



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	Not For New Designs
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/st72f325k4t6tr">https://www.e-xfl.com/product-detail/stmicroelectronics/st72f325k4t6tr</a>

SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.3 Clock Security System (CSS)

The Clock Security System (CSS) protects the ST7 against breakdowns, spikes and overfrequencies occurring on the main clock source ( $f_{OSC}$ ). It is based on a clock filter and a clock detection control with an internal safe oscillator ( $f_{SFOSC}$ ).

6.4.3.1 Clock Filter Control

The PLL has an integrated glitch filtering capability making it possible to protect the internal clock from overfrequencies created by individual spikes. This feature is available only when the PLL is enabled. If glitches occur on  $f_{OSC}$  (for example, due to loose connection or noise), the CSS filters these automatically, so the internal CPU frequency ( $f_{CPU}$ ) continues deliver a glitch-free signal (see Figure 20).

6.4.3.2 Clock detection Control

If the clock signal disappears (due to a broken or disconnected resonator...), the safe oscillator delivers a low frequency clock signal ( $f_{SFOSC}$ ) which allows the ST7 to perform some rescue operations.

Automatically, the ST7 clock source switches back from the safe oscillator ( $f_{SFOSC}$ ) if the main clock source ( $f_{OSC}$ ) recovers.

When the internal clock ( $f_{CPU}$ ) is driven by the safe oscillator ( $f_{SFOSC}$ ), the application software is notified by hardware setting the CSSD bit in the SICSR register. An interrupt can be generated if the

CSSIE bit has been previously set. These two bits are described in the SICSR register description.

6.4.4 Low Power Modes

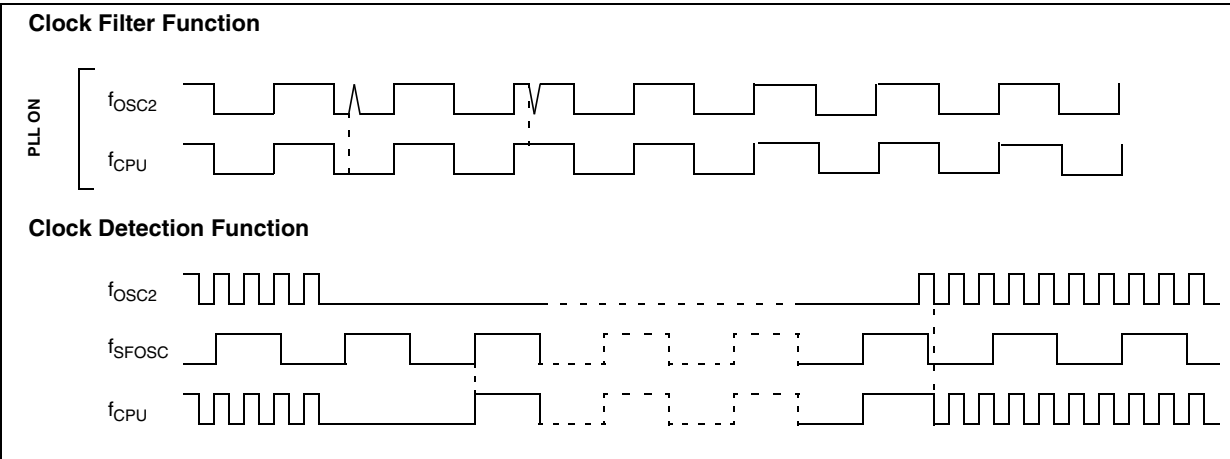
Mode	Description
WAIT	No effect on SI. CSS and AVD interrupts cause the device to exit from Wait mode.
HALT	The SICSR register is frozen. The CSS (including the safe oscillator) is disabled until HALT mode is exited. The previous CSS configuration resumes when the MCU is woken up by an interrupt with "exit from HALT mode" capability or from the counter reset value when the MCU is woken up by a RESET.

6.4.4.1 Interrupts

The CSS orAVD interrupt events generate an interrupt if the corresponding Enable Control Bit (CSSIE or AVDIE) 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
CSS event detection (safe oscillator activated as main clock)	CSSD	CSSIE	Yes	No
AVD event	AVDF	AVDIE	Yes	No

Figure 20. Clock Filter Function



## 10.2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC)

The Main Clock Controller consists of three different functions:

- a programmable CPU clock prescaler
- a clock-out signal to supply external devices
- a real time clock timer with interrupt capability

Each function can be used independently and simultaneously.

### 10.2.1 Programmable CPU Clock Prescaler

The programmable CPU clock prescaler supplies the clock for the ST7 CPU and its internal peripherals. It manages SLOW power saving mode (See [Section 8.2 SLOW MODE](#) for more details).

The prescaler selects the  $f_{CPU}$  main clock frequency and is controlled by three bits in the MCCR register: CP[1:0] and SMS.

### 10.2.2 Clock-out Capability

The clock-out capability is an alternate function of an I/O port pin that outputs a  $f_{CPU}$  clock to drive

external devices. It is controlled by the MCO bit in the MCCR register.

**CAUTION:** When selected, the clock out pin suspends the clock during ACTIVE-HALT mode.

### 10.2.3 Real Time Clock Timer (RTC)

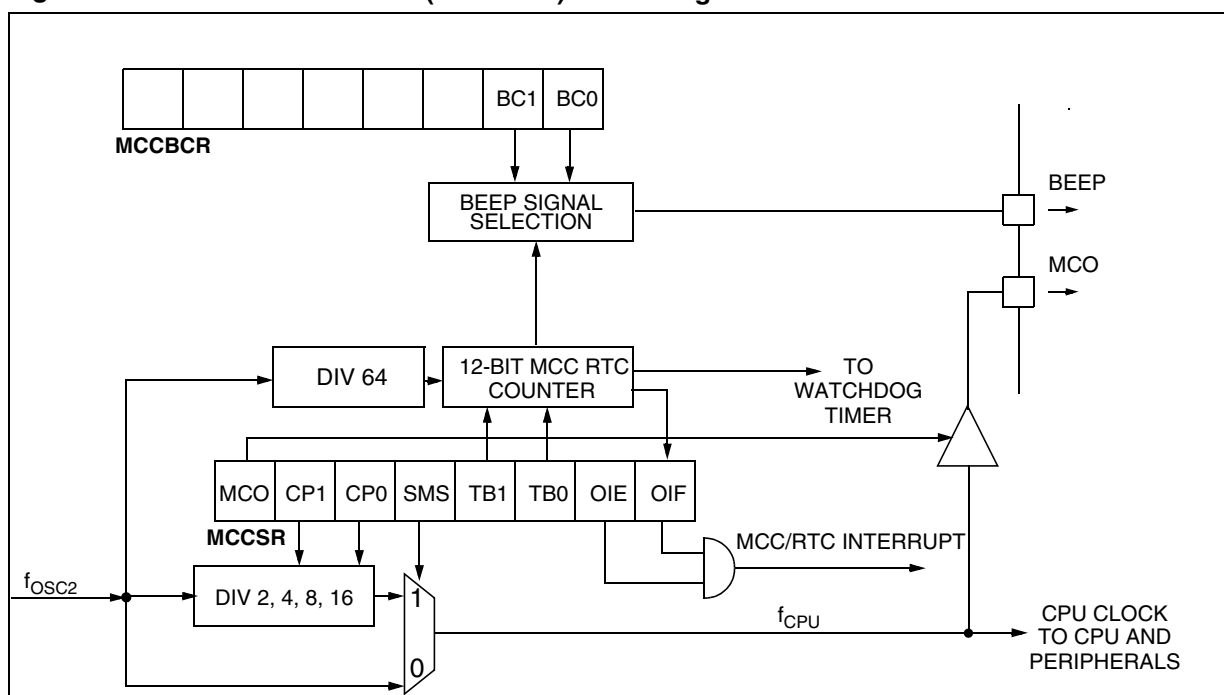
The counter of the real time clock timer allows an interrupt to be generated based on an accurate real time clock. Four different time bases depending directly on  $f_{OSC2}$  are available. The whole functionality is controlled by four bits of the MCCR register: TB[1:0], OIE and OIF.

When the RTC interrupt is enabled (OIE bit set), the ST7 enters ACTIVE-HALT mode when the HALT instruction is executed. See [Section 8.4 ACTIVE-HALT AND HALT MODES](#) for more details.

### 10.2.4 Beeper

The beep function is controlled by the MCCBCR register. It can output three selectable frequencies on the BEEP pin (I/O port alternate function).

Figure 38. Main Clock Controller (MCC/RTC) Block Diagram



16-BIT TIMER (Cont'd)

Figure 46. Counter Timing Diagram, Internal Clock Divided by 2

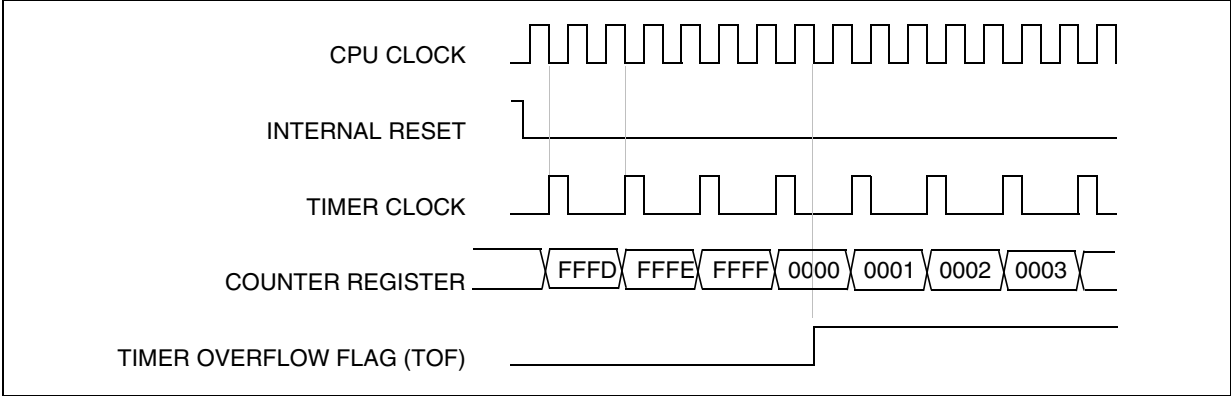


Figure 47. Counter Timing Diagram, Internal Clock Divided by 4

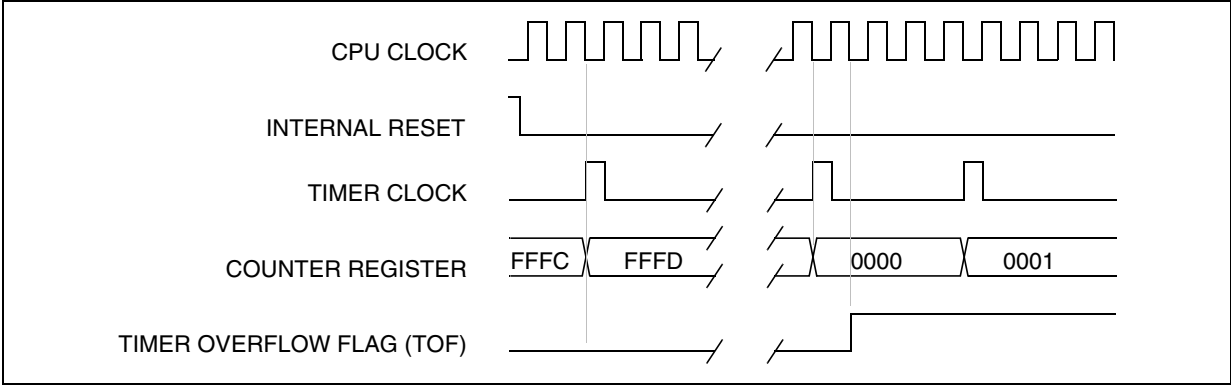
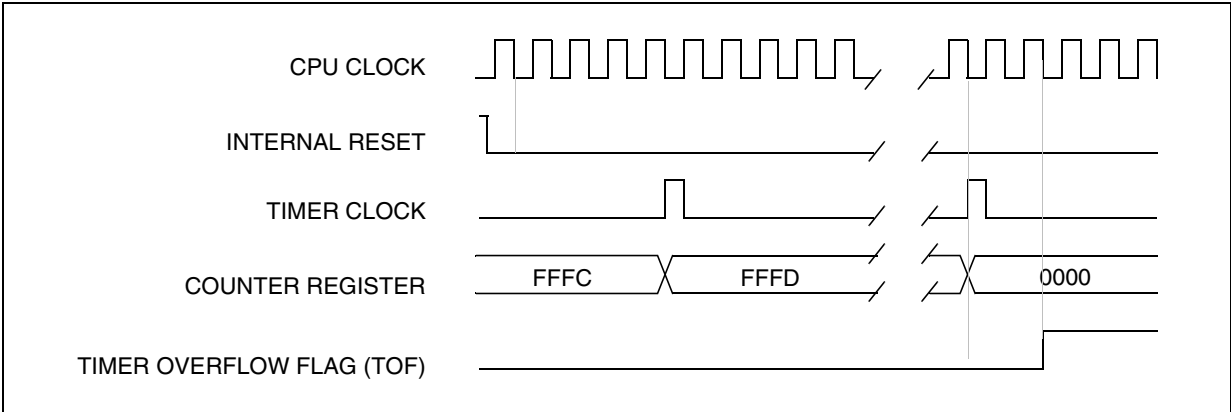


Figure 48. Counter Timing Diagram, Internal Clock Divided By 8



**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)****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.

## SERIAL PERIPHERAL INTERFACE (Cont'd)

## 10.5.4 Clock Phase and Clock Polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (See Figure 60).

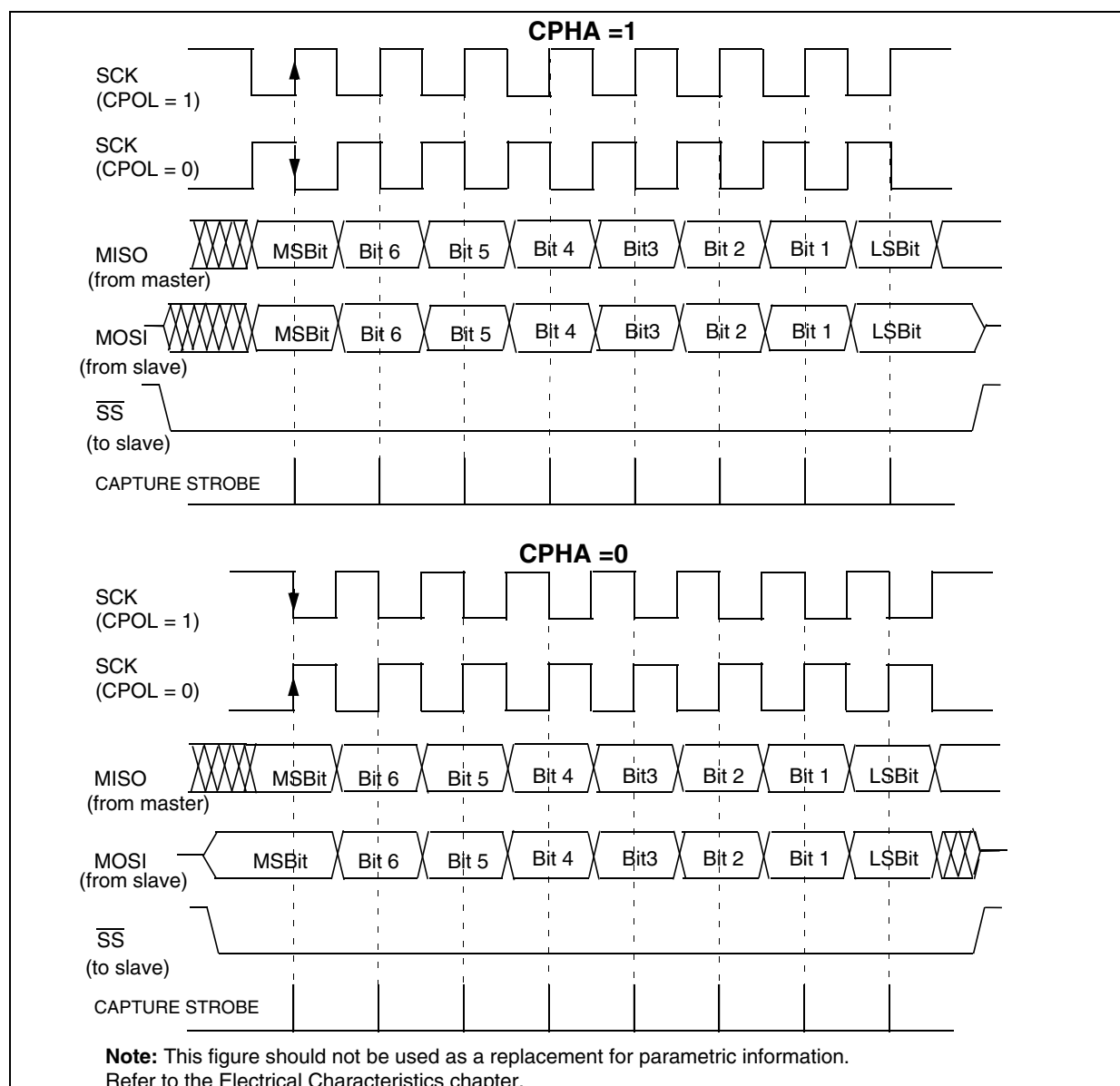
**Note:** The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL=1 or pulling down SCK if CPOL=0).

The combination of the CPOL clock polarity and CPHA (clock phase) bits selects the data capture clock edge

Figure 60, shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK pin, the MISO pin, the MOSI pin are directly connected between the master and the slave device.

**Note:** If CPOL is changed at the communication byte boundaries, the SPI must be disabled by re-setting the SPE bit.

Figure 60. Data Clock Timing Diagram





**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.



**SERIAL COMMUNICATIONS 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 1.](#)).

The RDR register provides the parallel interface between the input shift register and the internal bus (see [Figure 1.](#)).

**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 in conventional Baud Rate Generator mode.

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 in conventional Baud Rate Generator mode.

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

## SERIAL COMMUNICATION INTERFACE (Cont'd)

Table 24. SCI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0050h	<b>SCISR</b> Reset Value	TDRE 1	TC 1	RDRF 0	IDLE 0	OVR 0	NF 0	FE 0	PE 0
0051h	<b>SCIDR</b> Reset Value	MSB x	x	x	x	x	x	x	LSB x
0052h	<b>SCIBRR</b> Reset Value	SCP1 0	SCP0 0	SCT2 0	SCT1 0	SCT0 0	SCR2 0	SCR1 0	SCR0 0
0053h	<b>SCICR1</b> Reset Value	R8 x	T8 0	SCID 0	M 0	WAKE 0	PCE 0	PS 0	PIE 0
0054h	<b>SCICR2</b> Reset Value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0
0055h	<b>SCIERPR</b> Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	<b>SCIPETPR</b> Reset Value	MSB 0	0	0	0	0	0	0	LSB 0

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

### 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.

### SMBus Compatibility

ST7 I<sup>2</sup>C is compatible with SMBus V1.1 protocol. It supports all SMBus addressing modes, SMBus bus protocols and CRC-8 packet error checking. Refer to AN1713: SMBus Slave Driver For ST7 I<sup>2</sup>C Peripheral.

### 10.7.4.2 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, **holding the SCL line low** (see [Figure 69](#) Transfer sequencing EV5).

#### Slave address transmission

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

In 7-bit addressing mode, one address byte is sent.

In 10-bit addressing mode, sending the first byte including the header sequence causes the following event:

- 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 DR register, **holding the SCL line low** (see [Figure 69](#) Transfer sequencing EV9).

Then the second address byte is sent by the interface.

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 69](#) Transfer sequencing EV6).

Next the master must enter Receiver or Transmitter mode.

**Note:** In 10-bit addressing mode, to switch the master to Receiver mode, software must generate a repeated Start condition and resend the header sequence with the least significant bit set (11110xx1).

#### Master Receiver

Following the address transmission and after 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 69](#) 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.

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

### 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 69 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<sup>2</sup>C 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<sup>2</sup>C master mode transmission. 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 due to possible «0» bits transmitted last. It is then necessary to release both lines by software.

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

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

**I<sup>2</sup>C STATUS REGISTER 2 (SR2)**

Read Only

Reset Value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	ARLO	BERR	GCAL

Bit 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).

The SCL line is not held low while AF=1 but by other flags (SB or BTF) that are set at the same time.

0: No acknowledge failure

1: Acknowledge failure

Note:

- When an AF event occurs, the SCL line is not held low; however, the SDA line can remain low if the last bits transmitted are all 0. It is then necessary to release both lines by software.

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<sup>2</sup>C 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

**10-BIT A/D CONVERTER** (Cont'd)**Table 26. ADC Register Map and Reset Values**

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

## INSTRUCTION SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	I1	H	I0	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	$A = A \cdot M$	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	$A = \text{FFH} - A$	reg, M					N	Z	1
DEC	Decrement	dec Y	reg, M					N	Z	
HALT	Halt				1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC			I1	H	I0	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. INT pin = 1	(ext. INT pin high)								
JRIL	Jump if ext. INT pin = 0	(ext. INT pin low)								
JRH	Jump if H = 1	H = 1 ?								
JRNH	Jump if H = 0	H = 0 ?								
JRM	Jump if I1:0 = 11	I1:0 = 11 ?								
JRNM	Jump if I1:0 <> 11	I1:0 <> 11 ?								
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 >								

**CLOCK AND TIMING CHARACTERISTICS** (Cont'd)

Supplier	f <sub>osc</sub> (MHz)	Typical Ceramic Resonators <sup>2)</sup>
Murata	2	CSTCC2M00G56Z-R0
	4	SMD CSTCR4M00G53Z-R0 Lead CSTLS4M00G53Z-R0
	8	SMD CSTCE8M00G52Z-R0 Lead CSTLS4M0052Z-R0
	16	SMD CSTCE16M0V51Z-R0 Lead CSTLS16M0X51Z-R0

**Notes:**

1. Resonator characteristics given by the ceramic resonator manufacturer.

2. SMD = [-R0: Plastic tape package (Ø = 180mm), -B0: Bulk]

LEAD = [-A0: Flat pack package (Radial taping Ho = 18mm), -B0: Bulk]

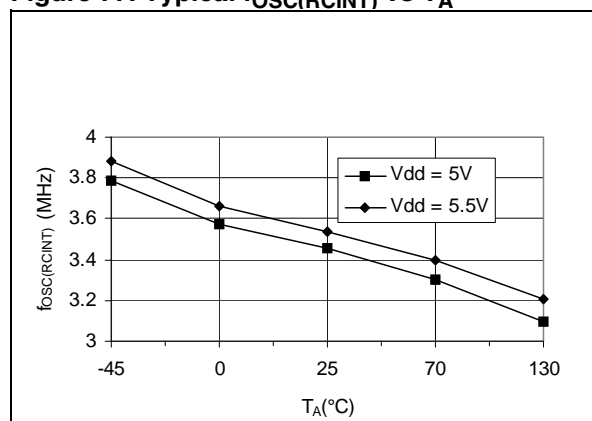
For more information on these resonators, please consult [www.murata.com](http://www.murata.com)



## CLOCK CHARACTERISTICS (Cont'd)

## 12.5.4 RC Oscillators

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{\text{OSC (RCINT)}}$	Internal RC oscillator frequency See <a href="#">Figure 77</a>	$T_A=25^\circ\text{C}$ , $V_{\text{DD}}=5\text{V}$	2	3.5	5.6	MHz

Figure 77. Typical  $f_{\text{OSC(RCINT)}}$  vs  $T_A$ 

**Note:** To reduce disturbance to the RC oscillator, it is recommended to place decoupling capacitors between  $V_{\text{DD}}$  and  $V_{\text{SS}}$  as shown in [Figure 97](#)

## 12.10 TIMER PERIPHERAL CHARACTERISTICS

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (output compare, input capture, external clock, PWM output...).

### 12.10.1 8-Bit PWM-ART Auto-Reload Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{res(PWM)}$	PWM resolution time		1			$t_{CPU}$
		$f_{CPU}=8MHz$	125			ns
$f_{EXT}$	ART external clock frequency		0		$f_{CPU}/2$	MHz
$f_{PWM}$	PWM repetition rate		0		$f_{CPU}/2$	
$Res_{PWM}$	PWM resolution				8	bit
$V_{OS}$	PWM/DAC output step voltage	$V_{DD}=5V$ , Res=8-bits		20		mV

### 12.10.2 16-Bit Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{w(ICAP)in}$	Input capture pulse time		1			$t_{CPU}$
$t_{res(PWM)}$	PWM resolution time		2			$t_{CPU}$
		$f_{CPU}=8MHz$	250			ns
$f_{EXT}$	Timer external clock frequency		0		$f_{CPU}/4$	MHz
$f_{PWM}$	PWM repetition rate		0		$f_{CPU}/4$	MHz
$Res_{PWM}$	PWM resolution				16	bit

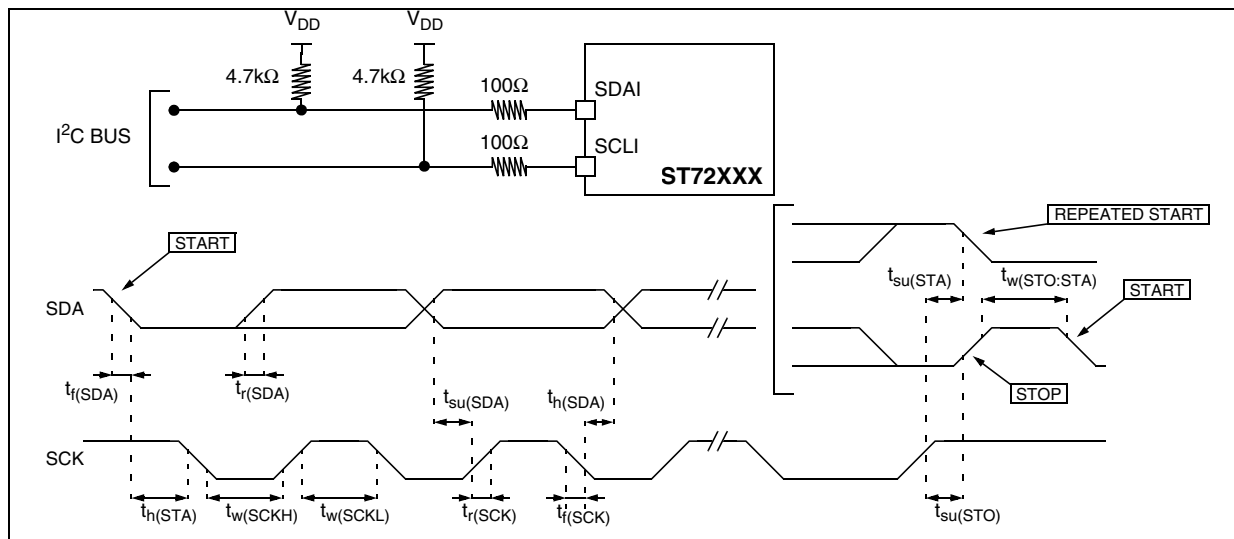
## COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

12.11.2 I<sup>2</sup>C - Inter IC Control 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 (SDAI and SCLI). The ST7 I<sup>2</sup>C interface meets the requirements of the Standard I<sup>2</sup>C communication protocol described in the following table.

Symbol	Parameter	Standard mode I <sup>2</sup> C		Fast mode I <sup>2</sup> C <sup>5)</sup>		Unit
		Min <sup>1)</sup>	Max <sup>1)</sup>	Min <sup>1)</sup>	Max <sup>1)</sup>	
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		$\mu s$
$t_{w(SCLH)}$	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 <sup>3)</sup>		0 <sup>2)</sup>	900 <sup>3)</sup>	
$t_r(SDA)$ $t_r(SCL)$	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		$\mu s$
$t_{su(STA)}$	Repeated START condition setup time	4.7		0.6		
$t_{su(STO)}$	STOP condition setup time	4.0		0.6		$\mu s$
$t_{w(STO:STA)}$	STOP to START condition time (bus free)	4.7		1.3		$\mu s$
$C_b$	Capacitive load for each bus line		400		400	pF

Figure 93. Typical Application with I<sup>2</sup>C Bus and Timing Diagram <sup>4)</sup>

## Notes:

1. Data based on standard I<sup>2</sup>C 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. Measurement points are done at CMOS levels:  $0.3 \times V_{DD}$  and  $0.7 \times V_{DD}$ .
5. At 4MHz  $f_{CPU}$ , max. I<sup>2</sup>C speed (400kHz) is not achievable. In this case, max. I<sup>2</sup>C speed will be approximately 260kHz.

Table 32. 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
<b>SYSTEM OPTIMIZATION</b>	
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

**KNOWN LIMITATIONS (Cont'd)****15.1.8 Pull-up always active on PE2**

The I/O port internal pull-up is always active on I/O port E2. As a result, if PE2 is in output mode low level, current consumption in Halt/Active Halt mode is increased.

**15.1.9 ADC accuracy 16/32K Flash devices**

The ADC accuracy in 16/32K Flash Devices deviates from table in [section 12.12.3 on page 173](#) as follows:

Symbol	Max	Unit
$ E_T $	6	LSB
$ E_O $	5	
$ E_G $	4.5	
$ E_D $	2	
$ E_L $	3	