Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbusSCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f561ar9tae

INTERRUPTS (Cont'd)

7.6 EXTERNAL INTERRUPTS

7.6.1 I/O Port Interrupt Sensitivity

The external interrupt sensitivity is controlled by the ISxx bits in the EICR register (Figure 21). This control allows up to four fully independent external interrupt source sensitivities.

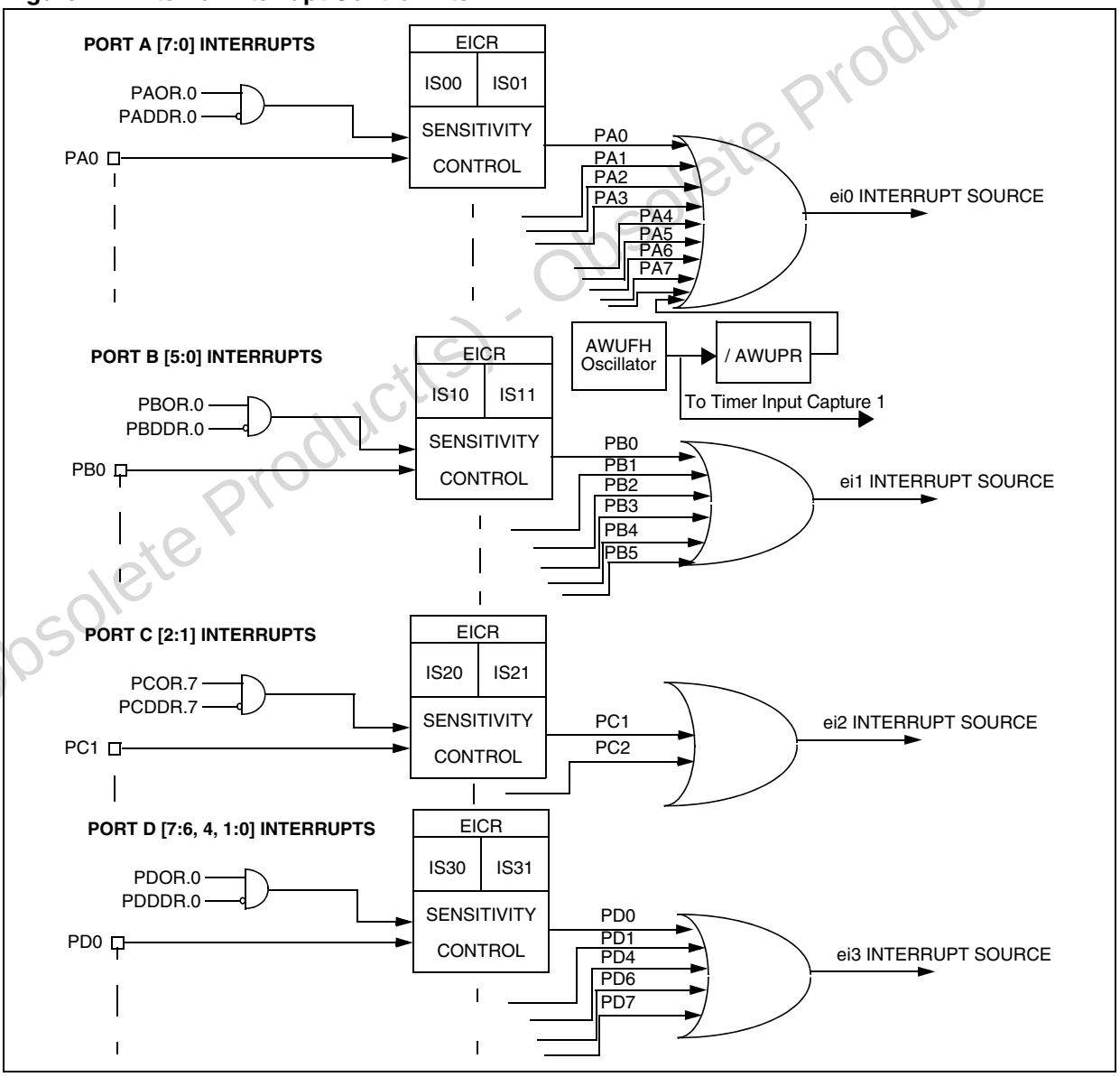
Each external interrupt source can be generated on four (or five) different events on the pin:

- Falling edge
- Rising edge
- Falling and rising edge
- Falling edge and low level

To guarantee correct functionality, the sensitivity bits in the EICR register can be modified only when the I1 and I0 bits of the CC register are both set to 1 (level 3). This means that interrupts must be disabled before changing sensitivity.

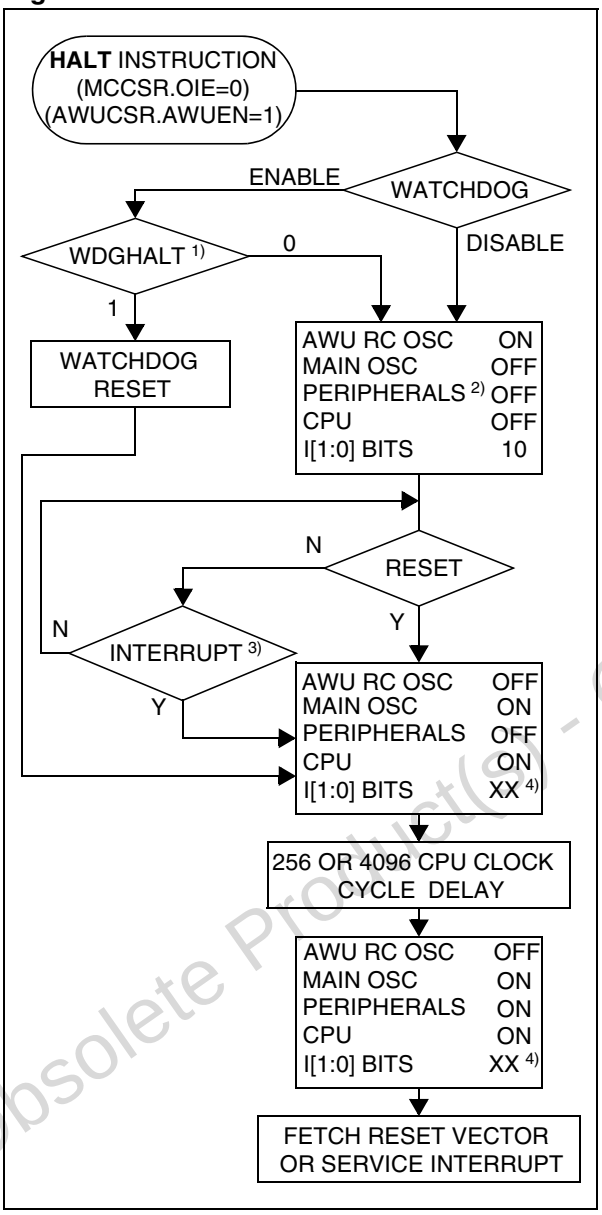
The pending interrupts are cleared by writing a different value in the ISx[1:0] of the EICR.

Figure 21. External Interrupt Control Bits



POWER SAVING MODES (Cont'd)

Figure 31. AWUFH Mode Flow-chart



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 an AWUFH interrupt and some specific interrupts can exit the MCU from HALT mode (such as external interrupt). Refer to Table 9, "Interrupt Mapping," on page 34 for more details.

4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

10 ON-CHIP PERIPHERALS

10.1 WINDOW WATCHDOG (WWDG)

10.1.1 Introduction

The Window Watchdog 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 contents of the downcounter before the T6 bit becomes cleared. An MCU reset is also generated if the 7-bit downcounter value (in the control register) is refreshed before the downcounter has reached the window register value. This implies that the counter must be refreshed in a limited window.

10.1.2 Main Features

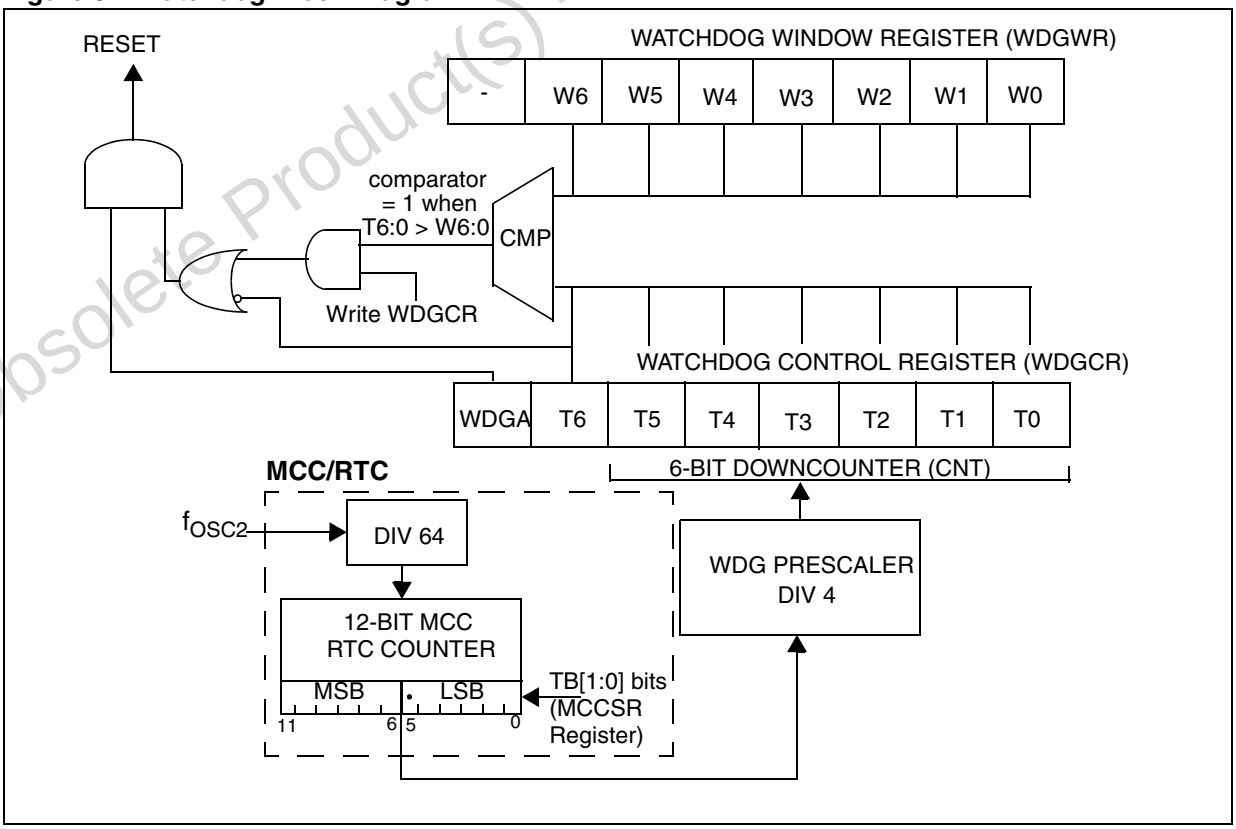
- Programmable free-running downcounter
- Conditional reset
  - Reset (if watchdog activated) when the downcounter value becomes less than 40h
  - Reset (if watchdog activated) if the downcounter is reloaded outside the window (see Figure 4)
- Hardware/Software Watchdog activation (selectable by option byte)
- Optional reset on HALT instruction (configurable by option byte)

10.1.3 Functional Description

The counter value stored in the WDGCR register (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 downcounter (T[6:0] bits) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 30µs. If the software reloads the counter while the counter is greater than the value stored in the window register, then a reset is generated.

Figure 34. Watchdog Block Diagram



ON-CHIP PERIPHERALS (Cont'd)

10.3.3 Register Description

CONTROL / STATUS REGISTER (ARTCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
EXCL	CC2	CC1	CC0	TCE	FCRL	OIE	OVF

Bit 7 = **EXCL** *External Clock*
This bit is set and cleared by software. It selects the input clock for the 7-bit prescaler.
0: CPU clock.
1: External clock.

Bit 6:4 = **CC[2:0]** *Counter Clock Control*
These bits are set and cleared by software. They determine the prescaler division ratio from f_{INPUT}.

$f_{COUNTER}$	With f_{INPUT}=8 MHz	CC2	CC1	CC0
f_{INPUT}	8 MHz	0	0	0
f_{INPUT} / 2	4 MHz	0	0	1
f_{INPUT} / 4	2 MHz	0	1	0
f_{INPUT} / 8	1 MHz	0	1	1
f_{INPUT} / 16	500 kHz	1	0	0
f_{INPUT} / 32	250 kHz	1	0	1
f_{INPUT} / 64	125 kHz	1	1	0
f_{INPUT} / 128	62.5 kHz	1	1	1

Bit 3 = **TCE** *Timer Counter Enable*
This bit is set and cleared by software. It puts the timer in the lowest power consumption mode.
0: Counter stopped (prescaler and counter frozen).
1: Counter running.

Bit 2 = **FCRL** *Force Counter Re-Load*
This bit is write-only and any attempt to read it will yield a logical zero. When set, it causes the contents of ARTARR register to be loaded into the counter, and the content of the prescaler register to be cleared in order to initialize the timer before starting to count.

Bit 1 = **OIE** *Overflow Interrupt Enable*
This bit is set and cleared by software. It allows to enable/disable the interrupt which is generated when the OVF bit is set.
0: Overflow Interrupt disable.
1: Overflow Interrupt enable.

Bit 0 = **OVF** *Overflow Flag*
This bit is set by hardware and cleared by software reading the ARTCSR register. It indicates the transition of the counter from FFh to the ARTARR value.
0: New transition not yet reached
1: Transition reached

COUNTER ACCESS REGISTER (ARTCAR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0

Bit 7:0 = **CA[7:0]** *Counter Access Data*

These bits can be set and cleared either by hardware or by software. The ARTCAR register is used to read or write the auto-reload counter "on the fly" (while it is counting).

AUTO-RELOAD REGISTER (ARTARR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bit 7:0 = **AR[7:0]** *Counter Auto-Reload Data*

These bits are set and cleared by software. They are used to hold the auto-reload value which is automatically loaded in the counter when an overflow occurs. At the same time, the PWM output levels are changed according to the corresponding OPx bit in the PWMCR register.

This register has two PWM management functions:

– Adjusting the PWM frequency
– Setting the PWM duty cycle resolution

PWM Frequency vs Resolution:

ARTARR value	Resolution	f_{PWM}	
		Min	Max
0	8-bit	~0.244 kHz	31.25 kHz
[0..127]	> 7-bit	~0.244 kHz	62.5 kHz
[128..191]	> 6-bit	~0.488 kHz	125 kHz
[192..223]	> 5-bit	~0.977 kHz	250 kHz
[224..239]	> 4-bit	~1.953 kHz	500 kHz

16-BIT TIMER (Cont'd)

10.4.7 Register Description

Each Timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 7 = **ICIE** *Input Capture Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.

Bit 6 = **OCIE** *Output Compare Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.

Bit 5 = **TOIE** *Timer Overflow Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Bit 4 = **FOLV2** *Forced Output Compare 2.*
This bit is set and cleared by software.
0: No effect on the OCMP2 pin.
1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison.

Bit 3 = **FOLV1** *Forced Output Compare 1.*
This bit is set and cleared by software.
0: No effect on the OCMP1 pin.
1: Forces OLVL1 to be copied to the OCMP1 pin, if the OC1E bit is set and even if there is no successful comparison.

Bit 2 = **OLVL2** *Output Level 2.*
This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse mode and Pulse Width Modulation mode.

Bit 1 = **IEDG1** *Input Edge 1.*
This bit determines which type of level transition on the ICAP1 pin will trigger the capture.
0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **OLVL1** *Output Level 1.*
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

8-BIT TIMER (Cont'd)

Figure 63. Input Capture Block Diagram

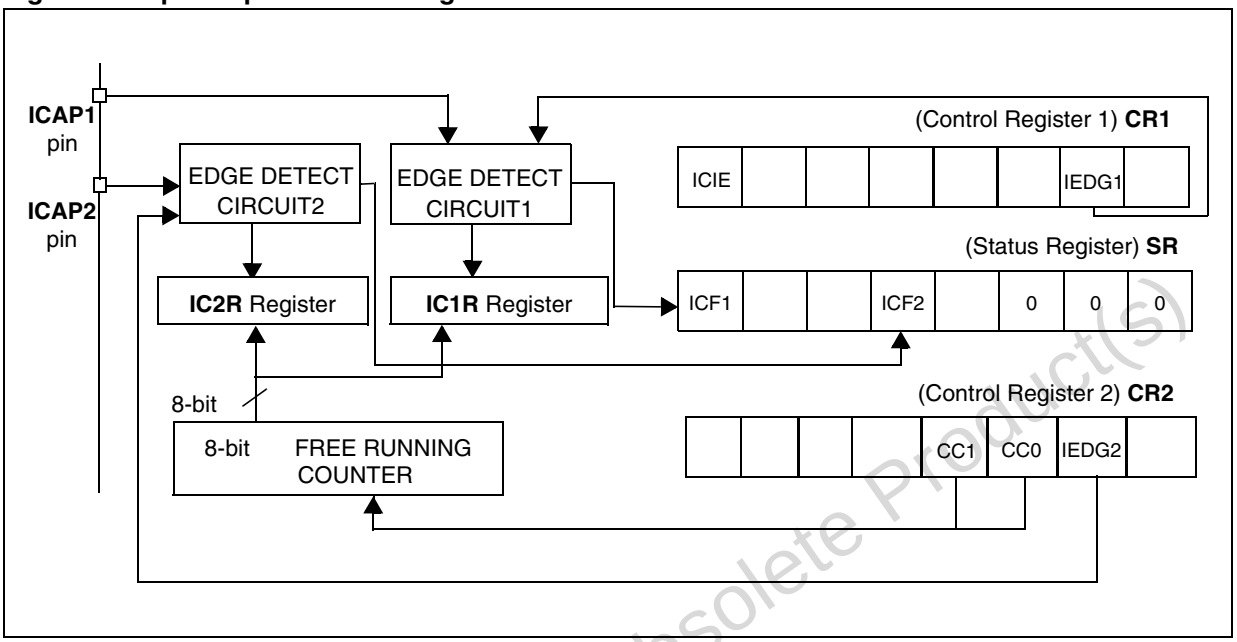
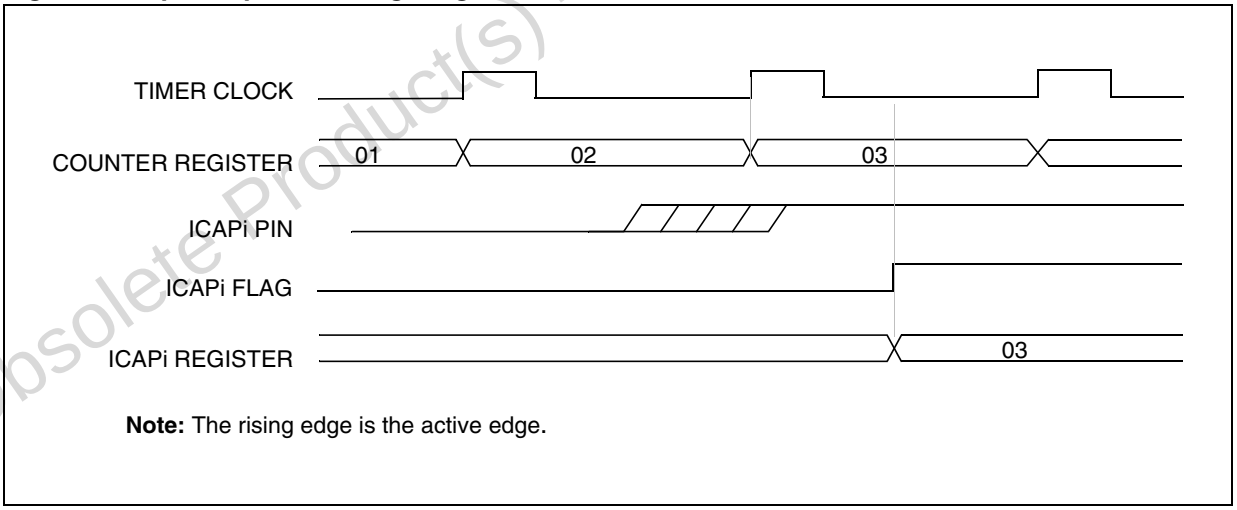


Figure 64. Input Capture Timing Diagram



Note: The rising edge is the active edge.

SERIAL PERIPHERAL INTERFACE (cont'd)

10.6.3.1 Functional Description

A basic example of interconnections between a single master and a single slave is illustrated in Figure 71.

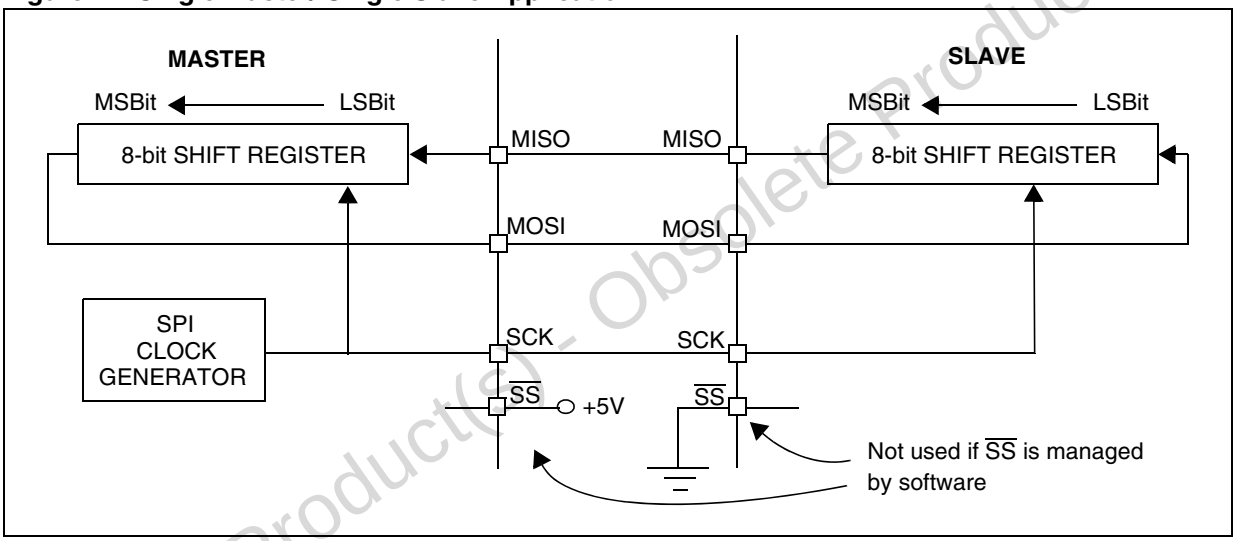
The MOSI pins are connected together and the MISO pins are connected together. In this way data is transferred serially between master and slave (most significant bit first).

The communication is always initiated by the master. 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 communication with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

To use a single data line, the MISO and MOSI pins must be connected at each node (in this case only simplex communication is possible).

Four possible data/clock timing relationships may be chosen (see Figure 74 on page 114) but master and slave must be programmed with the same timing mode.

Figure 71. Single Master/ Single Slave Application



SERIAL PERIPHERAL INTERFACE (cont'd)

SPI CONTROL/STATUS REGISTER (SPICSR)

Read/Write (some bits Read Only)
Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF** *Serial Peripheral Data Transfer Flag (Read only)*
This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register).
0: Data transfer is in progress or the flag has been cleared.
1: Data transfer between the device and an external device has been completed.

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Bit 6 = **WCOL** *Write Collision status (Read only)*
This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 75).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = **OVR** S*PI Overrun error (Read only)*
This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 10.6.5.2). An interrupt is generated if SPIE = 1 in the SPICR register. The OVR bit is cleared by software reading the SPICSR register.
0: No overrun error
1: Overrun error detected

Bit 4 = **MODF** *Mode Fault flag (Read only)*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 10.6.5.1 "Master Mode Fault (MODF)"). An SPI interrupt can be generated if SPIE = 1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF = 1 followed by a write to the SPICR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bit 3 = Reserved, must be kept cleared.

Bit 2 = **SOD** *SPI Output Disable*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode)
0: SPI output enabled (if SPE = 1)
1: SPI output disabled

Bit 1 = **SSM** $\overline{SS}$ *Management*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI $\overline{SS}$ pin and uses the SSI bit value instead. See Section 10.6.3.2 "Slave Select Management".
0: Hardware management ($\overline{SS}$ managed by external pin)
1: Software management (internal $\overline{SS}$ signal controlled by SSI bit. External $\overline{SS}$ pin free for general-purpose I/O)

Bit 0 = **SSI** $\overline{SS}$ *Internal Mode*
This bit is set and cleared by software. It acts as a 'chip select' by controlling the level of the $\overline{SS}$ slave select signal when the SSM bit is set.
0: Slave selected
1: Slave deselected

SPI DATA I/O REGISTER (SPIDR)

Read/Write
Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

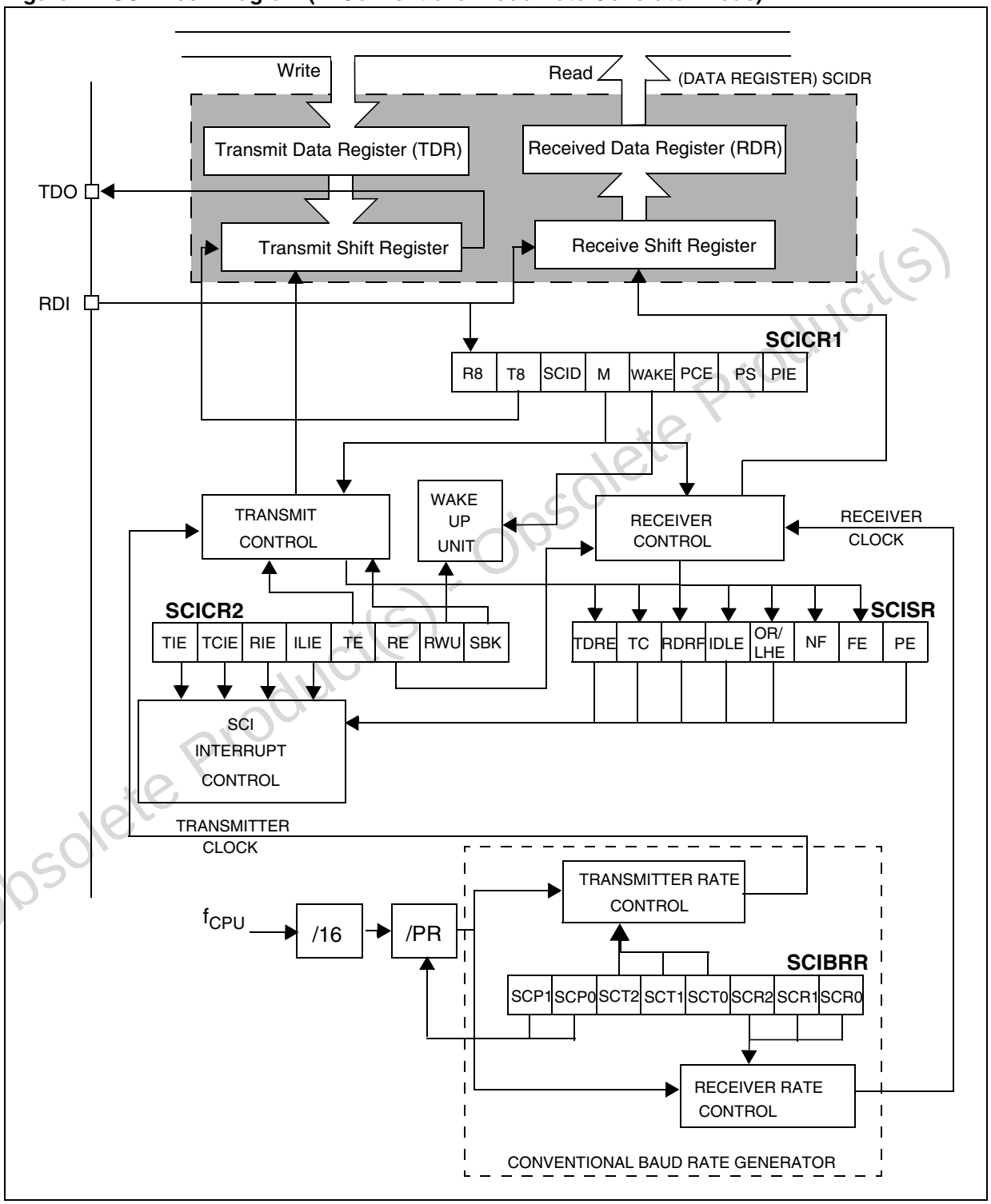
While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see Figure 70).

LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)

Figure 77. SCI Block Diagram (in Conventional Baud Rate Generator Mode)



LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode)

10.7.9 LIN Mode - Functional Description.

The block diagram of the Serial Control Interface, in LIN slave mode is shown in Figure 5.

It uses six registers:

– 3 control registers: SCICR1, SCICR2 and SCICR3

– 2 status registers: the SCISR register and the LHLR register mapped at the SCIERPR address

– A baud rate register: LPR mapped at the SCIBRR address and an associated fraction register LPFR mapped at the SCIETPR address

The bits dedicated to LIN are located in the SCICR3. Refer to the register descriptions in Section 0.1.10 for the definitions of each bit.

10.7.9.1 Entering LIN Mode

To use the LINSCI in LIN mode the following configuration must be set in SCICR3 register:

– Clear the M bit to configure 8-bit word length.

– Set the LINE bit.

Master

To enter master mode the LSLV bit must be reset In this case, setting the SBK bit will send 13 low bits.

Then the baud rate can programmed using the SCIBRR, SCIERPR and SCIETPR registers.

In LIN master mode, the Conventional and / or Extended Prescaler define the baud rate (as in standard SCI mode)

Slave

Set the LSLV bit in the SCICR3 register to enter LIN slave mode. In this case, setting the SBK bit will have no effect.

In LIN Slave mode the LIN baud rate generator is selected instead of the Conventional or Extended Prescaler. The LIN baud rate generator is common to the transmitter and the receiver.

Then the baud rate can be programmed using LPR and LPRF registers.

Note: It is mandatory to set the LIN configuration first before programming LPR and LPRF, because the LIN configuration uses a different baud rate generator from the standard one.

10.7.9.2 LIN Transmission

In LIN mode the same procedure as in SCI mode has to be applied for a LIN transmission.

To transmit the LIN Header the proceed as follows:

– First set the SBK bit in the SCICR2 register to start transmitting a 13-bit LIN Synch Break

– reset the SBK bit

– Load the LIN Synch Field (0x55) in the SCIDR register to request Synch Field transmission

– Wait until the SCIDR is empty (TDRE bit set in the SCISR register)

– Load the LIN message Identifier in the SCIDR register to request Identifier transmission.

LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode) (cont'd)

If LHE bit is set due to this error during Fields other than LIN Synch Field or if LASE bit is reset then the current received Header is discarded and the SCI searches for a new Break Field.

Note on LIN Header Time-out Limit

According to the LIN specification, the maximum length of a LIN Header which does not cause a timeout is equal to 1.4 * (34 + 1) = 49 T_{BIT_MASTER}.

T_{BIT_MASTER} refers to the master baud rate.

When checking this timeout, the slave node is desynchronized for the reception of the LIN Break and Synch fields. Consequently, a margin must be allowed, taking into account the worst case: This occurs when the LIN identifier lasts exactly 10 T_{BIT_MASTER} periods. In this case, the LIN Break and Synch fields last 49 - 10 = 39T_{BIT_MASTER} periods.

Assuming the slave measures these first 39 bits with a desynchronized clock of 15.5%. This leads to a maximum allowed Header Length of:

39 x (1/0.845) T_{BIT_MASTER} + 10T_{BIT_MASTER}

= 56.15 T_{BIT_SLAVE}

A margin is provided so that the time-out occurs when the header length is greater than 57 T_{BIT_SLAVE} periods. If it is less than or equal to 57 T_{BIT_SLAVE} periods, then no timeout occurs.

LIN Header Length

Even if no timeout occurs on the LIN Header, it is possible to have access to the effective LIN header Length (T_{HEADER}) through the LHL register. This allows monitoring at software level the T_{FRAME_MAX} condition given by the LIN protocol.

This feature is only available when LHDM bit = 1 or when LASE bit = 1.

Mute Mode and Errors

In mute mode when LHDM bit = 1, if an LHE error occurs during the analysis of the LIN Synch Field or if a LIN Header Time-out occurs then the LHE bit is set but it does not wake up from mute mode. In this case, the current header analysis is discarded. If needed, the software has to reset LSF bit. Then the SCI searches for a new LIN header.

In mute mode, if a framing error occurs on a data (which is not a break), it is discarded and the FE bit is not set.

When LHDM bit = 1, any LIN header which respects the following conditions causes a wake-up from mute mode:

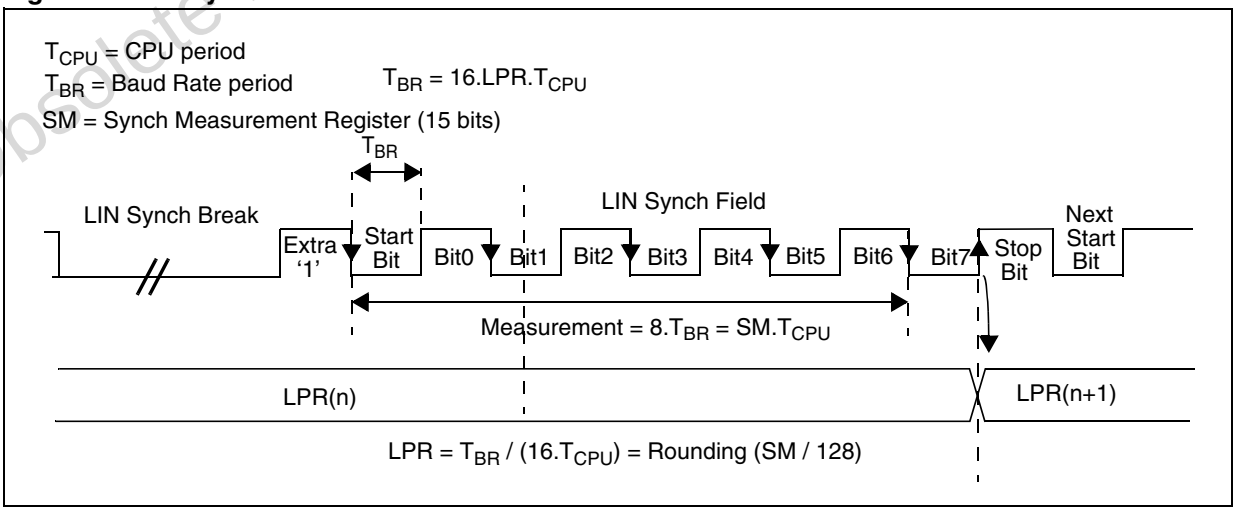
- A valid LIN Break Field (at least 11 dominant bits followed by a recessive bit)

- A valid LIN Synch Field (without deviation error)

- A LIN Identifier Field without framing error. Note that a LIN parity error on the LIN Identifier Field does not prevent wake-up from mute mode.

- No LIN Header Time-out should occur during Header reception.

Figure 83. LIN Synch Field Measurement



beCAN CONTROLLER (Cont'd)

10.9.8.2 Mailbox Registers

This chapter describes the registers of the transmit and receive mailboxes. Refer to Section 0.1.4.4 Message Storage for detailed register mapping.

Transmit and receive mailboxes have the same registers except:

– MCSR register in a transmit mailbox is replaced by MFMI register in a receive mailbox.

– A receive mailbox is always write protected.

– A transmit mailbox is write enable only while empty, corresponding TME bit in the CTPR register set.

MAILBOX CONTROL STATUS REGISTER (MCSR)

Read / Write
Reset Value: 0000 0000 (00h)

7							0
0	0	TERR	ALST	TXOK	RQCP	ABRQ	TXRQ

Bits 7:6 = Reserved. Forced to 0 by hardware.

Bit 5 = **TERR** *Transmission Error*
- Read
This bit is updated by hardware after each transmission attempt.
0: The previous transmission was successful
1: The previous transmission failed due to an error

Bit 4 = **ALST** *Arbitration Lost*
- Read
This bit is updated by hardware after each transmission attempt.
0: The previous transmission was successful
1: The previous transmission failed due to an arbitration lost

Bit 3 = **TXOK** *Transmission OK*
- Read
The hardware updates this bit after each transmission attempt.
0: The previous transmission failed
1: The previous transmission was successful

Note: This bit has the same value as the corresponding TXOKx bit in the CTSR register.

Bit 2 = **RQCP** *Request Completed*
- Read/Clear
Set by hardware when the last request (transmit or abort) has been performed.

Cleared by software writing a "1" or by hardware on transmission request.

Note: This bit has the same value as the corresponding RQCPx bit of the CTSR register.

Clearing this bit clears all the status bits (TXOK, ALST and TERR) in the MCSR register and the RQCP and TXOK bits in the CTSR register.

Bit 1 = **ABRQ** *Abort Request for Mailbox*
- Read/Set
Set by software to abort the transmission request for the corresponding mailbox.

Cleared by hardware when the mailbox becomes empty.

Setting this bit has no effect when the mailbox is not pending for transmission.

Bit 0 = **TXRQ** *Transmit Mailbox Request*

- Read/Set
Set by software to request the transmission for the corresponding mailbox.

Cleared by hardware when the mailbox becomes empty.

Note: This register is implemented only in transmit mailboxes. In receive mailboxes, the MFMI register is mapped at this location.

beCAN CONTROLLER (Cont'd)

CAN FILTER CONFIGURATION REG.1 (CFCR2)

All bits of this register are set and cleared by software.
Read / Write
Reset Value: 0000 0000 (00h)

7							0
0	FSC51	FSC50	FACT5	0	FSC41	FSC40	FACT4

Bit 7 = Reserved. Forced to 0 by hardware.

Bits 6:5 = **FSC5[1:0]** *Filter Scale Configuration*
These bits define the scale configuration of Filter 5.

Bit 4 = **FACT5** *Filter Active*
The software sets this bit to activate filter 5. To modify the Filter 5 registers (CF5R[0:7]) the FACT5 bit must be cleared.
0: Filter 5 is not active
1: Filter 5 is active

Bit 3 = Reserved. Forced to 0 by hardware.

Bits 2:1 = **FSC4[1:0]** *Filter Scale Configuration*
These bits define the scale configuration of Filter 4.

Bit 0 = **FACT4** *Filter Active*
The software sets this bit to activate Filter 4. To modify the Filter 4 registers (CF4R[0:7]), the FACT4 bit must be cleared.
0: Filter 4 is not active
1: Filter 4 is active

CAN FILTER MODE REGISTER (CFMR0)

All bits of this register are set and cleared by software.
Read / Write
Reset Value: 0000 0000 (00h)

7							0
FMH3	FML3	FMH2	FML2	FMH1	FML1	FMH0	FML0

Bit 7 = **FMH3** *Filter Mode High*
Mode of the high registers of Filter 3.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 6 = **FML3** *Filter Mode Low*
Mode of the low registers of Filter 3.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

Bit 5 = **FMH2** *Filter Mode High*
Mode of the high registers of Filter 2.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 4 = **FML2** *Filter Mode Low*
Mode of the low registers of Filter 2.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

Bit 3 = **FMH1** *Filter Mode High*
Mode of the high registers of Filter 1.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 2 = **FML1** *Filter Mode Low*
Mode of the low registers of filter 1.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

Bit 1 = **FMH0** *Filter Mode High*
Mode of the high registers of filter 0.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 0 = **FML0** *Filter Mode Low*
Mode of the low registers of filter 0.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

11 INSTRUCTION SET

11.1 CPU ADDRESSING MODES

The CPU features 17 different addressing modes which can be classified in seven main groups:

Addressing Mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The CPU Instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two submodes called long and short:

– Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.

– Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 37. CPU Addressing Mode Overview

Mode			Syntax	Destination	Pointer Address (Hex.)	Pointer Size (Hex.)	Length (Bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00..FF			+ 0
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC+/-127			+ 1
Relative	Indirect		jrne [$10]	PC+/-127	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

12.3.2 Operating Conditions with Low Voltage Detector (LVD)

Subject to general operating conditions for T_A.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)		4.0[1)]	4.2	4.5	V
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)		3.8	4.0	4.25[1)]	
$V_{hys(LVD)}$	LVD voltage threshold hysteresis[1)]	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$	150	200	250	mV
Vt_{POR}	V_{DD} rise time rate[1)]		6			µs/V
					100	ms/V
$t_{g(VDD)}$	V_{DD} glitches filtered (not detected) by LVD[1)]	Measured at $V_{IT-(LVD)}$			40	ns

Notes:

1. Data based on characterization results, not tested in production.

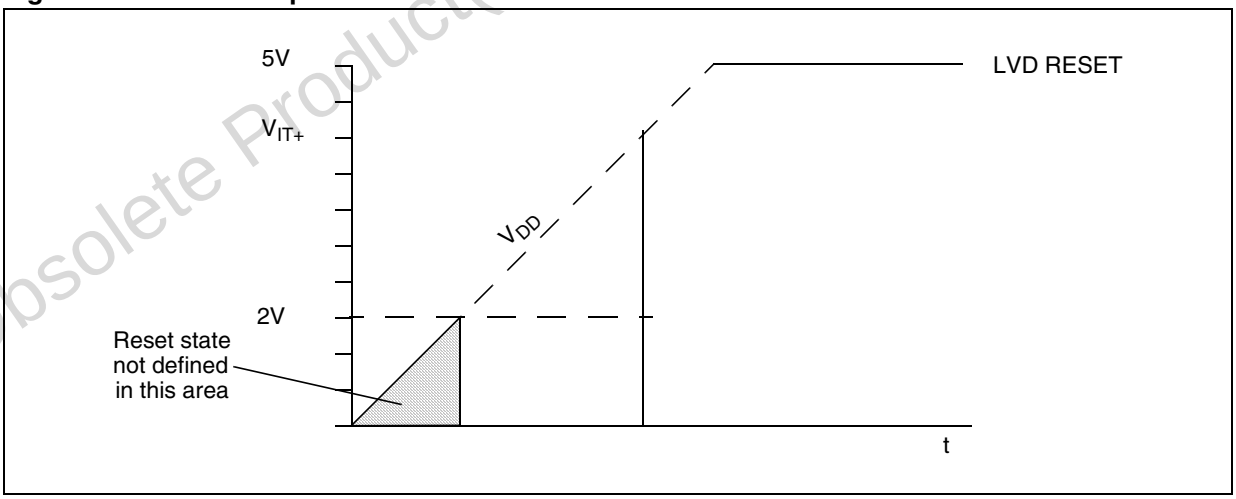
12.3.3 Auxiliary Voltage Detector (AVD) Thresholds

Subject to general operating conditions for T_A.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(AVD)}$	1⇒0 AVDF flag toggle threshold (V_{DD} rise)		4.4[1)]	4.6	4.9	V
$V_{IT-(AVD)}$	0⇒1 AVDF flag toggle threshold (V_{DD} fall)		4.2	4.4	4.65[1)]	
$V_{hys(AVD)}$	AVD voltage threshold hysteresis	$V_{IT+(AVD)}$-$V_{IT-(AVD)}$		250		mV
ΔV_{IT-}	Voltage drop between AVD flag set and LVD reset activated	$V_{IT-(AVD)}$-$V_{IT-(LVD)}$		450		

1. Data based on characterization results, not tested in production.

Figure 120. LVD Startup Behavior



Note: When the LVD is enabled, the MCU reaches its authorized operating voltage from a reset state. However, in some devices, the reset signal may be undefined until V_{DD} is approximately 2V. As a consequence, the I/Os may toggle when V_{DD} is below this voltage.

Because Flash write access is impossible below this voltage, the Flash memory contents will not be corrupted.

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

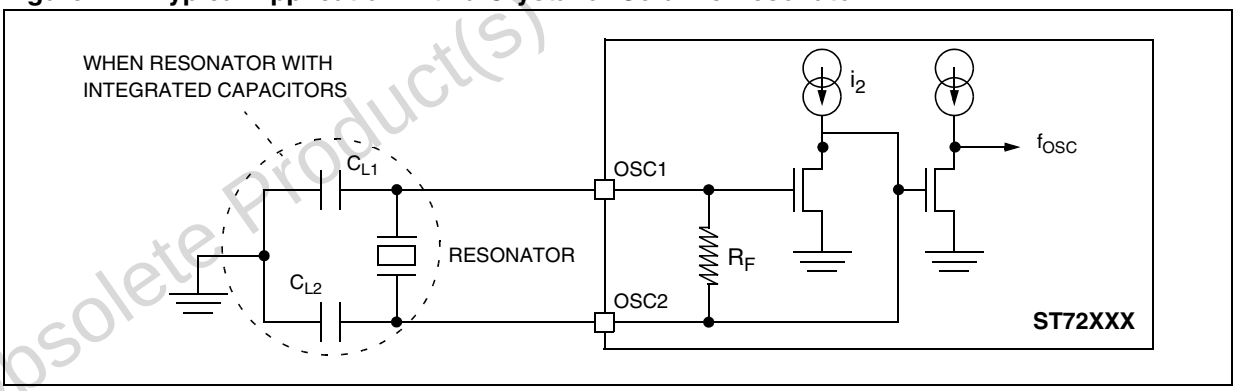
12.5.3 Crystal and Ceramic Resonator Oscillators

The ST7 internal clock can be supplied with four different Crystal/Ceramic resonator oscillators. All the information given in this paragraph is based on characterization results with specified typical external components. In the application, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...). [1)2)]

Symbol	Parameter	Conditions		Min	Max	Unit
f_{OSC}	Oscillator Frequency[3)]	LP: Low power oscillator MP: Medium power oscillator MS: Medium speed oscillator HS: High speed oscillator		1 >2 >4 >8	2 4 8 16	MHz
R_F	Feedback resistor			20	40	kΩ
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)	R_S = 200Ω R_S = 200Ω R_S = 200Ω R_S = 100Ω	LP oscillator MP oscillator MS oscillator HS oscillator	22 22 18 15	56 46 33 33	pF

Symbol	Parameter	Conditions		Typ	Max	Unit
i_2	OSC2 driving current	V_{DD} = 5V V_{IN} = V_{SS}	LP oscillator MP oscillator MS oscillator HS oscillator	80 160 310 610	150 250 460 910	µA

Figure 122. Typical Application with a Crystal or Ceramic Resonator



Notes:

1. Resonator characteristics given by the crystal/ceramic resonator manufacturer.

2. $t_{SU(OSC)}$ is the typical oscillator start-up time measured between V_{DD} = 2.8V and the fetch of the first instruction (with a quick V_{DD} ramp-up from 0 to 5V (< 50µs).

3. The oscillator selection can be optimized in terms of supply current using an high quality resonator with small R_S value. Refer to crystal/ceramic resonator manufacturer for more details.

I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 131. Typical V_{OL} vs V_{DD} (Standard I/Os)

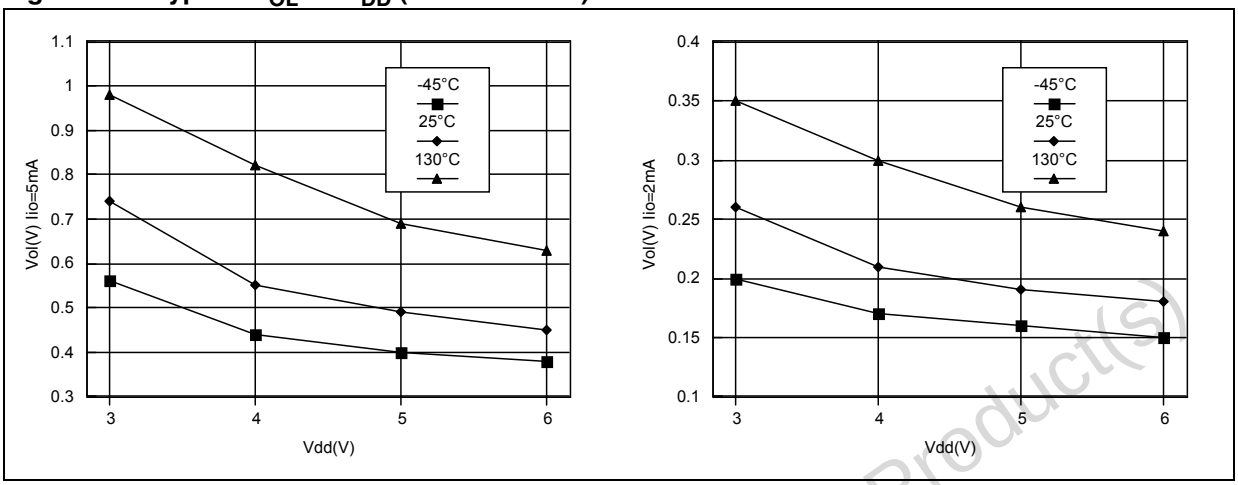
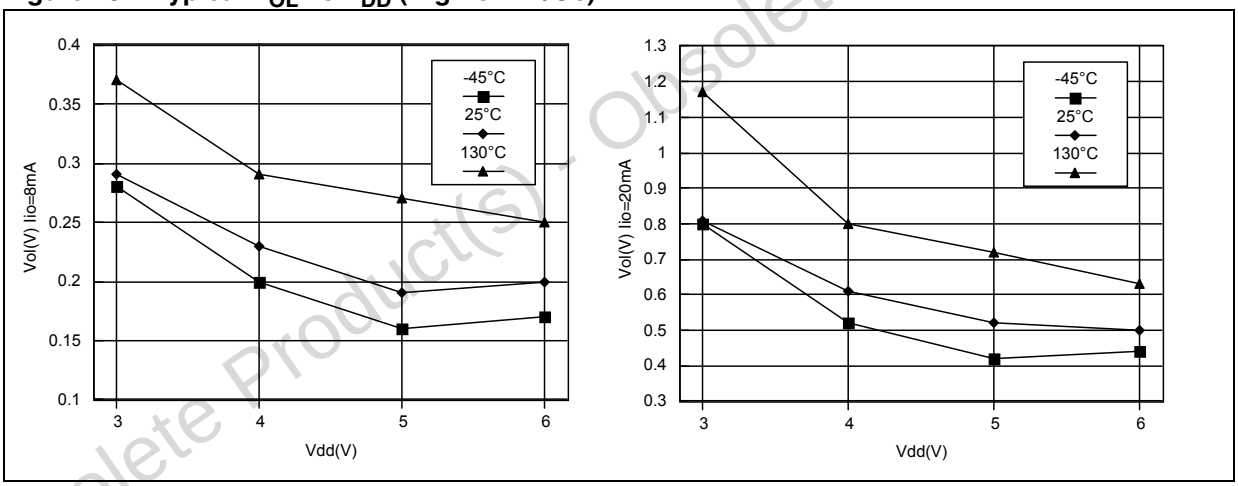
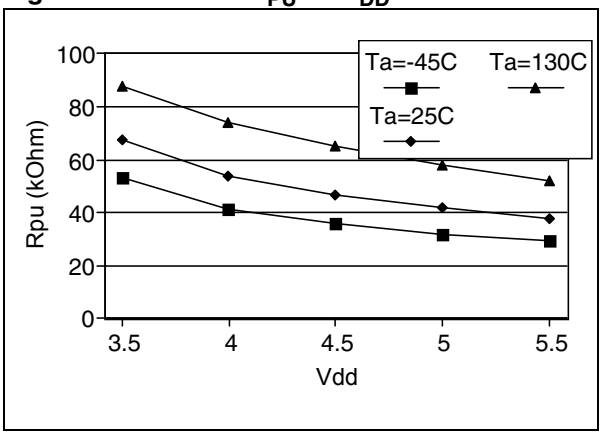


Figure 132. Typical V_{OL} vs V_{DD} (High-sink I/Os)



CONTROL PIN CHARACTERISTICS (Cont'd)

Figure 136. RESET R_{PU} vs V_{DD}

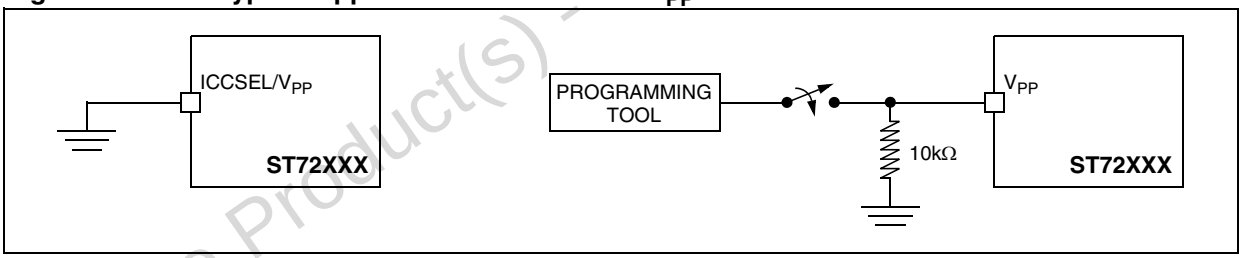


12.10.2 ICCSEL/V_{PP} Pin

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

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input low level voltage[1)]		V_{SS}	0.2	V
V_{IH}	Input high level voltage[1)]		V_{DD}-0.1	12.6	
I_L	Input leakage current	$V_{IN} = V_{SS}$		±1	µA

Figure 137. Two Typical Applications with ICCSEL/V_{PP} Pin[2)]



Notes:

1. Data based on design simulation and/or technology characteristics, not tested in production.
2. When ICC mode is not required by the application ICCSEL/V_{PP} pin must be tied to V_{SS}.

TRANSFER OF CUSTOMER CODE (Cont'd)

ST72561 MICROCONTROLLER OPTION LIST
(Last update: September 2006)

Customer . . . . . .
Address . . . . . .
. . . . . .
Contact . . . . . .
Phone No . . . . . .
Reference/ROM Code* . . . . . .

*The ROM/FASTROM code name is assigned by STMicroelectronics.
ROM/FASTROM code must be sent in .S19 format. .Hex extension cannot be processed.

Device Type/Memory Size/Package (check only one option)

ROM:	Package	60K	48K	32K	16K
	LQFP64 10x10:	[ ] ST72561AR9	[ ] ST72561AR7	[ ] ST72561AR6	[ ] ST72561AR4
	LQFP44:	[ ] ST72561J9	[ ] ST72561J7	[ ] ST72561J6	[ ] ST72561J4
	LQFP32:	[ ] ST72561K9	[ ] ST72561K7	[ ] ST72561K6	[ ] ST72561K4

FASTROM:	Package	60K	48K	32K	16K
	LQFP64 10x10:	[ ] ST72P561AR9	[ ] ST72P561AR7	[ ] ST72P561AR6	[ ] ST72P561AR4
	LQFP44:	[ ] ST72P561J9	[ ] ST72P561J7	[ ] ST72P561J6	[ ] ST72P561J4
	LQFP32:	[ ] ST72P561K9	[ ] ST72P561K7	[ ] ST72P561K6	[ ] ST72P561K4

Conditioning: [ ] Tray [ ] Tape & Reel
Special Marking: [ ] No [ ] Yes " _ _ _ _ _ _ _ _ _ " (10 char. max)
Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Temp. Range. Please refer to datasheet for specific sales conditions:

-	Temp. Range
[ ]	-40°C to +85°C
[ ]	-40°C to +125°C

Clock Source Selection: [ ] Resonator:
[ ] External Source

Oscillator/External source range:
[ ] LP: Low power (1 to 2 MHz)
[ ] MP: Medium power (2 to 4 MHz)
[ ] MS: Medium speed (4 to 8 MHz)
[ ] HS: High speed (8 to 16 MHz)

LVD [ ] Disabled [ ] Enabled
PLL[1] [ ] Disabled [ ] Enabled
Watchdog Selection [ ] Software Activation [ ] Hardware Activation
Watchdog Reset on Halt [ ] Reset [ ] No Reset

Read-out Protection [ ] Disabled [ ] Enabled

Reset Delay [ ] 256 Cycles [ ] 4096 Cycles

LINSCI2 Mapping [ ] Not available (AFIMAP[1] = 0) [ ] Mapped (AFIMAP[1] = 1)
T16_ICAP2 Mapping [ ] On PD1 (AFIMAP[0] = 0) [ ] On PC1 (AFIMAP[0] = 1)

Comments:
Supply Operating Range in the application: . . . . . .

Notes . . . . . .
Signature . . . . . .
Date . . . . . .

[1] If PLL is enabled, medium power (2 to 4 MHz range) has to be selected (MP)
Please download the latest version of this option list from:
http://www.st.com

17 REVISION HISTORY

Date	Revision	Main changes
24-Oct-2005	4	Added standard version 16K ROM/Flash devices Modified data retention in Section 12.7 Added “6” and “3” standard version device type coding to Figure 152 and Figure 150 on page 254 Modified power consumption Section 12.4 Added CDM in Section 12.8.3.1 Added “External interrupt missed” Section 16.1.3
26-Sep-2006	5	Replaced TQFP with LQFP packages throughout document Changed device summary on page 1 Changed Section 9.2.1 on page 46 Changed title of Section 9.6 on page 50 from “I/O Port Implementation” to “I/O Port Register Configurations” Changed Section 10.6.3.3 on page 114 Corrected name of bit 5 in SPICSR register in Table 22 on page 121 Changed Section 12.5.4 on page 228 Added links to Table 39 and Table 40 in Section 12.8.3 on page 233 Removed EMC protection circuitry in Figure 135 on page 240 (device works correctly without these components) Changed Section 12.12.1 on page 243 Changed title of Figure 146 on page 249 Changed title of Figure 124 on page 210 Changed title of Figure 147 on page 249 Changed notes in Section 13.2 on page 251 Changed AFI mapping for “OPTION BYTE 1” on page 253 Changed Figure 150 on page 254, Figure 151 on page 255 and Figure 152 on page 255 Changed “ST72561 MICROCONTROLLER OPTION LIST” on page 256 Deleted Section 15.1.5 “Clearing active interrupts outside interrupt routine” (text already exists in Section 16.1.2 on page 258) Added Section 16.1.5 on page 260 Removed automotive part numbers, see separate ST72561-auto datasheet
19-Jun-2007	6	Added EMI characteristics for LQFP44 in Section 12.8. Updated ADC accuracy characteristics in Section 12.13.
02-Oct-2008	7	Updated static latchup Section 12.8.3.2 on page 233 Updated Flash data retention characteristics in Section 12.7.2. Added Section 16.1.6 on page 261