E·XFL



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	16MHz
Connectivity	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	22
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-50
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f260g1m6-tr

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

2 PIN DESCRIPTION

Figure 2. 28-Pin SO Package Pinout



Figure 3. 32-Pin SDIP Package Pinout



FLASH PROGRAM MEMORY (Cont'd)

4.4 ICC interface

ICP needs a minimum of 4 and up to 7 pins to be connected to the programming tool. These pins are:

- RESET: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- ICCSEL: ICC selection (not required on devices without ICCSEL pin)
- OSC1: main clock input for external source (not required on devices without OSC1/OSC2 pins)
- V_{DD}: application board power supply (optional, see Note 3)

Notes:

1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.

2. During the ICP session, the programming tool must control the RESET pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (push pull output or pull-up resistor<1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R>1K or a reset management IC with open drain output and pull-up resistor>1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.

3. The use of Pin 7 of the ICC connector depends on the Programming Tool architecture. This pin must be connected when using most ST Programming Tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

4. Pin 9 has to be connected to the OSC1 pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.



Figure 6. Typical ICC Interface

6.3 RESET SEQUENCE MANAGER (RSM)

6.3.1 Introduction

The reset sequence manager includes three RE-SET sources as shown in Figure 12:

- External RESET source pulse
- Internal LVD RESET (Low Voltage Detection)
- Internal WATCHDOG RESET

These sources act on the RESET pin and it is always kept low during the delay phase.

The RESET service routine vector is fixed at addresses FFFEh-FFFFh in the ST7 memory map.

The basic RESET sequence consists of 3 phases as shown in Figure 11:

- Active Phase depending on the RESET source
- 4096 CPU clock cycle delay (selected by option byte)
- RESET vector fetch

The 4096 CPU clock cycle delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state. The shorter or longer clock cycle delay should be selected by option byte to correspond to the stabilization time of the external oscillator used in the application.

Figure 12. Reset Block Diagram

The RESET vector fetch phase duration is 2 clock cycles.

Figure 11. RESET Sequence Phases



6.3.2 Asynchronous External RESET pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor. This pull-up has no fixed value but varies in accordance with the input voltage. It can be pulled low by external circuitry to reset the device. See Electrical Characteristic section for more details.

A RESET signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized (see Figure 13). This detection is asynchronous and therefore the MCU can enter reset state even in HALT mode.



SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.2 Auxiliary Voltage Detector (AVD)

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

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

Caution: The AVD functions only if the LVD is enabled through the option byte.

6.4.2.1 Monitoring the V_{DD} Main Supply

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see Section 15.1 on page 162).

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

Figure 15. Using the AVD to Monitor V_{DD}

57/

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See Figure 15.

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

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

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

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



7 INTERRUPTS

7.1 INTRODUCTION

The ST7 enhanced interrupt management provides the following features:

- Hardware interrupts
- Software interrupt (TRAP)
- Nested or concurrent interrupt management with flexible interrupt priority and level management:
 - Up to 4 software programmable nesting levels
 - Up to 16 interrupt vectors fixed by hardware
- 2 non-maskable events: RESET and TRAP
 This interrupt management is based on:
- Bit 5 and bit 3 of the CPU CC register (I1:0),
- Interrupt software priority registers (ISPRx),
- Fixed interrupt vector addresses located at the high addresses of the memory map (FFE0h to FFFFh) sorted by hardware priority order.

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

7.2 MASKING AND PROCESSING FLOW

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see Table 4). The processing flow is shown in Figure 16

Figure 16. Interrupt Processing Flowchart

When an interrupt request has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to "Interrupt Mapping" table for vector addresses).

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

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

Table 4. Interrupt Software Priority Levels

Interrupt software priority	Level	l1	10
Level 0 (main)	Low	1	0
Level 1		0	1
Level 2	•	0	0
Level 3 (= interrupt disable)	High	1	1



MISCELLANEOUS REGISTERS (Cont'd)

10.3 MISCELLANEOUS REGISTER DESCRIPTION

MISCELLANEOUS REGISTER 1 (MISCR1)

Read/Write

Reset Value: 0000 0000 (00h)

7

0

	l.		l.				l.
IS11	IS10	мсо	IS01	IS00	CP1	CP0	SMS

Bits 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[1:0] bits in the CC register are 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)

Bits 4:3 = ISO[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[1:0] bits in-the CC register are 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

Bits 2:1 = CP[1:0] CPU clock prescaler

These bits select the CPU clock prescaler which is applied in the various 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 _{OSC2} / 2	0	0
f _{OSC2} / 4	1	0
f _{OSC2} / 8	0	1
f _{OSC2} / 16	1	1

Bit 0 = **SMS** *Slow mode select*

This bit is set and cleared by software.

0: Normal mode. $f_{CPU} = f_{OSC2}$

1: Slow mode. f_{CPU} is given by CP1, CP0

See low power consumption mode and MCC chapters for more details.



11.2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK (MCC/RTC)

The Main Clock Controller consists of a real time clock timer with interrupt capability

11.2.1 Real Time Clock Timer (RTC)

5/

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 MCC-SR 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" on page 35 for more details.

Figure 34. Main Clock Controller (MCC/RTC) Block Diagram



16-BIT TIMER (Cont'd)

Related Documentation

AN 973: SCI software communications using 16bit timer

AN 974: Real Time Clock with ST7 Timer Output Compare

AN 976: Driving a buzzer through the ST7 Timer PWM function

AN1041: Using ST7 PWM signal to generate analog input (sinusoid)

AN1046: UART emulation software

AN1078: PWM duty cycle switch implementing true 0 or 100 per cent duty cycle

AN1504: Starting a PWM signal directly at high level using the ST7 16-Bit timer



SERIAL PERIPHERAL INTERFACE (Cont'd)

11.4.5.4 Single Master and Multimaster Configurations

There are two types of SPI systems:

- Single Master System
- Multimaster System

Single Master System

A typical single master system may be configured, using a device as the master and four devices as slaves (see Figure 52).

The master device selects the individual slave devices by using four pins of a parallel port to control the four 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.

Multi-Master System

A multi-master system may also be configured by the user. Transfer of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system.

The multi-master system is principally handled by the MSTR bit in the SPICR register and the MODF bit in the SPICSR register.





SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 53. SCI Block Diagram



۸Ţ/

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

11.5.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 53. It contains 6 dedicated registers:

- Two control registers (SCICR1 & SCICR2)
- A status register (SCISR)

57

- A baud rate register (SCIBRR)
- An extended prescaler receiver register (SCIER-PR)
- An extended prescaler transmitter register (SCI-ETPR)

Refer to the register descriptions in Section 11.5.7for the definitions of each bit.

11.5.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see Figure 53).

The TDO pin is in low state during the start bit.

The TDO pin is in high state during the stop bit.

An Idle character is interpreted as an entire frame of "1"s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving "0"s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra "1" bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.



Figure 54. Word Length Programming

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 55. SCI Baud Rate and Extended Prescaler Block Diagram



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 53).

The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 53).

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

I²C BUS INTERFACE (Cont'd)

11.6.5 Low Power Modes

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

11.6.6 Interrupts

57/

Figure 60. Event Flags and Interrupt Generation



Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
10-bit Address Sent Event (Master mode)	ADD10		Yes	No
End of Byte Transfer Event	BTF]	Yes	No
Address Matched Event (Slave mode)	ADSEL		Yes	No
Start Bit Generation Event (Master mode)	SB	ITE	Yes	No
Acknowledge Failure Event	AF		Yes	No
Stop Detection Event (Slave mode)	STOPF		Yes	No
Arbitration Lost Event (Multimaster configuration)	ARLO		Yes	No
Bus Error Event	BERR		Yes	No

Note: The l^2C interrupt events are connected to the same interrupt vector (see Interrupts chapter). They generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

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

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 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



INSTRUCTION SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	11	Н	10	Ν	Ζ	С
ADC	Add with Carry	A=A+M+C	А	М		Н		Ν	Ζ	С
ADD	Addition	A = A + M	А	М		Н		Ν	Ζ	С
AND	Logical And	A = A . M	А	М				Ν	Ζ	
BCP	Bit compare A, Memory	tst (A . M)	А	М				Ν	Ζ	
BRES	Bit Reset	bres Byte, #3	М							
BSET	Bit Set	bset Byte, #3	М							
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	М							С
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	М							С
CALL	Call subroutine									
CALLR	Call subroutine relative									
CLR	Clear		reg, M					0	1	
СР	Arithmetic Compare	tst(Reg - M)	reg	М				Ν	Ζ	С
CPL	One Complement	A = FFH-A	reg, M					Ν	Ζ	1
DEC	Decrement	dec Y	reg, M					Ν	Ζ	
HALT	Halt				1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC			11	Н	10	Ν	Ζ	С
INC	Increment	inc X	reg, M					Ν	Ζ	
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	l1:0 = 11 ?								
JRNM	Jump if I1:0 <> 11	11:0 <> 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 >								



SUPPLY CURRENT CHARACTERISTICS (Cont'd)

13.4.4 On-chip peripherals

Symbol	Parameter	Conditions		Тур	Unit
I _{DD(TIM)}	16-bit Timer supply current ¹⁾	f _{CPU} =4MHz	V _{DD} =3.0V	200	
		f _{CPU} =8MHz	V _{DD} =5.0V	300	
1	SPI supply current ²⁾	f _{CPU} =4MHz	V _{DD} =3.0V	200	
DD(SPI)		f _{CPU} =8MHz	V _{DD} =5.0V	250	
I _{DD(SCI)}	SCI supply current ³⁾	f _{CPU} =4MHz	V _{DD} =3.0V	350	
		f _{CPU} =8MHz	V _{DD} =5.0V	650	μΑ
I _{DD(I2C)}	I2C supply current ⁴⁾	f _{CPU} =4MHz	V _{DD} =3.0V	350	
		f _{CPU} =8MHz	V _{DD} =5.0V	500	
I _{DD(ADC)}	ADC supply current when converting ⁵⁾	f 4MHz	V _{DD} =3.0V	500	
		ADC-4MINZ	V _{DD} =5.0V	600	

Notes:

1. Data based on a differential I_{DD} measurement between reset configuration (timer counter running at $f_{CPU}/2$) and timer counter stopped (only TIMD bit set). Data valid for one timer.

 Data based on a differential I_{DD} measurement between reset configuration (SPI disabled) and a permanent SPI master communication at maximum speed (data sent equal to FFh). This measurement includes the pad toggling consumption.

 Data based on a differential I_{DD} measurement between SCI running at maximum speed configuration (500 kbaud, continuous transmission of AA +RE enabled and SCI off. This measurement includes the pad toggling consumption.

4. Data based on a differential I_{DD} measurement between reset configuration (I2C disabled) and a permanent I2C master communication at 300kHz (data sent equal to AAh). This measurement includes the pad toggling consumption (4.7kOhm external pull-up on clock and data lines).

5. Data based on a differential I_{DD} measurement between reset configuration (ADC off) and continuous A/D conversion (f_{ADC}=4MHz).

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

	f _{OSC} (MHz)	Typical Ceramic Resonators				
Supplier		Reference ²⁾	Recommended OSCRNGE Option bit Configuration			
Murata	2	CSTCC2M00G56A-R0	MP Mode ³⁾			
	4	CSTCR4M00G55B-R0	MS Mode			
	8	CSTCE8M00G55A-R0	HS Mode			
	16	CSTCE16M0G53A-R0	HS Mode			

Notes:

<u>(</u>ح)

1. Resonator characteristics given by the ceramic resonator manufacturer.

SMD = [-R0: Plastic tape package (Ø =180mm), -B0: Bulk] LEAD = [-A0: Flat pack package (Radial taping Ho= 18mm), -B0: Bulk]

3. LP mode is not recommended for 2 MHz resonator because the peak to peak amplitude is too small (>0.8V) For more information on these resonators, please consult www.murata.com

DEVICE CONFIGURATION (Cont'd)

OPT 0 = **FMP_W** FLASH write protection This option indicates if the FLASH program memory is write protected.

Warning: When this option is selected, the program memory (and the option bit itself) can never be erased or programmed again.

- 0: Write protection off
- 1: Write protection on

USER OPTION BYTE 1

OPT 7 = **EXTIT** *Port C External Interrupt Configuration.*

This option bit allows the Port C external interrupt mapping to be configured as ei0 or ei1.

Table 27. External Interrupt Configuration

ei0	ei1	EXTIT option bit
PA[7:0] Ports	PB[7:0] Ports PC[5:0] Ports	1
PA[7:0] Ports PC[5:0] Ports	PB[7:0] Ports	0

OPT 6 = Reserved, must be kept at default value.

OPT 5:4 = **OSCTYPE[1:0]** Oscillator Type selection

These option bits select the Oscillator Type.

57

Clock Source	OSCTYPE1	OSCTYPE0
Resonator Oscillator	0	0
Reserved	0	1
Internal RC Oscillator	1	0
External Source	1	1

OPT 3:1 = **OSCRNGE[2:0]** Oscillator Range selection

These option bits select the oscillator range.

Тур.	Freq. Range	OSC RNGE2	OSC RNGE1	OSC RNGE0
VLP	32~100kHz	1	х	х
LP	1~2MHz	0	0	0
MP	2~4MHz	0	0	1
MS	4~8MHz	0	1	0
HS	8~16MHz	0	1	1

OPT 0 = PLL PLL selection

This option bit selects the PLL which allows multiplication by two of the oscillator frequency. The PLL must not be used with the internal RC oscillator. It is guaranteed only with a f_{OSC} input frequency between 2 and 4MHz.

0: PLL x2 enabled

1: PLL x2 disabled

CAUTION: the PLL can be enabled only if the "OSC RANGE" (OPT3:1) bits are configured to "MP - 2~4MHz". Otherwise, the device functionality is not guaranteed.

bit) accuracy does not meet the accuracy specified in the data sheet.

Workaround

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

16.2.11 Negative injection impact on ADC accuracy

Injecting a negative current on an analog input pins significantly reduces the accuracy of the AD Converter. Whenever necessary, the negative injection should be prevented by the addition of a Schottky diode between the concerned I/Os and ground. Injecting a negative current on digital input pins degrades ADC accuracy especially if performed on a pin close to ADC channel in use.

16.2.12 ADC conversion spurious results

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

Workaround

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