



Welcome to [E-XFL.COM](https://www.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	I <sup>2</sup> C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/stmicroelectronics/st72f325k4t6">https://www.e-xfl.com/product-detail/stmicroelectronics/st72f325k4t6</a>

---

## Table of Contents

---

8.3	WAIT MODE	46
8.4	ACTIVE-HALT AND HALT MODES	47
8.4.1	ACTIVE-HALT MODE	47
8.4.2	HALT MODE	48
<b>9</b>	<b>I/O PORTS</b>	<b>50</b>
9.1	INTRODUCTION	50
9.2	FUNCTIONAL DESCRIPTION	50
9.2.1	Input Modes	50
9.2.2	Output Modes	50
9.2.3	Alternate Functions	50
9.3	I/O PORT IMPLEMENTATION	53
9.4	LOW POWER MODES	53
9.5	INTERRUPTS	53
9.5.1	I/O Port Implementation	54
<b>10</b>	<b>ON-CHIP PERIPHERALS</b>	<b>56</b>
10.1	WATCHDOG TIMER (WDG)	56
10.1.1	Introduction	56
10.1.2	Main Features	56
10.1.3	Functional Description	56
10.1.4	How to Program the Watchdog Timeout	57
10.1.5	Low Power Modes	59
10.1.6	Hardware Watchdog Option	59
10.1.7	Using Halt Mode with the WDG (WDGHALT option)	59
10.1.8	Interrupts	59
10.1.9	Register Description	59
10.2	MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC)	61
10.2.1	Programmable CPU Clock Prescaler	61
10.2.2	Clock-out Capability	61
10.2.3	Real Time Clock Timer (RTC)	61
10.2.4	Beeper	61
10.2.5	Low Power Modes	62
10.2.6	Interrupts	62
10.2.7	Register Description	62
10.3	PWM AUTO-RELOAD TIMER (ART)	64
10.3.1	Introduction	64
10.3.2	Functional Description	65
10.3.3	Register Description	69
10.4	16-BIT TIMER	73
10.4.1	Introduction	73
10.4.2	Main Features	73
10.4.3	Functional Description	73
10.4.4	Low Power Modes	85
10.4.5	Interrupts	85
10.4.6	Summary of Timer Modes	85
10.4.7	Register Description	86
10.5	SERIAL PERIPHERAL INTERFACE (SPI)	92

## 1 DESCRIPTION

The ST72F325 Flash and ST72325 ROM devices are members of the ST7 microcontroller family designed for mid-range applications.

They are derivatives of the ST72321 and ST72324 devices, with enhanced characteristics and robust Clock Security System.

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

The on-chip peripherals include an A/D converter, a PWM Autoreload timer, 2 general purpose timers, I<sup>2</sup>C bus, SPI interface and an SCI interface.

For power economy, microcontroller can switch dynamically into WAIT, SLOW, ACTIVE-HALT or

HALT mode when the application is in idle or stand-by state.

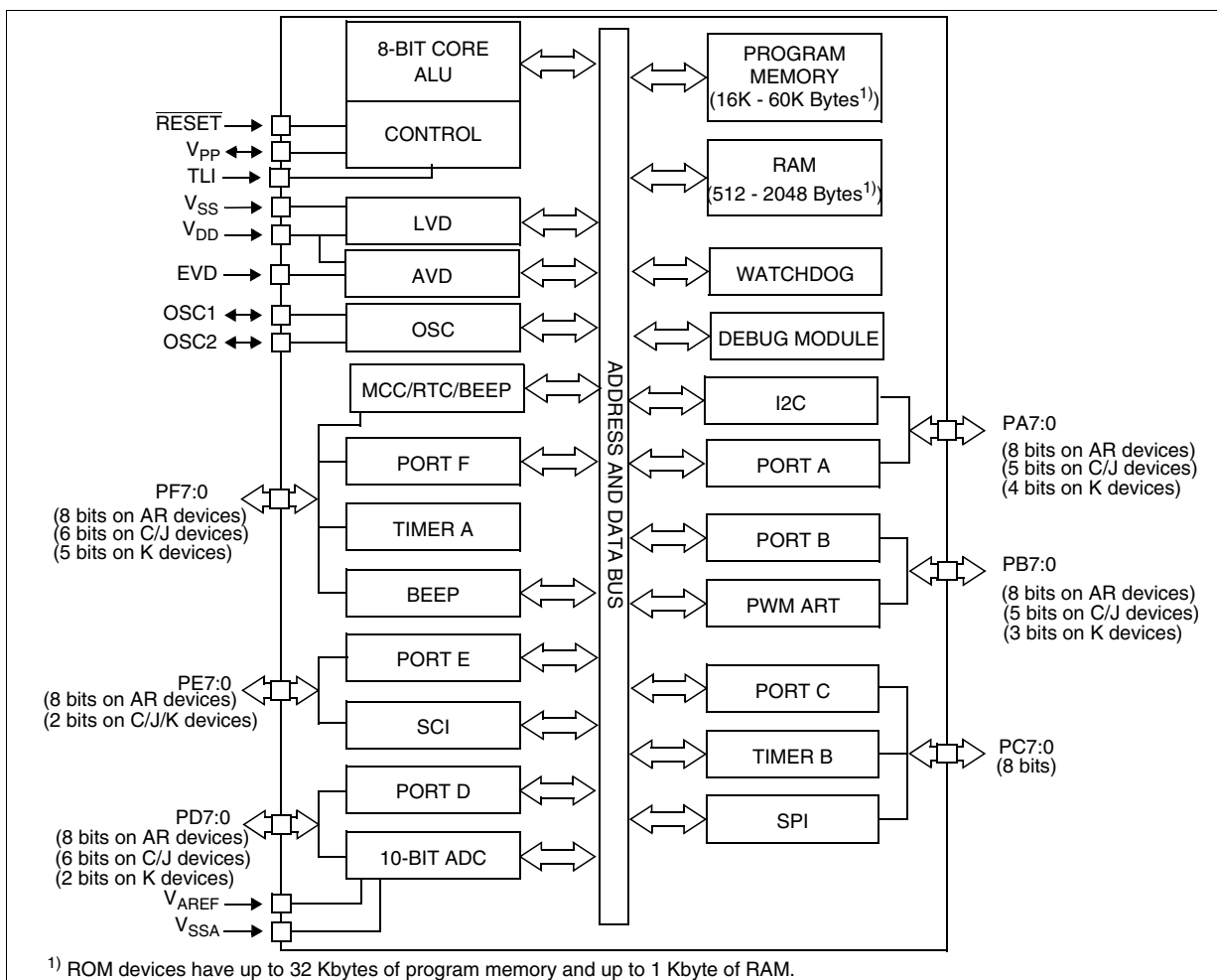
Typical applications are consumer, home, office and industrial products.

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

### Main Differences with ST72321:

- LQFP48 and LQFP32 packages
- Clock Security System
- Internal RC, Readout protection, LVD and PLL without limitations
- Negative current injection not allowed on I/O port PB0 (instead of PC6).
- External interrupts have Exit from Active Halt mode capability.

**Figure 1. Device Block Diagram**



Pin n°		Pin Name	Type	Level		Port						Main function (after reset)	Alternate function
LQFP32	DIP32			Input	Output	Input				Output			
						float	wpu	int	ana	OD	PP		
28	31	PB0/PWM3	I/O	C <sub>T</sub>		X	ei2			X	X	Port B0	PWM Output 3
													<b>Caution:</b> Negative current injection not allowed on this pin
29	32	PB3/PWM0	I/O	C <sub>T</sub>		X		ei2		X	X	Port B3	PWM Output 0
30	1	PB4 (HS)/ARTCLK	I/O	C <sub>T</sub>	HS	X	ei3			X	X	Port B4	PWM-ART External Clock
31	2	PD0/AIN0	I/O	C <sub>T</sub>		X	X		X	X	X	Port D0	ADC Analog Input 0
32	3	PD1/AIN1	I/O	C <sub>T</sub>		X	X		X	X	X	Port D1	ADC Analog Input 1

**Notes for Table 2 and Table 3:**

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<sub>DD</sub> are not implemented). See "I/O PORTS" on page 50. and Section 12.8 I/O PORT PIN CHARACTERISTICS for more details.
3. OSC1 and OSC2 pins connect a crystal/ceramic resonator, or an external source to the on-chip oscillator; see Section 1 DESCRIPTION and Section 12.5 CLOCK AND TIMING CHARACTERISTICS for more details.
4. On the chip, each I/O port may have up to 8 pads:
  - In all devices except 48-pin ST72325C, pads that are not bonded to external pins are forced by hardware in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.
  - In 48-pin ST72325C devices, unbonded pads PA0, PA1, PB6, PB7, PD6, PD7, PE3, PE5, PE6, PE7, PF3 and PF5) are in **input floating** configuration after reset. To avoid added current consumption, the application must force these ports in input pull-up state by writing to the OR and DDR registers after reset. This initialization is not necessary in 48-pin ST72325S devices.
5. Pull-up always activated on PE2 see limitation Section 15.1.8.
6. It is mandatory to connect all available V<sub>DD</sub> and V<sub>REF</sub> pins to the supply voltage and all V<sub>SS</sub> and V<sub>SSA</sub> pins to ground.

## SYSTEM INTEGRITY MANAGEMENT (Cont'd)

## 6.4.2 Auxiliary Voltage Detector (AVD)

The Voltage Detector function (AVD) is based on an analog comparison between a  $V_{IT-(AVD)}$  and  $V_{IT+(AVD)}$  reference value and the  $V_{DD}$  main supply or the external EVD pin voltage level ( $V_{EVD}$ ). The  $V_{IT-}$  reference value for falling voltage is lower than the  $V_{IT+}$  reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real time status bit (AVDF) in the SICSR register. This bit is read only.

**Caution:** The AVD function is active only if the LVD is enabled through the option byte.

6.4.2.1 Monitoring the  $V_{DD}$  Main Supply

This mode is selected by clearing the AVDS bit in the SICSR register.

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see section 14.1 on page 181).

If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the  $V_{IT+(AVD)}$  or  $V_{IT-(AVD)}$  threshold (AVDF bit toggles).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See Figure 18.

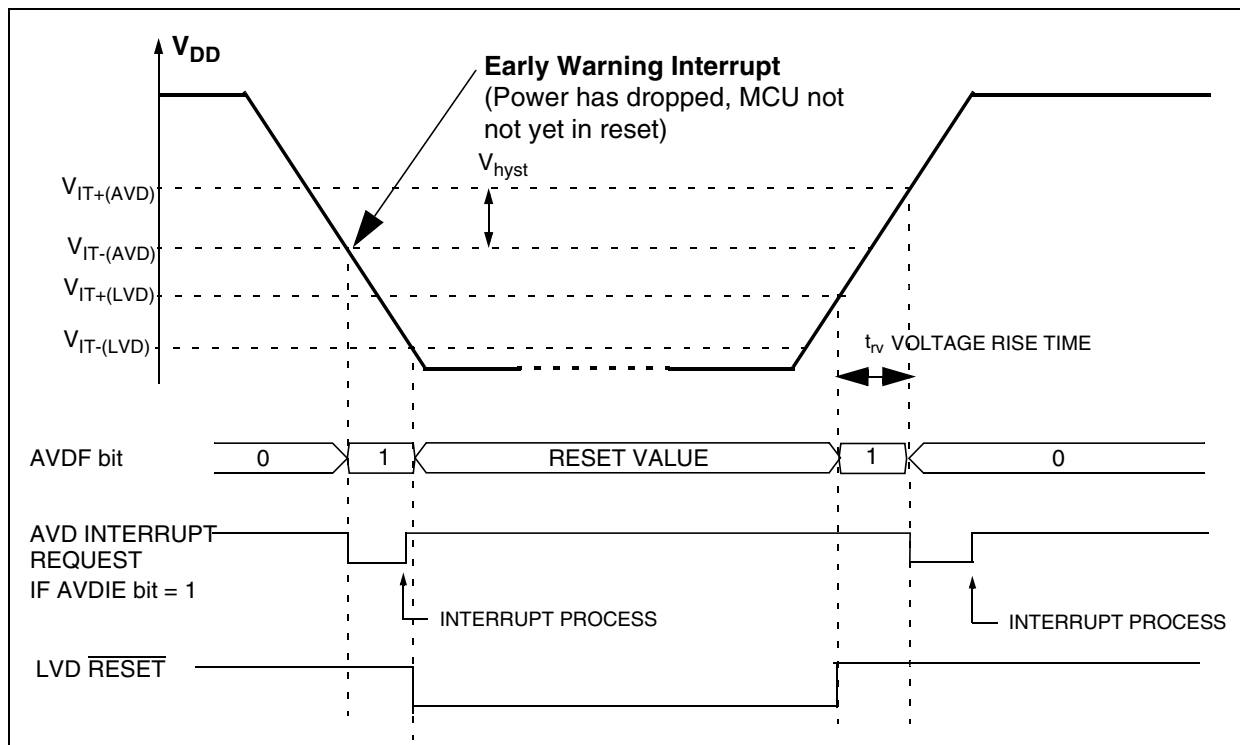
The interrupt on the rising edge is used to inform the application that the  $V_{DD}$  warning state is over.

If the voltage rise time  $t_{rv}$  is less than 256 or 4096 CPU cycles (depending on the reset delay selected by option byte), no AVD interrupt will be generated when  $V_{IT+(AVD)}$  is reached.

If  $t_{rv}$  is greater than 256 or 4096 cycles then:

- If the AVD interrupt is enabled before the  $V_{IT+(AVD)}$  threshold is reached, then 2 AVD interrupts will be received: the first when the AVDIE bit is set, and the second when the threshold is reached.
- If the AVD interrupt is enabled after the  $V_{IT+(AVD)}$  threshold is reached then only one AVD interrupt will occur.

Figure 18. Using the AVD to Monitor  $V_{DD}$  (AVDS bit=0)



## 7.7 EXTERNAL INTERRUPT CONTROL REGISTER (EICR)

Read/Write

Reset Value: 0000 0000 (00h)

7				0			
IS11	IS10	IPB	IS21	IS20	IPA	TLIS	TLIE

Bit 7:6 = **IS1[1:0]** *ei2 and ei3 sensitivity*

The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the following external interrupts:  
- ei2 (port B3..0)

IS11	IS10	External Interrupt Sensitivity	
		IPB bit =0	IPB bit =1
0	0	Falling edge & low level	Rising edge & high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

- ei3 (port B4)

IS11	IS10	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 5 = **IPB** *Interrupt polarity for port B*

This bit is used to invert the sensitivity of the port B [3:0] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3).

0: No sensitivity inversion  
1: Sensitivity inversion

Bit 4:3 = **IS2[1:0]** *ei0 and ei1 sensitivity*

The interrupt sensitivity, defined using the IS2[1:0] bits, is applied to the following external interrupts:

- ei0 (port A3)

IS21	IS20	External Interrupt Sensitivity	
		IPA bit =0	IPA bit =1
0	0	Falling edge & low level	Rising edge & high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

- ei1 (port F2..0)

IS21	IS20	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 2 = **IPA** *Interrupt polarity for port A*

This bit is used to invert the sensitivity of the port A [3:0] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3).

0: No sensitivity inversion  
1: Sensitivity inversion

Bit 1 = **TLIS** *TLI sensitivity*

This bit allows to toggle the TLI edge sensitivity. It can be set and cleared by software only when TLIE bit is cleared.

0: Falling edge  
1: Rising edge

Bit 0 = **TLIE** *TLI enable*

This bit allows to enable or disable the TLI capability on the dedicated pin. It is set and cleared by software.

0: TLI disabled  
1: TLI enabled

**Note:** a parasitic interrupt can be generated when clearing the TLIE bit.

**WATCHDOG TIMER (Cont'd)****10.1.5 Low Power Modes**

Mode	Description		
SLOW	No effect on Watchdog.		
WAIT	No effect on Watchdog.		
HALT	<b>OIE bit in MCCR register</b>	<b>WDGHALT bit in Option Byte</b>	
	0	0	No Watchdog reset is generated. The MCU enters Halt mode. The Watchdog counter is decremented once and then stops counting and is no longer able to generate a watchdog reset until the MCU receives an external interrupt or a reset. If an external interrupt is received, the Watchdog restarts counting after 256 or 4096 CPU clocks. If a reset is generated, the Watchdog is disabled (reset state) unless Hardware Watchdog is selected by option byte. For application recommendations see Section 0.1.7 below.
	0	1	A reset is generated.
	1	x	No reset is generated. The MCU enters Active Halt mode. The Watchdog counter is not decremented. It stop counting. When the MCU receives an oscillator interrupt or external interrupt, the Watchdog restarts counting immediately. When the MCU receives a reset the Watchdog restarts counting after 256 or 4096 CPU clocks.

**10.1.6 Hardware Watchdog Option**

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

**10.1.7 Using Halt Mode with the WDG (WDGHALT option)**

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

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

**10.1.8 Interrupts**

None.

**10.1.9 Register Description****CONTROL REGISTER (WDGCR)**

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

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

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

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

## 10.4 16-BIT TIMER

### 10.4.1 Introduction

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

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

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

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

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

### 10.4.2 Main Features

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

The Block Diagram is shown in Figure 1.

**\*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'.

### 10.4.3 Functional Description

#### 10.4.3.1 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 1. 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.



## 10.5 SERIAL PERIPHERAL INTERFACE (SPI)

### 10.5.1 Introduction

The Serial Peripheral Interface (SPI) allows full-duplex, synchronous, serial communication with external devices. An SPI system may consist of a master and one or more slaves however the SPI interface can not be a master in a multi-master system.

### 10.5.2 Main Features

- Full duplex synchronous transfers (on 3 lines)
- Simplex synchronous transfers (on 2 lines)
- Master or slave operation
- Six master mode frequencies ( $f_{CPU}/4$  max.)
- $f_{CPU}/2$  max. slave mode frequency (see note)
- $\overline{SS}$  Management by software or hardware
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision, Master Mode Fault and Overrun flags

**Note:** In slave mode, continuous transmission is not possible at maximum frequency due to the software overhead for clearing status flags and to initiate the next transmission sequence.

### 10.5.3 General Description

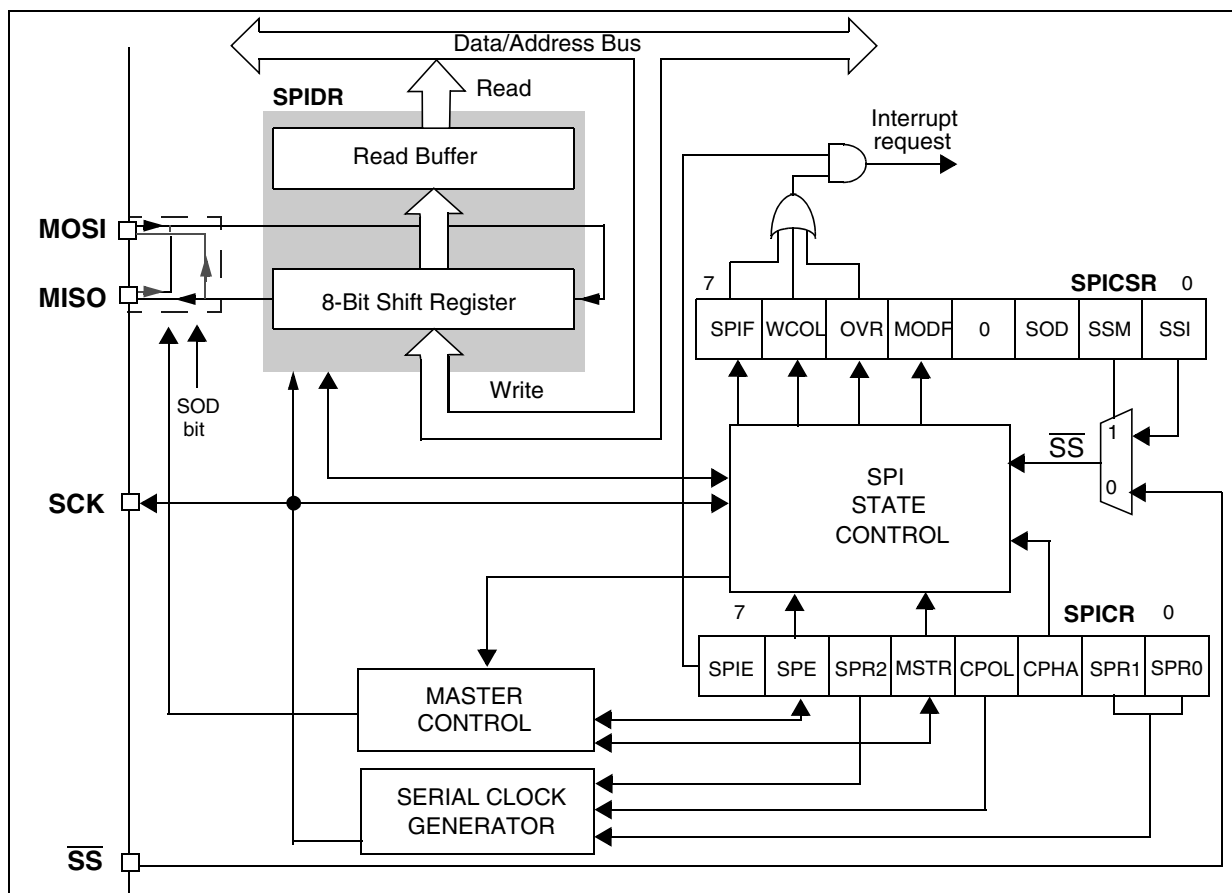
Figure 56 shows the serial peripheral interface (SPI) block diagram. There are 3 registers:

- SPI Control Register (SPICR)
- SPI Control/Status Register (SPICSR)
- SPI Data Register (SPIDR)

The SPI is connected to external devices through 4 pins:

- MISO: Master In / Slave Out data
- MOSI: Master Out / Slave In data
- SCK: Serial Clock out by SPI masters and input by SPI slaves

Figure 56. Serial Peripheral Interface Block Diagram



## SERIAL PERIPHERAL INTERFACE (Cont'd)

### 10.5.3.3 Master Mode Operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

**Note:** The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL=1 or pulling down SCK if CPOL=0).

To operate the SPI in master mode, perform the following steps in order (**if the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account**):

- Write to the SPICR register:
  - Select the clock frequency by configuring the SPR[2:0] bits.
  - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. Figure 60 shows the four possible configurations.  
**Note:** The slave must have the same CPOL and CPHA settings as the master.
- Write to the SPICSR register:
  - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the SS pin high for the complete byte transmit sequence.
- Write to the SPICR register:
  - Set the MSTR and SPE bits  
**Note:** MSTR and SPE bits remain set only if SS is high).

The transmit sequence begins when software writes a byte in the SPIDR register.

### 10.5.3.4 Master Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

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

- An access to the SPICSR register while the SPIF bit is set
- A read to the SPIDR register.

**Note:** While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

### 10.5.3.5 Slave Mode Operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

- Write to the SPICSR register to perform the following actions:
  - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see Figure 60).  
**Note:** The slave must have the same CPOL and CPHA settings as the master.
  - Manage the  $\overline{SS}$  pin as described in Section 10.5.3.2 and Figure 58. If CPHA=1  $\overline{SS}$  must be held low continuously. If CPHA=0  $\overline{SS}$  must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
- Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

### 10.5.3.6 Slave Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

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

- An access to the SPICSR register while the SPIF bit is set.
- A write or a read to the SPIDR register.

**Notes:** While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see Section 10.5.5.2).

## SERIAL PERIPHERAL INTERFACE (Cont'd)

Table 21. SPI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0021h	<b>SPIDR</b> Reset Value	MSB x	x	x	x	x	x	x	LSB x
0022h	<b>SPICR</b> Reset Value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0023h	<b>SPICSR</b> Reset Value	SPIF 0	WCOL 0	OVR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

**SERIAL COMMUNICATIONS 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 wake-up 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 sends a BREAK word at the end of the current word.

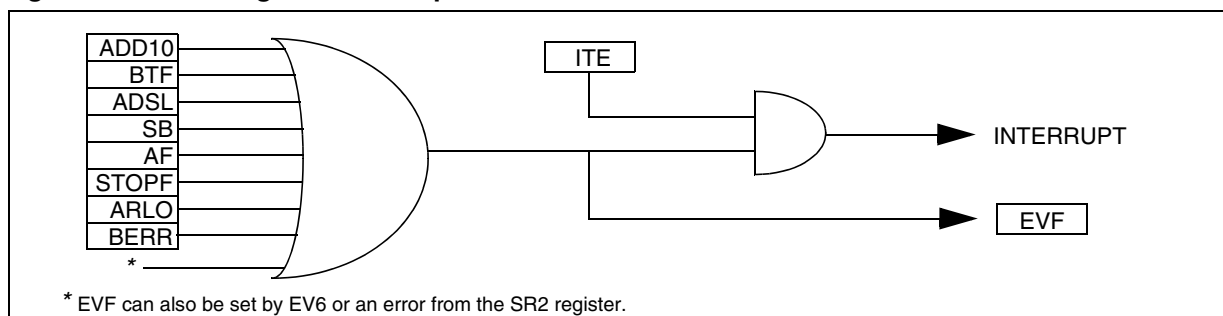
I<sup>2</sup>C BUS INTERFACE (Cont'd)

## 10.7.5 Low Power Modes

Mode	Description
WAIT	No effect on I <sup>2</sup> C interface. I <sup>2</sup> C interrupts cause the device to exit from WAIT mode.
HALT	I <sup>2</sup> C registers are frozen. In HALT mode, the I <sup>2</sup> C interface is inactive and does not acknowledge data on the bus. The I <sup>2</sup> C interface resumes operation when the MCU is woken up by an interrupt with "exit from HALT mode" capability.

## 10.7.6 Interrupts

Figure 70. Event Flags and Interrupt Generation



Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
10-bit Address Sent Event (Master mode)	ADD10	ITE	Yes	No
End of Byte Transfer Event	BTF		Yes	No
Address Matched Event (Slave mode)	ADSEL		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

**Note:** The I<sup>2</sup>C 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).

## INSTRUCTION SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	I1	H	I0	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	I1	H	I0	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	I1:0 = 10 (level 0)			1		0			
RLC	Rotate left true C	C <= A <= C	reg, M					N	Z	C
RRC	Rotate right true C	C => A => 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	I1:0 = 11 (level 3)			1		1			
SLA	Shift left Arithmetic	C <= A <= 0	reg, M					N	Z	C
SLL	Shift left Logic	C <= A <= 0	reg, M					N	Z	C
SRL	Shift right Logic	0 => A => C	reg, M					0	Z	C
SRA	Shift right Arithmetic	A7 => A => C	reg, M					N	Z	C
SUB	Subtraction	A = A - M	A	M				N	Z	C
SWAP	SWAP nibbles	A7-A4 <=> A3-A0	reg, M					N	Z	
TNZ	Test for Neg & Zero	tnz lbl1						N	Z	
TRAP	S/W trap	S/W interrupt			1		1			
WFI	Wait for Interrupt				1		0			
XOR	Exclusive OR	A = A XOR M	A	M				N	Z	

**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 is 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...).

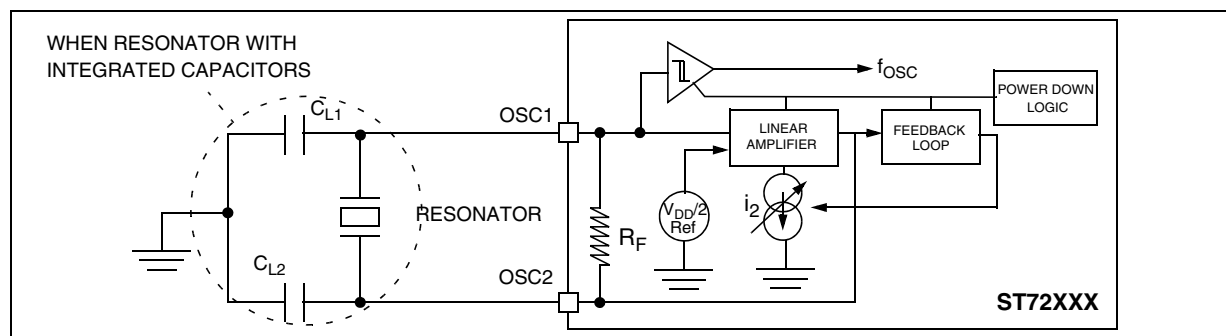
Symbol	Parameter	Conditions	Min	Max	Unit
$f_{OSC}$	Oscillator Frequency <sup>1)</sup>		1	16	MHz
$R_F$	Feedback resistor <sup>2)</sup>		20	40	k $\Omega$
$C_{L1}$ $C_{L2}$	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator ( $R_S$ ) <sup>3)</sup>	$f_{OSC} = 1$ to 2 MHz $f_{OSC} = 2$ to 4 MHz $f_{OSC} = 4$ to 8 MHz $f_{OSC} = 8$ to 16 MHz	20 20 15 15	60 50 35 35	pF

Symbol	Parameter	Conditions	Typ	Max	Unit
$i_2$	OSC2 driving current	$V_{DD}=5V$ : $f_{OSC} = 2MHz$ , $C_0 = 6pF$ , $C_{I1} = C_{I2} = 68pF$ $f_{OSC} = 4MHz$ , $C_0 = 6pF$ , $C_{I1} = C_{I2} = 68pF$ $f_{OSC} = 8MHz$ , $C_0 = 6pF$ , $C_{I1} = C_{I2} = 40pF$ $f_{OSC} = 16MHz$ , $C_0 = 7pF$ , $C_{I1} = C_{I2} = 20pF$	426 425 456 660		$\mu A$

**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.
2. Data based on characterisation results, not tested in production.

Figure 76. Typical Application with a Crystal or Ceramic Resonator



1. The relatively low value of the RF resistor, offers a good protection against issues resulting from use in a humid environment, due to the induced leakage and the bias condition change. However, it is recommended to take this point into account if the  $\mu C$  is used in tough humidity conditions.
2. For  $C_{L1}$  and  $C_{L2}$  it is recommended to use high-quality ceramic capacitors in the 5-pF to 25-pF range (typ.) designed for high-frequency applications and selected to match the requirements of the crystal or resonator.  $C_{L1}$  and  $C_{L2}$  are usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of  $C_{L1}$  and  $C_{L2}$ . PCB and MCU pin capacitance must be included when sizing  $C_{L1}$  and  $C_{L2}$  (10 pF can be used as a rough estimate of the combined pin and board capacitance).



# I/O PORT PIN CHARACTERISTICS (Cont'd)

## 12.8.2 Output Driving Current

Subject to general operating conditions 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 8 pins are sunk at same time (see Figure 81)	$I_{IO}=+5mA$		1.2	V
		$I_{IO}=+2mA$		0.5	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see Figure 82 and Figure 84)	$I_{IO}=+20mA, T_A \leq 85^\circ C$ $T_A \geq 85^\circ C$		1.3 1.5	
		$I_{IO}=+8mA$		0.6	
$V_{OH}^{2)}$	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see Figure 83 and Figure 86)	$I_{IO}=-5mA, T_A \leq 85^\circ C$ $T_A \geq 85^\circ C$	$V_{DD}-1.4$ $V_{DD}-1.6$		
		$I_{IO}=-2mA$	$V_{DD}-0.7$		

Figure 81. Typical  $V_{OL}$  at  $V_{DD}=5V$  (standard)

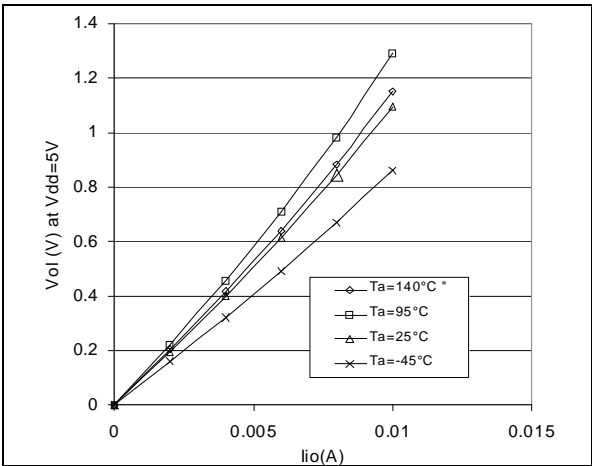


Figure 83. Typical  $V_{OH}$  at  $V_{DD}=5V$

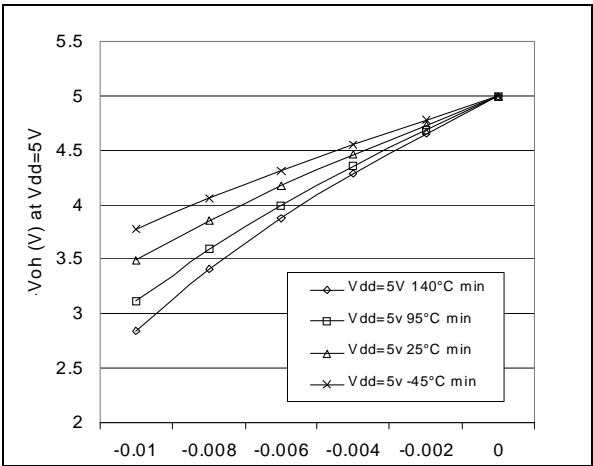
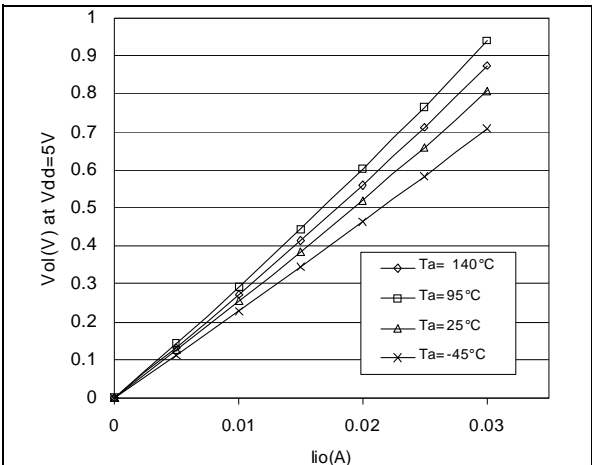


Figure 82. Typical  $V_{OL}$  at  $V_{DD}=5V$  (high-sink)



### Notes:

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

## 10-BIT ADC CHARACTERISTICS (Cont'd)

## 12.12.3 ADC Accuracy

Conditions:  $V_{DD}=5V$  <sup>1)</sup>

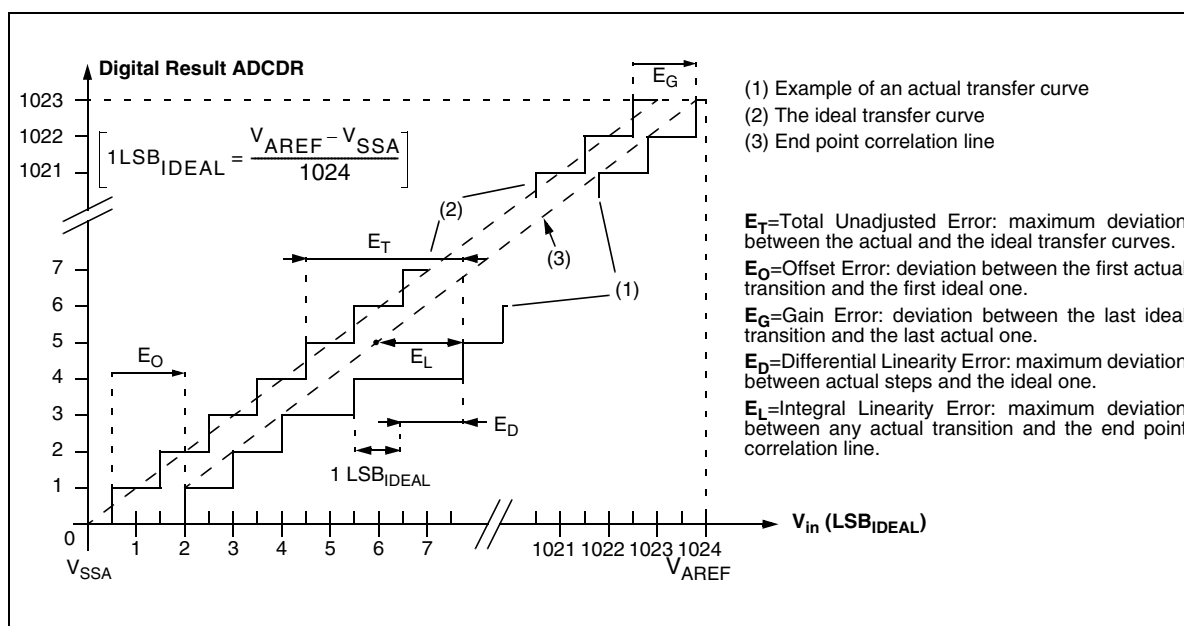
Symbol	Parameter	Conditions	Typ	Max <sup>2)</sup>	Unit
$ E_T $	Total unadjusted error <sup>1)</sup>		3	4	LSB
$ E_O $	Offset error <sup>1)</sup>		2	3	
$ E_G $	Gain Error <sup>1)</sup>		0.5	3	
$ E_D $	Differential linearity error <sup>1)</sup>	CPU in run mode @ $f_{ADC}$ 2 MHz.	1	2	
$ E_L $	Integral linearity error <sup>1)</sup>	CPU in run mode @ $f_{ADC}$ 2 MHz.	1	2	

**Notes:**

1. ADC Accuracy vs. Negative Injection Current: Injecting negative current may reduce the accuracy of the conversion being performed on another analog input.

Any positive injection current within the limits specified for  $I_{INJ(PIN)}$  and  $\Sigma I_{INJ(PIN)}$  in Section 12.8 does not affect the ADC accuracy.

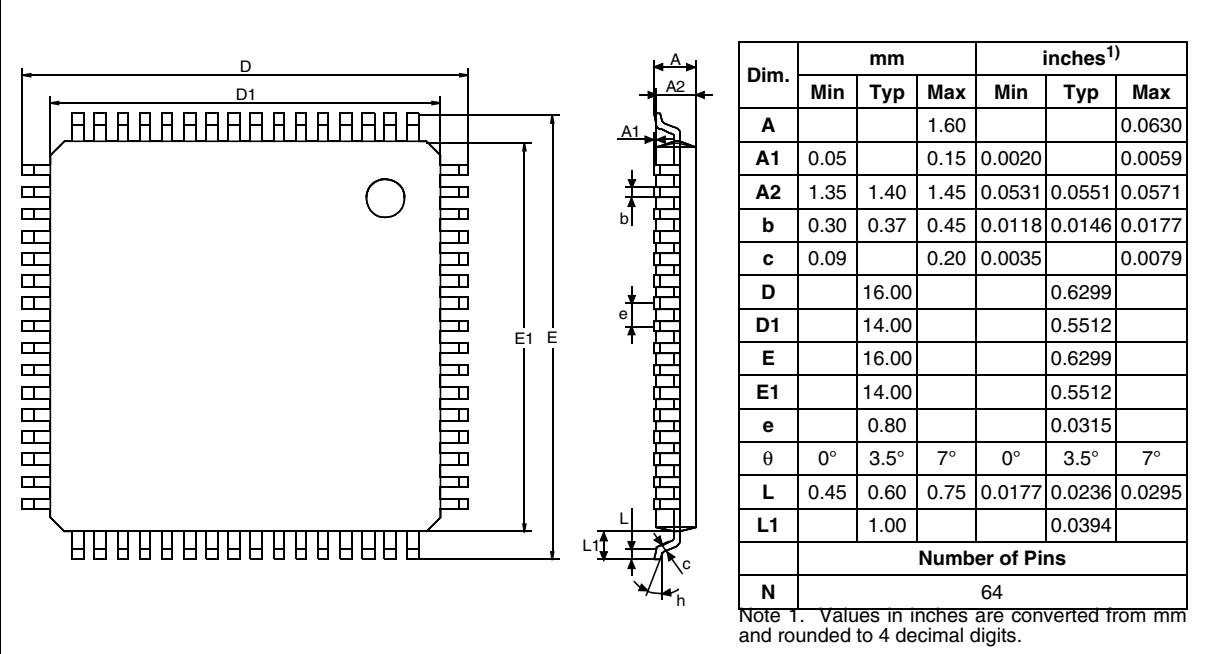
2. Data based on characterization results, monitored in production to guarantee 99.73% within  $\pm$  max value from -40°C to 125°C ( $\pm 3\sigma$  distribution limits).

**Figure 98. ADC Accuracy Characteristics**

13 PACKAGE CHARACTERISTICS

13.1 PACKAGE MECHANICAL DATA

Figure 99. 64-Pin Low Profile Quad Flat Package (14x14)



## 13.2 THERMAL CHARACTERISTICS

Symbol	Ratings	Value	Unit
$R_{thJA}$	Package thermal resistance (junction to ambient)		
	LQFP64 10x10	50	°C/W
	LQFP48 7x7	80	
	LQFP44 10x10	52	
	SDIP42	55	
	LQFP32 7x7	70	
	SDIP32	50	
$P_D$	Power dissipation <sup>1)</sup>	500	mW
$T_{Jmax}$	Maximum junction temperature <sup>2)</sup>	150	°C

**Notes:**

1. The maximum chip-junction temperature is based on technology characteristics.

2. The maximum power dissipation is obtained from the formula  $P_D = (T_J - T_A) / R_{thJA}$ .

The power dissipation of an application can be defined by the user with the formula:  $P_D = P_{INT} + P_{PORT}$  where  $P_{INT}$  is the chip internal power ( $I_{DD} \times V_{DD}$ ) and  $P_{PORT}$  is the port power dissipation depending on the ports used in the application.

**KNOWN LIMITATIONS (Cont'd)****15.1.4 SCI Wrong Break duration****Description**

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

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

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

**Occurrence**

The occurrence of the problem is random and proportional to the baudrate. With a transmit frequency of 19200 baud ( $f_{CPU}=8\text{MHz}$  and  $\text{SCI-BRR}=0\text{x}C9$ ), 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

**15.1.5 16-bit Timer PWM Mode**

In PWM mode, the first PWM pulse is missed after writing the value  $\text{FFFCh}$  in the OC1R register (OC1HR, OC1LR). It leads to either full or no PWM

during a period, depending on the OLVL1 and OLVL2 settings.

**15.1.6 TIMD set simultaneously with OC interrupt**

If the 16-bit timer is disabled at the same time the output compare event occurs then output compare flag gets locked and cannot be cleared before the timer is enabled again.

Impact on the application

If output compare interrupt is enabled, then the output compare flag cannot be cleared in the timer interrupt routine. Consequently the interrupt service routine is called repeatedly.

**Workaround**

Disable the timer interrupt before disabling the timer. Again while enabling, first enable the timer then the timer interrupts.

Perform the following to disable the timer:

TACR1 or TBCR1 =  $0\text{x}00\text{h}$ ; // Disable the compare interrupt

TACSR1 or TBCSR1 =  $0\text{x}40$ ; // Disable the timer

Perform the following to enable the timer again:

TACSR & or TBCSR &=  $\sim 0\text{x}40$ ; // Enable the timer

TACR1 or TBCR1 =  $0\text{x}40$ ; // Enable the compare interrupt

**15.1.7 I2C Multimaster**

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