



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	I ² C, SCI, 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	A/D 6x10b
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-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f264g1m6

Table of Contents

10.1	I/O PORT INTERRUPT SENSITIVITY	45
10.2	I/O PORT ALTERNATE FUNCTIONS	45
10.3	MISCELLANEOUS REGISTER DESCRIPTION	46
11	ON-CHIP PERIPHERALS	48
11.1	WATCHDOG TIMER (WDG)	48
11.2	MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK (MCC/RTC)	53
11.3	16-BIT TIMER	55
11.4	SERIAL PERIPHERAL INTERFACE (SPI)	75
11.5	SERIAL COMMUNICATIONS INTERFACE (SCI)	87
11.6	I2C BUS INTERFACE (I2C)	103
11.7	10-BIT A/D CONVERTER (ADC)	116
12	INSTRUCTION SET	120
12.1	CPU ADDRESSING MODES	120
12.2	INSTRUCTION GROUPS	123
13	ELECTRICAL CHARACTERISTICS	126
13.1	PARAMETER CONDITIONS	126
13.2	ABSOLUTE MAXIMUM RATINGS	127
13.3	OPERATING CONDITIONS	128
13.4	SUPPLY CURRENT CHARACTERISTICS	131
13.5	CLOCK AND TIMING CHARACTERISTICS	135
13.6	MEMORY CHARACTERISTICS	140
13.7	EMC CHARACTERISTICS	141
13.8	I/O PORT PIN CHARACTERISTICS	144
13.9	CONTROL PIN CHARACTERISTICS	150
13.10	TIMER PERIPHERAL CHARACTERISTICS	152
13.11	COMMUNICATION INTERFACE CHARACTERISTICS	153
13.12	10-BIT ADC CHARACTERISTICS	157
14	PACKAGE CHARACTERISTICS	159
14.1	PACKAGE MECHANICAL DATA	159
14.2	THERMAL CHARACTERISTICS	160
14.3	LEAD-FREE PACKAGE INFORMATION	161
15	DEVICE CONFIGURATION AND ORDERING INFORMATION	162
15.1	OPTION BYTES	162
15.2	DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE	164
15.3	DEVELOPMENT TOOLS	166
16	KNOWN LIMITATIONS	168
16.1	ALL FLASH AND ROM DEVICES	168
16.2	FLASH DEVICES ONLY	168
17	REVISION HISTORY	171

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:

- $\overline{\text{RESET}}$: 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

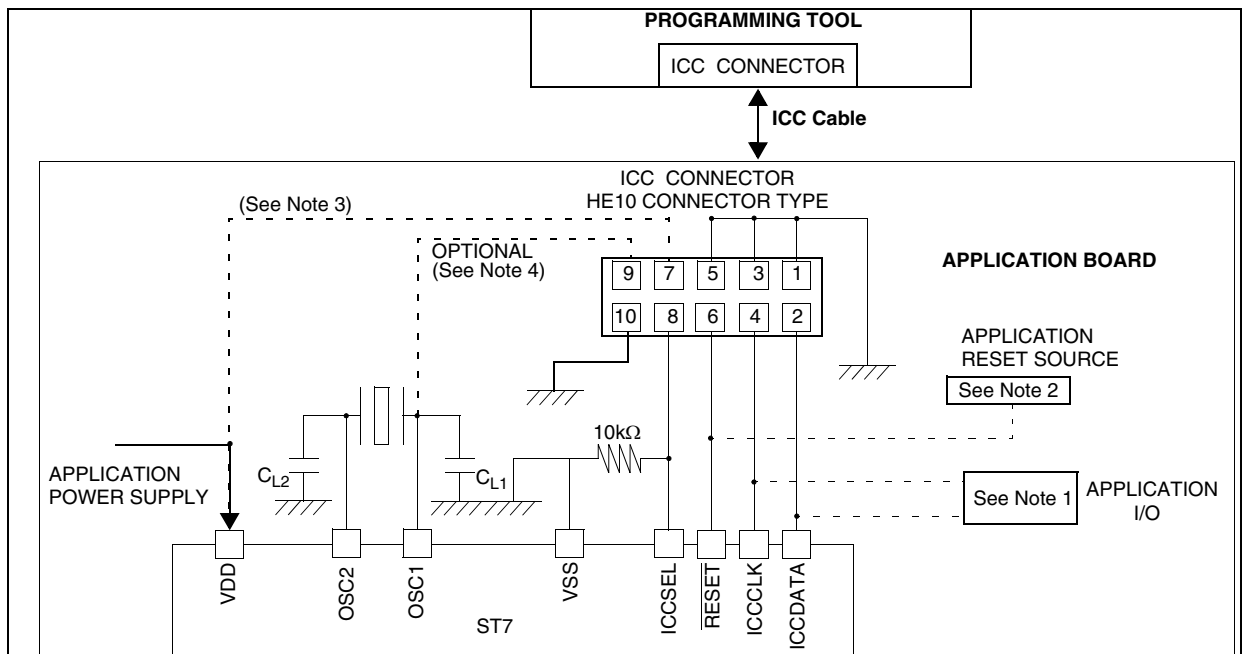
Tool documentation for recommended resistor values.

2. During the ICP session, the programming tool must control the $\overline{\text{RESET}}$ 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



INTERRUPTS (Cont'd)

7.3 INTERRUPTS AND LOW POWER MODES

All interrupts allow the processor to exit the WAIT low power mode. On the contrary, only external and other specified interrupts allow the processor to exit the HALT modes (see column “Exit from HALT” in “Interrupt Mapping” table). When several pending interrupts are present while exiting HALT mode, the first one serviced can only be an interrupt with exit from HALT mode capability and it is selected through the same decision process shown in Figure 17.

Note: If an interrupt, that is not able to Exit from HALT mode, is pending with the highest priority when exiting HALT mode, this interrupt is serviced after the first one serviced.

7.4 CONCURRENT & NESTED MANAGEMENT

The following Figure 18 and Figure 19 show two different interrupt management modes. The first is called concurrent mode and does not allow an interrupt to be interrupted, unlike the nested mode in Figure 19. The interrupt hardware priority is given in this order from the lowest to the highest: MAIN, IT4, IT3, IT2, IT1, IT0. The software priority is given for each interrupt.

Warning: A stack overflow may occur without notifying the software of the failure.

Note: TLI (Top Level Interrupt) is not available in this product.

Related Documentation

AN1044: Multiple interrupt source management for ST7 MCUs

Figure 18. Concurrent Interrupt Management

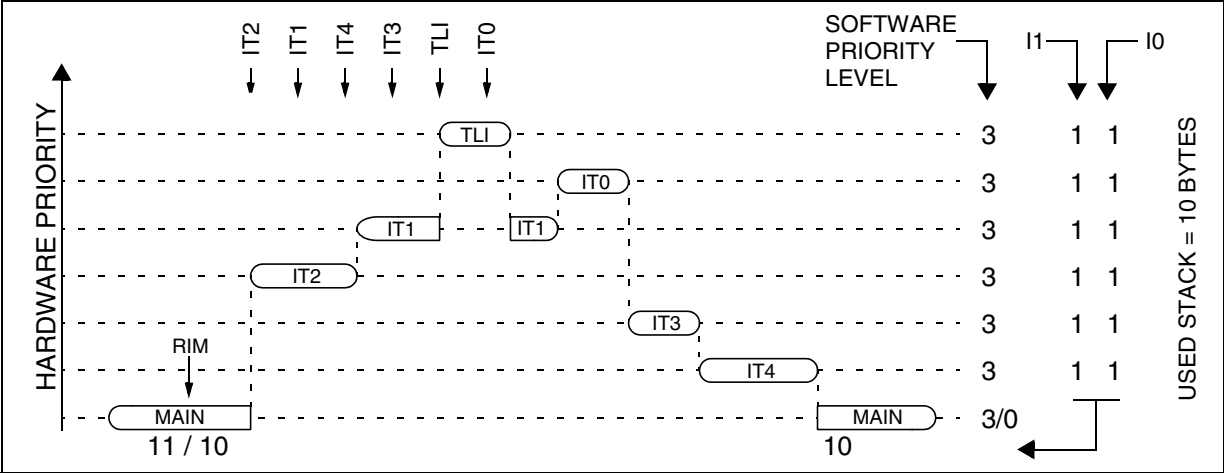
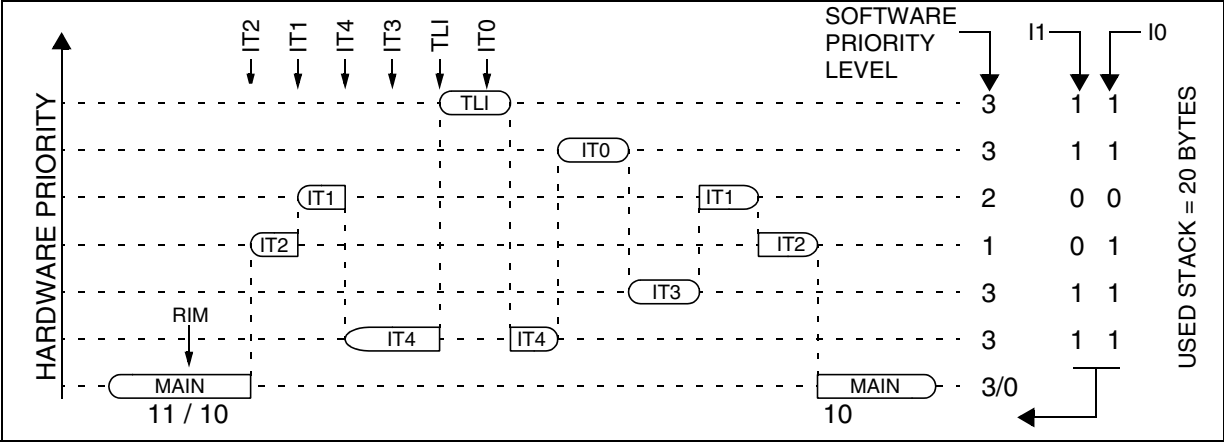


Figure 19. Nested Interrupt Management



POWER SAVING MODES (Cont'd)

8.5 HALT MODE

The HALT mode is the lowest power consumption mode of the MCU. It is entered by executing the ST7 HALT instruction (see Figure 26).

The MCU can exit HALT mode on reception of either a specific interrupt (see Table 5, "Interrupt Mapping," on page 32) or a RESET. When exiting HALT mode by means of a RESET or an interrupt, the oscillator is immediately turned on and the 4096 CPU cycle delay is used to stabilize the oscillator. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 25). When entering HALT mode, the I[1:0] bits in the CC register are forced to '10b' to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes immediately.

In the HALT mode the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

The compatibility of Watchdog operation with HALT mode is configured by the "WDGHALT" option bit of the option byte. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET (see Section 15.1 "OPTION BYTES" on page 162 for more details).

Figure 25. HALT Mode Timing Overview

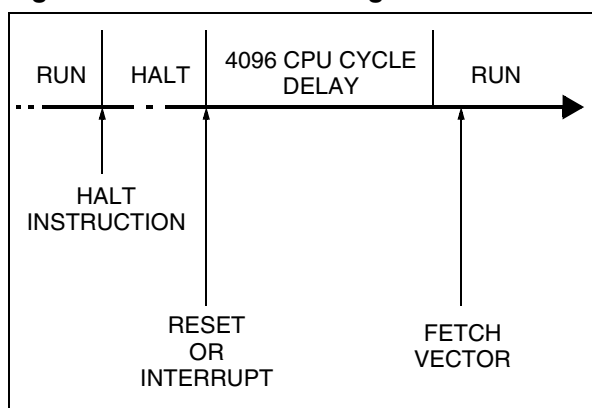
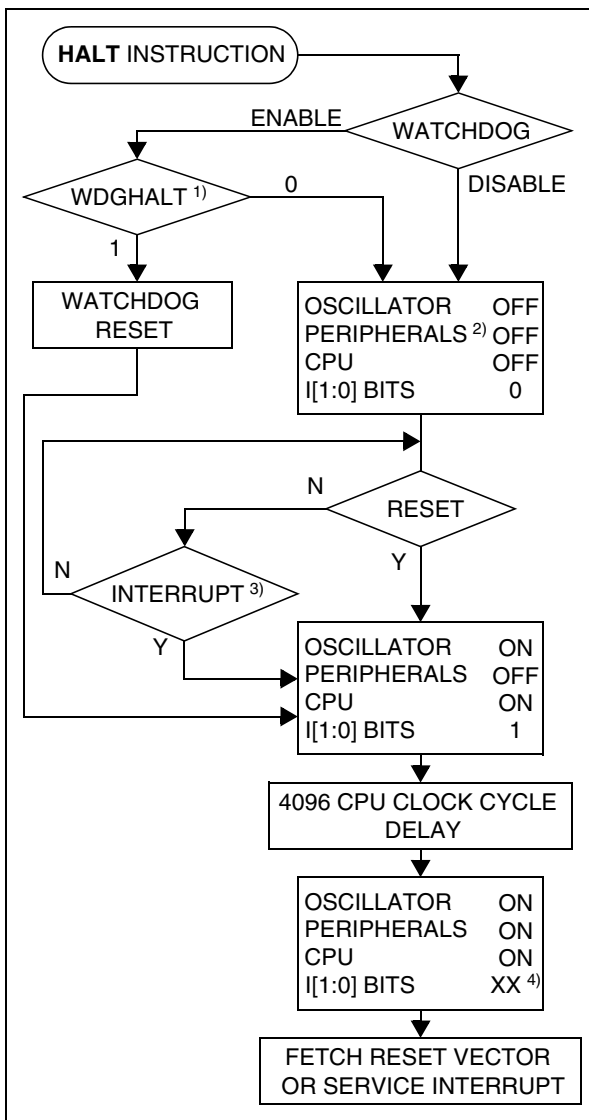


Figure 26. HALT Mode Flowchart



Notes:

1. WDGHALT is an option bit. See option byte section for more details.
2. Peripheral clocked with an external clock source can still be active.
3. Only some specific interrupts can exit the MCU from HALT mode (such as external interrupt). Refer to Table 5, "Interrupt Mapping," on page 32 for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits in the CC register are set during the interrupt routine and cleared when the CC register is popped.

11 ON-CHIP PERIPHERALS

11.1 WATCHDOG TIMER (WDG)

11.1.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

11.1.2 Main Features

- Programmable free-running downcounter
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero
- Optional reset on HALT instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte

11.1.3 Functional Description

The counter value stored in the Watchdog Control register (WDGCR bits T[6:0]), is decremented every $16384 f_{OSC2}$ cycles (approx.), and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

The application program must write in the WDGCR register at regular intervals during normal operation to prevent an MCU reset. This down-counter is free-running: it counts down even if the watchdog is disabled. The value to be stored in the WDGCR register must be between FFh and C0h:

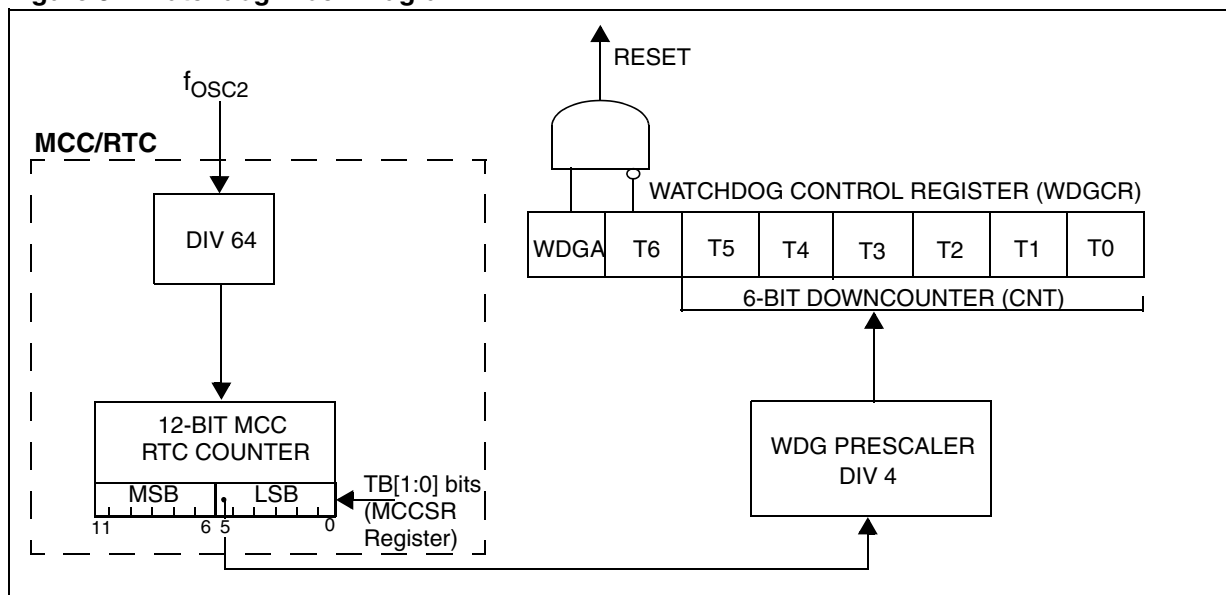
- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset (see [Figure 32. Approximate Timeout Duration](#)). The timing varies between a minimum and a maximum value due to the unknown status of the prescaler when writing to the WDGCR register (see [Figure 33](#)).

Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.

Figure 31. Watchdog Block Diagram



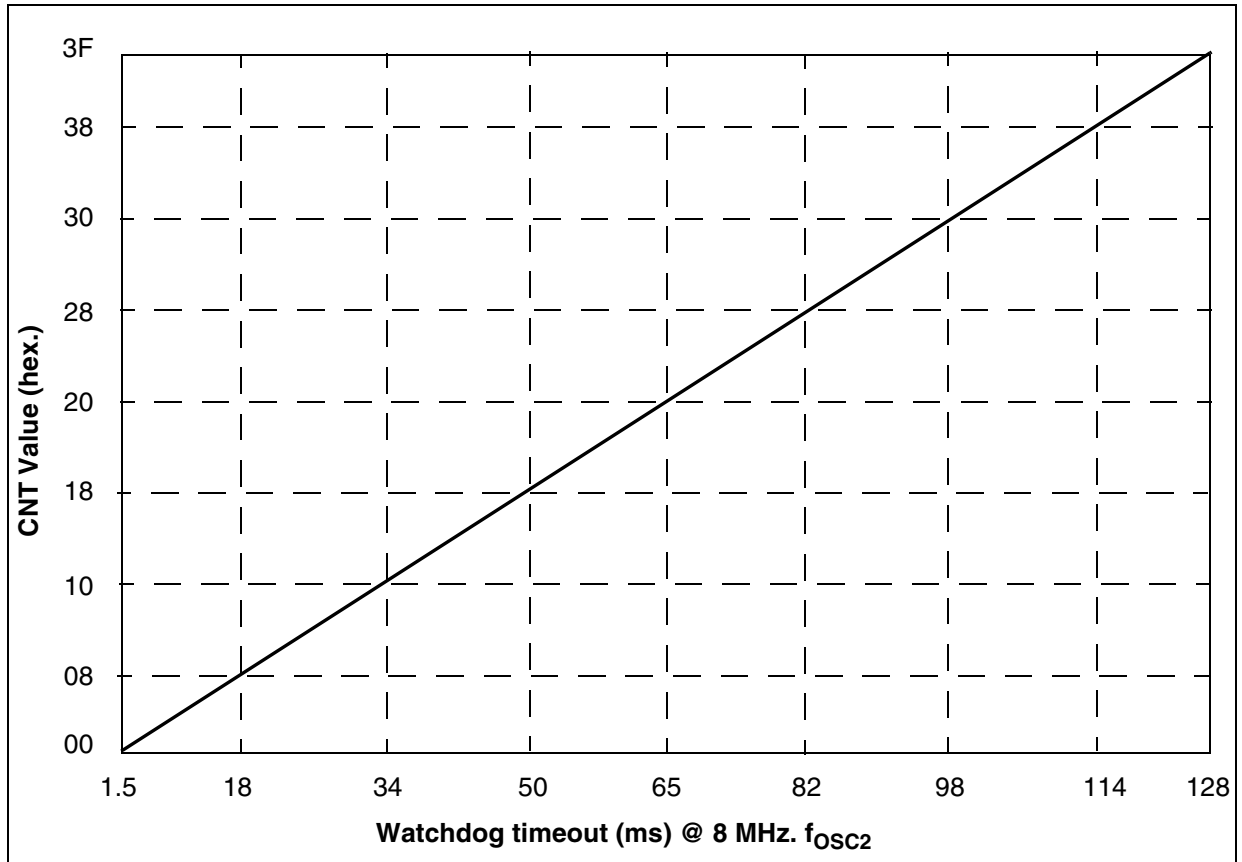
WATCHDOG TIMER (Cont'd)**11.1.4 How to Program the Watchdog Timeout**

Figure 32 shows the linear relationship between the 6-bit value to be loaded in the Watchdog Counter (CNT) and the resulting timeout duration in milliseconds. This can be used for a quick calculation without taking the timing variations into account. If

more precision is needed, use the formulae in Figure 33.

Caution: When writing to the WDGCR register, always write 1 in the T6 bit to avoid generating an immediate reset.

Figure 32. Approximate Timeout Duration



16-BIT TIMER (Cont'd)**11.3.3.5 One Pulse Mode**

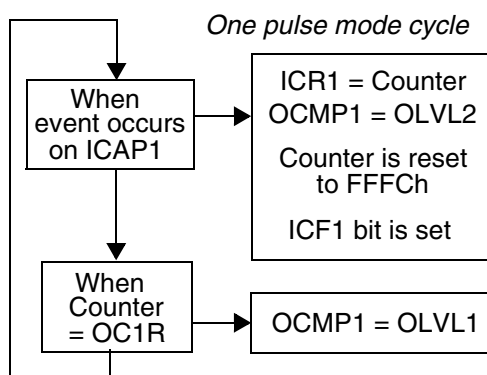
One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure:

To use one pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in the opposite column).
2. Select the following in the CR1 register:
 - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
 - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
 - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
3. Select the following in the CR2 register:
 - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
 - Set the OPM bit.
 - Select the timer clock CC[1:0] (see [Table 14 Clock Control Bits](#)).



When a valid event occurs on the ICAP1 pin, the counter value is loaded in the ICR1 register. The counter is then initialized to FFFCh, the OLVL2 bit is output on the OCMP1 pin and the ICF1 bit is set.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (i.e. clearing the ICF1 bit) is done in two steps:

1. Reading the SR register while the ICF1 bit is set.
2. An access (read or write) to the IC1LR register.

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

$$\text{OC1R Value} = \frac{t * f_{\text{CPU}}}{\text{PRESC}} - 5$$

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 14 Clock Control Bits](#))

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

$$\text{OC1R} = t * f_{\text{EXT}} - 5$$

Where:

t = Pulse period (in seconds)

f_{EXT} = External timer 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 44](#)).

Notes:

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 can not 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.

16-BIT TIMER (Cont'd)

11.3.3.6 Pulse Width Modulation Mode

Pulse Width Modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so this functionality can not be used when PWM mode is activated.

In PWM mode, double buffering is implemented on the output compare registers. Any new values written in the OC1R and OC2R registers are loaded in their respective shadow registers (double buffer) only at the end of the PWM period (OC2) to avoid spikes on the PWM output pin (OCMP1). The shadow registers contain the reference values for comparison in PWM “double buffering” mode.

Note: There is a locking mechanism for transferring the OCiR value to the buffer. After a write to the OCiHR register, transfer of the new compare value to the buffer is inhibited until OCiLR is also written.

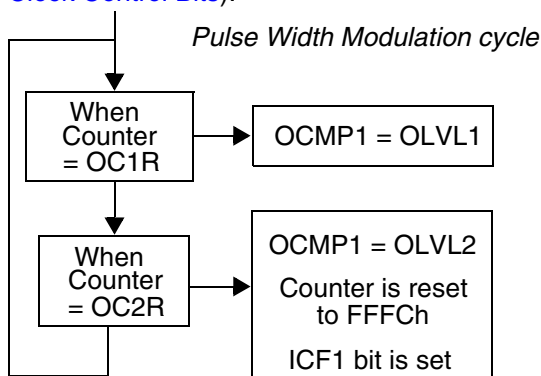
Unlike in Output Compare mode, the compare function is always enabled in PWM mode.

Procedure

To use pulse width modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal using the formula in the opposite column.
2. Load the OC1R register with the value corresponding to the period of the pulse if (OLVL1=0 and OLVL2=1) using the formula in the opposite column.
3. Select the following in the CR1 register:
 - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC1R register.
 - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC2R register.
4. Select the following in the CR2 register:
 - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
 - Set the PWM bit.
 - Select the timer clock (CC[1:0]) (see [Table 14](#)

Clock Control Bits).



If OLVL1=1 and OLVL2=0 the length of the positive pulse is the difference between the OC2R and OC1R registers.

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

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

$$\text{OCiR Value} = \frac{t * f_{\text{CPU}}}{\text{PRESC}} - 5$$

Where:

t = Signal or pulse period (in seconds)

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

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see [Table 14 Clock Control Bits](#))

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

$$\text{OCiR} = t * f_{\text{EXT}} - 5$$

Where:

t = Signal or pulse period (in seconds)

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

The Output Compare 2 event causes the counter to be initialized to FFFCh (See [Figure 45](#))

Notes:

1. The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.
2. The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.

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 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 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.

Bit 3, 2 = CC[1:0] Clock Control.

The timer clock mode depends on these bits:

Table 14. 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.

16-BIT TIMER (Cont'd)**CONTROL/STATUS REGISTER (CSR)**

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7			0				
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	0	0

Bit 7 = **ICF1** *Input Capture Flag 1*.

0: No input capture (reset value).

1: An input capture has occurred on the ICAP1 pin or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Bit 6 = **OCF1** *Output Compare Flag 1*.

0: No match (reset value).

1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.

Bit 5 = **TOF** *Timer Overflow Flag*.

0: No timer overflow (reset value).

1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

Note: Reading or writing the ACLR register does not clear TOF.

Bit 4 = **ICF2** *Input Capture Flag 2*.

0: No input capture (reset value).

1: An input capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 = **OCF2** *Output Compare Flag 2*.

0: No match (reset value).

1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2 = **TIMD** *Timer disable*.

This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed while it is disabled.

0: Timer enabled

1: Timer prescaler, counter and outputs disabled

Bits 1:0 = Reserved, must be kept cleared.

SERIAL PERIPHERAL INTERFACE (Cont'd)**11.4.5 Error Flags****11.4.5.1 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.

Notes: 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.

In a slave device, the MODF bit can not be set, but in a multi master configuration the Device can be in slave mode with the MODF bit set.

The MODF bit indicates that there might have been a multi-master conflict and allows software to handle this using an interrupt routine and either perform to a reset or return to an application default state.

11.4.5.2 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.

11.4.5.3 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 [Section 11.4.3.2 "Slave Select Management"](#) on page 78.

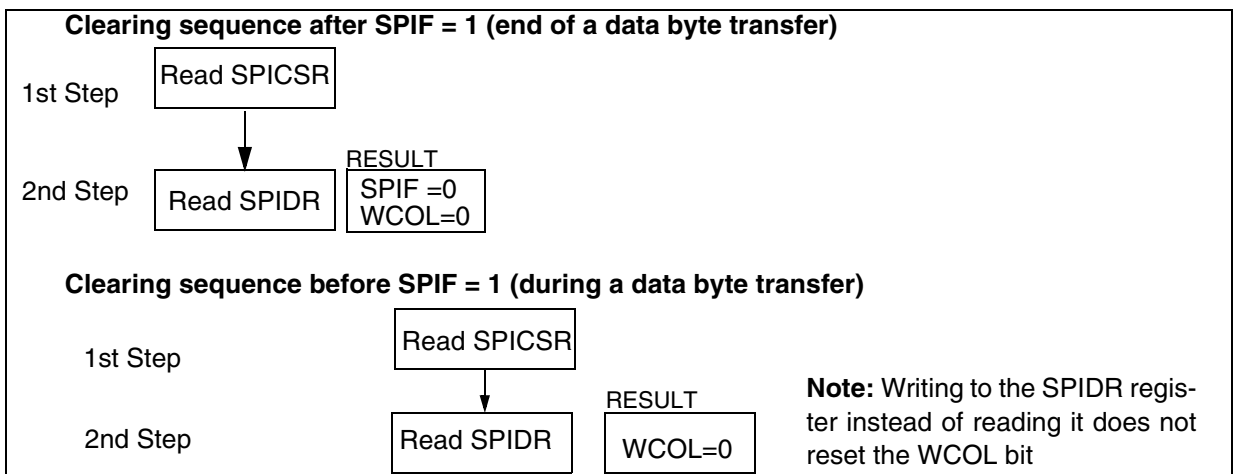
Note: a "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the CPU operation.

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).

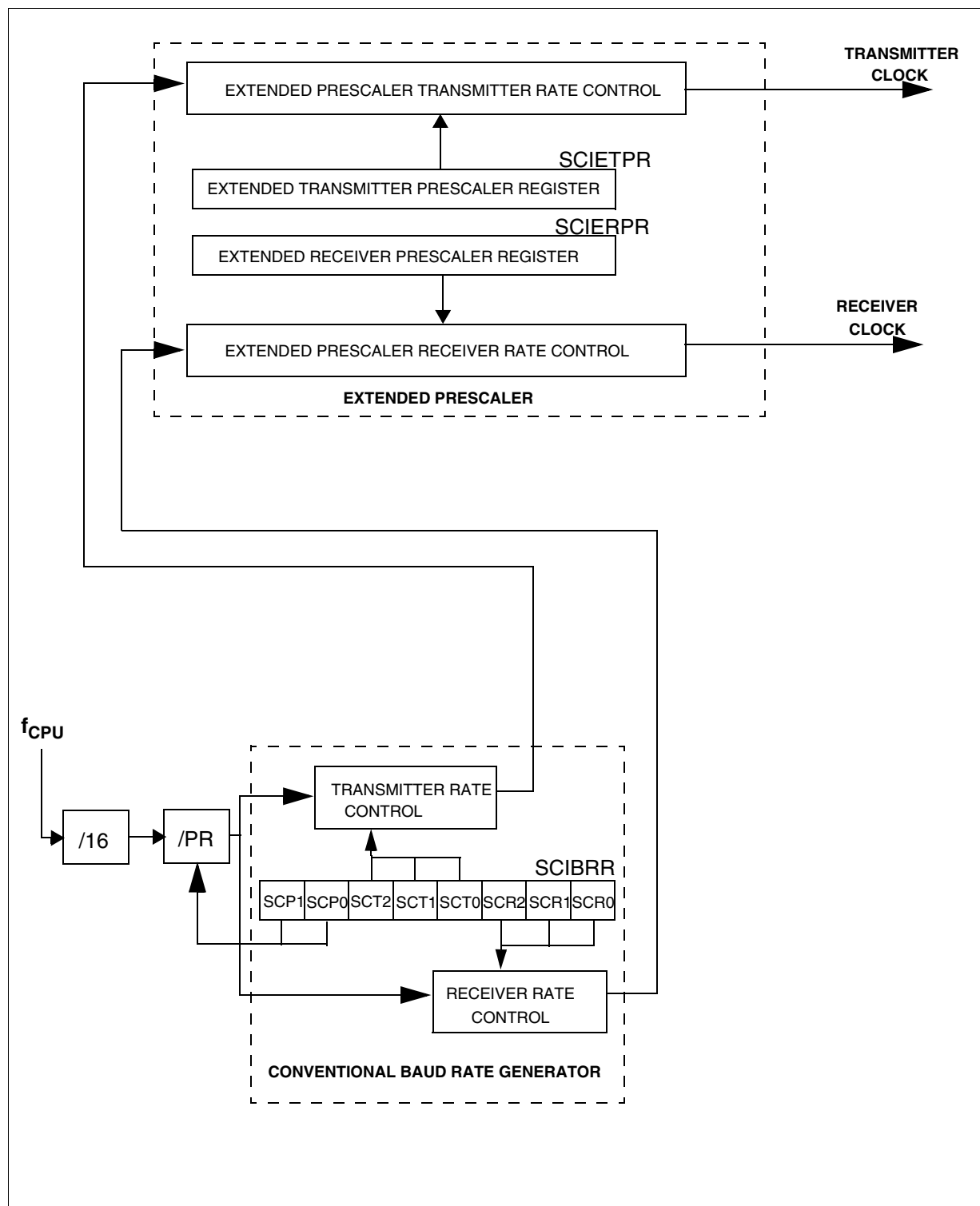
Clearing the WCOL bit is done through a software sequence (see [Figure 51](#)).

Figure 51. Clearing the WCOL bit (Write Collision Flag) Software Sequence



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 55. SCI Baud Rate and Extended Prescaler Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)**11.5.4.9 Clock Deviation Causes**

The causes which contribute to the total deviation are:

- D_{TRA} : Deviation due to transmitter error (Local oscillator error of the transmitter or the transmitter is transmitting at a different baud rate).
- D_{QUANT} : Error due to the baud rate quantisation of the receiver.
- D_{REC} : Deviation of the local oscillator of the receiver: This deviation can occur during the reception of one complete SCI message assuming that the deviation has been compensated at the beginning of the message.
- D_{TCL} : Deviation due to the transmission line (generally due to the transceivers)

All the deviations of the system should be added and compared to the SCI clock tolerance:

$$D_{TRA} + D_{QUANT} + D_{REC} + D_{TCL} < 3.75\%$$

11.5.4.10 Noise Error Causes

See also description of Noise error in [Section 11.5.4.3](#).

Start bit

The noise flag (NF) is set during start bit reception if one of the following conditions occurs:

1. A valid falling edge is not detected. A falling edge is considered to be valid if the 3 consecutive samples before the falling edge occurs are detected as '1' and, after the falling edge occurs, during the sampling of the 16 samples, if one of the samples numbered 3, 5 or 7 is detected as a "1".
2. During sampling of the 16 samples, if one of the samples numbered 8, 9 or 10 is detected as a "1".

Therefore, a valid Start Bit must satisfy both the above conditions to prevent the Noise Flag getting set.

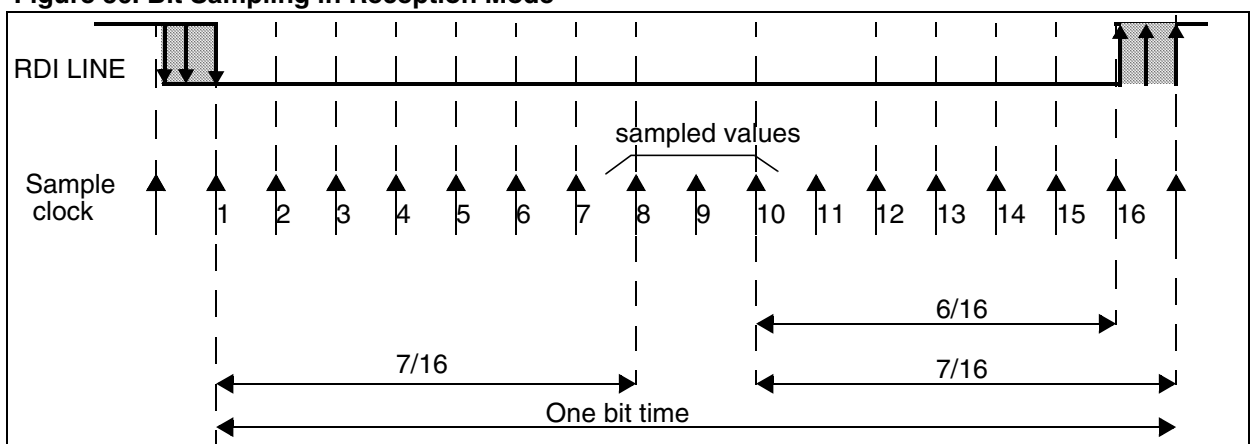
Data Bits

The noise flag (NF) is set during normal data bit reception if the following condition occurs:

- During the sampling of 16 samples, if all three samples numbered 8, 9 and 10 are not the same. The majority of the 8th, 9th and 10th samples is considered as the bit value.

Therefore, a valid Data Bit must have samples 8, 9 and 10 at the same value to prevent the Noise Flag getting set.

Figure 56. Bit Sampling in Reception Mode



SERIAL COMMUNICATIONS INTERFACE (Cont'd)**11.5.7 Register Description****STATUS REGISTER (SCISR)**

Read Only

Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE

Bit 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 will not be transferred to the shift register unless the TDRE bit is cleared.**Bit 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.**Bit 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

Bit 4 = IDLE *Idle line detect.*

This bit is set by hardware when an Idle Line is detected. An interrupt is generated if the ILIE=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: No Idle Line is detected

1: Idle Line is detected

Note: The IDLE bit will not be set again until the RDRF bit has been set itself (i.e. a new idle line occurs).**Bit 3 = OR** *Overrun error.*

This bit is set by hardware when the word currently being received in the shift register is ready to be transferred into the RDR register while RDRF=1. 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: No Overrun error

1: Overrun error is detected

Note: When this bit is set RDR register content will not be lost but the shift register will be overwritten.**Bit 2 = NF** *Noise flag.*

This bit is set by hardware when noise is detected on a received frame. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No noise is detected

1: Noise is detected

Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt.**Bit 1 = FE** *Framing error.*

This bit is set by hardware when a de-synchronization, excessive noise or a break character is detected. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No Framing error is detected

1: Framing error or break character is detected

Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt. If the word currently being transferred causes both frame error and overrun error, it will be transferred and only the OR bit will be set.**Bit 0 = PE** *Parity error.*

This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by a software sequence (a read to the status register followed by an access to the SCIDR data register). An interrupt is generated if PIE=1 in the SCICR1 register.

0: No parity error

1: Parity error

I²C BUS INTERFACE (Cont'd)**I²C CLOCK CONTROL REGISTER (CCR)**

Read / Write

Reset Value: 0000 0000 (00h)

7							0
FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0

Bit 7 = **FM/SM** *Fast/Standard I²C mode*.

This bit is set and cleared by software. It is not cleared when the interface is disabled (PE=0).

0: Standard I²C mode1: Fast I²C modeBit 6:0 = **CC[6:0]** *7-bit clock divider*.These bits select the speed of the bus (F_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE=0).

Refer to the Electrical Characteristics section for the table of values.

Note: The programmed F_{SCL} assumes no load on SCL and SDA lines.**I²C DATA REGISTER (DR)**

Read / Write

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[7:0]** *8-bit Data Register*.

These bits contain the byte to be received or transmitted on the bus.

- Transmitter mode: Byte transmission start automatically when the software writes in the DR register.
- Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address.
Then, the following data bytes are received one by one after reading the DR register.

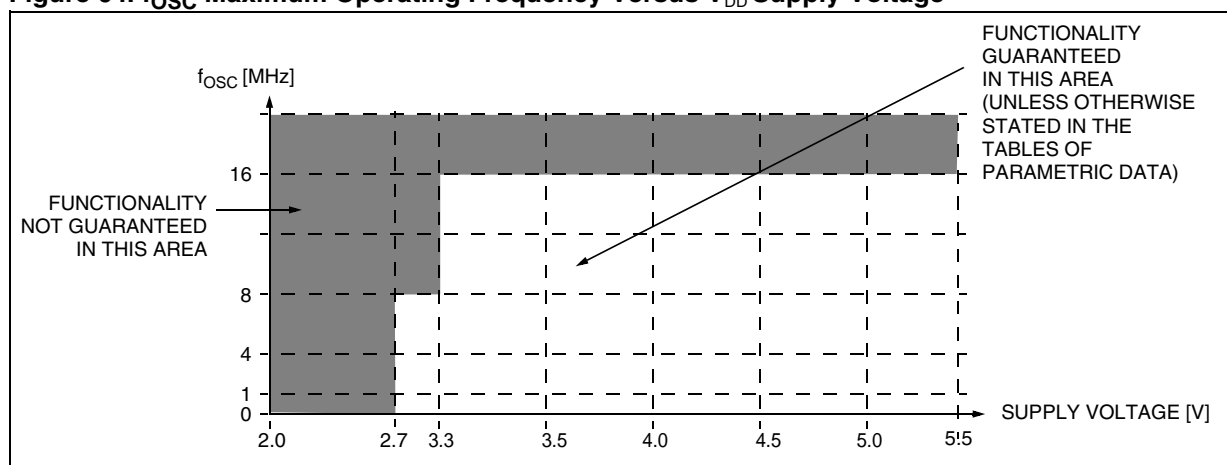
13.3 OPERATING CONDITIONS

13.3.1 General Operating Conditions

$T_A = -40$ to $+85^\circ\text{C}$ unless otherwise specified.

Symbol	Parameter	Conditions	Min	Max	Unit
V _{DD}	Supply voltage	f _{OSC} = 8 MHz. max.	2.7	5.5	V
		f _{OSC} = 16 MHz. max.	3.3	5.5	
f _{OSC}	External clock frequency on OSC1 pin	V _{DD} ≥3.3V	up to 16		MHz
		V _{DD} ≥2.7V	up to 8		

Figure 64. f_{OSC} Maximum Operating Frequency Versus V_{DD} Supply Voltage



SUPPLY CURRENT CHARACTERISTICS (Cont'd)

Figure 66. Typical I_{DD} in RUN at $T_A=25^{\circ}\text{C}$

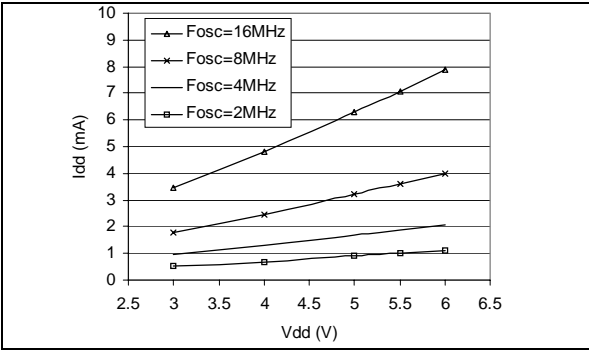


Figure 68. Typical I_{DD} in WAIT at $T_A=25^{\circ}\text{C}$

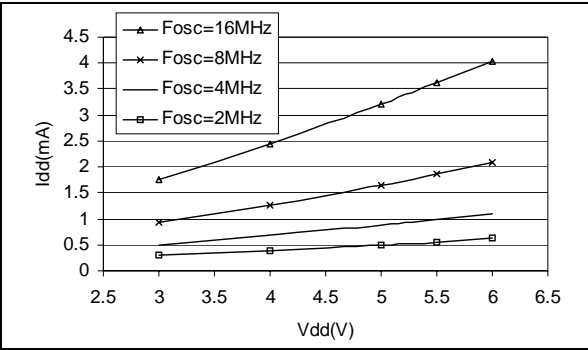


Figure 67. Typical I_{DD} in SLOW at $T_A=25^{\circ}\text{C}$

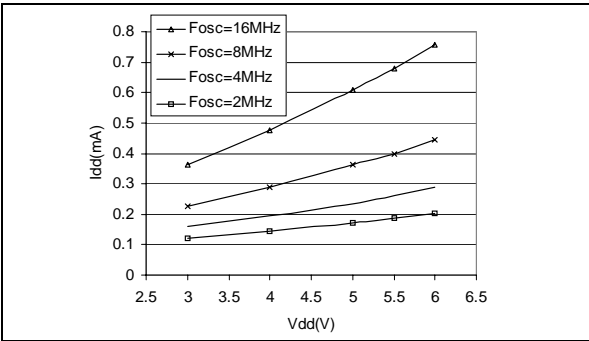
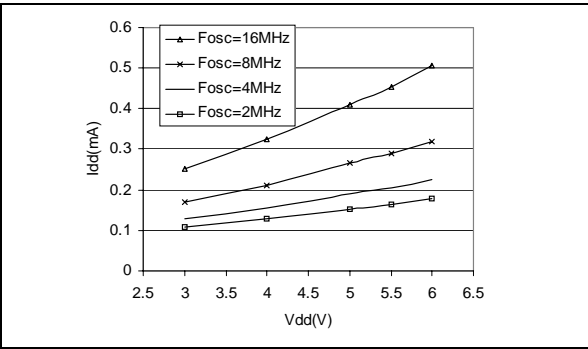


Figure 69. Typ. I_{DD} in SLOW-WAIT at $T_A=25^{\circ}\text{C}$



13.5 CLOCK AND TIMING CHARACTERISTICS

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

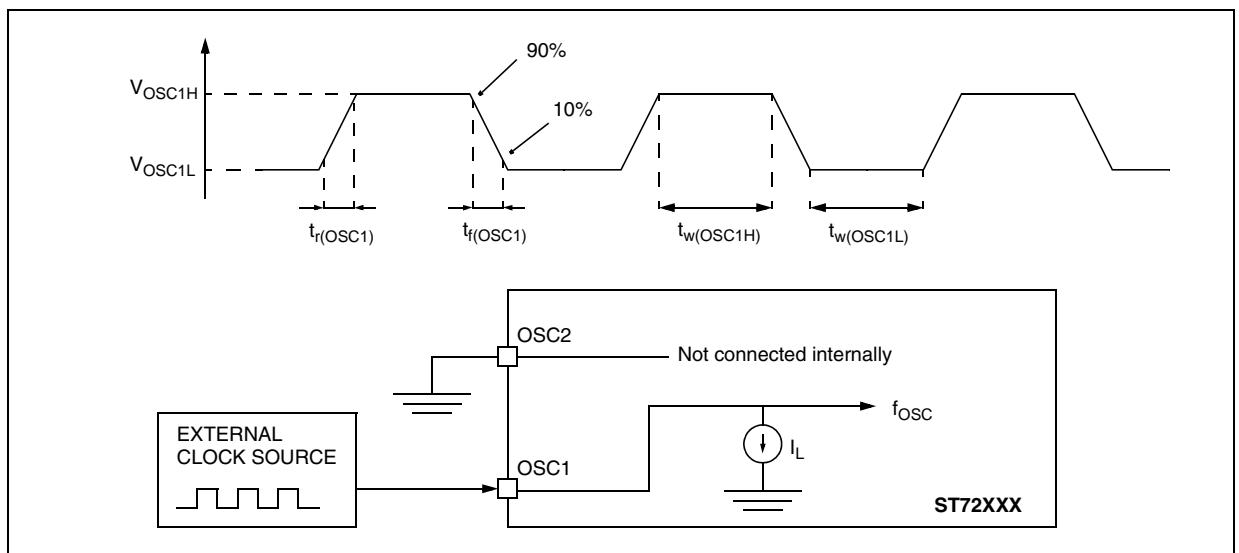
13.5.1 General Timings

Symbol	Parameter	Conditions	Min	Typ ¹⁾	Max	Unit
$t_{c(INST)}$	Instruction cycle time		2	3	12	t_{CPU}
		$f_{CPU}=8\text{MHz}$	250	375	1500	ns
$t_{V(IT)}$	Interrupt reaction time ²⁾ $t_{V(IT)} = \Delta t_{c(INST)} + 10$		10		22	t_{CPU}
		$f_{CPU}=8\text{MHz}$	1.25		2.75	μs

13.5.2 External Clock Source

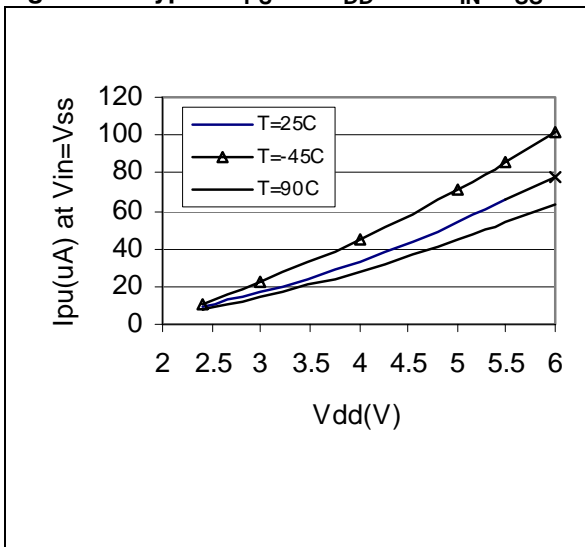
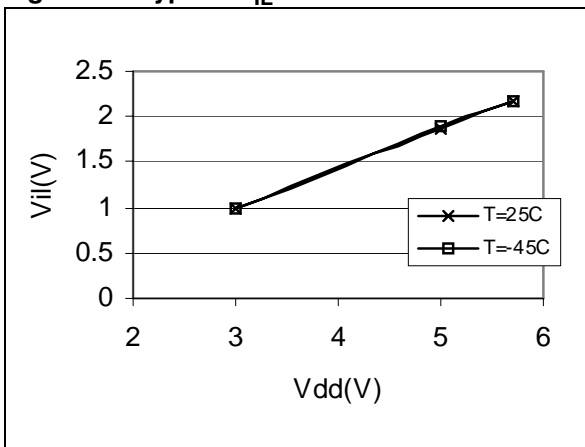
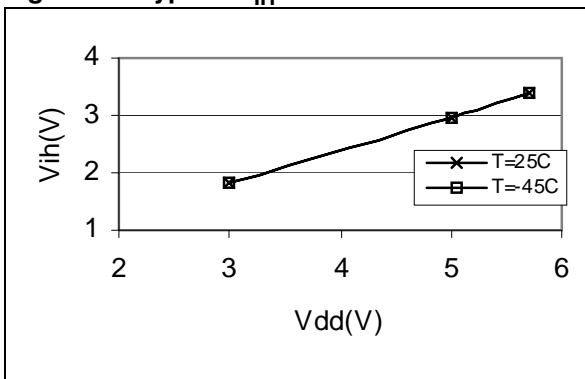
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSC1H}	OSC1 input pin high level voltage	see Figure 70	$0.7 \times V_{DD}$		V_{DD}	V
V_{OSC1L}	OSC1 input pin low level voltage		V_{SS}		$0.3 \times V_{DD}$	
$t_w(OSC1H)$ $t_w(OSC1L)$	OSC1 high or low time ³⁾		15			ns
$t_r(OSC1)$ $t_f(OSC1)$	OSC1 rise or fall time ³⁾				15	
I_L	OSCx Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			± 1	μA

Figure 70. Typical Application with an External Clock Source



Notes:

1. Data based on typical application software.
2. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.
3. Data based on design simulation and/or technology characteristics, not tested in production.

I/O PORT PIN CHARACTERISTICS (Cont'd)**Figure 77. Typical I_{PU} vs. V_{DD} with $V_{IN}=V_{SS}$** **Figure 78. Typical V_{IL}** **Figure 79. Typical V_{IH}** 

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/FASTROM).

ST7226x devices are ROM versions. ST72P26x devices are Factory Advanced Service Technique ROM (FASTROM) versions: they are factory-programmed XFlash devices.

ST72F26x XFlash devices are shipped to customers with a default program memory content (FFh). The option bytes are programmed to enable the internal RC oscillator. The ROM/FASTROM 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/FASTROM 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

OPT 7 = **WDG HALT** *Watchdog reset on HALT*

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

OPT 6 = **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)

OPT 5:4 = **VD[1:0]** *Voltage detection selection*

These option bits enable the voltage detection block (LVD and AVD) with a selected threshold of the LVD and AVD.

Configuration	VD1	VD0
LVD Off	1	1
Lowest Voltage Threshold (~3.05V)	1	0
Medium Voltage Threshold (~3.6V)	0	1
Highest Voltage Threshold (~4.1V)	0	0

OPT 3:2 = **SEC[1:0]** *Sector 0 size definition*

These option bits indicate the size of sector 0 according to the following table.

Sector 0 Size	SEC1	SEC0
0.5k	0	0
1k	0	1
2	1	0
4k ¹⁾	1	1

Note 1: 4k available on FASTROM devices only.

OPT 1 = **FMP_R** *Read-out protection*

Read-out protection, when selected, provides a protection against Program Memory content extraction and against write access to Flash memory.

Erasing the option bytes when the FMP_R option is selected will cause the whole memory to be erased first, and the device can be reprogrammed. Refer to [Section 4.5](#) and the ST7 Flash Programming Reference Manual for more details.

0: Read-out protection off

1: Read-out protection on

	USER OPTION BYTE 0								USER OPTION BYTE 1							
	7							0	7							0
	WDG HALT	WDG SW	VD1	VD0	SEC 1	SEC 0	FMP R	FMP W	EXTIT	Res.	OSC TYPE 1	OSC TYPE 0	OSC RNGE 2	OSC RNGE 1	OSC RNGE 0	PLL OFF
Default Value	1	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1