**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 | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3.8V ~ 5.5V |
| Data Converters | A/D 16x10b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/st72f561ar6t3 |

**PIN DESCRIPTION** (Cont'd)

For external pin connection guidelines, refer to "ELECTRICAL CHARACTERISTICS" on page 219.

**Legend / Abbreviations for Table 2:**

Type: I = input, O = output, S = supply

In/Output level: $C_T$= CMOS $0.3V_{DD}/0.7V_{DD}$ with Schmitt trigger

$T_T$= TTL 0.8V / 2V with Schmitt trigger

Output level: HS = 20mA high sink (on N-buffer only)

Port and control configuration:

– Input: float = floating, wpu = weak pull-up, int = interrupt[1], ana = analog, RB = robust

– Output: OD = open drain, PP = push-pull

Refer to "I/O PORTS" on page 46 for more details on the software configuration of the I/O ports.

The RESET configuration of each pin is shown in bold which is valid as long as the device is in reset state.

**Table 2. Device Pin Description**

| Pin n° | | | Pin Name | Type | Level | | Port | | | | | | Main function (after reset) | Alternate function | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Input | | | | Output | | | | |
| LQFP64 | LQFP44 | LQFP32 | | | Input | Output | float | wpu | int | ana | OD | PP | | | |
| 1 | 1 | 1 | OSC1[3] | I | | | | | | | | | External clock input or Resonator oscillator inverter input | | |
| 2 | 2 | 2 | OSC2[3] | I/O | | | | | | | | | Resonator oscillator inverter output | | |
| 3 | - | - | PA0 / ARTIC1 | I/O | $C_T$ | | **X** | ei0 | | | X | X | Port A0 | ART Input Capture 1 | |
| 4 | 3 | 3 | PA1 / PWM0 | I/O | $C_T$ | | **X** | | ei0 | | X | X | Port A1 | ART PWM Output 0 | |
| 5 | 4 | 4 | PA2 (HS) / PWM1 | I/O | $C_T$ | HS | **X** | ei0 | | | X | X | Port A2 | ART PWM Output 1 | |
| 6 | 5 | - | PA3 / PWM2 | I/O | $C_T$ | | **X** | | ei0 | | X | X | Port A3 | ART PWM Output 2 | |
| 7 | 6 | - | PA4 / PWM3 | I/O | $C_T$ | | **X** | ei0 | | | X | X | Port A4 | ART PWM Output 3 | |
| 8 | - | - | $V_{SS\_3}$ | S | | | | | | | | | Digital Ground Voltage | | |
| 9 | - | - | $V_{DD\_3}$ | S | | | | | | | | | Digital Main Supply Voltage | | |
| 10 | 7 | 5 | PA5 (HS) / ARTCLK | I/O | $C_T$ | HS | **X** | | ei0 | | X | X | Port A5 | ART External Clock | |
| 11 | 8 | - | PA6 (HS) / ARTIC2 | I/O | $C_T$ | HS | **X** | ei0 | | | X | X | Port A6 | ART Input Capture 2 | |
| 12 | - | - | PA7 / T8_OCMP2 | I/O | $C_T$ | | **X** | | ei0 | | X | X | Port A7 | TIM8 Output Compare 2 | |
| 13 | - | - | PB0 /T8_ICAP2 | I/O | $C_T$ | | **X** | ei1 | | | X | X | Port B0 | TIM8 Input Capture 2 | |
| 14 | 9 | 6 | PB1 /T8_OCMP1 | I/O | $C_T$ | | **X** | | ei1 | | X | X | Port B1 | TIM8 Output Compare 1 | |
| 15 | 10 | 7 | PB2 / T8_ICAP1 | I/O | $C_T$ | | **X** | ei1 | | | X | X | Port B2 | TIM8 Input Capture 1 | |
| 16 | 11 | 8 | PB3 / MCO | I/O | $C_T$ | | **X** | | ei1 | | X | X | Port B3 | Main clock out ($f_{OSC2}$) | |
| 17 | - | - | PE0 / AIN12 | I/O | $T_T$ | | **X** | X | | RB | X | X | Port E0 | ADC Analog Input 12 | |
| 18 | - | - | PE1 / AIN13 | I/O | $T_T$ | | **X** | X | | RB | X | X | Port E1 | ADC Analog Input 13 | |
| 19 | 12 | 9 | PB4 / AIN0 / ICCCLK | I/O | $C_T$ | | **X** | ei1 | | RB | X | X | Port B4 | ICC Clock input | ADC Analog Input 0 |
| 20 | - | - | PE2 / AIN14 | I/O | $T_T$ | | **X** | X | | RB | X | X | Port E2 | ADC Analog Input 14 | |
| 21 | - | - | PE3 / AIN15 | I/O | $T_T$ | | **X** | X | | RB | X | X | Port E3 | ADC Analog Input 15 | |
| 22 | 13 | 10 | PB5 / AIN1 / ICCDATA | I/O | $C_T$ | | **X** | | ei1 | RB | X | X | Port B5 | ICC Data input | ADC Analog Input 1 |

# 6 SUPPLY, RESET AND CLOCK MANAGEMENT

The device includes a range of utility features for securing the application in critical situations (for example, in case of a power brown-out), and reducing the number of external components. An overview is shown in Figure 11.

For more details, refer to dedicated parametric section.

### Main features

- Optional PLL for multiplying the frequency by 2
- Reset Sequence Manager (RSM)
- Multi-Oscillator Clock Management (MO)
  - 4 Crystal/Ceramic resonator oscillators
- System Integrity Management (SI)
  - Main supply Low voltage detection (LVD)
  - Auxiliary Voltage detector (AVD) with interrupt capability for monitoring the main supply

## 6.1 PHASE LOCKED LOOP

If the clock frequency input to the PLL is in the range 2 to 4 MHz, the PLL can be used to multiply the frequency by two to obtain an $f_{OSC2}$ of 4 to 8 MHz. The PLL is enabled by option byte. If the PLL is disabled, then $f_{OSC2} = f_{OSC}/2$.

**Caution:** The PLL is not recommended for applications where timing accuracy is required. See "PLL Characteristics" on page 228.

**Figure 10. PLL Block Diagram**

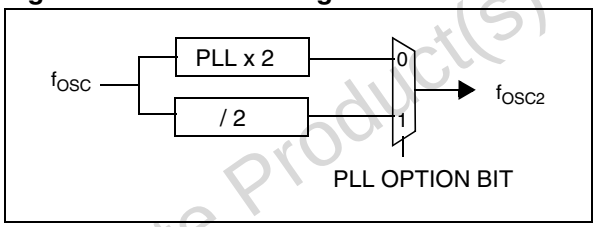
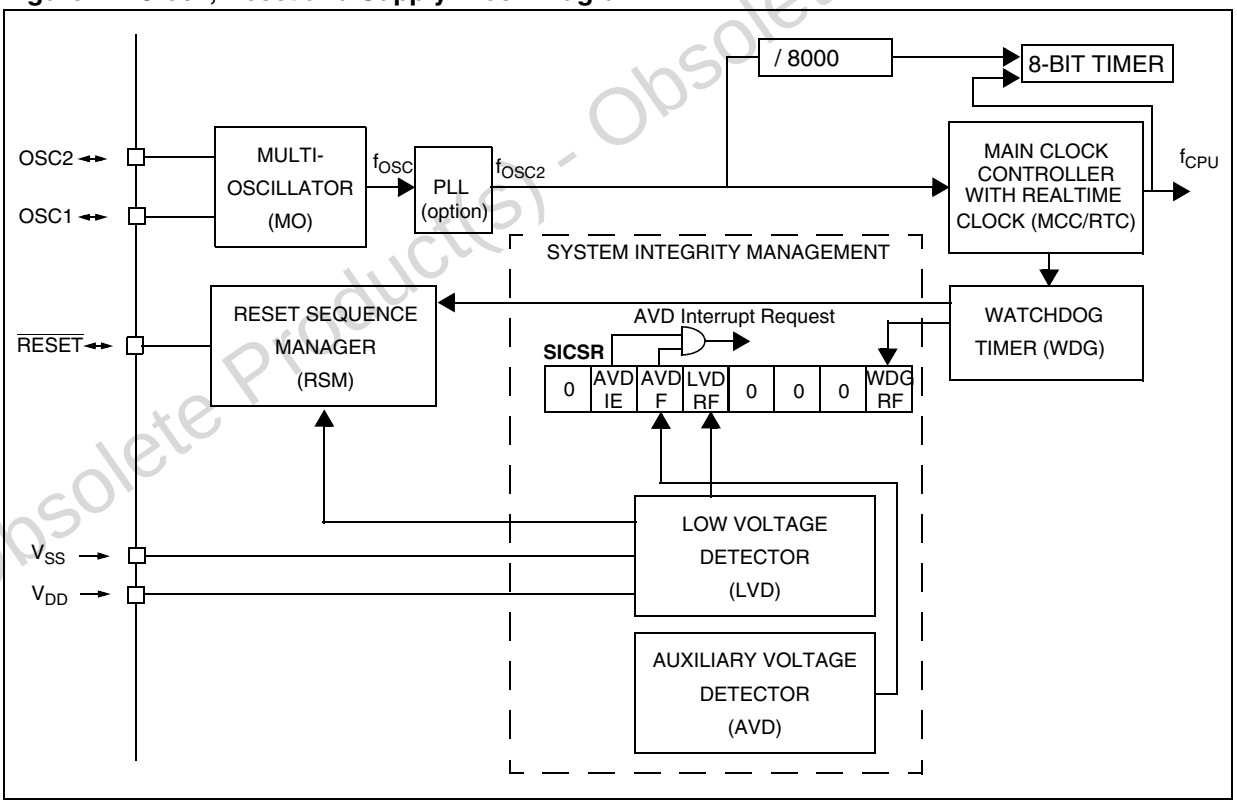



**Figure 11. Clock, Reset and Supply Block Diagram**



**INTERRUPTS** (Cont'd)

### 7.5 INTERRUPT REGISTER DESCRIPTION

#### CPU CC REGISTER INTERRUPT BITS

Read/Write

Reset Value: 111x 1010 (xAh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | **I1** | H | **I0** | N | Z | C |

Bit 5, 3 = **I1, I0** *Software Interrupt Priority*

These two bits indicate the current interrupt software priority.

| Interrupt Software Priority | Level | I1 | I0 |
|---|---|---|---|
| Level 0 (main) | Low | 1 | 0 |
| Level 1 | ↓ | 0 | 1 |
| Level 2 | ↓ | | 0 |
| Level 3 (= interrupt disable*) | High | 1 | 1 |

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (ISPRx).

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see "Interrupt Dedicated Instruction Set" table).

***Note**: TLI, TRAP and RESET events can interrupt a level 3 program.

#### INTERRUPT SOFTWARE PRIORITY REGISTERS (ISPRX)

Read/Write (bit 7:4 of **ISPR3** are read only)

Reset Value: 1111 1111 (FFh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| ISPR0 | I1_3 | I0_3 | I1_2 | I0_2 | I1_1 | I0_1 | I1_0 | I0_0 |
| ISPR1 | I1_7 | I0_7 | I1_6 | I0_6 | I1_5 | I0_5 | I1_4 | I0_4 |
| ISPR2 | I1_11 | I0_11 | I1_10 | I0_10 | I1_9 | I0_9 | I1_8 | I0_8 |
| ISPR3 | 1 | 1 | 1 | 1 | I1_13 | I0_13 | I1_12 | I0_12 |

These four registers contain the interrupt software priority of each interrupt vector.

– Each interrupt vector (except RESET and TRAP) has corresponding bits in these registers where its own software priority is stored. This correspondence is shown in the following table.

| Vector address | ISPRx bits |
|---|---|
| FFFBh-FFFAh | I1_0 and I0_0 bits* |
| FFF9h-FFF8h | I1_1 and I0_1 bits |
| ... | ... |
| FFE1h-FFE0h | I1_13 and I0_13 bits |

– Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.

– Level 0 cannot be written (I1_x = 1, I0_x = 0). In this case, the previously stored value is kept (Example: previous = CFh, write = 64h, result = 44h)

The RESET, TRAP and TLI vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

***Note**: Bits in the ISPRx registers which correspond to the TLI can be read and written but they are not significant in the interrupt process management.

**Caution**: If the I1_x and I0_x bits are modified while the interrupt x is executed the following behavior has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

**WINDOW WATCHDOG** (Cont'd)

### 10.1.9 Interrupts

None.

### 10.1.10 Register Description

**CONTROL REGISTER (WDGCR)**

Read/Write

Reset Value: 0111 1111 (7Fh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| WDGA | T6 | T5 | T4 | T3 | T2 | T1 | T0 |

Bit 7 = **WDGA** *Activation bit.*

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled

**Note:** This bit is not used if the hardware watchdog option is enabled by option byte.

Bits 6:0 = **T[6:0]** *7-bit counter (MSB to LSB).*
These bits contain the value of the watchdog counter. It is decremented every 16384 $f_{OSC2}$ cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

**WINDOW REGISTER (WDGWR)**

Read/Write

Reset Value: 0111 1111 (7Fh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| - | W6 | W5 | W4 | W3 | W2 | W1 | W0 |

Bit 7 = Reserved

Bits 6:0 = **W[6:0]** *7-bit window value*
These bits contain the window value to be compared to the downcounter.



**Figure 38. Watchdog Timer Register Map and Reset Values**

| Address (Hex.) | Register Label | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 2F | **WDGCR** | WDGA | T6 | T5 | T4 | T3 | T2 | T1 | T0 |
| | Reset Value | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30 | **WDGWR** | - | W6 | W5 | W4 | W3 | W2 | W1 | W0 |
| | Reset Value | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

## 10.4 16-BIT TIMER

### 10.4.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

Some ST7 devices have two on-chip 16-bit timers. They are completely independent, and do not share any resources. They are synchronized after a MCU reset as long as the timer clock frequencies are not modified.

This description covers one or two 16-bit timers. In ST7 devices with two timers, register names are prefixed with TA (Timer A) or TB (Timer B).

### 10.4.2 Main Features

- Programmable prescaler: $f_{CPU}$ divided by 2, 4 or 8
- Overflow status flag and maskable interrupt
- External clock input (must be at least four times slower than the CPU clock speed) with the choice of active edge
- 1 or 2 Output Compare functions each with:
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- 1 or 2 Input Capture functions each with:
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One Pulse mode
- Reduced Power Mode
- 5 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)*

The Block Diagram is shown in Figure 48.

***Note:** Some timer pins may not be available (not bonded) in some ST7 devices. Refer to the device pin out description.

When reading an input signal on a non-bonded pin, the value will always be '1'.

### 10.4.3 Functional Description

#### 10.4.3.1 Counter

The main block of the Programmable Timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high and low.

Counter Register (CR):

- Counter High Register (CHR) is the most significant byte (MS Byte).
- Counter Low Register (CLR) is the least significant byte (LS Byte).

Alternate Counter Register (ACR)

- Alternate Counter High Register (ACHR) is the most significant byte (MS Byte).
- Alternate Counter Low Register (ACLR) is the least significant byte (LS Byte).

These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register, (SR), (see note at the end of paragraph titled 16-bit read sequence).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.
Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in One Pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in Table 17 Clock Control Bits. The value in the counter register repeats every 131072, 262144 or 524288 CPU clock cycles depending on the CC[1:0] bits.
The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

**16-BIT TIMER** (Cont'd)

**Figure 49. Counter Timing Diagram, Internal Clock Divided by 2**

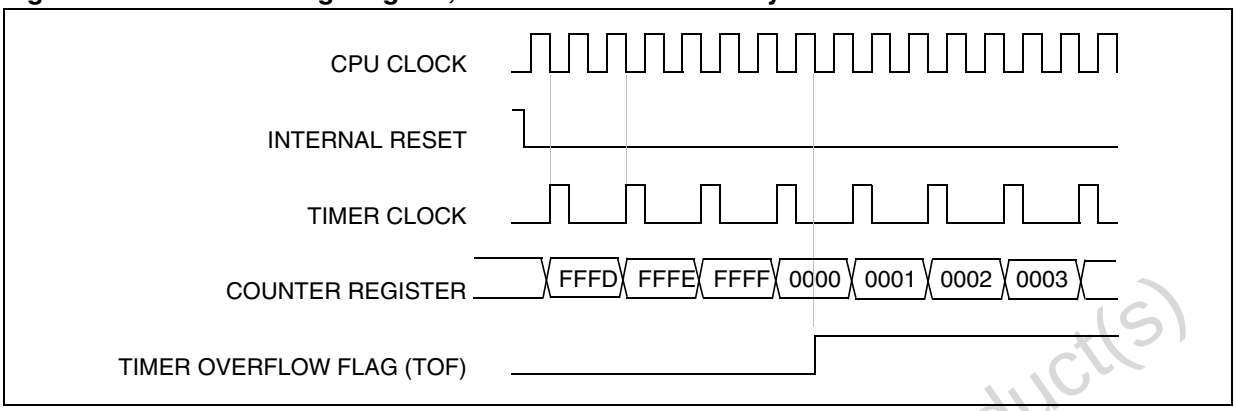



**Figure 50. Counter Timing Diagram, Internal Clock Divided by 4**

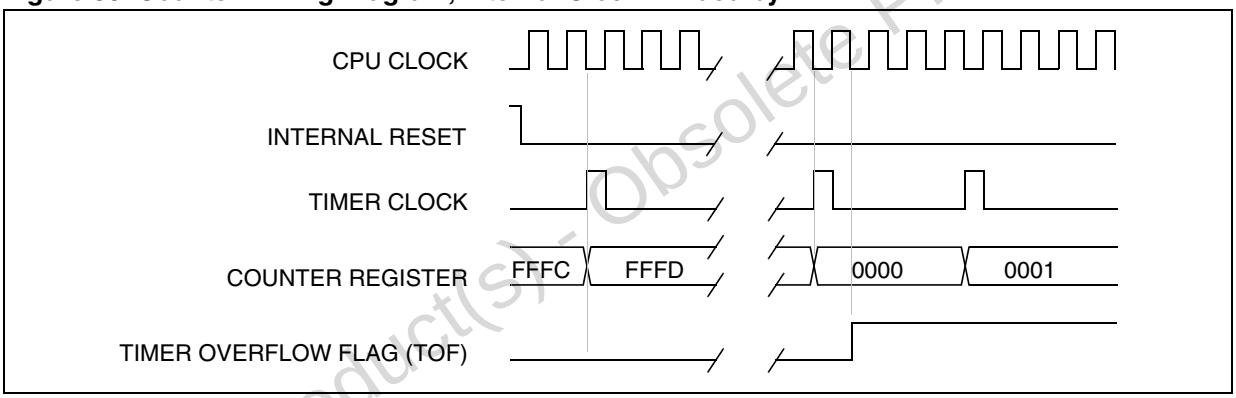
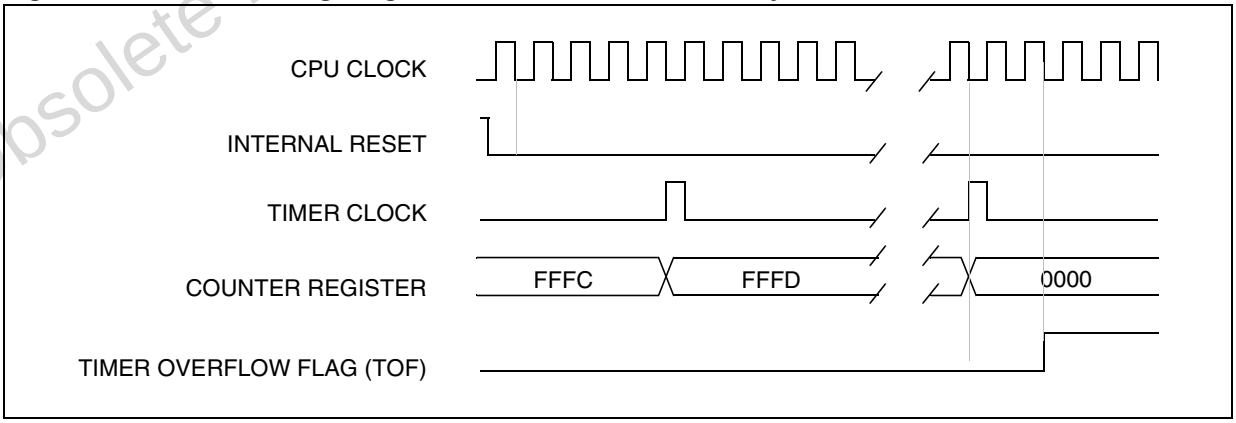



**Figure 51. Counter Timing Diagram, Internal Clock Divided By 8**




**Note:** The MCU is in reset state when the internal reset signal is high, when it is low the MCU is running.

**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 18. Clock Control Bits**

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

**OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)**

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**OUTPUT COMPARE 2 LOW REGISTER (OC2LR)**

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

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**COUNTER HIGH REGISTER (CHR)**

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**COUNTER LOW REGISTER (CLR)**

Read Only
Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**ALTERNATE COUNTER HIGH REGISTER (ACHR)**

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**ALTERNATE COUNTER LOW REGISTER (ACLR)**

Read Only
Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**INPUT CAPTURE 2 HIGH REGISTER (IC2HR)**

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**INPUT CAPTURE 2 LOW REGISTER (IC2LR)**

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the Input Capture 2 event).

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |

**8-BIT TIMER** (Cont'd)

**Figure 63. Input Capture Block Diagram**

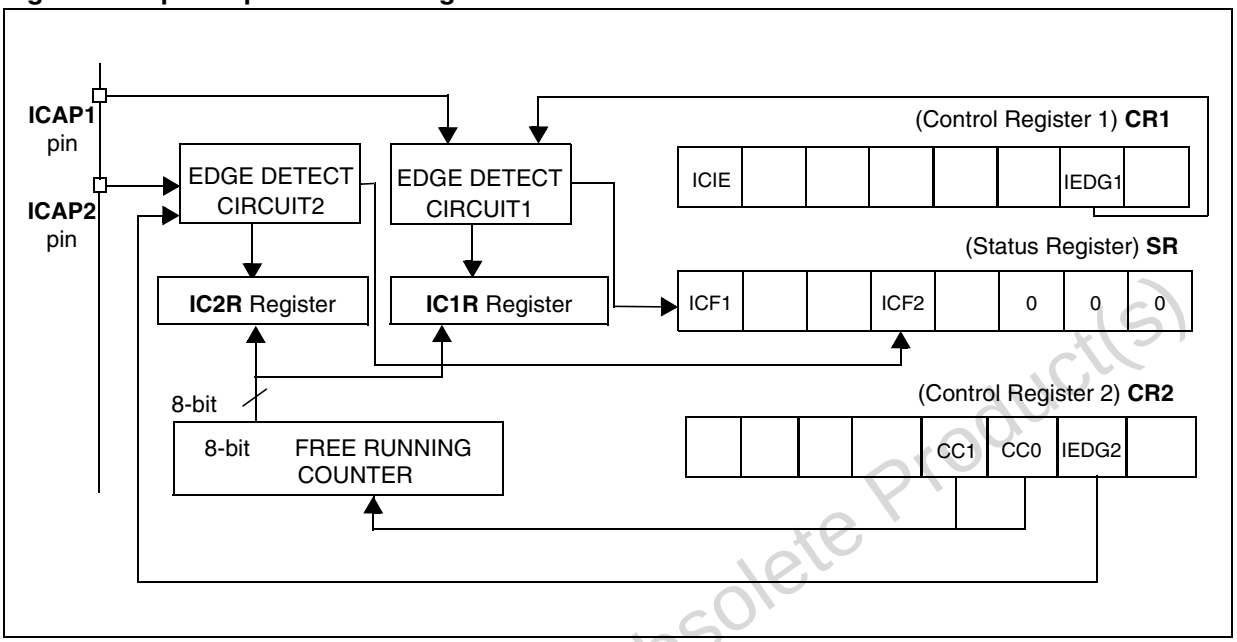
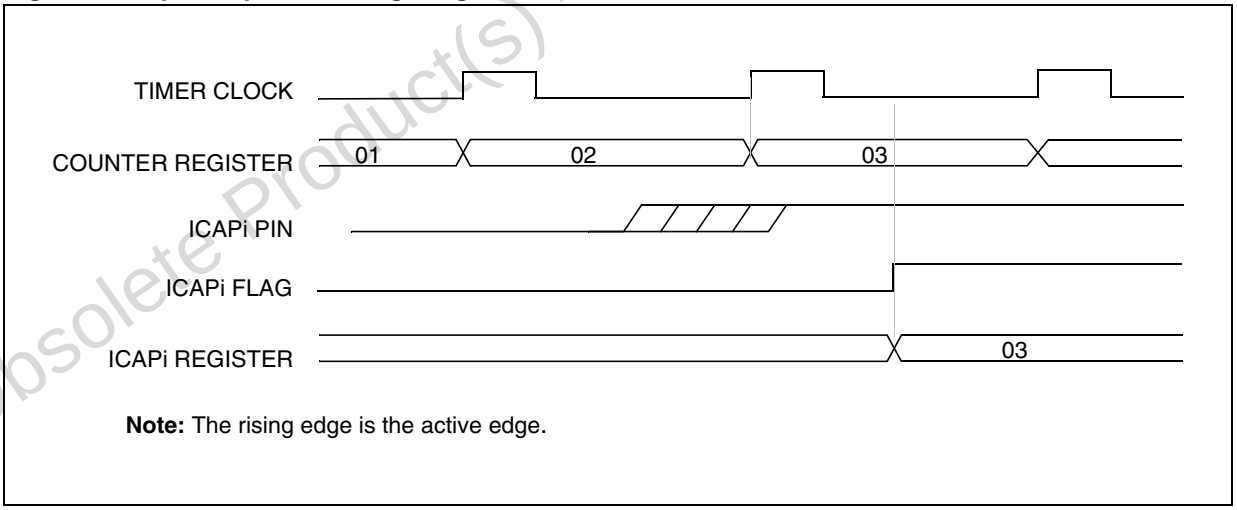



**Figure 64. Input Capture Timing Diagram**




**Note:** The rising edge is the active edge.

**8-BIT TIMER** (Cont'd)

### 10.5.8 8-bit Timer Register Map

| Address (Hex.) | Register Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 3C | **CR2** | OC1E | OC2E | OPM | PWM | CC1 | CC0 | IEDG2 | 0 |
| 3D | **CR1** | ICIE | OCIE | TOIE | FOLV2 | FOLV1 | OLVL2 | IEDG1 | OLVL1 |
| 3E | **CSR** | ICF1 | OCF1 | TOF | ICF2 | OCF2 | TIMD | | |
| 3F | **IC1R** | MSB | | | | | | | LSB |
| 40 | **OC1R** | MSB | | | | | | | LSB |
| 41 | **CTR** | MSB | | | | | | | LSB |
| 42 | **ACTR** | MSB | | | | | | | LSB |
| 43 | **IC2R** | MSB | | | | | | | LSB |
| 44 | **OC2R** | MSB | | | | | | | LSB |

**beCAN CONTROLLER** (Cont'd)

**Filter Bank Scale and Mode Configuration**

The filter banks are configured by means of the corresponding CFCRx register. To configure a filter bank this must be deactivated by clearing the FACT bit in the CFCR register. The filter scale is configured by means of the FSC[1:0] bits in the corresponding CFCR register, refer to Figure 9. Filter Bank Scale Configuration - Register Organisation. The **identifier list** or **identifier mask** mode for the corresponding Mask/Identifier registers is configured by means of the FMCLx and FMCHx bits in the CFMR register. The FMCLx bit defines the mode for the two least significant bytes, and the FMCHx bit the mode for the two most significant bytes of filter bank x. **Examples**:

– If filter bank 1 is configured as two 16-bit filters, then the FMCL1 bit defines the mode of the CF1R2 and CF1R3 registers and the FMCH1 bit defines the mode of the CF1R6 and CF1R7 registers.

– If filter bank 2 is configured as four 8-bit filters, then the FMCL2 bit defines the mode of the CF2R1 and CF2R3 registers and the FMCH2 bit defines the mode of the CF2R5 and CF2R7 registers.

**Note**: In 32-bit configuration, the FMCLx and FMCHx bits must have the same value to ensure that the four Mask/Identifier registers are in the same mode.

To filter a group of identifiers, configure the Mask/Identifier registers in mask mode.

To select single identifiers, configure the Mask/Identifier registers in identifier list mode.

Filters not used by the application should be left deactivated.

**Filter Match Index**

Once a message has been received in the FIFO it is available to the application. Typically application data are copied into RAM locations. To copy the data to the right location the application has to identify the data by means of the identifier. To avoid this and to ease the access to the RAM locations, the CAN controller provides a Filter Match Index.

This index is stored in the mailbox together with the message according to the filter priority rules. Thus each received message has its associated Filter Match Index.

The Filter Match Index can be used in two ways:

– Compare the Filter Match Index with a list of expected values.

– Use the Filter Match Index as an index on an array to access the data destination location.

For non-masked filters, the software no longer has to compare the identifier.

If the filter is masked the software reduces the comparison to the masked bits only.

**Filter Priority Rules**

Depending on the filter combination it may occur that an identifier passes successfully through several filters. In this case the filter match value stored in the receive mailbox is chosen according to the following rules:

– A filter in identifier list mode prevails on an filter in mask mode.

– A filter with full identifier coverage prevails over filters covering part of the identifier, e.g. 16-bit filters prevail over 8-bit filters.

– Filters configured in the same mode and with identical coverage are prioritized by filter number and register number. The lower the number the higher the priority.

**beCAN CONTROLLER** (Cont'd)

Bit 4 = **TXOK0** *Transmission OK for mailbox 0*
- Read
This bit is set by hardware when the transmission request on mailbox 0 has been completed successfully. Please refer to Figure 7.

This bit is cleared by hardware when mailbox 0 is requested for transmission or when the software clears the RQCP0 bit.

Bits 3:2 = Reserved. Forced to 0 by hardware.

Bit 1 = **RQCP1** *Request Completed for Mailbox 1*
- Read/Clear
This bit is set by hardware to signal that the last request for mailbox 1 has been completed. The request could be a transmit or an abort request.

This bit is cleared by software.

Bit 0 = **RQCP0** *Request Completed for Mailbox 0*
- Read/Clear
This bit is set by hardware to signal that the last request for mailbox 0 has been completed. The request could be a transmit or an abort request.

This bit is cleared by software.

**CAN TRANSMIT PRIORITY REGISTER (CTPR)**
All bits of this register are read only.

Reset Value: 0000 1100 (0Ch)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 0 | LOW1 | LOW0 | 0 | TME1 | TME0 | 0 | CODE |

Bit 7 = Reserved. Forced to 0 by hardware.

Bit 6 = **LOW1** *Lowest Priority Flag for Mailbox 1*
- Read
This bit is set by hardware when more than one mailbox are pending for transmission and mailbox 1 has the lowest priority.

Bit 5 = **LOW0** *Lowest Priority Flag for Mailbox 0*
- Read
This bit is set by hardware when more than one mailbox are pending for transmission and mailbox 0 has the lowest priority.

**Note**: These bits are set to zero when only one mailbox is pending.

Bit 4 = Reserved. Forced to 0 by hardware.

Bit 3 = **TME1** *Transmit Mailbox 1 Empty*
- Read
This bit is set by hardware when no transmit request is pending for mailbox 1.

Bit 2 = **TME0** *Transmit Mailbox 0 Empty*
- Read
This bit is set by hardware when no transmit request is pending for mailbox 0.

Bit 1:0 = **CODE** *Mailbox Code*
- Read
In case at least one transmit mailbox is free, the code value is equal to the number of the next transmit mailbox free.

In case all transmit mailboxes are pending, the code value is equal to the number of the transmit mailbox with the lowest priority.

**beCAN CONTROLLER** (Cont'd)

**Table 32. beCAN Control and Status Page - Register Map and Reset Values**

| Address (Hex.) | Register Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 68h | **CMCR** | | ABOM | AWUM | NART | RFLM | TXFP | SLEEP | INRQ |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 69h | **CMSR** | | | REC | TRAN | WKUI | ERRI | SLAK | INAK |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 6Ah | **CTSR** | | | TXOK1 | TXOK0 | | | RQCP1 | RQCP0 |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6Bh | **CTPR** | | LOW1 | LOW0 | | TME1 | TME0 | | CODE0 |
| | Reset Value | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 6Ch | **CRFR** | | | RFOM | FOVR | FULL | | FMP1 | FMP0 |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6Dh | **CIER** | WKUIE | 0 | 0 | 0 | FOVIE0 | FFIE0 | FMPIE0 | TMEIE |
| | Reset Value | 0 | | | | 0 | 0 | 0 | 0 |
| 6Eh | **CDGR** | | | | | RX | SAMP | SILM | LBKM |
| | Reset Value | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 6Fh | **CFPSR** | | | | | | FPS2 | FPS1 | FPS0 |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 33. beCAN Mailbox Pages - Register Map and Reset Values**

| Address (Hex.) | Register Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 70h | **MFMI** | FMI7 | FMI6 | FMI5 | FMI4 | FMI3 | FMI2 | FMI1 | FMI0 |
| Receive | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 70h | **MCSR** | | | TERR | ALST | TXOK | RQCP | ABRQ | TXRQ |
| Transmit | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 71h | **MDLC** | 0 | | | | DLC3 | DLC2 | DLC1 | DLC0 |
| | Reset Value | x | x | x | x | x | x | x | x |
| 72h | **MIDR0** | | IDE | RTR | STID10 | STID9 | STID8 | STID7 | STID6 |
| | Reset Value | x | x | x | x | x | x | x | x |
| 73h | **MIDR1** | STID5 | STID4 | STID3 | STID2 | STID1 | STID0 | EXID17 | EXID16 |
| | Reset Value | x | x | x | x | x | x | x | x |
| 74h | **MIDR2** | EXID15 | EXID14 | EXID13 | EXID12 | EXID11 | EXID10 | EXID9 | EXID8 |
| | Reset Value | x | x | x | x | x | x | x | x |
| 75h | **MIDR3** | EXID7 | EXID6 | EXID5 | EXID4 | EXID3 | EXID2 | EXID1 | EXID0 |
| | Reset Value | x | x | x | x | x | x | x | x |
| 76h:7Dh | **MDAR[0:7]** | MDAR7 | MDAR6 | MDAR5 | MDAR4 | MDAR3 | MDAR2 | MDAR1 | MDAR0 |
| | Reset Value | x | x | x | x | x | x | x | x |

**INSTRUCTION SET OVERVIEW** (Cont'd)

### 11.2 INSTRUCTION GROUPS

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

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Load and Transfer | LD | CLR | | | | | | |
| Stack operation | PUSH | POP | RSP | | | | | |
| Increment/Decrement | INC | DEC | | | | | | |
| Compare and Tests | CP | TNZ | BCP | | | | | |
| Logical operations | AND | OR | XOR | CPL | NEG | | | |
| Bit Operation | BSET | BRES | | | | | | |
| Conditional Bit Test and Branch | BTJT | BTJF | | | | | | |
| Arithmetic operations | ADC | ADD | SUB | SBC | MUL | | | |
| Shift and Rotates | SLL | SRL | SRA | RLC | RRC | SWAP | SLA | |
| Unconditional Jump or Call | JRA | JRT | JRF | JP | CALL | CALLR | NOP | RET |
| Conditional Branch | JRxx | | | | | | | |
| Interruption management | TRAP | WFI | HALT | IRET | | | | |
| Condition Code Flag modification | SIM | RIM | SCF | RCF | | | | |

**Using a prebyte**

The instructions are described with one to four opcodes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC Opcode

PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

## SUPPLY CURRENT CHARACTERISTICS (Cont'd)

### 12.4.1 Supply and Clock Managers

The previous current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To obtain the total device consumption, the two current values must be added (except for HALT mode).

| Symbol | Parameter | Conditions | Typ | Max[1)] | Unit |
|---|---|---|---|---|---|
| $I_{DD(RES)}$ | Supply current of resonator oscillator[2)3)] | | See Section 12.5.3 on page 227 | | µA |
| $I_{DD(PLL)}$ | PLL supply current | $V_{DD}$ = 5V | 360 | | |
| $I_{DD(LVD)}$ | LVD supply current | HALT mode, $V_{DD}$ = 5V | 150 | 300 | |

**Notes:**

1. Data based on characterization results, not tested in production.
2. Data based on characterization results done with the external components specified in Section 12.5.3, not tested in production.
3. As the oscillator is based on a current source, the consumption does not depend on the voltage.

**CLOCK AND TIMING CHARACTERISTICS** (Cont'd)

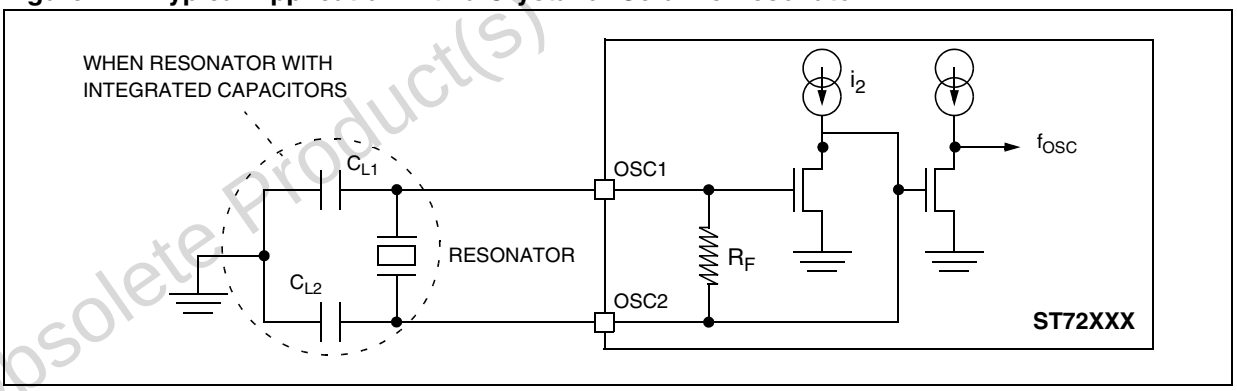
### 12.5.3 Crystal and Ceramic Resonator Oscillators

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<br>MP: Medium power oscillator<br>MS: Medium speed oscillator<br>HS: High speed oscillator | | 1<br>>2<br>>4<br>>8 | 2<br>4<br>8<br>16 | MHz |
| $R_F$ | Feedback resistor | | | 20 | 40 | kΩ |
| $C_{L1}$<br>$C_{L2}$ | Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator ($R_S$) | $R_S$ = 200Ω<br>$R_S$ = 200Ω<br>$R_S$ = 200Ω<br>$R_S$ = 100Ω | LP oscillator<br>MP oscillator<br>MS oscillator<br>HS oscillator | 22<br>22<br>18<br>15 | 56<br>46<br>33<br>33 | pF |

| Symbol | Parameter | Conditions | | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $i_2$ | OSC2 driving current | $V_{DD}$ = 5V<br>$V_{IN}$ = $V_{SS}$ | LP oscillator<br>MP oscillator<br>MS oscillator<br>HS oscillator | 80<br>160<br>310<br>610 | 150<br>250<br>460<br>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.

**CLOCK CHARACTERISTICS** (Cont'd)

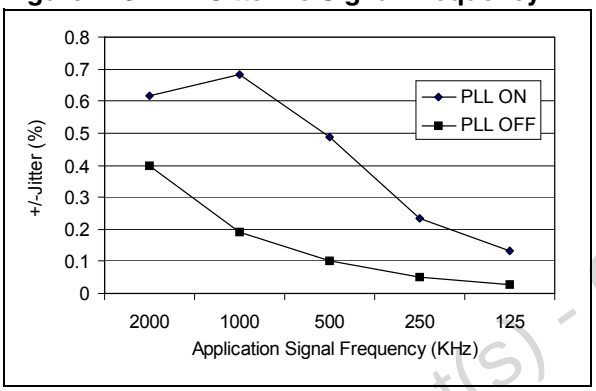
### 12.5.4 PLL Characteristics

Operating conditions: $V_{DD}$ 3.8 to 5.5V @ $T_A$ 0 to 70°C[1)] or $V_{DD}$ 4.5 to 5.5V @ $T_A$ -40 to 125°C

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DD(PLL)}$ | PLL Voltage Range | $T_A$ = 0 to +70°C | 3.8 | | 5.5 | |
| | | $T_A$ = -40 to +125°C | 4.5 | | | |
| $f_{OSC}$ | PLL input frequency range | | 2 | | 4 | MHz |
| $\Delta f_{CPU}/f_{CPU}$ | PLL jitter[1)] | $f_{OSC}$ = 4 MHz, $V_{DD}$ = 4.5 to 5.5V | | Note 2 | | % |
| | | $f_{OSC}$ = 2 MHz, $V_{DD}$ = 4.5 to 5.5V | | | | |

**Notes:**
1. Data characterized but not tested.
2. Under characterization

**Figure 123. PLL Jitter vs Signal Frequency[1)]**




**Notes:**
1. Measurement conditions: $f_{CPU}$ = 4 MHz, $T_A$ = 25°C

The user must take the PLL jitter into account in the application (for example in serial communication or sampling of high frequency signals). The PLL jitter is a periodic effect, which is integrated over several CPU cycles. Therefore, the longer the period of the application signal, the less it is impacted by the PLL jitter.

Figure 123 shows the PLL jitter integrated on application signals in the range 125 kHz to 2 MHz. At frequencies of less than 125 kHz, the jitter is negligible.

**I/O PORT PIN CHARACTERISTICS** (Cont'd)

**Figure 125. Connecting Unused I/O Pins**

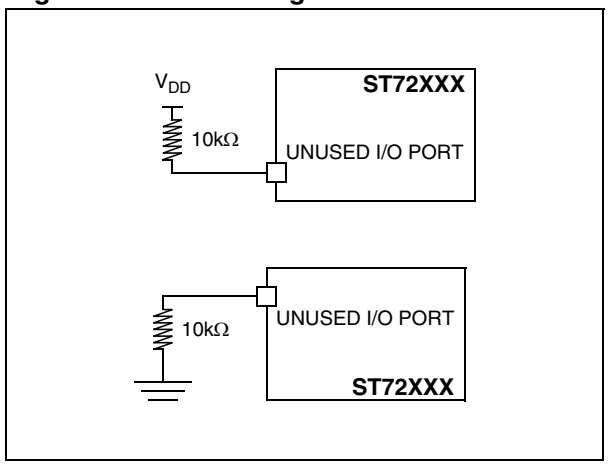



**Figure 126. $R_{PU}$ vs $V_{DD}$ with $V_{IN} = V_{SS}$**

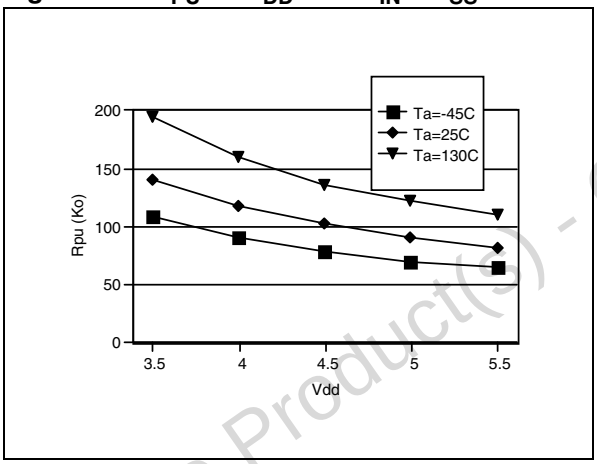
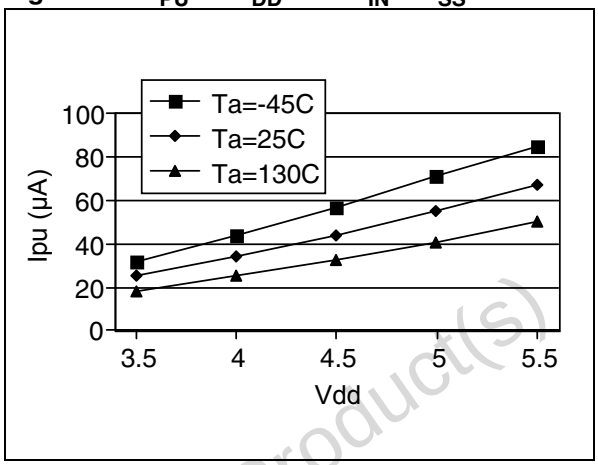



**Figure 127. $I_{PU}$ vs $V_{DD}$ with $V_{IN} = V_{SS}$**



# 15 DEVELOPMENT TOOLS

Full details of tools available for the ST7 from third party manufacturers can be obtained from the STMicroelectronics Internet site:

➟ www.st.com/mcu

Tools from iSystem and Hitex include C compliers, emulators and gang programmers.

**Note:** Before designing the board layout, it is recommended to check the overall dimensions of the socket as they may be greater than the dimensions of the device.

For footprint and other mechanical information about these sockets and adapters, refer to the manufacturer's datasheet.

**ST Programming Tools**

- ST7MDT25-EPB: For in-socket or ICC programming
- ST7-STICK: For ICC programming

**IMPORTANT NOTES** (Cont'd)

### 16.1.5 Header Time-out Does Not Prevent Wake-up from Mute Mode

Normally, when LINSCI is configured in LIN slave mode, if a header time-out occurs during a LIN header reception (that is, header length > 57 bits), the LIN Header Error bit (LHE) is set, an interrupt occurs to inform the application but the LINSCI should stay in mute mode, waiting for the next header reception.

**Problem Description**

The LINSCI sampling period is Tbit / 16. If a LIN Header time-out occurs between the 9th and the 15th sample of the Identifier Field Stop Bit (refer to Figure 153), the LINSCI wakes up from mute mode. Nevertheless, LHE is set and LIN Header Detection Flag (LHDF) is kept cleared.

In addition, if LHE is reset by software before this 15th sample (by accessing the SCISR register and reading the SCIDR register in the LINSCI interrupt routine), the LINSCI will generate another LINSCI interrupt (due to the RDRF flag setting).

**Impact on application**

Software may execute the interrupt routine twice after header reception.

Moreover, in reception mode, as the receiver is no longer in mute mode, an interrupt will be generated on each data byte reception.

**Workaround**

The problem can be detected in the LINSCI interrupt routine. In case of timeout error (LHE is set and LHLR is loaded with 00h), the software can check the RWU bit in the SCICR2 register. If RWU is cleared, it can be set by software. Refer to Figure 154 on page 261. Workaround is shown in bold characters.

**Figure 153. Header Reception Event Sequence**