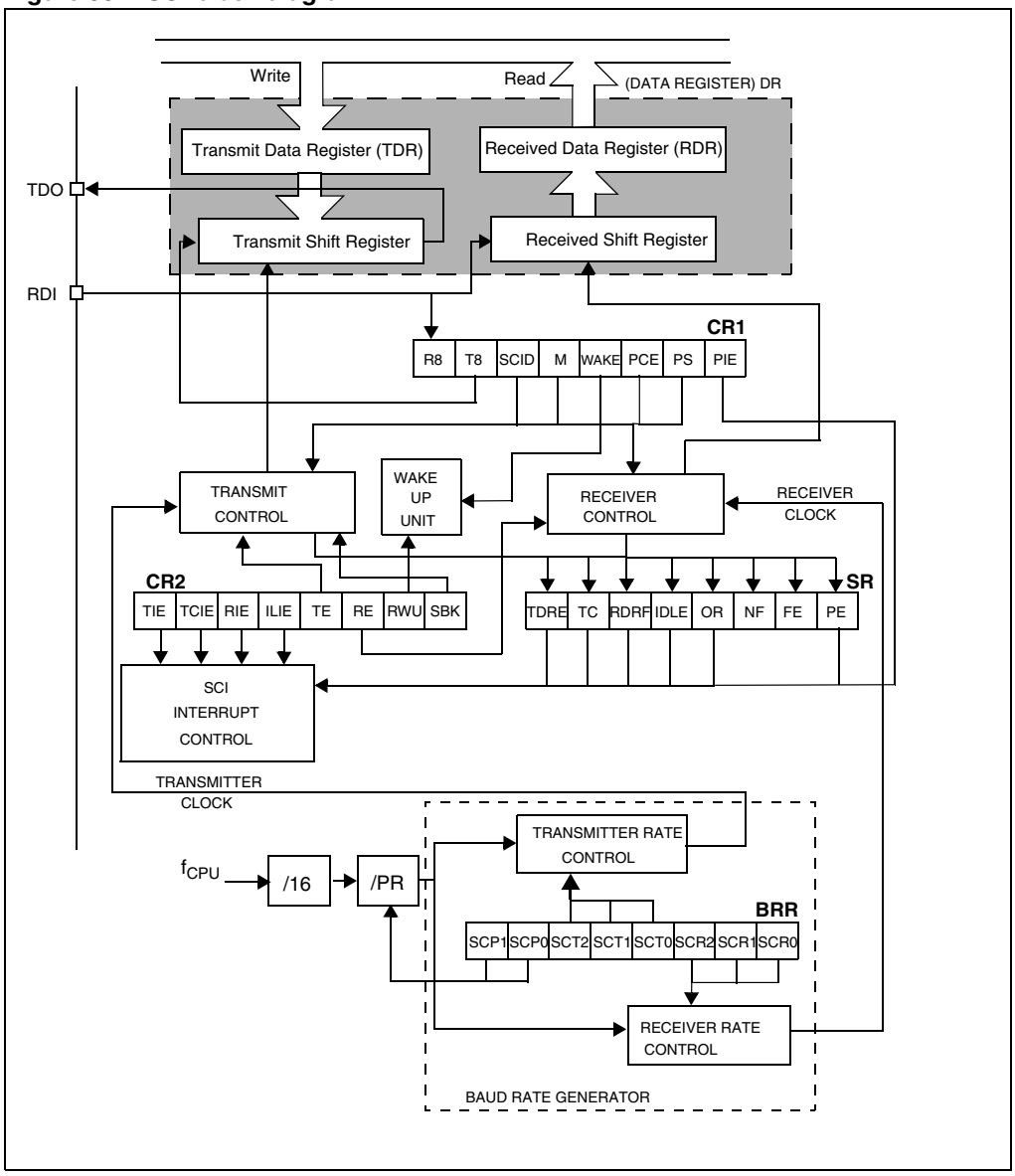
Through these pins, serial data is transmitted and received as frames comprising:

- 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.

Figure 38. SCI block diagram



13.2.2 Functional description

The block diagram of the Serial Control Interface, is shown in *Figure 38*. It contains 6 dedicated registers:

- Two control registers (SCICR1 & SCICR2)
- A status register (SCISR)
- A baud rate register (SCIBRR)

Refer to the register descriptions in *Section 13.3* for the definitions of each bit.

Serial data format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see *Figure 38*).

The TDO pin is in low state during the start bit.

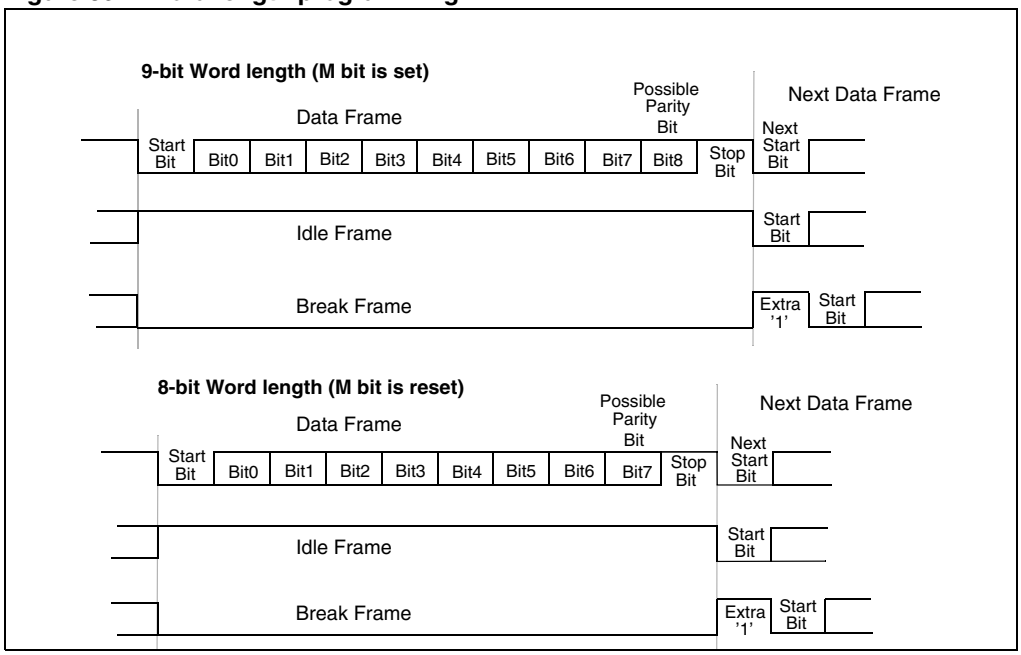
The TDO pin is in high state during the stop bit.

An Idle character is interpreted as an entire frame of “1”s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving “0”s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra “1” bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.

Figure 39. Word length programming



Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. When the M bit is set, word length is 9 bits and the 9th bit (the MSB) has to be stored in the T8 bit in the SCICR1 register.

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see *Figure 38*).

Procedure

- Select the M bit to define the word length.
- Select the desired baud rate using the SCIBRR and the SCIETPR registers.
- Set the TE bit to assign the TDO pin to the alternate function and to send a idle frame as first transmission.
- Access the SCISR register and write the data to send in the SCIDR register (this sequence clears the TDRE bit). Repeat this sequence for each data to be transmitted.

Clearing the TDRE bit is always performed by the following software sequence:

1. An access to the SCISR register
2. A write to the SCIDR register

The TDRE bit is set by hardware and it indicates:

- The TDR register is empty.
- The data transfer is beginning.
- The next data can be written in the SCIDR register without overwriting the previous data.

This flag generates an interrupt if the TIE bit is set and the I bit is cleared in the CCR register.

When a transmission is taking place, a write instruction to the SCIDR register stores the data in the TDR register and which is copied in the shift register at the end of the current transmission.

When no transmission is taking place, a write instruction to the SCIDR register places the data directly in the shift register, the data transmission starts, and the TDRE bit is immediately set.

When a frame transmission is complete (after the stop bit or after the break frame) the TC bit is set and an interrupt is generated if the TCIE is set and the I bit is cleared in the CCR register.

Clearing the TC bit is performed by the following software sequence:

1. An access to the SCISR register
2. A write to the SCIDR register

Note: The TDRE and TC bits are cleared by the same software sequence.

Break characters

Setting the SBK bit loads the shift register with a break character. The break frame length depends on the M bit (see *Figure 39*).

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

Idle characters

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

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

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

Receiver

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

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, the SCIDR register consists or a buffer (RDR) between the internal bus and the received shift register (see <Blue HT>Figure 38).

Procedure

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

When a character is received:

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

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

1. An access to the SCISR register
2. A read to the SCIDR register

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

Break Character

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

Idle Character

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

Overrun Error

An overrun error occurs when a character is received when RDRF has not been reset. Data can not be transferred from the shift register to the RDR register as long as the RDRF bit is not cleared.

When a overrun error occurs:

- The OR bit is set.
- The RDR content will not be lost.
- The shift register will be overwritten.
- An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.

The OR bit is reset by an access to the SCISR register followed by a SCIDR register read operation.

Noise error

Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise. Normal data bits are considered valid if three consecutive samples (8th, 9th, 10th) have the same bit value, otherwise the NF flag is set. In the case of start bit detection, the NF flag is set on the basis of an algorithm combining both valid edge detection and three samples (8th, 9th, 10th). Therefore, to prevent the NF flag getting set during start bit reception, there should be a valid edge detection as well as three valid samples.

When noise is detected in a frame:

- The NF flag is set at the rising edge of the RDRF bit.
- 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 NF flag is reset by a SCISR register read operation followed by a SCIDR register read operation.

During reception, if a false start bit is detected (e.g. 8th, 9th, 10th samples are 011,101,110), the frame is discarded and the receiving sequence is not started for this frame. There is no RDRF bit set for this frame and the NF flag is set internally (not accessible to the user). This NF flag is accessible along with the RDRF bit when a next valid frame is received.

Note: If the application Start Bit is not long enough to match the above requirements, then the NF Flag may get set due to the short Start Bit. In this case, the NF flag may be ignored by the application software when the first valid byte is received.

See also *Noise error causes*.

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 a SCIDR register read operation.

Baud rate generation

The baud rate for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(16*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(16*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits)

TR = 1, 2, 4, 8, 16, 32, 64,128

(see SCT[2:0] bits)

RR = 1, 2, 4, 8, 16, 32, 64,128

(see SCR[2:0] bits)

All these bits are in the SCIBRR register.

Example: If f_{CPU} is 8 MHz (normal mode) and if PR=13 and TR=RR=1, the transmit and receive baud rates are 38400 baud.

Note: The baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.

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 can not 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.

Receiver wakes-up by Idle Line detection when the Receive line has recognised an Idle Frame. Then the RWU bit is reset by hardware but the IDLE bit is not set.

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 a 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 35*.

Table 35. Frame formats

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

Legend: SB = Start Bit, STB = Stop Bit,
PB = Parity Bit

Note: In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Even parity: the parity bit is calculated to obtain an even number of “1s” inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Ex: data=00110101; 4 bits set => parity bit will be 0 if even parity is selected (PS bit = 0).

Odd parity: the parity bit is calculated to obtain an odd number of “1s” inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Ex: data=00110101; 4 bits set => parity bit will be 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an even number of “1s” if even parity is selected (PS=0) or an odd number of “1s” if odd parity is selected (PS=1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PIE is set in the SCICR1 register.

SCI clock tolerance

During reception, each bit is sampled 16 times. The majority of the 8th, 9th and 10th samples is considered as the bit value. For a valid bit detection, all the three samples should have the same value otherwise the noise flag (NF) is set. For example: if the 8th, 9th and 10th samples are 0, 1 and 1 respectively, then the bit value will be “1”, but the Noise Flag bit is be set because the three samples values are not the same.

Consequently, the bit length must be long enough so that the 8th, 9th and 10th samples have the desired bit value. This means the clock frequency should not vary more than 6/16 (37.5%) within one bit. The sampling clock is resynchronized at each start bit, so that when receiving 10 bits (one start bit, 1 data byte, 1 stop bit), the clock deviation must not exceed 3.75%.

Note: The internal sampling clock of the microcontroller samples the pin value on every falling edge. Therefore, the internal sampling clock and the time the application expects the sampling to take place may be out of sync. For example: If the baud rate is 15.625 kbaud (bit length is 64 µs), then the 8th, 9th and 10th samples will be at 28 µs, 32 µs & 36 µs respectively (the first sample starting ideally at 0µs). But if the falling edge of the internal

clock occurs just before the pin value changes, the samples would then be out of sync by ~4 µs. This means the entire bit length must be at least 40 µs (36µs for the 10th sample + 4 µs for synchronization with the internal sampling clock).

Clock deviation causes

The causes which contribute to the total deviation are:

- D_{TRA}: Deviation due to transmitter error (Local oscillator error of the transmitter or the transmitter is transmitting at a different baud rate).
- D_{QUANT}: Error due to the baud rate quantisation of the receiver.
- D_{REC}: Deviation of the local oscillator of the receiver: This deviation can occur during the reception of one complete SCI message assuming that the deviation has been compensated at the beginning of the message.
- D_{TCL}: Deviation due to the transmission line (generally due to the transceivers)

All the deviations of the system should be added and compared to the SCI clock tolerance:

$$D_{TRA} + D_{QUANT} + D_{REC} + D_{TCL} < 3.75\%$$

Noise error causes

See also description of Noise error in *Receiver on page 76*.

Start bit

The noise flag (NF) is set during start bit reception if one of the following conditions occurs:

Note: 1 A valid falling edge is not detected. A falling edge is considered to be valid if the 3 consecutive samples before the falling edge occurs are detected as '1' and, after the falling edge occurs, during the sampling of the 16 samples, if one of the samples numbered 3, 5 or 7 is detected as a "1".

2 During sampling of the 16 samples, if one of the samples numbered 8, 9 or 10 is detected as a "1".

Therefore, a valid Start Bit must satisfy both the above conditions to prevent the Noise Flag getting set.

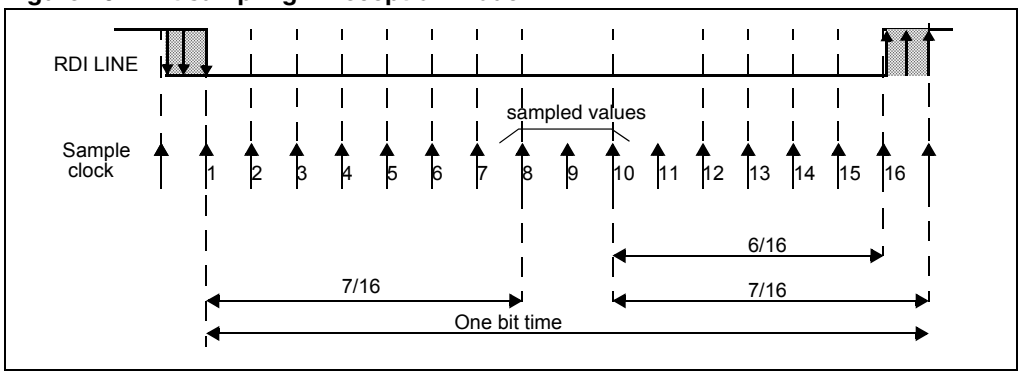
Data bits

The noise flag (NF) is set during normal data bit reception if the following condition occurs:

- During the sampling of 16 samples, if all three samples numbered 8, 9 and10 are not the same. The majority of the 8th, 9th and 10th samples is considered as the bit value.

Therefore, a valid Data Bit must have samples 8, 9 and 10 at the same value to prevent the Noise Flag getting set.

Figure 40. Bit sampling in reception mode



13.2.3 Low power modes

Table 36. Effect of low power modes on SCI

Mode	Description
Wait	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
Halt	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

13.2.4 Interrupts

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 37. SCI interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Transmit data register empty	TDRE	TIE	Yes	No
Transmission complete	TC	TCIE	Yes	No
Received data ready to be read	RDRF	RIE	Yes	No
Overrun error detected	OR		Yes	No
Idle line detected	IDLE	ILIE	Yes	No
Parity error	PE	PIE	Yes	No

13.3 Register description

13.3.1 Status register (SCISR)

SCISR Reset value: 1100 0000 (C0h)

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE
R	R	R	R	R	R	R	R

Table 38. 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).*

Table 38. SCISR register description (continued)

Bit	Name	Function
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.*
1	FE	Framing Error This bit is set by hardware when a desynchronization, excessive noise or a break character is detected. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No framing error is detected 1: Framing error or break character 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. If the word currently being transferred causes both Frame Error and Overrun error, it is transferred and only the OR bit will be set.*
0	PE	Parity Error This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by a software sequence (a read to the status register followed by an access to the SCIDR data register). An interrupt is generated if PIE = 1 in the SCICR1 register. 0: No parity error 1: Parity error

13.3.2 Control register 1 (SCICR1)

SCICR1 Reset value: x000 0000 (x0h)

7	6	5	4	3	2	1	0
R8	T8	SCID	M	WAKE	PCE	PS	PIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39. SCICR1 register description

Bit	Name	Function
7	R8	Receive data bit 8 This bit is used to store the 9th bit of the received word when M = 1.
6	T8	Transmit data bit 8 This bit is used to store the 9th bit of the transmitted word when M = 1.
5	SCID	Disabled for low power consumption When this bit is set the SCI prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption.This bit is set and cleared by software. 0: SCI enabled 1: SCI prescaler and outputs disabled
4	M	Word length This bit determines the word length. It is set or cleared by software. 0: 1 Start bit, 8 data bits, 1 Stop bit 1: 1 Start bit, 9 data bits, 1 Stop bit *Note: The M bit must not be modified during a data transfer (both transmission and reception).*
3	WAKE	Wake-Up method This bit determines the SCI Wake-Up method, it is set or cleared by software. 0: Idle line 1: Address mark
2	PCE	Parity Control Enable This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M = 1; 8th bit if M = 0) and parity is checked on the received data. This bit is set and cleared by software. Once it is set, PCE is active after the current byte (in reception and in transmission). 0: Parity control disabled 1: Parity control enabled

Table 39. SCICR1 register description (continued)

Bit	Name	Function
1	PS	Parity Selection This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity will be selected after the current byte. 0: Even parity 1: Odd parity
0	PIE	Parity Interrupt Enable This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). It is set and cleared by software. 0: Parity error interrupt disabled 1: Parity error interrupt enabled

13.3.3 Control register 2 (SCICR2)

SCICR2 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40. SCICR2 register description

Bit	Name	Function
7	TIE	Transmitter Interrupt Enable This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever TDRE = 1 in the SCISR register.
6	TCIE	Transmission Complete Interrupt Enable This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever TC = 1 in the SCISR register.
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.

Table 40. SCICR2 register description (continued)

Bit	Name	Function
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.*

13.3.4 Data register (SCIDR)

SCIDR — Reset value: undefined (xxh)

7	6	5	4	3	2	1	0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).
The TDR register provides the parallel interface between the internal bus and the output shift register (see *Figure 38*).
The RDR register provides the parallel interface between the input shift register and the internal bus (see *Figure 38*).

13.3.5 Baud rate register (SCIBRR)

SCIBRR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SCP[1:0]		SCT[2:0]			SCR[2:0]		
R/W		R/W			R/W		

Table 41. SCIBRR register description

Bit	Name	Function
7:6	SCP[1:0]	First SCI Prescaler These 2 prescaling bits allow several standard clock division ranges. 00: PR prescaling factor = 1 01: PR prescaling factor = 3 10: PR prescaling factor = 4 11: PR prescaling factor = 13
5:3	SCT[2:0]	SCI Transmitter rate divisor These 3 bits, in conjunction with the SCP1 and SCP0 bits, define the total division applied to the bus clock to yield the transmit rate clock in conventional baud rate generator mode. 000: TR dividing factor = 1 001: TR dividing factor = 2 010: TR dividing factor = 4 011: TR dividing factor = 8 100: TR dividing factor = 16 101: TR dividing factor = 32 110: TR dividing factor = 64 111: TR dividing factor = 128
2:0	SCR[2:0]	SCI Receiver rate divisor These 3 bits, in conjunction with the SCP[1:0] bits, define the total division applied to the bus clock to yield the receive rate clock in conventional baud rate generator mode. 000: RR dividing factor = 1 001: RR dividing factor = 2 010: RR dividing factor = 4 011: RR dividing factor = 8 100: RR dividing factor = 16 101: RR dividing factor = 32 110: RR dividing factor = 64 111: RR dividing factor = 128

Table 42. SCI register map and reset values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
20	SCISR Reset Value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
21	SCIDR Reset Value	DR7 x	DR6 x	DR5 x	DR4 x	DR3 x	DR2 x	DR1 x	DR0 x

Table 42. SCI register map and reset values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
22	SCIBRR	SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0
	Reset Value	0	0	x	x	x	x	x	x
23	SCICR1	R8	T8	SCID	M	WAKE	PCE	PS	PIE
	Reset Value	x	x	0	x	x	0	0	0
24	SCICR2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	Reset Value	0	0	0	0	0	0	0	0

14 USB interface (USB)

14.1 Introduction

The USB Interface implements a low-speed function interface between the USB and the ST7 microcontroller. It is a highly integrated circuit which includes the transceiver, 3.3 voltage regulator, SIE and DMA. No external components are needed apart from the external pull-up on USBDM for low speed recognition by the USB host. The use of DMA architecture allows the endpoint definition to be completely flexible. Endpoints can be configured by software as in or out.

14.2 Main features

- USB specification version 1.1 compliant
- Supports Low-Speed USB protocol
- Two or three Endpoints (including default one) depending on the device (see device feature list and register map)
- CRC generation/checking, NRZI encoding/decoding and bit-stuffing
- USB Suspend/Resume operations
- DMA data transfers
- On-chip 3.3V regulator
- On-chip USB transceiver

14.3 Functional description

The block diagram in *Figure 41*, gives an overview of the USB interface hardware.

For general information on the USB, refer to the “Universal Serial Bus Specifications” document available at http//:www.usb.org.

Serial interface engine

The SIE (Serial Interface Engine) interfaces with the USB, via the transceiver.

The SIE processes tokens, handles data transmission/reception, and handshaking as required by the USB standard. It also performs frame formatting, including CRC generation and checking.

Endpoints

The Endpoint registers indicate if the microcontroller is ready to transmit/receive, and how many bytes need to be transmitted.

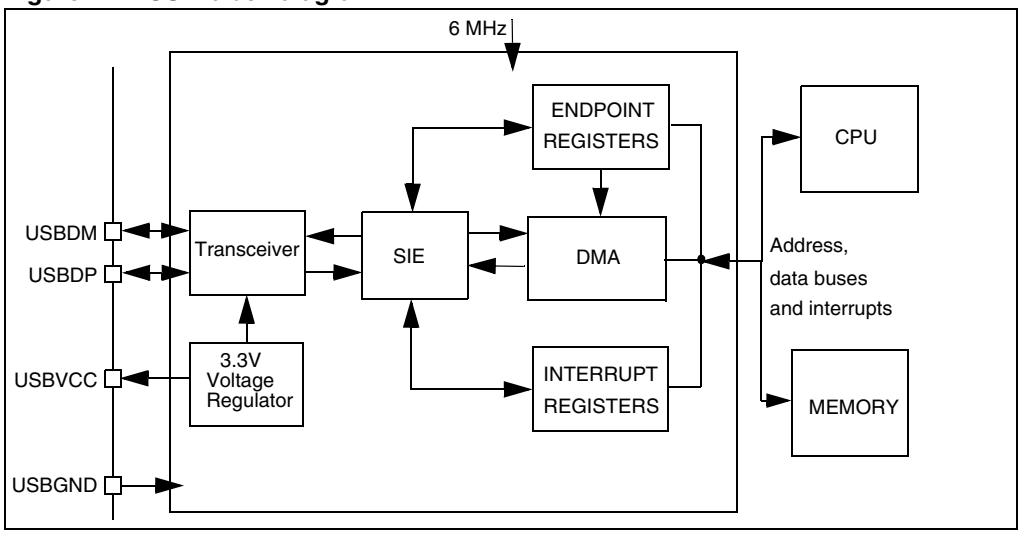
DMA

When a token for a valid Endpoint is recognized by the USB interface, the related data transfer takes place, using DMA. At the end of the transaction, an interrupt is generated.

Interrupts

By reading the Interrupt Status register, application software can know which USB event has occurred.

Figure 41. USB block diagram



14.4 Register description

14.4.1 DMA address register (DMAR)

DMAR — Reset value: undefined (xxh)

7	6	5	4	3	2	1	0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7:0=**DA[15:8]** *DMA address bits 15-8.*
Software must write the start address of the DMA memory area whose most significant bits are given by DA15-DA6. The remaining 6 address bits are set by hardware. See the description of the IDR register and *Figure 42*.

14.4.2 Interrupt/DMA register (IDR)

IDR — Reset value: xxxx 0000 (x0h)

7	6	5	4	3	2	1	0
DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7:6 = **DA[7:6]** *DMA address bits 7-6.*
Software must reset these bits. See the description of the DMAR register and *Figure 42*.

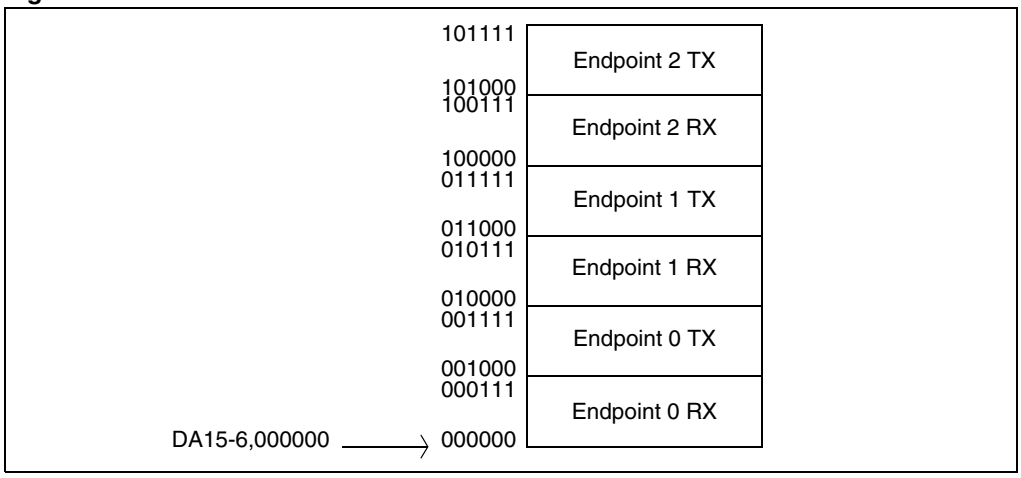
Bits 5:4 = **EP[1:0]** *Endpoint number* (read-only). These bits identify the endpoint which required attention.
00: Endpoint 0
01: Endpoint 1
10: Endpoint 2

When a CTR interrupt occurs (see register ISTR) the software should read the EP bits to identify the endpoint which has sent or received a packet.

Bits 3:0 = **CNT[3:0]** *Byte count* (read only).
This field shows how many data bytes have been received during the last data reception.

Note: Not valid for data transmission.

Figure 42. DMA buffers



14.4.3 PID register (PIDR)

PIDR Reset value: xxxx 0000 (x0h)

7	6	5	4	3	2	1	0
TP3	TP2	0	0	0	RX_ SEZ	RXD	0
R	R	R	R	R	R	R	R

Bits 7:6 = **TP[3:2]** *Token PID bits 3 & 2*.
USB token PIDs are encoded in four bits. **TP[3:2]** correspond to the variable token PID bits 3 & 2.

Note: PID bits 1 & 0 have a fixed value of 01.

When a CTR interrupt occurs (see register ISTR) the software should read the TP3 and TP2 bits to retrieve the PID name of the token received.
The USB standard defines TP bits as:

Table 43. TP bits

TP3	TP2	PID name
0	0	OUT
1	0	IN
1	1	SETUP

Bits 5:3 Reserved. Forced by hardware to 0.

Bit 2 = **RX_SEZ** *Received single-ended zero*
This bit indicates the status of the RX_SEZ transceiver output.
0: No SE0 (single-ended zero) state
1: USB lines are in SE0 (single-ended zero) state

Bit 1 = **RXD** *Received data*
0: No K-state
1: USB lines are in K-state

This bit indicates the status of the RXD transceiver output (differential receiver output).

Note: If the environment is noisy, the RX_SEZ and RXD bits can be used to secure the application. By interpreting the status, software can distinguish a valid End Suspend event from a spurious wake-up due to noise on the external USB line. A valid End Suspend is followed by a Resume or Reset sequence. A Resume is indicated by RXD=1, a Reset is indicated by RX_SEZ=1.

Bit 0 = Reserved. Forced by hardware to 0.

14.4.4 Interrupt status register (ISTR)

ISTR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SUSP	DOVR	CTR	ERR	IOVR	ESUSP	RESET	SOF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When an interrupt occurs these bits are set by hardware. Software must read them to determine the interrupt type and clear them after servicing.

Note: These bits cannot be set by software.

Bit 7 = **SUSP** *Suspend mode request*.
This bit is set by hardware when a constant idle state is present on the bus line for more than 3 ms, indicating a suspend mode request from the USB bus. The suspend request check is active immediately after each USB reset event and its disabled by hardware when suspend mode is forced (FSUSP bit of CTLR register) until the end of resume sequence.

Bit 6 = **DOVR** *DMA over/underrun*.
This bit is set by hardware if the ST7 processor can't answer a DMA request in time.
0: No over/underrun detected
1: Over/underrun detected

Bit 5 = **CTR** *Correct Transfer.* This bit is set by hardware when a correct transfer operation is performed. The type of transfer can be determined by looking at bits TP3-TP2 in register PIDR. The Endpoint on which the transfer was made is identified by bits EP1-EP0 in register IDR.
0: No Correct Transfer detected
1: Correct Transfer detected

Note: A transfer where the device sent a NAK or STALL handshake is considered not correct (the host only sends ACK handshakes). A transfer is considered correct if there are no errors in the PID and CRC fields, if the DATA0/DATA1 PID is sent as expected, if there were no data overruns, bit stuffing or framing errors.

Bit 4 = **ERR** *Error.*
This bit is set by hardware whenever one of the errors listed below has occurred:

0: No error detected
1: Timeout, CRC, bit stuffing or nonstandard framing error detected

Bit 3 = **IOVR** *Interrupt overrun.*
This bit is set when hardware tries to set ERR, or SOF before they have been cleared by software.
0: No overrun detected
1: Overrun detected

Bit 2 = **ESUSP** *End suspend mode.*
This bit is set by hardware when, during suspend mode, activity is detected that wakes the USB interface up from suspend mode.

This interrupt is serviced by a specific vector, in order to wake up the ST7 from HALT mode.
0: No End Suspend detected
1: End Suspend detected

Bit 1 = **RESET** *USB reset.*
This bit is set by hardware when the USB reset sequence is detected on the bus.
0: No USB reset signal detected
1: USB reset signal detected

Note: The DADDR, EP0RA, EP0RB, EP1RA, EP1RB, EP2RA and EP2RB registers are reset by a USB reset.

Bit 0 = **SOF** *Start of frame.*
This bit is set by hardware when a low-speed SOF indication (keep-alive strobe) is seen on the USB bus. It is also issued at the end of a resume sequence.
0: No SOF signal detected
1: SOF signal detected

Note: To avoid spurious clearing of some bits, it is recommended to clear them using a load instruction where all bits which must not be altered are set, and all bits to be cleared are reset. Avoid read-modify-write instructions like AND , XOR..

14.4.5 Interrupt mask register (IMR)

IMR — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SUSPM	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7:0 = These bits are mask bits for all interrupt condition bits included in the ISTR. Whenever one of the IMR bits is set, if the corresponding ISTR bit is set, and the I bit in the CC register is cleared, an interrupt request is generated. For an explanation of each bit, please refer to the corresponding bit description in ISTR.

14.4.6 Control register (CTLR)

CTLR — Reset value: 0000 0110 (06h)

7	6	5	4	3	2	1	0
0	0	0	0	RESUME	PDWN	FSUSP	FRES
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7:4 = Reserved. Forced by hardware to 0.

Bit 3 = **RESUME** *Resume*.
This bit is set by software to wake-up the Host when the ST7 is in suspend mode.
0: Resume signal not forced
1: Resume signal forced on the USB bus.

Software should clear this bit after the appropriate delay.

Bit 2 = **PDWN** *Power down*.
This bit is set by software to turn off the 3.3V on-chip voltage regulator that supplies the external pull-up resistor and the transceiver.
0: Voltage regulator on
1: Voltage regulator off

Note: After turning on the voltage regulator, software should allow at least 3 µs for stabilisation of the power supply before using the USB interface.

Bit 1 = **FSUSP** *Force suspend mode*.
This bit is set by software to enter Suspend mode. The ST7 should also be halted allowing at least 600 ns before issuing the HALT instruction.
0: Suspend mode inactive
1: Suspend mode active

When the hardware detects USB activity, it resets this bit (it can also be reset by software).

Bit 0 = **FRES** *Force reset.*
This bit is set by software to force a reset of the USB interface, just as if a RESET sequence came from the USB.
0: Reset not forced
1: USB interface reset forced.

The USB is held in RESET state until software clears this bit, at which point a “USB-RESET” interrupt will be generated if enabled.

14.4.7 Device address register (DADDR)

DADDR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7 = Reserved. Forced by hardware to 0.

Bits 6:0 = **ADD[6:0]** *Device address, 7 bits.*

Software must write into this register the address sent by the host during enumeration.

Note: This register is also reset when a USB reset is received from the USB bus or forced through bit FRES in the CTLR register.

14.4.8 Endpoint n register A (EPnRA)

EPnRA Reset value: 0000 xxxx (0xh)

7	6	5	4	3	2	1	0
ST_ OUT	DTOG _TX	STAT _TX1	STAT _TX0	TBC3	TBC2	TBC1	TBC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These registers (**EP0RA**, **EP1RA** and **EP2RA**) are used for controlling data transmission. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RA register are not available on some devices (see device feature list and register map).

Bit 7 = **ST_OUT** *Status out.*
This bit is set by software to indicate that a status out packet is expected: in this case, all nonzero OUT data transfers on the endpoint are STALLed instead of being ACKed. When ST_OUT is reset, OUT transactions can have any number of bytes, as needed.

Bit 6 = **DTOG_TX** *Data Toggle, for transmission transfers.*
It contains the required value of the toggle bit (0=DATA0, 1=DATA1) for the next transmitted data packet. This bit is set by hardware at the reception of a SETUP PID. DTOG_TX toggles only when the transmitter has received the ACK signal from the USB host. DTOG_TX and also DTOG_RX (see EPnRB) are normally updated by hardware, at the receipt of a relevant PID. They can be also written by software.

Bits 5:4 = **STAT_TX[1:0]** *Status bits, for transmission transfers.*
These bits contain the information about the endpoint status, which are listed below:

Table 44. STAT_TX bits

STAT_TX1	STAT_TX0	Meaning
0	0	**DISABLED:** transmission transfers cannot be executed.
0	1	**STALL**: the endpoint is stalled and all transmission requests result in a STALL handshake.
1	0	**NAK**: the endpoint is naked and all transmission requests result in a NAK handshake.
1	1	**VALID**: this endpoint is enabled for transmission.

These bits are written by software. Hardware sets the STAT_TX bits to NAK when a correct transfer has occurred (CTR=1) related to a IN or SETUP transaction addressed to this endpoint; this allows the software to prepare the next set of data to be transmitted.

Bits 3:0 = **TBC[3:0]** *Transmit byte count for Endpoint n.*
Before transmission, after filling the transmit buffer, software must write in the TBC field the transmit packet size expressed in bytes (in the range 0-8).

Caution: Any value outside the range 0-8 willinduce undesired effects (such as continuous data transmission).

14.4.9 Endpoint n register B (EPnRB)

EPnRB Reset value: 0000 xxxx (0xh)

7	6	5	4	3	2	1	0
CTRL	DTOG _RX	STAT _RX1	STAT _RX0	EA3	EA2	EA1	EA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These registers (**EP1RB** and **EP2RB**) are used for controlling data reception on Endpoints 1 and 2. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RB register are not available on some devices (see device feature list and register map).

Bit 7 = **CTRL** *Control.*
This bit should be 0.

Note: If this bit is 1, the Endpoint is a control endpoint. (Endpoint 0 is always a control Endpoint, but it is possible to have more than one control Endpoint).

Bit 6 = **DTOG_RX** *Data toggle, for reception transfers.*
It contains the expected value of the toggle bit (0=DATA0, 1=DATA1) for the next data packet. This bit is cleared by hardware in the first stage (Setup Stage) of a control transfer (SETUP transactions start always with DATA0 PID). The receiver toggles DTOG_RX only if it receives a correct data packet and the packet's data PID matches the receiver sequence bit.

Bits 5:4 = **STAT_RX [1:0]** *Status bits, for reception transfers.*
These bits contain the information about the endpoint status, which are listed below:

Table 45. STAT_RX bits

STAT_RX1	STAT_RX0	Meaning
0	0	**DISABLED**: reception transfers cannot be executed.
0	1	**STALL:** the endpoint is stalled and all reception requests result in a STALL handshake.
1	0	**NAK**: the endpoint is naked and all reception requests result in a NAK handshake.
1	1	**VALID**: this endpoint is enabled for reception.

These bits are written by software. Hardware sets the STAT_RX bits to NAK when a correct transfer has occurred (CTR=1) related to an OUT or SETUP transaction addressed to this endpoint, so the software has the time to elaborate the received data before acknowledging a new transaction.

Bits 3:0 = **EA[3:0]** *Endpoint address*.
Software must write in this field the 4-bit address used to identify the transactions directed to this endpoint. Usually EP1RB contains “0001” and EP2RB contains “0010”.

14.4.10 Endpoint 0 register B (EP0RB)

EP0RB Reset value: 1000 0000 (80h)

7	6	5	4	3	2	1	0
1	DTOG RX	STAT RX1	STAT RX0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used for controlling data reception on Endpoint 0. It is also reset by the USB bus reset.

Bit 7 = Forced by hardware to 1.

Bits 6:4 = Refer to the EPnRB register for a description of these bits.

Bits 3:0 = Forced by hardware to 0.

14.5 Programming considerations

The interaction between the USB interface and the application program is described below. Apart from system reset, action is always initiated by the USB interface, driven by one of the USB events associated with the Interrupt Status Register (ISTR) bits.

14.5.1 Initializing the registers

At system reset, the software must initialize all registers to enable the USB interface to properly generate interrupts and DMA requests.

1. Initialize the DMAR, IDR, and IMR registers (choice of enabled interrupts, address of DMA buffers). Refer the paragraph titled initializing the DMA Buffers.
2. Initialize the EP0RA and EP0RB registers to enable accesses to address 0 and endpoint 0 to support USB enumeration. Refer to the paragraph titled Endpoint Initialization.
3. When addresses are received through this channel, update the content of the DADDR.
4. If needed, write the endpoint numbers in the EA fields in the EP1RB and EP2RB register.

14.5.2 Initializing DMA buffers

The DMA buffers are a contiguous zone of memory whose maximum size is 48 bytes. They can be placed anywhere in the memory space to enable the reception of messages. The 10 most significant bits of the start of this memory area are specified by bits DA15-DA6 in registers DMAR and IDR, the remaining bits are 0. The memory map is shown in *Figure 42*.

Each buffer is filled starting from the bottom (last 3 address bits=000) up.

14.5.3 Endpoint initialization

To be ready to receive:

Set STAT_RX to VALID (11b) in EP0RB to enable reception.

To be ready to transmit:

1. Write the data in the DMA transmit buffer.
2. In register EPnRA, specify the number of bytes to be transmitted in the TBC field
3. Enable the endpoint by setting the STAT_TX bits to VALID (11b) in EPnRA.

Note: Once transmission and/or reception are enabled, registers EPnRA and/or EPnRB (respectively) must not be modified by software, as the hardware can change their value on the fly.

When the operation is completed, they can be accessed again to enable a new operation.

14.5.4 Interrupt handling

Start of frame (SOF)

The interrupt service routine may monitor the SOF events for a 1 ms synchronization event to the USB bus. This interrupt is generated at the end of a resume sequence and can also be used to detect this event.

USB reset (RESET)

When this event occurs, the DADDR register is reset, and communication is disabled in all endpoint registers (the USB interface will not respond to any packet). Software is responsible for reenabling endpoint 0 within 10 ms of the end of reset. To do this, set the STAT_RX bits in the EP0RB register to VALID.

Suspend (SUSP)

The CPU is warned about the lack of bus activity for more than 3 ms, which is a suspend request. The software should set the USB interface to suspend mode and execute an ST7 HALT instruction to meet the USB-specified power constraints.

End suspend (ESUSP)

The CPU is alerted by activity on the USB, which causes an ESUSP interrupt. The ST7 automatically terminates HALT mode.

Correct transfer (CTR)

1. When this event occurs, the hardware automatically sets the STAT_TX or STAT_RX to NAK.

Note: Every valid endpoint is NAKed until software clears the CTR bit in the ISTR register, independently of the endpoint number addressed by the transfer which generated the CTR interrupt.

Note: If the event triggering the CTR interrupt is a SETUP transaction, both STAT_TX and STAT_RX are set to NAK.

2. Read the PIDR to obtain the token and the IDR to get the endpoint number related to the last transfer.

Note: When a CTR interrupt occurs, the TP3-TP2 bits in the PIDR register and EP1-EP0 bits in the IDR register stay unchanged until the CTR bit in the ISTR register is cleared.

3. Clear the CTR bit in the ISTR register.

Table 46. USB register map and reset values

Address (Hex.)	Register name	7	6	5	4	3	2	1	0
25	PIDR Reset Value	TP3 x	TP2 x	0 0	0 0	0 0	RX_SEZ 0	RXD 0	0 0
26	DMAR Reset Value	DA15 x	DA14 x	DA13 x	DA12 x	DA11 x	DA10 x	DA9 x	DA8 x
27	IDR Reset Value	DA7 x	DA6 x	EP1 x	EP0 x	CNT3 0	CNT2 0	CNT1 0	CNT0 0
28	ISTR Reset Value	SUSP 0	DOVR 0	CTR 0	ERR 0	IOVR 0	ESUSP 0	RESET 0	SOF 0
29	IMR Reset Value	SUSPM 0	DOVRM 0	CTRM 0	ERRM 0	IOVRM 0	ESUSPM 0	RESETM 0	SOFM 0
2A	CTLR Reset Value	0 0	0 0	0 0	0 0	RESUME 0	PDWN 1	FSUSP 1	FRES 0
2B	DADDR Reset Value	0 0	ADD6 0	ADD5 0	ADD4 0	ADD3 0	ADD2 0	ADD1 0	ADD0 0
2C	EP0RA Reset Value	ST_OUT 0	DTOG_TX 0	STAT_TX1 0	STAT_TX0 0	TBC3 x	TBC2 x	TBC1 x	TBC0 x
2D	EP0RB Reset Value	1 1	DTOG_RX 0	STAT_RX 1 0	STAT_RX 0 0	0 0	0 0	0 0	0 0

Table 46. USB register map and reset values (continued)

Address (Hex.)	Register name	7	6	5	4	3	2	1	0
2E	EP1RA Reset Value	ST_OUT 0	DTOG_TX 0	STAT_TX1 0	STAT_TX0 0	TBC3 x	TBC2 x	TBC1 x	TBC0 x
2F	EP1RB Reset Value	CTRL 0	DTOG_RX 0	STAT_RX 1 0	STAT_RX 0 0	EA3 x	EA2 x	EA1 x	EA0 x
30	EP2RA Reset Value	ST_OUT 0	DTOG_TX 0	STAT_TX1 0	STAT_TX0 0	TBC3 x	TBC2 x	TBC1 x	TBC0 x
31	EP2RB Reset Value	CTRL 0	DTOG_RX 0	STAT_RX 1 0	STAT_RX 0 0	EA3 x	EA2 x	EA1 x	EA0 x

15 Instruction set

15.1 ST7 addressing modes

The ST7 Core features 17 different addressing modes which can be classified in 7 main groups:

Table 47. Addressing mode groups

Addressing mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 Instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two sub-modes called long and short:

- Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.
- Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 48. ST7 addressing mode overview

Mode			Syntax	Destination/ source	Pointer address	Pointer size	Length (bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00..FF			+ 0 (with X register) + 1 (with Y register)
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2

Table 48. ST7 addressing mode overview (continued)

Mode			Syntax	Destination/ source	Pointer address	Pointer size	Length (bytes)
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC-128/PC+127(1)			+ 1
Relative	Indirect		jrne [$10]	PC-128/PC+127(1)	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

1. At the time the instruction is executed, the Program Counter (PC) points to the instruction following JRxx.

15.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Table 49. Inherent instructions

Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask
RIM	Reset Interrupt Mask
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication

Table 49. Inherent instructions

Instruction	Function
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

15.1.2 Immediate

Immediate instructions have two bytes, the first byte contains the opcode, the second byte contains the operand value.

Table 50. Immediate instructions

Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

15.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two sub-modes:

Direct (short)

The address is a byte, thus requires only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

15.1.4 Indexed (no offset, short, long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three sub-modes:

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 one 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.

15.1.5 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 sub-modes:

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.

15.1.6 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 sub-modes:

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 51. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes

Instructions		Function
Long and short instructions	LD	Load
	CP	Compare
	AND, OR, XOR	Logical Operations
	ADC, ADD, SUB, SBC	Arithmetic Addition/subtraction operations
	BCP	Bit Compare
Short instructions only	CLR	Clear
	INC, DEC	Increment/Decrement
	TNZ	Test Negative or Zero
	CPL, NEG	1 or 2 Complement
	BSET, BRES	Bit Operations
	BTJT, BTJF	Bit Test and Jump Operations
	SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
	SWAP	Swap Nibbles
	CALL, JP	Call or Jump subroutine

15.1.7 Relative mode (direct, indirect)

This addressing mode is used to modify the PC register value by adding an 8-bit signed offset to it.

Table 52. Available relative direct/indirect instructions

Instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two sub-modes:

Relative (direct)

The offset follows the opcode.

Relative (indirect)

The offset is defined in memory, of which the address follows the opcode.

15.2 Instruction groups

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Table 53. Instruction groups

Group	Instruction							
Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx							
Interruption management	TRAP	WFI	HALT	IRET				
Condition Code Flag modification	SIM	RIM	SCF	RCF				

Using a pre-byte

The instructions are described with one to four bytes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC Opcode

PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode. It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

Table 54. Instruction set

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A . M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							

Table 54. Instruction set (continued)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRUGT	Jump if (C + Z = 0)	Unsigned >							
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg	reg	M					
		pop CC	CC	M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= Dst <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => Dst => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= Dst <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= Dst <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => Dst => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	Dst7 => Dst => C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	Dst[7..4] <=> Dst[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz lbl1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

16 Electrical characteristics

16.1 Parameter conditions

Unless otherwise specified, all voltages are referred to V_{SS}.

16.1.1 Minimum and maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at T_A=25°C and T_A=T_{Amax} (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean±3s).

16.1.2 Typical values

Unless otherwise specified, typical data are based on T_A=25°C, V_{DD}=5V. They are given only as design guidelines and are not tested.

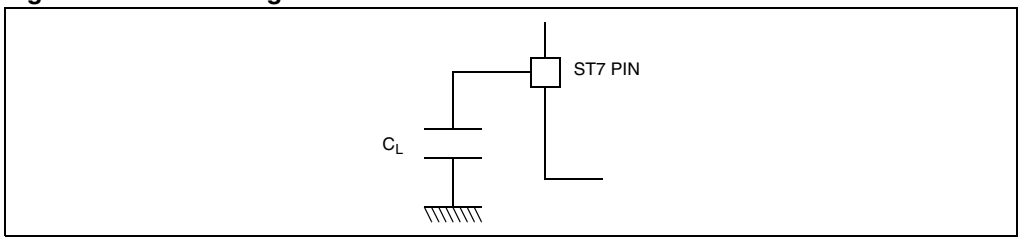
16.1.3 Typical curves

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

16.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 43*.

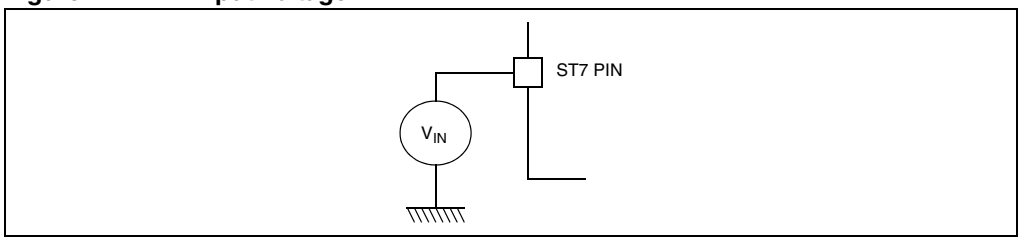
Figure 43. Pin loading conditions



16.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 44*.

Figure 44. Pin input voltage



16.2 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 55. Voltage characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.0	V
V_{IN} (1) & (2)	Input voltage on true open drain pins	V_{SS}-0.3 to 6.0	
	Input voltage on any other pin	V_{SS}-0.3 to V_{DD}+0.3	
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human Body Model)	See “Absolute maximum ratings (electrical sensitivity)” on page 119.	

1. Directly connecting the RESET and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 4.7kΩ for RESET, 10kΩ for I/Os). Unused I/O pins must be tied in the same way to V_{DD} or V_{SS} according to their reset configuration.

Table 56. Current characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) (1)	80	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) (1)	80	
I_{IO}	Output current sunk by any standard I/O and control pin	25	
	Output current sunk by any high sink I/O pin	50	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}$ (2)	Injected current on V_{PP} pin	± 5	
	Injected current on $\overline{\text{RESET}}$ pin	± 5	
	Injected current on OSCIN and OSCOUT pins	± 5	
	Injected current on any other pin (3) & (4)	± 5	
$\Sigma I_{INJ(PIN)}$ (2)	Total injected current (sum of all I/O and control pins) (3)	± 20	
$I_{INJ(PIN)}$ (2)	Negative injected current to PB0(10mA) pin	- 80	µA

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. 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.
4. True open drain I/O port pins do not accept positive injection.

Table 57. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature: See *Section 17.1.1 on page 130* for T_{Jmax}		

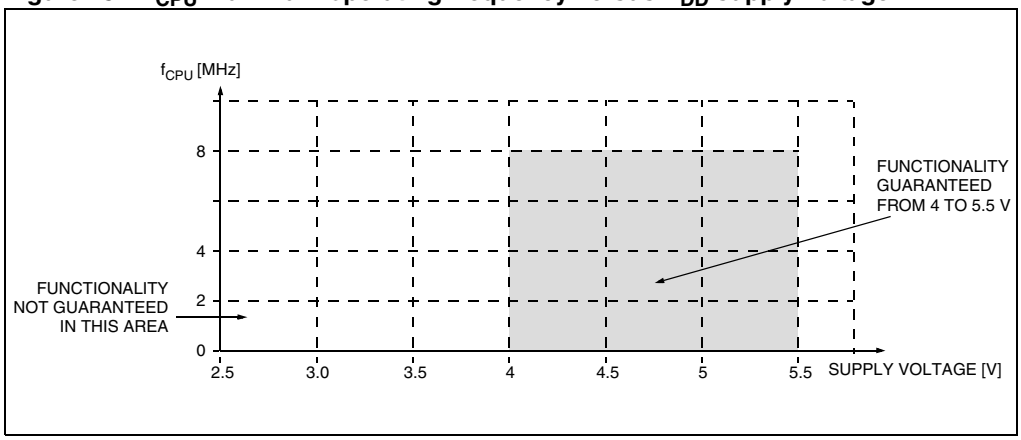
16.3 Operating conditions

16.3.1 General operating conditions

Table 58. General operating condition

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating supply voltage	f_{CPU} = 8 MHz	4	5	5.5	V
V_{DDA}	Analog supply voltage		V_{DD}		V_{DD}	
V_{SSA}	Analog supply voltage		V_{SS}		V_{SS}	
f_{CPU}	Operating frequency	f_{OSC} = 24 MHz			8	MHz
		f_{OSC} = 12 MHz			4	
T_A	Ambient temperature range		0		70	°C

Figure 45. f_{CPU} maximum operating frequency versus V_{DD} supply voltage



16.3.2 Operating conditions with low voltage detector (LVD)

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A. Refer to *Figure 9 on page 26*.

Table 59. Operating conditions at power-up/power-down

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IT+}	Low Voltage Reset Threshold (V_{DD} rising)	V_{DD} Max. Variation 50 V/ms	3.4	3.7	4.0	V
V_{IT-}	Low Voltage Reset Threshold (V_{DD} falling)	V_{DD} Max. Variation 50 V/ms	3.2	3.5	3.8	V
V_{hyst}	Hysteresis (V_{IT+} - V_{IT-}) (1)		100	175	220	mV
Vt_{POR}	V_{DD} rise time rate (2)		0.5		50	V/ms

1. Guaranteed by characterization - not tested in production
2. The V_{DD} rise time rate condition is needed to insure a correct device power-on and LVD reset. Not tested in production.

16.4 Supply current characteristics

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

Table 60. Supply current consumption

Symbol	Parameter	Conditions		Typ	Max	Unit
$\Delta I_{DD(\Delta Ta)}$	Supply current variation vs. temperature	Constant V_{DD} and f_{CPU}			10 (1)	%
I_{DD}	CPU RUN mode	I/Os in input mode	f_{CPU} = 4 MHz	7.5	9 (2)(1)	mA
			f_{CPU} = 8 MHz	10.5	13 (2)	
	CPU WAIT mode		f_{CPU} = 4 MHz	6	8 (1)	mA
			f_{CPU} = 8 MHz	8.5	11 (2)	
	CPU HALT mode (3)	LVD disabled		25	40 (1)	µA
	USB Suspend mode (4)	LVD disabled		100	120	µA
		LVD enabled		230		

1. Not tested in production, guaranteed by characterization.
2. Oscillator and watchdog running. All others peripherals disabled.
3. USB Transceiver is powered down.
4. CPU in Halt mode. Current consumption of external pull-up (1.5 Kohms to USBVCC) and pull-down (15 Kohms to V_{SSA}) not included.

Figure 46. Typ. I_{DD} in RUN at 4 and 8 MHz f_{CPU}

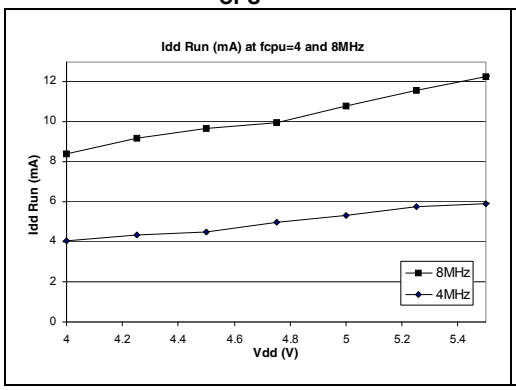
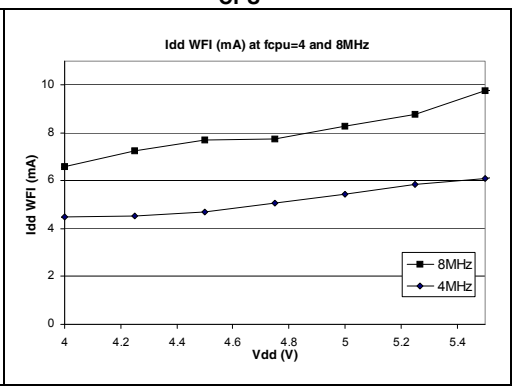


Figure 47. Typ. I_{DD} in WAIT at 4 and 8 MHz f_{CPU}



16.5 Clock and timing characteristics

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

16.5.1 General timings

Table 61. CPU timings

Symbol	Parameter	Conditions	Min	Typ (1)	Max	Unit
$t_{c(INST)}$	Instruction cycle time	f_{CPU}=8 MHz	2	3	12	t_{CPU}
			250	375	1500	ns
$t_{v(IT)}$	Interrupt reaction time (2) $t_{v(IT)} = \Delta t_{c(INST)} + 10\ t_{CPU}$	f_{CPU}=8 MHz	10		22	t_{CPU}
			1.25		2.75	µs

1. Data based on typical application software.
2. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.

16.5.2 Control timing characteristics

Table 62. Control timings

Symbol	Parameter	Conditions	Value			Unit
			Min	Typ	Max	
f_{OSC}	Oscillator frequency				24	MHz
f_{CPU}	Operating frequency				8	MHz
t_{RL}	External RESET input pulse width		2520			ns
t_{PORL}	Internal power reset duration		4096			t_{CPU}
t_{DOGL}	Watchdog or low voltage reset output pulse width		200	300		ns

Table 62. Control timings (continued)

Symbol	Parameter	Conditions	Value			Unit
			Min	Typ	Max	
t_{DOG}	Watchdog time-out	f_{CPU} = 8 MHz	49152 6.144		3145728 393.216	t_{CPU} ms
t_{OXOV}	Crystal oscillator start-up time		20(1)	30	40(1)	ms
t_{DDR}	Power up rise time	from V_{DD} = 0 to 4 V			100(1)	ms

1. Not tested in production, guaranteed by characterization.

16.5.3 External clock source

Table 63. External clock characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSCINH}	OSCIN input pin high level voltage	see *Figure 48*	$0.7xV_{DD}$		V_{DD}	V
V_{OSCINL}	OSCIN input pin low level voltage		V_{SS}		$0.3xV_{DD}$	
$t_{w(OSCINH)}$ $t_{w(OSCINL)}$	OSCIN high or low time (1)		15			ns
$t_{r(OSCIN)}$ $t_{f(OSCIN)}$	OSCIN rise or fall time(1)				15	
I_L	OSCx Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$			±1	µA

1. Data based on design simulation and/or technology characteristics, not tested in production.

Figure 48. Typical application with an external clock source

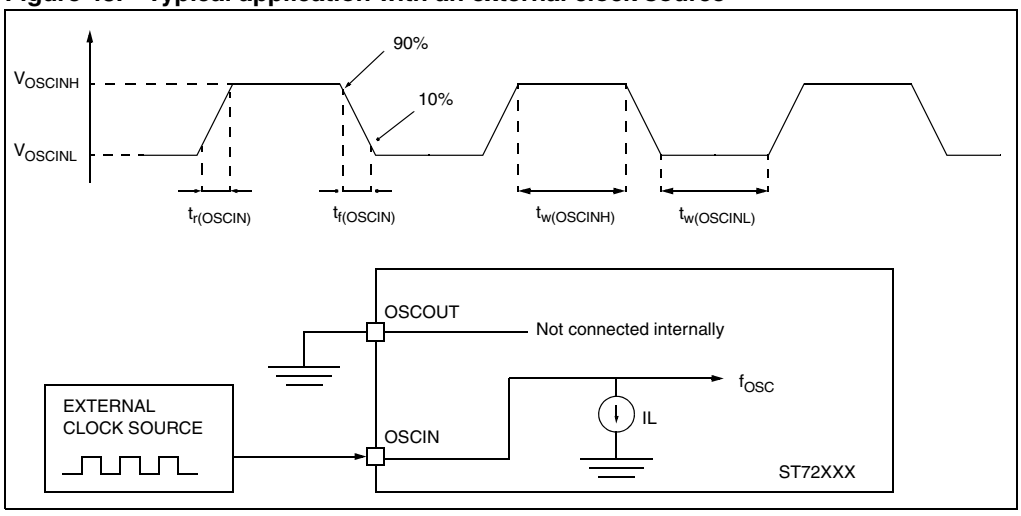
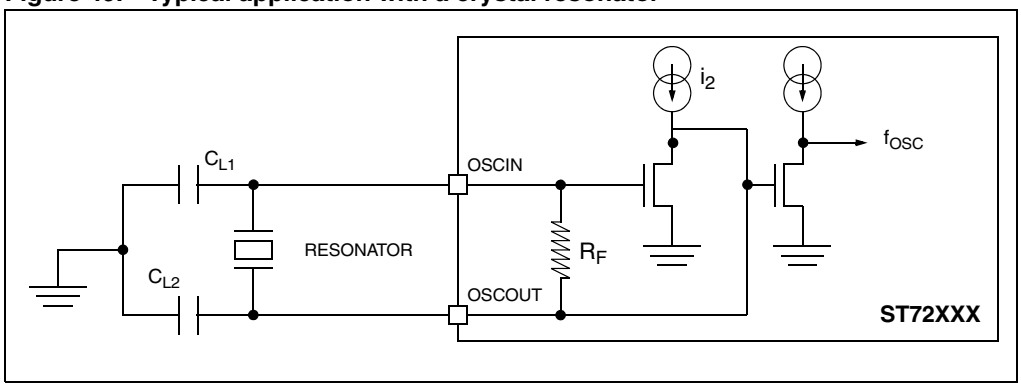


Figure 49. Typical application with a crystal resonator



16.6 Memory characteristics

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

16.6.1 RAM and hardware registers

Table 64. RAM and hardware registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode (1)	HALT mode (or RESET)	2.0			V

1. Guaranteed by design. Not tested in production.

16.6.2 Flash memory

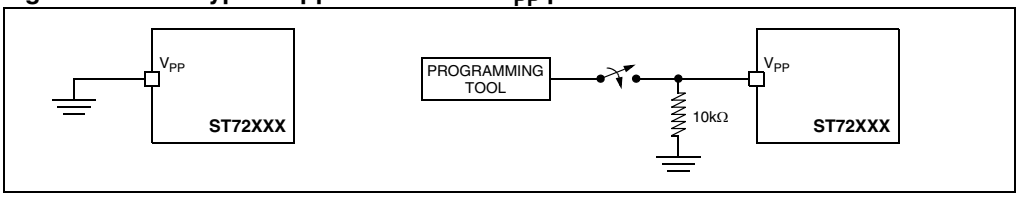
Operating Conditions: f_{CPU} = 8 MHz.

Table 65. Dual voltage Flash memory (1)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CPU}	Operating Frequency	Read mode			8	MHz
		Write / Erase mode, T_A=25°C			8	
V_{PP}	Programming Voltage	4.0V ≤V_{DD} ≤ 5.5V	11.4		12.6	V
I_{PP}	V_{PP} Current	Write / Erase		30		mA
t_{VPP}	Internal V_{PP} Stabilization Time			10		µs
t_{RET}	Data Retention	T_A ≤ 55°C	40			years
N_{RW}	Write Erase Cycles	T_A=25°C	100			cycles

1. Refer to the Flash Programming Reference Manual for the typical HDFlash programming and erase timing values.

Figure 50. Two typical applications with V_{PP} pin(b)



b. When the ICP mode is not required by the application, V_{PP} pin must be tied to V_{SS}.

16.7 EMC characteristics

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

16.7.1 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**: 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 VDD and VSS 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.

16.7.2 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 behaviour is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

Table 66. EMS data

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	4B
V_{FFTB}	Fast transient voltage burst limits to be applied through 100pF 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	4A

16.7.3 Electro magnetic interference (EMI)

Based on a simple application running on the product (toggling 2 LEDs through the I/O ports), the product is monitored in terms of emission. This emission test is in line with the norm SAE J 1752/3 which specifies the board and the loading of each pin.

Table 67. EMC data

Symbol	Parameter	Conditions	Monitored frequency band	Max vs. [f_{OSC}/f_{CPU}]	Unit
				16/8 MHz	
S_{EMI}	Peak level (1)	V_{DD}=5V, T_A=+25°C, conforming to SAE J 1752/3 **Note:** Refer to Application Note AN1709 for data on other package types.	0.1 MHz to 30 MHz	36	dBµV
			30 MHz to 130 MHz	39	
			130 MHz to 1 GHz	26	
			SAE EMI Level	3.5	-

1. Data based on characterization results, not tested in production.

16.7.4 Absolute maximum ratings (electrical sensitivity)

Based on two different tests (ESD and LU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity.

16.7.5 Electrostatic discharge (ESD)

Electrostatic discharges (a positive then a negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends on the number of supply pins in the device (3 parts*(n+1) supply pin). This test conforms to the AEC-Q100-002 standard.

Table 68. ESD absolute maximum ratings

Symbol	Ratings	Conditions	Class	Maximum value(1)	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human body model)	T_A = +25°C, conforming to AEC-Q100-002	H1C	2000	V

1. Data based on characterization results, not tested in production

Static latch-up

Two complementary static tests are required on six parts to assess the latch-up performance.

- A supply overvoltage (applied to each power supply pin) and
- A current injection (applied to each input, output and configurable I/O pin) are performed on each sample.

This test conforms to the EIA/JESD 78 IC latch-up standard.

Table 69. Electrical sensitivities

Symbol	Parameter	Conditions	Class
LU	Static latch-up class	T_A = +25 °C conforming to JESD 78	II level A

16.8 I/O port pin characteristics

16.8.1 General characteristics

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

Table 70. I/O port characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage					$0.3xV_{DD}$	V
V_{IH}	Input high level voltage			$0.7xV_{DD}$			
V_{IN}	Input voltage	True open drain I/O pins		V_{SS}		6.0	V
		Other I/O pins				V_{DD}	
V_{hys}	Schmitt trigger voltage hysteresis				400		mV
I_L	Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$				±1	µA
I_S	Static current consumption induced by each floating input pin (1)	Floating input mode			400		
R_{PU}	Weak pull-up equivalent resistor (2)	$V_{IN}=V_{SS}$	V_{DD}=5 V	50	90	120	kΩ
C_{IO}	I/O pin capacitance				5		pF
$t_{f(IO)out}$	Output high to low level fall time	C_L=50 pF Between 10% and 90%			25		ns
$t_{r(IO)out}$	Output low to high level rise time				25		
$t_{w(IT)in}$	External interrupt pulse time (3)			1			t_{CPU}

1. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example or an external pull-up or pull-down resistor (see *Figure 51*). Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values.
2. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in *Figure 52*).
3. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

Figure 51. Two typical applications with unused I/O pin

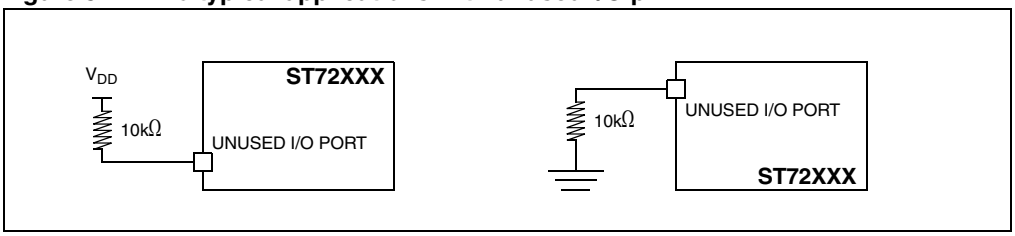


Figure 52. Typ. I_{PU} vs. V_{DD}

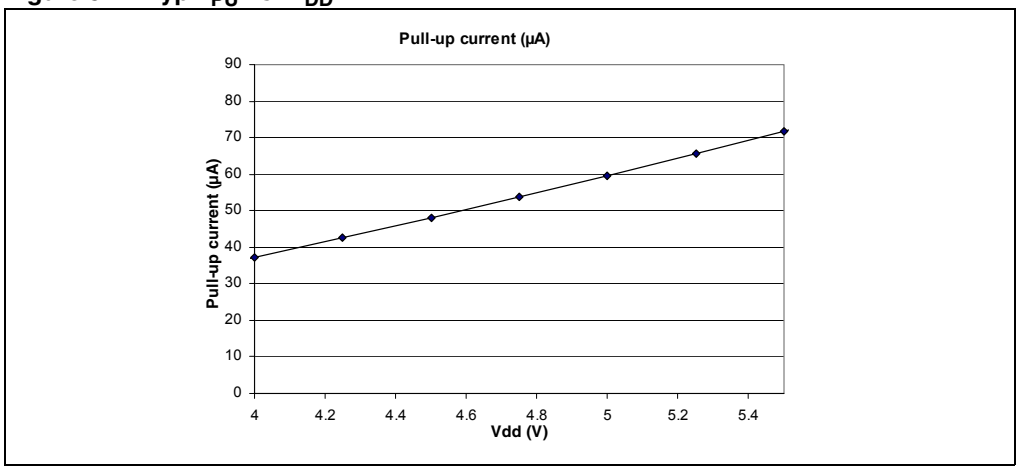
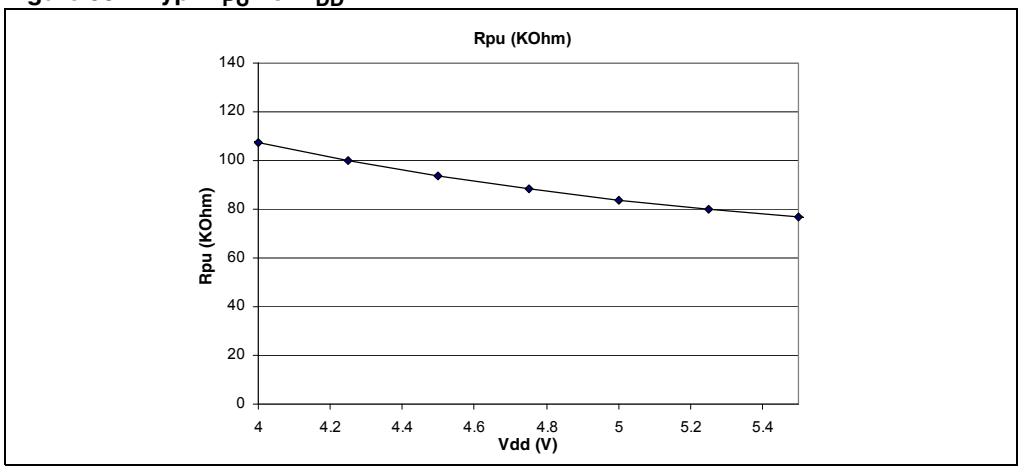


Figure 53. Typ. R_{PU} vs. V_{DD}



16.8.2 Output driving current

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

Table 71. Output current characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} (1)	Output low level voltage for a standard I/O pin when up to 8 pins are sunk at the same time, Port A0, Port A(3:7), Port C(0:2)	V_{DD}=5 V	I_{IO}=+1.6 mA		0.4	V
	Output low level voltage for a high sink I/O pin when up to 4 pins are sunk at the same time, Port B(0:7)		I_{IO}=+10 mA		1.3	
	Output low level voltage for a very high sink I/O pin when up to 2 pins are sunk at the same time, Port A1, Port A2		I_{IO}=+25 mA		1.5	
V_{OH} (2)	Output high level voltage for an I/O pin when up to 8 pins are sourced at same time		I_{IO}=-10 mA	V_{DD}-1.3		
			I_{IO}=-1.6 mA	V_{DD}-0.8		

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 16.2* 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 16.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins does not have V_{OH}.

Figure 54. V_{OL} standard V_{DD}=5 V

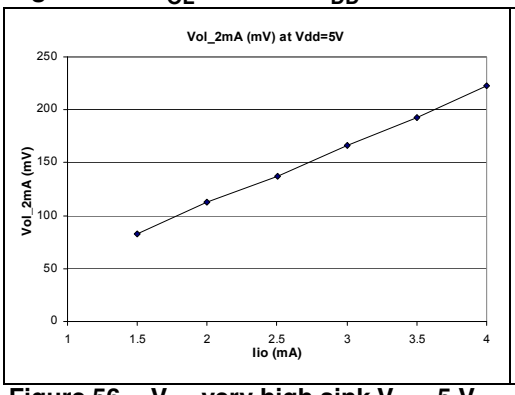


Figure 55. V_{OL} high sink V_{DD}=5 V

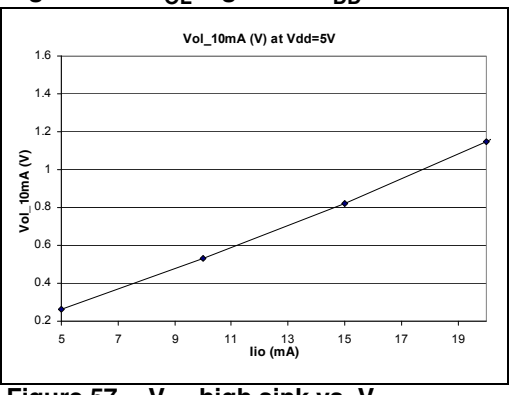


Figure 56. V_{OL} very high sink V_{DD}=5 V

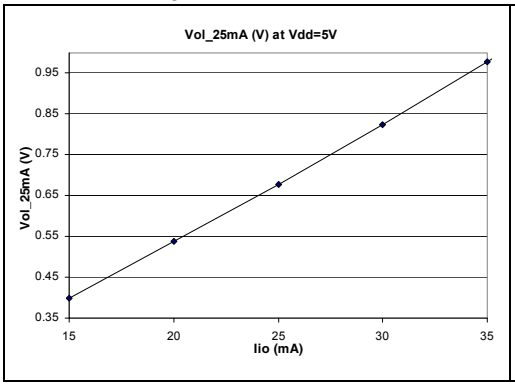


Figure 57. V_{OL} high sink vs. V_{DD}

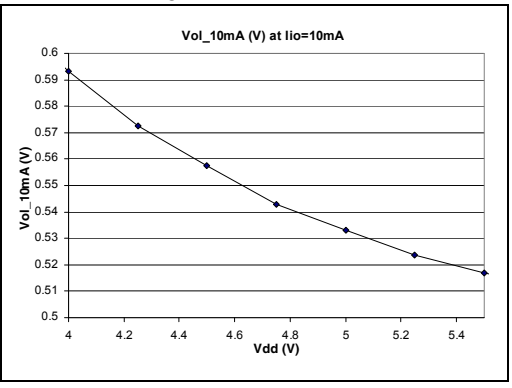


Figure 58. V_{OL} standard vs. V_{DD}

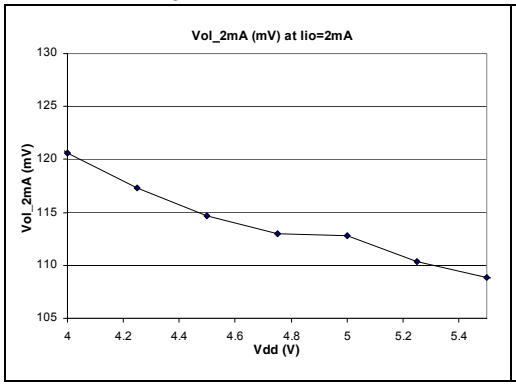


Figure 59. V_{OL} very high sink vs. V_{DD}

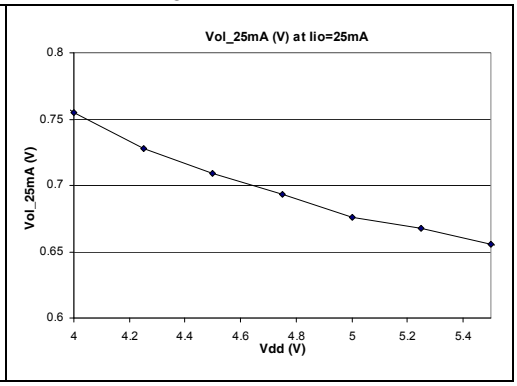


Figure 60. $|V_{DD}-V_{OH}|$ @ V_{DD}=5 V (low current)

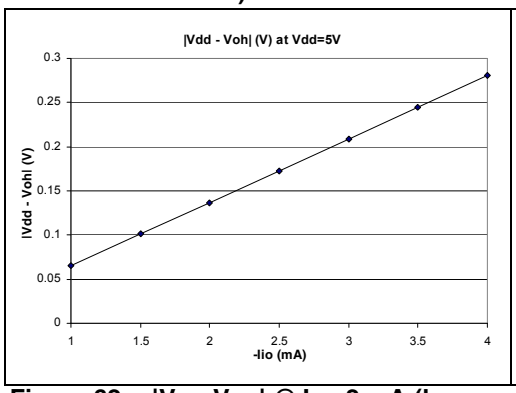


Figure 61. $|V_{DD}-V_{OH}|$ @ V_{DD}=5 V (high current)

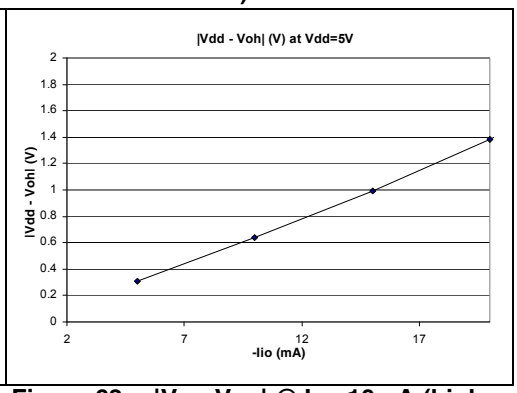


Figure 62. $|V_{DD}-V_{OH}|$ @ I_{IO}=2 mA (low current)

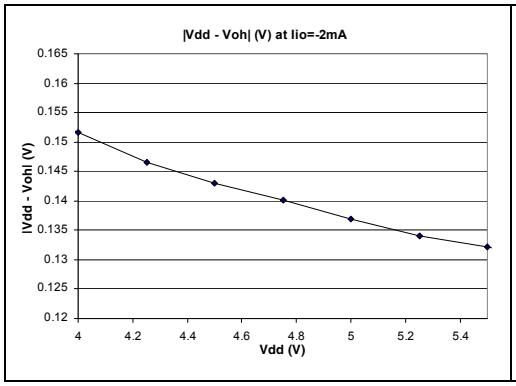
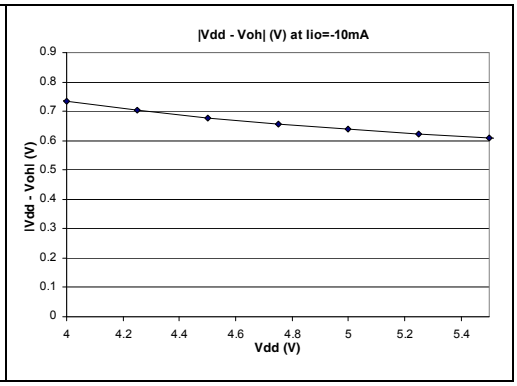


Figure 63. $|V_{DD}-V_{OH}|$ @ I_{IO}=10mA (high current)



16.9 Control pin characteristics

16.9.1 Asynchronous $\overline{\text{RESET}}$ pin

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

Table 72. RESET pin characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IH}	Input high level voltage			$0.7xV_{DD}$		V_{DD}	V
V_{IL}	Input low voltage			V_{SS}		$0.3xV_{DD}$	V
V_{hys}	Schmitt trigger voltage hysteresis (1)				400		mV
V_{OL}	Output low level voltage (2)	V_{DD}=5V	I_{IO}=5 mA			0.8	V
			I_{IO}=7.5 mA			1.3	
R_{ON}	Weak pull-up equivalent resistor (3)	V_{IN}=V_{SS}	V_{DD}=5 V	50	80	100	kΩ
$t_{w(RSTL)out}$	Generated reset pulse duration	External pin or internal reset sources			6 30		$1/f_{SFOSC}$ µs
$t_{h(RSTL)in}$	External reset pulse hold time (4)			5			µs

1. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
2. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 16.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
3. The R_{ON} pull-up equivalent resistor is based on a resistive transistor. This data is based on characterization results, not tested in production.
4. To guarantee the reset of the device, a minimum pulse has to be applied to $\overline{\text{RESET}}$ pin. All short pulses applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.

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

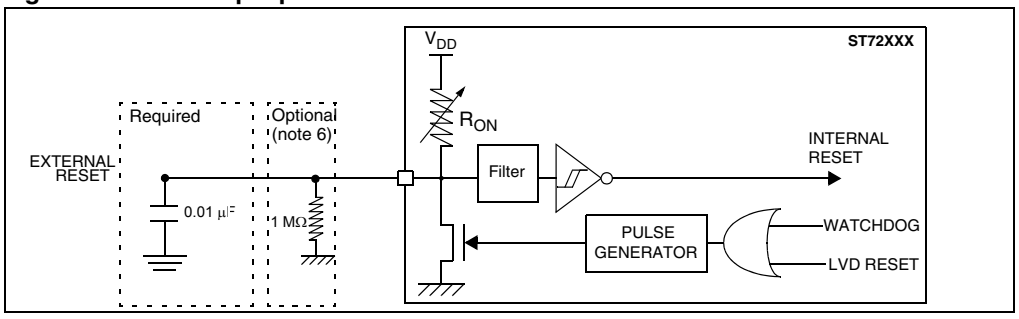
When the LVD is enabled, it is recommended to protect the $\overline{\text{RESET}}$ pin as shown in *Figure 64* and follow these guidelines:

1. The reset network protects the device against parasitic resets.
2. The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
3. Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in *Section 16.9.1*. Otherwise the reset will not be taken into account internally.
4. Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin (by an external pull-up for example) is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in *Section 16.2 on page 111*.
5. When the LVD is enabled, it is mandatory not to connect a pull-up resistor. A 10nF pull-down capacitor is recommended to filter noise on the reset line.
6. In case a capacitive power supply is used, it is recommended to connect a 1M ohm pull-down resistor to the $\overline{\text{RESET}}$ pin to discharge any residual voltage induced by this capacitive power supply (this will add 5µA to the power consumption of the MCU).

Tips when using the LVD:

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

Figure 64. $\overline{\text{RESET}}$ pin protection when LVD is enabled

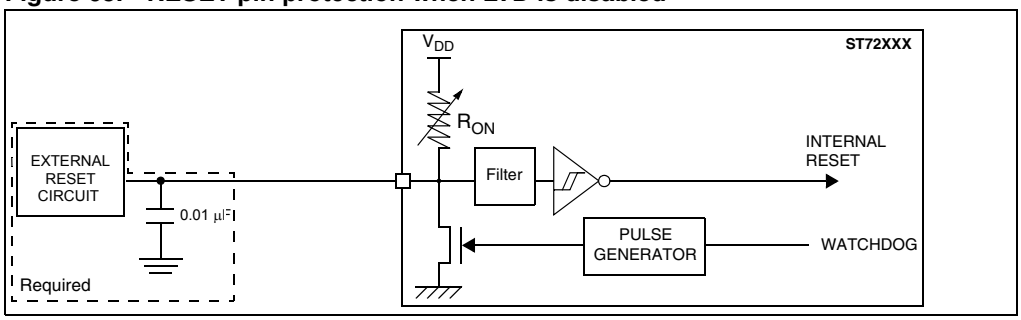


$\overline{\text{RESET}}$ pin protection when LVD is disabled

When the LVD is disabled, it is recommended to protect the $\overline{\text{RESET}}$ pin as shown in *Figure 65* and follow these guidelines:

1. The reset network protects the device against parasitic resets.
2. The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
3. Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in *Section 16.9.1*. Otherwise the reset will not be taken into account internally.
4. Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin (by an external pull-up for example) is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in *Section 16.2 on page 111*.

Figure 65. $\overline{\text{RESET}}$ pin protection when LVD is disabled



16.9.2 USB - universal bus interface

Operating conditions T_A = 0 to +70 °C, V_{DD} = 4.0 to 5.25 V unless otherwise specified.

Table 73. USB DC electrical characteristics

Symbol	Parameter	Conditions(1)	Min.	Max.	Unit
V_{DI}	Differential Input Sensitivity	I(D+, D-)	0.2		V
V_{CM}	Differential Common Mode Range	Includes VDI range	0.8	2.5	
V_{SE}	Single Ended Receiver Threshold		0.8	2.0	
V_{OL}	Static Output Low	RL(2) of 1.5 Kohms to 3.6v		0.3	
V_{OH}	Static Output High	RL (2) of 15 Kohms to V_{SS}	2.8	3.6	
USBV	USBVCC: voltage level (3)	V_{DD}=5 V	3.00	3.60	

1. All the voltages are measured from the local ground potential.
2. RL is the load connected on the USB drivers.
3. To improve EMC performance (noise immunity), it is recommended to connect a 100nF capacitor to the USBVCC pin.

Figure 66. USB: data signal rise and fall time

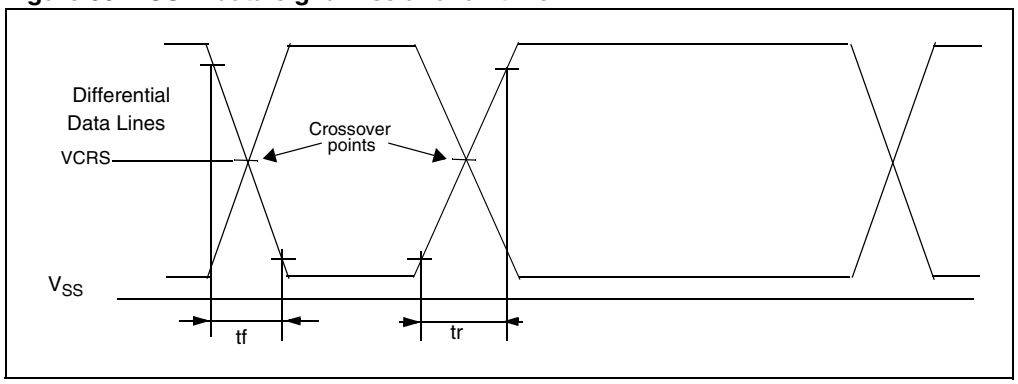


Table 74. USB: low-speed electrical characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
	Driver characteristics:				
t_r	Rise time	CL=50 pF (1)	75		ns
		CL=600 pF (1)		300	ns
t_f	Fall Time	CL=50 pF (1)	75		ns
		CL=600 pF (1)		300	ns
t_{rfm}	Rise/ Fall Time matching	tr/tf	80	120	%
V_{CRS}	Output signal Crossover Voltage		1.3	2.0	V

1. For more detailed information, please refer to Chapter 7 (Electrical) of the USB specification (version 1.1).

16.9.3 SCI - serial communications interface

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (RDI and TDO).

Table 75. SCI characteristics

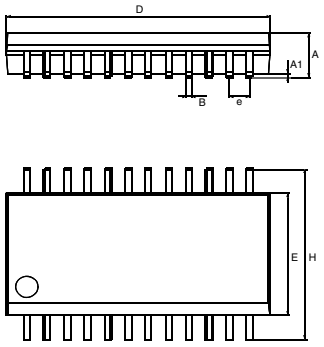
Symbol	Parameter	Conditions			Standard	Baud Rate	Unit
		f_{CPU}	Accuracy vs. Standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8 MHz	~0.16%	Conventional Mode TR (or RR)=128, PR=13 TR (or RR)= 32, PR=13 TR (or RR)= 16, PR=13 TR (or RR)= 8, PR=13 TR (or RR)= 4, PR=13 TR (or RR)= 16, PR= 3 TR (or RR)= 2, PR=13 TR (or RR)= 1, PR=13	 300 1200 2400 4800 9600 10400 19200 38400	 ~300.48 ~1201.92 ~2403.84 ~4807.69 ~9615.38 ~10416.67 ~19230.77 ~38461.54	Hz

17 Package characteristics

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: *www.st.com*. ECOPACK® is an ST trademark.

17.1 Package mechanical data

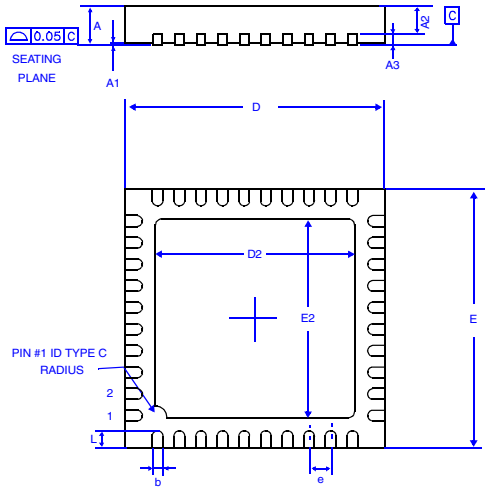
Figure 67. 24-pin plastic small outline package, 300-mil width



Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.0925		0.1043
A1	0.10		0.30	0.0039		0.0118
B	0.33		0.51	0.0130		0.0201
C	0.23		0.32	0.0091		0.0126
D	15.20		15.60	0.5984		0.6142
E	7.40		7.60	0.2913		0.2992
e		1.27			0.0500	
H	10.00		10.65	0.3937		0.4193
h	0.25		0.75	0.098		0.0295
α	0°		8°	0°		8°
L	0.40		1.27	0.0157		0.0500
	Number of Pins					
N	24					

1. Values in inches are converted from mm and rounded to 4 decimal digits.

Figure 68. 40-lead very thin fine pitch quad flat no-lead package



Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A	0.80	0.90	1.00	0.0315	0.0354	0.0394
A1		0.02	0.05		0.0008	0.0020
A2		0.65	1.00		0.0256	0.0394
A3		0.20			0.0079	
b	0.18	0.25	0.30	0.0071	0.0098	0.0118
D	5.85	6.00	6.15	0.2303	0.2362	0.2421
D2	2.75	2.9	3.05	0.1083	0.1142	0.1201
E	5.85	6	6.15	0.2303	0.2362	0.2421
E2	2.75	2.9	3.05	0.1083	0.1142	0.1201
e		0.50			0.0197	
L	0.30	0.40	0.50	0.0118	0.0157	0.0197
	Number of Pins					
N	40					

1. Values in inches are converted from mm and rounded to 4 decimal digits.

17.1.1 Thermal characteristics

Table 76. Package thermal characteristics

Symbol	Ratings	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient) SO24 QFN40	 70 34	°C/W
P_D	Power dissipation (1)	500	mW
T_{Jmax}	Maximum junction temperature (2)	150	°C

1. The maximum power dissipation is obtained from the formula $P_D = (T_J - T_A) / R_{thJA}$. The power dissipation of an application can be defined by the user with the formula: $P_D = PI_{NT} + P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD} x V_{DD}$) and P_{PORT} is the port power dissipation depending on the ports used in the application.
2. The maximum chip-junction temperature is based on technology characteristics.

18 Device configuration and ordering information

Each device is available for production in user programmable versions (High Density FLASH) as well as in factory coded versions (FASTROM).

ST72P60 devices are Factory Advanced Service Technique ROM (FASTROM) versions: they are factory programmed FLASH devices.

ST72F60 FLASH devices are shipped to customers with a default content (FFh).

This implies that FLASH devices have to be configured by the customer using the Option Byte while the FASTROM devices are factory-configured.

18.1 Option byte

The Option Byte allows the hardware configuration of the microcontroller to be selected.

The Option Byte has no address in the memory map and can be accessed only in programming mode using a standard ST7 programming tool. The default contents of the FLASH is fixed to F7h. This means that all the options have “1” as their default value, except LVD.

Table 77. Flash option byte

7							0
--	--	WDG SW	WD HALT	LVD	--	OSC 24/12	FMP_R

OPT 7:6 = Reserved.

OPT 5 = **WDGSW** *Hardware or Software Watchdog*
This option bit selects the watchdog type.
0: Hardware enabled
1: Software enabled

OPT 4 = **WDHALT** *Watchdog and HALT mode*
This option bit determines if a RESET is generated when entering HALT mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

OPT 3 = **LVD** *Low Voltage Detector selection*
This option bit selects the LVD.
0: LVD enabled
1: LVD disabled

OPT 2 = Reserved.

OPT 1 = **OSC24/12** *Oscillator Selection*
This option bit selects the clock divider used to drive the USB interface at 6 MHz.
0: 24 MHz oscillator
1: 12 Mhz oscillator

OPT 0 = **FMP_R** Flash memory readout protection
This option indicates if the user flash memory is protected against readout. readout protection, when selected, provides a protection against Program Memory content

extraction and against write access to Flash memory. Erasing the option bytes when the FMP_R option is selected, causes the whole user memory to be erased first and the device can be reprogrammed. Refer to the ST7 Flash Programming Reference Manual and *Section 5.3.1 on page 18* for more details.
0: Readout protection enabled
1: Readout protection disabled

18.2 Device ordering information and transfer of customer code

Customer code is made up of the FASTROM contents and the list of the selected options (if any). The FASTROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file in .S19 format generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended. See page 110.

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

The STMicroelectronics sales organization will be pleased to provide detailed information on contractual points.

Table 78. Supported part numbers

Sales type (1)	Program memory (bytes)	RAM (bytes)	Package
ST72F60K2U1	8 K Flash	384	QFN40
ST72F60E2M1		384	SO24
ST72F60K1U1	4 K Flash	384	QFN40
ST72F60E1M1		384	SO24
ST72P60K2U1/xxx	8 K FASTROM	384	QFN40
ST72P60E2M1/xxx		384	SO24
ST72P60K1U1/xxx	4 K FASTROM	384	QFN40
ST72P60E1M1/xxx		384	SO24

1. /xxx stands for the ROM code name assigned by STMicroelectronics. Contact ST sales office for product availablity of FASTROM types (shaded in table).

18.3 Development tools

STMicroelectronics offers a range of hardware and software development tools for the ST7 microcontroller family. Full details of tools available for the ST7 from third party manufacturers can be obtain from the STMicroelectronics Internet site:
➟ http//www.st.com.

Tools from these manufacturers include C compliers, emulators and gang programmers.

STMicroelectronics Tools

Three types of development tool are offered by ST see *Table 79* and *Table 80* for more details.

Table 79. STMicroelectronics tools features

	In-Circuit Emulation	Programming capability(1)	Software Included
ST7 Emulator	Yes, powerful emulation features including trace/ logic analyzer	No	ST7 CD ROM with: ST7 Assembly toolchain STVD7 powerful Source Level Debugger for Win 3.1, Win 9x and NT C compiler demo versions Windows Programming Tools for Win 3.1, Win 9x and NT
ST7 Programming Board	No	Yes (All packages)	

1. In-Circuit Programming (ICP) interface for FLASH devices.

Table 80. Dedicated STMicroelectronics development tools

Supported products	Evaluation board	ST7 emulator	ST7 programming board
ST7260	ST7MDTULS-EVAL	ST7MDTU3-EMU3	ST7MDTU3-EPB (1)

1. Add Suffix /EU or /US for the power supply for your region.

ST7260 MICROCONTROLLER OPTION LIST
(Last update: January 2009)

Customer: . . . . . .
Address: . . . . . .
. . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference: . . . . . .

FASTROM code must be sent in .S19 format.
Hex extension cannot be processed.
STMicroelectronics references:

Device Type/Memory Size/Package (check only one option):

FASTROM:	4K	8K
SO24:	[ ] ST72P60E1M1	[ ] ST72P60E2M1
QFN40:	[ ] ST72P60K1U1	[ ] ST72P60K2U1

Conditioning (check only one option):
Packaged Product

[ ] Tape & Reel (SO package only)
[ ] Tube

Special Marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ _ _ _"
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
For marking, one line is possible with a maximum of 13 characters.

Watchdog Selection: [ ] Software activation [ ] Hardware activation
Halt when Watchdog on: [ ] Reset [ ] No reset
LVD Reset * [ ] Disabled* [ ] Enabled*

Oscillator Selection: [ ] 24 MHz. [ ] 12 MHz.
Readout Protection: [ ] Disabled [ ] Enabled

Date . . . . . .

Signature . . . . . .
Please download the latest version of this option list from:
http://www.st.com/

19 Known limitations

19.1 PA2 limitation with OCMP1 enabled

Description

This limitation affects only Rev B Flash devices (with Internal Sales Type 72F60xxxxx$x7); it has been corrected in Rev W Flash devices (with Internal Sales Type 72F60xxxxx$x9).

Refer to *Figure 69 on page 137*.

When Output Compare 1 function (OCMP1) on pin PA6 is enabled by setting the OC1E bit in the TCR2 register, pin PA2 is also affected.

In particular, the PA2 pin is forced to be floating even if port configuration (PADDR+PADR) has set it as output low. However, it can be still used as an input.

19.2 Unexpected reset fetch

If an interrupt request occurs while a "POP CC" instruction is executed, the interrupt controller does not recognise 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.

19.3 SCI wrong break duration

Description

A single break character is sent by setting and resetting the SBK bit in the SCICR2 register. In some cases, the break character may have a longer duration than expected:

- 20 bits instead of 10 bits if M=0
- 22 bits instead of 11 bits if M=1

In the same way, as long as the SBK bit is set, break characters are sent to the TDO pin. This may lead to generate one break more than expected.

Occurrence

The occurrence of the problem is random and proportional to the baudrate. With a transmit frequency of 19200 baud (f_{CPU}=8 MHz and SCIBRR=0xC9), the wrong break duration occurrence is around 1%.

Workaround

If this wrong duration is not compliant with the communication protocol in the application, software can request that an Idle line be generated before the break character. In this case, the break duration is always correct assuming the application is not doing anything between the idle and the break. This can be ensured by temporarily disabling interrupts.

The exact sequence is:

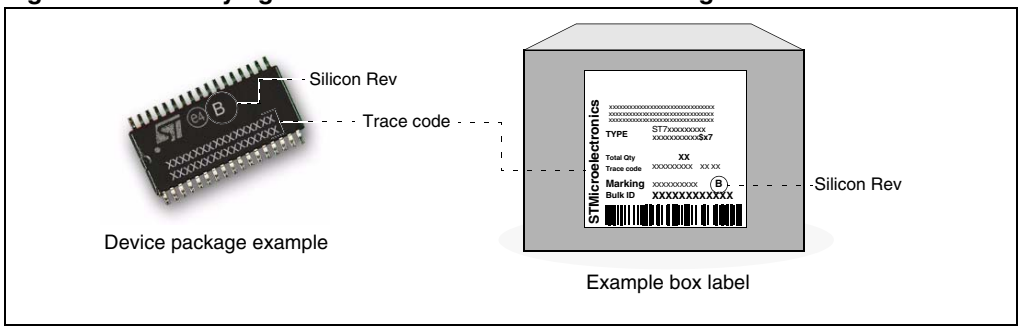
- Disable interrupts
- Reset and Set TE (IDLE request)
- Set and Reset SBK (Break Request)
- Re-enable interrupts

20 Device marking

The silicon revision can be identified either by Rev letter or obtained via a trace code. Follow the procedure below:

1. Identify the silicon revision letter from either the device package or the box label. For example, “**B**”, etc. Refer to *Figure 69*.
2. If the revision letter is not present, obtain the silicon revision by contacting your local ST office with the trace code information printed on either the box label or the device package.

Figure 69. Identifying silicon revision from device marking and box label



21 Revision history

Table 81. Document revision history

Date	Revision	Changes
13-Feb-2006	1	Initial release.
18-Oct-2006	2	Added known limitations section
05-Feb-2009	3	Added caution in *Section 7.1 on page 25* Added reference to watchdog reset pulse t_{DOG} in *Section 12.3 on page 44* Removed EMC protective circuitry in *Figure 65 on page 127* (device works correctly without these components) Modified notes below *Table 76: Package thermal characteristics on page 130* Replaced soldering information with ECOPACK reference in *Section 17 on page 129*

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com