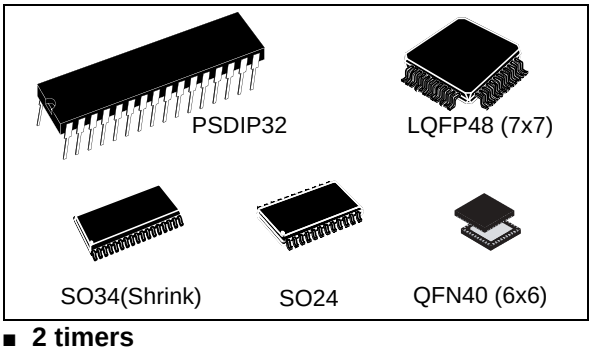
ST7263BDx ST7263BHx ST7263BKx ST7263BE

Low speed USB 8-bit MCU family with up to 32K Flash/ROM, DFU capability, 8-bit ADC, WDG, timer, SCI & I²C

- **Memories**
  - 4, 8, 16 or 32 Kbytes program memory: high density Flash (HDFlash), FastROM or ROM with Read-Out and Write protection
  - In-application programming (IAP) and in-circuit programming (ICP)
  - 384, 512 or 1024 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 Input/Output configuration
- **Up to 27 I/O ports**
  - Up to 8 high sink I/Os (10 mA at 1.3 V)
  - 2 very high sink true open drain I/Os (25 mA at 1.5 V)
  - Up to 8 lines individually programmable as interrupt inputs
- **1 analog peripheral**
  - 8-bit A/D converter with 8 or 12 channels



- **2 timers**
  - Programmable watchdog
  - 16-bit timer with 2 Input Captures, 2 Output Compares, PWM output and clock input
- **2 communication Interfaces**
  - Asynchronous serial communications Interface
  - I²C multimaster Interface up to 400 kHz
- **Instruction Set**
  - 63 basic instructions
  - 17 main addressing modes
  - 8 x 8 unsigned multiply instruction
  - True bit manipulation
- **Development tools**
  - Versatile development tools (under Windows) including assembler, linker, C-compiler, archiver, source level debugger, software library, hardware emulator, programming boards and gang programmers, HID and DFU software layers

Table 1. Device summary

Features	ST7263BH2, ST7263BH6			ST7263BD6	ST7263BK1, ST7263BK2, ST7263BK4, ST7263BK6				ST7263BE1, ST7263BE2, ST7263BE4, ST7263BE6			
Program memory (Flash / ROM) - bytes	32K	16K	8K	32K	32K	16K	8K	4K	32K	16K	8K	4K
RAM (stack) - bytes	1024 (128)	512 (128)	384 (128)	1024 (128)	1024 (128)	512 (128)	384 (128)	384 (128)	1024 (128)	512 (128)	384 (128)	384 (128)
Standard peripherals	Watchdog timer, 16-bit timer, USB											
Other peripherals	SCI, I²C, ADC				SCI, ADC			ADC	SCI, I²C			
I/Os (high current)	27 (10)				19 (10)				14 (6)			
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 temp.	0 °C to +70 °C											
Packages	LQFP48 (7x7)			QFN40 (6x6)	SDIP32/ SO34		QFN40 (6x6)	SDIP32/SO34	SO24			

Rev. 7.0

Table of Contents

1 Introduction 6
2 Pin description 7
3 Register & memory map 14
4 Flash program memory 17
4.1 Introduction 17
4.2 Structure 17
4.2.1 Read-Out protection 17
4.3 ICC interface 19
4.4 ICP (in-circuit programming) 20
4.5 IAP (in-application programming) 20
4.6 Related documentation 20
4.7 Register description 20
5 Central processing unit 21
5.1 Introduction 21
5.2 Features 21
5.3 CPU registers 21
6 CPU registers (Cont'd) 23
7 Reset and clock management 24
7.1 Reset 24
7.1.1 Low voltage detector (LVD) 24
7.1.2 Watchdog reset 24
7.1.3 External reset 24
7.2 Clock system 26
7.2.1 General description 26
7.2.2 External clock 26
8 Interrupts 27
8.1 Interrupt register 29
9 Power saving modes 30
9.1 Introduction 30
9.2 Halt mode 30
9.3 Slow mode 31
9.4 Wait mode 31
10 I/O ports 32
10.1 Introduction 32
10.2 Functional description 32
10.3 I/O port implementation 33
10.3.1 Port A 34
10.3.2 Port B 36
10.3.3 Port C 38
10.3.4 Port D 39
10.3.5 Register description 40
11 Miscellaneous register 41
12 On-chip peripherals 42
12.1 Watchdog timer (WDG) 42

12.1.1 Introduction . . . 42
12.1.2 Main features . . . 42
12.1.3 Functional description . . . 42
12.1.4 Software Watchdog option . . . 43
12.1.5 Hardware Watchdog option . . . 43
12.1.6 Low power modes . . . 43
12.1.7 Interrupts . . . 43
12.1.8 Register description . . . 44
12.2 16-bit timer . . . 45
12.2.1 Introduction . . . 45
12.2.2 Main features . . . 45
12.2.3 Functional description . . . 45
12.2.4 Low power modes . . . 57
12.2.5 Interrupts . . . 57
12.2.6 Summary of timer modes . . . 57
12.2.7 Register description . . . 58
12.3 Serial communications interface (SCI) . . . 64
12.3.1 Introduction . . . 64
12.3.2 Main features . . . 64
12.3.3 General description . . . 64
12.3.4 Functional description . . . 66
12.3.5 Low power modes . . . 72
12.3.6 Interrupts . . . 72
12.3.7 Register description . . . 73
12.4 USB interface (USB) . . . 78
12.4.1 Introduction . . . 78
12.4.2 Main features . . . 78
12.4.3 Functional description . . . 78
12.4.4 Register description . . . 79
12.4.5 Programming considerations . . . 84
12.5 I²C bus interface (I²C) . . . 86
12.5.1 Introduction . . . 86
12.5.2 Main features . . . 86
12.5.3 General description . . . 86
12.5.4 Functional description . . . 88
12.5.5 Low power modes . . . 92
12.5.6 Interrupts . . . 92
12.5.7 Register description . . . 93
12.6 8-bit A/D converter (ADC) . . . 98
12.6.1 Introduction . . . 98
12.6.2 Main features . . . 98
12.6.3 Functional description . . . 98
12.6.4 Low power modes . . . 99
12.6.5 Interrupts . . . 99
12.6.6 Register description . . . 100
13 Instruction set . . . 102
13.1 ST7 addressing modes . . . 102
13.1.1 Inherent . . . 103

13.1.2 Immediate . . . 103
13.1.3 Direct . . . 103
13.1.4 Indexed (No Offset, Short, Long) . . . 103
13.1.5 Indirect (Short, Long) . . . 103
13.1.6 Indirect Indexed (Short, Long) . . . 104
13.1.7 Relative mode (Direct, Indirect) . . . 104
13.2 insTruction groups . . . 105
14 Electrical characteristics . . . 108
14.1 Parameter conditions . . . 108
14.1.1 Minimum and maximum values . . . 108
14.1.2 Typical values . . . 108
14.1.3 Typical curves . . . 108
14.1.4 Loading capacitor . . . 108
14.1.5 Pin input voltage . . . 108
14.2 Absolute maximum ratings . . . 109
14.2.1 Voltage characteristics . . . 109
14.2.2 Current characteristics . . . 109
14.2.3 Thermal characteristics . . . 109
14.3 Operating conditions . . . 110
14.3.1 General operating conditions . . . 110
14.3.2 Operating conditions with Low Voltage Detector (LVD) . . . 110
14.4 Supply current characteristics . . . 111
14.5 Clock and timing characteristics . . . 112
14.5.1 General timings . . . 112
14.5.2 Control timing characteristics . . . 112
14.5.3 External clock source . . . 113
14.6 Memory characteristics . . . 114
14.6.1 RAM and hardware registers . . . 114
14.6.2 Flash memory . . . 114
14.7 EMC characteristics . . . 115
14.7.1 Functional EMS (Electromagnetic susceptibility) . . . 115
14.7.2 Electromagnetic interference (EMI) . . . 115
14.7.3 Absolute maximum ratings (electrical sensitivity) . . . 116
14.8 I/O port pin characteristics . . . 117
14.8.1 General characteristics . . . 117
14.8.2 Output driving current . . . 118
14.9 Control pin characteristics . . . 121
14.9.1 Asynchronous RESET pin . . . 121
14.10Communication interface characteristics . . . 123
14.10.1 USB - universal bus interface . . . 123
14.10.2 SCI - serial communication interface . . . 124
14.10.3 I2C - Inter IC control interface . . . 124
14.118-bit ADC characteristics . . . 126
15 Package characteristics . . . 128
15.1 Package mechanical data . . . 128
15.2 Thermal characteristics . . . 131

Table of Contents

15.3 Soldering and glueability information 132
16 Device configuration and ordering information 133
16.1 Option Byte 133
16.2 Device ordering information and transfer of customer code 134
16.3 Development tools 137
16.3.1 Evaluation tools and starter kits 137
16.3.2 Development and debugging tools 137
16.3.3 Programming tools 137
16.3.4 Order codes for ST7263B development tools 137
16.4 ST7 application notes 139
17 Known limitations 142
17.1 PA2 limitation with OCMP1 enabled 142
17.2 Unexpected reset fetch 142
17.3 USB behavior with LVD disabled 142
17.4 I2C multimaster 142
17.5 Halt mode power consumption with ADC on 142
17.6 SCI wrong break duration 142
18 Revision history 144

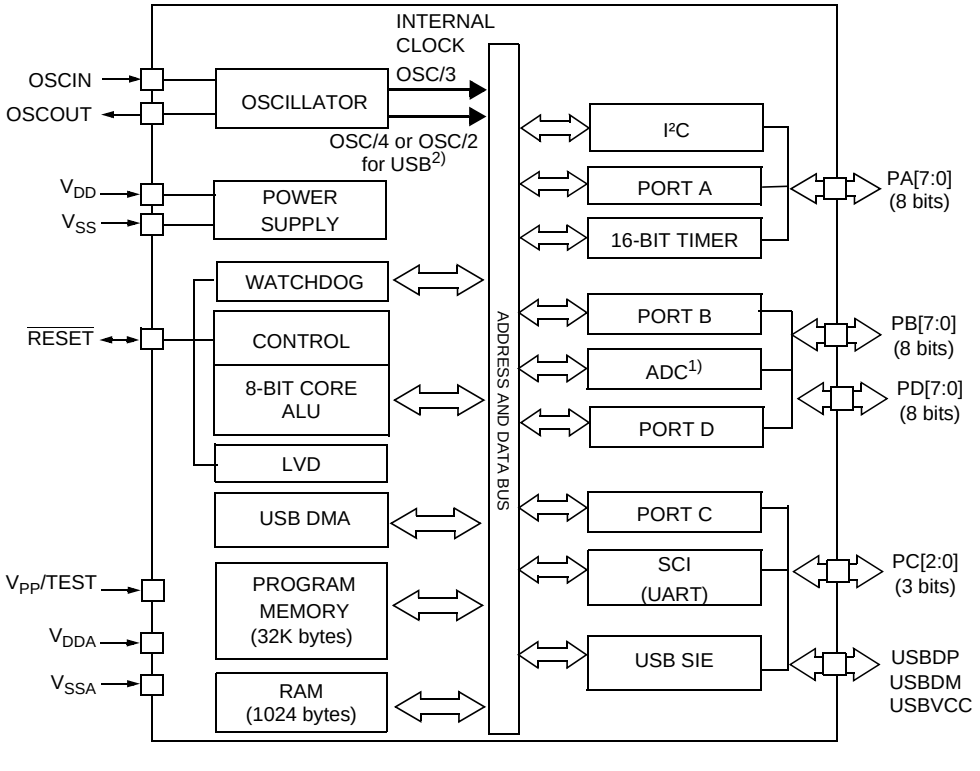
1 Introduction

The ST7263B microcontrollers form a sub-family of the ST7 MCUs dedicated to USB applications. The devices are based on an industry-standard 8-bit core and feature an enhanced instruction set. They operate at a 24 MHz or 12 MHz oscillator frequency. Under software control, the ST7263B MCUs may be placed in either Wait or Halt modes, thus reducing power consumption. The enhanced instruction set and addressing modes afford real programming potential. In addition to standard 8-bit data management, the ST7263B MCUs feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes. The devices include an ST7 Core, up to 32Kbytes of program memory, up to 1024 bytes of RAM, 27 I/O lines and the following on-chip peripherals:

– USB low speed interface with 3 endpoints with programmable in/out configuration using the DMA architecture with embedded 3.3V voltage regulator and transceivers (no external components are needed).
– 8-bit analog-to-digital converter (ADC) with 12 multiplexed analog inputs
– Industry standard asynchronous SCI serial interface
– Watchdog
– 16-bit timer featuring an external clock input, 2 Input Captures, 2 Output Compares with Pulse Generator capabilities
– Fast I²C multimaster interface
– Low voltage reset (LVD) ensuring proper power-on or power-off of the device

The ST72F63B devices are Flash versions. They support programming in IAP mode (in-application programming) via the on-chip USB interface.

Figure 1. General block diagram



1) ADC channels:
12 on 48-pin devices (Port B and Port D[3:0])
8 on 34 and 32-pin devices (Port B)
None on 24-pin devices

2) 12 or 24 MHz OSCIN frequency required to generate 6 MHz USB clock.

3) The drive from USBVCC is sufficient to only drive an external pull-up in additoin to the internal transceiver

2 Pin description

Figure 2. 48-pin LQFP package pinout

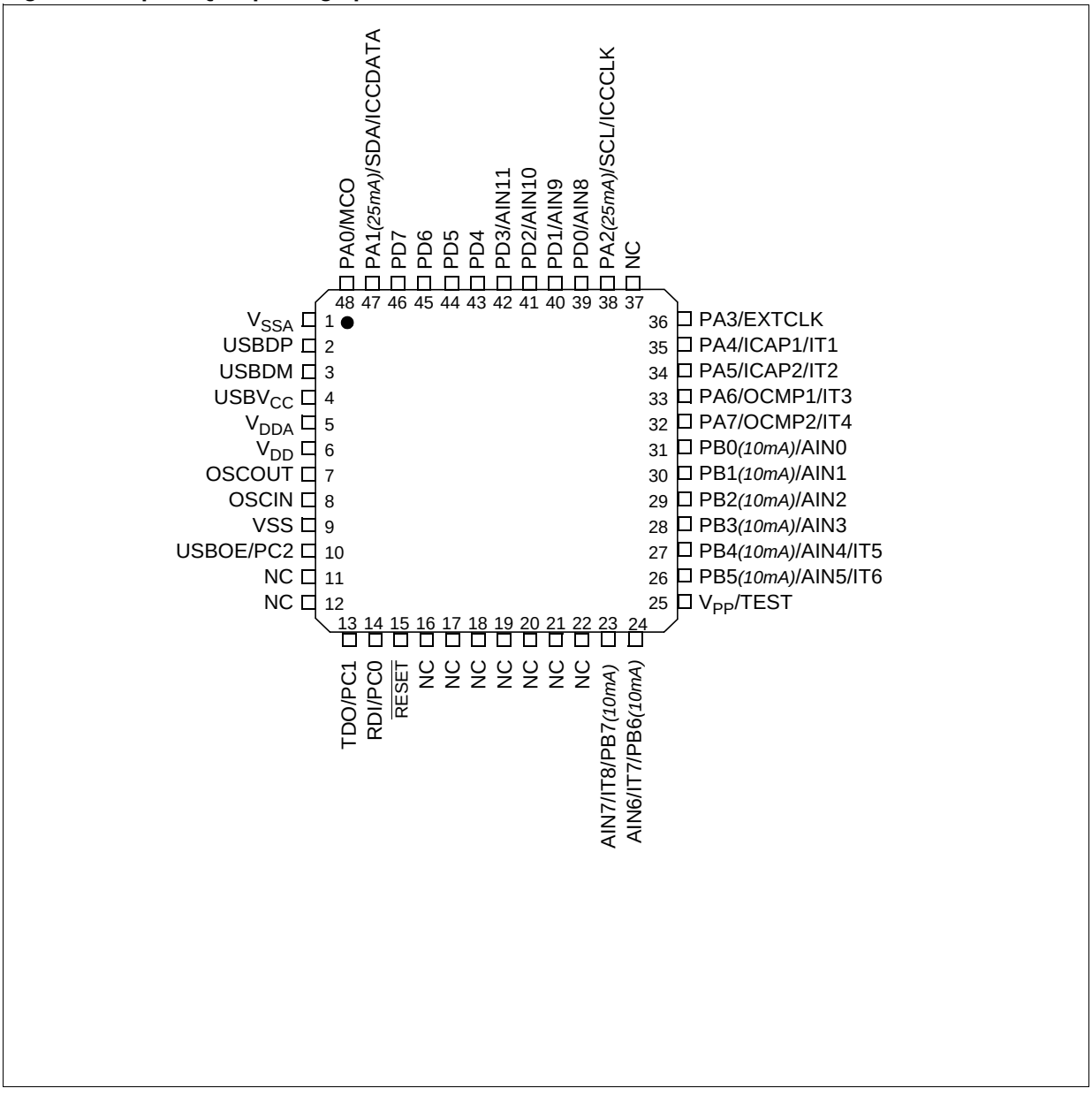
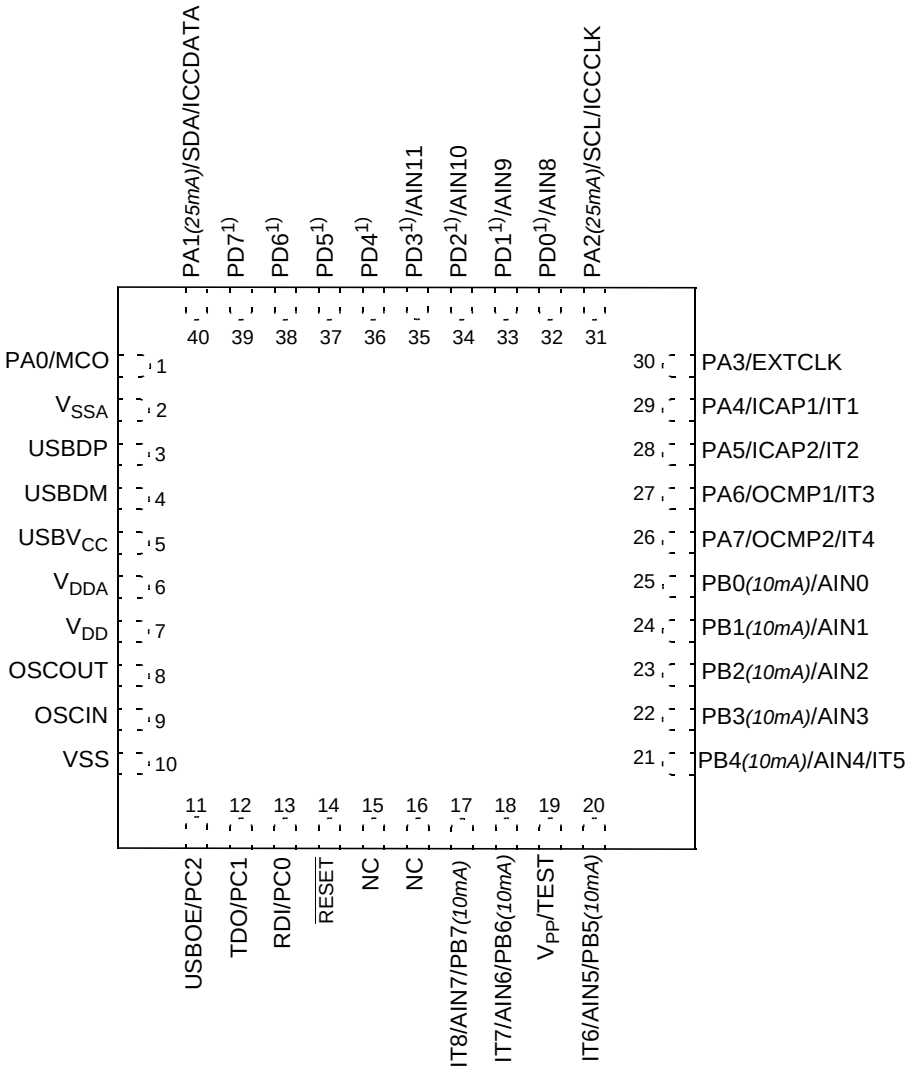


Figure 3. 40-lead QFN package pinout



Note:
1. Port D functions are not available on the 8K version of the QFN40 package (ST7263BK2) and should not be connected.

Pin description (Cont'd)

Figure 4. 34-pin SO package pinout

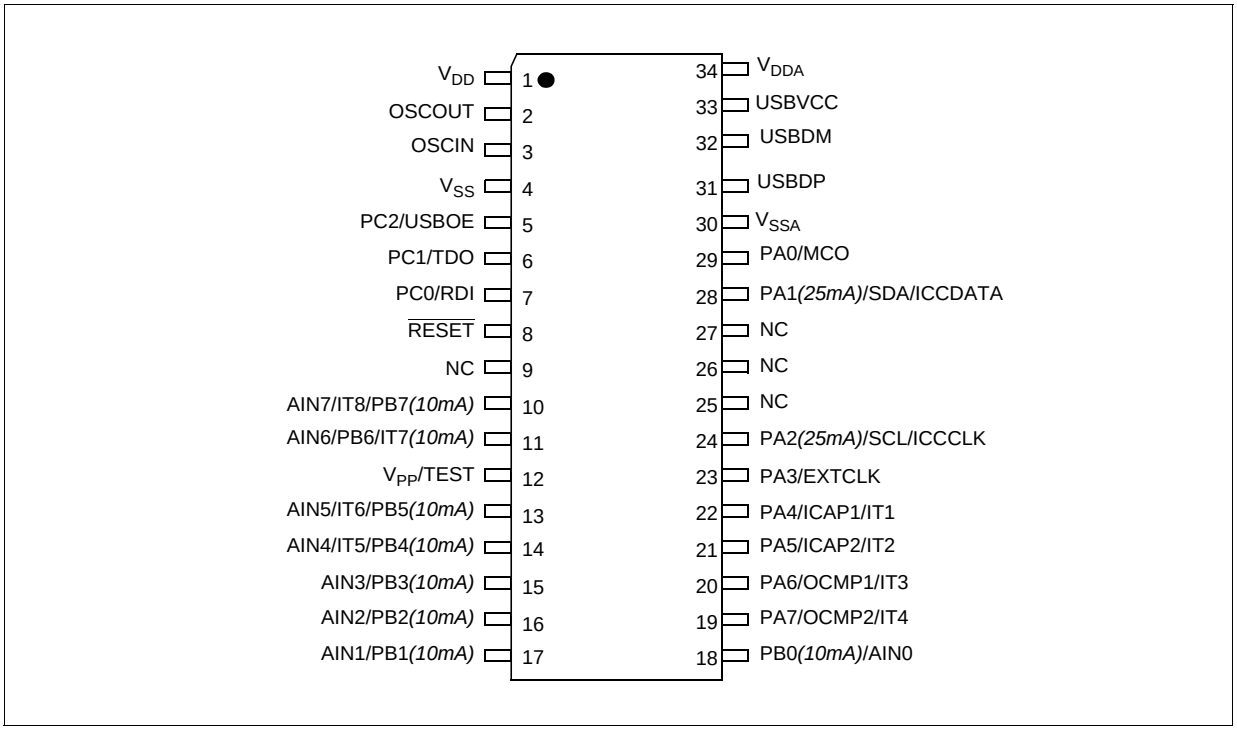


Figure 5. 32-pin SDIP package pinout

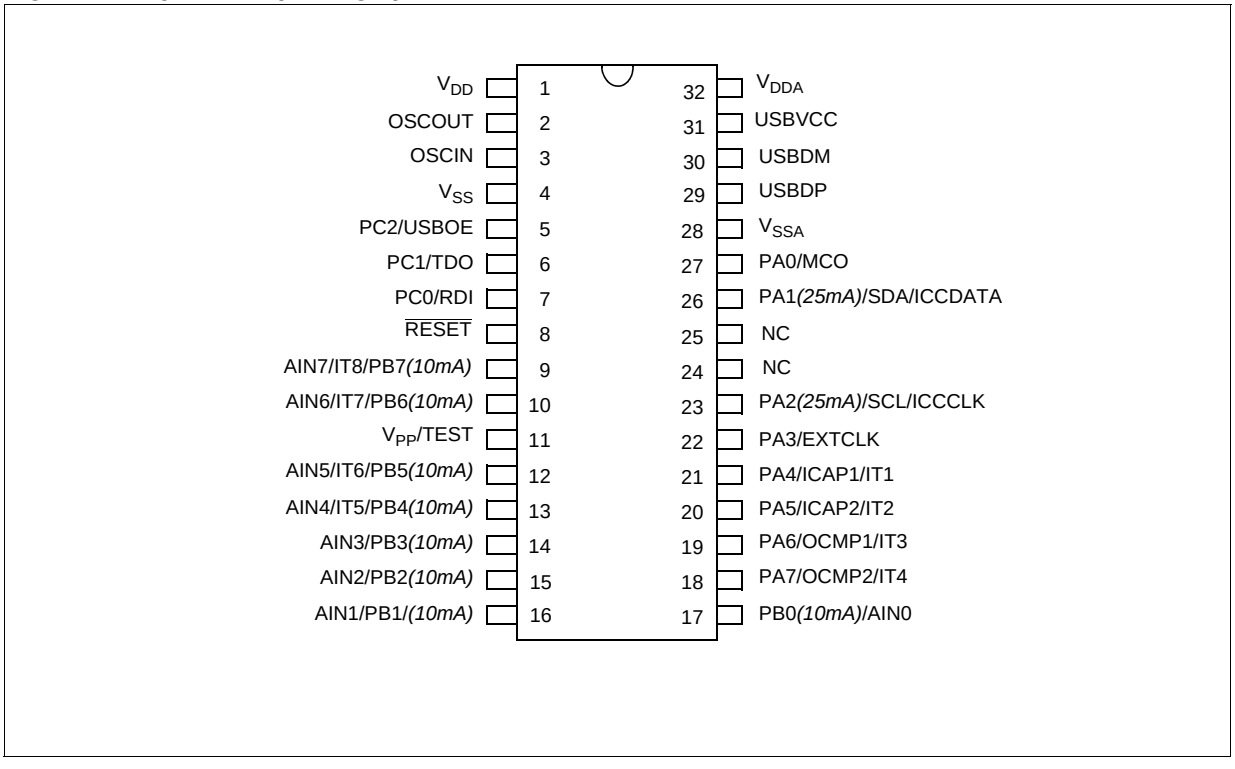
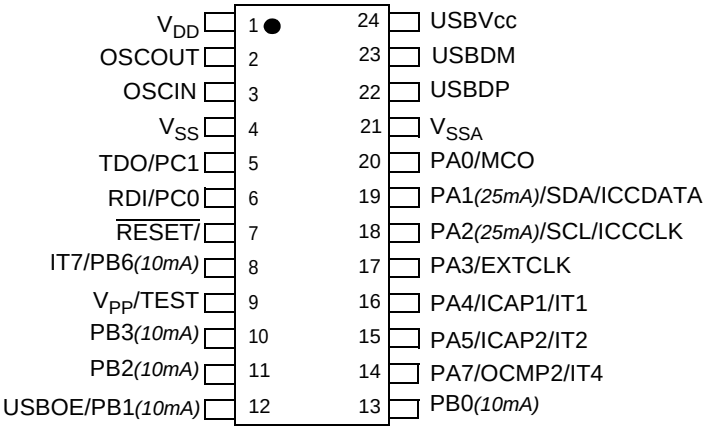


Figure 6. 24-pin SO package pinout



$\overline{\text{RESET}}$ (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: Adding two 100 nF decoupling capacitors on the Reset pin (respectively connected to V_{DD} and V_{SS}) will significantly improve product electromagnetic susceptibility performance.

Note 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}.

Note 3: The USBOE alternate function is mapped on port C2 in 32/34/48 pin devices. In SO24 devices it is mapped on Port B1.

Note 4: The timer OCMP1 alternate function is mapped on Port A6 in 32/34/48 pin devices. In SO24 devices it is not available.

Legend / Abbreviations for Table 2 and Table 3:

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

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

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

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

Port and control configuration:

– Input: float = floating, wpu = weak pull-up, int = interrupt, ana = analog

– 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, LQFP48, SO34 and SDIP32)

Pin n°				Pin name	Type	Level		Port / Control						Main function (after reset)	Alternate function
								Input				Output			
SDIP32	SO34	QFN40	LQFP48			Input	Output	float	wpu	int	ana	OD	PP		
1	1	7	6	V_{DD}	S									Power supply voltage (4V - 5.5V)	
2	2	8	7	OSCOUT	O									Oscillator output	
3	3	9	8	OSCIN	I									Oscillator input	
4	4	10	9	V_{SS}	S									Digital ground	
5	5	11	10	PC2/USBOE	I/O	C_T			**X**				X	**Port C2**	USB Output Enable
6	6	12	13	PC1/TDO	I/O	C_T			**X**				X	**Port C1**	SCI Transmit Data Output
7	7	13	14	PC0/RDI	I/O	C_T			**X**				X	**Port C0**	SCI Receive Data Input
8	8	14	15	$\overline{\text{RESET}}$	I/O				**X**			X		Reset	
-	9	15	16	NC	--									Not connected	
-	-	16	17	NC	--									Not connected	
-	-	-	18	NC	--									Not connected	
-	-	-	19	NC	--									Not connected	
-	-	-	20	NC	--									Not connected	
-	-	-	21	NC	--									Not connected	
-	-	-	22	NC	--									Not connected	
9	10	17	23	PB7/AIN7/IT8	I/O	C_T	10mA	**X**		X	X		X	**Port B7**	ADC analog input 7
10	11	18	24	PB6/AIN6/IT7	I/O	C_T	10mA	**X**		X	X		X	**Port B6**	ADC analog input 6
11	12	19	25	V_{PP}/TEST	S									Programming supply	
12	13	20	26	PB5/AIN5/IT6	I/O	C_T	10mA	**X**		X	X		X	**Port B5**	ADC analog input 5
13	14	21	27	PB4/AIN4/IT5	I/O	C_T	10mA	**X**		X	X		X	**Port B4**	ADC analog input 4
14	15	22	28	PB3/AIN3	I/O	C_T	10mA	**X**			X		X	**Port B3**	ADC analog input 3
15	16	23	29	PB2/AIN2	I/O	C_T	10mA	**X**			X		X	**Port B2**	ADC analog input 2
16	17	24	30	PB1/AIN1	I/O	C_T	10mA	**X**			X		X	**Port B1**	ADC analog input 1
17	18	25	31	PB0/AIN0	I/O	C_T	10mA	**X**			X		X	**Port B0**	ADC analog Input 0
18	19	26	32	PA7/OCMP2/IT4	I/O	C_T			**X**	X			X	**Port A7**	Timer Output Compare 2
19	20	27	33	PA6/OCMP1/IT3	I/O	C_T			**X**	X			X	**Port A6**	Timer Output Compare 1
20	21	28	34	PA5/ICAP2/IT2	I/O	C_T			**X**	X			X	**Port A5**	Timer Input Capture 2
21	22	29	35	PA4/ICAP1/IT1	I/O	C_T			**X**	X			X	**Port A4**	Timer Input Capture 1
22	23	30	36	PA3/EXTCLK	I/O	C_T			**X**				X	**Port A3**	Timer External Clock
23	24	31	38	PA2/SCL/ICCCLK	I/O	C_T	25mA	**X**				T		**Port A2**	I²C serial clock, ICC Clock
-	-	32	39	PD0[1)]/AIN8	I/O	C_T		**X**			X		X	**Port D0**	ADC analog Input 8

Pin n°				Pin name	Type	Level		Port / Control						Main function (after reset)	Alternate function
								Input				Output			
SDIP32	SO34	QFN40	LQFP48			Input	Output	float	wpu	int	ana	OD	PP		
-	-	33	40	PD1[1]/AIN9	I/O	C_T		**X**			X		X	**Port D1**	ADC analog Input 9
-	-	34	41	PD2[1]/AIN10	I/O	C_T		**X**			X		X	**Port D2**	ADC analog Input 10
-	-	35	42	PD3[1]/AIN11	I/O	C_T		**X**			X		X	**Port D3**	ADC analog Input 11
-	-	36	43	PD4[1]	I/O	C_T			X				X	**Port D4**	
-	-	37	44	PD5[1]	I/O	C_T			X				X	**Port D5**	
-	-	38	45	PD6[1]	I/O	C_T			X				X	**Port D6**	
-	-	39	46	PD7[1]	I/O	C_T			X				X	**Port D7**	
-	25	-	-	NC	--									Not connected	
24	26	-	-	NC	--									Not connected	
25	27	-	-	NC	--									Not connected	
26	28	40	47	PA1/SDA/ICCDATA	I/O	C_T	25mA	**X**				T		**Port A1**	I²C serial data, ICC Data
27	29	1	48	PA0/MCO	I/O	C_T				X			X	**Port A0**	Main Clock Output
28	30	2	1	V_{SSA}	S									Analog ground	
29	31	3	2	USBDP	I/O									USB bidirectional data (data +)	
30	32	4	3	USBDM	I/O									USB bidirectional data (data -)	
31	33	5	4	USBVCC [2]	O									USB power supply [2]	
32	34	6	5	V_{DDA}	S									Analog supply voltage	

Note:

1. Port D functions are not available on the 8K version of the QFN40 package (ST7263BK2) and should not be connected.

2. The drive from USBVcc is sufficient to only drive an external pull-up in addition to the internal transceiver.

Table 3. Device pin description (SO24)

Pin n°	Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
					Input				Output			
SO24			Input	Output	float	wpu	int	ana	OD	PP		
1	V_{DD}	S									Power supply voltage (4V - 5.5V)	
2	OSCOUT	O									Oscillator output	
3	OSCIN	I									Oscillator input	
4	V_{SS}	S									Digital ground	
5	PC1/TDO	I/O	C_T			**X**				X	**Port C1**	SCI Transmit Data Output
6	PC0/RDI	I/O	C_T			**X**				X	**Port C0**	SCI Receive Data Input
7	RESET	I/O				**X**			X		Reset	

Pin n°	Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
					Input				Output			
SO24			Input	Output	float	wpu	int	ana	OD	PP		
8	PB6/IT7	I/O	C_T	10mA	**X**		X	X		X	**Port B6**	
9	V_{PP}/TEST	S									Programming supply	
10	PB3	I/O	C_T	10mA	**X**			X		X	**Port B3**	
11	PB2	I/O	C_T	10mA	**X**			X		X	**Port B2**	
12	PB1/USBOE	I/O	C_T	10mA	**X**			X		X	**Port B1**	USB Output Enable
13	PB0	I/O	C_T	10mA	**X**			X		X	**Port B0**	
14	PA7/OCMP2/IT4	I/O	C_T			**X**	X			X	**Port A7**	Timer Output Compare 2
15	PA5/ICAP2/IT2	I/O	C_T			**X**	X			X	**Port A5**	Timer Input Capture 2
16	PA4/ICAP1/IT1	I/O	C_T			**X**	X			X	**Port A4**	Timer Input Capture 1
17	PA3/EXTCLK	I/O	C_T			**X**				X	**Port A3**	Timer External Clock
18	PA2/SCL/ ICCCLK	I/O	C_T	25mA	**X**				T		**Port A2**	I^2C serial clock, ICC Clock
19	PA1/SDA/ICCDATA	I/O	C_T	25mA	**X**				T		**Port A1**	I^2C serial data, ICC Data
20	PA0/MCO	I/O	C_T				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	

3 Register & memory map

As shown in Figure 7, the MCU is capable of addressing 32 Kbytes of memories and I/O registers.

The available memory locations consist of up to 1024 bytes of RAM including 64 bytes of register locations, and up to 32K bytes 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.

IMPORTANT: Memory locations noted "Reserved" must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Figure 7. Memory map

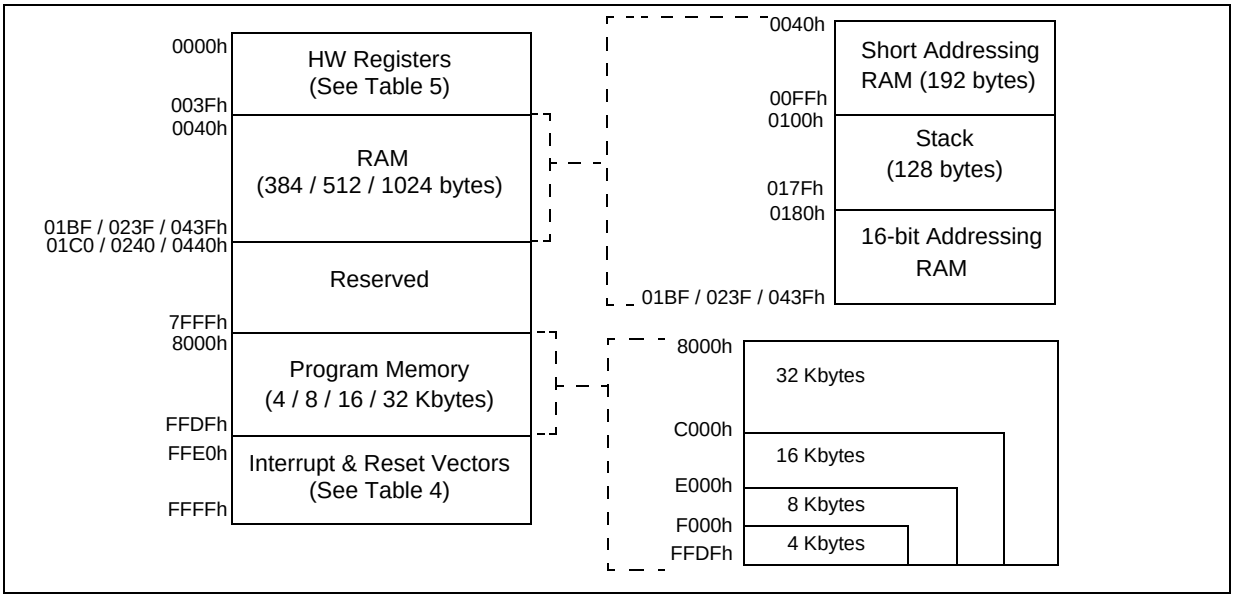


Table 4. interrupt vector map

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	I²C Interrupt Vector	I- bit	Internal Interrupt	No
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	Port A	PADR	Port A Data Register	00h	R/W
0001h		PADDR	Port A Data Direction Register	00h	R/W
0002h	Port B	PBDR	Port B Data Register	00h	R/W
0003h		PBDDR	Port B Data Direction Register	00h	R/W
0004h	Port C	PCDR	Port C Data Register	1111 x000b	R/W
0005h		PCDDR	Port C Data Direction Register	1111 x000b	R/W
0006h	Port D	PDDR	Port D Data Register	00h	R/W
0007h		PDDDR	Port D Data Direction Register	00h	R/W
0008h	ITC	ITIFRE	Interrupt Register	00h	R/W
0009h	MISC	MISCR	Miscellaneous Register	00h	R/W
000Ah	ADC	ADCDR	ADC Data Register	00h	Read only
000Bh		ADCCSR	ADC control Status register	00h	R/W
000Ch	WDG	WDGCR	Watchdog Control Register	7Fh	R/W
000Dh to 0010h	Reserved (4 bytes)				
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

Address	Block	Register label	Register name	Reset status	Remarks
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)				
0032h 0036h	Reserved (5 bytes)				
0037h	Flash	FCSR	Flash Control /Status Register	00h	R/W
0038h	Reserved (1 byte)				
0039h	I²C	I2CDR	I²C Data Register	00h	R/W
003Ah			Reserved	-	
003Bh		I2COAR	I²C (7 Bits) Slave Address Register	00h	R/W
003Ch		I2CCCR	I²C Clock Control Register	00h	R/W
003Dh		I2CSR2	I²C 2nd Status Register	00h	Read only
003Eh		I2CSR1	I²C 1st Status Register	00h	Read only
003Fh		I2CCR	I²C Control Register	00h	R/W

4 Flash program memory

4.1 Introduction

The ST7 dual voltage High Density Flash (HDFlash) is a non-volatile memory that can be electrically erased as a single block or by individual sectors and programmed on a byte-by-byte basis using an external V_{PP} supply.

The HDFlash devices can be programmed and erased off-board (plugged in a programming tool) or on-board using ICP (in-circuit programming) or IAP (in-application programming).

The array matrix organisation allows each sector to be erased and reprogrammed without affecting other sectors.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
- Read-Out protection
- Register access security system (RASS) to prevent accidental programming or erasing

4.2 Structure

The Flash memory is organised 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 8). 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

4.2.1 Read-Out protection

Read-Out 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.

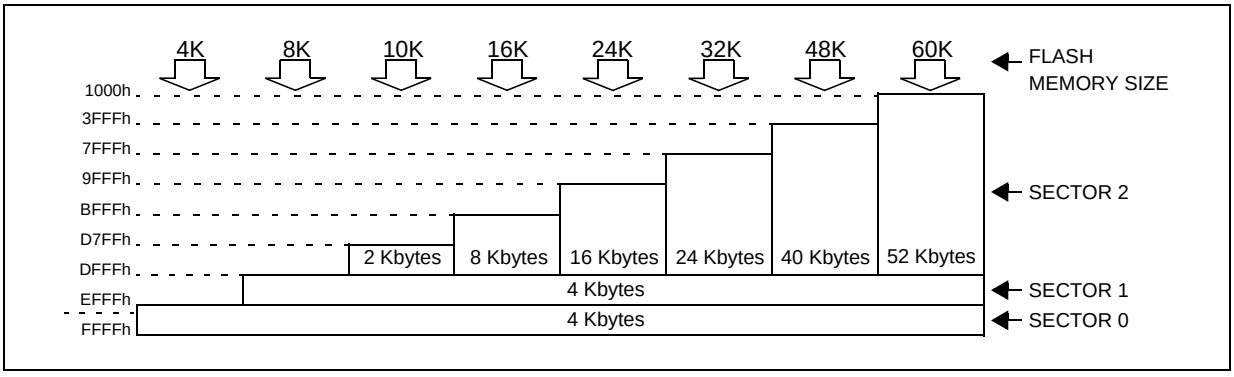
In Flash devices, this protection is removed by reprogramming the option. In this case, the entire program memory is first automatically erased and the device can be reprogrammed.

Read-Out protection selection depends on the device type:

– In Flash devices it is enabled and removed through the FMP_R bit in the option byte.

– In ROM devices it is enabled by mask option specified in the Option List.

Figure 8. Memory map and sector address



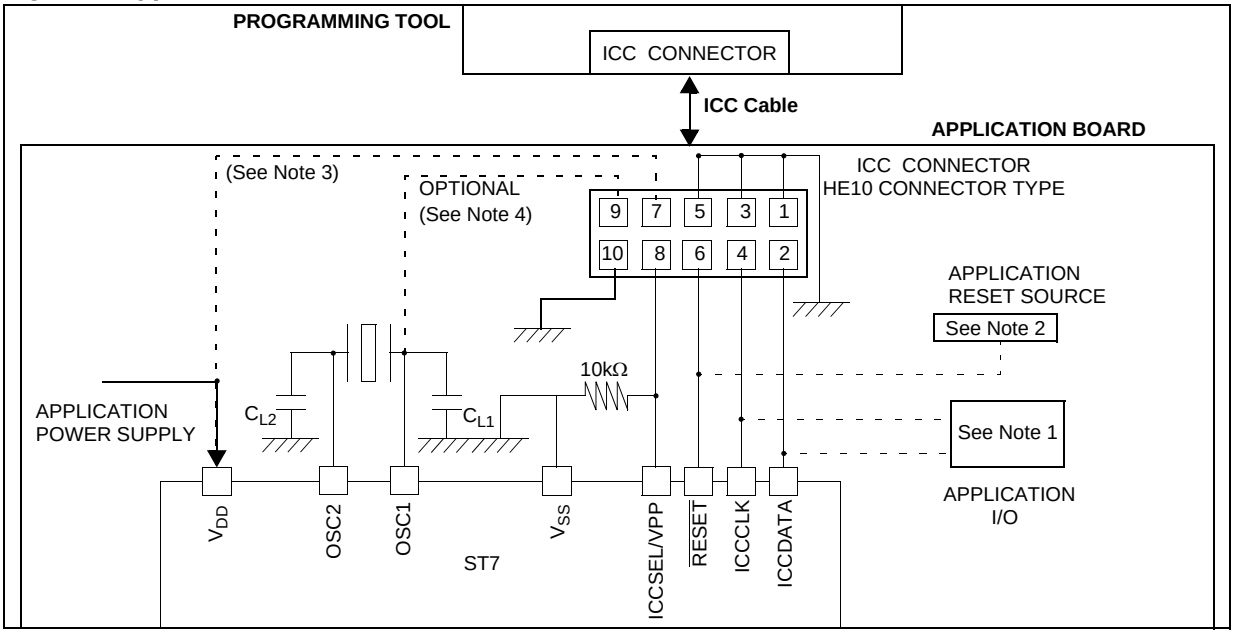
Flash program memory (Cont'd)

4.3 ICC interface

ICC (in-circuit communication) needs a minimum of four and up to six pins to be connected to the programming tool (see Figure 9). 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 9, Note 3)

Figure 9. Typical ICC interface



Notes:

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 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 $\overline{\text{RESET}}$ 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 or 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 multioscillator capability need to have OSC2 grounded in this case.

Flash program memory (Cont'd)

4.4 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 9). For more details on the pin locations, refer to the device pinout description.

4.5 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, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored, etc.). For example, it is possible to download code from the SPI, SCI or other type of serial 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.

4.6 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*.

4.7 Register description

Flash Control/Status register (FCSR)

Read/Write

Reset value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	0	0

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

5 Central processing unit

5.1 Introduction

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

5.2 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

5.3 CPU registers

The six CPU registers shown in Figure 10 are not present in the memory mapping and are accessed by specific instructions.

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.

Index registers (X and Y)

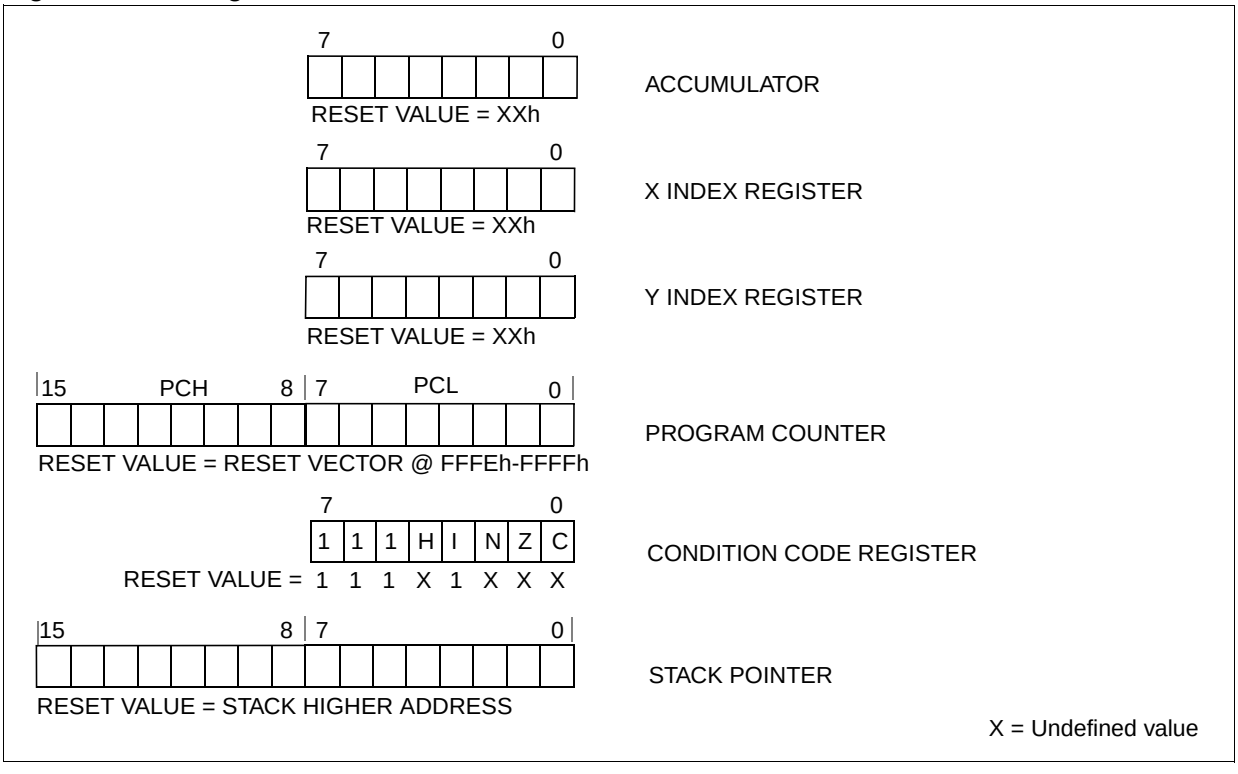
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The cross-assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

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

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).

Figure 10. CPU registers



CPU registers (cont'd)

Condition Code register (CC)

Read/Write

Reset value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

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.

Bit 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 instruction. 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.

Bit 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.

Bit 2 = **N** *Negative*

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7^{th} bit of the result.
0: The result of the last operation is positive or null.
1: The result of the last operation is negative (that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero*

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.
0: The result of the last operation is different from zero.
1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = **C** *Carry/borrow*

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.
0: No overflow or underflow has occurred.
1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

6 CPU registers (Cont'd)

Stack Pointer (SP)

Read/Write

Reset value: 017Fh

15							8
0	0	0	0	0	0	0	1

7							0
0	SP6	SP5	SP4	SP3	SP2	SP1	SP0

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 11).

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 SP6 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 a 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.

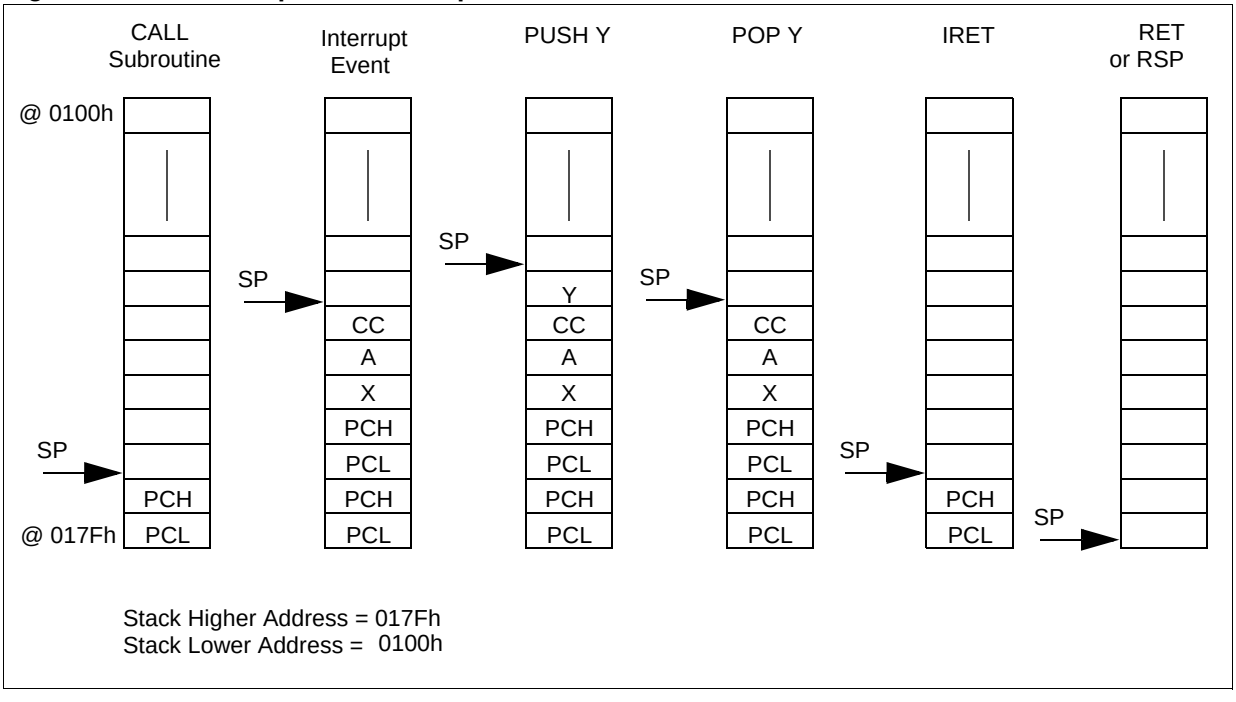
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 11.

– 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 11. 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.

7.1.1 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.

7.1.2 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 12).

7.1.3 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 15, 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 12. Low voltage detector functional diagram

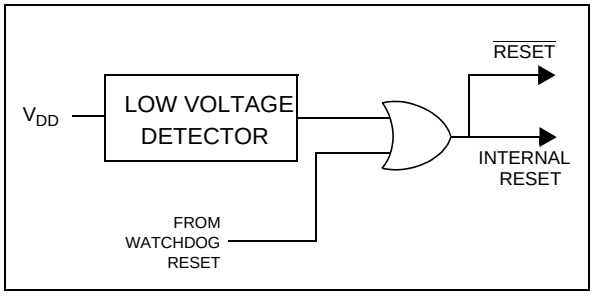
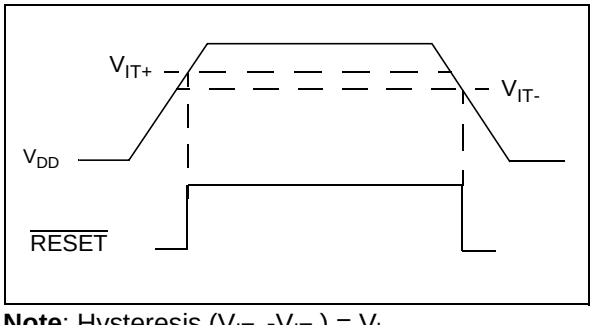
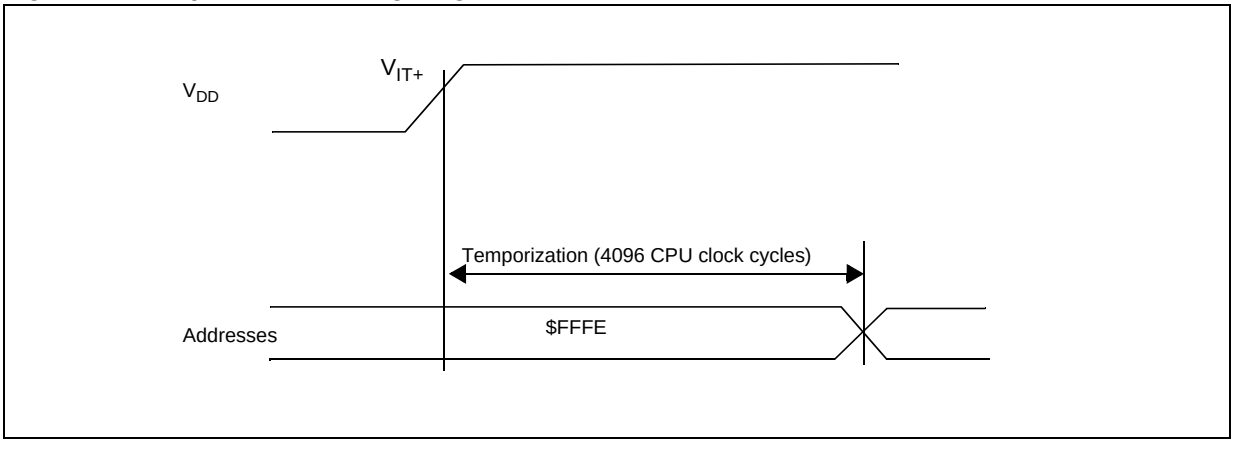


Figure 13. Low voltage reset signal output



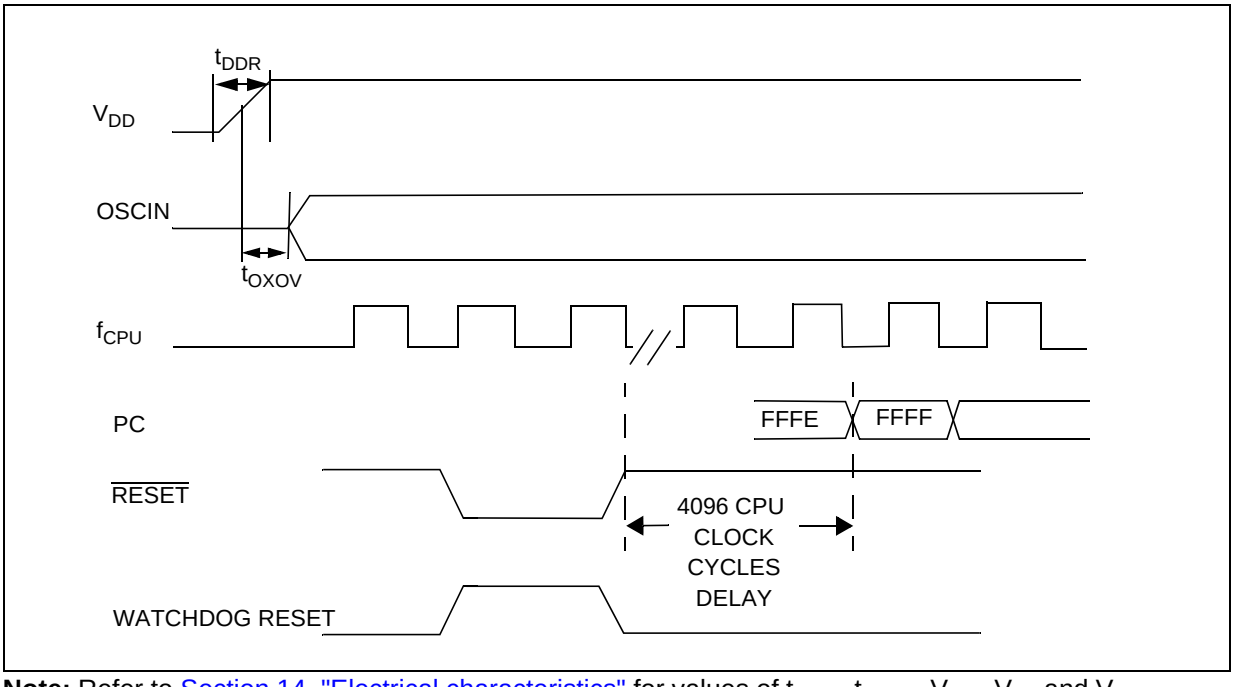
Note: Hysteresis (V_{IT+}-V_{IT-}) = V_{hys}

Figure 14. Temporization timing diagram after an internal Reset



Reset (Cont'd)

Figure 15. Reset timing diagram



Note: Refer to Section 14, "Electrical characteristics" for values of t_{DDR}, t_{OXOV}, V_{IT+}, V_{IT-} and V_{hys}

7.2 Clock system

7.2.1 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 18).

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 17 is recommended when using a crystal, and Table 7, "Recommended values for 24 MHz crystal resonator" 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 7. Recommended values for 24 MHz crystal resonator

R_{SMAX}	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

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

7.2.2 External clock

An external clock may be applied to the OSCIN input with the OSCOUT pin not connected, as shown on Figure 16. 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 6.5 CONTROL TIMING).

Figure 16. External clock source connections

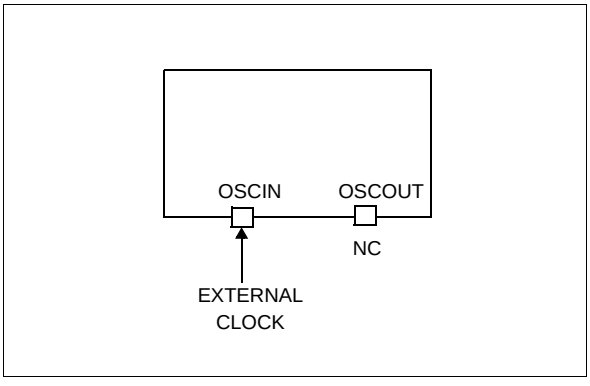


Figure 17. Crystal/ceramic resonator

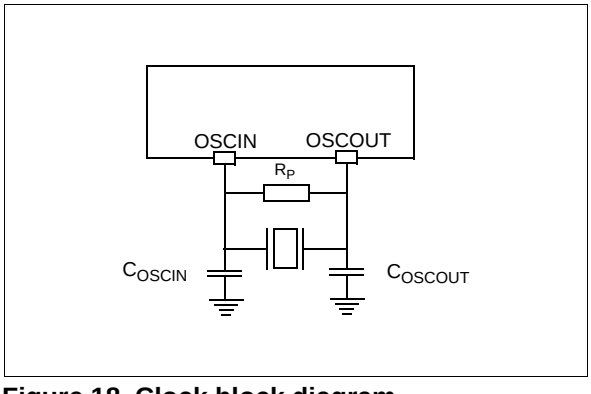
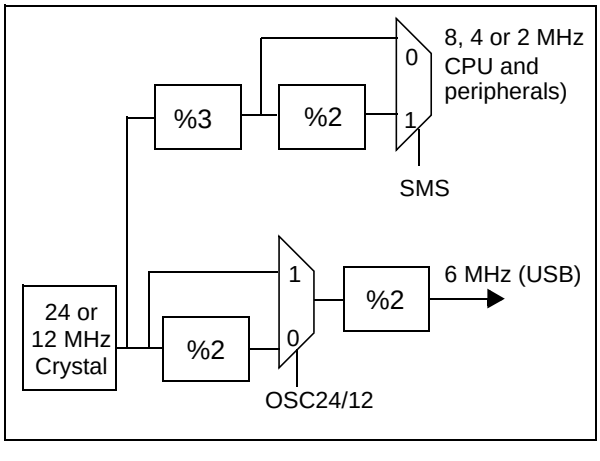


Figure 18. Clock block diagram



8 Interrupts

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in Table 8, "Interrupt mapping" and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 19.

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 8, "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 8, "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 19.

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 8, "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.

Notes:

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.

Interrupts (Cont'd)

Figure 19. Interrupt processing flowchart

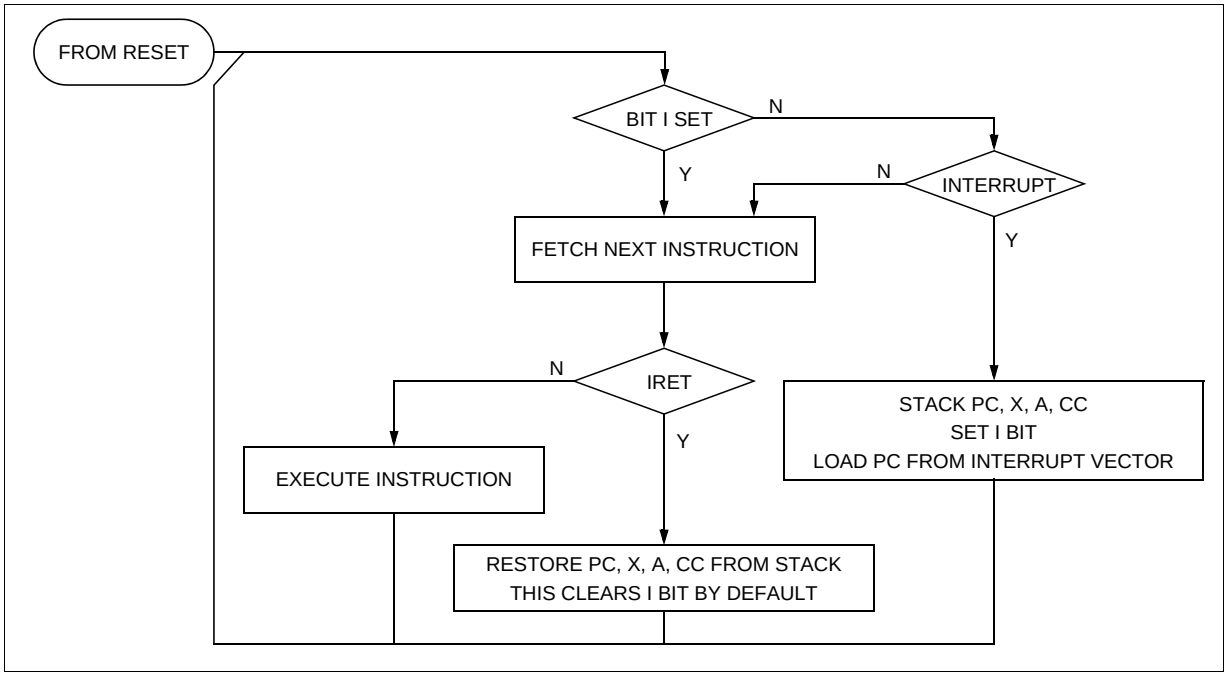


Table 8. Interrupt mapping

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	I²C	I²C Peripheral Interrupts	I²CSR1			FFF2h-FFF3h
			I²CSR2			
4	SCI	SCI Peripheral Interrupts	SCISR	Lowest Priority		FFF0h-FFF1h
5	USB	USB Peripheral Interrupts	ISTR			FFEEh-FFEFh

Interrupts (Cont'd)

8.1 Interrupt register

Interrupt register (ITRFRE)

Address: 0008h — Read/Write
Reset value: 0000 0000 (00h)

7							0
IT8E	IT7E	IT6E	IT5E	IT4E	IT3E	IT2E	IT1E

Bit 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.

Note: Analog input must be disabled for interrupts coming from port B.

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.

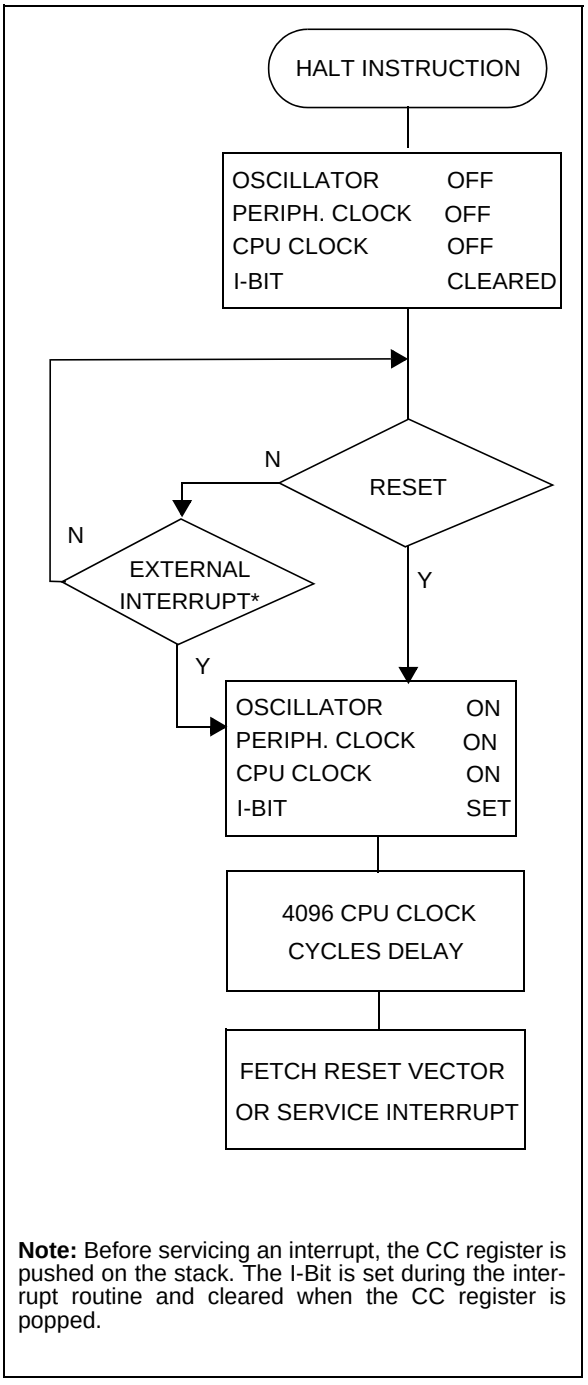
9.2 Halt mode

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 20. 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.

Power saving modes (Cont'd)

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 21.

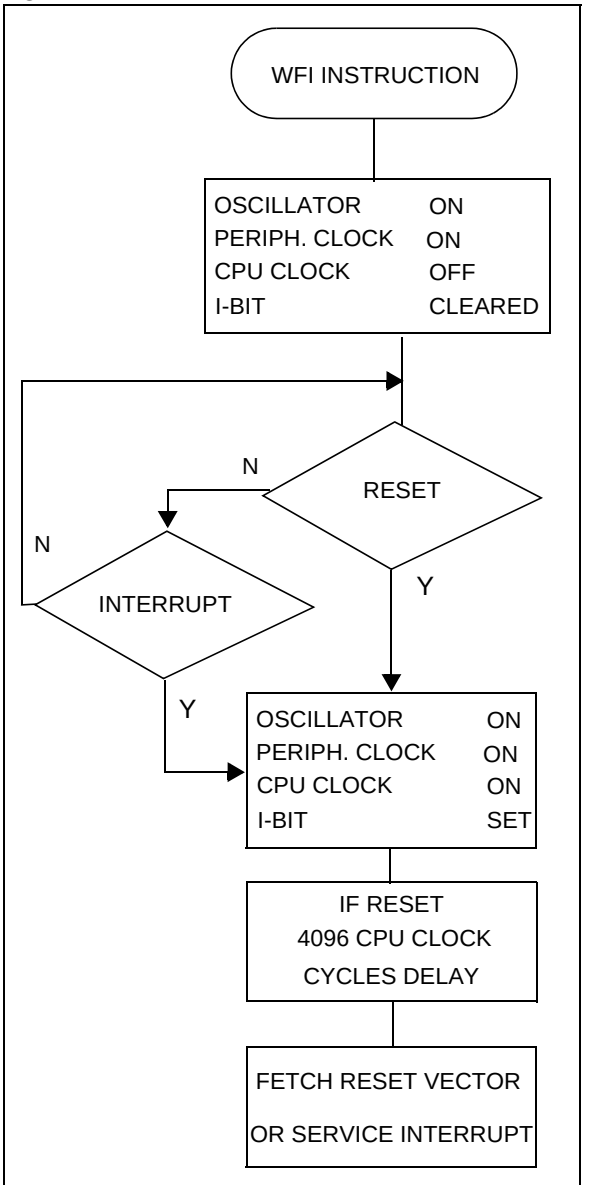
Related documentation

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 21. 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

– Analog signal input (ADC)

– 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 9. 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.
Note 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 7).

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.

Digital 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.

Notes:
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).

***Warning*:** 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.

I/O ports (Cont'd)

Analog alternate function

When the pin is used as an ADC input the I/O must be configured as a floating input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail which is connected to the ADC input.

It is recommended not to change the voltage level or loading on any port pin while conversion is in progress. Furthermore it is recommended not to have clocking pins located close to a selected analog pin.

***Warning**:* The analog input voltage level must be within the limits stated in the Absolute Maximum Ratings.

10.3 I/O port implementation

The hardware implementation on each I/O port depends on the settings in the DDR register and specific feature of the I/O port such as ADC Input or true open drain.

I/O ports (Cont'd)

10.3.1 Port A

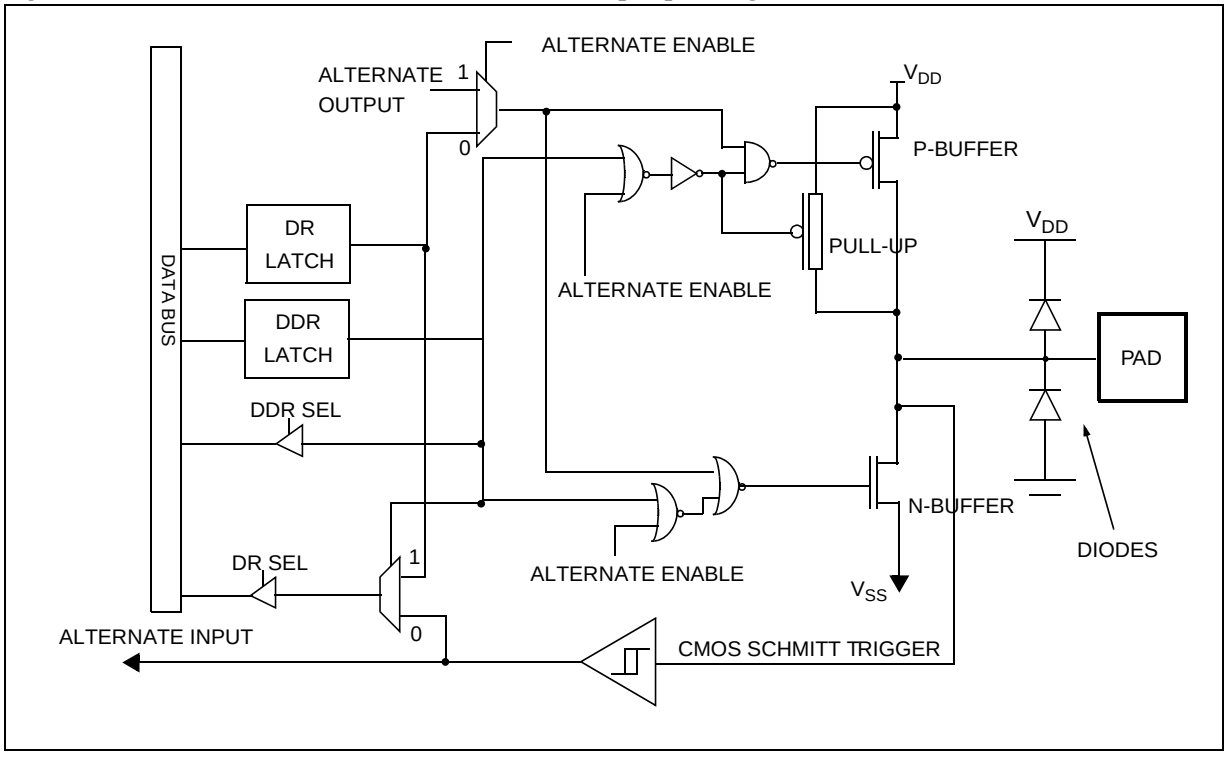
Table 10. Port A0, A3, A4, A5, A6, A7 description

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 22. PA0, PA3, PA4, PA5, PA6, PA7 and PD[7:4] configuration



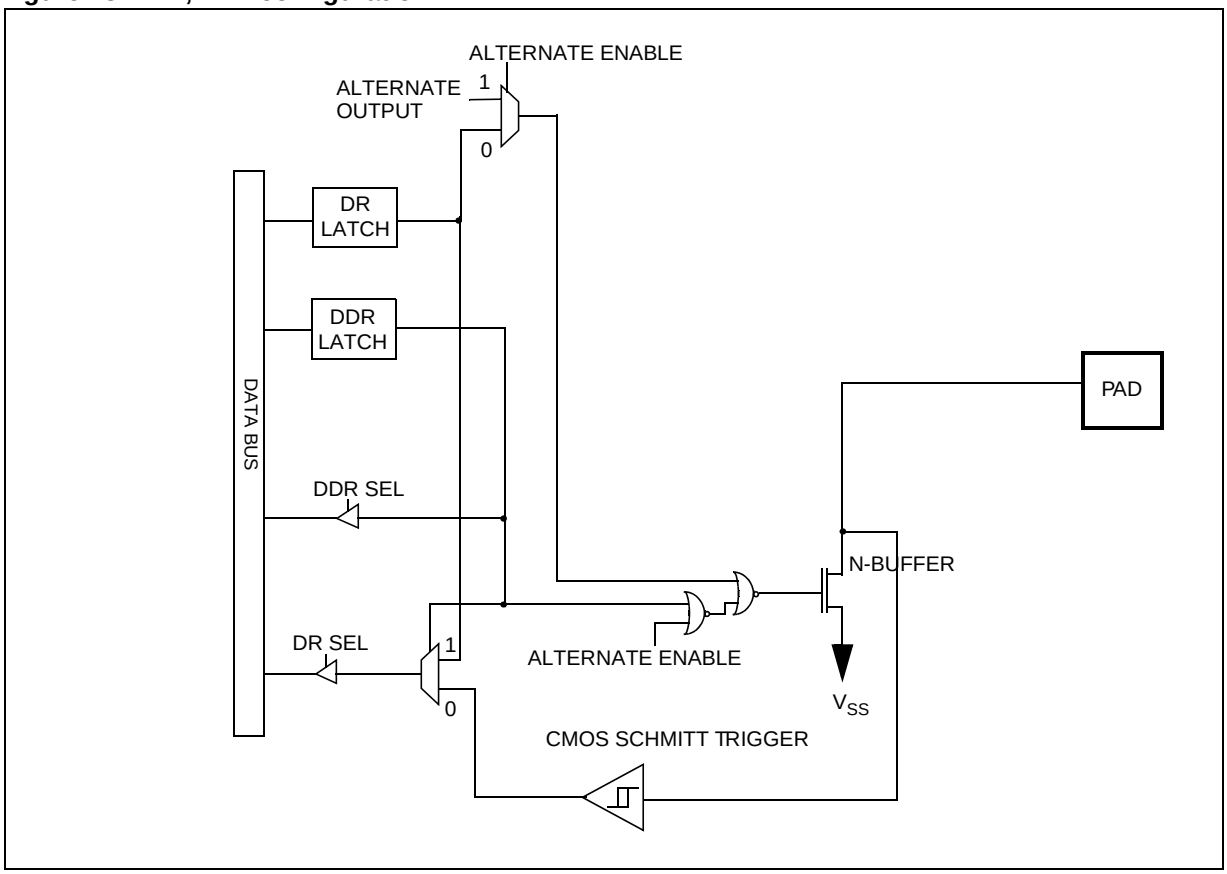
I/O ports (Cont'd)

Table 11. PA1, PA2 description

PORT A	I / O		Alternate function	
	Input[1]	Output	Signal	Condition
PA1	without pull-up	Very High Current open drain	SDA (I^2C data)	I^2C enable
PA2	without pull-up	Very High Current open drain	SCL (I^2C clock)	I^2C enable

[1]Reset State

Figure 23. PA1, PA2 configuration



I/O ports (Cont'd)

10.3.2 Port B

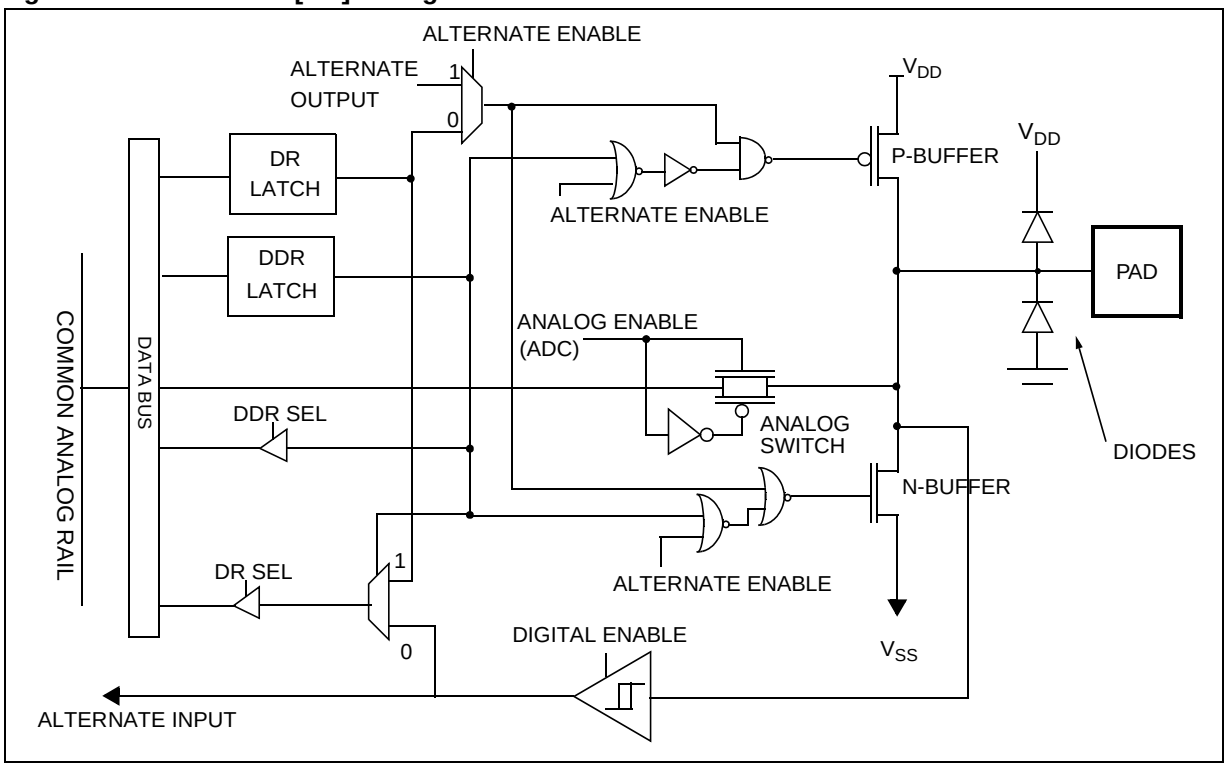
Table 12. Port B description

PORT B	I/O		Alternate function	
	Input[1]	Output	Signal	Condition
PB0	without pull-up	push-pull	Analog input (ADC)	CH[3:0] = 000 (ADCCSR)
PB1	without pull-up	push-pull	Analog input (ADC)	CH[3:0] = 001 (ADCCSR)
			USBOE (USB output enable)[2]	USBOE =1 (MISCR)
PB2	without pull-up	push-pull	Analog input (ADC)	CH[3:0]= 010 (ADCCSR)
PB3	without pull-up	push-pull	Analog input (ADC)	CH[3:0]= 011 (ADCCSR)
PB4	without pull-up	push-pull	Analog input (ADC)	CH[3:0]= 100 (ADCCSR)
			IT5 Schmitt triggered input	IT4E = 1 (ITIFRE)
PB5	without pull-up	push-pull	Analog input (ADC)	CH[3:0]= 101 (ADCCSR)
			IT6 Schmitt triggered input	IT5E = 1 (ITIFRE)
PB6	without pull-up	push-pull	Analog input (ADC)	CH[3:0]= 110 (ADCCSR)
			IT7 Schmitt triggered input	IT6E = 1 (ITIFRE)
PB7	without pull-up	push-pull	Analog input (ADC)	CH[3:0]= 111 (ADCCSR)
			IT8 Schmitt triggered input	IT7E = 1 (ITIFRE)

[1] Reset State

[2] On SO24 only

Figure 24. Port B and D[3:0] configuration



I/O ports (Cont'd)

10.3.3 Port C

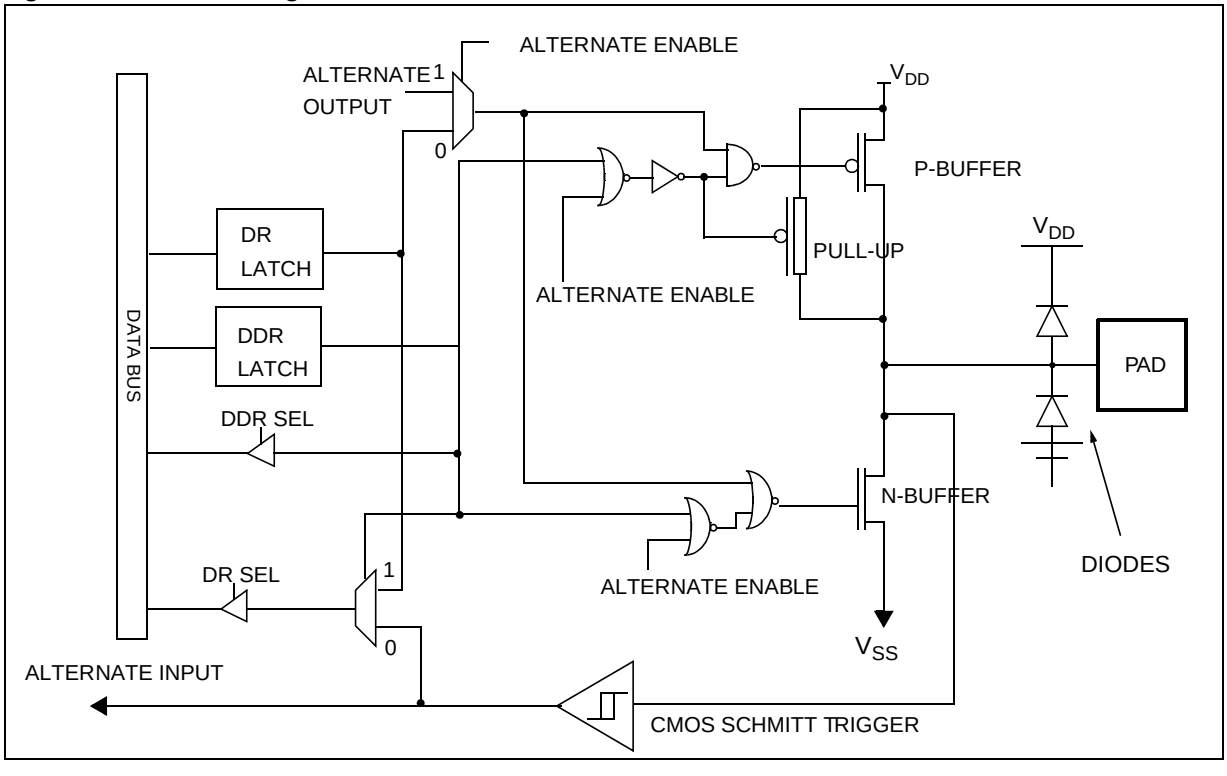
Table 13. Port C description

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 25. Port C configuration



10.3.4 Port D

Table 14. Port D description

PORT D	I / O		Alternate function	
	Input*	Output	Signal	Condition
PD0	without pull-up	push-pull	Analog input (ADC)	CH[3:0] = 1000 (ADCCSR)
PD1	without pull-up	push-pull	Analog input (ADC)	CH[3:0] = 1001 (ADCCSR)
PD2	without pull-up	push-pull	Analog input (ADC)	CH[3:0] = 1010 (ADCCSR)
PD3	without pull-up	push-pull	Analog input (ADC)	CH[3:0] = 1011 (ADCCSR)
PD4	with pull-up	push-pull		
PD5	with pull-up	push-pull		
PD6	with pull-up	push-pull		
PD7	with pull-up	push-pull		

*Reset State

I/O ports (Cont'd)

10.3.5 Register description

Data registers (PxDR)

Port A Data Register (PADR): 0000h

Port B Data Register (PBDR): 0002h

Port C Data Register (PCDR): 0004h

Port D Data Register (PDDR): 0006h

Read/Write

Reset value Port A: 0000 0000 (00h)

Reset value Port B: 0000 0000 (00h)

Reset value Port C: 1111 x000 (FXh)

Reset value Port D: 0000 0000 (00h)

Note: For Port C, unused bits (7-3) are not accessible.

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = D[7:0] *Data Register 8 bits.*

The DR register has a specific behaviour 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).

Note: When using open-drain I/Os in output configuration, the value read in DR is the digital value applied to the I/Opin.

Data Direction Register (PxDDR)

Port A Data Direction Register (PADDR): 0001h

Port B Data Direction Register (PBDDR): 0003h

Port C Data Direction Register (PCDDR): 0005h

Port D Data Direction Register (PDDDR): 0007h

Read/Write

Reset value Port A: 0000 0000 (00h)

Reset value Port B: 0000 0000 (00h)

Reset value Port C: 1111 x000 (FXh)

Reset value Port D: 0000 0000 (00h)

Note: For Port C, unused bits (7-3) are not accessible

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Bit 7:0 = DD[7:0] *Data Direction Register 8 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

Table 15. I/O ports register map

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
00	**PADR**	MSB							LSB
01	**PADDR**	MSB							LSB
02	**PBDR**	MSB							LSB
03	**PBDDR**	MSB							LSB
04	**PCDR**	MSB							LSB
05	**PCDDR**	MSB							LSB
06	**PDDR**	MSB							LSB
07	**PDDDR**	MSB							LSB

Related documentation

AN 970: SPI communication between ST7 and EEPROM

AN1045: S/W implementation of I2C bus master

AN1048: Software LCD driver

11 Miscellaneous register

Address: 0009h — Read/Write
Reset value: 0000 0000 (00h)

7							0
-	-	-	-	-	SMS	USBOE	MCO

Bit 7:3 = Reserved

Bit 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 18 on page 26). 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.

Bit 1 = **USBOE** *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.

Bit 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)

12 On-chip peripherals

12.1 Watchdog timer (WDG)

12.1.1 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.

12.1.2 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.

12.1.3 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 typically 30µs.

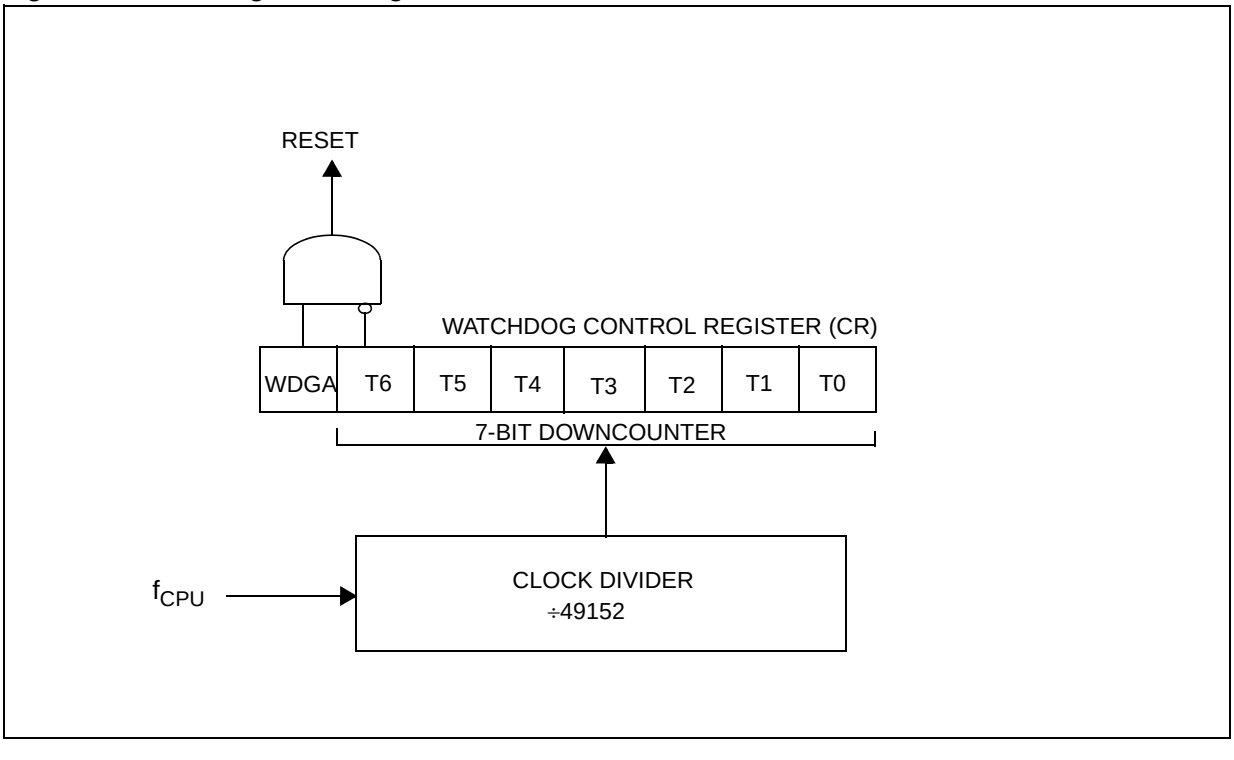
The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is free-running: it counts down even if the watchdog is disabled. The value to be stored in the CR register must be between FFh and C0h (see Table 16, ". Watchdog timing (fCPU = 8 MHz)"):

– 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 26. Watchdog block diagram



Watchdog timer (Cont'd)

Table 16. Watchdog timing (f_{CPU} = 8 MHz)

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

Notes: Following a reset, the watchdog is disabled. 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).

12.1.4 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).

12.1.5 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.

12.1.6 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).

12.1.6.1 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 in ROM 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).

12.1.7 Interrupts

None.

Watchdog timer (Cont'd)

12.1.8 Register description

Control Register (CR)

Read/Write

Reset value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7 = **WDGA** *Activation bit*.
This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.
0: Watchdog disabled
1: Watchdog enabled

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

Table 17. Watchdog timer register map and reset values

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

12.2 16-bit timer

12.2.1 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).

12.2.2 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)*

The block diagram is shown in Figure 27.

***Note:** Some timer pins may not be available (not bonded) in some ST7 devices. Refer to the device pin out description.

When reading an input signal on a non-bonded pin, the value will always be '1'.

12.2.3 Functional description

12.2.3.1 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 (MS byte).
- Counter Low Register (CLR) is the least significant byte (LS byte).

Alternate Counter Register (ACR)

- Alternate Counter High Register (ACHR) is the most significant byte (MS byte).
- Alternate Counter Low Register (ACLR) is the least significant byte (LS byte).

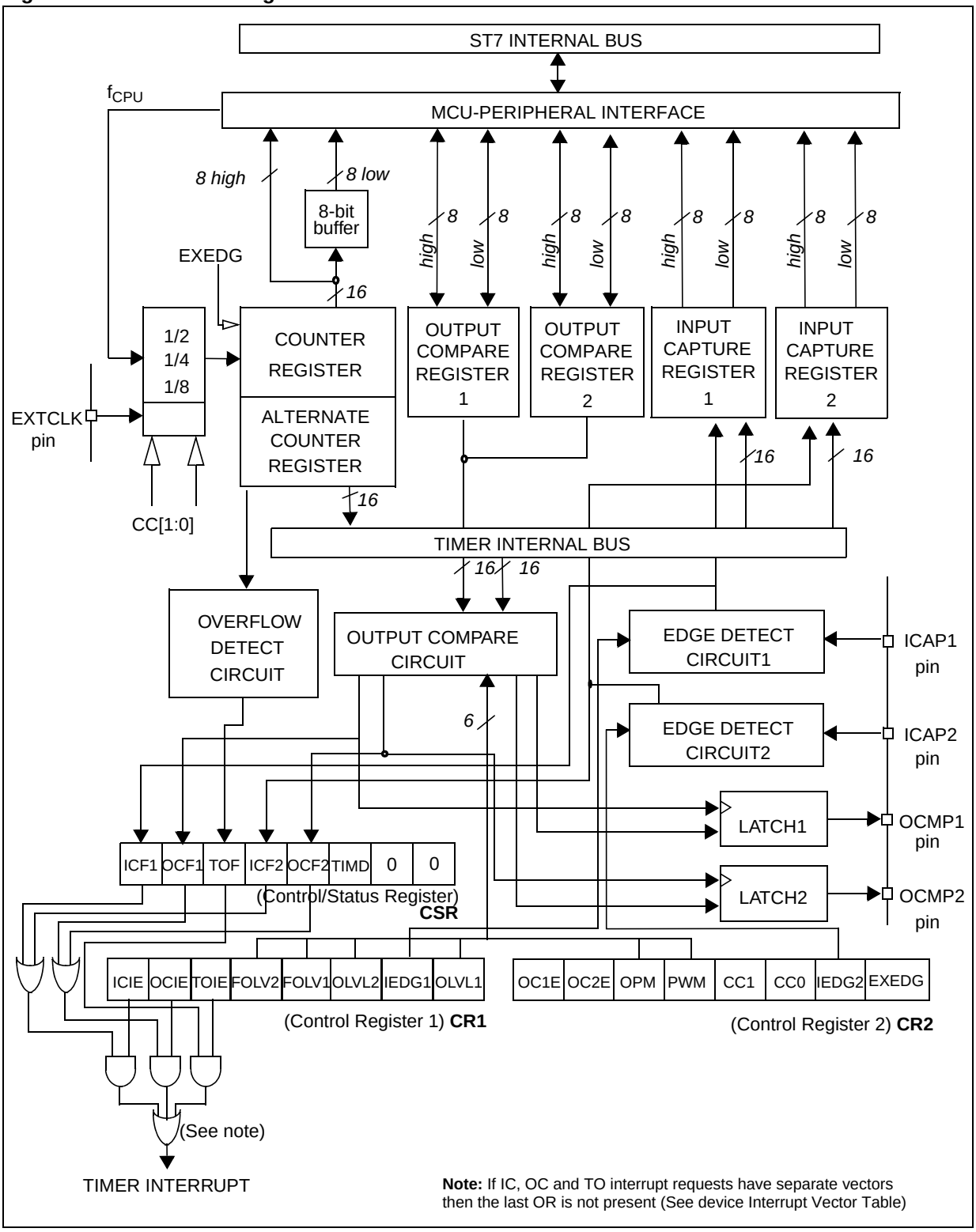
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register, (SR), (see note at the end of paragraph titled 16-bit read sequence).

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 18, "Clock Control Bits". 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.

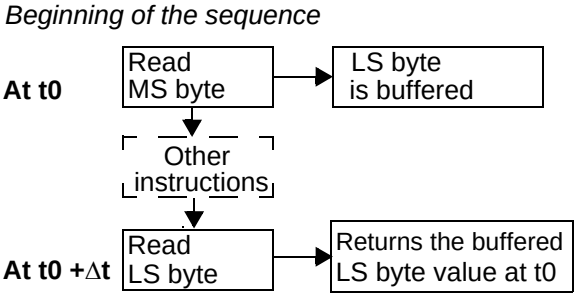
16-bit timer (Cont'd)

Figure 27. Timer block diagram



16-bit timer (Cont'd)

16-bit read sequence: (from either the Counter Register or the Alternate Counter Register).



The user must read the MS byte first, then the LS byte value is buffered automatically.

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

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LS byte 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.

Notes: The TOF bit is not cleared by accesses to 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).

12.2.3.2 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.

16-bit timer (Cont'd)

Figure 28. Counter timing diagram, internal clock divided by 2

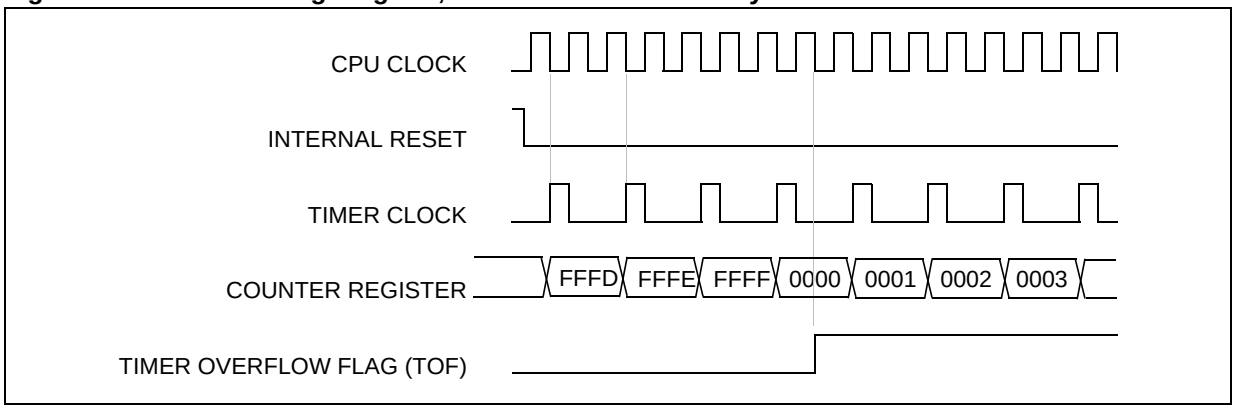


Figure 29. Counter timing diagram, internal clock divided by 4

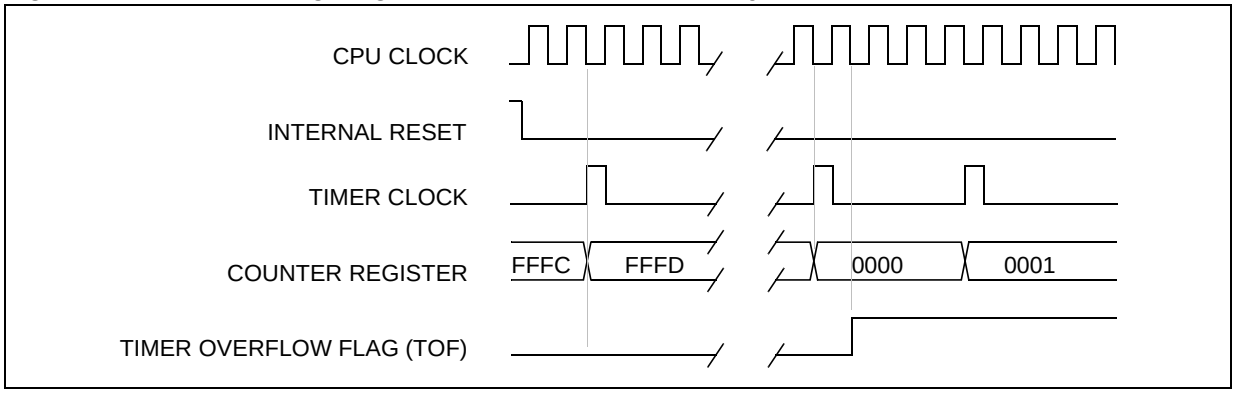
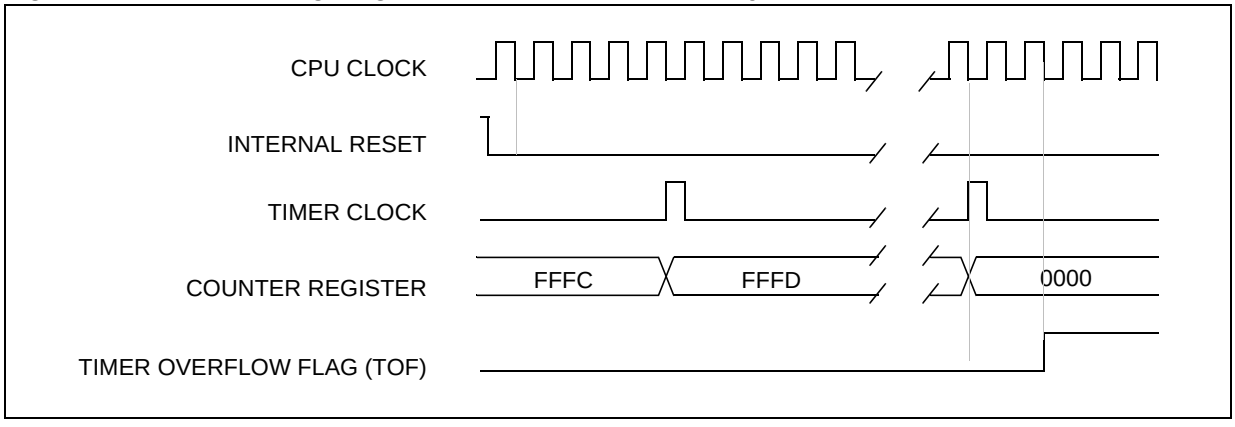


Figure 30. Counter timing diagram, internal clock divided by 8



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

16-bit timer (Cont'd)

12.2.3.3 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 and IC2R) are used to latch the value of the free running counter after a transition is detected on the ICAP*i* pin (see Figure 31).

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

IC*i*R register is a read-only register.

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 18, "Clock Control Bits").

– 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).

And 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 32).

– 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.

Notes:

1. After reading the IC*i*HR register, transfer of input capture data is inhibited and ICF*i* will never be set until the IC*i*LR register is also read.
2. The IC*i*R 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 ICAP*i* 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 IC*i*HR (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).

16-bit timer (Cont'd)

Figure 31. Input Capture block diagram

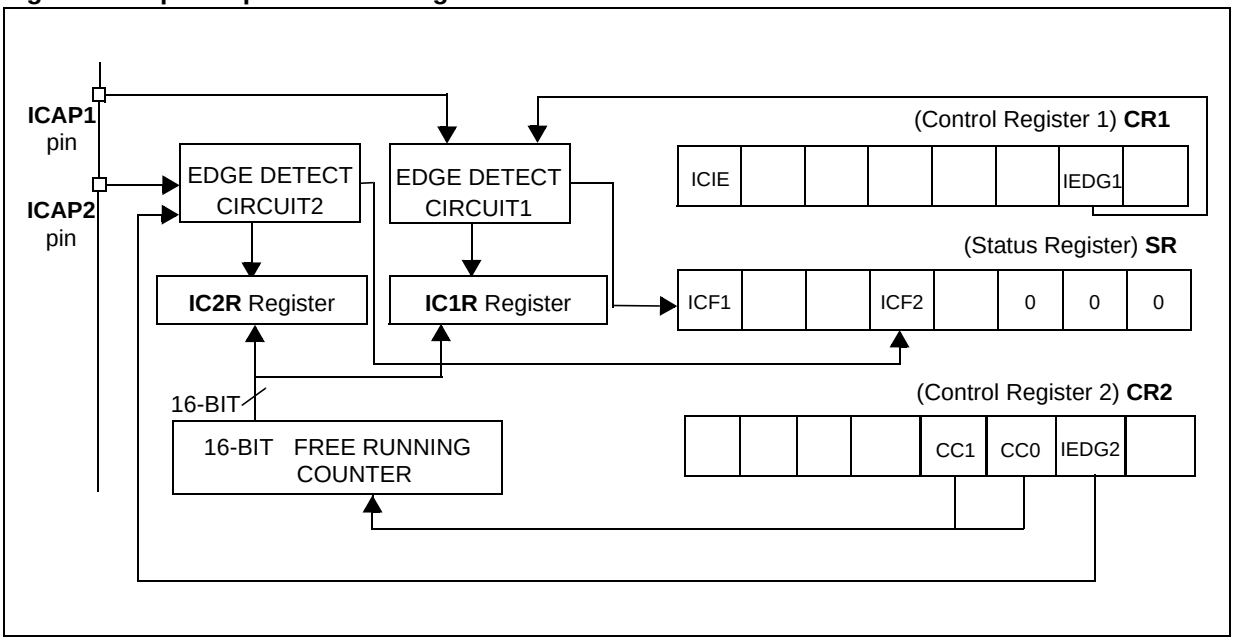
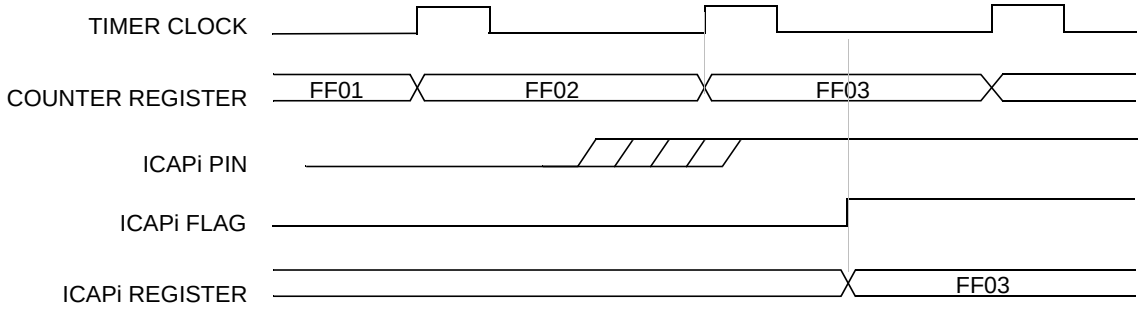


Figure 32. Input Capture timing diagram



Note: The rising edge is the active edge.

16-bit timer (Cont'd)

12.2.3.4 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.

	MS byte	LS byte
OC*i*R	OC*i*HR	OC*i*LR

These registers are readable and writable 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 18, "Clock Control Bits").

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 OC*i*R 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:

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

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 18, "Clock Control Bits")

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

$$\Delta \text{ OC}i\text{R} = \Delta t * f_{EXT}$$

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).

16-bit timer (Cont'd)

Notes:

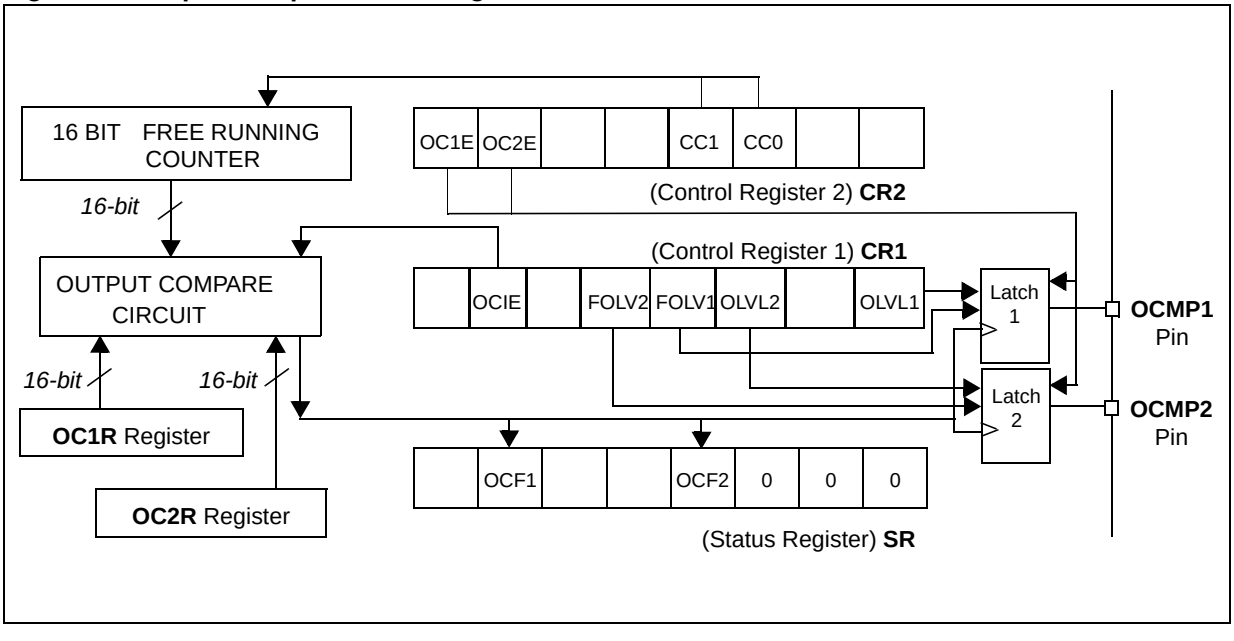
1. After a processor write cycle to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
2. If the OC*i*E bit is not set, the OCMP*i* pin is a general I/O port and the OLVL*i* bit will not appear when a match is found but an interrupt could be generated 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 34 on page 53 for an example with $f_{CPU}/2$ and Figure 35 on page 53 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 OCMP*i* pins even if the input capture mode is also used.
5. The value in the 16-bit OC*i*R register and the OLV*i* bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

Forced Compare output 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 33. Output Compare block diagram



16-bit timer (Cont'd)

Figure 34. Output Compare timing diagram, $f_{TIMER} = f_{CPU}/2$

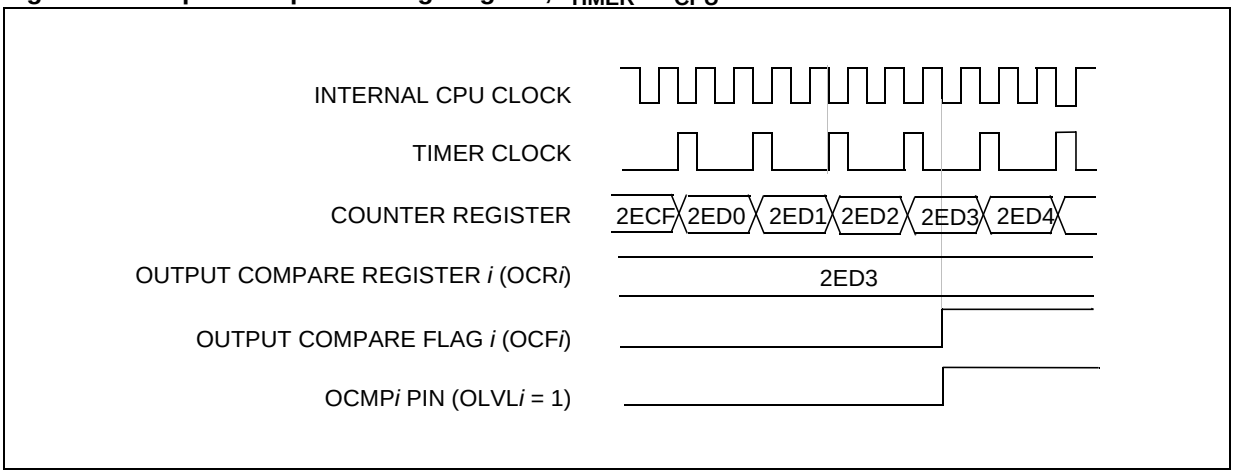
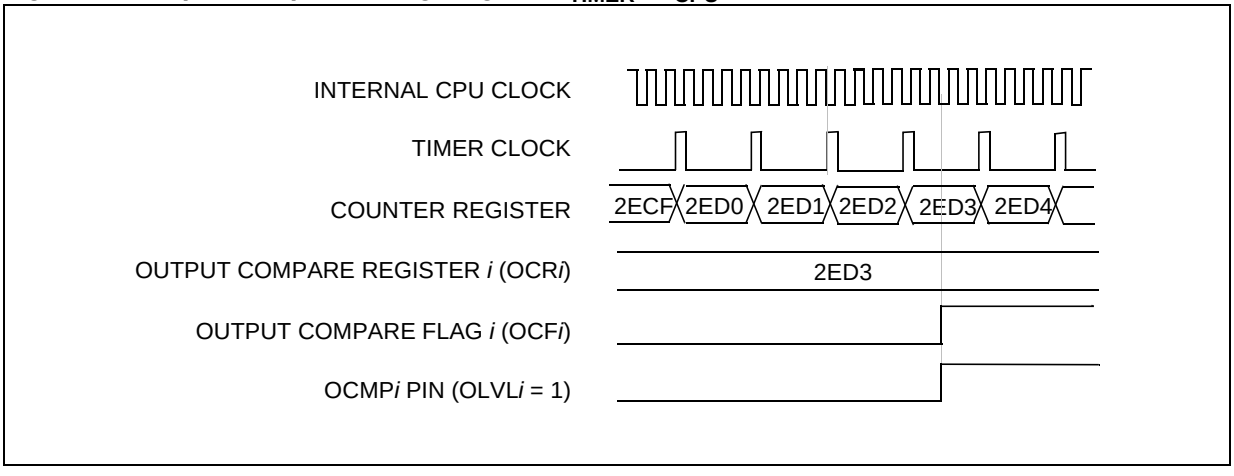


Figure 35. Output Compare timing diagram, $f_{TIMER} = f_{CPU}/4$



16-bit timer (Cont'd)

12.2.3.5 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.

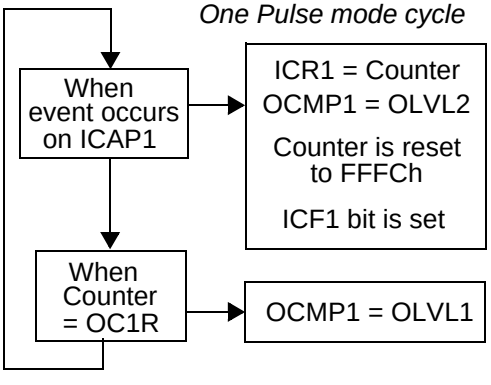
The One Pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure:

To use One Pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in the opposite column).
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 18, "Clock Control Bits").

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 18, "Clock Control Bits")

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)

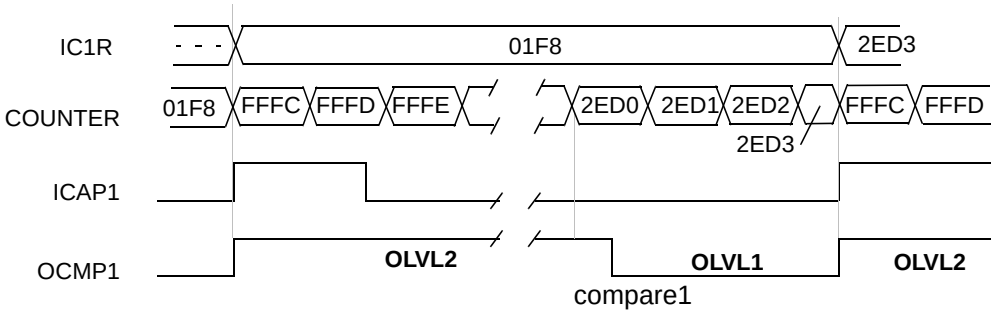
When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin, (See Figure 36).

Notes:

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.

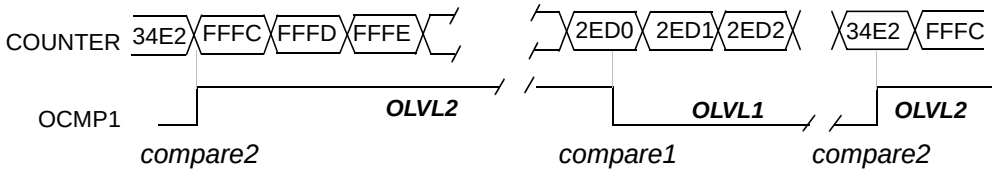
16-bit timer (Cont'd)

Figure 36. One Pulse mode timing example



Note: IEDG1 = 1, OC1R = 2ED0h, OLVL1 = 0, OLVL2 = 1

Figure 37. Pulse Width Modulation mode timing example with 2 Output Compare functions



Note: OC1R = 2ED0h, OC2R = 34E2, OLVL1 = 0, OLVL2 = 1

Note: 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.

16-bit timer (Cont'd)

12.2.3.6 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.

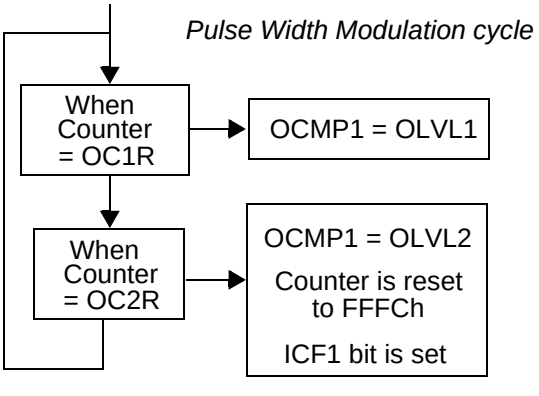
Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so this functionality can not be used when PWM mode is activated.

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 in the opposite column.
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 18, "Clock Control Bits").



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 OC*i*R 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 = Signal or pulse 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 18, "Clock Control Bits")

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

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

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 37)

Notes:

1. After a write instruction to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR 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.

16-bit timer (Cont'd)

12.2.4 Low power modes

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.

12.2.5 Interrupts

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		

Note: The 16-bit timer interrupt events are connected to the same interrupt vector (see Interrupts chapter). 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).

12.2.6 Summary of timer modes

MODES	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 Section 12.2.3.5, "One Pulse mode"

2) See note 5 in Section 12.2.3.5, "One Pulse mode"

3) See note 4 in Section 12.2.3.6, "Pulse Width Modulation mode"

16-bit timer (Cont'd)

12.2.7 Register description

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)

Read/Write

Reset value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 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.

Bit 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.

Bit 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.

Bit 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.

Bit 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.

Bit 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.

Bit 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.

Bit 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.

16-bit timer (Cont'd)

Control Register 2 (CR2)

Read/Write

Reset value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *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.

Bit 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.

Bit 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.

Bit 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.

Bit 3, 2 = **CC[1:0]** *Clock Control.*

The timer clock mode depends on these bits:

Table 18. Clock Control Bits

Timer Clock	CC1	CC0
f_{CPU} / 4	0	0
f_{CPU} / 2		1
f_{CPU} / 8	1	0
External Clock (where available)		1

Note: If the external clock pin is not available, programming the external clock configuration stops the counter.

Bit 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.

Bit 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.

16-bit timer (Cont'd)

Control/status Register (CSR)

Read/Write (bits 7:3 read only)

Reset value: xxxx x0xx (xxh)

7							0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	0	0

Bit 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.

Bit 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.

Bit 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.

Bit 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.

Bit 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.

Bit 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

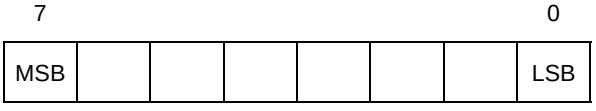
Bits 1:0 = Reserved, must be kept cleared.

16-bit timer (Cont'd)

Input Capture 1 High Register (IC1HR)

Read Only
Reset value: Undefined

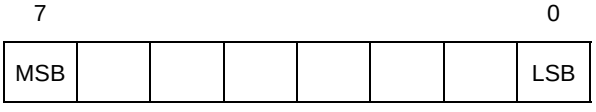
This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).



Input Capture 1 Low Register (IC1LR)

Read Only
Reset value: Undefined

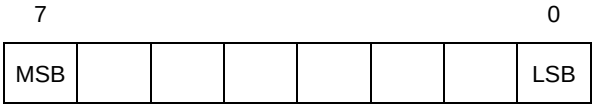
This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).



Output Compare 1 High Register (OC1HR)

Read/Write
Reset value: 1000 0000 (80h)

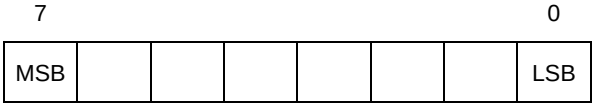
This is an 8-bit register that contains the high part of the value to be compared to the CHR register.



Output Compare 1 Low Register (OC1LR)

Read/Write
Reset value: 0000 0000 (00h)

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

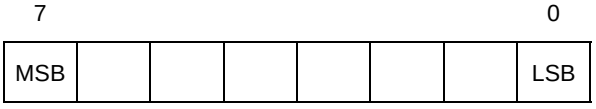


16-bit timer (Cont'd)

Output Compare 2 High Register (OC2HR)

Read/Write
Reset value: 1000 0000 (80h)

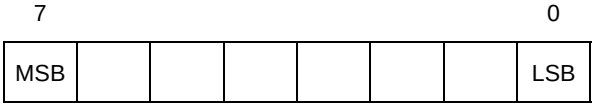
This is an 8-bit register that contains the high part of the value to be compared to the CHR register.



Output Compare 2 Low Register (OC2LR)

Read/Write
Reset value: 0000 0000 (00h)

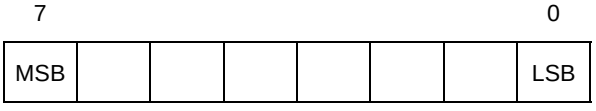
This is an 8-bit register that contains the low part of the value to be compared to the CLR register.



Counter High Register (CHR)

Read Only
Reset value: 1111 1111 (FFh)

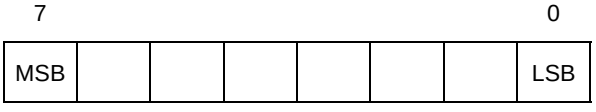
This is an 8-bit register that contains the high part of the counter value.



Counter Low Register (CLR)

Read Only
Reset value: 1111 1100 (FCh)

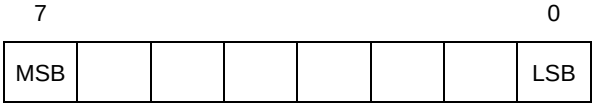
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.



Alternate Counter High Register (ACHR)

Read Only
Reset value: 1111 1111 (FFh)

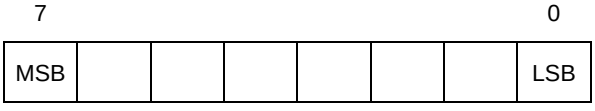
This is an 8-bit register that contains the high part of the counter value.



Alternate Counter Low Register (ACLR)

Read Only
Reset value: 1111 1100 (FCh)

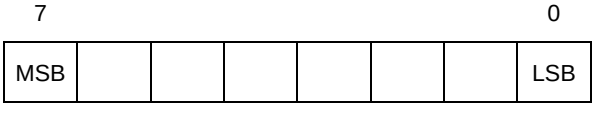
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.



Input Capture 2 High Register (IC2HR)

Read Only
Reset value: Undefined

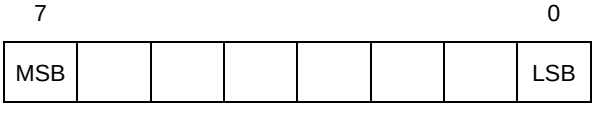
This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).



Input Capture 2 Low Register (IC2LR)

Read Only
Reset value: Undefined

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



16-bit timer (Cont'd)

Table 19. 16-bit timer register map and reset values

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

12.3 Serial communications interface (SCI)

12.3.1 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.

12.3.2 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

12.3.3 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

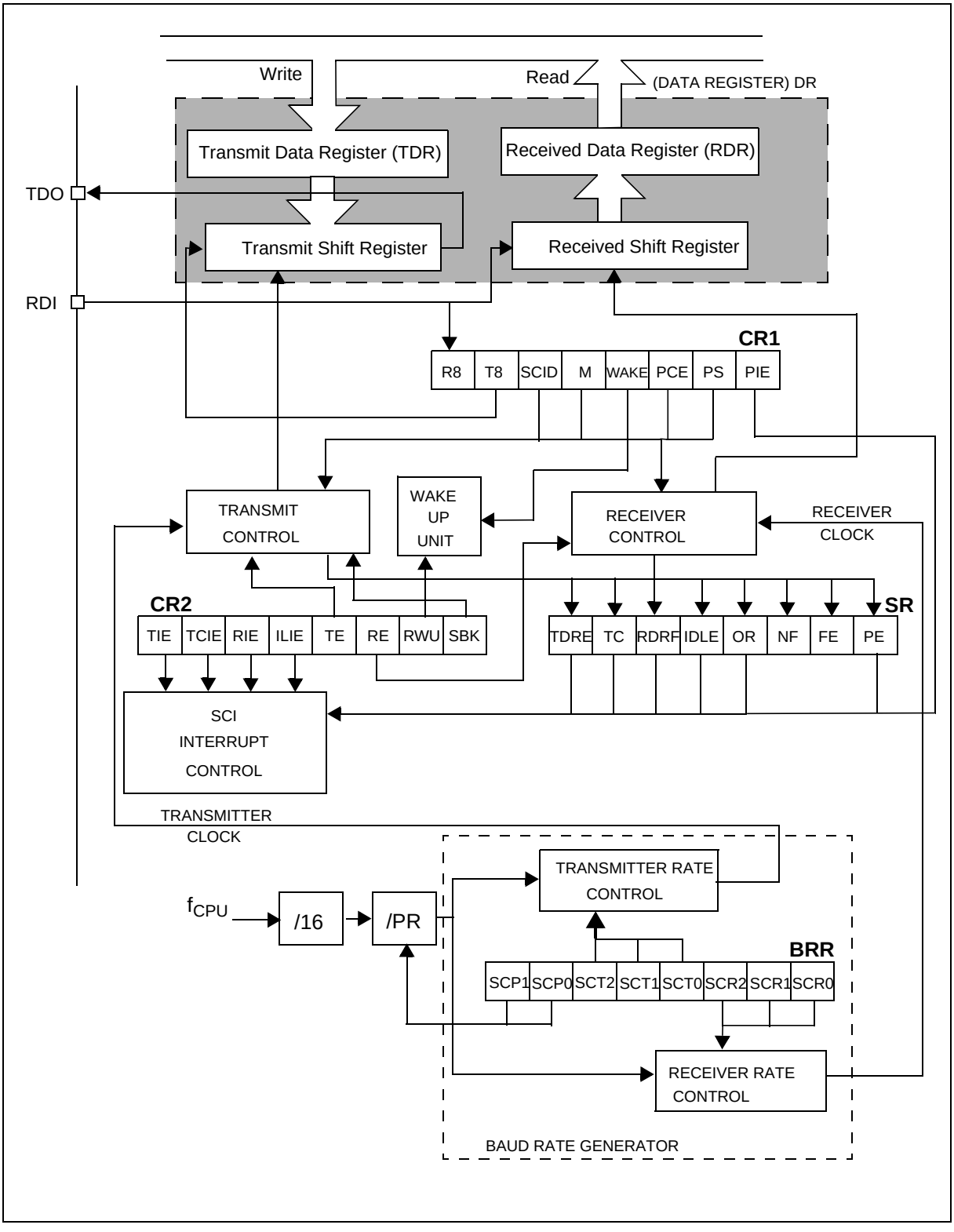
– A Stop bit indicating that the frame is complete.

This interface uses two types of baud rate generator:

– A conventional type for commonly-used baud rates.

Serial communications interface (Cont'd)

Figure 38. SCI block diagram



Serial communication interface (Cont'd)

12.3.4 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 12.3.7 for the definitions of each bit.

12.3.4.1 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.

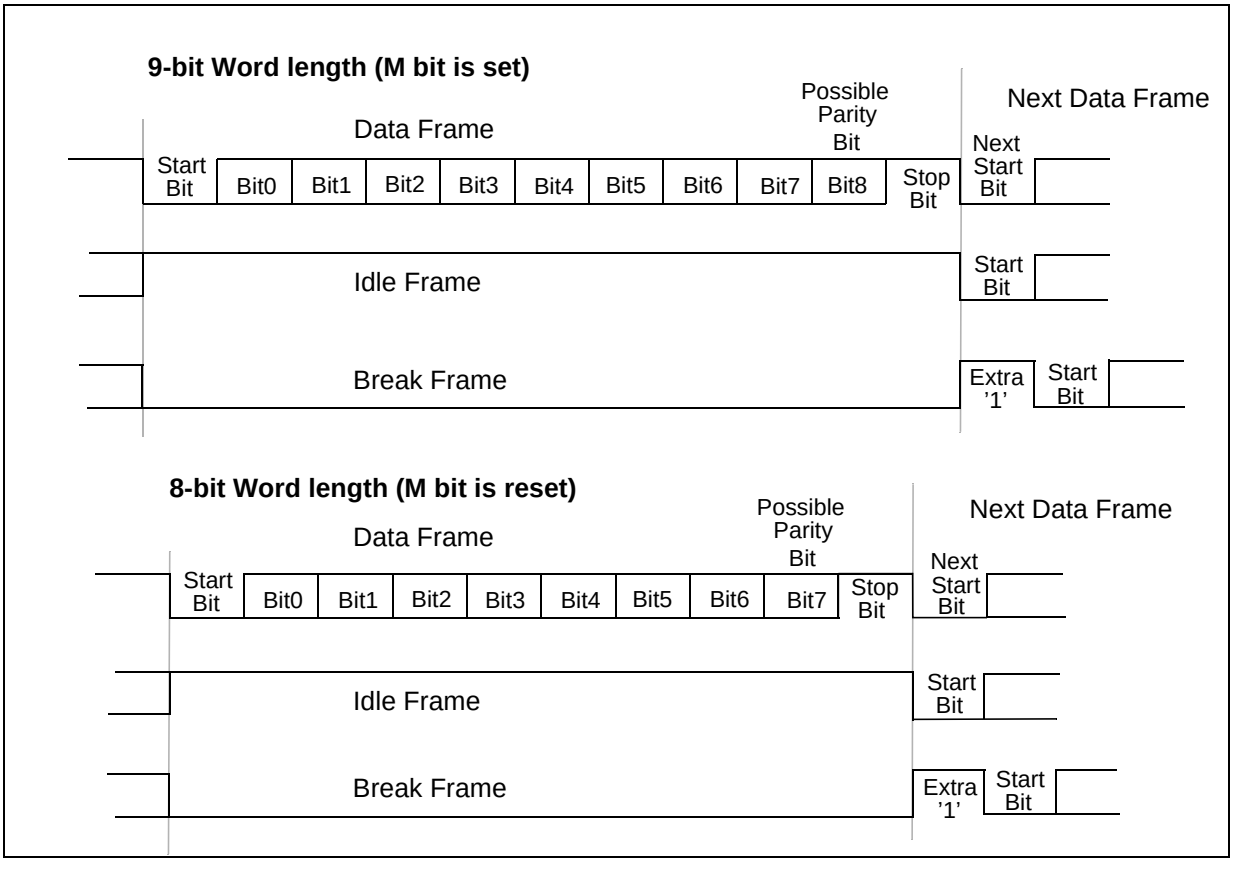
The TDO pin is in high state during the stop bit.

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



Serial communication interface (Cont'd)

12.3.4.2 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.

Serial communication interface (Cont'd)

12.3.4.3 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 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 characters

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

Idle characters

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 Section 12.3.4.9.

Serial communication interface (Cont'd)

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.

12.3.4.4 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.

12.3.4.5 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.

Serial communication interface (Cont'd)

12.3.4.6 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 20.

Table 20. Frame formats

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

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.

12.3.4.7 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 64µs), then the 8th, 9th and 10th samples will be at 28µs, 32µs & 36µs 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 ~4us. This means the entire bit length must be at least 40µs (36µs for the 10th sample + 4µs for synchronization with the internal sampling clock).

Serial communication interface (Cont'd)

12.3.4.8 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:

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

12.3.4.9 Noise error causes

See also description of Noise error in Section 12.3.4.3.

Start bit

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

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.

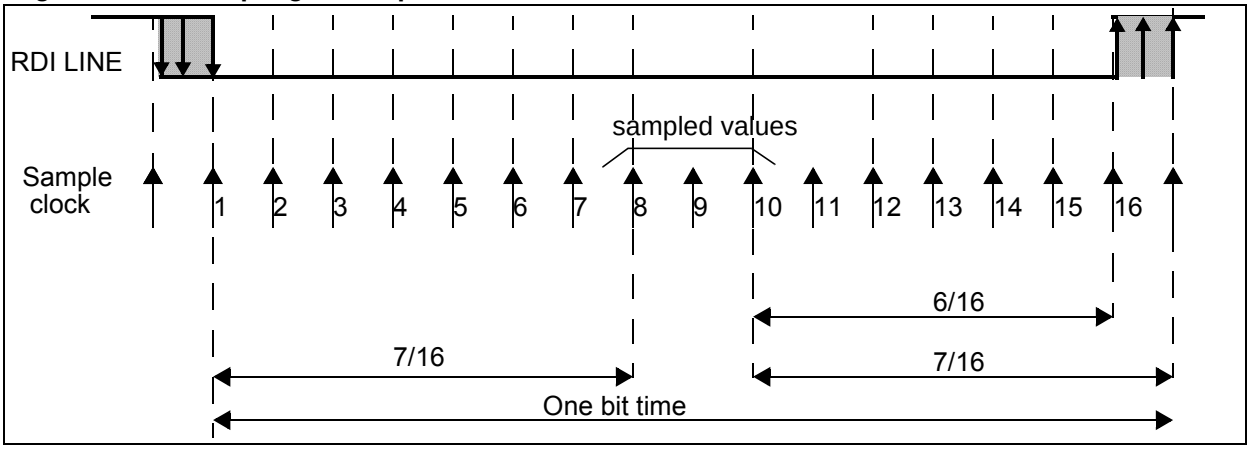
Data bits

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



Serial communication interface (Cont'd)

12.3.5 Low power modes

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.

12.3.6 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).

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

Serial communication interface (Cont'd)

12.3.7 Register description

Status Register (SCISR)
Read Only
Reset value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE

Bit 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.

Bit 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.

Bit 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

Bit 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 will not be set again until the RDRF bit has been set itself (i.e. a new idle line occurs).

Bit 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 will not be lost but the shift register will be overwritten.

Bit 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.

Bit 1 = **FE** *Framing error.*
This bit is set by hardware when a de-synchronization, 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 will be transferred and only the OR bit will be set.

Bit 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

Serial communication interface (Cont'd)

Control Register 1 (SCICR1)

Read/Write

Reset value: x000 0000 (x0h)

7							0
R8	T8	SCID	M	WAKE	PCE	PS	PIE

Bit 7 = **R8** *Receive data bit 8.*
This bit is used to store the 9th bit of the received word when M=1.

Bit 6 = **T8** *Transmit data bit 8.*
This bit is used to store the 9th bit of the transmitted word when M=1.

Bit 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

Bit 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).

Bit 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

Bit 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

Bit 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

Bit 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.

Serial communication interface (Cont'd)

Control Register 2 (SCICR2)

Read/Write

Reset value: 0000 0000 (00h)

7							0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

Bit 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

Bit 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

Bit 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

Bit 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.

Bit 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).

Bit 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

Bit 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

Note: Before selecting Mute mode (setting the RWU bit), the SCI must receive some data first, otherwise it cannot function in Mute mode with wakeup by idle line detection.

Bit 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.

Serial communication interface (Cont'd)

Data Register (SCIDR)

Read/Write

Reset value: Undefined

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

7							0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

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).

Baud Rate Register (SCIBRR)

Read/Write

Reset value: 0000 0000 (00h)

7							0
SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

Bits 7:6= **SCP[1:0]** *First SCI Prescaler*
These 2 prescaling bits allow several standard clock division ranges:

PR prescaling factor	SCP1	SCP0
1	0	0
3	0	1
4	1	0
13	1	1

Bits 5:3 = **SCT[2:0]** *SCI Transmitter rate divisor*
These 3 bits, in conjunction with the SCP1 & SCP0 bits define the total division applied to the bus clock to yield the transmit rate clock.

TR dividing factor	SCT2	SCT1	SCT0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

Bits 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.

RR dividing factor	SCR2	SCR1	SCR0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

Table 21. SCI register map and reset values

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

12.4 USB interface (USB)

12.4.1 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.

12.4.2 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

12.4.3 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.

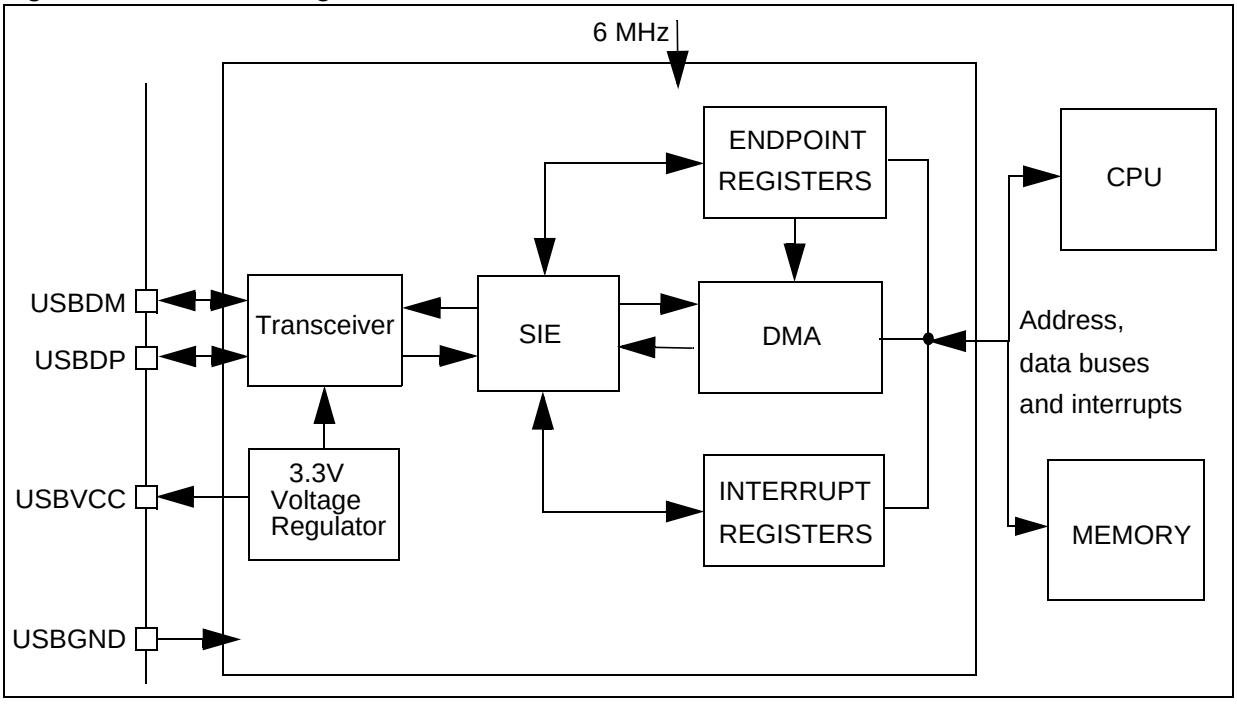
DMA

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



USB interface (Cont'd)

12.4.4 Register description

DMA address register (DMAR)

Read / Write

Reset value: Undefined

7							0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8

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.

Interrupt/DMA register (IDR)

Read / Write

Reset value: xxxx 0000 (x0h)

7							0
DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0

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.

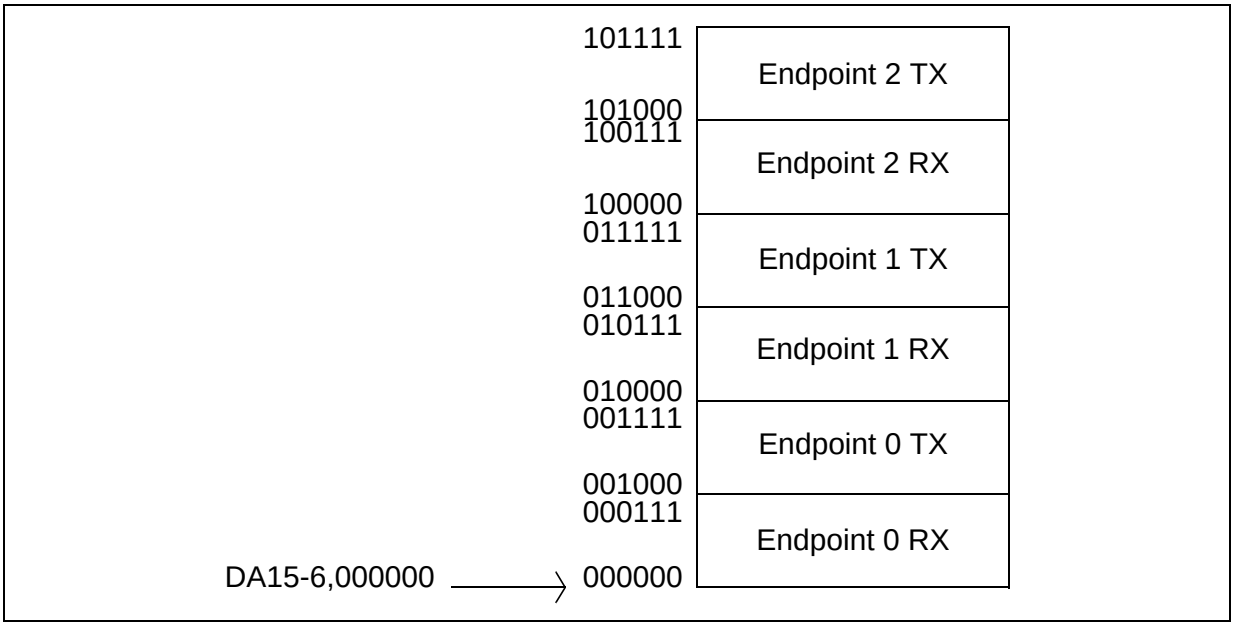
Bits 5:4 = **EP[1:0]** *Endpoint number* (read-only).
These bits identify the endpoint which required attention.
00: Endpoint 0
01: Endpoint 1
10: Endpoint 2

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



USB interface (Cont'd)

PID Register (PIDR)

Read only

Reset value: xx00 0000 (x0h)

7							0
TP3	TP2	0	0	0	RX_ SEZ	RXD	0

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:

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.

INTERRUPT Status Register (ISTR)

Read / Write

Reset value: 0000 0000 (00h)

7							0
SUSP	DOVR	CTR	ERR	IOVR	ESUSP	RESET	SOF

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

USB interface (Cont'd)

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..

Interrupt Mask Register (IMR)

Read / Write

Reset value: 0000 0000 (00h)

7							0
SUSPM	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM

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.

Control Register (CTLR)

Read / Write

Reset value: 0000 0110 (06h)

7							0
0	0	0	0	RESUME	PDWN	FSUSP	FRES

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.

USB interface (Cont'd)

Device Address Register (DADDR)

Read / Write

Reset value: 0000 0000 (00h)

7							0
0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

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.

Endpoint n Register A (EPnRA)

Read / Write

Reset value: 0000 xxxx (0xh)

7							0
ST_ OUT	DTOG _TX	STAT _TX1	STAT _TX0	TBC 3	TBC 2	TBC 1	TBC 0

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:

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).

Warning: Any value outside the range 0-8 will-induce undesired effects (such as continuous data transmission).

USB INTERFACE (Cont'd)

Endpoint n Register B (EPnRB)

Read / Write

Reset value: 0000 xxxx (0xh)

7							0
CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0

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:

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".

Endpoint 0 Register B (EP0RB)

Read / Write

Reset value: 1000 0000 (80h)

7							0
1	DTOG_RX	STAT_RX1	STAT_RX0	0	0	0	0

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.

USB interface (Cont'd)

12.4.5 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.

12.4.5.1 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.

12.4.5.2 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.

12.4.5.3 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.

12.4.5.4 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.

USB interface (Cont'd)

Table 22. USB register map and reset values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
25	PIDR	TP3	TP2	0	0	0	RX_SEZ	RXD	0
	Reset value	x	x	0	0	0	0	0	0
26	DMAR	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
	Reset value	x	x	x	x	x	x	x	x
27	IDR	DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0
	Reset value	x	x	x	x	0	0	0	0
28	ISTR	SUSP	DOVR	CTR	ERR	IOVR	ESUSP	RESET	SOF
	Reset value	0	0	0	0	0	0	0	0
29	IMR	SUSPM	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM
	Reset value	0	0	0	0	0	0	0	0
2A	CTLR	0	0	0	0	RESUME	PDWN	FSUSP	FRES
	Reset value	0	0	0	0	0	1	1	0
2B	DADDR	0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	Reset value	0	0	0	0	0	0	0	0
2C	EP0RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset value	0	0	0	0	x	x	x	x
2D	EP0RB	1	DTOG_RX	STAT_RX1	STAT_RX0	0	0	0	0
	Reset value	1	0	0	0	0	0	0	0
2E	EP1RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset value	0	0	0	0	x	x	x	x
2F	EP1RB	CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0
	Reset value	0	0	0	0	x	x	x	x
30	EP2RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset value	0	0	0	0	x	x	x	x
31	EP2RB	CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0
	Reset value	0	0	0	0	x	x	x	x

12.5 I²C bus interface (I²C)

12.5.1 Introduction

The I²C Bus Interface serves as an interface between the microcontroller and the serial I²C bus. It provides both multimaster and slave functions, and controls all I²C bus-specific sequencing, protocol, arbitration and timing. It supports fast I²C mode (400 kHz).

12.5.2 Main features

- Parallel-bus/I²C protocol converter
- Multi-master capability
- 7-bit Addressing
- Transmitter/Receiver flag
- End-of-byte transmission flag
- Transfer problem detection

I²C master features:

- Clock generation
- I²C bus busy flag
- Arbitration Lost flag
- End of byte transmission flag
- Transmitter/Receiver flag
- Start bit detection flag
- Start and Stop generation

I²C slave features:

- Stop bit detection
- I²C bus busy flag
- Detection of misplaced start or stop condition
- Programmable I²C Address detection
- Transfer problem detection
- End-of-byte transmission flag
- Transmitter/Receiver flag

12.5.3 General description

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa, using either an interrupt or polled handshake. The interrupts are enabled or disabled by software. The interface is connected to the I²C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I²C bus and a Fast I²C bus. This selection is made by software.

Mode selection

The interface can operate in the four following modes:

– Slave transmitter/receiver

– Master transmitter/receiver

By default, it operates in slave mode.

The interface automatically switches from slave to master after it generates a START condition and from master to slave in case of arbitration loss or a STOP generation, allowing then Multi-Master capability.

Communication flow

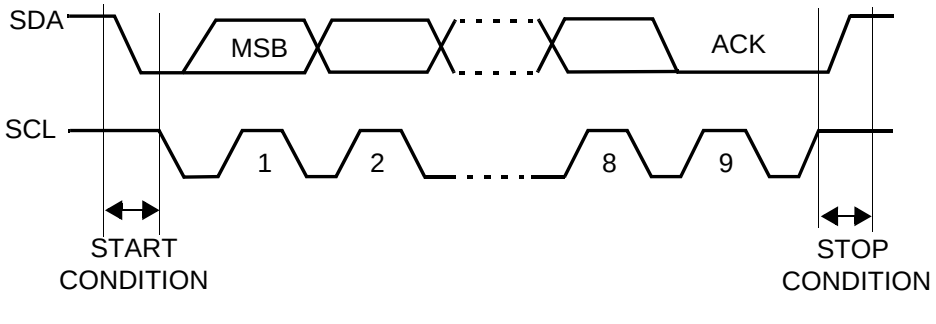
In Master mode, it initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated in master mode by software.

In Slave mode, the interface is capable of recognising its own address (7-bit), and the General Call address. The General Call address detection may be enabled or disabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte following the start condition is the address byte; it is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to Figure 43.

Figure 43. I²C bus protocol



I²C bus interface (Cont'd)

Acknowledge may be enabled and disabled by software.

The I²C interface address and/or general call address can be selected by software.

The speed of the I²C interface may be selected between Standard (up to 100kHz) and Fast I²C (up to 400kHz).

SDA/SCL Line Control

Transmitter mode: the interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the Data Register.

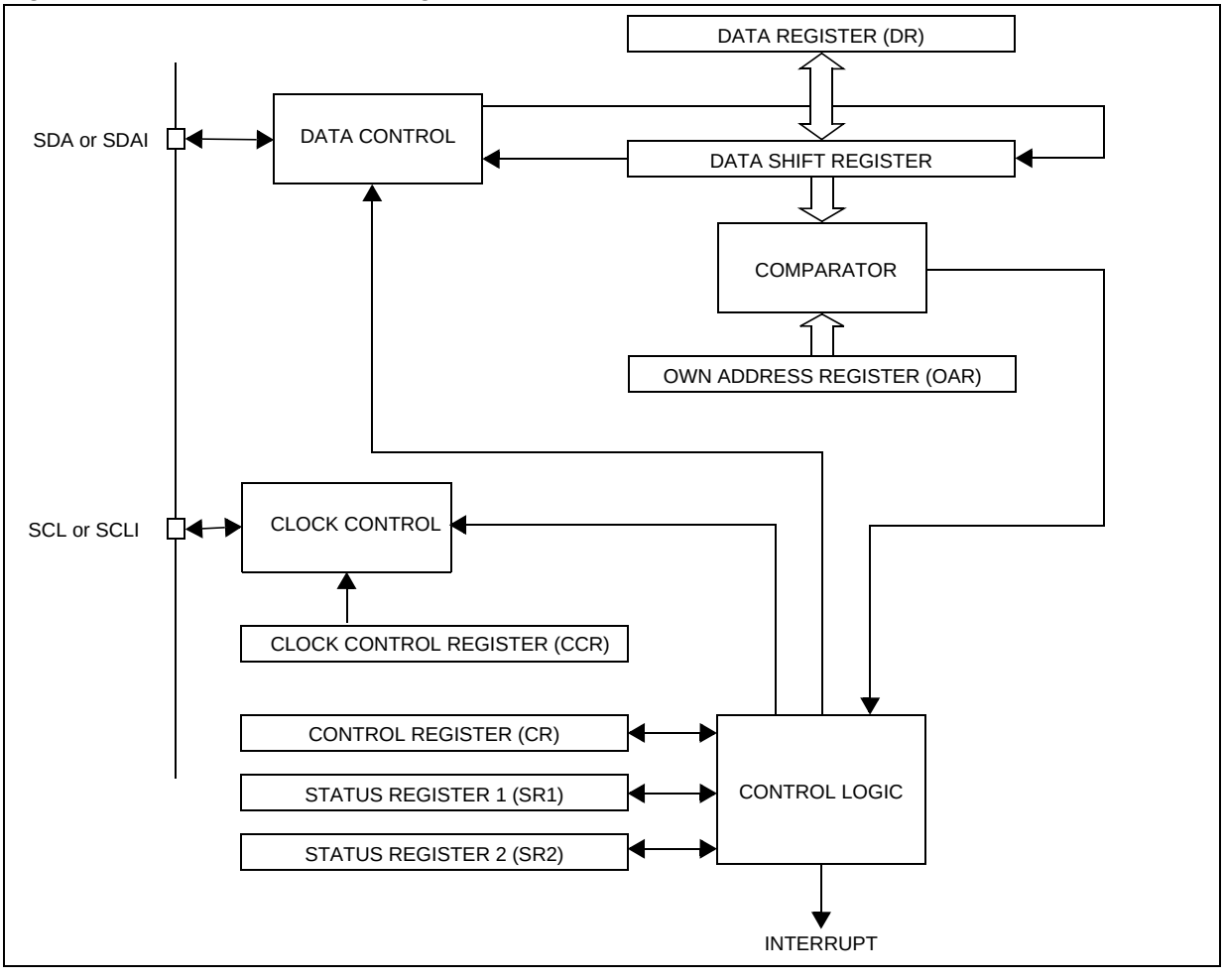
Receiver mode: the interface holds the clock line low after reception to wait for the microcontroller to read the byte in the Data Register.

The SCL frequency (F_{SCL}) is controlled by a programmable clock divider which depends on the I²C bus mode.

When the I²C cell is enabled, the SDA and SCL ports must be configured as floating inputs. In this case, the value of the external pull-up resistor used depends on the application.

When the I²C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 44. I²C Interface block diagram



I²C bus interface (Cont'd)

12.5.4 Functional description

Refer to the CR, SR1 and SR2 registers in Section 12.5.7. for the bit definitions.

By default the I²C interface operates in Slave mode (M/SL bit is cleared) except when it initiates a transmit or receive sequence.

12.5.4.1 Slave mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register; then it is compared with the address of the interface or the General Call address (if selected by software).

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the interface generates in sequence:

– Acknowledge pulse if the ACK bit is set.

– EVF and ADSL bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register, **holding the SCL line low** (see Figure 45 Transfer sequencing EV1).
Next, software must read the DR register to determine from the least significant bit (Data Direction Bit) if the slave must enter Receiver or Transmitter mode.

Slave receiver

Following the address reception and after SR1 register has been read, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

– Acknowledge pulse if the ACK bit is set

– EVF and BTF bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see Figure 45 Transfer sequencing EV2).

Slave transmitter

Following the address reception and after the SR1 register has been read, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 45 Transfer sequencing EV3).

When the acknowledge pulse is received:

– The EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Closing slave communication

After the last data byte is transferred a Stop Condition is generated by the master. The interface detects this condition and sets:

– EVF and STOPF bits with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR2 register (see Figure 45 Transfer sequencing EV4).

Error cases

– **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and the BERR bits are set with an interrupt if the ITE bit is set.
If it is a Stop, then the interface discards the data, released the lines and waits for another Start condition.
If it is a Start, then the interface discards the data and waits for the next slave address on the bus.

– **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set with an interrupt if the ITE bit is set.
The AF bit is cleared by reading the I2CSR2 register. However, if read before the completion of the transmission, the AF flag will be set again, thus possibly generating a new interrupt. Software must ensure either that the SCL line is back at 0 before reading the SR2 register, or be able to correctly handle a second interrupt during the 9th pulse of a transmitted byte.

Note: In case of errors, SCL line is not held low; however, the SDA line can remain low if the last bits transmitted are all 0. While AF=1, the SCL line may be held low due to SB or BTF flags that are set at the same time. It is then necessary to release both lines by software.

How to release the SDA / SCL lines

Set and subsequently clear the STOP bit while BTF is set. The SDA/SCL lines are released after the transfer of the current byte.

I²C bus interface (Cont'd)

12.5.4.2 Master mode

To switch from default Slave mode to Master mode, a Start condition generation is needed.

Start condition

Setting the START bit while the BUSY bit is cleared causes the interface to switch to Master mode (M/SL bit set) and generates a Start condition.

Once the Start condition is sent:

– The EVF and SB bits are set by hardware with an interrupt if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address byte, **holding the SCL line low** (see Figure 45 Transfer sequencing EV5).

Slave address transmission

Then the slave address byte is sent to the SDA line via the internal shift register.

After completion of this transfer (and acknowledge from the slave if the ACK bit is set):

– The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see Figure 45 Transfer sequencing EV6).

Next the master must enter Receiver or Transmitter mode.

Master receiver

Following the address transmission and after the SR1 and CR registers have been accessed, the master receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

– Acknowledge pulse if the ACK bit is set

– EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see Figure 45 Transfer sequencing EV7).

To close the communication: before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

Master transmitter

Following the address transmission and after SR1 register has been read, the master sends bytes from the DR register to the SDA line via the internal shift register.

The master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 45 Transfer sequencing EV8).

When the acknowledge bit is received, the interface sets:

– EVF and BTF bits with an interrupt if the ITE bit is set.

To close the communication: after writing the last byte to the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Error cases

– **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and BERR bits are set by hardware with an interrupt if ITE is set.
Note that BERR will not be set if an error is detected during the first or second pulse of each 9-bit transaction:
Single Master mode
If a Start or Stop is issued during the first or second pulse of a 9-bit transaction, the BERR flag will not be set and transfer will continue however the BUSY flag will be reset. To work around this, slave devices should issue a NACK when they receive a misplaced Start or Stop. The reception of a NACK or BUSY by the master in the middle of communication gives the possibility to reinitiate transmission.
Multimaster mode
Normally the BERR bit would be set whenever unauthorized transmission takes place while transfer is already in progress. However, an issue will arise if an external master generates an unauthorized Start or Stop while the I^2C master is on the first or second pulse of a 9-bit transaction. It is possible to work around this by polling the BUSY bit during I^2C master mode transmis-

sion. The resetting of the BUSY bit can then be handled in a similar manner as the BERR flag being set.

– **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the START or STOP bit.
The AF bit is cleared by reading the I2CSR2 register. However, if read before the completion of the transmission, the AF flag will be set again, thus possibly generating a new interrupt. Software must ensure either that the SCL line is back at 0 before reading the SR2 register, or be able to correctly handle a second interrupt during the 9th pulse of a transmitted byte.

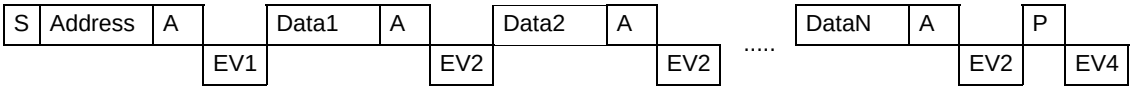
– **ARLO:** Detection of an arbitration lost condition. In this case the ARLO bit is set by hardware (with an interrupt if the ITE bit is set and the interface goes automatically back to slave mode (the M/SL bit is cleared).

Note: In all these cases, the SCL line is not held low; however, the SDA line can remain low if the last bits transmitted are all 0. While AF=1, the SCL line may be held low due to SB or BTF flags that are set at the same time. It is then necessary to release both lines by software.

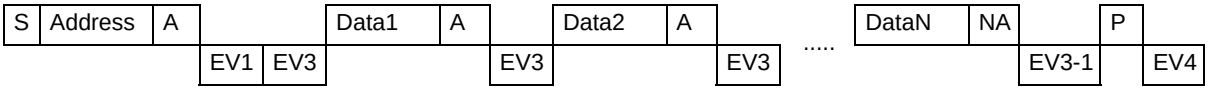
I²C bus interface (Cont'd)

Figure 45. Transfer sequencing

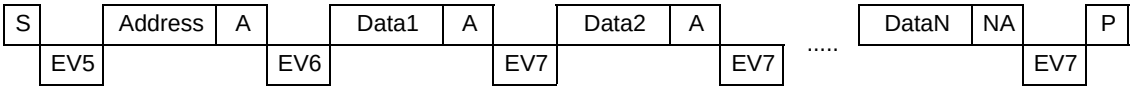
Slave Receiver



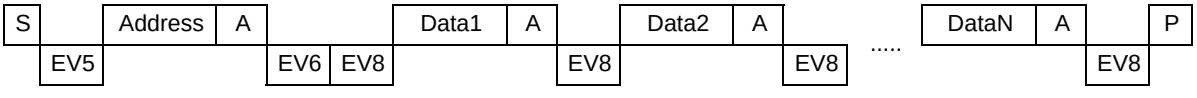
Slave Transmitter



Master Receiver



Master Transmitter



Legend:

S=Start, P=Stop, A=Acknowledge, NA=Non-acknowledge

EVx=Event (with interrupt if ITE=1)

EV1: EVF=1, ADSL=1, cleared by reading the SR1 register.

EV2: EVF=1, BTF=1, cleared by reading the SR1 register followed by reading the DR register.

EV3: EVF=1, BTF=1, cleared by reading the SR1 register followed by writing the DR register.

EV3-1: EVF=1, AF=1, BTF=1; AF is cleared by reading the SR1 register. The BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing the DR register (DR=FFh).
Note: If lines are released by STOP=1, STOP=0, the subsequent EV4 is not seen.

EV4: EVF=1, STOPF=1, cleared by reading the SR2 register.

EV5: EVF=1, SB=1, cleared by reading the SR1 register followed by writing the DR register.

EV6: EVF=1, cleared by reading the SR1 register followed by writing the CR register (for example PE=1).

EV7: EVF=1, BTF=1, cleared by reading the SR1 register followed by reading the DR register.

EV8: EVF=1, BTF=1, cleared by reading the SR1 register followed by writing the DR register.

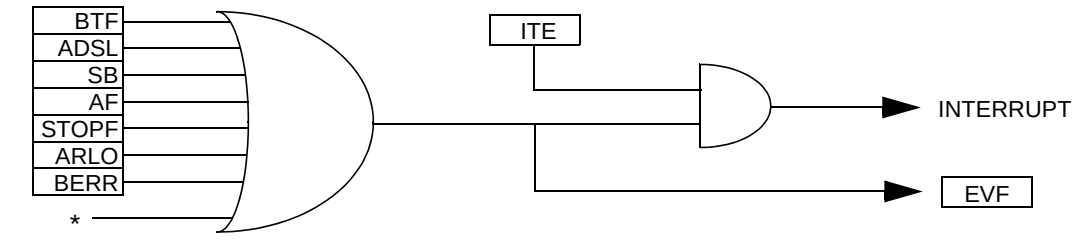
I²C bus interface (Cont'd)

12.5.5 Low power modes

Mode	Description
Wait	No effect on I²C interface. I²C interrupts cause the device to exit from Wait mode.
Halt	I²C registers are frozen. In Halt mode, the I²C interface is inactive and does not acknowledge data on the bus. The I²C interface resumes operation when the MCU is woken up by an interrupt with "exit from Halt mode" capability.

12.5.6 Interrupts

Figure 46. Event flags and interrupt generation



* EVF can also be set by EV6 or an error from the SR2 register.

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
End of byte Transfer event	BTF	ITE	Yes	No
Address Matched event (Slave mode)	ADSL		Yes	No
Start Bit Generation event (Master mode)	SB		Yes	No
Acknowledge Failure event	AF		Yes	No
Stop Detection event (Slave mode)	STOPF		Yes	No
Arbitration Lost event (Multimaster configuration)	ARLO		Yes	No
Bus Error event	BERR		Yes	No

The I²C interrupt events are connected to the same interrupt vector (see Interrupts chapter).

They generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

I²C bus interface (Cont'd)

12.5.7 Register description

I²C Control Register (CR)
Read / Write
Reset value: 0000 0000 (00h)

7							0
0	0	PE	ENGC	START	ACK	STOP	ITE

Bits 7:6 = Reserved. Forced to 0 by hardware.

Bit 5 = **PE** *Peripheral enable.*
This bit is set and cleared by software.
0: Peripheral disabled
1: Master/Slave capability
Notes:
– When PE=0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released while PE=0
– When PE=1, the corresponding I/O pins are selected by hardware as alternate functions.
– To enable the I²C interface, write the CR register **TWICE** with PE=1 as the first write only activates the interface (only PE is set).

Bit 4 = **ENGC** *Enable General Call.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0). The 00h General Call address is acknowledged (01h ignored).
0: General Call disabled
1: General Call enabled

Note: In accordance with the I2C standard, when GCAL addressing is enabled, an I2C slave can only receive data. It will not transmit data to the master.

Bit 3 = **START** *Generation of a Start condition*. This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0) or when the Start condition is sent (with interrupt generation if ITE=1).

– In master mode:
0: No start generation
1: Repeated start generation

– In slave mode:
0: No start generation
1: Start generation when the bus is free

Bit 2 = **ACK** *Acknowledge enable.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0).
0: No acknowledge returned
1: Acknowledge returned after an address byte or a data byte is received

Bit 1 = **STOP** *Generation of a Stop condition*.
This bit is set and cleared by software. It is also cleared by hardware in master mode. Note: This bit is not cleared when the interface is disabled (PE=0).

– In Master mode:
0: No stop generation
1: Stop generation after the current byte transfer or after the current Start condition is sent. The STOP bit is cleared by hardware when the Stop condition is sent.

– In Slave mode:
0: No stop generation
1: Release the SCL and SDA lines after the current byte transfer (BTF=1). In this mode the STOP bit has to be cleared by software.

Bit 0 = **ITE** *Interrupt enable.*
This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE=0).
0: Interrupts disabled
1: Interrupts enabled

Refer to Figure 46 for the relationship between the events and the interrupt.

SCL is held low when the SB, BTF or ADSL flags or an EV6 event (See Figure 45) is detected.

I²C bus interface (Cont'd)

I²C Status Register 1 (SR1)

Read Only

Reset value: 0000 0000 (00h)

7							0
EVF	0	TRA	BUSY	BTF	ADSL	M/SL	SB

Bit 7 = **EVF** *Event flag.*
This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in Figure 45. It is also cleared by hardware when the interface is disabled (PE=0).
0: No event
1: One of the following events has occurred:

- BTF=1 (byte received or transmitted)
- ADSL=1 (Address matched in Slave mode while ACK=1)
- SB=1 (Start condition generated in Master mode)
- AF=1 (No acknowledge received after byte transmission)
- STOPF=1 (Stop condition detected in Slave mode)
- ARLO=1 (Arbitration lost in Master mode)
- BERR=1 (Bus error, misplaced Start or Stop condition detected)
- Address byte successfully transmitted in Master mode.

Bit 6 = Reserved. Forced to 0 by hardware.

Bit 5 = **TRA** *Transmitter/Receiver.*
When BTF is set, TRA=1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware after detection of Stop condition (STOPF=1), loss of bus arbitration (ARLO=1) or when the interface is disabled (PE=0).
0: Data byte received (if BTF=1)
1: Data byte transmitted

Bit 4 = **BUSY** *Bus busy*.
This bit is set by hardware on detection of a Start condition and cleared by hardware on detection of a Stop condition. It indicates a communication in progress on the bus. The BUSY flag of the I2CSR1 register is cleared if a Bus Error occurs.
0: No communication on the bus
1: Communication ongoing on the bus

Note:
– The BUSY flag is NOT updated when the interface is disabled (PE=0). This can have consequences when operating in Multimaster mode; i.e. a second active I^2C master commencing a transfer with an unset BUSY bit can cause a conflict resulting in lost data. A software workaround consists of checking that the I^2C is not busy before enabling the I^2C Multimaster cell.

Bit 3 = **BTF** *Byte transfer finished.*
This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE=1. It is cleared by software reading SR1 register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE=0).

- Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV6 event (See Figure 45). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.
- Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

0: byte transfer not done
1: byte transfer succeeded

Bit 2 = **ADSL** *Address matched (Slave mode).*
This bit is set by hardware as soon as the received slave address matched with the OAR register content or a general call is recognized. An interrupt is generated if ITE=1. It is cleared by software reading SR1 register or by hardware when the interface is disabled (PE=0).

The SCL line is held low while ADSL=1.

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

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

Bit 0 = **SB** *Start bit (Master mode).*
This bit is set by hardware as soon as the Start condition is generated (following a write START=1). An interrupt is generated if ITE=1. It is cleared by software reading SR1 register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE=0).
0: No Start condition
1: Start condition generated

I²C Status Register 2 (SR2)
Read Only
Reset value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	ARLO	BERR	GCAL

Bits 7:5 = Reserved. Forced to 0 by hardware.

Bit 4 = **AF** *Acknowledge failure*.
This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

Note: While AF=1, the SCL line may be held low due to SB or BTF flags that are set at the same time. It is then necessary to release both lines by software.

0: No acknowledge failure
1: Acknowledge failure

Bit 3 = **STOPF** *Stop detection (Slave mode).*
This bit is set by hardware when a Stop condition is detected on the bus after an acknowledge (if ACK=1). An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while STOPF=1.

0: No Stop condition detected
1: Stop condition detected

Bit 2 = **ARLO** *Arbitration lost*.
This bit is set by hardware when the interface loses the arbitration of the bus to another master. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

After an ARLO event the interface switches back automatically to Slave mode (M/SL=0).

The SCL line is not held low while ARLO=1.

0: No arbitration lost detected
1: Arbitration lost detected
Note:
– In a Multimaster environment, when the interface is configured in Master Receive mode it does not perform arbitration during the reception of the Acknowledge Bit. Mishandling of the ARLO bit from the I2CSR2 register may occur when a second master simultaneously requests the same data from the same slave and the I^2C master does not acknowledge the data. The ARLO bit is then left at 0 instead of being set.

Bit 1 = **BERR** *Bus error.*
This bit is set by hardware when the interface detects a misplaced Start or Stop condition. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while BERR=1.

0: No misplaced Start or Stop condition
1: Misplaced Start or Stop condition
Note:
– If a Bus Error occurs, a Stop or a repeated Start condition should be generated by the Master to re-synchronize communication, get the transmission acknowledged and the bus released for further communication

Bit 0 = **GCAL** *General Call (Slave mode).*
This bit is set by hardware when a general call address is detected on the bus while ENGC=1. It is cleared by hardware detecting a Stop condition (STOPF=1) or when the interface is disabled (PE=0).

0: No general call address detected on bus
1: general call address detected on bus

I²C bus interface (Cont'd)

I²C Clock Control Register (CCR)

Read / Write
Reset value: 0000 0000 (00h)

7							0
FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0

Bit 7 = **FM/SM** *Fast/Standard I²C mode.*
This bit is set and cleared by software. It is not cleared when the interface is disabled (PE=0).
0: Standard I²C mode
1: Fast I²C mode

Bits 6:0 = **CC[6:0]** *7-bit clock divider.*
These bits select the speed of the bus (F_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE=0).

Refer to the Electrical Characteristics section for the table of value.

Note: The programmed F_{SCL} assumes no load on SCL and SDA lines.

I²C Data Register (DR)

Read / Write
Reset value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bits 7:0 = **D[7:0]** *8-bit Data Register.*
These bits contain the byte to be received or transmitted on the bus.

– Transmitter mode: byte transmission start automatically when the software writes in the DR register.

– Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address.
Then, the following data bytes are received one by one after reading the DR register.

I²C Own Address Register (OAR)

Read / Write
Reset value: 0000 0000 (00h)

7							0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

Bits 7:1 = **ADD[7:1]** *Interface address*.
These bits define the I²C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

Bit 0 = **ADD0** *Address direction bit.*
This bit is don't care, the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE=0).

Note: Address 01h is always ignored.

Table 23. I²C register map

Address (Hex.)	Register name	7	6	5	4	3	2	1	0
39	**DR**	DR7 .. DR0							
3B	**OAR**	ADD7 .. ADD0							
3C	**CCR**	FM/SM	CC6 .. CC0						
3D	**SR2**				AF	STOPF	ARLO	BERR	GCAL
3E	**SR1**	EVF		TRA	BUSY	BTF	ADSL	M/SL	SB
3F	**CR**			PE	ENGC	START	ACK	STOP	ITE

Note: Refer to Section 17, "Known limitations" for information regarding a limitation on the alternate function on pin PA2 (SCL).

12.6 8-bit A/D converter (ADC)

12.6.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 8-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

The result of the conversion is stored in a 8-bit Data Register. The A/D converter is controlled through a Control/Status Register.

12.6.2 Main features

- 8-bit conversion
- Up to 12 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in Figure 47.

12.6.3 Functional description

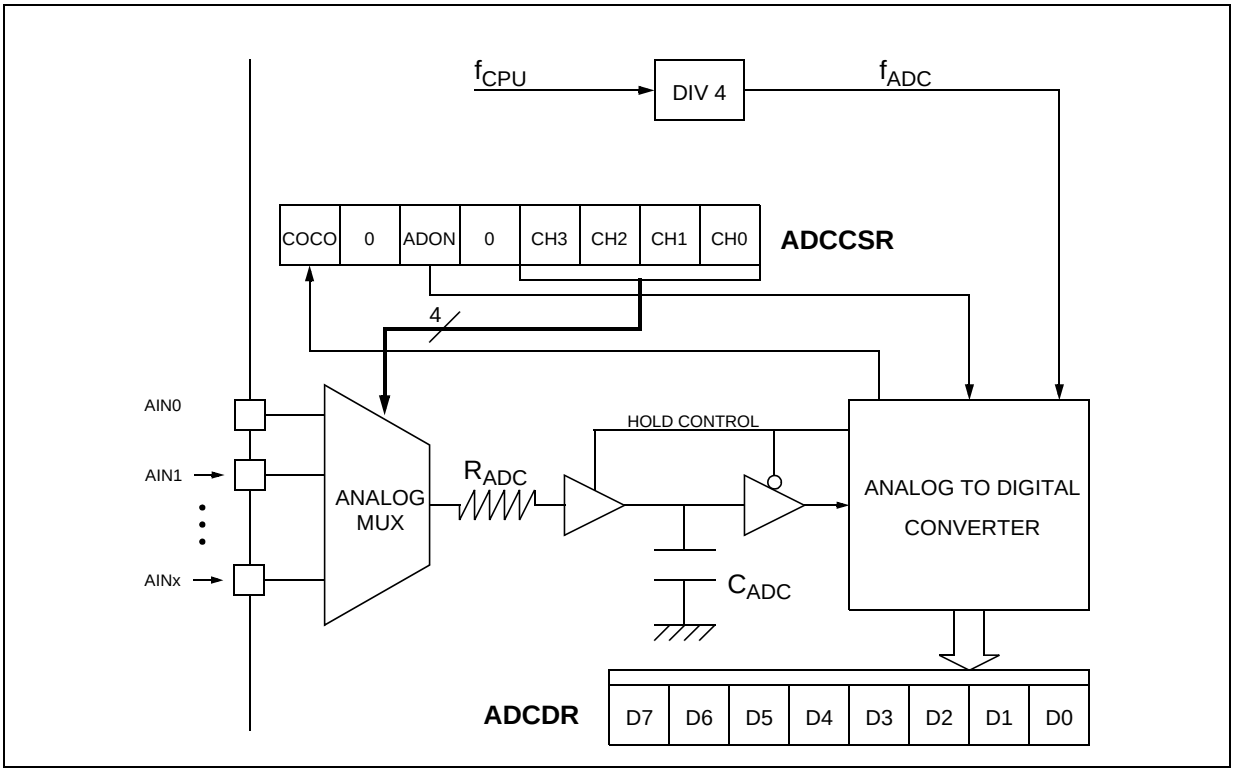
12.6.3.1 Analog power supply

V_{DDA} and V_{SSA} are the high and low level reference voltage pins. In some devices (refer to device pin out description) they are internally connected to the V_{DD} and V_{SS} pins.

Conversion accuracy may therefore be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

See electrical characteristics section for more details.

Figure 47. ADC block diagram



8-bit A/D converter (ADC) (Cont'd)

12.6.3.2 Digital A/D conversion result

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than or equal to V_{DDA} (high-level voltage reference) then the conversion result in the DR register is FFh (full scale) without overflow indication.

If input voltage (V_{AIN}) is lower than or equal to V_{SSA} (low-level voltage reference) then the conversion result in the DR register is 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDR register. The accuracy of the conversion is described in the parametric section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

12.6.3.3 A/D conversion phases

The A/D conversion is based on two conversion phases as shown in Figure 48:

- Sample capacitor loading [duration: t_{LOAD}]
  During this phase, the V_{AIN} input voltage to be measured is loaded into the C_{ADC} sample capacitor.
- A/D conversion [duration: t_{CONV}]
  During this phase, the A/D conversion is computed (8 successive approximations cycles) and the C_{ADC} sample capacitor is disconnected from the analog input pin to get the optimum analog to digital conversion accuracy.

While the ADC is on, these two phases are continuously repeated.

At the end of each conversion, the sample capacitor is kept loaded with the previous measurement load. The advantage of this behaviour is that it minimizes the current consumption on the analog pin in case of single input channel measurement.

12.6.3.4 Software Procedure

Refer to the control/status register (CSR) and data register (DR) in Section 12.6.6 for the bit definitions and to Figure 48 for the timings.

ADC configuration

The total duration of the A/D conversion is 12 ADC clock periods ($1/f_{ADC}=4/f_{CPU}$).

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the «I/O ports» chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the CSR register:

– Select the CH[3:0] bits to assign the analog channel to be converted.

ADC conversion

In the CSR register:

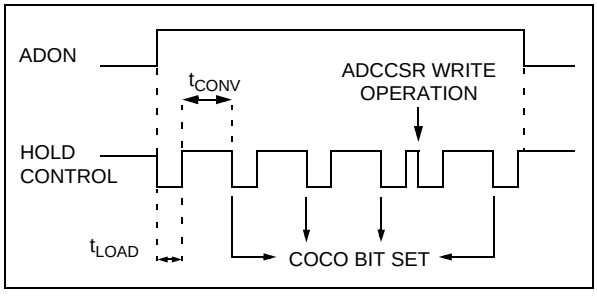
– Set the ADON bit to enable the A/D converter and to start the first conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete

– The COCO bit is set by hardware.
– No interrupt is generated.
– The result is in the DR register and remains valid until the next conversion has ended.

A write to the CSR register (with ADON set) aborts the current conversion, resets the COCO bit and starts a new conversion.

Figure 48. ADC conversion timings



12.6.4 Low power modes

Mode	Description
Wait	No effect on A/D Converter
Halt	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilisation time before accurate conversions can be performed.

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

12.6.5 Interrupts

None

8-bit A/D converter (ADC) (Cont'd)

12.6.6 Register description

Control/Status Register (CSR)

Read/Write

Reset value: 0000 0000 (00h)

7							0
COCO	0	ADON	0	CH3	CH2	CH1	CH0

Bit 7 = **COCO** *Conversion Complete*
This bit is set by hardware. It is cleared by software reading the result in the DR register or writing to the CSR register.
0: Conversion is not complete
1: Conversion can be read from the DR register

Bit 6 = **Reserved.** *must always be cleared.*

Bit 5 = **ADON** *A/D Converter On*
This bit is set and cleared by software.
0: A/D converter is switched off
1: A/D converter is switched on

Bit 4 = **Reserved.** *must always be cleared.*

Bits 3:0 = **CH[3:0]** *Channel Selection*
These bits are set and cleared by software. They select the analog input to convert.

Channel Pin[1]	CH3[2]	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
AIN6	0	1	1	0
AIN7	0	1	1	1
AIN8	1	0	0	0
AIN9	1	0	0	1
AIN10	1	0	1	0
AIN11	1	0	1	1

1 The number of pins AND the channel selection varies according to the device. Refer to the device pinout.
2 For SDIP/SO34 devices, the CH3 bit is always at '0'. If, however, set to '1' on error, channel (11:8) becomes enabled which may result in a higher and unnecessary level of consumption.

Data Register (DR)

Read Only

Reset value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bits 7:0 = **D[7:0]** *Analog Converted value*
This register contains the converted analog value in the range 00h to FFh.

Note: Reading this register reset the COCO flag.

8-bit A/D converter (ADC) (Cont'd)

Table 24. ADC register map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0Ah	DR	AD7 .. AD0							
0Bh	CSR	COCO	0	ADON	0	CH3	CH2	CH1	CH0

13 Instruction set

13.1 ST7 addressing modes

The ST7 Core features 17 different addressing modes which can be classified in 7 main 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 25. ST7 addressing mode overview

Mode			Syntax	Destination/ source	Pointer address (Hex.)	Pointer size (Hex.)	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
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

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

ST7 addressing modes (Cont'd)

13.1.1 Inherent

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

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

13.1.2 Immediate

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

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

13.1.3 Direct

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

The direct addressing mode consists of two submodes:

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.

13.1.4 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.

13.1.5 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.

ST7 addressing modes (Cont'd)

13.1.6 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 26. Instructions supporting Direct, Indexed, Indirect and Indirect Indexed addressing modes

Long and Short instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Addition/subtraction operations
BCP	Bit Compare

Short instructions only	Function
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

13.1.7 Relative mode (Direct, Indirect)

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

Available Relative Direct/ Indirect instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset follows the opcode.

Relative (Indirect)

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

13.2 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:

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.

Instruction groups (Cont'd)

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 >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

Instruction groups(Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
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	

14 Electrical characteristics

14.1 Parameter conditions

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

14.1.1 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=$T_A$max (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±3Σ).

14.1.2 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.

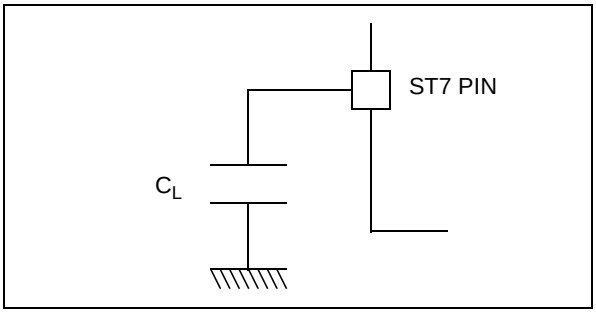
14.1.3 Typical curves

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

14.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in Figure 49.

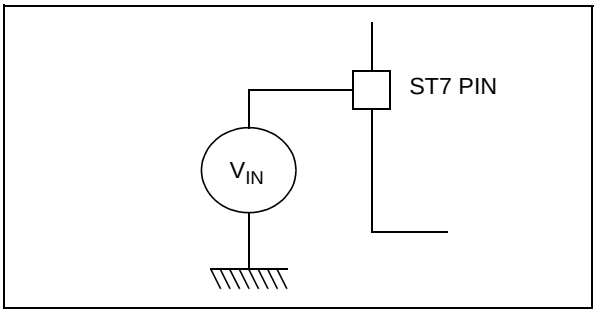
Figure 49. Pin loading conditions



14.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in Figure 50.

Figure 50. Pin input voltage



14.2 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.

14.2.1 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 116.	

14.2.2 Current characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) 3)	80	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) 3)	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) & 4)	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 5) & 6)	± 5	
$\Sigma I_{INJ(PIN)}$ 2)	Total injected current (sum of all I/O and control pins) 5)	± 20	
$I_{INJ(PIN)}$ 2) & 4)	Negative injected current to PB0(10mA)/AIN0 pin	- 80	µA

Notes:

1. Directly connecting the $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$, 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.

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. All power (V_{DD}) and ground (V_{SS}) lines must always be connected to the external supply.

4. Negative injection disturbs the analog performance of the device. In particular, it induces leakage currents throughout the device including the analog inputs. To avoid undesirable effects on the analog functions, care must be taken:
- Analog input pins must have a negative injection less than 0.8 mA (assuming that the impedance of the analog voltage is lower than the specified limits)
- Pure digital pins must have a negative injection less than 1.6mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.

5. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterization with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

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

14.2.3 Thermal characteristics

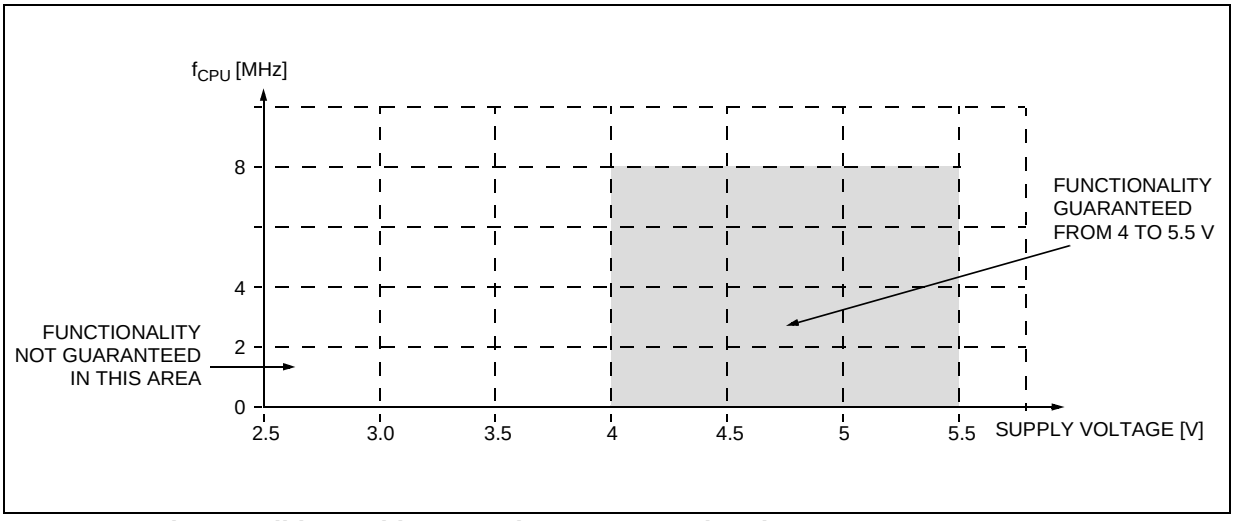
Symbol	Ratings	value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature: See section 15.2 on page 131 for T_{Jmax}		

14.3 Operating conditions

14.3.1 General operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating Supply Voltage	f_{CPU} = 8 MHz	4	5	5.5	V
V_{DDA}	Analog reference voltage		V_{DD}		V_{DD}	
V_{SSA}	Analog reference 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 51. f_{CPU} maximum operating frequency versus V_{DD} supply voltage



14.3.2 Operating conditions with Low Voltage Detector (LVD)

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A. Refer to Figure 12 on page 24.

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

Notes:

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

14.4 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).

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

Notes:

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

Figure 52. Typ. I_{DD} in Run at 4 and 8 MHz f_{CPU}

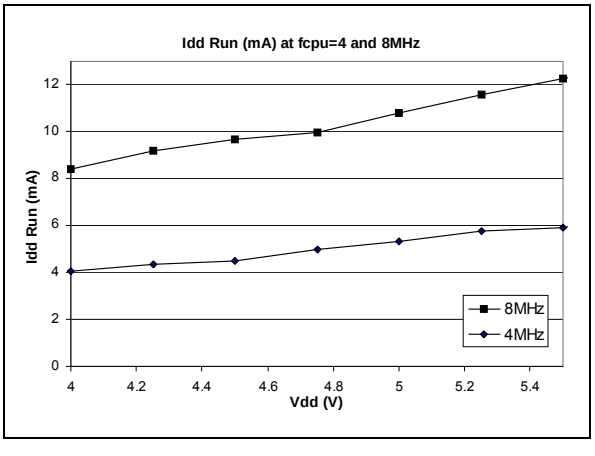
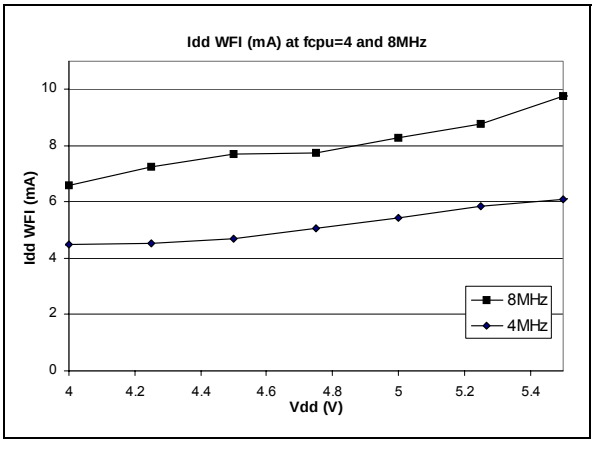


Figure 53. Typ. I_{DD} in Wait at 4 and 8 MHz f_{CPU}



14.5 Clock and timing characteristics

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

14.5.1 General timings

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
$t_{c(INST)}$	Instruction cycle time	f_{CPU}=8MHz	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}=8MHz	10		22	t_{CPU}
			1.25		2.75	µs

Notes:

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.

14.5.2 Control timing characteristics

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
t_{DOG}	Watchdog Time-out	 f_{cpu} = 8MHz	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 4V			100 [1]	ms

Note:

1. Not tested in production, guaranteed by characterization.

Clock and timing characteristics (Cont'd)

14.5.3 External clock source

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSCINH}	OSCIN input pin high level voltage	see Figure 54	$0.7xV_{DD}$		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 time1)				15	
I_L	OSCx Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			±1	µA

Note:

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

Figure 54. Typical application with an external clock source

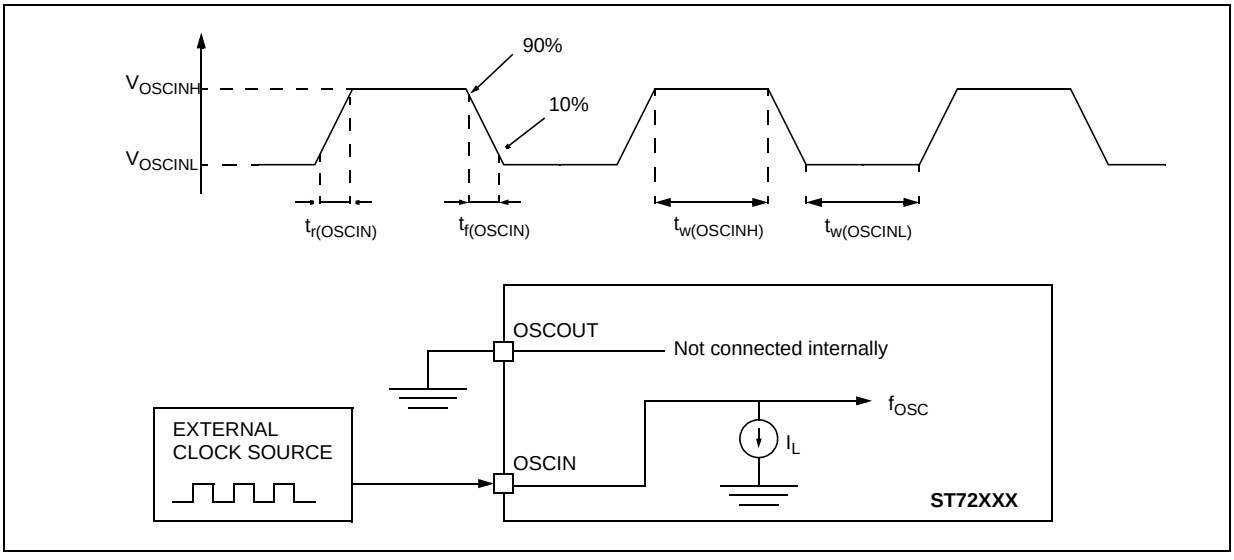
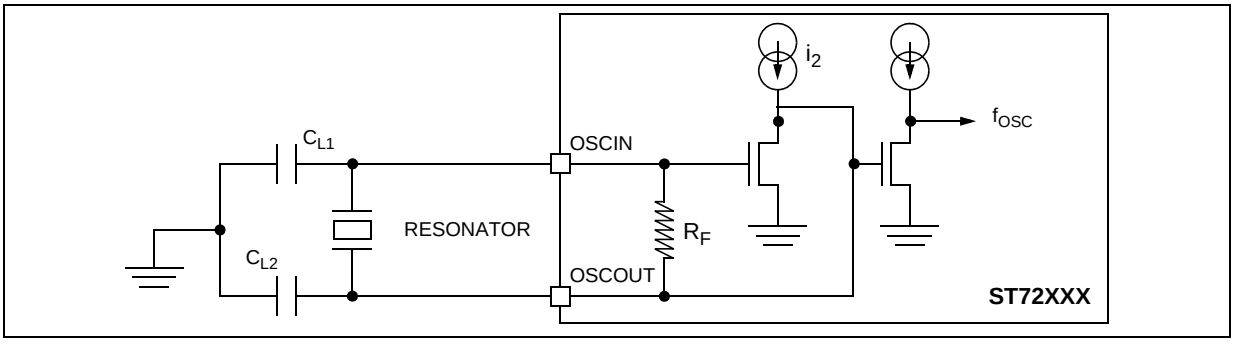


Figure 55. Typical application with a crystal resonator



14.6 Memory characteristics

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

14.6.1 RAM and hardware registers

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

Note:
1. Guaranteed by design. Not tested in production.

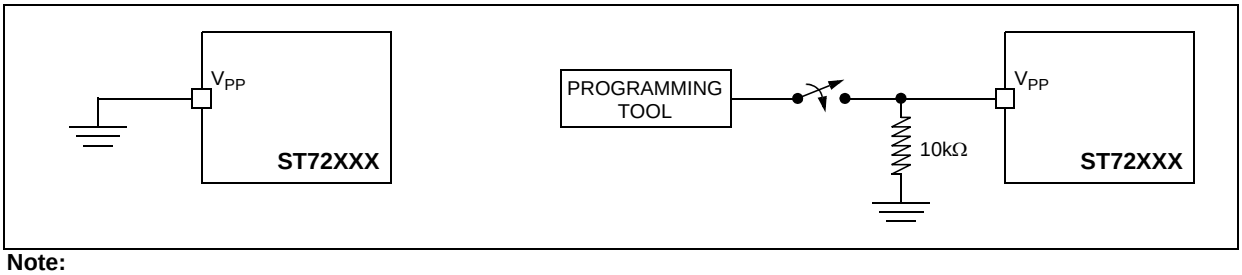
14.6.2 Flash memory

Operating Conditions: f_{CPU} = 8 MHz.

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

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

Figure 56. Two typical applications with V_{PP} pin[1]



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

14.7 EMC characteristics

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

14.7.1 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 electro magnetic events until a failure occurs (indicated by the LEDs).

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

A device reset allows normal operations to be resumed. The test results are given in the table below based on the EMS levels and classes defined in application note AN1709.

14.7.1.1 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).

Symbol	Parameter	Conditions	Level/ Class
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD}=5V, T_A=+25°C, f_{OSC}=8MHz, PSDIP32 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}=5V, T_A=+25°C, f_{OSC}=8MHz, PSDIP32 conforms to IEC 1000-4-4	4A

14.7.2 Electromagnetic 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.

Symbol	Parameter	Conditions	Monitored frequency band	Max vs. [f_{OSC}/ f_{CPU}]	Unit
				16/8MHz	
S_{EMI}	Peak level [1]	V_{DD}=5V, T_A=+25°C, PSDIP32 package conforming to SAE J 1752/3 **Note:** Refer to Application Note AN1709 for data on other package types.	0.1MHz to 30MHz	36	dBµV
			30MHz to 130MHz	39	
			130MHz to 1GHz	26	
			SAE EMI Level	3.5	-

Note:

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

EMC characteristics (Cont'd)

14.7.3 Absolute maximum ratings (electrical sensitivity)

Based on three different tests (ESD, LU and DLU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity. For more details, refer to the application note AN1181.

14.7.3.1 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 JESD22-A114A/A115A standard.

Absolute maximum ratings

Symbol	Ratings	Conditions	Maximum value [1]	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human body model)	T_A=+25°C	2000	V

Note:

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

14.7.3.2 Static and dynamic Latch-Up

- **LU**: 3 complementary static tests are required on 10 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. For more details, refer to the application note AN1181.
- **DLU**: Electrostatic discharges (one positive then one negative test) are applied to each pin of 3 samples when the micro is running to assess the latch-up performance in dynamic mode. Power supplies are set to the typical values, the oscillator is connected as near as possible to the pins of the micro and the component is put in reset mode. This test conforms to the IEC1000-4-2 and SAEJ1752/3 standards. For more details, refer to the application note AN1181.

Electrical sensitivities

Symbol	Parameter	Conditions	Class [1]
LU	Static latch-up class	T_A=+25°C	A
DLU	Dynamic latch-up class	V_{DD}=5.5V, f_{OSC}=4MHz, T_A=+25°C	A

Note:

1. Class description: A Class is an STMicroelectronics internal specification. All its limits are higher than the JEDEC specifications, that means when a device belongs to Class A it exceeds the JEDEC standard. B Class strictly covers all the JEDEC criteria (international standard).

14.8 I/O port pin characteristics

14.8.1 General characteristics

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

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}=5V	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=50pF 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}

Notes:

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 57). 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 58).

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 57. Two typical applications with unused I/O pin

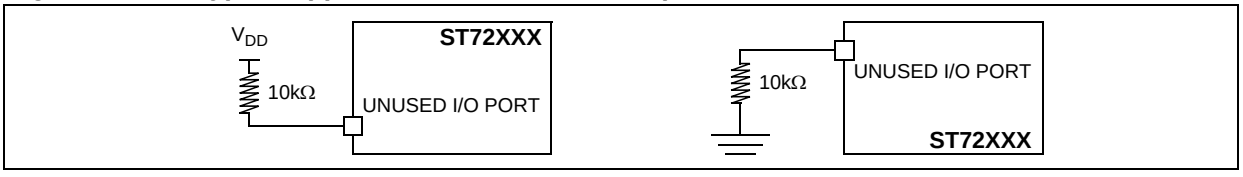


Figure 58. Typ. I_{PU} vs. V_{DD}

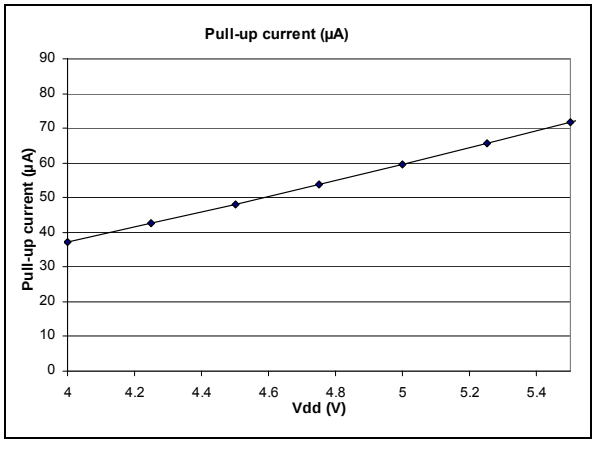
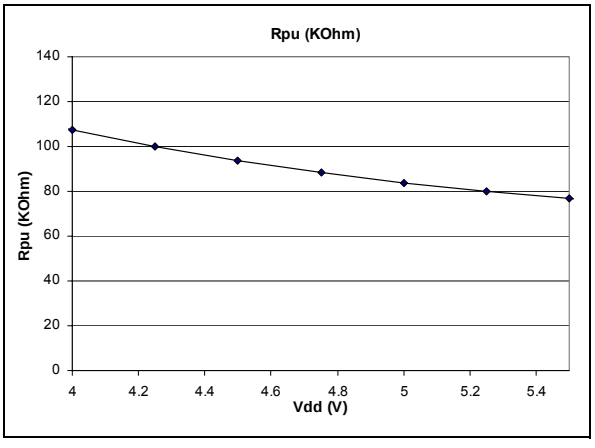


Figure 59. Typ. R_{PU} vs. V_{DD}



14.8.2 Output driving current

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

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), Port D(0:7)	V_{DD}=5V	I_{IO}=+1.6mA		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}=+10mA		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}=+25mA		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}=-10mA	V_{DD}-1.3 [3]		
			I_{IO}=-1.6mA	V_{DD}-0.8		

Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 14.2 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 14.2 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}.

3. The minimum V_{OH} value (with I_{IO}=-10mA) depends on the chosen device type. For Flash devices, min = V_{DD} - 1.3 V and for ROM devices, min = V_{DD} - 1.7 V

Figure 60. V_{OL} standard V_{DD}=5 V

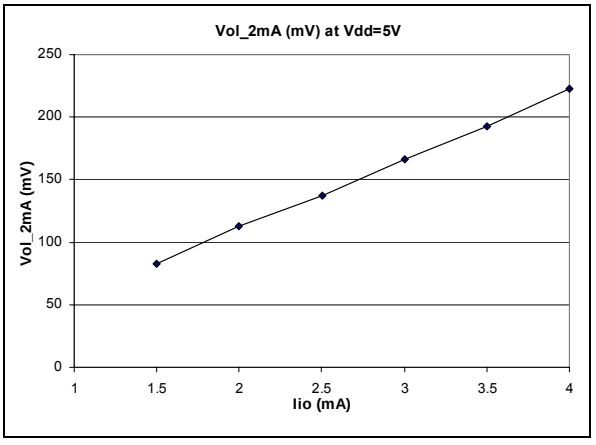


Figure 61. V_{OL} high sink V_{DD}=5 V

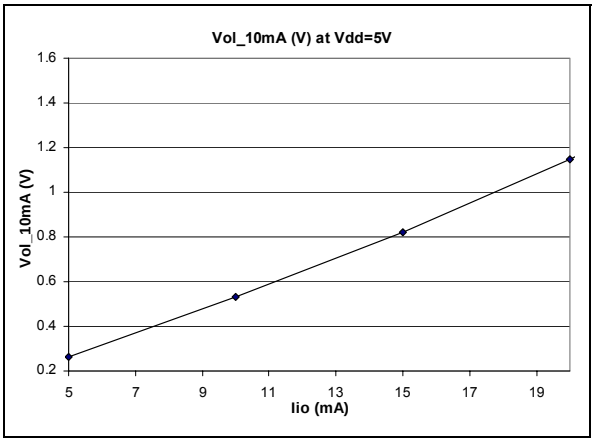


Figure 62. V_{OL} very high sink V_{DD}=5 V

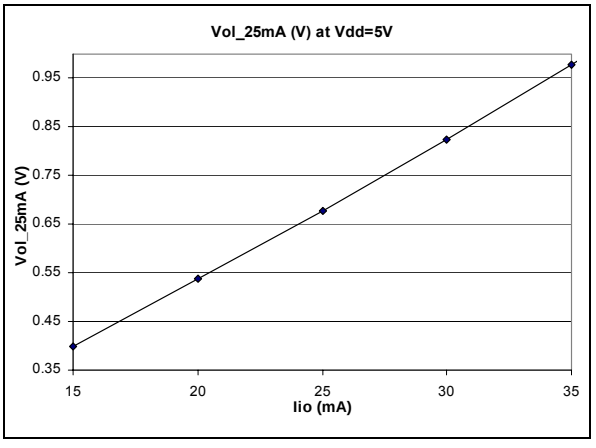


Figure 64. V_{OL} high sink vs. V_{DD}

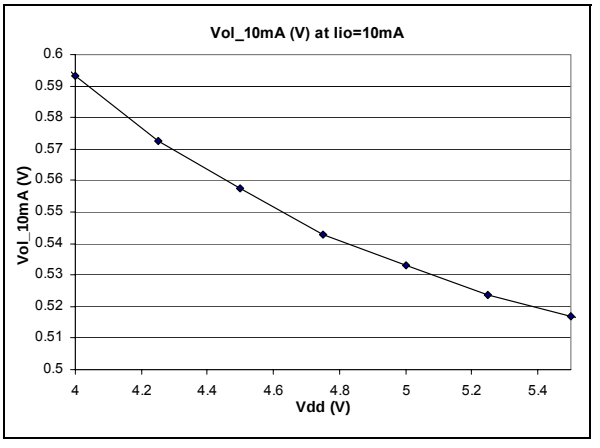


Figure 63. V_{OL} standard vs. V_{DD}

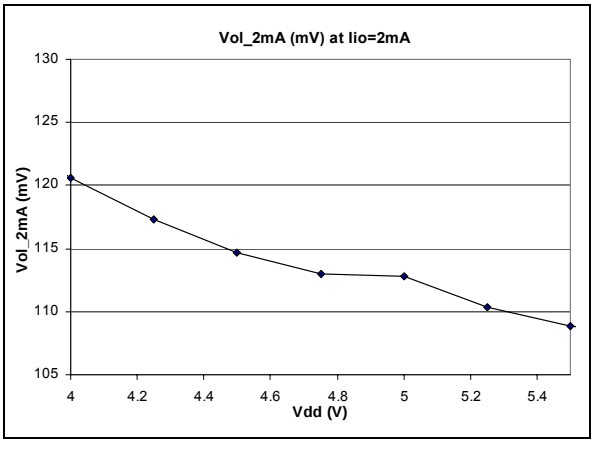


Figure 65. V_{OL} very high sink vs. V_{DD}

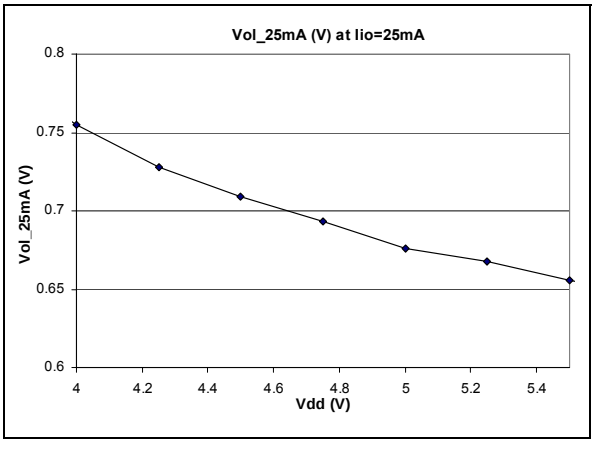


Figure 66. $|V_{DD}-V_{OH}|$ @ V_{DD}=5 V (low current)

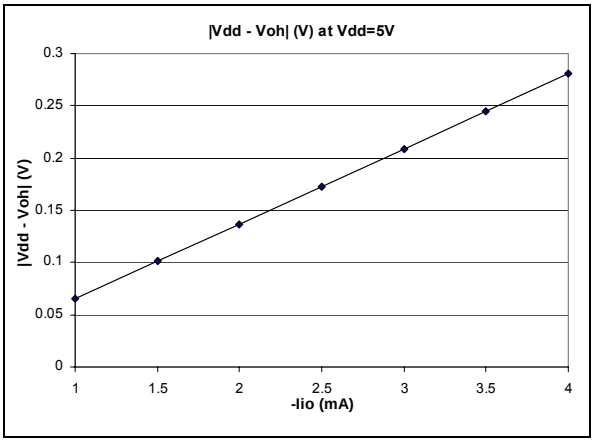


Figure 67. $|V_{DD}-V_{OH}|$ @ V_{DD}=5 V (high current)

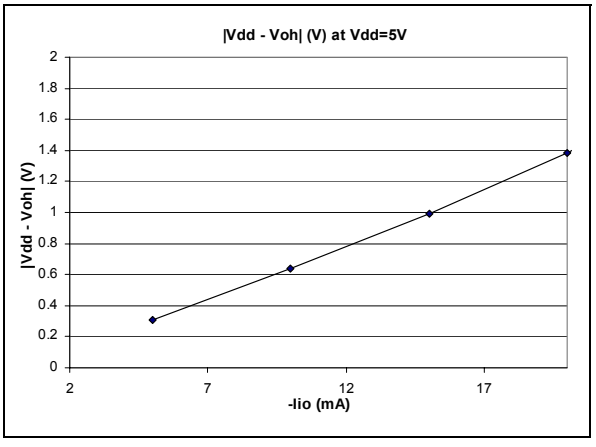


Figure 68. $|V_{DD}-V_{OH}|$ @ I_{IO}=2 mA (low current)

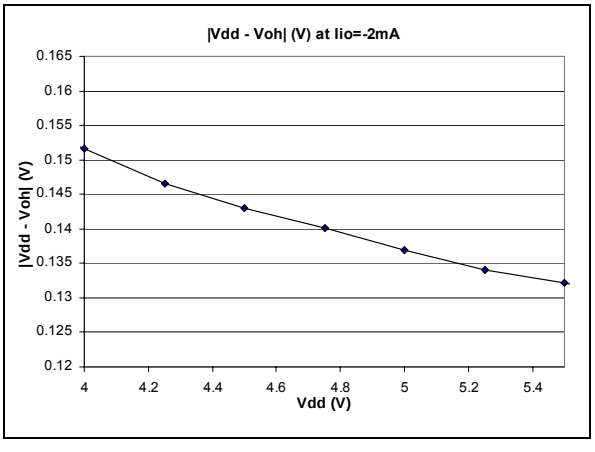
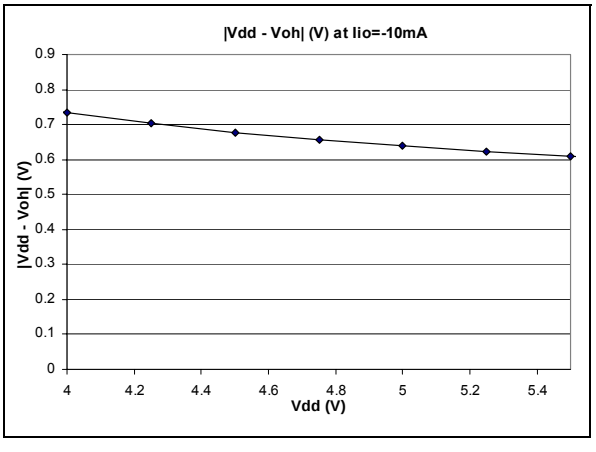


Figure 69. $|V_{DD}-V_{OH}|$ @ I_{IO}=10 mA (high current)



14.9 Control pin characteristics

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

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

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}=5mA			0.8	V
			I_{IO}=7.5mA			1.3	
R_{ON}	Weak pull-up equivalent resistor [3]	$V_{IN}=V_{SS}$	V_{DD}=5V	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

Notes:

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 14.2 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.

Control pin characteristics (Cont'd)

Figure 70. $\overline{\text{RESET}}$ pin protection when LVD is enabled.[1)2)3)4)]

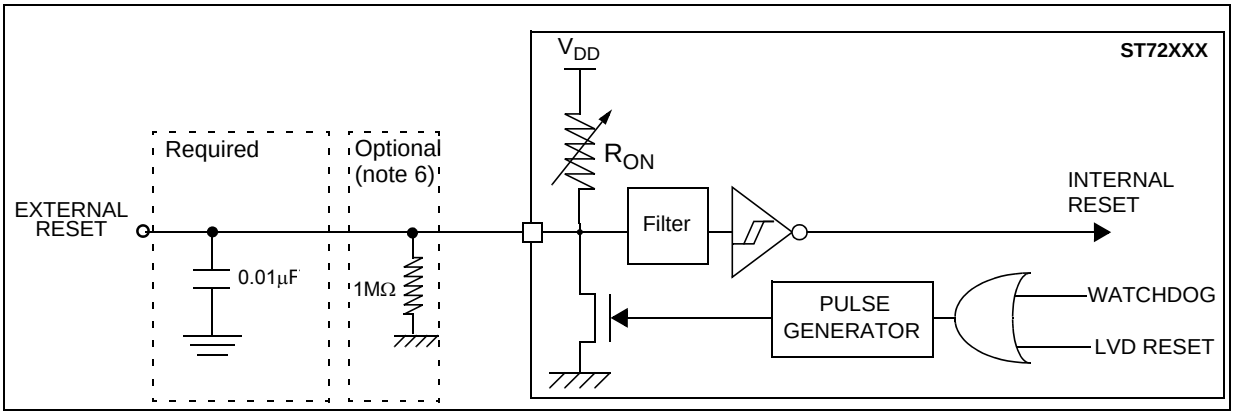
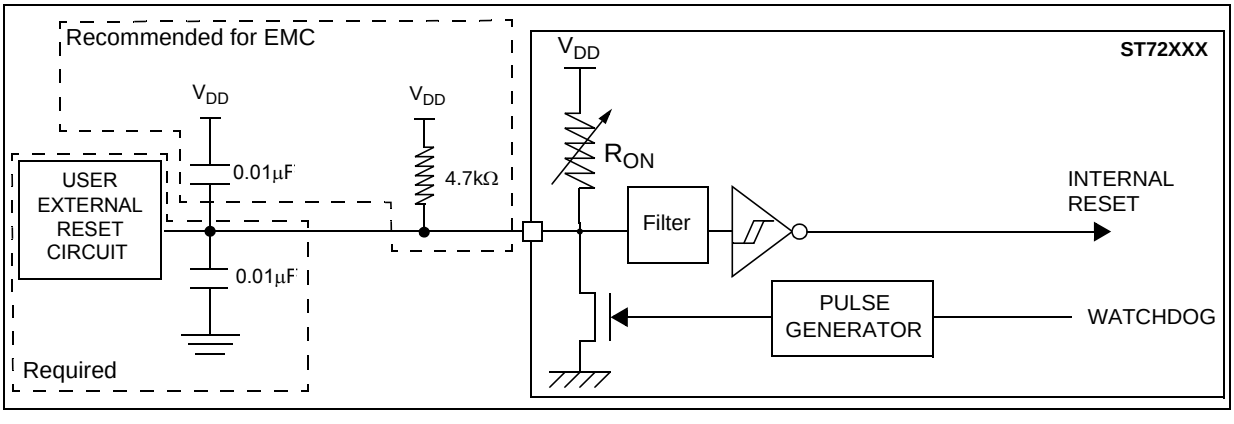


Figure 71. $\overline{\text{RESET}}$ pin protection when LVD is disabled.[1)]



Note 1:

- The reset network protects the device against parasitic resets.
- 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).
- 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 14.9.1 on page 121. Otherwise the reset will not be taken into account internally.
- 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 is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in section 14.2.2 on page 109.

Note 2: When the LVD is enabled, it is recommended not to connect a pull-up resistor or capacitor. A 10nF pull-down capacitor is required to filter noise on the reset line.

Note 3: In case a capacitive power supply is used, it is recommended to connect a 1MΩ pull-down resistor to the $\overline{\text{RESET}}$ pin to discharge any residual voltage induced by the capacitive effect of the power supply (this will add 5µA to the power consumption of the MCU).

Note 4: Tips when using the LVD:

- 1. Check that all recommendations related to ICCCLK and reset circuit have been applied (see notes above).
- 2. Check that the power supply is properly decoupled (100nF + 10µF close to the MCU). Refer to AN1709 and AN2017. If this cannot be done, it is recommended to put a 100nF + 1MΩ pull-down on the $\overline{\text{RESET}}$ pin.
- 3. The capacitors connected on the RESET pin and also the power supply are key to avoid 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 5µF to 20µF capacitor."

14.10 Communication interface characteristics

14.10.1 USB - universal bus interface

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

USB DC electrical characteristics					
Symbol	**Parameter**	**Conditions**	**Min.**	**Max.**	**Unit**
VDI	Differential Input Sensitivity	I(D+, D-)	0.2		V [2]
VCM	Differential Common mode Range	Includes VDI range	0.8	2.5	
VSE	Single Ended Receiver Threshold		0.8	2.0	
VOL	Static Output Low	RL [1] of 1.5K ohms to 3.6v		0.3	
VOH	Static Output High	RL [1] of 15K ohms to V_{SS}	2.8	3.6	
USBV	USBVCC: voltage level [3]	V_{DD}=5v	3.00	3.60	

Notes:

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

Figure 72. USB: data signal rise and fall time

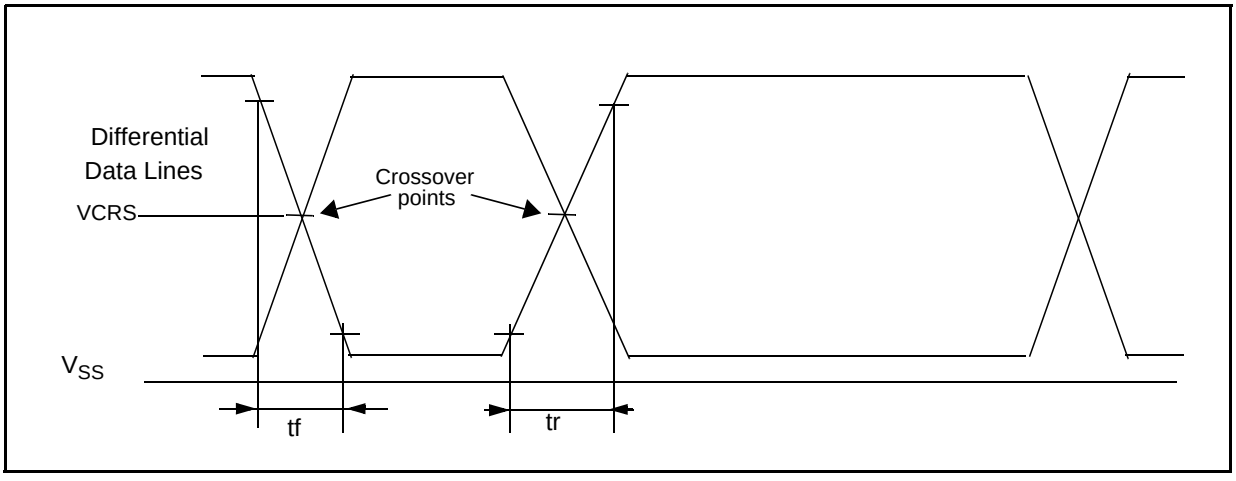


Table 27. USB: low-speed electrical characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
	Driver characteristics:				
tr	Rise time	CL=50 pF [1]	75		ns
		CL=600 pF [1]		300	ns
tf	Fall Time	CL=50 pF [1]	75		ns
		CL=600 pF [1]		300	ns
trfm	Rise/ Fall Time matching	tr/tf	80	120	%
VCRS	Output signal Crossover Voltage		1.3	2.0	V

Note:

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

Communication interface characteristics (Cont'd)

14.10.2 SCI - serial communication 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).

Symbol	Parameter	Conditions			Standard	Baud Rate	Unit
		f_{CPU}	Accuracy vs. standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8MHz	~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

14.10.3 I^2C - Inter IC control interface

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

The ST7 I^2C interface meets the requirements of the Standard I^2C communication protocol described in the following table.

(Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified)

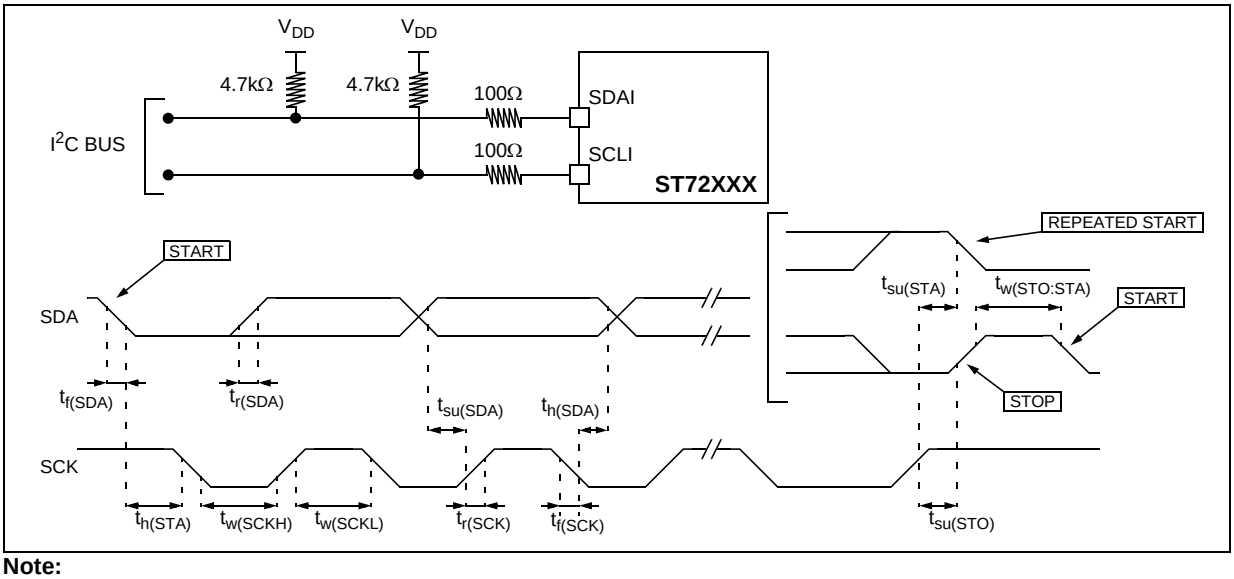
Symbol	Parameter	Standard mode I^2C		Fast mode I^2C [4]		Unit
		Min [1]	Max [1]	Min [1]	Max [1]	
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		µs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		
$t_{su(SDA)}$	SDA setup time	250		100		ns
$t_{h(SDA)}$	SDA data hold time	0 [3]		0 [2]	900 [3]	
$t_{r(SDA)}$ $t_{r(SCL)}$	SDA and SCL rise time		1000	$20+0.1C_b$	300	
$t_{f(SDA)}$ $t_{f(SCL)}$	SDA and SCL fall time		300	$20+0.1C_b$	300	
$t_{h(STA)}$	START condition hold time	4.0		0.6		µs
$t_{su(STA)}$	Repeated START condition setup time	4.7		0.6		
$t_{su(STO)}$	STOP condition setup time	4.0		0.6		µs
$t_{w(STO:STA)}$	STOP to START condition time (bus free)	4.7		1.3		µs
C_b	Capacitive load for each bus line		400		400	pF

Notes:

1. Data based on standard I^2C protocol requirement, not tested in production.
2. The device must internally provide a hold time of at least 300ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.
3. The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal.
4. At 4MHz f_{CPU}, max.I^2C speed (400kHz) is not achievable. In this case, max. I^2C speed will be approximately 260KHz.

Communication interface characteristics (Cont'd)

Figure 73. Typical application with I²C Bus and timing diagram 1)



Note:

1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

The following table gives the values to be written in the I2CCCR register to obtain the required I²C SCL line frequency .

Table 28. SCL frequency table

f_{SCL} (kHz)	I2CCCR value							
	f_{CPU}=4 MHz.				f_{CPU}=8 MHz.			
	V_{DD} = 4.1 V		V_{DD} = 5 V		V_{DD} = 4.1 V		V_{DD} = 5 V	
	R_P=3.3kΩ	R_P=4.7kΩ	R_P=3.3kΩ	R_P=4.7kΩ	R_P=3.3kΩ	R_P=4.7kΩ	R_P=3.3kΩ	R_P=4.7kΩ
400	NA	NA	NA	NA	83h	83	83h	83h
300	NA	NA	NA	NA	85h	85h	85h	85h
200	83h	83h	83h	83h	8Ah	89h	8Ah	8Ah
100	10h	10h	10h	10h	24h	23h	24h	23h
50	24h	24h	24h	24h	4Ch	4Ch	4Ch	4Ch
20	5Fh	5Fh	5Fh	5Fh	FFh	FFh	FFh	FFh

Legend:

R_P = External pull-up resistance

f_{SCL} = I²C speed

NA = Not achievable

Notes:

– For speeds around 200 kHz, achieved speed can have ±5% tolerance

– For other speed ranges, achieved speed can have ±2% tolerance

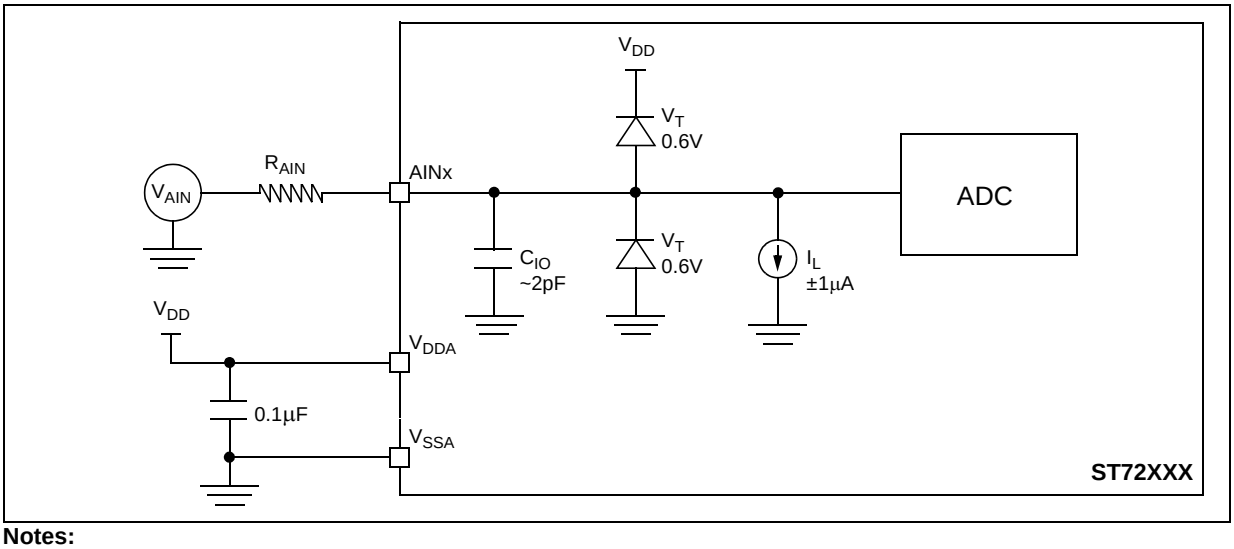
The above variations depend on the accuracy of the external components used.

14.11 8-bit ADC characteristics

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

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
f_{ADC}	ADC clock frequency				4	MHz
V_{AIN}	Conversion range voltage [2]		V_{SSA}		V_{DDA}	V
R_{AIN}	External input resistor				10 [3]	kΩ
C_{ADC}	Internal sample and hold capacitor			6		pF
t_{STAB}	Stabilization time after ADC enable	f_{CPU}=8MHz, f_{ADC}=2MHz	0 [4]			µs
t_{ADC}	Conversion time (Sample+Hold)		6			$1/f_{ADC}$
	- Sample capacitor loading time		4			
	- Hold conversion time		8			

Figure 74. Typical application with ADC



Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V.
2. When V_{DDA} and V_{SSA} pins are not available on the pinout, the ADC refer to V_{DD} and V_{SS}.
3. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.
4. The stabilization time of the AD converter is masked by the first t_{LOAD}. The first conversion after the enable is then always valid.

8-bit ADC characteristics (Cont'd)

ADC accuracy with V_{DD}=5V, f_{CPU}=8 MHz, f_{ADC}=4 MHz R_{AIN}< 10kΩ

Symbol	Parameter	Typ	Max [2) 3)]
$\|E_T\|$	Total unadjusted error [1)]	1.5	2
$\|E_O\|$	Offset error [1)]	0.5	1
$\|E_G\|$	Gain Error [1)]	0.5	1.5
$\|E_D\|$	Differential linearity error [1)]	1	1.5
$\|E_L\|$	Integral linearity error [1)]	1	1.5

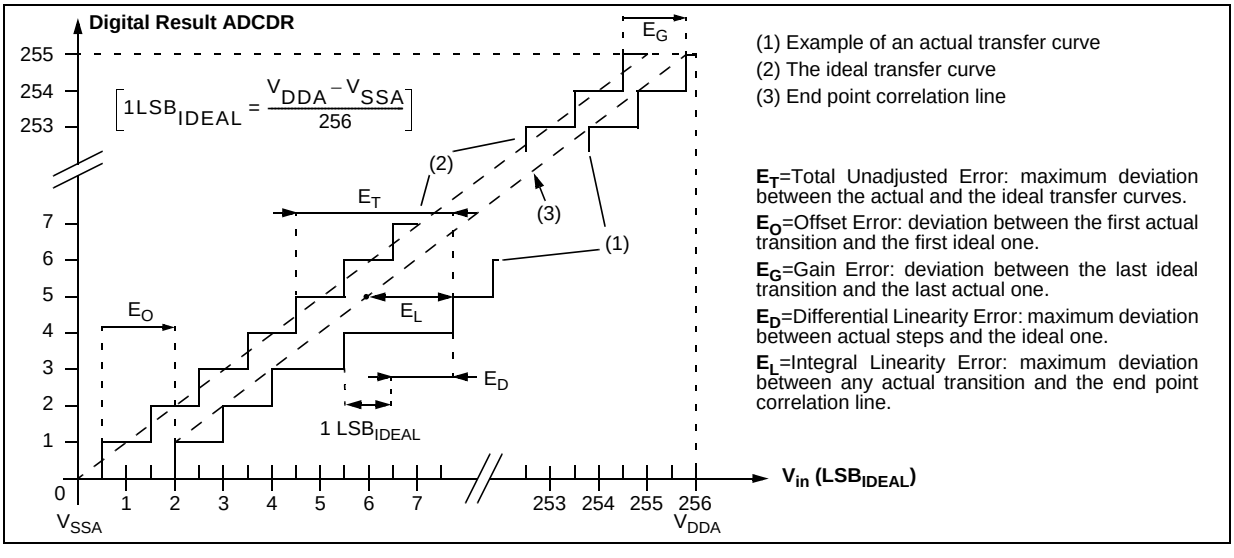
Notes:

1. ADC Accuracy vs. Negative Injection Current:
For I_{INJ-}=0.8mA, the typical leakage induced inside the die is 1.6µA and the effect on the ADC accuracy is a loss of 1 LSB for each 10KΩ increase of the external analog source impedance. This effect on the ADC accuracy has been observed under worst-case conditions for injection:
- negative injection
- injection to an Input with analog capability, adjacent to the enabled Analog Input
- at 5V V_{DD} supply, and worst case temperature.

2. Data based on characterization results over the whole temperature range, not tested in production.

3. Data based on characterization results, to guarantee 99.73% within ± max value from 0°C to 70°C (± 3σ distribution limits).

Figure 75. ADC accuracy characteristics



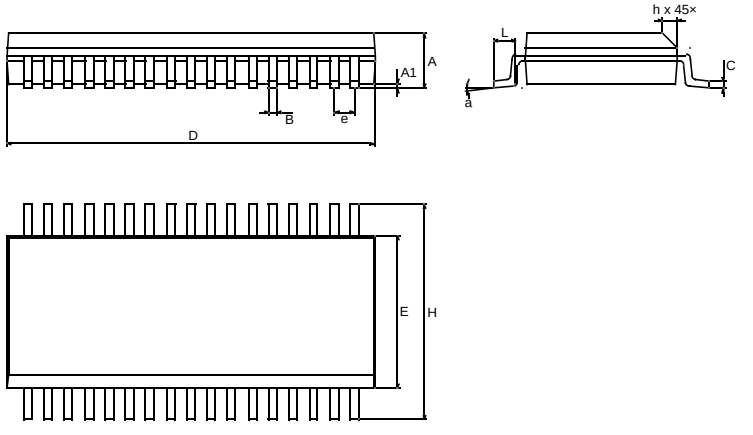
15 Package characteristics

In order to meet environmental requirements, ST offers these devices in ECOPACK® packages. These packages have a Lead-free second level interconnect. The category of second Level Interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

ECOPACK is an ST trademark. ECOPACK specifications are available at: www.st.com

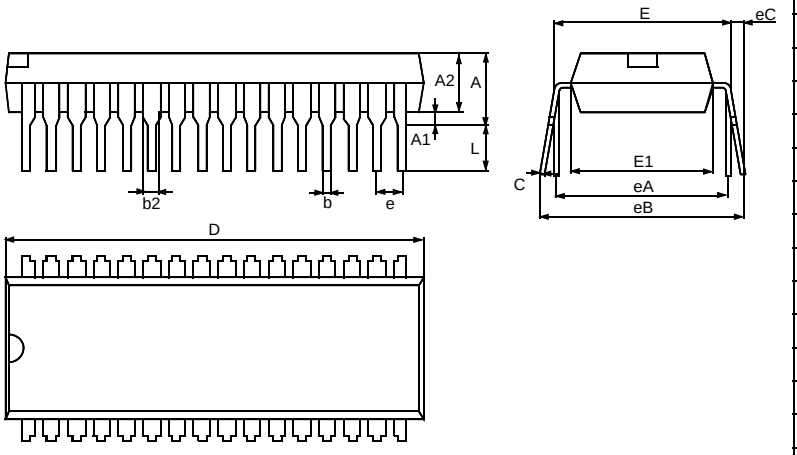
15.1 Package mechanical data

Figure 76. 34-pin Plastic Small Outline Package, shrink 300-mil width



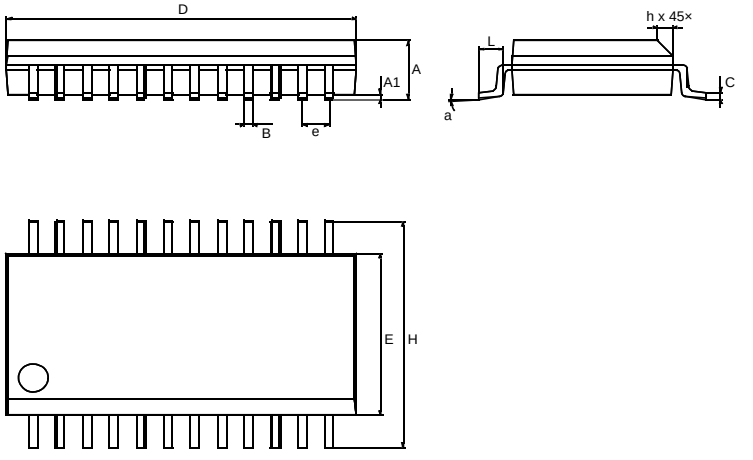
Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.464		2.642	0.097		0.104
A1	0.127		0.292	0.005		0.012
B	0.356		0.483	0.014		0.019
C	0.231		0.318	0.009		0.013
D	17.729		18.059	0.698		0.711
E	7.417		7.595	0.292		0.299
e		1.016			0.040	
H	10.160		10.414	0.400		0.410
h	0.635		0.737	0.025		0.029
α	0°		8°	0°		8°
L	0.610		1.016	0.024		0.040
	Number of Pins					
N	34					

Figure 77. 32-pin Plastic Dual In-Line Package, shrink 400-mil width



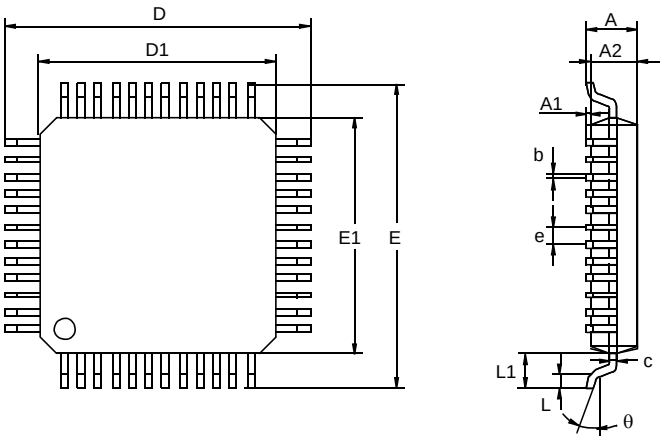
Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	3.56	3.76	5.08	0.140	0.148	0.200
A1	0.51			0.020		
A2	3.05	3.56	4.57	0.120	0.140	0.180
b	0.36	0.46	0.58	0.014	0.018	0.023
b1	0.76	1.02	1.40	0.030	0.040	0.055
C	0.20	0.25	0.36	0.008	0.010	0.014
D	27.43		28.45	1.080	1.100	1.120
E	9.91	10.41	11.05	0.390	0.410	0.435
E1	7.62	8.89	9.40	0.300	0.350	0.370
e		1.78			0.070	
eA		10.16			0.400	
eB			12.70			0.500
eC			1.40			0.055
L	2.54	3.05	3.81	0.100	0.120	0.150
	Number of Pins					
N	32					

Figure 78. 24-pin Plastic Small Outline Package, 300-mil width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	15.20		15.60	0.599		0.614
E	7.40		7.60	0.291		0.299
e		1.27			0.050	
H	10.00		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
α	0°		8°	0°		8°
L	0.40		1.27	0.016		0.050
	Number of Pins					
N	24					

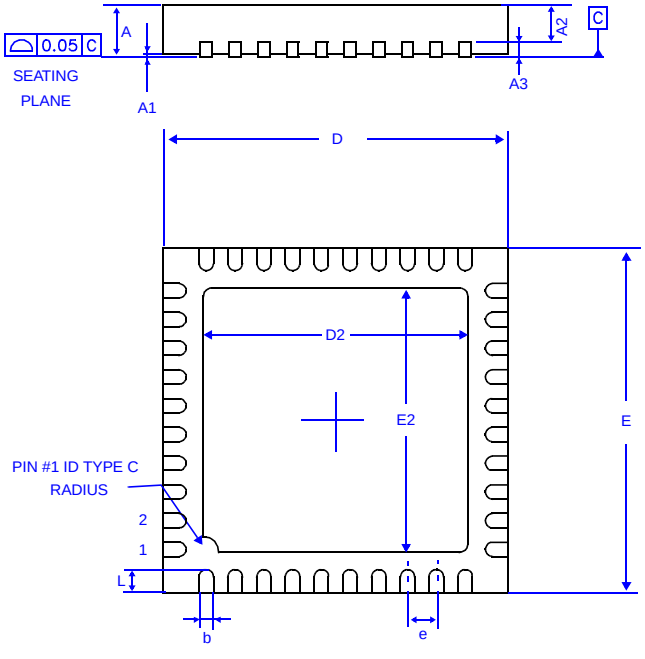
Figure 79. 48-pin Low profile Quad Flat package



Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.007	0.009	0.011
C	0.09		0.20	0.004		0.008
D		9.00			0.354	
D1		7.00			0.276	
E		9.00			0.354	
E1		7.00			0.276	
e		0.50			0.020	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
	Number of Pins					
N	48					

Note 1. values in inches are converted from mm and rounded to 3 decimal digits.

Figure 80. 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	0.031	0.035	0.039
A1		0.02	0.05		0.001	0.002
A2		0.65	1.00		0.026	0.039
A3		0.20			0.008	
b	0.18	0.25	0.30	0.007	0.010	0.012
D	5.85	6.00	6.15	0.230	0.236	0.242
D2	2.75	2.9	3.05	0.108	0.114	0.120
E	5.85	6	6.15	0.230	0.236	0.242
E2	2.75	2.9	3.05	0.108	0.114	0.120
e		0.50			0.020	
L	0.30	0.40	0.50	0.012	0.016	0.020
	Number of Pins					
N	40					

Note 1. values in inches are converted from mm and rounded to 3 decimal digits.

15.2 Thermal characteristics

Symbol	Ratings	value	Unit
R_{thJA}	Package thermal resistance (junction to ambient) SDIP32 SO34 SO24 LQFP48 QFN40	 60 75 70 80 34	°C/W
P_D	Power dissipation [1]	500	mW
T_{Jmax}	Maximum junction temperature [2]	150	°C

Notes:

1. The power dissipation is obtained from the formula $P_D=P_{INT}+P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD}xV_{DD}$) and P_{PORT} is the port power dissipation determined by the user.
2. The average chip-junction temperature can be obtained from the formula $T_J = T_A + P_D$ x RthJA.

15.3 Soldering and glueability information

Recommended soldering information given only as design guidelines in Figure 81 and Figure 82.

Recommended glue for SMD plastic packages dedicated to molding compound with silicone:

- Heraeus: PD945, PD955
- Loctite: 3615, 3298

Figure 81. Recommended wave soldering profile (with 37% Sn and 63% Pb)

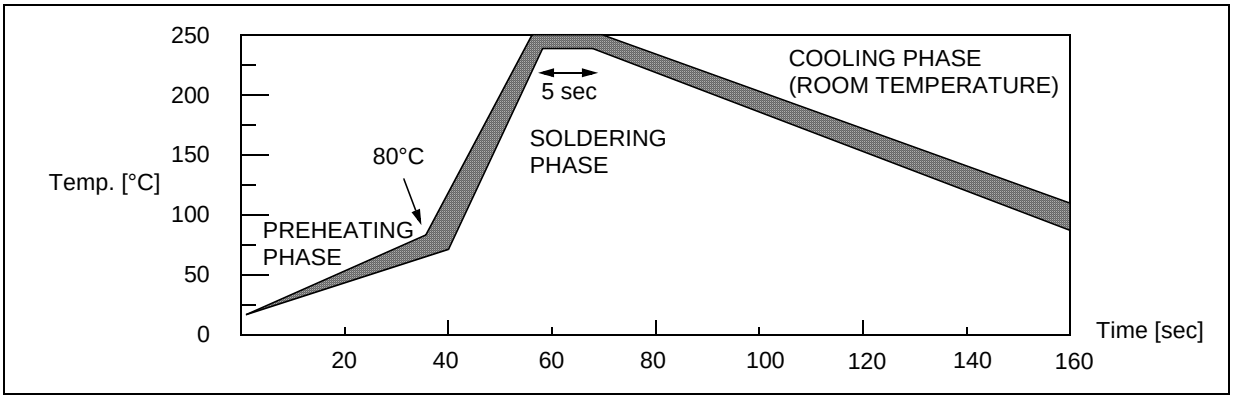
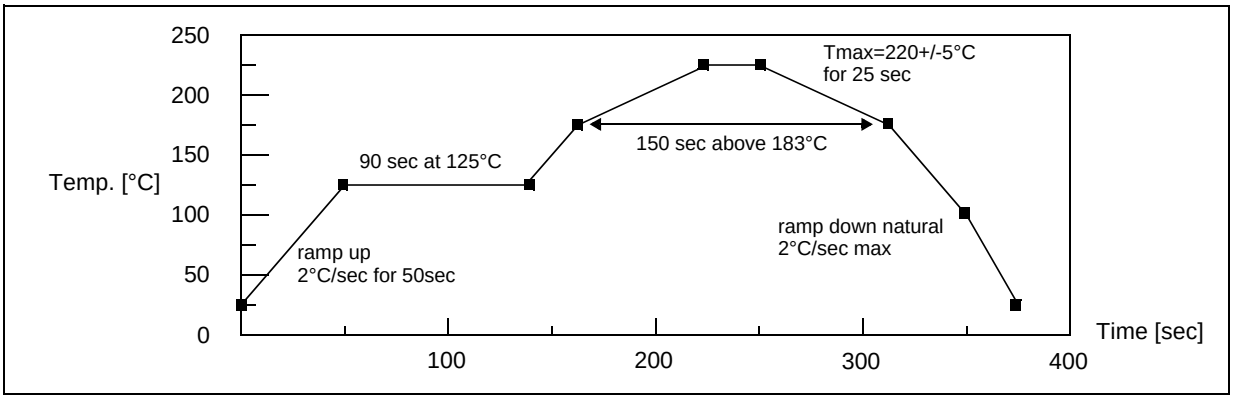


Figure 82. Recommended reflow soldering oven profile (MID JEDEC)



16 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).

ST72P63B devices are Factory Advanced Service Technique ROM (FASTROM) versions: they are factory programmed FLASH devices.

ST72F63B 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 ROM devices are factory-configured.

16.1 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.
In ROM devices, the Option Byte is fixed in hardware by the ROM code.

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
Important note: on 4K and 8K ROM devices listed below, this option bit is forced by ST to 0 (LVD always enabled):
ST7263BK1M1, ST7263BK2M1
ST7263BK2B1, ST7263BK2B1

OPT 2 = Reserved.

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

OPT 0 = **FMP_R** Flash memory Read-Out protection

This option indicates if the user flash memory is protected against Read-Out. Read-Out 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 4.2.1 on page 17 for more details.
0: Read-Out protection enabled
1: Read-Out protection disabled

16.2 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 138.

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 29. Supported part numbers

Sales Type [1)]	Program memory (bytes)	RAM (bytes)	Package
ST72F63BH6T1	32K Flash	1024	LQFP48
ST72F63BD6U1			QFN40
ST72F63BK6M1			SO34
ST72F63BK6B1			SDIP32
ST72F63BE6M1			SO24
ST72F63BH4T1	16K Flash	512	LQFP48
ST72F63BK4M1			SO34
ST72F63BK4B1			SDIP32
ST72F63BE4M1			SO24
ST72F63BH2T1	8K Flash	384	LQFP48
ST72F63BK2U1			QFN40
ST72F63BK2M1			SO34
ST72F63BK2B1			SDIP32
ST72F63BE2M1			SO24
ST72F63BK1M1	4K Flash	384	SO34
ST72F63BK1B1			SDIP32
ST72F63BE1M1			SO24
ST7263BH6T1/xxx	32K ROM	1024	LQFP48
ST7263BD6U1/xxx			QFN40
ST7263BK6M1/xxx			SO34
ST7263BK6B1/xxx			SDIP32
ST7263BE6M1/xxx			SO24
ST7263BH4T1/xxx	16K ROM	512	LQFP48
ST7263BK4M1/xxx			SO34
ST7263BK4B1/xxx			SDIP32
ST7263BE4M1/xxx			SO24
ST7263BH2T1/xxx	8K ROM	384	LQFP48
ST7263BK2U1/xxx			QFN40
ST7263BK2M1/xxx			SO34
ST7263BK2B1/xxx			SDIP32
ST7263BE2M1/xxx			SO24
ST7263BK1M1/xxx	4K ROM	384	SO34
ST7263BK1B1/xxx			SDIP32
ST7263BE1M1/xxx			SO24
ST72P63BH6T1	32K FASTROM	1024	LQFP48
ST72P63BD6U1			QFN40
ST72P63BK6M1			SO34
ST72P63BK6B1			SDIP32
ST72P63BE6M1			SO24
ST72P63BH4T1	16K FASTROM	512	LQFP48
ST72P63BK4M1			SO34
ST72P63BK4B1			SDIP32
ST72P63BE4M1			SO24

ST72P63BH2T1	8K FASTROM	384	LQFP48
ST72P63BK2U1			QFN40
ST72P63BK2M1			SO34
ST72P63BK2B1			SDIP32
ST72P63BE2M1			SO24
ST72P63BK1M1	4K FASTROM	384	SO34
ST72P63BK1B1			SDIP32
ST72P63BE1M1			SO24

Note:

Contact ST sales office for product availability

1. /xxx stands for the ROM code name assigned by STMicroelectronics

16.3 Development tools

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

16.3.1 Evaluation tools and starter kits

ST offers complete, affordable **starter kits** and full-featured **evaluation boards** that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete, affordable hardware/software tool packages that include features and samples to to help you quickly start developing your application. ST evaluation boards are open-design, embedded systems, which are developed and documented to serve as references for your application design. They include sample application software to help you demonstrate, learn about and implement your ST7's features.

16.3.2 Development and debugging tools

Application development for ST7 is supported by fully optimizing **C Compilers** and the **ST7 Assembler-Linker** toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16K of code.

The range of hardware tools includes full-featured **ST7-EMU3 series emulators** and the low-cost **RLink** in-circuit debugger/programmer. These tools are supported by the **ST7 Toolset** from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

16.3.3 Programming tools

During the development cycle, the **ST7-EMU3 series emulators** and the **RLink** provide in-circuit programming capability for programming the Flash microcontroller on your application board.

In addition ST provides dedicated programming tools including the **ST7-EPB programming boards**, which include all the sockets required to program any of the devices in a specific ST7 subfamily.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

16.3.4 Order codes for ST7263B development tools

Table 30. Development tool order codes for the ST7263B family

MCU	Starter kit	Evaluation board	Emulator	In-circuit debugger/ programmer	Dedicated programmer
ST7263B	ST72F63B-SK/ RAIS	ST7MDTULS-EVAL	ST7MDTU3-EMU3	STX-RLINK	ST7MDTU3-EPB

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

ST7263B microcontroller option list
(Last update: Mar 2007)

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

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

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

ROM DEVICE:	4K	8K	16K	32K
SO24:	[ ] ST7263BE1M1	[ ] ST7263BE2M1	[ ] ST7263BE4B1	[ ] ST7263BE6M1
PSDIP32:	[ ] ST7263BK1B1	[ ] ST7263BK2B1	[ ] ST7263BK4B1	[ ] ST7263BK6B1
SO34:	[ ] ST7263BK1M1	[ ] ST7263BK2M1	[ ] ST7263BK4M1	[ ] ST7263BK6M1
QFN40:		[ ] ST7263BK2U1		[ ] ST7263BD6U1
LQFP48:		[ ] ST7263BH2T1	[ ] ST7263BH4T1	[ ] ST7263BH6T1

FASTROM:	4K	8K	16K	32K
SO24:	[ ] ST72P63BE1M1	[ ] ST72P63BE2M1	[ ] ST72P63BE4M1	[ ] ST72P63BE6M1
PSDIP32:	[ ] ST72P63BK1B1	[ ] ST72P63BK2B1	[ ] ST72P63BK4B1	[ ] ST72P63BK6B1
SO34:	[ ] ST72P63BK1M1	[ ] ST72P63BK2M1	[ ] ST72P63BK4M1	[ ] ST72P63BK6M1
QFN40:		[ ] ST72P63BK2U1		[ ] ST72P63BD6U1
LQFP48:		[ ] ST72P63BH2T1	[ ] ST72P63BH4T1	[ ] ST72P63BH6T1

DIE FORM:	4K	8K	16K	32K
24-pin:	[ ] (as E1M1)	[ ] (as E2M1)	[ ] (as E4M1)	[ ] (as E6M1)
32-pin:	[ ] (as K1B1)	[ ] (as K2B1)	[ ] (as K4B1)	[ ] (as K6B1)
34-pin:	[ ] (as K1M1)	[ ] (as K2M1)	[ ] (as K4M1)	[ ] (as K6M1)
40-pin:		[ ] (as K2U1)		[ ] (as D6U1)
48-pin:		[ ] (as H2T1)	[ ] (as H4T1)	[ ] (as H6T1)

Conditioning (check only one option):

Packaged Product	Die Product (dice tested at 25°C only)
[ ] Tape & Reel (SO package only)	[ ] Tape & Reel
[ ] Tube	[ ] Inked wafer
	[ ] Sawn wafer on sticky foil

Special Marking (ROM only): [ ] 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*
* LVD is forced to 0 (LVD always enabled) for 4K and 8K ROM devices
(sales types ST7263BK1B1, ST7263BK2B1, ST7263BK1M1, ST72BK2M1 only)

Oscillator Selection: [ ] 24 MHz. [ ] 12 MHz.
Read-Out Protection: [ ] Disabled [ ] Enabled

Date . . . . . .

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

16.4 ST7 application notes

Table 31. ST7 application notes

IDENTIFICATION	DESCRIPTION
APPLICATION EXAMPLES	
AN1658	SERIAL NUMBERING IMPLEMENTATION
AN1720	MANAGING THE READ-OUT PROTECTION IN FLASH MICROCONTROLLERS
AN1755	A HIGH RESOLUTION/PRECISION THERMOMETER USING ST7 AND NE555
AN1756	CHOOSING A DALI IMPLEMENTATION STRATEGY WITH ST7DALI
AN1812	A HIGH PRECISION, LOW COST, SINGLE SUPPLY ADC FOR POSITIVE AND NEGATIVE INPUT VOLTAGES
EXAMPLE DRIVERS	
AN 969	SCI COMMUNICATION BETWEEN ST7 AND PC
AN 970	SPI COMMUNICATION BETWEEN ST7 AND EEPROM
AN 971	I²C COMMUNICATION BETWEEN ST7 AND M24CXX EEPROM
AN 972	ST7 SOFTWARE SPI MASTER COMMUNICATION
AN 973	SCI SOFTWARE COMMUNICATION WITH A PC USING ST72251 16-BIT TIMER
AN 974	REAL TIME CLOCK WITH ST7 TIMER OUTPUT COMPARE
AN 976	DRIVING A BUZZER THROUGH ST7 TIMER PWM FUNCTION
AN 979	DRIVING AN ANALOG KEYBOARD WITH THE ST7 ADC
AN 980	ST7 KEYPAD DECODING TECHNIQUES, IMPLEMENTING WAKE-UP ON KEYSTROKE
AN1017	USING THE ST7 UNIVERSAL SERIAL BUS MICROCONTROLLER
AN1041	USING ST7 PWM SIGNAL TO GENERATE ANALOG OUTPUT (SINUSOÏD)
AN1042	ST7 ROUTINE FOR I²C SLAVE MODE MANAGEMENT
AN1044	MULTIPLE INTERRUPT SOURCES MANAGEMENT FOR ST7 MCUS
AN1045	ST7 S/W IMPLEMENTATION OF I²C BUS MASTER
AN1046	UART EMULATION SOFTWARE
AN1047	MANAGING RECEPTION ERRORS WITH THE ST7 SCI PERIPHERALS
AN1048	ST7 SOFTWARE LCD DRIVER
AN1078	PWM DUTY CYCLE SWITCH IMPLEMENTING TRUE 0% & 100% DUTY CYCLE
AN1082	DESCRIPTION OF THE ST72141 MOTOR CONTROL PERIPHERALS REGISTERS
AN1083	ST72141 BLDC MOTOR CONTROL SOFTWARE AND FLOWCHART EXAMPLE
AN1105	ST7 PCAN PERIPHERAL DRIVER
AN1129	PWM MANAGEMENT FOR BLDC MOTOR DRIVES USING THE ST72141
AN1130	AN INTRODUCTION TO SENSORLESS BRUSHLESS DC MOTOR DRIVE APPLICATIONS WITH THE ST72141
AN1148	USING THE ST7263 FOR DESIGNING A USB MOUSE
AN1149	HANDLING SUSPEND MODE ON A USB MOUSE
AN1180	USING THE ST7263 KIT TO IMPLEMENT A USB GAME PAD
AN1276	BLDC MOTOR START ROUTINE FOR THE ST72141 MICROCONTROLLER
AN1321	USING THE ST72141 MOTOR CONTROL MCU IN SENSOR MODE
AN1325	USING THE ST7 USB LOW-SPEED FIRMWARE V4.X
AN1445	EMULATED 16-BIT SLAVE SPI
AN1475	DEVELOPING AN ST7265X MASS STORAGE APPLICATION
AN1504	STARTING A PWM SIGNAL DIRECTLY AT HIGH LEVEL USING THE ST7 16-BIT TIMER
AN1602	16-BIT TIMING OPERATIONS USING ST7262 OR ST7263B ST7 USB MCUS
AN1633	DEVICE FIRMWARE UPGRADE (DFU) IMPLEMENTATION IN ST7 NON-USB APPLICATIONS
AN1712	GENERATING A HIGH RESOLUTION SINEWAVE USING ST7 PWMART
AN1713	SMBUS SLAVE DRIVER FOR ST7 I2C PERIPHERALS
AN1753	SOFTWARE UART USING 12-BIT ART

Table 31. ST7 application notes

IDENTIFICATION	DESCRIPTION
AN1947	ST7MC PMAC SINE WAVE MOTOR CONTROL SOFTWARE LIBRARY
GENERAL PURPOSE	
AN1476	LOW COST POWER SUPPLY FOR HOME APPLIANCES
AN1526	ST7FLITE0 QUICK REFERENCE NOTE
AN1709	EMC DESIGN FOR ST MICROCONTROLLERS
AN1752	ST72324 QUICK REFERENCE NOTE
PRODUCT EVALUATION	
AN 910	PERFORMANCE BENCHMARKING
AN 990	ST7 BENEFITS VS INDUSTRY STANDARD
AN1077	OVERVIEW OF ENHANCED CAN CONTROLLERS FOR ST7 AND ST9 MCUS
AN1086	U435 CAN-DO SOLUTIONS FOR CAR MULTIPLEXING
AN1103	IMPROVED B-EMF DETECTION FOR LOW SPEED, LOW VOLTAGE WITH ST72141
AN1150	BENCHMARK ST72 VS PC16
AN1151	PERFORMANCE COMPARISON BETWEEN ST72254 & PC16F876
AN1278	LIN (LOCAL INTERCONNECT NETWORK) SOLUTIONS
PRODUCT MIGRATION	
AN1131	MIGRATING APPLICATIONS FROM ST72511/311/214/124 TO ST72521/321/324
AN1322	MIGRATING AN APPLICATION FROM ST7263 REV.B TO ST7263B
AN1365	GUIDELINES FOR MIGRATING ST72C254 APPLICATIONS TO ST72F264
AN1604	HOW TO USE ST7MDT1-TRAIN WITH ST72F264
AN2200	GUIDELINES FOR MIGRATING ST7LITE1X APPLICATIONS TO ST7FLITE1XB
PRODUCT OPTIMIZATION	
AN 982	USING ST7 WITH CERAMIC RESONATOR
AN1014	HOW TO MINIMIZE THE ST7 POWER CONSUMPTION
AN1015	SOFTWARE TECHNIQUES FOR IMPROVING MICROCONTROLLER EMC PERFORMANCE
AN1040	MONITORING THE VBUS SIGNAL FOR USB SELF-POWERED DEVICES
AN1070	ST7 CHECKSUM SELF-CHECKING CAPABILITY
AN1181	ELECTROSTATIC DISCHARGE SENSITIVE MEASUREMENT
AN1324	CALIBRATING THE RC OSCILLATOR OF THE ST7FLITE0 MCU USING THE MAINS
AN1502	EMULATED DATA EEPROM WITH ST7 HDFLASH MEMORY
AN1529	EXTENDING THE CURRENT & VOLTAGE CAPABILITY ON THE ST7265 VDDF SUPPLY
AN1530	ACCURATE TIMEBASE FOR LOW-COST ST7 APPLICATIONS WITH INTERNAL RC OSCILLATOR
AN1605	USING AN ACTIVE RC TO WAKEUP THE ST7LITE0 FROM POWER SAVING MODE
AN1636	UNDERSTANDING AND MINIMIZING ADC CONVERSION ERRORS
AN1828	PIR (PASSIVE INFRARED) DETECTOR USING THE ST7FLITE05/09/SUPERLITE
AN1946	SENSORLESS BLDC MOTOR CONTROL AND BEMF SAMPLING METHODS WITH ST7MC
AN1953	PFC FOR ST7MC STARTER KIT
AN1971	ST7LITE0 MICROCONTROLLED BALLAST
PROGRAMMING AND TOOLS	
AN 978	ST7 VISUAL DEVELOP SOFTWARE KEY DEBUGGING FEATURES
AN 983	KEY FEATURES OF THE COSMIC ST7 C-COMPILER PACKAGE
AN 985	EXECUTING CODE IN ST7 RAM
AN 986	USING THE INDIRECT ADDRESSING MODE WITH ST7
AN 987	ST7 SERIAL TEST CONTROLLER PROGRAMMING
AN 988	STARTING WITH ST7 ASSEMBLY TOOL CHAIN
AN1039	ST7 MATH UTILITY ROUTINES

Table 31. ST7 application notes

IDENTIFICATION	DESCRIPTION
AN1071	HALF DUPLEX USB-TO-SERIAL BRIDGE USING THE ST72611 USB MICROCONTROLLER
AN1106	TRANSLATING ASSEMBLY CODE FROM HC05 TO ST7
AN1179	PROGRAMMING ST7 FLASH MICROCONTROLLERS IN REMOTE ISP MODE (IN-SITU PROGRAMMING)
AN1446	USING THE ST72521 EMULATOR TO DEBUG AN ST72324 TARGET APPLICATION
AN1477	EMULATED DATA EEPROM WITH XFLASH MEMORY
AN1527	DEVELOPING A USB SMARTCARD READER WITH ST7SCR
AN1575	ON-BOARD PROGRAMMING METHODS FOR XFLASH AND HDFLASH ST7 MCUS
AN1576	IN-APPLICATION PROGRAMMING (IAP) DRIVERS FOR ST7 HDFLASH OR XFLASH MCUS
AN1577	DEVICE FIRMWARE UPGRADE (DFU) IMPLEMENTATION FOR ST7 USB APPLICATIONS
AN1601	SOFTWARE IMPLEMENTATION FOR ST7DALI-EVAL
AN1603	USING THE ST7 USB DEVICE FIRMWARE UPGRADE DEVELOPMENT KIT (DFU-DK)
AN1635	ST7 CUSTOMER ROM CODE RELEASE INFORMATION
AN1754	DATA LOGGING PROGRAM FOR TESTING ST7 APPLICATIONS VIA ICC
AN1796	FIELD UPDATES FOR FLASH BASED ST7 APPLICATIONS USING A PC COMM PORT
AN1900	HARDWARE IMPLEMENTATION FOR ST7DALI-EVAL
AN1904	ST7MC THREE-PHASE AC INDUCTION MOTOR CONTROL SOFTWARE LIBRARY
AN1905	ST7MC THREE-PHASE BLDC MOTOR CONTROL SOFTWARE LIBRARY
SYSTEM OPTIMIZATION	
AN1711	SOFTWARE TECHNIQUES FOR COMPENSATING ST7 ADC ERRORS
AN1827	IMPLEMENTATION OF SIGMA-DELTA ADC WITH ST7FLITE05/09
AN2009	PWM MANAGEMENT FOR 3-PHASE BLDC MOTOR DRIVES USING THE ST7FMC
AN2030	BACK EMF DETECTION DURING PWM ON TIME BY ST7MC

17 Known limitations

17.1 PA2 limitation with OCMP1 enabled

Description

This limitation affects only Rev B Flash devices (with Internal Sales Type 72F63Bxxxxx$x7); it has been corrected in Rev W Flash devices (with Internal Sales Type 72F63Bxxxxx$x9).

Note: Refer to Figure 83 on page 143

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, PA2 is switched to its alternate function mode, SCL. As a consequence, the PA2 pin is forced to be floating (steady level of I^2C clock) even if port configuration (PADDR+PADR) has set it as output low. However, it can be still used as an input or can be controlled by the I^2C cell when enabled (where I^2C is available).

17.2 Unexpected reset fetch

Description

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.

17.3 USB behavior with LVD disabled

Description

On 4K and 8K ROM devices (ST7263BK1M1, ST72BK2M1, ST7263BKB1, ST7263BK2B1 only) if the LVD is disabled, the USB is disabled by hardware. So, the LVD is forced by ST to 0 (LVD enabled). Refer to the ST7263B option list for details.

17.4 I2C multimaster

Description

In multimaster configurations, if the ST7 I2C receives a START condition from another I2C master after the START bit is set in the I2CCR register and before the START condition is generated by the ST7 I2C, it may ignore the START condition from the other I2C master. In this case, the ST7 master will receive a NACK from the other device. On reception of the NACK, ST7 can send a re-start and Slave address to re-initiate communication

17.5 Halt mode power consumption with ADC on

Description

If the A/D converter is being used when Halt mode is entered, the power consumption in Halt mode may exceed the maximum specified in the datasheet.

Workaround

Switch off the ADC by software (ADON=0) before executing a HALT instruction.

17.6 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 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 (fCPU=8MHz 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:

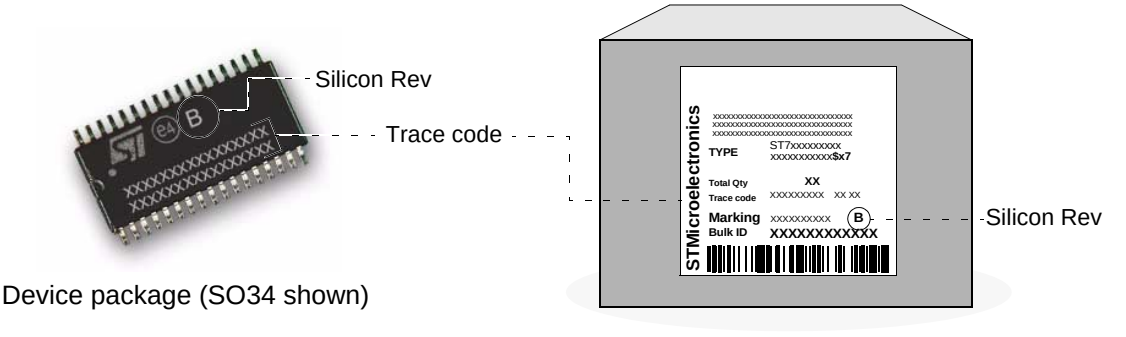
- Disable interrupts
- Reset and Set TE (IDLE request)
- Set and Reset SBK (Break Request)
- Re-enable interrupts

Figure 83. Identifying silicon revision from device marking and box label

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.

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.



Device package (SO34 shown)

Example box label

18 Revision history

Date	Revision	Main changes
27-May-05	3.0	New revision created by merging 32K Flash and non-32K Flash datasheets together Memory Map, Figure 7, expanded to handle all devices and memory sizes Operating conditions with LVD values modified, section 14.3.2 on page 110 Supply current characteristics values and notes updated, section 14.4 on page 111 IDD Run and Wait graphs replaced, Figure 52 and Figure 53 on page 111 Control timing characteristics modified, section 14.5.2 on page 112 Flash memory table notes and t_{PROG} typical value updated, section 14.6.2 on page 114 Notes added for I/O Port Pin characteristics table, section 14.8.1 on page 117 Note for R_{PU} modified, removing reference to data characterization, Section 14.8.1 I_{PU} and R_{PU} graphs added, Figure 58 and Figure 59 on page 118 Notes updated for USB low speed electrical characteristics Output voltage/current graphs added, Figures 60-69 Thermal Characteristics added for SO24 and TQFP48 packages, section 15.2 on page 131 Important note added for OPT 3 Option Byte (LVD), section 16.1 on page 133 Supported Part Numbers table updated with full sales type codes, Table 29 Option List updated with all device options, page 138 Important notes updated with 'USB behavior with LVD disabled', section 17.3 on page 142 Clock block diagram redrawn, Figure 18 on page 26 DFU added to title and features list, page 1 Removed unnecessary notes related to Typical values (already mentioned in section 14.1.2 on page 108) in electrical characteristic tables sections: Section 14.3.2, Section 14.4, Section 14.6.2, Section 14.8.1, Section 14.9.1 and Section 14.11 Added note for max values in ADC Accuracy, Section 14.11 Static Latch Up (LU) class tested only for T_A=25°C, section 14.7.3.2 on page 116
19-Sep-05	4.0	Flash memory minimum data retention increased to 40 years, section 14.6.2 on page 114 AF bit text modified concerning SCL, I2C Chapter Section 12.5.7, "Register description" Reference made to the Flash Programming Reference Manual for Flash timing values Reset pulse generated by WDG changed to 30µs, section 12.1 on page 42 Modified text in section 12.3 on page 64, adding Parity error as an interrupt Added ECOPACK information in section 15 on page 128 Modified I_S value and corresponding note in section 14.8.1 on page 117
06-Apr-06	5.0	32K and 8K QFN40 Packages added 4K SO24 Package added TQFP package renamed to LQFP
03-Oct-06	6.0	Important Notes section renamed to Known Limitations, section 17 on page 142 New PA2 limitation added, section 17 on page 142 Figure 83. on page 143 added for silicon revision identification
20-Aug-07	7.0	Root part numbers added in datasheet header and in Table 1, "Device summary". New 16K LQFP48 package added to product family. Note added to V_{OH} data in section 14.8.2 on page 118 List of supported partnumber availability updated, Table 29, "Supported part numbers" Download address updated in Section 16.3.4, "Order codes for ST7263B development tools" and Option list.

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.

© 2007 STMicroelectronics - All rights reserved

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