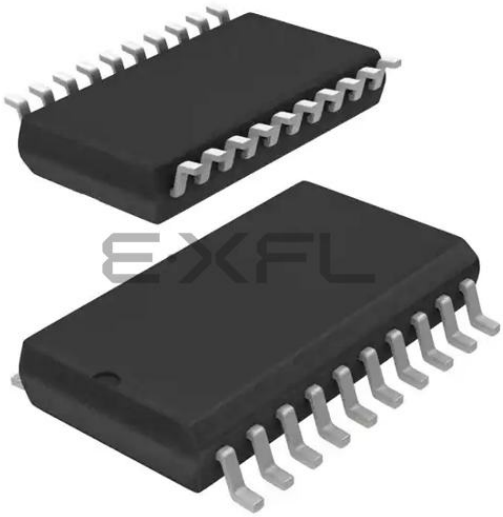


**Welcome to E-XFL.COM**

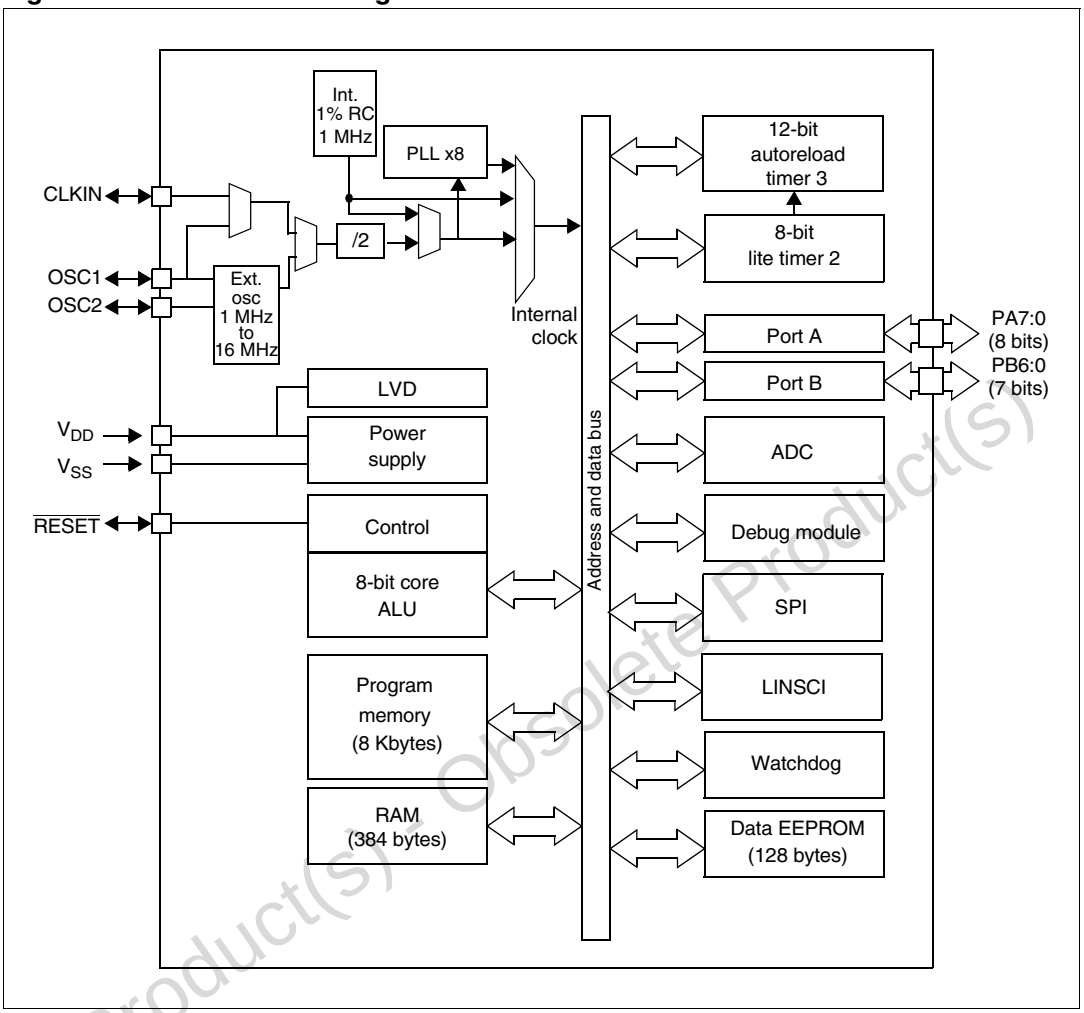
## What is "Embedded - Microcontrollers"?

"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 | LINbusSCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 15 |
| Program Memory Size | 8KB (8K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 384 x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 5.5V |
| Data Converters | A/D 7x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 20-SO |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/st7fl34f2mae |

**Figure 1. General block diagram**



# 6 Central processing unit

## 6.1 Introduction

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

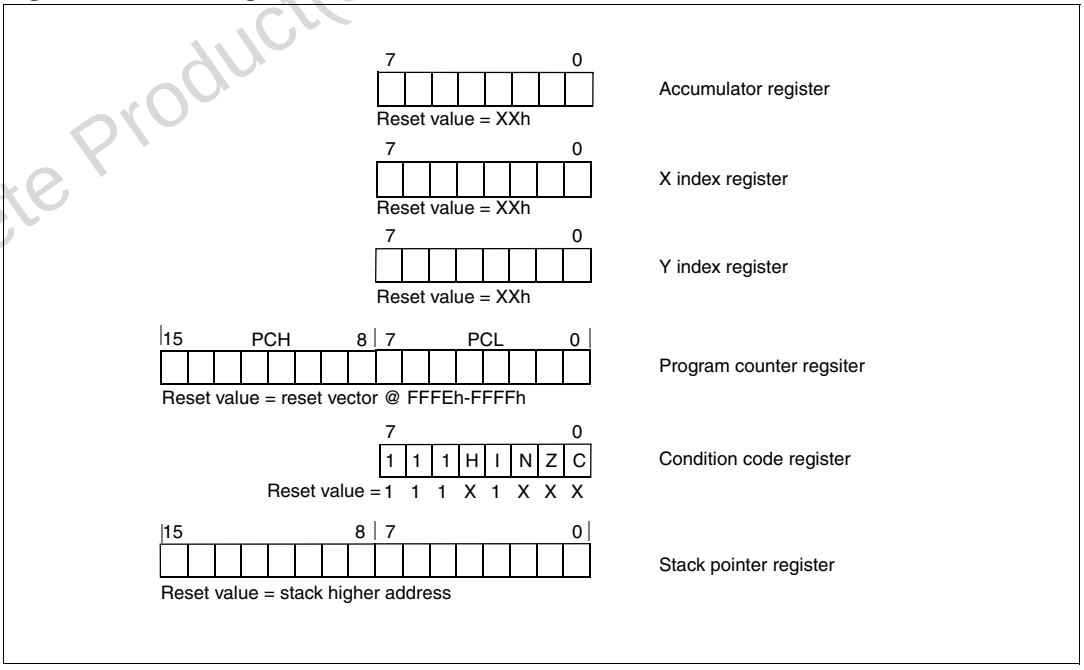
## 6.2 Main features

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

## 6.3 CPU registers

The six CPU registers shown in *Figure 10* are not present in the memory mapping and are accessed by specific instructions.

**Figure 10. CPU registers**




1. X = undefined value

### External interrupt control register (EICR)

EICR Reset value: 0000 0000 (00h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| IS3[1:0] | | IS2[1:0] | | IS1[1:0] | | IS0[1:0] | |
| R/W | | R/W | | R/W | | R/W | |

**Table 15. EICR register description**

| Bit | Bit name | Function |
|---|---|---|
| 7:6 | IS3[1:0] | ei3 sensitivity<br>These bits define the interrupt sensitivity for ei3 (port B0) according to *Table 16* |
| 5:4 | IS2[1:0] | ei2 sensitivity<br>These bits define the interrupt sensitivity for ei2 (port B3) according to *Table 16* |
| 3:2 | IS1[1:0] | ei1 sensitivity<br>These bits define the interrupt sensitivity for ei1 (port A7) according to *Table 16* |
| 1:0 | IS0[1:0] | ei0 sensitivity<br>These bits define the interrupt sensitivity for ei0 (port A0) according to *Table 16* |

*Note: 1 These 8 bits can be written only when the I bit in the CC register is set.*

**Table 16. Interrupt sensitivity bits**

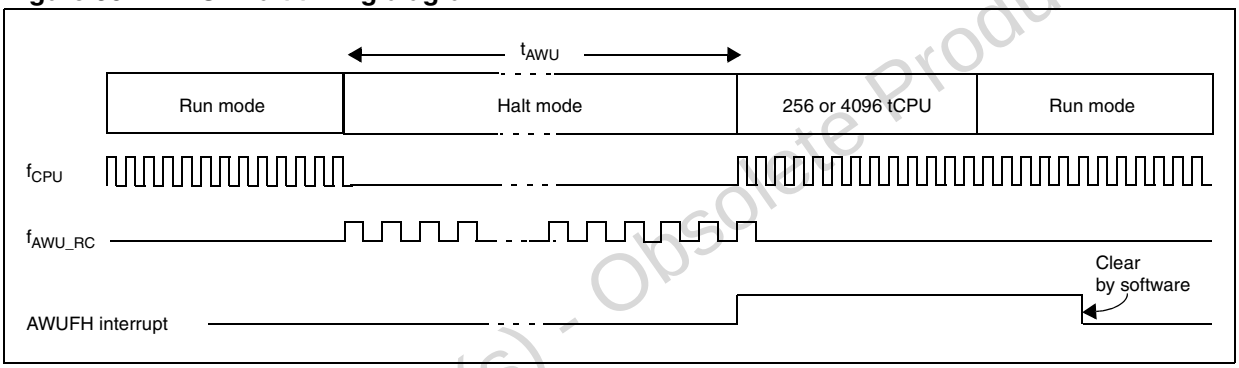
| ISx1 | ISx0 | External interrupt sensitivity |
|---|---|---|
| 0 | 0 | Falling edge and low level |
| 0 | 1 | Rising edge only |
| 1 | 0 | Falling edge only |
| 1 | 1 | Rising and falling edge |

### Similarities with halt mode

The following AWUFH mode behavior is the same as normal halt mode:

- The MCU can exit AWUFH mode by means of any interrupt with exit from halt capability or a reset (see *Section 9.4: Halt mode on page 59*).
- When entering AWUFH mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.
- In AWUFH mode, the main oscillator is turned off, stopping all internal processing, including the operation of the on-chip peripherals. None of the peripherals are clocked except those which receive their clock supply from another clock generator (such as an external or auxiliary oscillator like the AWU oscillator).
- The compatibility of watchdog operation with AWUFH mode is configured by the WDGHALT option bit in the option byte. Depending on this setting, the HALT instruction, when executed while the Watchdog system is enabled, can generate a watchdog reset.

**Figure 30. AWUF halt timing diagram**



### 11.1.6 Register description

#### Watchdog control register (WDGCR)

WDGCR Reset value: 0111 1111 (7Fh)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WDGA | T[6:0] | | | | | | |
| R/W | R/W | | | | | | |

**Table 32. WDGCR register description**

| Bit | Bit name | Function |
|---|---|---|
| 7 | WDGA | Activation bit(1)<br>This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.<br>0: Watchdog disabled<br>1: Watchdog enabled |
| 6:0 | T[6:0] | 7-bit counter (MSB to LSB)<br>These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared). |

1. The WDGA bit is not used if the hardware watchdog option is enabled by option byte.

**Table 33. Watchdog timer register map and reset values**

| Address (Hex.) | Register label | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 002Eh | WDGCR<br>Reset value | WDGA<br>0 | T6<br>1 | T5<br>1 | T4<br>1 | T3<br>1 | T2<br>1 | T1<br>1 | T0<br>1 |

**Table 34. Effect of low power modes on AT3 timer (continued)**

| | |
|---|---|
| Active halt | AT timer halted except if CK0 = 1, CK1 = 0 and OVFIE = 1 |
| Halt | AT timer halted |

### 11.2.5 Interrupts

**Table 35. AT3 interrupt control/wake-up capability**

| Interrupt event[1] | Event flag | Enable control bit | Exit from wait | Exit from Halt | Exit from active halt |
|---|---|---|---|---|---|
| Overflow event | OVF1 | OVFIE1 | Yes | No | Yes[2] |
| AT3 IC event | ICF | ICIE | | | No |
| CMP event | CMPFx | CMPIE | | | |

1. The CMP and AT3 IC events are connected to the same interrupt vector. The OVF event is mapped on a separate vector (see *Section 8: Interrupts*). They generate an interrupt if the enable bit is set in the ATCSR register and the interrupt mask in the CC register is reset (RIM instruction).
2. Only if CK0 = 1 and CK1 = 0 ($f_{COUNTER} = f_{LTIMER}$)

### 11.2.6 Register description

#### Timer control status register (ATCSR)

ATCSR Reset value: 0x00 0000 (x0h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | ICF | ICIE | CK[1:0] | | OVF1 | OVFIE1 | CMPIE |
| - | R/W | R/W | R/W | | R/W | R/W | R/W |

**Table 36. ATCSR register description**

| Bit | Bit name | Function |
|---|---|---|
| 7 | - | Reserved, must be kept cleared |
| 6 | ICF | Input capture flag<br>This bit is set by hardware and cleared by software by reading the ATICR register (a read access to ATICRH or ATICRL clears this flag). Writing to this bit does not change the bit value.<br>0: No input capture<br>1: An input capture has occurred |
| 5 | ICIE | IC interrupt enable<br>This bit is set and cleared by software.<br>0: Input capture interrupt disabled<br>1: Input capture interrupt enabled |

**Table 53. LTCSR1 register description (continued)**

| Bit | Bit name | Function |
|---|---|---|
| 3 | TB1F | Timebase interrupt flag<br>This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect.<br>0: No counter overflow<br>1: A counter overflow has occurred |
| 2:0 | - | Reserved, must be kept cleared |

## Lite timer input capture register (LTICR)

LTICR Reset value: 0000 0000 (00h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ICR[7:0] | | | | | | | |
| R | | | | | | | |

**Table 54. LTICR register description**

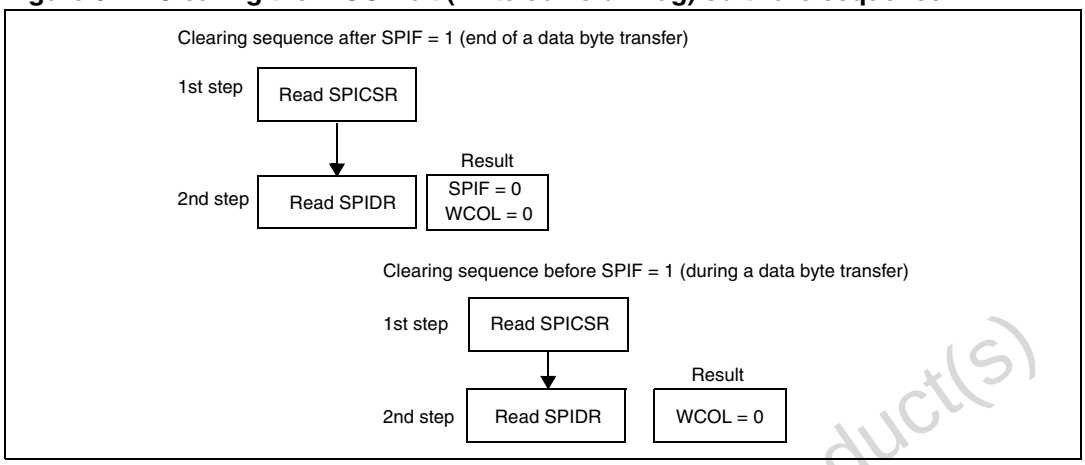
| Bit | Bit name | Function |
|---|---|---|
| 7:0 | ICR[7:0] | Input capture value<br>These bits are read by software and cleared by hardware after a reset. If the ICF bit in the LTCSR is cleared, the value of the 8-bit up-counter is captured when a rising or falling edge occurs on the LTIC pin. |

**Table 55. Lite timer register map and reset values**

| Address (Hex.) | Register label | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 08 | LTCSR2<br>Reset value | | | | | | | TB2IE<br>0 | TB2F<br>0 |
| 09 | LTARR<br>Reset value | AR7<br>0 | AR6<br>0 | AR5<br>0 | AR4<br>0 | AR3<br>0 | AR2<br>0 | AR1<br>0 | AR0<br>0 |
| 0A | LTCNTR<br>Reset value | CNT7<br>0 | CNT6<br>0 | CNT5<br>0 | CNT4<br>0 | CNT3<br>0 | CNT2<br>0 | CNT1<br>0 | CNT0<br>0 |
| 0B | LTCSR1<br>Reset value | ICIE<br>0 | ICF<br>x | TB<br>0 | TB1IE<br>0 | TB1F<br>0 | 0 | 0 | 0 |
| 0C | LTICR<br>Reset value | ICR7<br>0 | ICR6<br>0 | ICR5<br>0 | ICR4<br>0 | ICR3<br>0 | ICR2<br>0 | ICR1<br>0 | ICR0<br>0 |

Clearing the WCOL bit is done through a software sequence (see *Figure 54*).

**Figure 54. Clearing the WCOL bit (write collision flag) software sequence**




1. Writing to the SPIDR register instead of reading it does not reset the WCOL bit.

## Single master and multimaster configurations

There are two types of SPI systems:

- Single master system
- Multimaster system

Single Master System

A typical single master system may be configured using a device as the master and four devices as slaves (see *Figure 55: Single master/multiple slave configuration on page 117*).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

*Note: To prevent a bus conflict on the MISO line, the master allows only one active slave device during a transmission.*

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written to its SPIDR register.

Other transmission security methods can use ports for handshake lines or data bytes with command fields.

### Multimaster system

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

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

### Conventional baud rate generation

The baud rates for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(16*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(16*PR)*RR}$$

where:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits)

TR = 1, 2, 4, 8, 16, 32, 64,128 (see SCT[2:0] bits)

RR = 1, 2, 4, 8, 16, 32, 64,128 (see SCR[2:0] bits)

All these bits are in the *Baud rate register (SCIBRR) on page 139*.

Example: If $f_{CPU}$ is 8 MHz (normal mode) and if PR = 13 and TR = RR = 1, the transmit and receive baud rates are 38400 baud.

*Note: The baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.*

### Extended baud rate generation

The extended prescaler option gives a very fine tuning on the baud rate, using a 255 value prescaler, whereas the conventional baud rate generator retains industry standard software compatibility.

The extended baud rate generator block diagram is described in *Figure 58: SCI baud rate and extended prescaler block diagram on page 131*.

The output clock rate sent to the transmitter or to the receiver will be the output from the 16 divider divided by a factor ranging from 1 to 255 set in the SCIERPR or the SCIETPR register.

*Note: The extended prescaler is activated by setting the SCIETPR or SCIERPR register to a value other than zero.*

The baud rates are calculated as follows:

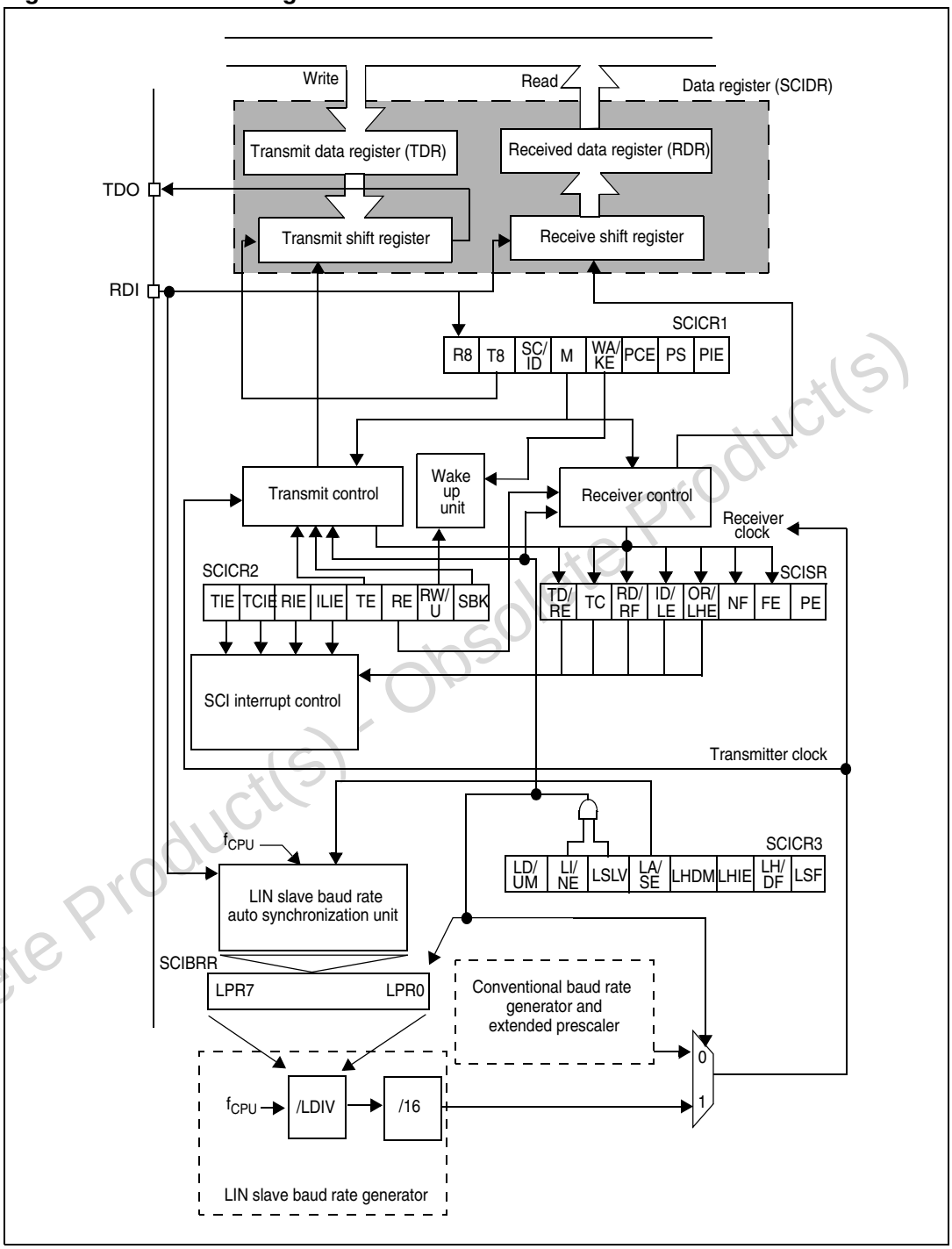
$$Tx = \frac{f_{CPU}}{16*ETPR*(PR*TR)} \qquad Rx = \frac{f_{CPU}}{16*ERPR*(PR*RR)}$$

where:

ETPR = 1, ..., 255 (see SCIETPR register *on page 140*)

ERPR = 1, ..., 255 (see SCIERPR register *on page 140*)

**Figure 60. SCI block diagram in LIN slave mode**






## LIN reception

In LIN mode the reception of a byte is the same as in SCI mode but the LINSCI has features for handling the LIN header automatically (identifier detection) or semi-automatically (synch break detection) depending on the LIN header detection mode. The detection mode is selected by the LHDM bit in the SCICR3.

# 13 Electrical characteristics

## 13.1 Parameter conditions

Unless otherwise specified, all voltages are referred to $V_{SS}$.

### 13.1.1 Minimum and maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at $T_A$ = 25 °C and $T_A$ = $T_A$max (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean ± 3σ).

### 13.1.2 Typical values

Unless otherwise specified, typical data are based on $T_A$ = 25 °C, $V_{DD}$ = 5 V (for the 4.5 V ≤ $V_{DD}$ ≤ 5.5 V voltage range) and $V_{DD}$ = 3.3 V (for the 3 V ≤ $V_{DD}$ ≤ 3.6 V voltage range). They are given only as design guidelines and are not tested.

### 13.1.3 Typical curves

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

### 13.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 70*.

**Figure 70. Pin loading conditions**

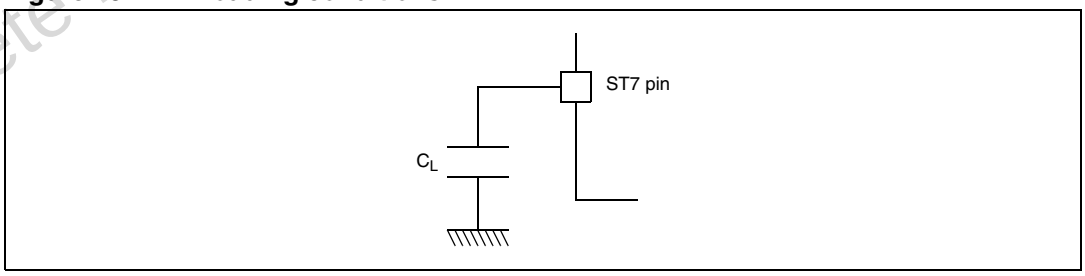


*Figure 76. $f_{RC}$ vs. $V_{DD}$ and temperature for calibrated RCCR1*

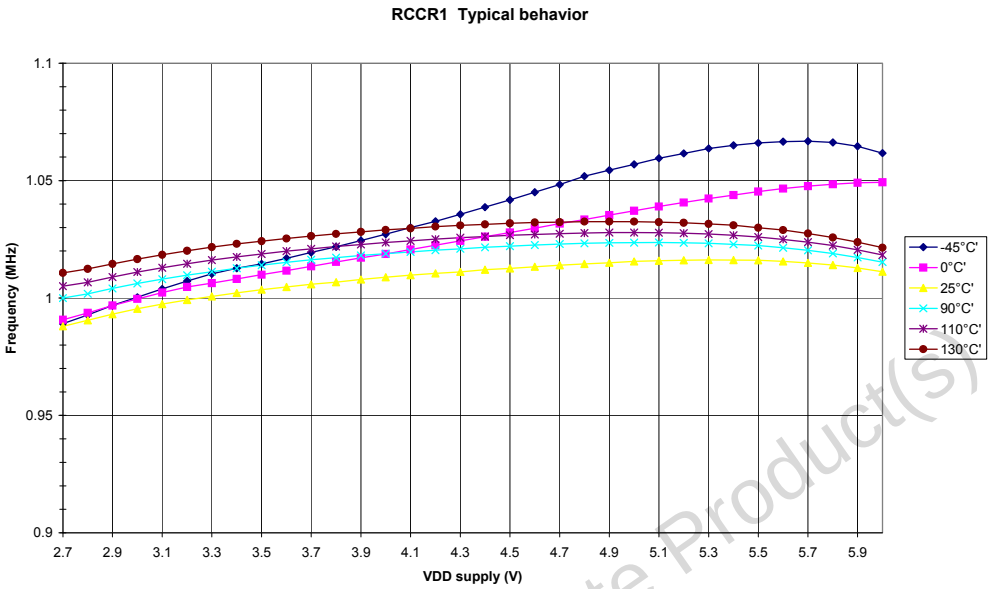
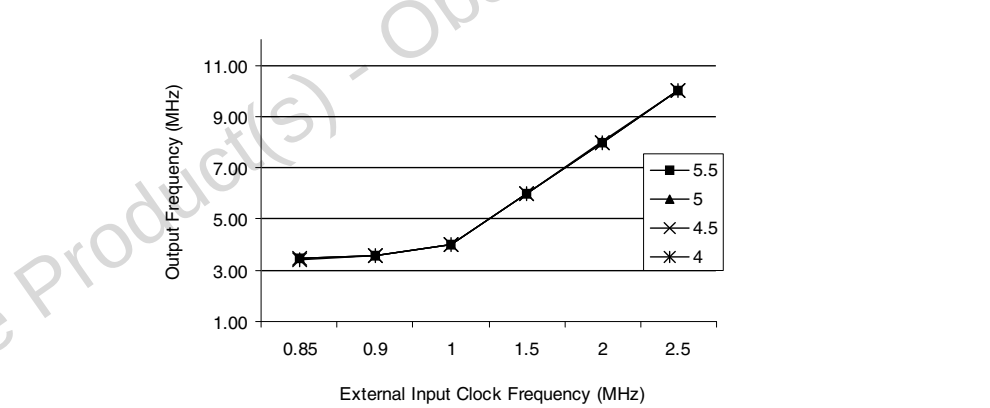



**Figure 77. PLL x 8 output vs. CLKIN frequency**




1. $f_{OSC} = f_{CLKIN}/2*PLL8$

## 13.3.2 Operating conditions with low voltage detector (LVD)

$T_A$ = -40 to +125 °C, unless otherwise specified

**Table 105. Operating conditions with low voltage detector**

| Symbol | Parameter | Conditions(1) | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{IT+(LVD)}$ | Reset release threshold ($V_{DD}$ rise) | High threshold | 3.60(2) | 4.15 | 4.50 | V |
| $V_{IT-(LVD)}$ | Reset generation threshold ($V_{DD}$ fall) | High threshold | 3.40 | 3.95 | 4.40(2) | |

## 13.4 Supply current characteristics

The following 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 for which the clock is stopped).

### 13.4.1 Supply current

$T_A$ = -40 to +125°C, unless otherwise specified

**Table 108. Supply current**

| Symbol | Parameter | Conditions | | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Supply current in run mode | (1) | $f_{CPU}$ = 8 MHz(2) | 6 | 9 | mA |
| | Supply current in wait mode | | $f_{CPU}$ = 8 MHz(3) | 2.4 | 4 | |
| | Supply current in slow mode | | $f_{CPU}$ = 250 kHz(4) | 0.7 | 1.1 | |
| | Supply current in slow wait mode | | $f_{CPU}$ = 250 kHz(5) | 0.6 | 1 | |
| $I_{DD}$ | Supply current in halt mode(6) | (1) | -40°C ≤ $T_A$ ≤ +85°C | <1 | 10 | µA |
| | | | -40°C ≤ $T_A$ ≤ +125°C | | 50 | |
| $I_{DD}$ | Supply current in AWUFH mode(7)(8)(9) | (1) | -40°C ≤ $T_A$ ≤ +85°C | 20 | 50 | µA |
| | | | -40°C ≤ $T_A$ ≤ +125°C | | 300 | |
| $I_{DD}$ | Supply current in active halt mode | (1) | -40°C ≤ $T_A$ ≤ +125°C | 0.7 | 1 | mA |

1. $V_{DD}$ = 5.5 V
2. CPU running with memory access, all I/O pins in input mode with a static value at $V_{DD}$ or $V_{SS}$ (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
3. All I/O pins in input mode with a static value at $V_{DD}$ or $V_{SS}$ (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
4. Slow mode selected with $f_{CPU}$ based on $f_{OSC}$ divided by 32. All I/O pins in input mode with a static value at $V_{DD}$ or $V_{SS}$ (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
5. Slow-wait mode selected with $f_{CPU}$ based on $f_{OSC}$ divided by 32. All I/O pins in input mode with a static value at $V_{DD}$ or $V_{SS}$ (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
6. All I/O pins in output mode with a static value at $V_{SS}$ (no load), LVD disabled. Data based on characterization results, tested in production at $V_{DD}$ max. and $f_{CPU}$ max.
7. All I/O pins in input mode with a static value at $V_{DD}$ or $V_{SS}$ (no load). Data tested in production at $V_{DD}$ max. and $f_{CPU}$ max.
8. This consumption refers to the Halt period only and not the associated run period which is software dependent.
9. If low consumption is required, AWUFH mode is recommended.

## 13.8 I/O port pin characteristics

### 13.8.1 General characteristics

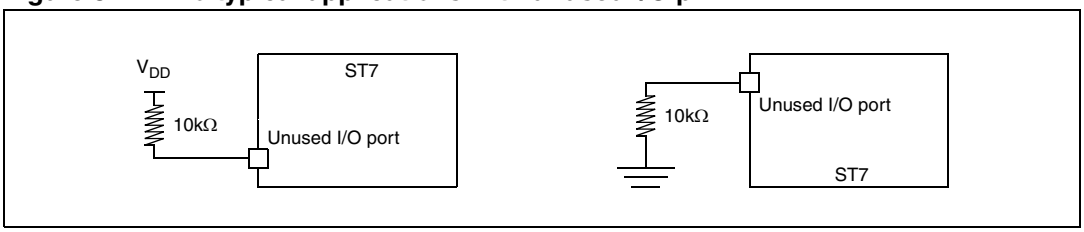
Subject to general operating conditions for $V_{DD}$, $f_{OSC}$, and $T_A$ (-40 to +125°C), unless otherwise specified.

**Table 120. I/O general port pin characteristics**

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{IL}$ | Input low level voltage | | $V_{SS}$ - 0.3 | | 0.3 x $V_{DD}$ | V |
| $V_{IH}$ | Input high level voltage | | 0.7 x $V_{DD}$ | | $V_{DD}$ + 0.3 | |
| $V_{hys}$ | Schmitt trigger voltage hysteresis(1) | | | 400 | | mV |
| $I_L$ | Input leakage current | $V_{SS} \le V_{IN} \le V_{DD}$ | | | ±1 | µA |
| $I_S$ | Static current consumption induced by each floating input pin(2) | Floating input mode | | 400 | | |
| $R_{PU}$ | Weak pull-up equivalent resistor(3) | $V_{IN} = V_{SS}$, $V_{DD}$ = 5 V | 50 | 100 | 170 | kΩ |
| $C_{IO}$ | I/O pin capacitance | | | 5 | | pF |
| $t_{f(IO)out}$ | Output high to low level fall time(1) | $C_L$ = 50 pF between 10% and 90% | | 25 | | ns |
| $t_{r(IO)out}$ | Output low to high level rise time(1) | | | | | |
| $t_{w(IT)in}$ | External interrupt pulse time(4) | | 1 | | | $t_{CPU}$ |

1. Data based on characterization results, not tested in production
2. Configuration not recommended, all unused pins must be kept at a fixed voltage: Using the output mode of the I/O for example or an external pull-up or pull-down resistor (see *Figure 84*). Static peak current value taken at a fixed $V_{IN}$ value, based on design simulation and technology characteristics, not tested in production. This value depends on $V_{DD}$ and temperature values
3. The $R_{PU}$ pull-up equivalent resistor is based on a resistive transistor (corresponding $I_{PU}$ current characteristics described in *Figure 84: Two typical applications with unused I/O pin on page 199*)
4. To generate an external interrupt, a minimum pulse width must be applied on an I/O port pin configured as an external interrupt source.

**Figure 84. Two typical applications with unused I/O pin**




1. I/O can be left unconnected if it is configured as output (0 or 1) by the software. This has the advantage of greater EMC robustness and lower cost.

**Figure 86. Typical $V_{OL}$ at $V_{DD}$ = 3 V**

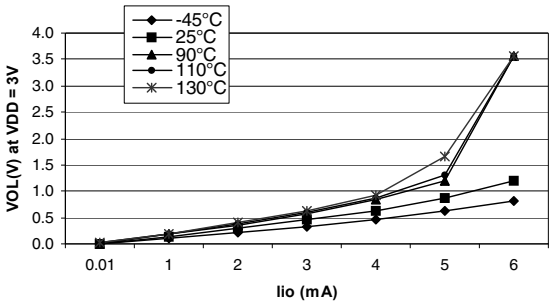



**Figure 87. Typical $V_{OL}$ at $V_{DD}$ = 4 V**

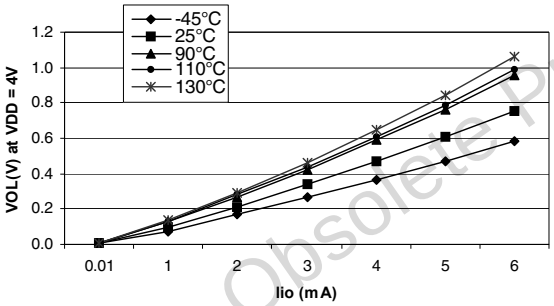



**Figure 88. Typical $V_{OL}$ at $V_{DD}$ = 5 V**

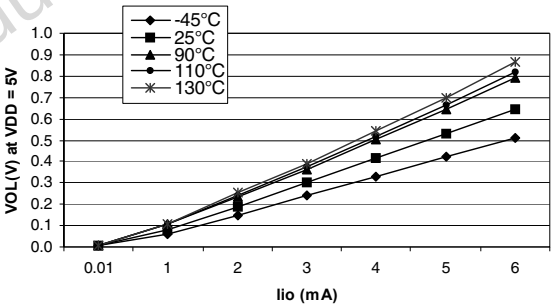
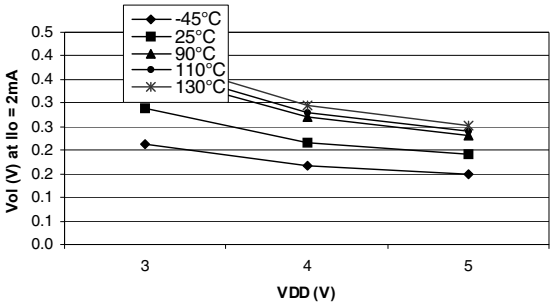


**Figure 95. Typical $V_{OL}$ vs. $V_{DD}$ (standard I/Os)**




**Typical $V_{OL}$ vs. $V_{DD}$ (standard I/Os)**

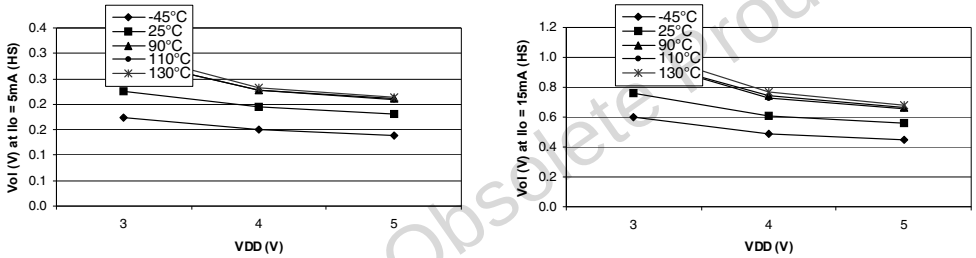
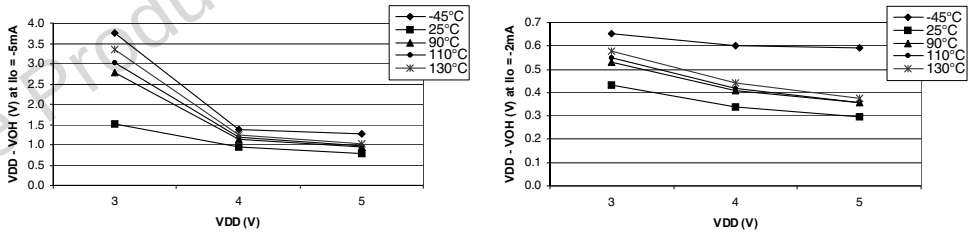



**Figure 96. Typical $V_{DD}$ - $V_{OH}$ vs. $V_{DD}$**



# 15 Device configuration and ordering information

## 15.1 Introduction

Each device is available for production in user programmable versions (Flash) as well as in factory coded versions (ROM). ST7L3x devices are ROM versions.

ST7PL3x devices are factory advanced service technique ROM (FASTROM) versions: They are factory programmed Flash devices.

ST7FL3 Flash devices are shipped to customers with a default program memory content (FFh), while 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.

## 15.2 Option bytes

The two option bytes allow the hardware configuration of the microcontroller to be selected. Differences in option byte configuration between Flash and ROM devices are presented in *Table 130* and are described in *Section 15.2.1: Flash option bytes on page 216* and *Section 15.2.2: ROM option bytes on page 217*.

**Table 130. Flash and ROM option bytes**

| | | Option byte 0 | | | | | | | | Option byte 1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | Flash | AW UCK | OSCRANGE 2:0 | | | SEC 1 | SEC 0 | FM PR | FM PW | Res | PLL OFF | Res | OSC | LVD 1:0 | | WDG SW | WDG HALT |
| | ROM | | | | | Res | | ROP _R | ROP _D | | | | | | | | |
| Default value | | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1(1) | 1 | 0(1) | 0 | 1 | 1 | 1 | 1 |

1. Contact your STMicroelectronics support