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	CANbus, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f521r6tc

Table 3. Device pin description (continued)

Pin No. LQFP80	Pin No. LQFP64	Name	Type	Level: Input	Level: Output	Port: Input: float	Port: Input: wpu	Port: Input: int	Port: Input: ana	Port: Output: OD	Port: Output: PP	Main function (after reset)	Alternate function
65	53	V_{PP}/ ICCSEL	I									Must be tied low. In Flash programming mode, this pin acts as the programming voltage input V_{PP}. See *Section 20.9.2 on page 242* for more details. High voltage must not be applied to ROM devices.	
66	54	$\overline{RESET}$	I/O	C_T								Top priority non-maskable interrupt	
67	55	EVD	I	A								External voltage detector	
68	56	TLI	I	C_T		**X**		X				Top level interrupt input pin	
69	- (1)	PH4	I/O	T_T		**X**	X			X	X	Port H4	
70	- (1)	PH5	I/O	T_T		**X**	X			X	X	Port H5	
71	- (1)	PH6	I/O	T_T		**X**	X			X	X	Port H6	
72	- (1)	PH7	I/O	T_T		**X**	X			X	X	Port H7	
73	57	V_{SS_2} (2)	S									Digital Ground Voltage	
74	58	OSC2 (3)	I/O									Resonator oscillator inverter output	
75	59	OSC1 (3)	I									External clock input or Resonator oscillator inverter input	
76	60	V_{DD_2} (2)	S									Digital Main Supply Voltage	
77	61	PE0/TDO	I/O	C_T		**X**	X			X	X	Port E0	SCI Transmit Data Out
78	62	PE1/RDI	I/O	C_T		**X**	X			X	X	Port E1	SCI Receive Data In
79	63	PE2/CANTX	I/O	C_T			**X**					Port E2	CAN Transmit Data Output
80	64	PE3/CANRX	I/O	C_T		**X**	X			X	X	Port E3	CAN Receive Data Input

1. On the chip, each I/O port may have up to 8 pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption
2. It is mandatory to connect all available V_{DD} and V_{AREF} pins to the supply voltage and all V_{SS} and V_{SSA} pins to ground.
3. OSC1 and OSC2 pins connect a crystal/ceramic resonator or an external source to the on-chip oscillator; see *Section 6.4: Multi-oscillator (MO) on page 41* and *Section 20.5: Clock and timing characteristics on page 228* for more details.

Legend / Abbreviations for *Table 3*:

Type: I = input
O = output
S = supply

Input level: A = dedicated analog input

Table 4. Hardware register map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0070h	ADC	ADCCSR	Control/Status Register	00h	R/W
0071h		ADCDRH	Data High Register	00h	Read Only
0072h		ADCDRL	Data Low Register	00h	Read Only
0073h	PWM ART	PWMDCR3	PWM AR Timer Duty Cycle Register 3	00h	R/W
0074h		PWMDCR2	PWM AR Timer Duty Cycle Register 2	00h	R/W
0075h		PWMDCR1	PWM AR Timer Duty Cycle Register 1	00h	R/W
0076h		PWMDCR0	PWM AR Timer Duty Cycle Register 0	00h	R/W
0077h		PWMCR	PWM AR Timer Control Register	00h	R/W
0078h		ARTCSR	Auto-Reload Timer Control/Status Register	00h	R/W
0079h		ARTCAR	Auto-Reload Timer Counter Access Register	00h	R/W
007Ah		ARTARR	Auto-Reload Timer Auto-Reload Register	00h	R/W
007Bh		ARTICCSR	AR Timer Input Capture Control/Status Reg.	00h	R/W
007Ch		ARTICR1	AR Timer Input Capture Register 1	00h	Read Only
007Dh		ARTICR2	AR Timer Input Capture Register 1	00h	Read Only
007Eh 007Fh	Reserved Area (2 bytes)				

1. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.
2. The bits associated with unavailable pins must always keep their reset value.

Note: Legend: x = undefined, R/W = read/write

5 Central processing unit (CPU)

5.1 Introduction

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

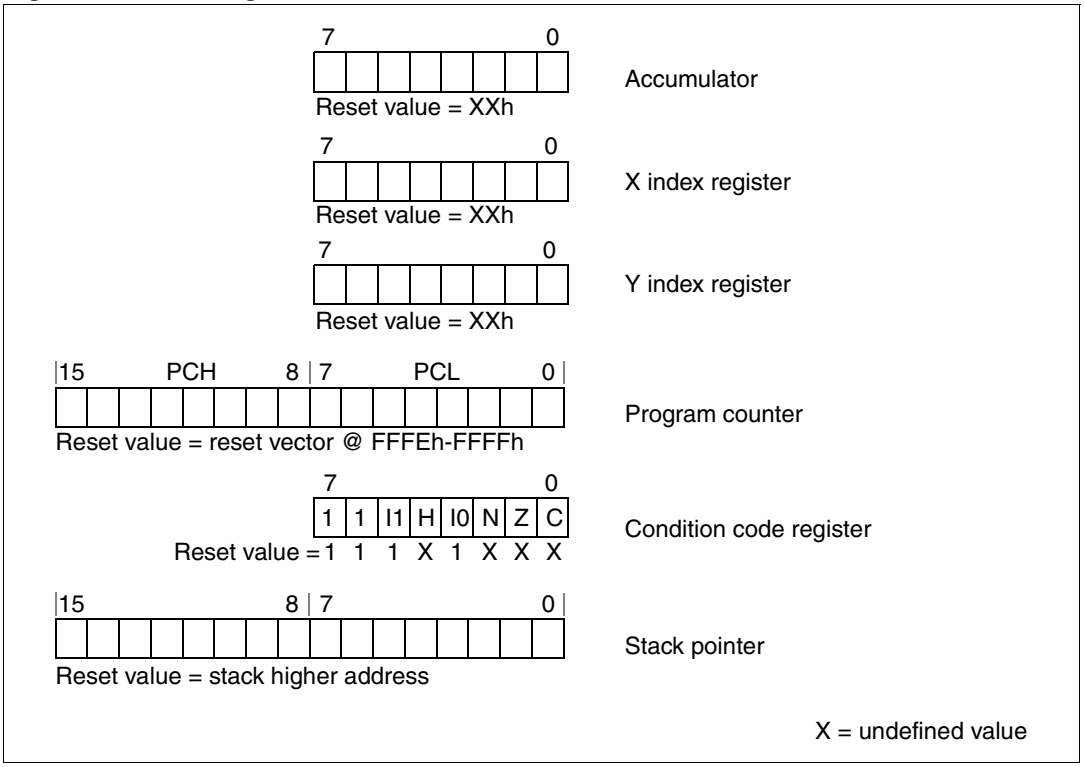
5.2 Main features

- Enable executing 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- Low power Halt and Wait modes
- Priority maskable hardware interrupts
- Non-maskable software/hardware interrupts

5.3 CPU registers

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

Figure 7. CPU registers



10.9 Register description

10.9.1 Control register (WDGCR)

WDGCR Reset value: 0111 1111 (7Fh)

7	6	5	4	3	2	1	0
WDGA	T[6:0]						
RW	RW						

Table 36. WDGCR register description

Bit	Name	Function
7	WDGA	*Activation bit* This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset. 0: Watchdog disabled 1: Watchdog enabled *Note: This bit is not used if the hardware watchdog option is enabled by option byte.*
6:0	T[6:0]	*7-bit counter (MSB to LSB)* These bits contain the value of the watchdog counter. It is decremented every 16384 f_{OSC2} cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

Table 37. Watchdog timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
002Ah	WDGCR Reset Value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

5 *The alternate inputs (ICAP1 and ICAP2) are always directly connected to the timer. So any transitions on these pins activates the input capture function.*

6 *Moreover if one of the ICAPi pins is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.*

7 *This can be avoided if the input capture function i is disabled by reading the ICiHR (see note 1).*

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

Figure 46. Input capture block diagram

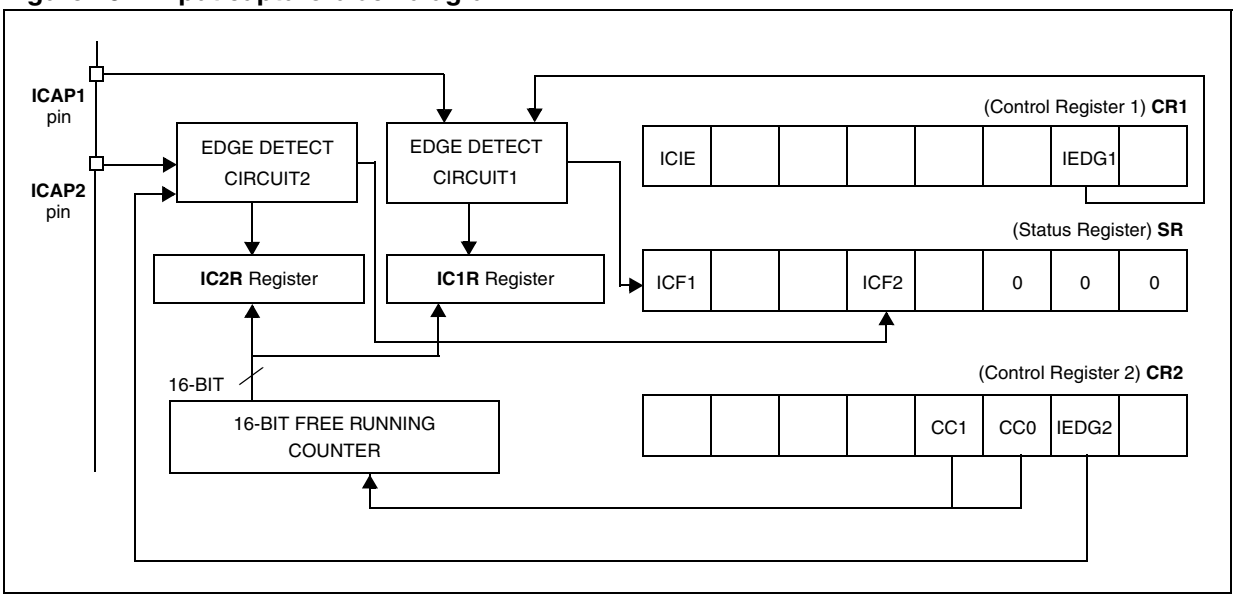
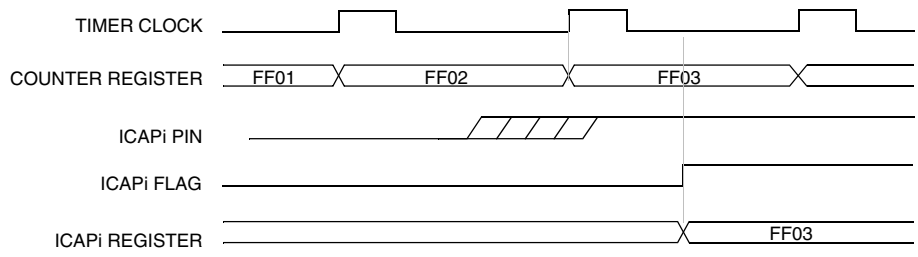


Figure 47. Input capture timing diagram



Note: The rising edge is the active edge.

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

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

Where:

t = Pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits; see *Table 61: Timer clock selection*)

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

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

Where:

t = Pulse period (in seconds)

f_{EXT} = External clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see *Figure 52*).

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

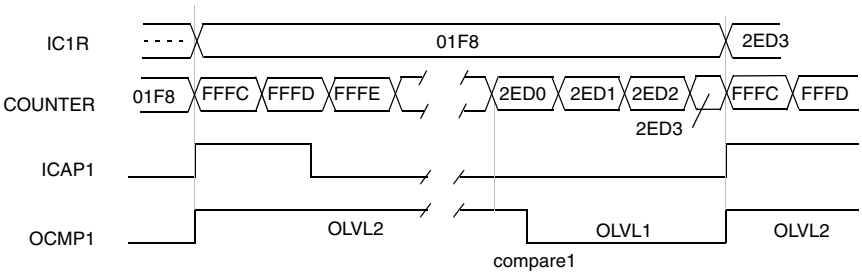
2 When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

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

4 The ICAP1 pin cannot be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.

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

Figure 52. One pulse mode timing example



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

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 *Overrun condition (OVR) on page 131*).

14.4 Clock phase and clock polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (see *Figure 59*).

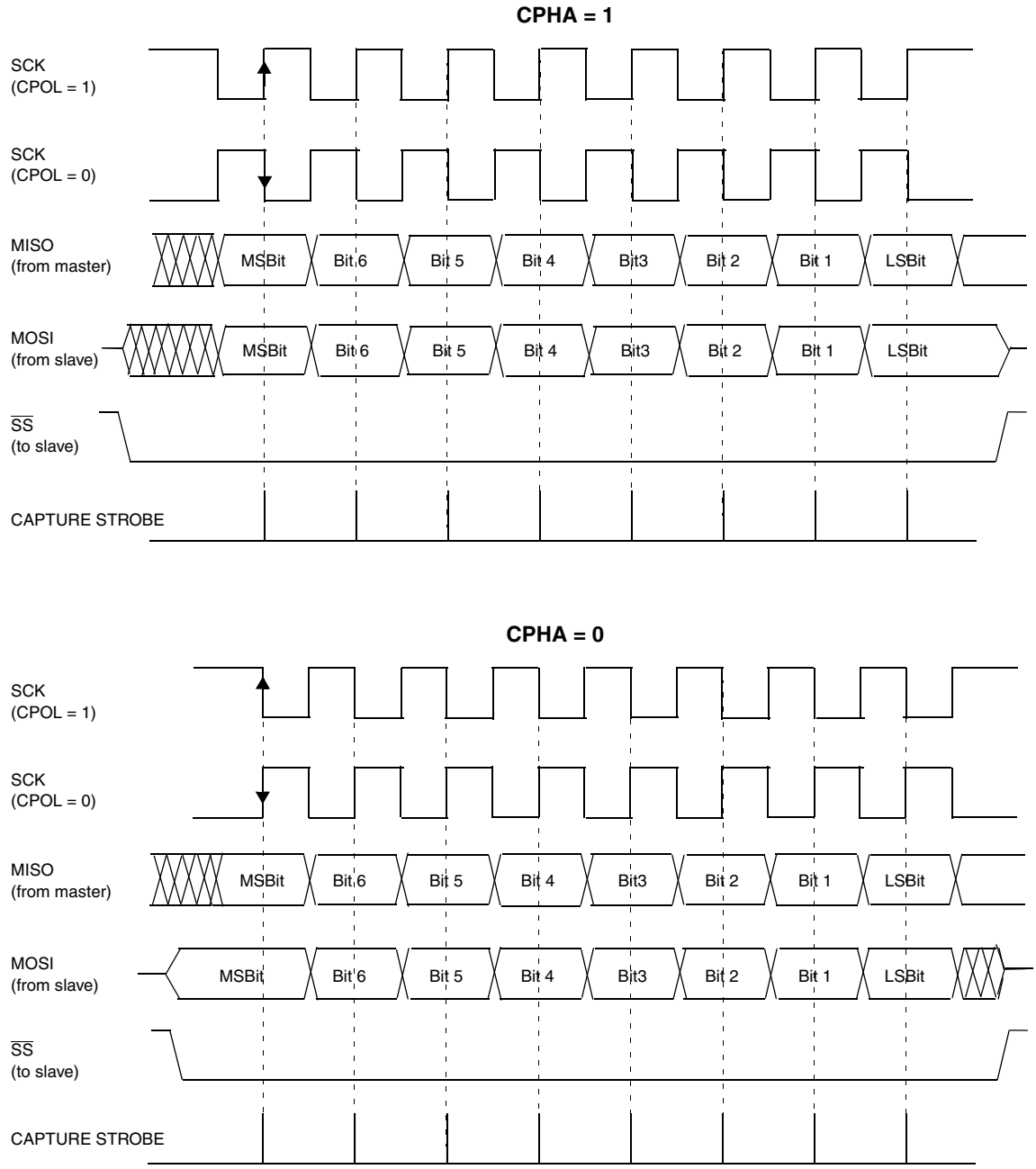
Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

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

Figure 59 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 resetting the SPE bit.

Figure 59. Data clock timing diagram



Note: This figure should not be used as a replacement for parametric information.
Refer to Section 20: Electrical characteristics.

14.5 Error flags

14.5.1 Master mode fault (MODF)

Master mode fault occurs when the master device has its $\overline{SS}$ pin pulled low.

When a Master mode fault occurs:

- The MODF bit is set and an SPI interrupt request is generated if the SPIE bit is set.
- The SPE bit is reset. This blocks all output from the device and disables the SPI peripheral.
- The MSTR bit is reset, thus forcing the device into slave mode.

Clearing the MODF bit is done through a software sequence:

1. A read access to the SPICSR register while the MODF bit is set.
2. A write to the SPICR register.

Note: To avoid any conflicts in an application with multiple slaves, the $\overline{SS}$ pin must be pulled high during the MODF bit clearing sequence. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

Hardware does not allow the user to set the SPE and MSTR bits while the MODF bit is set except in the MODF bit clearing sequence.

14.5.2 Overrun condition (OVR)

An overrun condition occurs, when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previously transmitted byte.

When an Overrun occurs:

- The OVR bit is set and an interrupt request is generated if the SPIE bit is set.

In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read to the SPIDR register returns this byte. All other bytes are lost.

The OVR bit is cleared by reading the SPICSR register.

14.5.3 Write collision error (WCOL)

A write collision occurs when the software tries to write to the SPIDR register while a data transfer is taking place with an external device. When this happens, the transfer continues uninterrupted; and the software write will be unsuccessful.

Write collisions can occur both in master and slave mode. See also *Slave select management on page 126*.

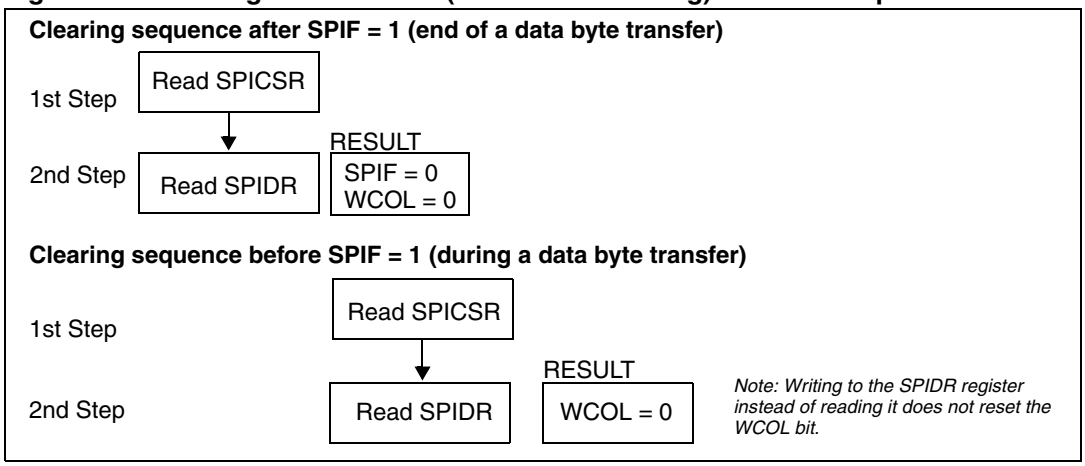
Note: A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation.

The WCOL bit in the SPICSR register is set if a write collision occurs.

No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see *Figure 60*).

Figure 60. Clearing the WCOL bit (Write Collision Flag) software sequence



14.5.4 Single master systems

A typical single master system may be configured, using an MCU as the master and four MCUs as slaves (see *Figure 61*).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

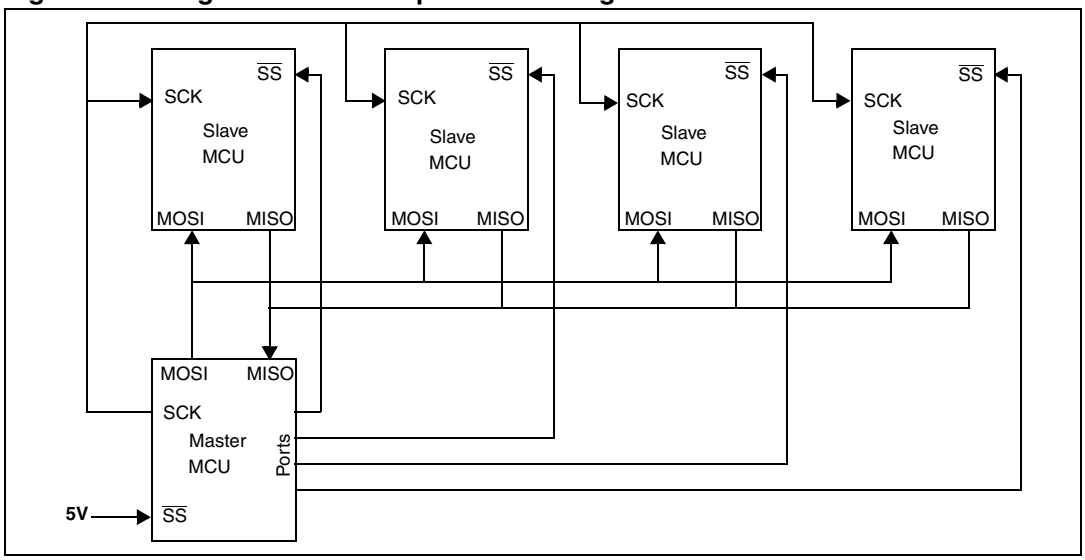
The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

Note: To prevent a bus conflict on the MISO line the master allows only one active slave device during a transmission.

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written to its SPIDR register.

Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Figure 61. Single master / multiple slave configuration



15.4.2 Transmitter

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

Character transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see *Figure 62*).

Procedure

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

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

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

The TDRE bit is set by hardware and it indicates:

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

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

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

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

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

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

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

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

Break characters

Setting the SBK bit loads the shift register with a break character. The break frame length depends on the M bit (see *Figure 63*).

As long as the SBK bit is set, the SCI send break frames to the TDO pin. After clearing this

Table 72. SCI interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Idle Line Detected	IDLE	ILIE	Yes	No
Parity Error	PE	PIE	Yes	No

15.7 SCI registers

15.7.1 Status register (SCISR)

SCISR Reset value: 1100 0000 (C0h)

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

Table 73. SCISR register description

Bit	Name	Function
7	TDRE	*Transmit data register empty* This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE bit = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Data is not transferred to the shift register 1: Data is transferred to the shift register *Note: Data is not transferred to the shift register unless the TDRE bit is cleared.*
6	TC	*Transmission complete* This bit is set by hardware when transmission of a frame containing Data is complete. An interrupt is generated if TCIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Transmission is not complete 1: Transmission is complete *Note: TC is not set after the transmission of a Preamble or a Break.*
5	RDRF	*Received data ready flag* This bit is set by hardware when the content of the RDR register has been transferred to the SCIDR register. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: Data is not received 1: Received data is ready to be read

16.3.3 SDA/SCL line control

Transmitter mode

The interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the data register.

Receiver mode

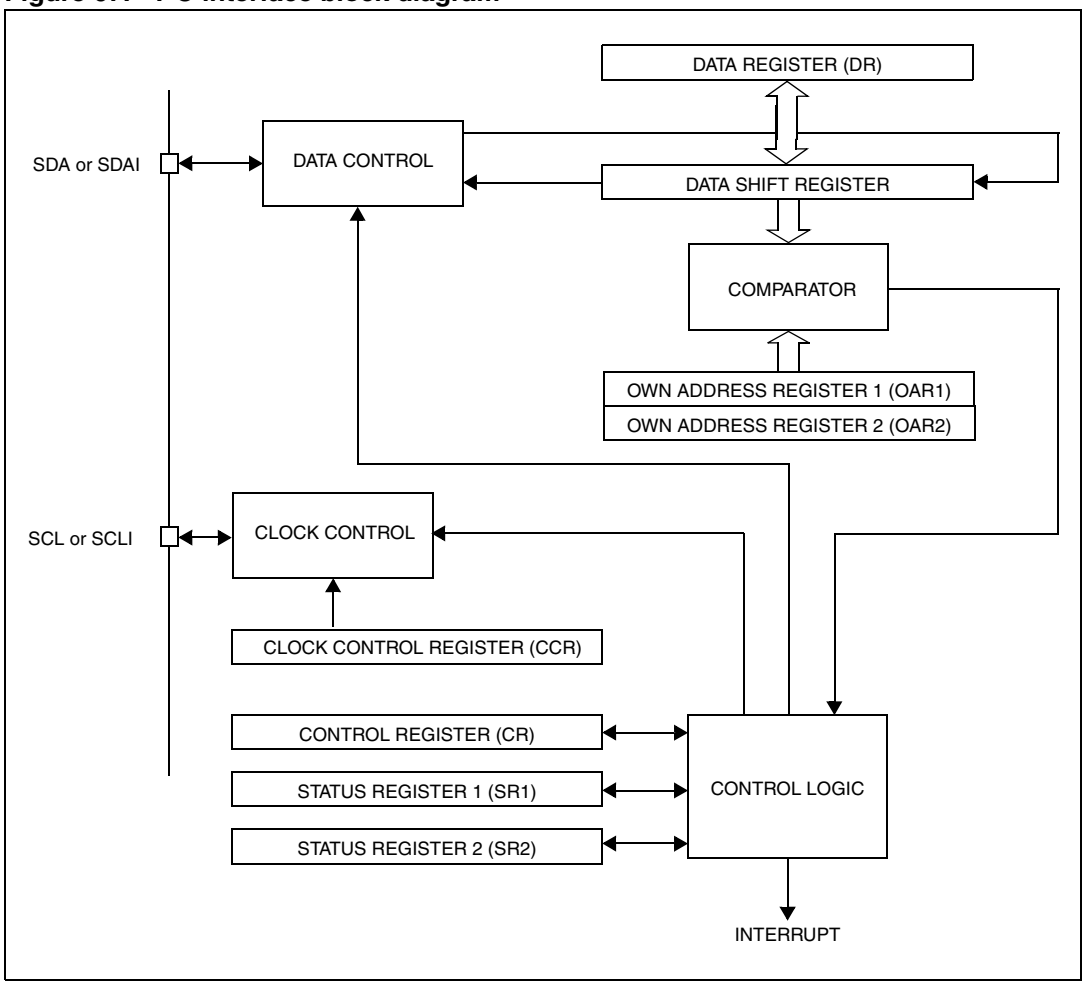
The interface holds the clock line low after reception to wait for the microcontroller to read the byte in the data register.

The SCL frequency (f_{SCL}) is controlled by a programmable clock divider which depends on the I^2C bus mode.

When the I^2C cell is enabled, the SDA and SCL ports must be configured as floating inputs. In this case, the value of the external pull-up resistor used depends on the application.

When the I^2C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 67. I^2C interface block diagram



Resync

The resynchronization mode is used to find the correct entry point for starting transmission or reception after the node has gone asynchronous either by going into the Standby or bus-off states.

Resynchronization is achieved when 128 sequences of 11 recessive bits have been monitored unless the node is not bus-off and the FSYN bit in the CSR register is set in which case a single sequence of 11 recessive bits needs to be monitored.

Idle

The CAN controller looks for one of the following events: The RUN bit is reset, a Start Of Frame appears on the CAN bus or the DATA7 register of the currently active page is written to.

Transmission

Once the LOCK bit of a Buffer Control/Status Register (BCSRx) has been set and read back as such, a transmit job can be submitted by writing to the DATA7 register. The message with the highest priority will be transmitted as soon as the CAN bus becomes idle. Among those messages with a pending transmission request, the highest priority is given to Buffer 3, then 2 and 1. If the transmission fails due to a lost arbitration or to an error while the NRTX bit of the CSR register is reset, then a new transmission attempt is performed. This goes on until the transmission ends successfully or until the job is cancelled by unlocking the buffer, by setting the NRTX bit or if the node ever enters bus-off or if a higher priority message becomes pending. The RDY bit in the BCSRx register, which was set since the job was submitted, gets reset. When a transmission is in progress, the BUSY bit in the BCSRx register is set. If it ends successfully then the TXIF bit in the Interrupt Status Register (ISR) is set, otherwise the TEIF bit is set. An interrupt is generated in either case provided the TXIE and TEIE bits of the ICR register are set.

Note: Setting the SRTE bit of the CSR register allows transmitted messages to be simultaneously received when they pass the acceptance filtering. This is particularly useful for checking the integrity of the communication path.

Reception

Once the CAN controller has synchronized itself onto the bus activity, it is ready for reception of new messages. The identifier of every incoming message is compared to the acceptance filters. If the bitwise comparison of the selected bits ends up with a match for at least one of the filters then that message is elected for reception and a target buffer is searched for. This buffer will be the first one - order is 1 to 3 - that has the LOCK and RDY bits of its BCSRx register reset.

- When no such buffer exists then an overrun interrupt is generated if the ORIE bit of the ICR register has been set. In this case the identifier of the last message is made available in the Last Identifier Register (LIDHR and LIDLR) at least until it is overwritten by a new identifier picked-up from the bus.
- When a buffer does exist, the accepted message gets written into it, the ACC bit in the BCSRx register gets the number of the matching filter, the RDY and RXIF bits get set and an interrupt is generated if the RXIE bit in the ISR register is set.

Up to three messages can be automatically received without intervention from the CPU because each buffer has its own set of status bits, greatly reducing the reactiveness requirements in the processing of the receive interrupts.

Table 120. Instructions supporting direct, indexed, indirect, and indirect indexed addressing modes (continued)

Type	Instruction	Function
Short instructions only	CLR	Clear
	INC, DEC	Increment/Decrement
	TNZ	Test Negative or Zero
	CPL, NEG	1 or 2 Complement
	BSET, BRES	Bit Operations
	BTJT, BTJF	Bit Test and Jump Operations
	SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
	SWAP	Swap Nibbles
	CALL, JP	Call or Jump subroutine

19.1.7 Relative (direct, indirect)

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

Table 121. Available relative direct/indirect instructions

Instruction	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

Relative (direct)

The offset is following the opcode.

Relative (indirect)

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

19.2 Instruction groups

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

Table 122. Instruction groups

Group	Instructions							
Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					

20.11 Communication interface characteristics

20.11.1 SPI (serial peripheral interface)

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

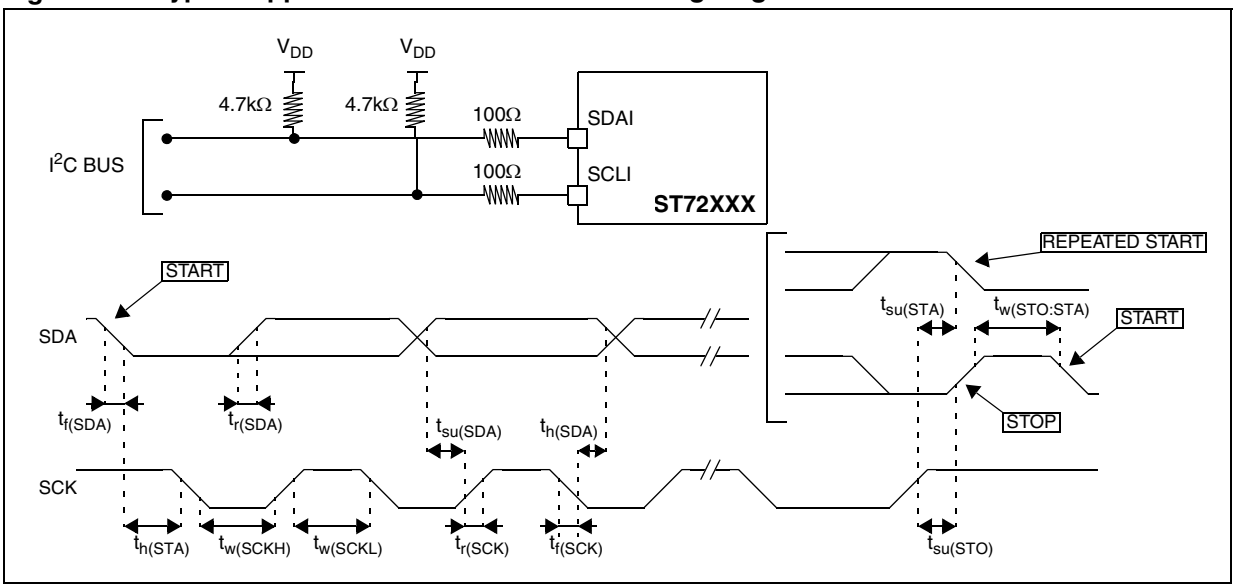
Refer to *Section 20.8: I/O port pin characteristics* for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Table 152. SPI characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master, f_{CPU} = 8 MHz	$f_{CPU}/128 = 0.0625$	$f_{CPU}/4 = 2$	MHz
		Slave, f_{CPU} = 8 MHz	0	$f_{CPU}/2 = 4$	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su(\overline{SS})}$ (1)	$\overline{SS}$ setup time(2)	Slave	t_{CPU} + 50		ns
$t_{h(\overline{SS})}$ (1)	$\overline{SS}$ hold time	Slave	120		
$t_{w(SCKH)}$ (1) $t_{w(SCKL)}$ (1)	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ (1) $t_{su(SI)}$ (1)	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ (1) $t_{h(SI)}$ (1)	Data input hold time	Master Slave	100 100		
$t_{a(SO)}$ (1)	Data output access time	Slave	0	120	
$t_{dis(SO)}$ (1)	Data output disable time	Slave		240	
$t_{v(SO)}$ (1)	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$ (1)	Data output hold time		0		
$t_{v(MO)}$ (1)	Data output valid time	Master (after enable edge)		120	t_{CPU}
$t_{h(MO)}$ (1)	Data output hold time		0		

1. Data based on design simulation and/or characterization results, not tested in production.
2. Depends on f_{CPU}. For example, if f_{CPU} = 8 MHz, then $t_{CPU} = 1 / f_{CPU}$ = 125 ns and $t_{su(\overline{SS})}$ = 175 ns.

Figure 111. Typical application with I²C BUS and timing diagram(1)



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

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

Table 154. SCL frequency table

f_{SCL} (kHz)	I2CCCR value							
	f_{CPU} = 4 MHz				f_{CPU} = 8 MHz			
	V_{DD} = 4.1V		V_{DD} = 5V		V_{DD} = 4.1V		V_{DD} = 5V	
	R_P = 3.3kΩ	R_P = 4.7kΩ	R_P = 3.3kΩ	R_P = 4.7kΩ	R_P = 3.3kΩ	R_P = 4.7kΩ	R_P = 3.3kΩ	R_P = 4.7kΩ
400	Not achievable				83h			
300	Not achievable				85h			
200	83h				8Ah	89h	8Ah	
100	10h				24h	23h	24h	23h
50	24h				4Ch			
20	5Fh				FFh			

Legend:

R_P = External pull-up resistance

f_{SCL} = I²C speed

Note: - For speeds around 200 kHz, the achieved speed can have a ±5% tolerance.
- For other speed ranges, the achieved speed can have a ±2% tolerance.

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

Figure 119. 64-pin (10x10) low profile quad flat package outline

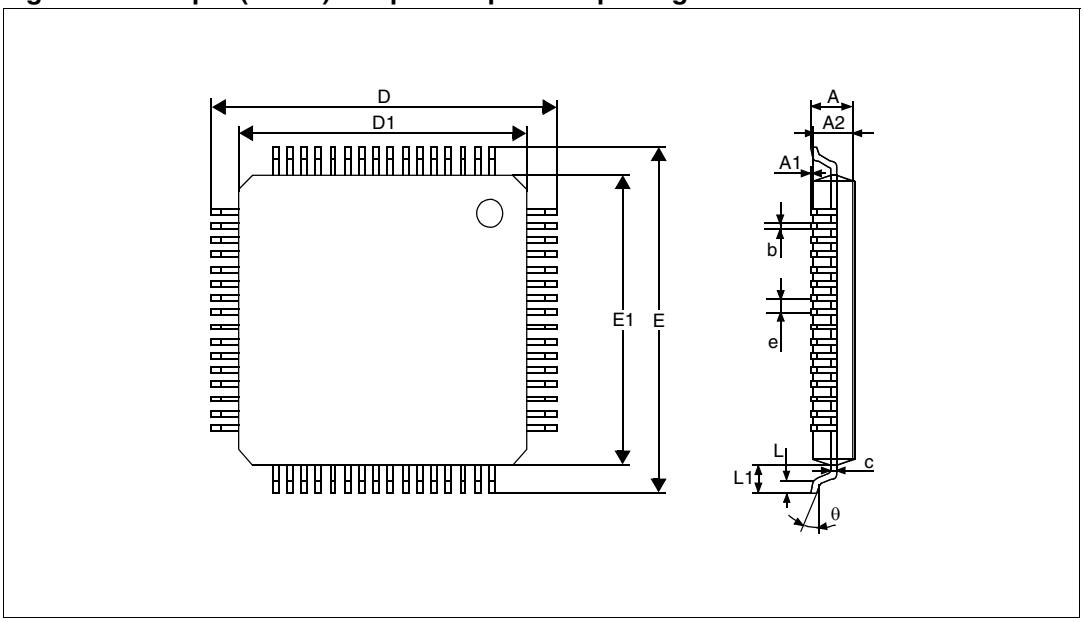


Table 160. 64-pin (10x10) low profile quad flat package mechanical data

Dimension	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A			1.600			0.0630
A1	0.050		0.150	0.0020		0.0059
A2	1.350	1.400	1.450	0.0531	0.0551	0.0571
b	0.170	0.220	0.270	0.0067	0.0087	0.0106
c	0.090		0.200	0.0035		0.0079
D		12.000			0.4724	
D1		10.000			0.3937	
E		12.000			0.4724	
E1		10.000			0.3937	
e		0.500			0.0197	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.450	0.600	0.750	0.0177	0.0236	0.0295
L1		1.000			0.0394	

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

22.2 ROM device ordering information and transfer of customer code

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

Complete the appended *ST72521-Auto Microcontroller FASTROM/ROM Option List on page 264* to communicate the selected options to STMicroelectronics and check for regular updates of the option list on the ST website or ask your ST representative.

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

The following *Figure 122* and *Figure 123* serve as guides for ordering. The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Caution: The Readout Protection binary value is inverted between ROM and Flash products. The option byte checksum will differ between ROM and Flash.

23.1.9 TIMD set simultaneously with OC interrupt

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

Impact on the application

If the 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 = 0x00h; // Disable the compare interrupt
  - TACSR | or TBCSR | = 0x40; // Disable the timer
- Perform the following to enable the timer again:
  - TACSR & or TBCSR & = ~0x40; // Enable the timer
  - TACR1 or TBCR1 = 0x40; // Enable the compare interrupt

23.1.10 CAN cell limitations

Table 169. CAN cell limitations

Limitation(1)	Flash	ROM
Omitted SOF bit	x	x
CPU write access (more than one cycle) corrupts CAN frame	x	x
Unexpected Message transmission	x(2)	
Bus Off State not entered		x(3)
WKPS functionality		x(4)

1. For details see *Section 17.5: List of CAN cell limitations on page 196*
2. Software workaround possible using modified WKPS bit.
3. Limitation present on ROM Rev W and Rev Z. Not present in Flash and ROM Rev Y.
4. Functionality modified for Unexpected Message Transmission workaround in Flash.

Legend:

x = limitation present

23.1.11 I^2C 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 restart and Slave address to re-initiate communication.