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	16MHz
Connectivity	I²C, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	22
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3.2V ~ 5.5V
Data Converters	A/D 6x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72c254g2m3-tr

3 REGISTER & MEMORY MAP

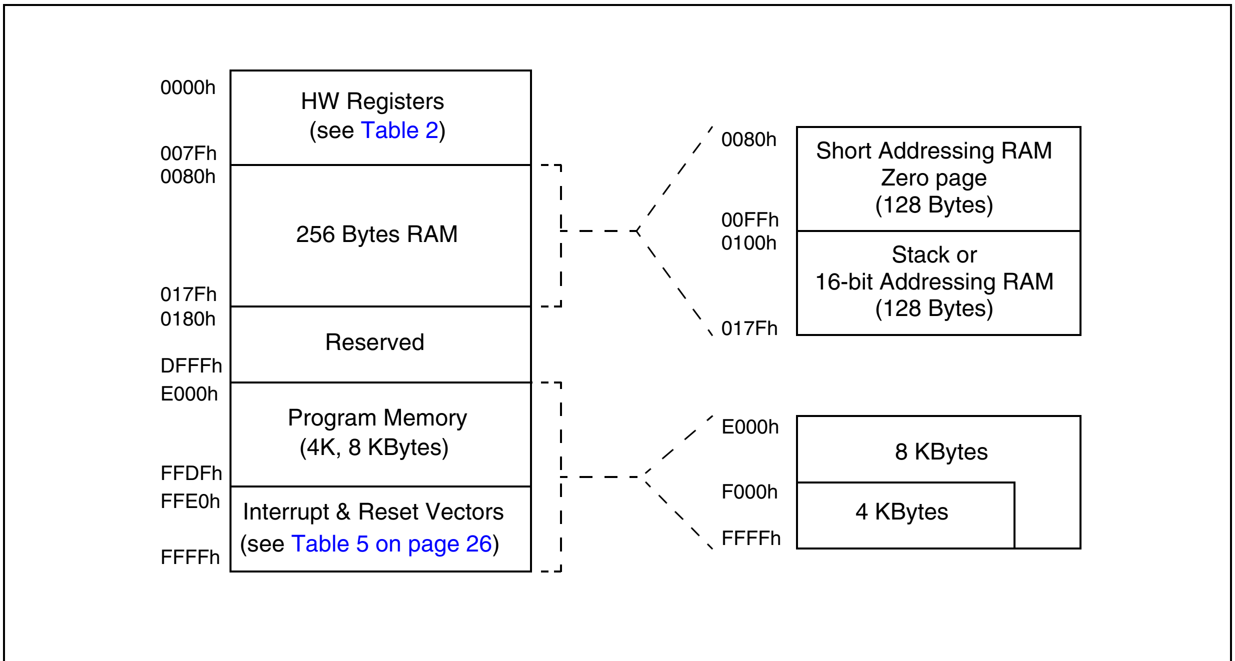
As shown in the Figure 4, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register location, 256 bytes of RAM and up to 8Kbytes of user program memory. The RAM space includes up to 128 bytes for the stack from 0100h to 017Fh.

The highest address bytes contain the user reset and interrupt vectors.

IMPORTANT: Memory locations marked as “Reserved” must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Figure 4. Memory Map



5 CENTRAL PROCESSING UNIT

5.1 INTRODUCTION

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

5.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

5.3 CPU REGISTERS

The six CPU registers shown in Figure 1 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

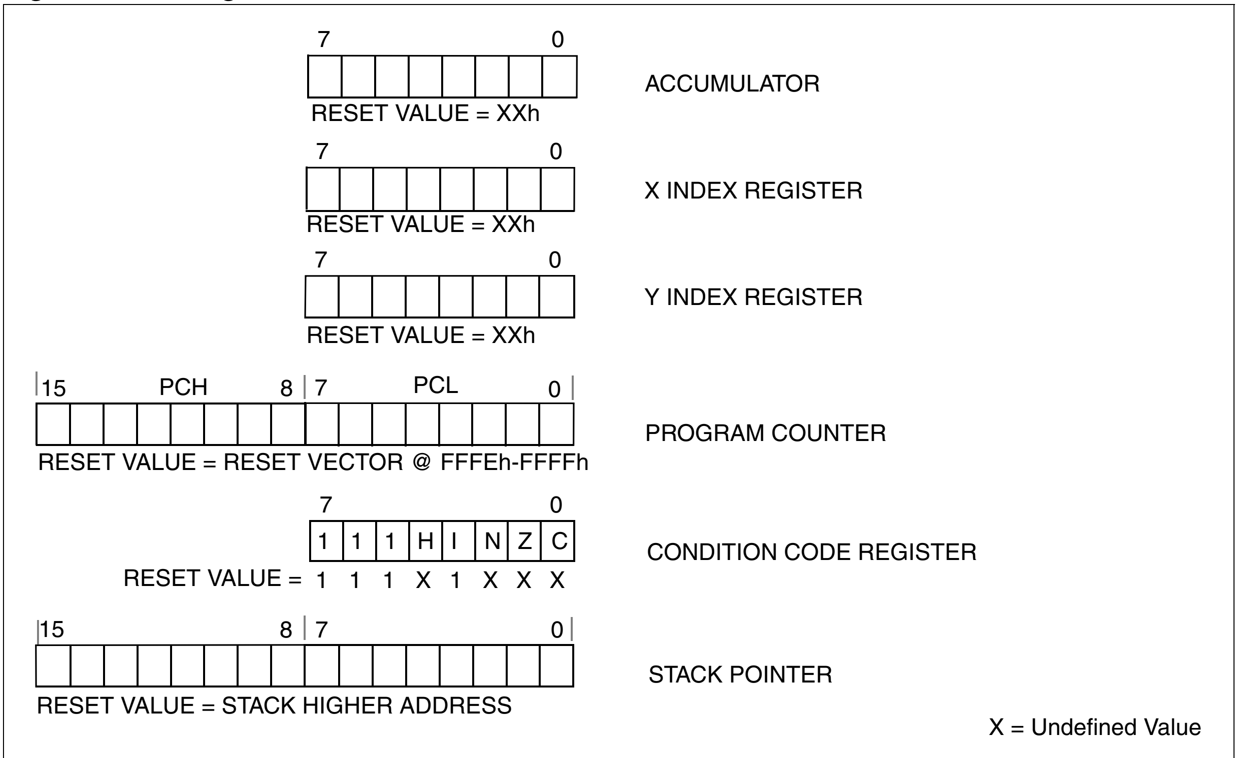
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

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

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 6. CPU Registers



6.4 CLOCK SECURITY SYSTEM (CSS)

The Clock Security System (CSS) protects the ST7 against main clock problems. To allow the integration of the security features in the applications, it is based on a clock filter control and an Internal safe oscillator. The CSS can be enabled or disabled by option byte.

6.4.1 Clock Filter Control

The clock filter is based on a clock frequency limitation function.

This filter function is able to detect and filter high frequency spikes on the ST7 main clock.

If the oscillator is not working properly (e.g. working at a harmonic frequency of the resonator), the current active oscillator clock can be totally filtered, and then no clock signal is available for the ST7 from this oscillator anymore. If the original clock source recovers, the filtering is stopped automatically and the oscillator supplies the ST7 clock.

6.4.2 Safe Oscillator Control

The safe oscillator of the CSS block is a low frequency back-up clock source (see Figure 13).

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

Automatically, the ST7 clock source switches back from the safe oscillator if the original clock source recovers.

Limitation detection

The automatic safe oscillator selection is notified by hardware setting the CSSD bit of the CRSR register. An interrupt can be generated if the CSSIE bit has been previously set.
These two bits are described in the CRSR register description.

6.4.3 Low Power Modes

Mode	Description
WAIT	No effect on CSS. CSS interrupt cause the device to exit from Wait mode.
HALT	The CRSR 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.

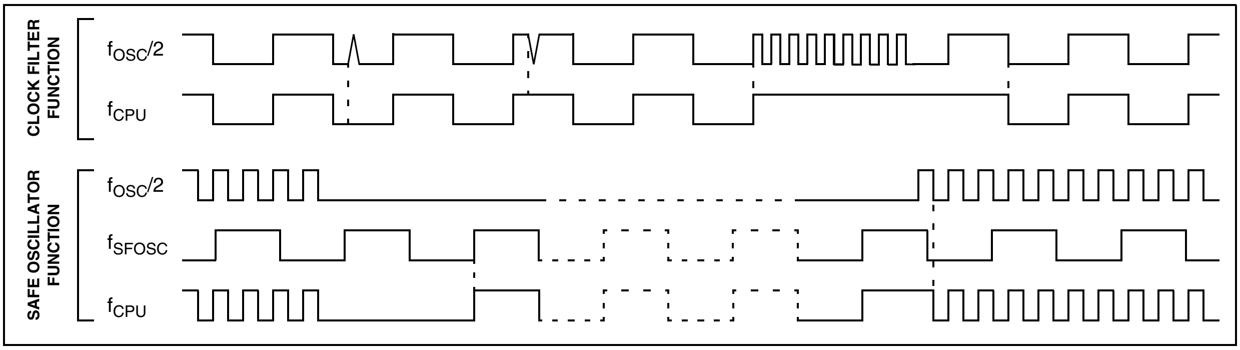
6.4.4 Interrupts

The CSS interrupt event generates an interrupt if the corresponding Enable Control Bit (CSSIE) 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[1)]
CSS event detection (safe oscillator activated as main clock)	CSSD	CSSIE	Yes	No

Note 1: This interrupt allows to exit from active-halt mode if this mode is available in the MCU.

Figure 13. Clock Filter Function and Safe Oscillator Function



I/O PORTS (Cont'd)

CAUTION: The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

Analog alternate function

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

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

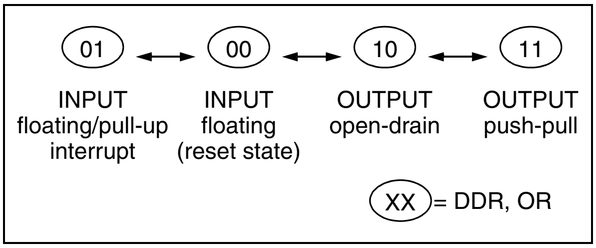
WARNING: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

9.3 I/O PORT IMPLEMENTATION

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

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in Figure 2. Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 22. Interrupt I/O Port State Transitions



The I/O port register configurations are summarized as follows.

Interrupt Ports

PA7, PA5, PA3:0, PB7:0, PC5:0 (with pull-up)

MODE	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	1

True Open Drain Interrupt Ports

PA6, PA4 (without pull-up)

MODE	DDR	OR
floating input	0	0
floating interrupt input	0	1
open drain (high sink ports)	1	X

Table 8. Port Configuration

Port	Pin name	Input (DDR = 0)		Output (DDR = 1)		
		OR = 0	OR = 1	OR = 0	OR = 1	High-Sink
Port A	PA7	floating	pull-up interrupt	open drain	push-pull	Yes
	PA6	floating	floating interrupt	true open-drain		
	PA5	floating	pull-up interrupt	open drain	push-pull	
	PA4	floating	floating interrupt	true open-drain		
	PA3:0	floating	pull-up interrupt	open drain	push-pull	
Port B	PB7:0	floating	pull-up interrupt	open drain	push-pull	No
Port C	PC7:0	floating	pull-up interrupt	open drain	push-pull	

MISCELLANEOUS REGISTERS (Cont'd)

10.3 MISCELLANEOUS REGISTER DESCRIPTION

MISCELLANEOUS REGISTER 1 (MISCR1)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
IS11	IS10	MCO	IS01	IS00	CP1	CP0	SMS

Bit 7:6 = **IS1[1:0]** *ei1 sensitivity*
The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the ei1 external interrupts. These two bits can be written only when the I bit of the CC register is set to 1 (interrupt masked).

ei1: Port B (C optional)

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

Bit 5 = **MCO** *Main clock out selection*
This bit enables the MCO alternate function on the PC2 I/O port. It is set and cleared by software.
0: MCO alternate function disabled (I/O pin free for general-purpose I/O)
1: MCO alternate function enabled (f_{CPU} on I/O port)

Bit 4:3 = **IS0[1:0]** *ei0 sensitivity*
The interrupt sensitivity, defined using the IS0[1:0] bits, is applied to the ei0 external interrupts. These two bits can be written only when the I bit of the CC register is set to 1 (interrupt masked).

ei0: Port A (C optional)

External Interrupt Sensitivity	IS01	IS00
Falling edge & low level	0	0
Rising edge only	0	1
Falling edge only	1	0
Rising and falling edge	1	1

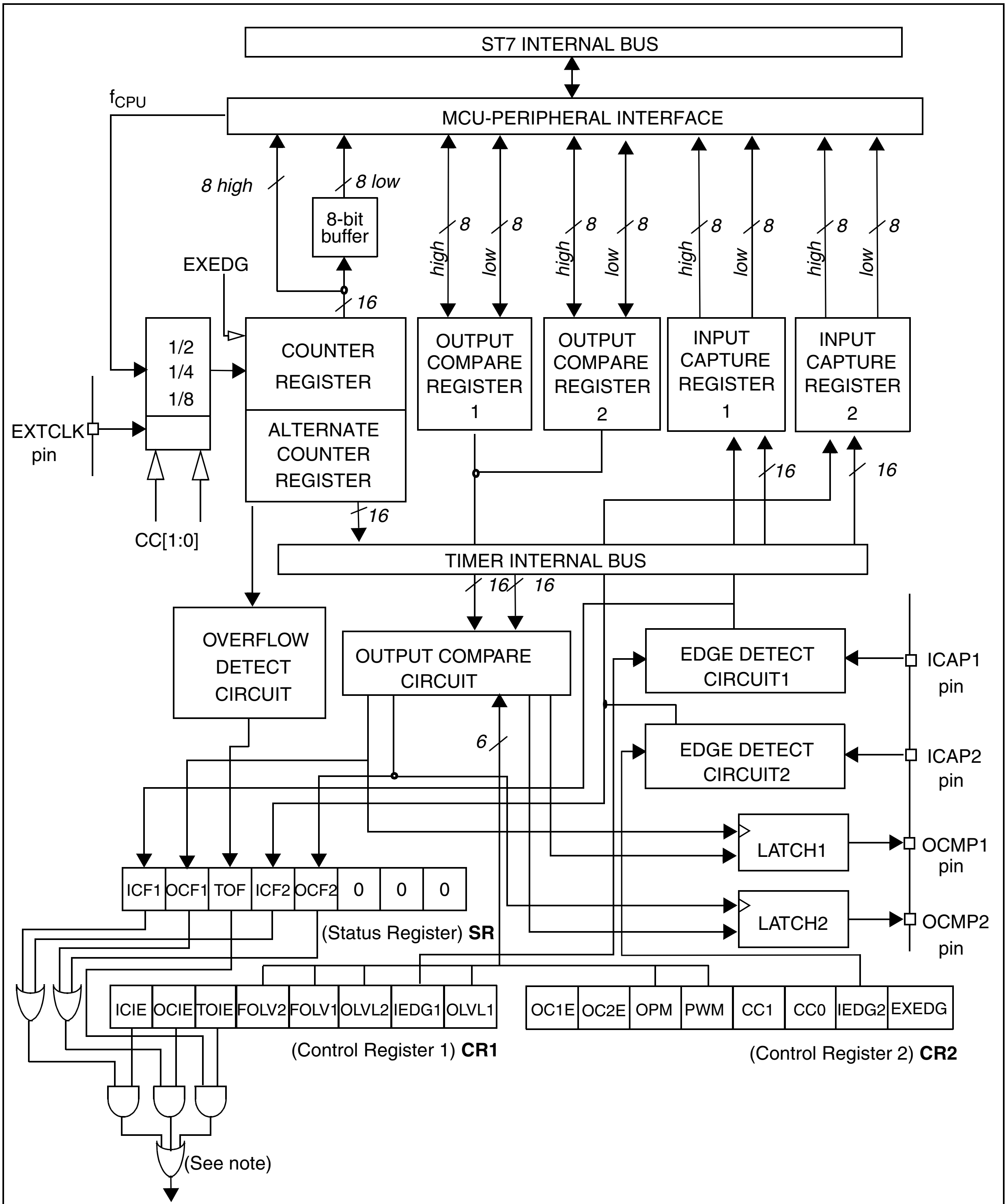
Bit 2:1 = **CP[1:0]** *CPU clock prescaler*
These bits select the CPU clock prescaler which is applied in the different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software

f_{CPU} in SLOW mode	CP1	CP0
f_{OSC} / 4	0	0
f_{OSC} / 8	1	0
f_{OSC} / 16	0	1
f_{OSC} / 32	1	1

Bit 0 = **SMS** *Slow mode select*
This bit is set and cleared by software.
0: Normal mode. $f_{CPU} = f_{OSC} / 2$
1: Slow mode. f_{CPU} is given by CP1, CP0
See low power consumption mode and MCC chapters for more details.

16-BIT TIMER (Cont'd)

Figure 26. Timer Block Diagram



Note: If IC, OC and TO interrupt requests have separate vectors then the last OR is not present (See device Interrupt Vector Table)

16-BIT TIMER (Cont'd)

CONTROL REGISTER 2 (CR2)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *Output Compare 1 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in Output Compare mode, both OLV1 and OLV2 in PWM and one-pulse mode). Whatever the value of the OC1E bit, the internal Output Compare 1 function of the timer remains active.
0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP1 pin alternate function enabled.

Bit 6 = **OC2E** *Output Compare 2 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in Output Compare mode). Whatever the value of the OC2E bit, the internal Output Compare 2 function of the timer remains active.
0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP2 pin alternate function enabled.

Bit 5 = **OPM** *One Pulse mode.*
0: One Pulse mode is not active.
1: One Pulse mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.

Bit 4 = **PWM** *Pulse Width Modulation.*
0: PWM mode is not active.
1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.

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

The timer clock mode depends on these bits:

Table 13. Clock Control Bits

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

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

Bit 1 = **IEDG2** *Input Edge 2.*
This bit determines which type of level transition on the ICAP2 pin will trigger the capture.
0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **EXEDG** *External Clock Edge.*
This bit determines which type of level transition on the external clock pin (EXTCLK) will trigger the counter register.
0: A falling edge triggers the counter register.
1: A rising edge triggers the counter register.

11.3 SERIAL PERIPHERAL INTERFACE (SPI)

11.3.1 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 or a system in which devices may be either masters or slaves.

The SPI is normally used for communication between the microcontroller and external peripherals or another microcontroller.

Refer to the Pin Description chapter for the device-specific pin-out.

11.3.2 Main Features

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- Four master mode frequencies
- Maximum slave mode frequency = $f_{CPU}/4$
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision flag protection
- Master mode fault protection capability

11.3.3 General description

The SPI is connected to external devices through 4 alternate pins:

– MISO: Master In Slave Out pin

– MOSI: Master Out Slave In pin

– SCK: Serial Clock pin

– $\overline{SS}$: Slave select pin

A basic example of interconnections between a single master and a single slave is illustrated on Figure 1.

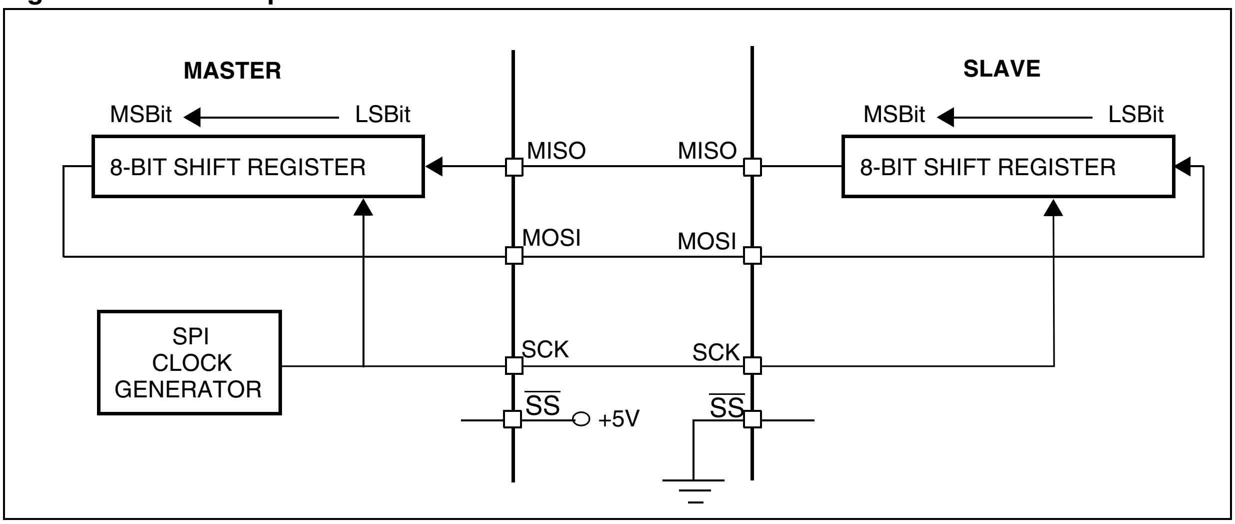
The MOSI pins are connected together as are MISO pins. In this way data is transferred serially between master and slave (most significant bit first).

When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full bits. A status flag is used to indicate that the I/O operation is complete.

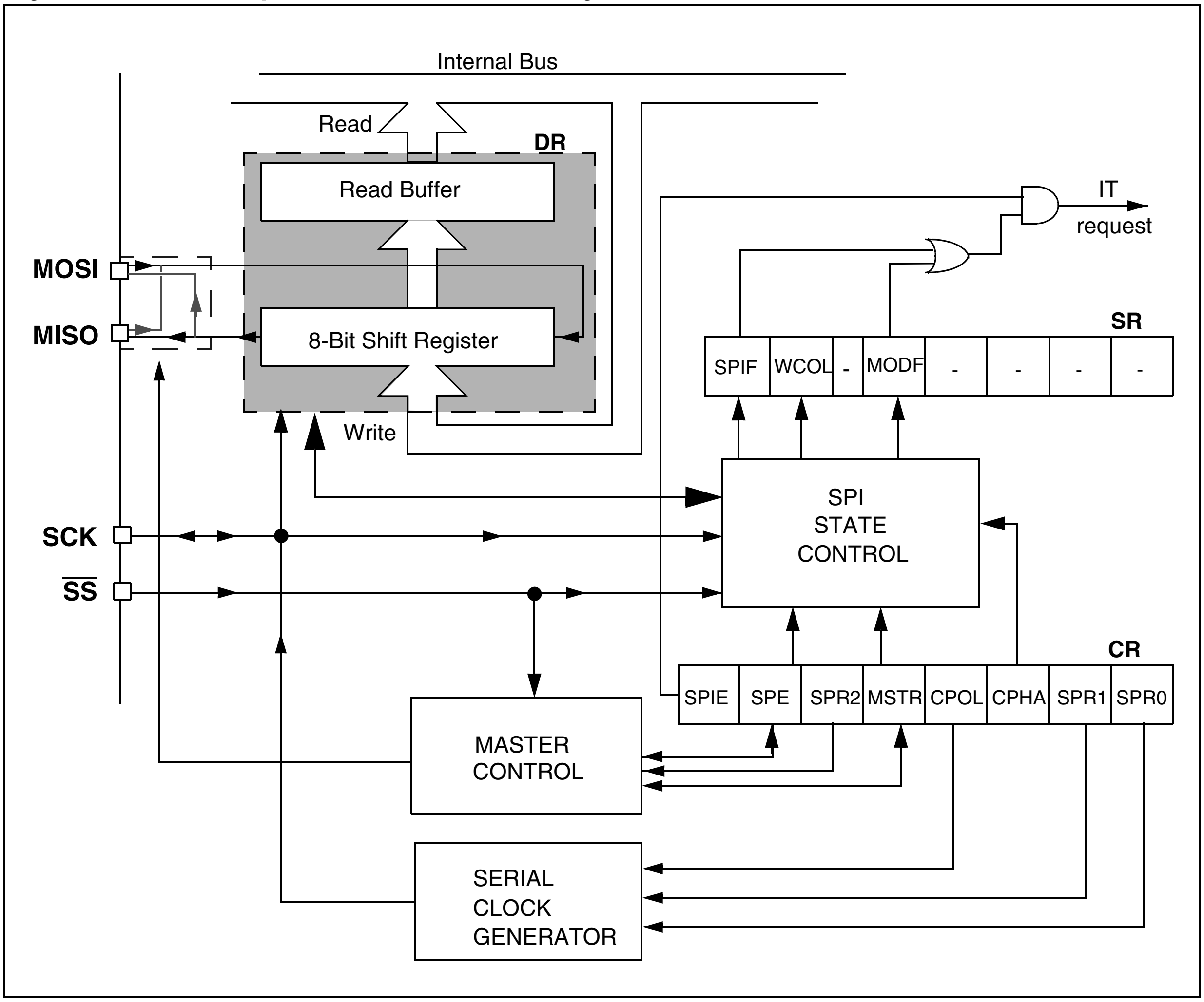
Four possible data/clock timing relationships may be chosen (see Figure 4) but master and slave must be programmed with the same timing mode.

Figure 37. Serial Peripheral Interface Master/Slave



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 38. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

11.3.5 Low Power Modes

Mode	Description
WAIT	No effect on SPI. SPI interrupt events cause the device to exit from WAIT mode.
HALT	SPI registers are frozen. In HALT mode, the SPI is inactive. SPI operation resumes when the MCU is woken up by an interrupt with "exit from HALT mode" capability.

11.3.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
SPI End of Transfer Event	SPIF	SPIE	Yes	No
Master Mode Fault Event	MODF		Yes	No

Note: The SPI 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 interrupt mask in the CC register is reset (RIM instruction).

I²C BUS INTERFACE (Cont'd)

I²C STATUS REGISTER 1 (SR1)

Read Only

Reset Value: 0000 0000 (00h)

7							0
EVF	ADD10	TRA	BUSY	BTF	ADSL	M/SL	SB

Bit 7 = **EVF** *Event flag.*
This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in Figure 3. It is also cleared by hardware when the interface is disabled (PE=0).
0: No event
1: One of the following events has occurred:

- BTF=1 (Byte received or transmitted)
- ADSL=1 (Address matched in Slave mode while ACK=1)
- SB=1 (Start condition generated in Master mode)
- AF=1 (No acknowledge received after byte transmission)
- STOPF=1 (Stop condition detected in Slave mode)
- ARLO=1 (Arbitration lost in Master mode)
- BERR=1 (Bus error, misplaced Start or Stop condition detected)
- ADD10=1 (Master has sent header byte)
- Address byte successfully transmitted in Master mode.

Bit 6 = **ADD10** *10-bit addressing in Master mode.*
This bit is set by hardware when the master has sent the first byte in 10-bit address mode. It is cleared by software reading SR2 register followed by a write in the DR register of the second address byte. It is also cleared by hardware when the peripheral is disabled (PE=0).

0: No ADD10 event occurred.
1: Master has sent first address byte (header)

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

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

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

- Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV6 event (See Figure 3). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.
- Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

0: Byte transfer not done
1: Byte transfer succeeded

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

The SCL line is held low while ADSL=1.

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

I²C 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

I²C 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

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 I²C 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

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

Table 18. ADC Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0070h	**ADCDR** Reset Value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0071h	**ADCCSR** Reset Value	COCO 0	0	ADON 0	0	CH3 0	CH2 0	CH1 0	CH0 0

INSTRUCTION GROUPS (Cont'd)

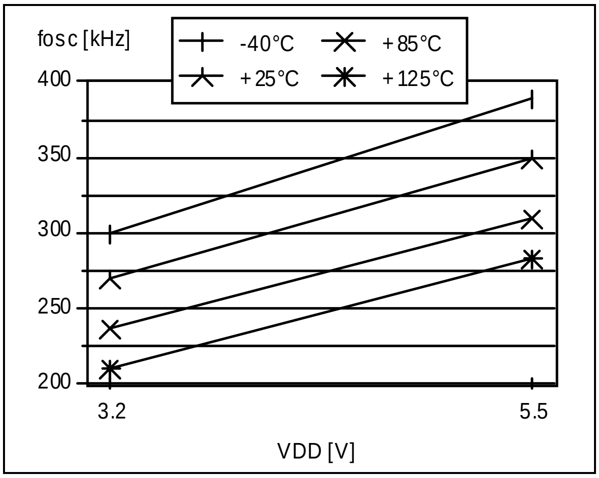
Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A . M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

CLOCK CHARACTERISTICS (Cont'd)

13.5.5 Clock Security System (CSS)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SFOSC}	Safe Oscillator Frequency [1)]	T_A=25°C, V_{DD}=5.0V	250	340	550	kHz
		T_A=25°C, V_{DD}=3.4V	190	260	450	
f_{GFOSC}	Glitch Filtered Frequency [2)]			30		MHz

Figure 69. Typical Safe Oscillator Frequencies



Note:

1. Data based on characterization results, tested in production between 90KHz and 600KHz.
2. Filtered glitch on the f_{OSC} signal. See functional description in Section 6.5 on page 23 for more details.

COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

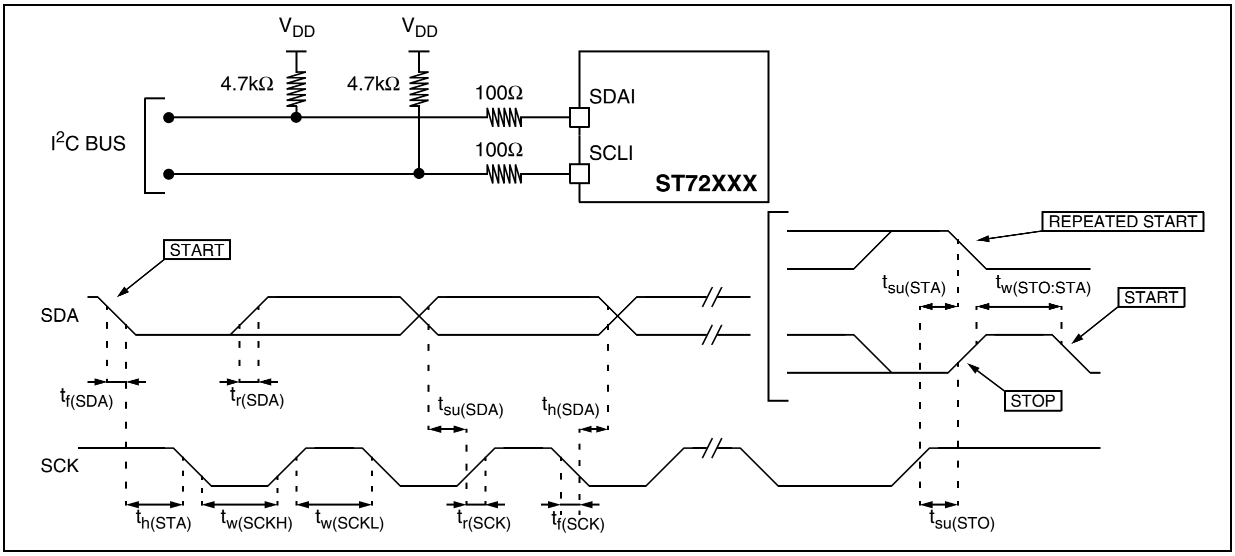
13.11.2 I²C - Inter IC Control Interface

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 (SDAI and SCLI). The ST7 I²C interface meets the requirements of the Standard I²C communication protocol described in the following table.

Symbol	Parameter	Standard mode I²C		Fast mode I²C		Unit
		Min [1)]	Max [1)]	Min [1)]	Max [1)]	
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		µs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		µs
$t_{su(SDA)}$	SDA setup time	250		100		ns
$t_{h(SDA)}$	SDA data hold time	0 [3)]		0 [2)]	900 [3)]	ns
$t_{r(SDA)}$ $t_{r(SCL)}$	SDA and SCL rise time		1000	$20+0.1C_b$	300	ns
$t_{f(SDA)}$ $t_{f(SCL)}$	SDA and SCL fall time		300	$20+0.1C_b$	300	ns
$t_{h(STA)}$	START condition hold time	4.0		0.6		µs
$t_{su(STA)}$	Repeated START condition setup time	4.7		0.6		µs
$t_{su(STO)}$	STOP condition setup time	4.0		0.6		ns
$t_{w(STO:STA)}$	STOP to START condition time (bus free)	4.7		1.3		ms
C_b	Capacitive load for each bus line		400		400	pF

Figure 94. Typical Application with I²C Bus and Timing Diagram [4)]



Notes:

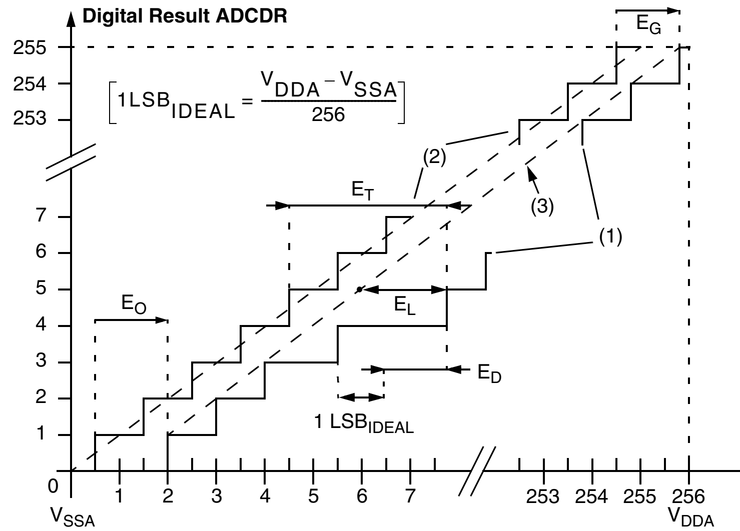
1. Data based on standard I²C protocol requirement, not tested in production.
2. The device must internally provide a hold time of at least 300ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.
3. The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal.
4. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

8-BIT ADC CHARACTERISTICS (Cont'd)

ADC Accuracy

Symbol	Parameter	V_{DD}=5V, [2)] f_{CPU}=1MHz		V_{DD}=5.0V, [3)] f_{CPU}=8MHz		V_{DD}=3.3V, [3)] f_{CPU}=8MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\|E_T\|$	Total unadjusted error [1)]		2.0		2.0		2.0	LSB
E_O	Offset error [1)]		1.5		1.5		1.5	
E_G	Gain Error [1)]		1.5		1.5		1.5	
$\|E_D\|$	Differential linearity error [1)]		1.5		1.5		1.5	
$\|E_L\|$	Integral linearity error [1)]		1.5		1.5		1.5	

Figure 96. ADC Accuracy Characteristics



(1) Example of an actual transfer curve
(2) The ideal transfer curve
(3) End point correlation line

E_T=Total Unadjusted Error: maximum deviation between the actual and the ideal transfer curves.
E_O=Offset Error: deviation between the first actual transition and the first ideal one.
E_G=Gain Error: deviation between the last ideal transition and the last actual one.
E_D=Differential Linearity Error: maximum deviation between actual steps and the ideal one.
E_L=Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

Notes:

1. ADC Accuracy vs. Negative Injection Current:
For I_{INJ-}=0.8mA, the typical leakage induced inside the die is 1.6µA and the effect on the ADC accuracy is a loss of 1 LSB for each 10KΩ increase of the external analog source impedance. This effect on the ADC accuracy has been observed under worst-case conditions for injection:
- negative injection
- injection to an Input with analog capability, adjacent to the enabled Analog Input
- at 5V V_{DD} supply, and worst case temperature.
2. Data based on characterization results with T_A=25°C.
3. Data based on characterization results over the whole temperature range.

14.3 SOLDERING INFORMATION

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

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

15 DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in user programmable versions (FLASH) as well as in factory coded versions (ROM). FLASH devices are shipped to customers with a default content (FFh), while ROM factory coded parts contain the code supplied by the customer. This implies that FLASH devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

15.1 OPTION BYTES

The two option bytes allow the hardware configuration of the microcontroller to be selected.
The option bytes have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The default content of the FLASH is fixed to FFh.
In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see option list).

USER OPTION BYTE 0

Bit 7:2 = **Reserved**, must always be 1.

Bit 1 = **EXTIT** *External Interrupt Configuration.*
This option bit allows the external interrupt mapping to be configured as shown in Table 23.

Table 23. External Interrupt Configuration

External IT0	External IT1	EXTIT
Ports PA7-PA0	Ports PB7-PB0 Ports PC5-PC0	1
Ports PA7-PA0 Ports PC5-PC0	Ports PB7-PB0	0

Bit 0 = **FMP** *Full memory protection.*
This option bit enables or disables external access to the internal program memory (read-out protection). Clearing this bit causes the erasing (by overwriting with the currently latched values) of the whole memory (not including the option bytes).
0: Program memory not read-out protected
1: Program memory read-out protected

USER OPTION BYTE 1

Bit 7 = **CFC** *Clock filter control on/off*
This option bit enables or disables the clock filter (CF) features.
0: Clock filter enabled
1: Clock filter disabled

Bit 6:4 = **OSC[2:0]** *Oscillator selection*
These three option bits can be used to select the main oscillator as shown in Table 24.

Bit 3:2 = **LVD[1:0]** *Low voltage detection selection*
These option bits enable the LVD block with a selected threshold as shown in Table 25.

Bit 1 = **WDG HALT** *Watchdog and halt mode*
This option bit determines if a RESET is generated when entering HALT mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

Bit 0 = **WDG SW** *Hardware or software watchdog*
This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

Table 24. Main Oscillator Configuration

Selected Oscillator	OSC2	OSC1	OSC0
External Clock (Stand-by)	1	1	1
~4 MHz Internal RC	1	1	0
1~14 MHz External RC	1	0	X
Low Power Resonator (LP)	0	1	1
Medium Power Resonator (MP)	0	1	0
Medium Speed Resonator (MS)	0	0	1
High Speed Resonator (HS)	0	0	0

Table 25. LVD Threshold Configuration

Configuration	LVD1	LVD0
LVD Off	1	1
Highest Voltage Threshold (~4.50V)	1	0
Medium Voltage Threshold (~4.05V)	0	1
Lowest Voltage Threshold (~3.45V)	0	0

	USER OPTION BYTE 0 7							0	USER OPTION BYTE 1 7							0
	Reserved						EXTIT	FMP	CFC	OSC 2	OSC 1	OSC 0	LVD1	LVD0	WDG HALT	WDG SW
Default Value	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1

DEVELOPMENT TOOLS (Cont'd)

15.3.1 PACKAGE/SOCKET FOOTPRINT PROPOSAL

Table 28. Suggested List of SDIP32 Socket Types

Package / Probe	Adaptor / Socket Reference		Same Footprint	Socket Type
SDIP32 EMU PROBE	TEXTOOL	232-1291-00	X	Textool

Table 29. Suggested List of SO28 Socket Types

Package / Probe	Adaptor / Socket Reference		Same Footprint	Socket Type
SO28	ENPLAS	OTS-28-1.27-04		Open Top
	YAMAICHI	IC51-0282-334-1		Clamshell
EMU PROBE	Adapter from SO28 to SDIP32 footprint (delivered with emulator)		X	SMD to SDIP