



Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.85V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f324lj4t6

Table of Contents

12.7	EMC CHARACTERISTICS	121
12.7.1	Functional EMS (Electro Magnetic Susceptibility)	121
12.7.2	Electro Magnetic Interference (EMI)	122
12.7.3	Absolute Maximum Ratings (Electrical Sensitivity)	123
12.8	I/O PORT PIN CHARACTERISTICS	124
12.8.1	General Characteristics	124
12.8.2	Output Driving Current	125
12.9	CONTROL PIN CHARACTERISTICS	127
12.9.1	Asynchronous RESET Pin	127
12.9.2	ICCSEL/VPP Pin	128
12.10	TIMER PERIPHERAL CHARACTERISTICS	129
12.10.1	16-Bit Timer	129
12.11	COMMUNICATION INTERFACE CHARACTERISTICS	130
12.11.1	SPI - Serial Peripheral Interface	130
12.12	10-BIT ADC CHARACTERISTICS	132
12.12.1	Analog Power Supply and Reference Pins	133
12.12.2	General PCB Design Guidelines	133
12.12.3	ADC Accuracy	135
13	PACKAGE CHARACTERISTICS	136
13.1	PACKAGE MECHANICAL DATA	136
13.2	THERMAL CHARACTERISTICS	138
13.3	SOLDERING INFORMATION	139
14	DEVICE CONFIGURATION AND ORDERING INFORMATION	140
14.1	FLASH OPTION BYTES	140
14.2	DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE	142
14.3	DEVELOPMENT TOOLS	145
14.3.1	Starter kits	145
14.3.2	Development and debugging tools	145
14.3.3	Programming tools	145
14.3.4	Socket and Emulator Adapter Information	146
14.4	ST7 APPLICATION NOTES	147
15	KNOWN LIMITATIONS	149
15.1	ALL FLASH AND ROM DEVICES	149
15.1.1	Safe Connection of OSC1/OSC2 Pins	149
15.1.2	Unexpected Reset Fetch	149
15.1.3	Clearing active interrupts outside interrupt routine	149
15.1.4	16-bit Timer PWM Mode	149
15.1.5	ADC Conversion Spurious Results	149
15.1.6	SCI Wrong Break duration	150
15.1.7	External interrupt missed	150
15.2	ROM DEVICES ONLY	151
15.2.1	I/O Port A and F Configuration	151
15.3	FLASH DEVICES ONLY	152
15.3.1	Timer A Restrictions in Flash Devices	152
15.3.2	External clock source with PLL	152
15.3.3	39-Pulse ICC Entry Mode	152
16	REVISION HISTORY	153

Pin n°				Pin Name	Type	Level		Port						Main function (after reset)	Alternate Function	
LQFP48	LQFP44	LQFP32	SDIP32			Input	Output	Input				Output				
								float	wpu	int	ana	OD	PP			
27	25	10	13	PC2 (HS)/ICAP2_B	I/O	C _T	HS	X	X			X	X	Port C2	Timer B Input Capture 2	
28	26	11	14	PC3 (HS)/ICAP1_B	I/O	C _T	HS	X	X			X	X	Port C3	Timer B Input Capture 1	
29	27	12	15	PC4/MISO/ICCDATA	I/O	C _T		X	X			X	X	Port C4	SPI Master In / Slave Out Data	ICC Data Input
30	28	13	16	PC5/MOSI/AIN14	I/O	C _T		X	X		X	X	X	Port C5	SPI Master Out / Slave In Data	ADC Analog Input 14
31	29	14	17	PC6/SCK/ICCCLK	I/O	C _T		X	X			X	X	Port C6	SPI Serial Clock	ICC Clock Output
32	30	15	18	PC7/ \overline{SS} /AIN15	I/O	C _T		X	X		X	X	X	Port C7	SPI Slave Select (active low)	ADC Analog Input 15
34	31	16	19	PA3 (HS)	I/O	C _T	HS	X		ei0		X	X	Port A3		
35	32			V _{DD_1}	S									Digital Main Supply Voltage ⁵⁾		
36	33			V _{SS_1}	S									Digital Ground Voltage ⁵⁾		
37	34	17	20	PA4 (HS)	I/O	C _T	HS	X	X			X	X	Port A4		
38	35			PA5 (HS)	I/O	C _T	HS	X	X			X	X	Port A5		
39	36	18	21	PA6 (HS)	I/O	C _T	HS	X				T		Port A6 ¹⁾		
40	37	19	22	PA7 (HS)	I/O	C _T	HS	X				T		Port A7 ¹⁾		
41	38	20	23	V _{PP} /ICCSEL	I									Must be tied low. In the flash programming mode, this pin acts as the programming voltage input V _{PP} . See Section 12.9.2 for more details. High voltage must not be applied to ROM devices.		
42	39	21	24	\overline{RESET}	I/O	C _T								Top priority non maskable interrupt.		
43	40	22	25	V _{SS_2}	S									Digital Ground Voltage ⁵⁾		
44	41	23	26	OSC2	O									Resonator oscillator inverter output		
45	42	24	27	OSC1	I									External clock input or Resonator oscillator inverter input		
46	43	25	28	V _{DD_2}	S									Digital Main Supply Voltage ⁵⁾		
47	44	26	29	PE0/TDO	I/O	C _T		X	X			X	X	Port E0	SCI Transmit Data Out	
48	1	27	30	PE1/RDI	I/O	C _T		X	X			X	X	Port E1	SCI Receive Data In	
3	2	28	31	PB0	I/O	C _T		X		ei2		X	X	Port B0		
4	3			PB1	I/O	C _T		X		ei2		X	X	Port B1		
5	4			PB2	I/O	C _T		X		ei2		X	X	Port B2		
6	5	29	32	PB3	I/O	C _T		X		ei2		X	X	Port B3		

Notes:

1. In the interrupt input column, “eiX” defines the associated external interrupt vector. If the weak pull-up column (wpu) is merged with the interrupt column (int), then the I/O configuration is pull-up interrupt input, else the configuration is floating interrupt input.

2. In the open drain output column, “T” defines a true open drain I/O (P-Buffer and protection diode to V_{DD}

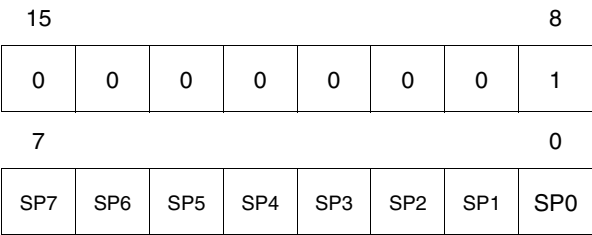
Address	Block	Register Label	Register Name	Reset Status	Remarks
0031h 0032h 0033h 0034h 0035h 0036h 0037h 0038h 0039h 003Ah 003Bh 003Ch 003Dh 003Eh 003Fh	TIMER A	TACR2	Timer A Control Register 2	00h	R/W
		TACR1	Timer A Control Register 1	00h	R/W
		TACSR	Timer A Control/Status Register ³⁾⁴⁾	xxxx x0xxb	R/W
		TAIC1HR	Timer A Input Capture 1 High Register	xxh	Read Only
		TAIC1LR	Timer A Input Capture 1 Low Register	xxh	Read Only
		TAOC1HR	Timer A Output Compare 1 High Register	80h	R/W
		TAOC1LR	Timer A Output Compare 1 Low Register	00h	R/W
		TACHR	Timer A Counter High Register	FFh	Read Only
		TACLR	Timer A Counter Low Register	FFh	Read Only
		TAACHR	Timer A Alternate Counter High Register	FFh	Read Only
		TAACLR	Timer A Alternate Counter Low Register	FFh	Read Only
		TAIC2HR	Timer A Input Capture 2 High Register ³⁾	xxh	Read Only
		TAIC2LR	Timer A Input Capture 2 Low Register ³⁾	xxh	Read Only
		TAOC2HR	Timer A Output Compare 2 High Register ⁴⁾	80h	R/W
		TAOC2LR	Timer A Output Compare 2 Low Register ⁴⁾	00h	R/W
0040h	Reserved Area (1 Byte)				
0041h 0042h 0043h 0044h 0045h 0046h 0047h 0048h 0049h 004Ah 004Bh 004Ch 004Dh 004Eh 004Fh	TIMER B	TBCR2	Timer B Control Register 2	00h	R/W
		TBCR1	Timer B Control Register 1	00h	R/W
		TBCSR	Timer B Control/Status Register	xxxx x0xxb	R/W
		TBIC1HR	Timer B Input Capture 1 High Register	xxh	Read Only
		TBIC1LR	Timer B Input Capture 1 Low Register	xxh	Read Only
		TBOC1HR	Timer B Output Compare 1 High Register	80h	R/W
		TBOC1LR	Timer B Output Compare 1 Low Register	00h	R/W
		TBCHR	Timer B Counter High Register	FFh	Read Only
		TBCLR	Timer B Counter Low Register	FFh	Read Only
		TBACHR	Timer B Alternate Counter High Register	FFh	Read Only
		TBACLR	Timer B Alternate Counter Low Register	FFh	Read Only
		TBIC2HR	Timer B Input Capture 2 High Register	xxh	Read Only
		TBIC2LR	Timer B Input Capture 2 Low Register	xxh	Read Only
		TBOC2HR	Timer B Output Compare 2 High Register	80h	R/W
		TBOC2LR	Timer B Output Compare 2 Low Register	00h	R/W
0050h 0051h 0052h 0053h 0054h 0055h 0056h 0057h	SCI	SCISR	SCI Status Register	C0h	Read Only
		SCIDR	SCI Data Register	xxh	R/W
		SCIBRR	SCI Baud Rate Register	00h	R/W
		SCICR1	SCI Control Register 1	x000 0000h	R/W
		SCICR2	SCI Control Register 2	00h	R/W
		SCIERPR	SCI Extended Receive Prescaler Register	00h	R/W
		SCIETPR	SCI Extended Transmit Prescaler Register	---	R/W
0058h to 006Fh	Reserved Area (24 Bytes)				
0070h 0071h 0072h	ADC	ADCCSR	Control/Status Register	00h	R/W
		ADCDRH	Data High Register	00h	Read Only
		ADCRL	Data Low Register	00h	Read Only
0073h 007Fh	Reserved Area (13 Bytes)				

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 FFh



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 10).

Since the stack is 256 bytes deep, the 8 most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

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

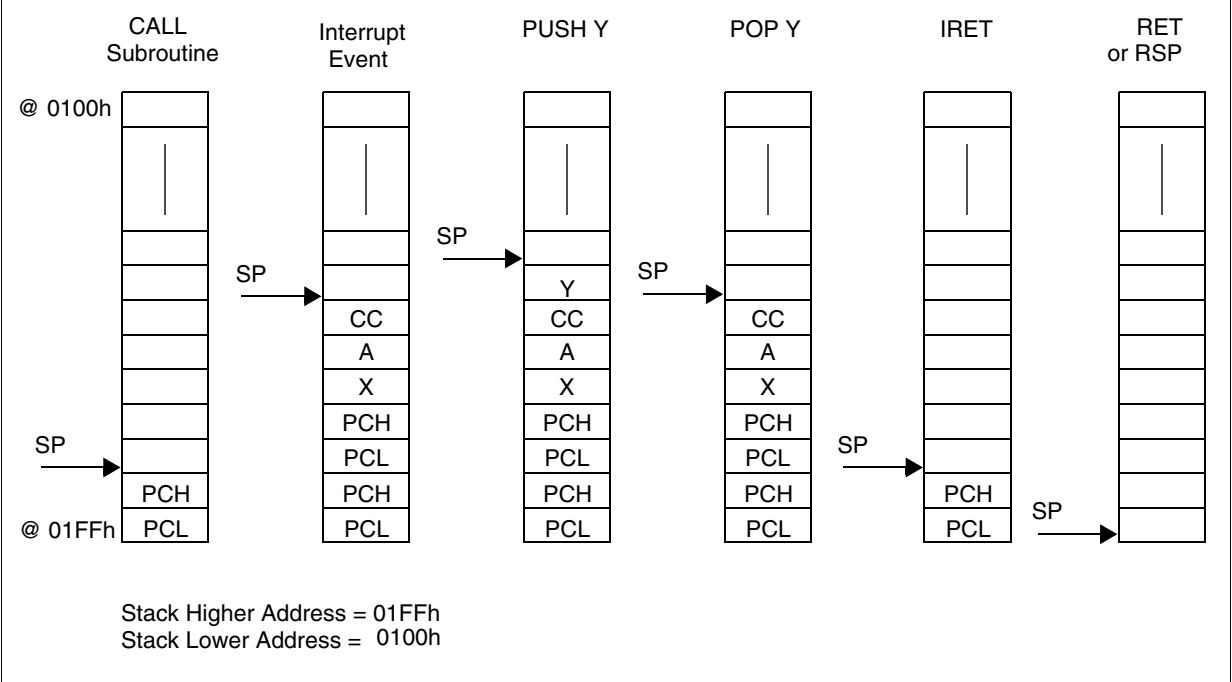
Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an under-flow.

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 10.

- 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 10. Stack Manipulation Example



7 INTERRUPTS

7.1 INTRODUCTION

The ST7 enhanced interrupt management provides the following features:

- Hardware interrupts
- Software interrupt (TRAP)
- Nested or concurrent interrupt management with flexible interrupt priority and level management:
 - Up to 4 software programmable nesting levels
 - Up to 16 interrupt vectors fixed by hardware
 - 2 non maskable events: RESET, TRAP

This interrupt management is based on:

- Bit 5 and bit 3 of the CPU CC register (I1:0),
- Interrupt software priority registers (ISPRx),
- Fixed interrupt vector addresses located at the high addresses of the memory map (FFE0h to FFFFh) sorted by hardware priority order.

This enhanced interrupt controller guarantees full upward compatibility with the standard (not nested) ST7 interrupt controller.

7.2 MASKING AND PROCESSING FLOW

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see Table 7). The processing flow is shown in Figure 14

When an interrupt request 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.
- I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.
- 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 “Interrupt Mapping” table for vector addresses).

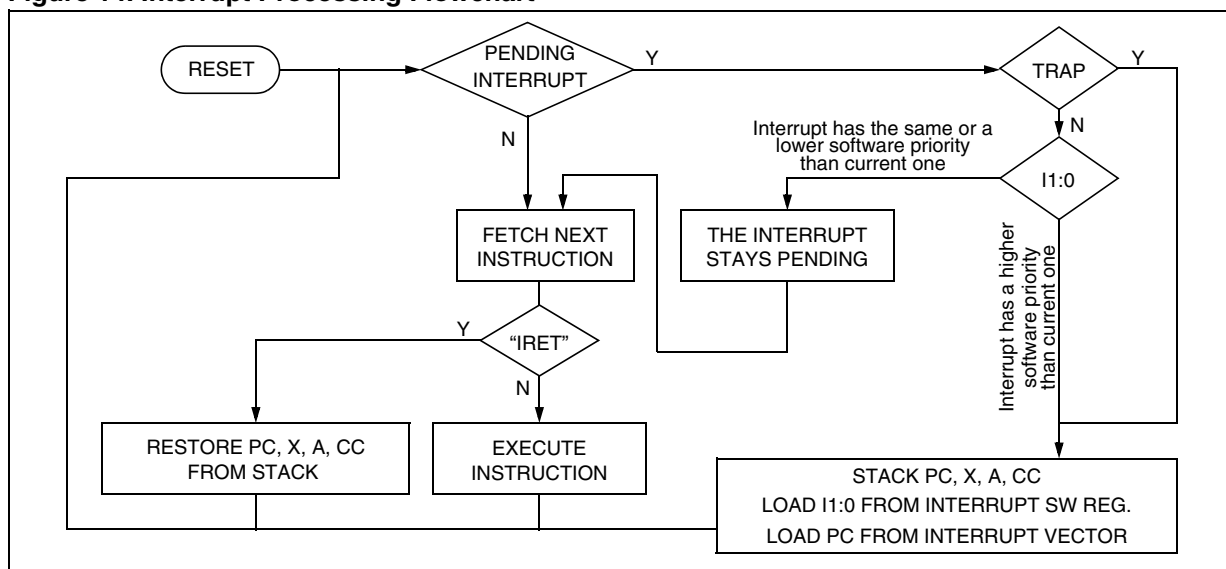
The interrupt service routine should end with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I1 and I0 bits will be restored from the stack and the program in the previous level will resume.

Table 7. Interrupt Software Priority Levels

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low ↓	1	0
Level 1		0	1
Level 2		0	0
Level 3 (= interrupt disable)	High	1	1

Figure 14. Interrupt Processing Flowchart



INTERRUPTS (Cont'd)

Table 8. Dedicated Interrupt Instruction Set

Instruction	New Description	Function/Example	I1	H	I0	N	Z	C
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC	I1	H	I0	N	Z	C
JRM	Jump if I1:0=11 (level 3)	I1:0=11 ?						
JRNM	Jump if I1:0<>11	I1:0<>11 ?						
POP CC	Pop CC from the Stack	Mem => CC	I1	H	I0	N	Z	C
RIM	Enable interrupt (level 0 set)	Load I0 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load I1 in I1:0 of CC	1		1			
TRAP	Software trap	Software NMI	1		1			
WFI	Wait for interrupt		1		0			

Note: During the execution of an interrupt routine, the HALT, POPCC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)**10.5.7 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 is not 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 an Idle Line is detected. An interrupt is generated if the ILIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No Idle Line is detected

1: Idle Line is detected

Note: The IDLE bit is not set again until the RDRF bit has been set itself (that is, 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 is not lost but the shift register is 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 COMMUNICATIONS 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 is 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.

10-BIT A/D CONVERTER (ADC) (Cont'd)

10.6.3 Functional Description

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 V_{AREF} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the Electrical Characteristics 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 allotted time.

10.6.3.1 A/D Converter Configuration

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 ADCCSR register:

- Select the CS[3:0] bits to assign the analog channel to convert.

10.6.3.2 Starting the Conversion

In the ADCCSR register:

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

When a conversion is complete:

- The EOC bit is set by hardware.
- The result is in the ADCDR registers.

A read to the ADCDRH resets the EOC bit.

To read the 10 bits, perform the following steps:

1. Poll the EOC bit
2. Read the ADCDRL register
3. Read the ADCDRH register. This clears EOC automatically.

Note: The data is not latched, so both the low and the high data register must be read before the next conversion is complete, so it is recommended to disable interrupts while reading the conversion result.

To read only 8 bits, perform the following steps:

1. Poll the EOC bit
2. Read the ADCDRH register. This clears EOC automatically.

10.6.3.3 Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[3:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

10.6.4 Low Power Modes

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

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 stabilization time t_{STAB} (see Electrical Characteristics) before accurate conversions can be performed.

10.6.5 Interrupts

None.

10-BIT A/D CONVERTER (ADC) (Cont'd)**10.6.6 Register Description****CONTROL/STATUS REGISTER (ADCCSR)**

Read/Write (Except bit 7 read only)

Reset Value: 0000 0000 (00h)

7				0			
EOC	SPEED	ADON	0	CH3	CH2	CH1	CH0

Bit 7 = **EOC** *End of Conversion*

This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register.

0: Conversion is not complete

1: Conversion complete

Bit 6 = **SPEED** *ADC clock selection*

This bit is set and cleared by software.

0: $f_{ADC} = f_{CPU}/4$ 1: $f_{ADC} = f_{CPU}/2$ Bit 5 = **ADON** *A/D Converter on*

This bit is set and cleared by software.

0: Disable ADC and stop conversion

1: Enable ADC and start conversion

Bit 4 = **Reserved**. Must be kept cleared.Bit 3:0 = **CH[3:0]** *Channel Selection*

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

Channel Pin*	CH3	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
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
AIN12	1	1	0	0
AIN13	1	1	0	1
AIN14	1	1	1	0
AIN15	1	1	1	1

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

DATA REGISTER (ADCDRH)

Read Only

Reset Value: 0000 0000 (00h)

7				0			
D9	D8	D7	D6	D5	D4	D3	D2

Bit 7:0 = **D[9:2]** *MSB of Converted Analog Value***DATA REGISTER (ADCDRL)**

Read Only

Reset Value: 0000 0000 (00h)

7				0			
0	0	0	0	0	0	D1	D0

Bit 7:2 = Reserved. Forced by hardware to 0.

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

10-BIT A/D CONVERTER (Cont'd)

Table 24. ADC Register Map and Reset Values

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

INSTRUCTION SET OVERVIEW (Cont'd)

11.2 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							
Interrupt management	TRAP	WFI	HALT	IRET				
Condition Code Flag modification	SIM	RIM	SCF	RCF				

Using a prebyte

The instructions are described with one to four opcodes.

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.

12.5 CLOCK AND TIMING CHARACTERISTICS

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

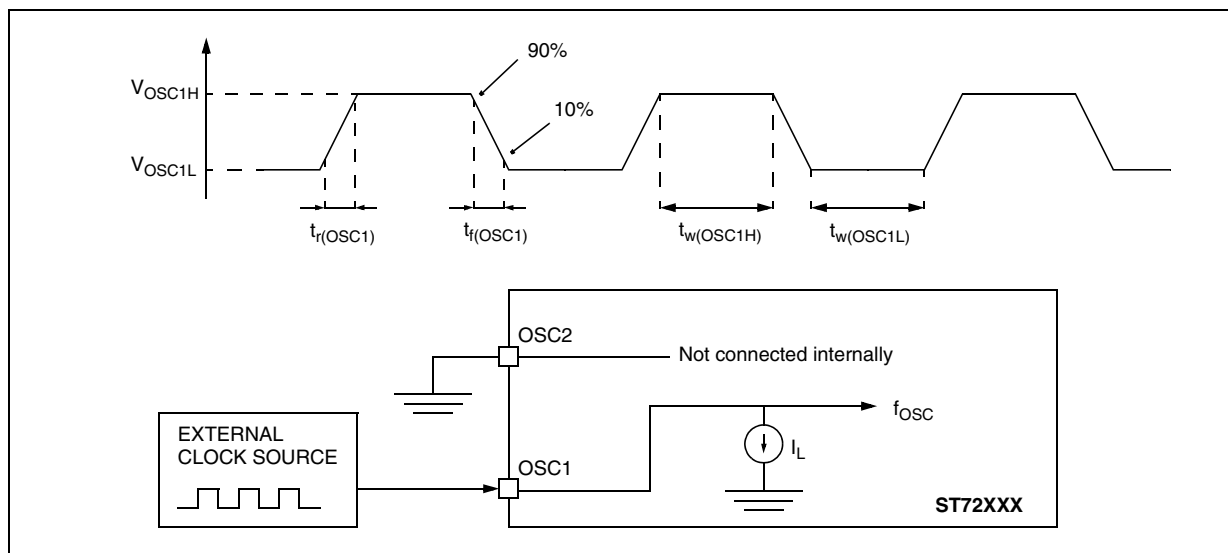
12.5.1 General Timings

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

12.5.2 External Clock Source

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSC1H}	OSC1 input pin high level voltage	see Figure 62	$0.9 \times V_{DD}$		V_{DD}	V
V_{OSC1L}	OSC1 input pin low level voltage		V_{SS}		$0.1 \times V_{DD}$	
$t_w(OSC1H)$ $t_w(OSC1L)$	OSC1 high or low time ³⁾		5			ns
$t_r(OSC1)$ $t_f(OSC1)$	OSC1 rise or fall time ³⁾				15	
I_L	OSC1 Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			± 1	μA

Figure 62. Typical Application with an External Clock Source



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

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

12.5.3 Crystal and Ceramic Resonator Oscillators

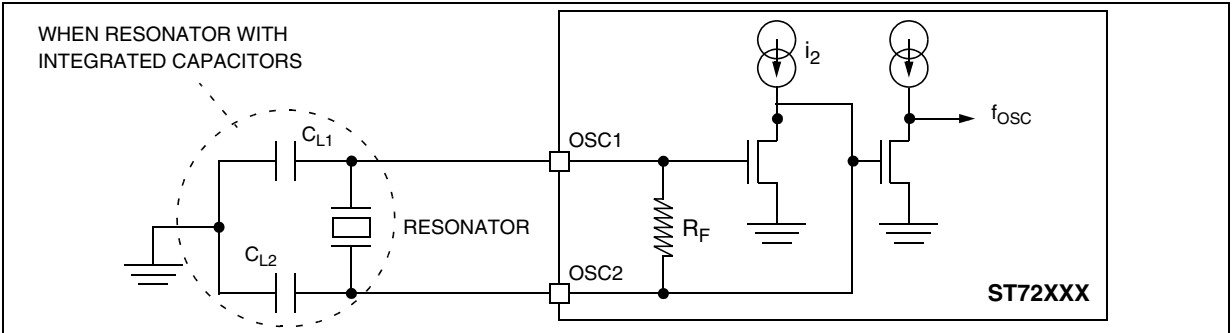
The ST7 internal clock can be supplied with four different Crystal/Ceramic resonator oscillators. All the information given in this paragraph are based on characterization results with specified typical external components. In the application, the resonator and the load capacitors have to be placed as

close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...).

Symbol	Parameter	Conditions	Min	Max	Unit
f_{OSC}	Oscillator Frequency ¹⁾	LP: Low power oscillator MP: Medium power oscillator MS: Medium speed oscillator HS: High speed oscillator	1 >2 >4 >8	2 4 8 16	MHz
R_F	Feedback resistor		20		k Ω
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)	$R_S=200\Omega$ LP osc. (1-2 MHz) $R_S=200\Omega$ MP osc. (2-4 MHz) $R_S=200\Omega$ MS osc. (4-8 MHz) $R_S=100\Omega$ HS osc. (8-16 MHz)	22 22 18 15	56 46 33 33	pF

Symbol	Parameter	Conditions	Typ	Max	Unit
i_2	OSC2 driving current	$V_{IN}=V_{SS}$ LP osc. (1-2 MHz) MP osc. (2-4 MHz) MS osc. (4-8 MHz) HS osc. (8-16 MHz)	80 160 310 610	150 250 460 910	μA

Figure 63. Typical Application with a Crystal or Ceramic Resonator



Notes:

1. The oscillator selection can be optimized in terms of supply current using an high quality resonator with small R_S value. Refer to crystal/ceramic resonator manufacturer for more details.

12.11 COMMUNICATION INTERFACE CHARACTERISTICS

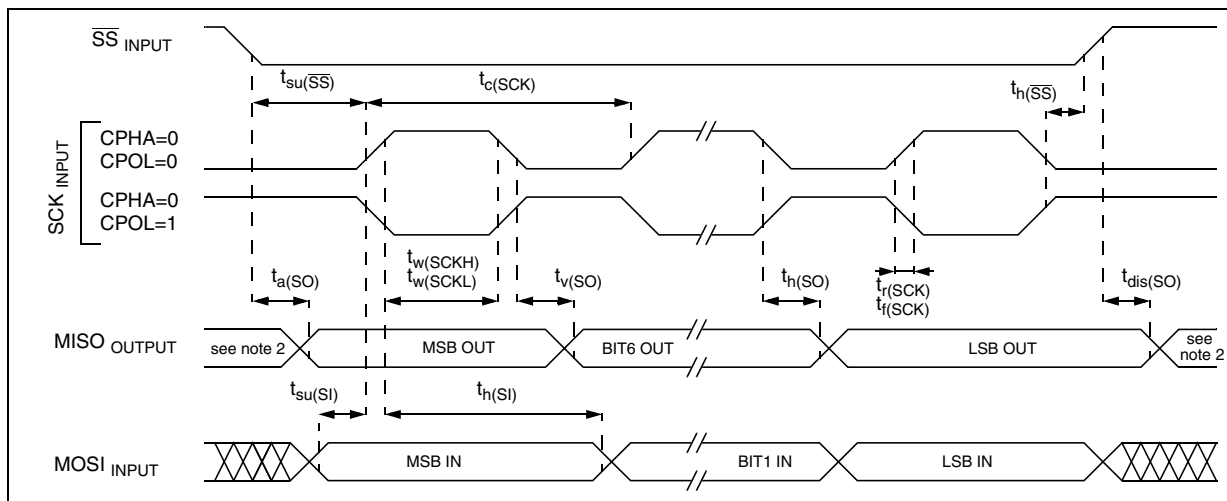
12.11.1 SPI - Serial Peripheral Interface

Subject to general operating conditions 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 (\overline{SS} , SCK, MOSI, MISO).

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master $f_{CPU}=8MHz$	$f_{CPU}/128$ 0.0625	$f_{CPU}/4$ 2	MHz
		Slave $f_{CPU}=8MHz$	0	$f_{CPU}/2$ 4	
$t_r(SCK)$ $t_f(SCK)$	SPI clock rise and fall time		see I/O port pin description		
$t_{su}(\overline{SS})$	\overline{SS} setup time	Slave	120		ns
$t_h(\overline{SS})$	\overline{SS} hold time	Slave	120		
$t_w(SCKH)$ $t_w(SCKL)$	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ $t_{su(SI)}$	Data input setup time	Master Slave	100 100		
$t_h(MI)$ $t_h(SI)$	Data input hold time	Master Slave	100 100		
$t_a(SO)$	Data output access time	Slave	0	120	
$t_{dis}(SO)$	Data output disable time	Slave		240	
$t_v(SO)$	Data output valid time	Slave (after enable edge)		90	
$t_h(SO)$	Data output hold time		0		
$t_v(MO)$	Data output valid time	Master (before capture edge)	0.25		t_{CPU}
$t_h(MO)$	Data output hold time		0.25		

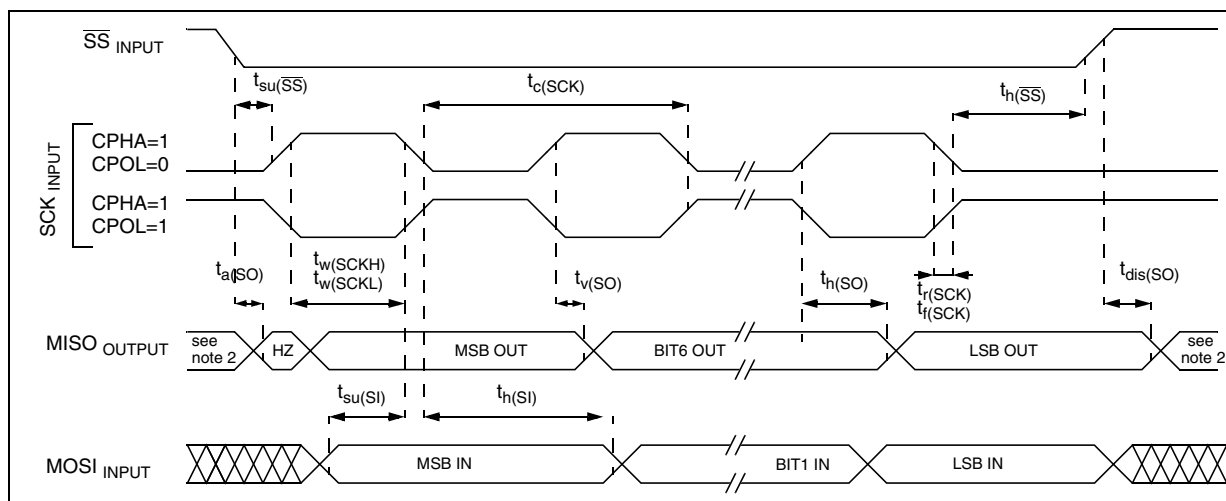
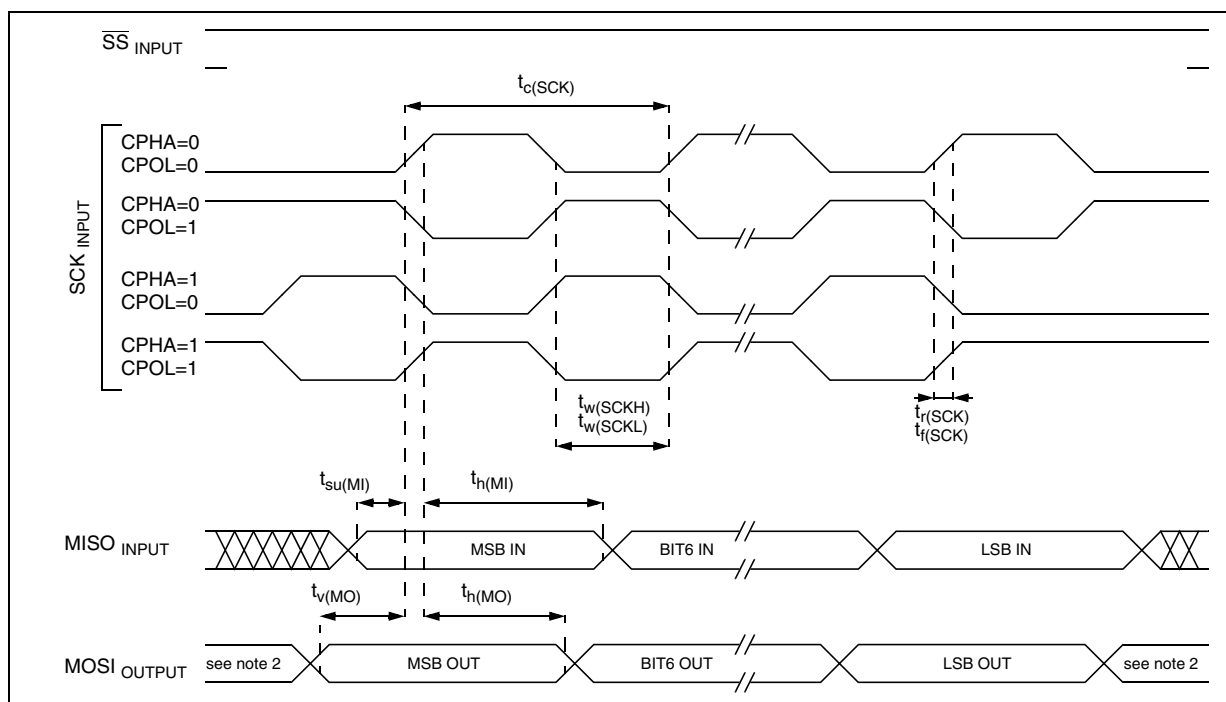
Figure 75. SPI Slave Timing Diagram with $CPHA=0$ ³⁾



Notes:

1. Data based on design simulation and/or characterisation results, not tested in production.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.
3. Measurement points are done at CMOS levels: $0.3 \times V_{DD}$ and $0.7 \times V_{DD}$.

COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

Figure 76. SPI Slave Timing Diagram with CPHA=1¹⁾Figure 77. SPI Master Timing Diagram¹⁾**Notes:**

1. Measurement points are done at CMOS levels: $0.3 \times V_{DD}$ and $0.7 \times V_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

14.3.4 Socket and Emulator Adapter Information

For information on the type of socket that is supplied with the emulator, refer to the suggested list of sockets in Table 29.

Note: Before designing the board layout, it is recommended to check the overall dimensions of the

Note:

- 1. Flash Programming interface for FLASH devices.

socket as they may be greater than the dimensions of the device.

For footprint and other mechanical information about these sockets and adapters, refer to the manufacturer’s datasheet (www.yamaichi.de for LQFP44 10 x 10 and www.ironwoodelectronics.com for LQFP32 7 x 7).

Table 29. Suggested List of Socket Types

Device	Socket (supplied with ST7MDT20J-EMU3)	Emulator Adapter (supplied with ST7MDT20J-EMU3)
LQFP32 7 X 7	IRONWOOD SF-QFE32SA-L-01	IRONWOOD SK-UGA06/32A-01
LQFP44 10 X10	YAMAICHI IC149-044-*52-*5	YAMAICHI ICP-044-5
LQFP48 7 X7	CAB 3303238	CAB 3303333

15 KNOWN LIMITATIONS

15.1 ALL FLASH AND ROM DEVICES

15.1.1 Safe Connection of OSC1/OSC2 Pins

The OSC1 and/or OSC2 pins must not be left unconnected otherwise the ST7 main oscillator may start and, in this configuration, could generate an f_{OSC} clock frequency in excess of the allowed maximum (>16MHz.), putting the ST7 in an unsafe/undefined state. Refer to Section 6.2 on page 24.

15.1.2 Unexpected Reset Fetch

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

Workaround

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

15.1.3 Clearing active interrupts outside interrupt routine

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

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

Concurrent interrupt context

The symptom does not occur when the interrupts are handled normally, i.e.

when:

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

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

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

Example:

```
SIM
reset interrupt flag
RIM
```

Nested interrupt context:

The symptom does not occur when the interrupts are handled normally, i.e.

when:

- The interrupt flag is cleared within its own interrupt routine
- The interrupt flag is cleared within any interrupt routine with higher or identical priority level
- The interrupt flag is cleared in any part of the code while this interrupt is disabled

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

```
PUSH CC
SIM
reset interrupt flag
POP CC
```

15.1.4 16-bit Timer PWM Mode

In PWM mode, the first PWM pulse is missed after writing the value FFFCh in the OC1R register (OC1HR, OC1LR). It leads to either full or no PWM during a period, depending on the OLVL1 and OLVL2 settings.

15.1.5 ADC Conversion Spurious Results

Spurious conversions occur with a rate lower than 50 per million. Such conversions happen when the measured voltage is just between 2 consecutive digital values.

Workaround

A software filter should be implemented to remove erratic conversion results whenever they may cause unwanted consequences.

In order to have the accuracy specified in the datasheet, the first conversion after a ADC switch-on has to be ignored.

16 REVISION HISTORY

Table 31. Revision History

Date	Revision	Description of Changes
01-Oct-2003	1.0	First release.
01-Jan-2004	1.1	<p>Changed Flash & ROM voltage range 2.85 to 3.6V in Section 12.3 on page 112</p> <p>Changed OSC1 high & low level in Section 12.5.1 on page 116</p> <p>Updated "EMC CHARACTERISTICS" on page 121</p> <p>Changed I/O Port: V_{OL}/V_{OH} values in Section 12.8.2 on page 125.</p> <p>Changed Reset pin: $t_{w(RSTL)out}$ & V_{OL} in Section 12.9.1 on page 127.</p> <p>Changed ADC Vref & Ilkg in Section 12.12 on page 132 and ADC accuracy Section 12.12.3 on page 135.</p> <p>Removed SDIP42 package</p>
01-Feb-2004	1.2	<p>Max. values TBD in Section 12.12.3 on page 135</p> <p>Removed 10pf cap. in Figure 81 on page 134</p> <p>Added Section 15.1.5 on page 149</p>
14-Jan-2005	3	<p>Revision number incremented from 1.2 to 3 due to Internal Document Management System change</p> <p>Modified I/O V_{IL} in Section 12.8 on page 124</p> <p>Modified R_{ON} in Section 12.9.1 on page 127</p> <p>Added Clearing active interrupts in Section 15.1.3 on page 149</p> <p>Modified Figure 8 and note 4 in "FLASH PROGRAM MEMORY" on page 17</p> <p>Added limitation on ICC entry mode with 39 pulses to "KNOWN LIMITATIONS" on page 149</p>
05-Mar-2007	4	Added LQFP48 package devices.
12-Sep-2007	5	<p>Added footnote 5 to Table 2 on page 10</p> <p>Updated LQFP48 device ordering information in Table 27 on page 143.</p>