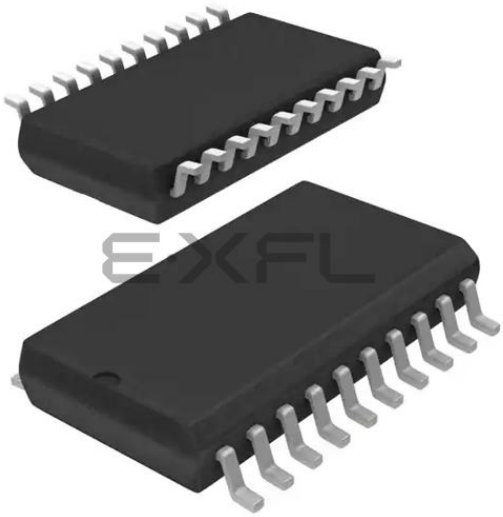


**Welcome to E-XFL.COM**

## 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 | 256 x 8 |
| 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/st7fl38f2mae |

# List of tables

# 1 Description

The ST7L3x is a member of the ST7 microcontroller family suitable for automotive applications. All ST7 devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

The ST7L3x features Flash memory with byte-by-byte in-circuit programming (ICP) and in-application programming (IAP) capability.

Under software control, the ST7L3x devices can be placed in wait, slow or halt mode, reducing power consumption when the application is in idle or standby state.

The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

**Table 1. Device summary**

| Feature | ST7L34 | ST7L35 | ST7L38 | ST7L39 |
|---|---|---|---|---|
| Program memory | 8 Kbytes | | | |
| RAM (stack) | 384 bytes (128 bytes) | | | |
| Data EEPROM | - | | 256 bytes | |
| Peripherals | Lite timer, autoreload timer, SPI, 10-bit ADC | Lite timer, autoreload timer, SPI, 10-bit ADC, LINSCI | Lite timer, autoreload timer, SPI, 10-bit ADC | Lite timer, autoreload timer, SPI, 10-bit ADC, LINSCI |
| Operating supply | 3.0 V to 5.5 V | | | |
| CPU frequency | Up to 8 MHz (with external resonator/clock or internal RC oscillator) | | | |
| Operating temperature | Up to -40 to 85°C / -40 to 125°C | | | |
| Packages | SO20 300mil, QFN20 | | | |



## 1.1 Parametric data

For easy reference, all parametric data is located in *Section 13: Electrical characteristics*.

## 1.2 Debug module (DM)

The devices feature an on-chip debug module (DM) to support in-circuit debugging (ICD). For a description of the DM registers, refer to the *ST7 ICC protocol reference manual*.

# 5 Data EEPROM

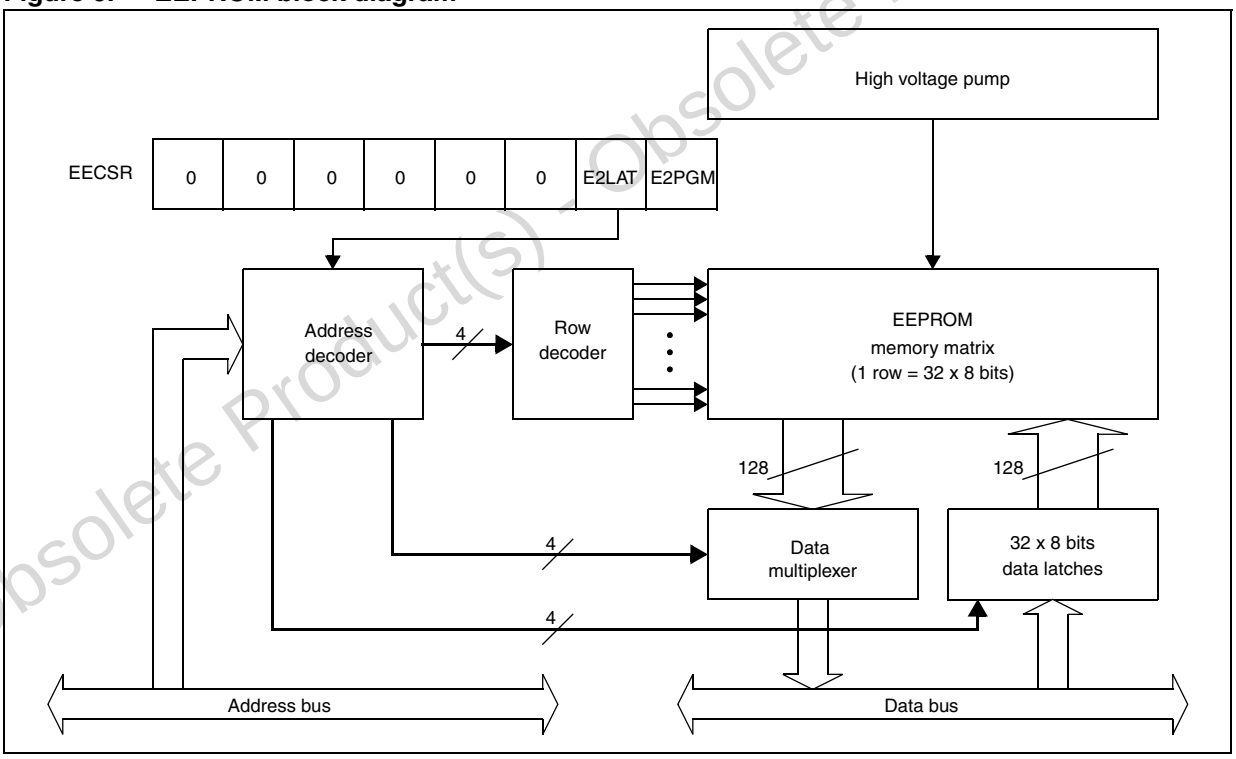
## 5.1 Introduction

The electrically erasable programmable read only memory can be used as a non volatile back-up for storing data. Using the EEPROM requires a basic access protocol described in this chapter.

## 5.2 Main features

- Up to 32 bytes programmed in the same cycle
- EEPROM mono-voltage (charge pump)
- Chained erase and programming cycles
- Internal control of the global programming cycle duration
- Wait mode management
- Readout protection

**Figure 6. EEPROM 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.

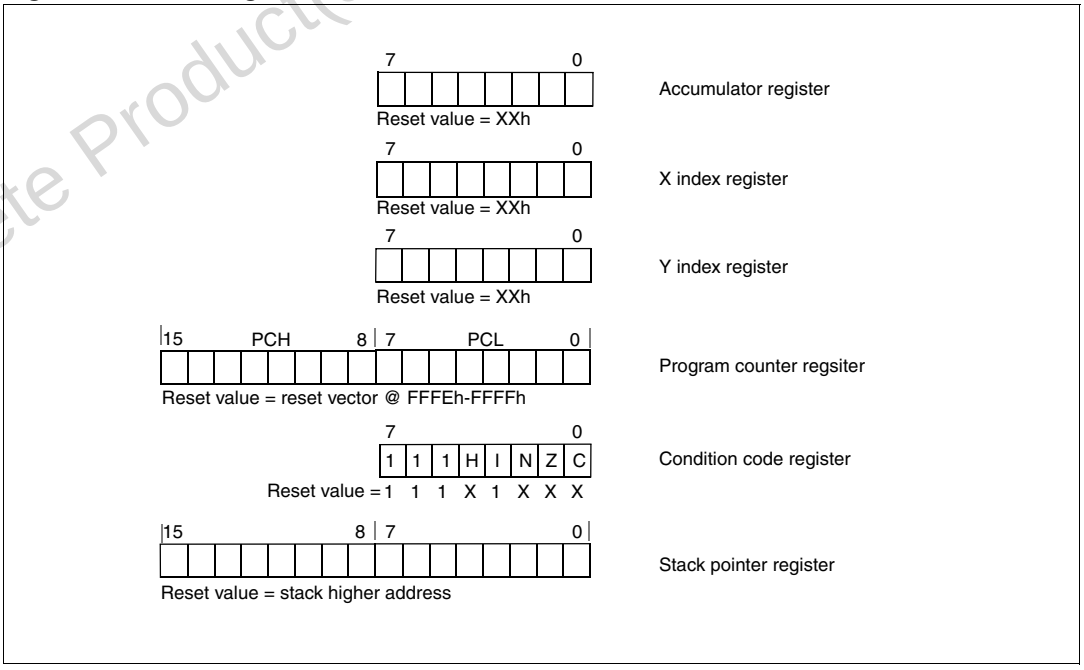
## 6.2 Main features

- 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

## 7.4 Multi-oscillator (MO)

The main clock of the ST7 can be generated by four different source types coming from the multi-oscillator block (1 to 16 MHz or 32 kHz):

- An external source
- 5 crystal or ceramic resonator oscillators
- An internal high frequency RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configurations are shown in *Table 14: ST7 clock sources on page 42*. Refer to *Section 13: Electrical characteristics* for more details.

### 7.4.1 External clock source

In external clock mode, a clock signal (square, sinus or triangle) with ~50% duty cycle must drive the OSC1 pin while the OSC2 pin is tied to ground.

*Note: When the multi-oscillator is not used, PB4 is selected by default as the external clock.*

### 7.4.2 Crystal/ceramic oscillators

This family of oscillators has the advantage of producing a very accurate rate on the main clock of the ST7. The selection within a list of four oscillators with different frequency ranges has to be done by option byte in order to reduce consumption (refer to *Section 15.2 on page 215* for more details on the frequency ranges). In this mode of the multi-oscillator, the resonator and the load capacitors must be placed as close as possible to the oscillator pins to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

These oscillators are not stopped during the reset phase to avoid losing time in the oscillator startup phase.

### 7.4.3 Internal RC oscillator

In this mode, the tunable 1%RC oscillator is the main clock source. The two oscillator pins must be tied to ground.

The calibration is done through the RCCR[7:0] and SICSR[6:5] registers.

## 8.2 External interrupts

External interrupt vectors can be loaded into the PC register if the corresponding external interrupt occurred and if the I bit is cleared. These interrupts allow the processor to leave the halt low power mode.

The external interrupt polarity is selected through the miscellaneous register or interrupt register (if available).

An external interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.

**Caution:** The type of sensitivity defined in the miscellaneous or interrupt register (if available) applies to the *ei* source. In case of a NANDed source (as described in *Section 10: I/O ports*), a low level on an I/O pin, configured as input with interrupt, masks the interrupt request even in case of rising-edge sensitivity.

## 8.3 Peripheral interrupts

Different peripheral interrupt flags in the status register are able to cause an interrupt when they are active if both the following conditions are met:

- The I bit of the CC register is cleared
- The corresponding enable bit is set in the control register

If either of these two conditions is false, the interrupt is latched and thus remains pending.

Clearing an interrupt request is done by:

- Writing ‘0’ to the corresponding bit in the status register or
- Access to the status register while the flag is set followed by a read or write of an associated register.

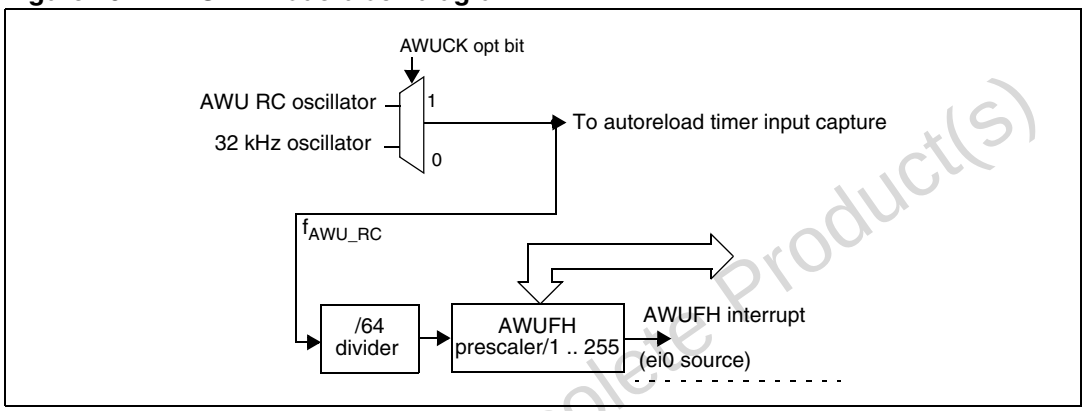
*Note: The clearing sequence resets the internal latch. A pending interrupt (that is, waiting for being enabled) will therefore be lost if the clear sequence is executed.*

## 9.6 Auto wakeup from halt mode

Auto wakeup from halt (AWUFH) mode is similar to halt mode with the addition of a specific internal RC oscillator for wakeup (auto wakeup from halt oscillator). Compared to active halt mode, AWUFH has lower power consumption (the main clock is not kept running but there is no accurate real-time clock available).

It is entered by executing the HALT instruction when the AWUEN bit in the AWUCSR register has been set.

**Figure 29. AWUFH mode block diagram**




As soon as halt mode is entered and if the AWUEN bit has been set in the AWUCSR register, the AWU RC oscillator provides a clock signal ($f_{AWU\_RC}$). Its frequency is divided by a fixed divider and a programmable prescaler controlled by the AWUPR register. The output of this prescaler provides the delay time. When the delay has elapsed, the AWUF flag is set by hardware and an interrupt wakes up the MCU from halt mode. At the same time, the main oscillator is immediately turned on and a 256-cycle delay is used to stabilize it. After this startup delay, the CPU resumes operation by servicing the AWUFH interrupt. The AWU flag and its associated interrupt are cleared by software reading the AWUCSR register.

To compensate for any frequency dispersion of the AWU RC oscillator, it can be calibrated by measuring the clock frequency $f_{AWU\_RC}$ and then calculating the right prescaler value. Measurement mode is enabled by setting the AWUM bit in the AWUCSR register in run mode. This connects $f_{AWU\_RC}$ to the input capture of the 12-bit autoreload timer, allowing the $f_{AWU\_RC}$ to be measured using the main oscillator clock as a reference timebase.

### 10.2.2 Output modes

Setting the DDRx bit selects output mode. Writing to the DR bits applies a digital value to the I/O through the latch. Reading the DR bits returns the previously stored value.

If an OR bit is available, different output modes can be selected by software: Push-pull or open-drain. Refer to *Section 10.3: I/O port implementation on page 72* for configuration.

**Table 23. DR value and output pin status**

| DR | Push-pull | Open-drain |
|---|---|---|
| 0 | $V_{OL}$ | $V_{OL}$ |
| 1 | $V_{OH}$ | Floating |

### 10.2.3 Alternate functions

Many ST7 I/Os have one or more alternate functions. These may include output signals from, or input signals to, on-chip peripherals. *Table 2: Device pin description on page 17* describes which peripheral signals can be input/output to which ports.

A signal coming from an on-chip peripheral can be output on an I/O. To do this, enable the on-chip peripheral as an output (enable bit in the peripheral's control register). The peripheral configures the I/O as an output and takes priority over standard I/O programming. The I/O's state is readable by addressing the corresponding I/O data register.

Configuring an I/O as floating enables alternate function input. It is not recommended to configure an I/O as pull-up as this increases current consumption. Before using an I/O as an alternate input, configure it without interrupt. Otherwise spurious interrupts can occur.

Configure an I/O as input floating for an on-chip peripheral signal which can be input and output.

**Caution:** I/Os which can be configured as both an analog and digital alternate function need special attention. The user must control the peripherals so that the signals do not arrive at the same time on the same pin. If an external clock is used, only the clock alternate function should be employed on that I/O pin and not the other alternate function.

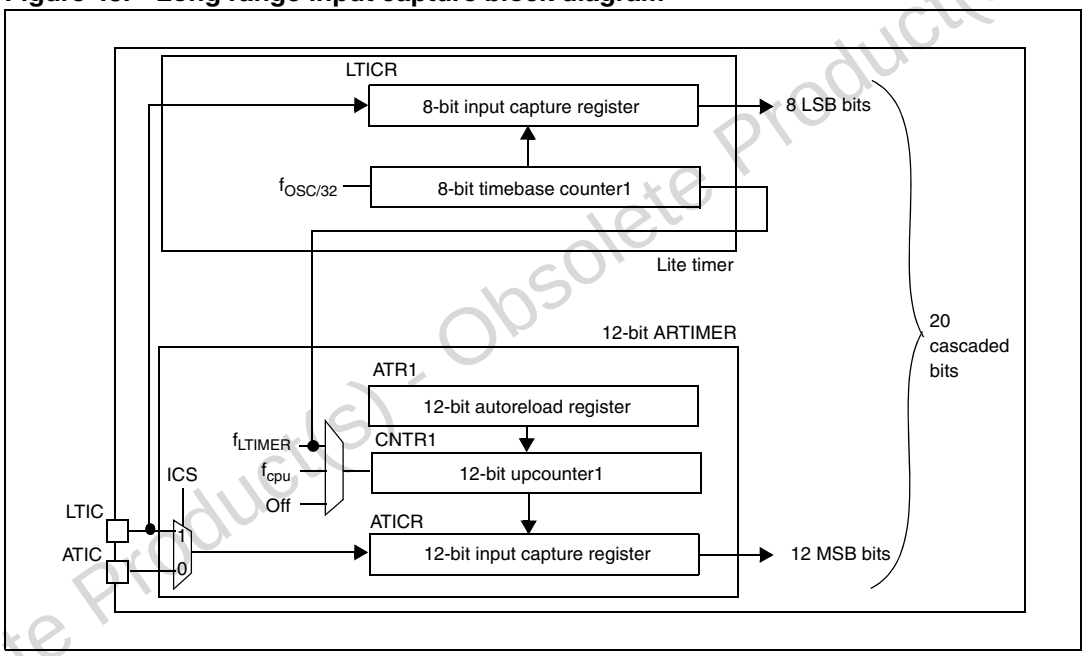
### Long input capture

Pulses that last between 8 µs and 2 s can be measured with an accuracy of 4µs if $f_{OSC}$ = 8 MHz under the following conditions:

- The 12-bit AT3 timer is clocked by the lite timer (RTC pulse: CK[1:0] = 01 in the ATCSR register)
- The ICS bit in the ATCSR2 register is set so that the LTIC pin is used to trigger the AT3 timer capture.
- The signal to be captured is connected to LTIC pin
- Input capture registers LTICR, ATICRH and ATICRL are read

This configuration allows to cascade the lite timer and the 12-bit AT3 timer to get a 20-bit input capture value. Refer to *Figure 45*.

**Figure 45. Long range input capture block diagram**






Since the input capture flags (ICF) for both timers (AT3 timer and LT timer) are set when signal transition occurs, software must mask one interrupt by clearing the corresponding ICIE bit before setting the ICS bit.

If the ICS bit changes (from 0 to 1 or from 1 to 0), a spurious transition might occur on the input capture signal because of different values on LTIC and ATIC. To avoid this situation, it is recommended to do the following:

- First, reset both ICIE bits
- Then set the ICS bit
- Reset both ICF bits
- Then set the ICIE bit of desired interrupt

### PWMx control status register (PWMxCSR)

PWMxCSR Reset value: 0000 0000 (00h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | OPx | CMPFx |
| - | - | - | - | - | - | R/W | R/W |

**Table 40. PWMxCSR register description**

| Bit | Bit name | Function |
|---|---|---|
| 7:2 | - | Reserved, must be kept cleared |
| 1 | OPx | PWMx output polarity<br>This bit is read/write by software and cleared by hardware after a reset. This bit selects the polarity of the PWM signal.<br>0: The PWM signal is not inverted<br>1: The PWM signal is inverted |
| 0 | CMPFx | PWMx *compare flag*<br>This bit is set by hardware and cleared by software by reading the PWMxCSR register. It indicates that the upcounter value matches the active DCRx register value.<br>0: Upcounter value does not match DCRx value<br>1: Upcounter value matches DCRx value |

### Timer control register2 (ATCSR2)

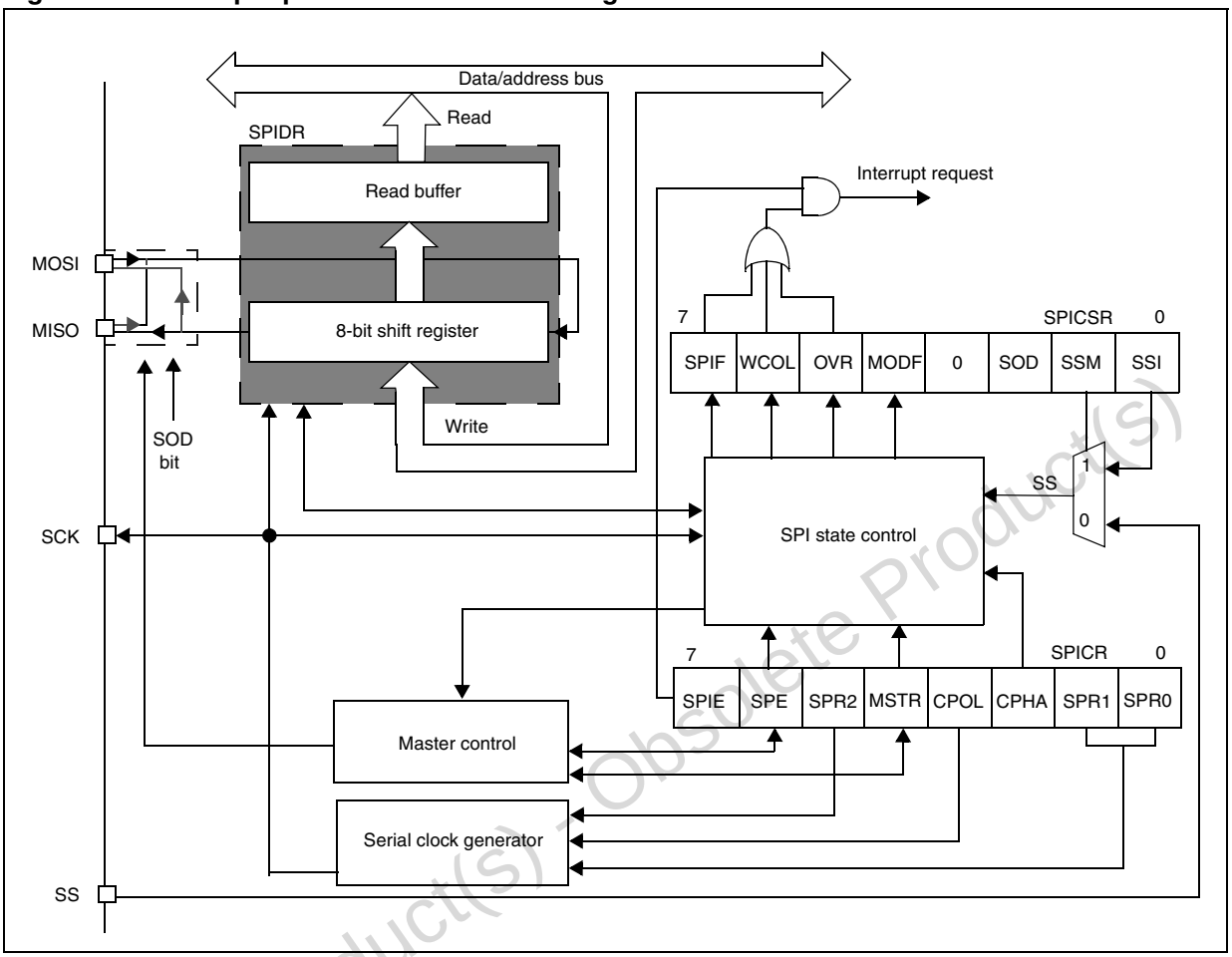
ATCSR2 Reset value: 0000 0011 (03h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | ICS | OVFIE2 | OVF2 | ENCNTR2 | TRAN2 | TRAN1 |
| - | - | R/W | R/W | R/W | R/W | R/W | R/W |

**Table 44. ATCSR2 register description**

| Bit | Bit name | Function |
|---|---|---|
| 7:6 | - | Reserved, must be kept cleared |
| 5 | ICS | Input capture shorted<br>This bit is read/write by software. It allows the AT timer CNTR1 to use the LTIC pin for long input capture.<br>0: ATIC for CNTR1 input capture<br>1: LTIC for CNTR1 input capture |
| 4 | OVFIE2 | Overflow interrupt 2 enable<br>This bit is read/write by software and controls the overflow interrupt of counter 2.<br>0: Overflow interrupt disabled<br>1: Overflow interrupt enabled |
| 3 | OVF2 | Overflow flag<br>This bit is set by hardware and cleared by software by reading the ATCSR2 register. It indicates the transition of the counter 2 from FFFh to ATR2 value.<br>0: No counter overflow occurred<br>1: Counter overflow occurred |
| 2 | ENCNTR2 | Enable counter 2<br>This bit is read/write by software and switches the second counter CNTR2. If this bit is set, PWM2/3 is generated using CNTR2<br>0: CNTR2 stopped<br>1: CNTR2 starts running |
| 1 | TRAN2 | Transfer enable 2<br>This bit is read/write by software, cleared by hardware after each completed transfer and set by hardware after reset. It controls the transfers on CNTR2. It allows the value of the preload DCRx registers to be transferred to the active DCRx registers after the next overflow event. The OPx bits are transferred to the shadow OPx bits in the same way.<br>*Note: Only DCR2/3 can be controlled using this bit* |

**Figure 49. Serial peripheral interface block diagram**






## Functional description

A basic example of interconnections between a single master and a single slave is illustrated in *Figure 50: Single master/single slave application on page 110*.

The MOSI pins are connected together and the MISO pins are connected together. In this way data are 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 53: Data clock timing diagram on page 114*) but master and slave must be programmed with the same timing mode.

**Table 60. SPI register map and reset values**

| Address (Hex.) | Register label | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0031h | SPIDR<br>Reset Value | MSB<br>x | x | x | x | x | x | x | LSB<br>x |
| 0032h | SPICR<br>Reset Value | SPIE<br>0 | SPE<br>0 | SPR2<br>0 | MSTR<br>0 | CPOL<br>x | CPHA<br>x | SPR1<br>x | SPR0<br>x |
| 0033h | SPICSR<br>Reset Value | SPIF<br>0 | WCOL<br>0 | OVR<br>0 | MODF<br>0 | 0 | SOD<br>0 | SSM<br>0 | SSI<br>0 |

## 11.5 LINSCI serial communication interface (LIN master/slave)

### 11.5.1 Introduction

The serial communications interface (SCI) offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The SCI offers a very wide range of baud rates using two baud rate generator systems.

The LIN-dedicated features support the LIN (local interconnect network) protocol for both master and slave nodes.

This chapter is divided into SCI Mode and LIN mode sections. For information on general SCI communications, refer to the SCI mode section. For LIN applications, refer to both the SCI mode and LIN mode sections.

The SCI interrupt events are connected to the same interrupt vector (see *Section 8: Interrupts*).

These events generate an interrupt if the corresponding enable control bit is set and the interrupt mask in the CC register is reset (RIM instruction).

## 11.5.8 SCI mode registers

### Status register (SCISR)

SCISR — Reset value: 1100 0000 (C0h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TDRE | TC | RDRF | IDLE | OR(1) | NF(1) | FE(1) | PE(1) |
| R | R | R | R | R | R | R | R |

1. This bit has a different function in LIN mode, please refer to the LIN mode register description

**Table 64. SCISR register description**

| Bit | Bit name | Function |
|---|---|---|
| 7 | TDRE | Transmit data register empty<br>This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE bit = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register).<br>0: Data is not transferred to the shift register<br>1: Data is transferred to the shift register |
| 6 | TC | Transmission complete<br>This bit is set by hardware when transmission of a character containing data is complete. An interrupt is generated if TCIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register).<br>0: Transmission is not complete<br>1: Transmission is complete<br>*Note: TC is not set after the transmission of a preamble or a break.* |
| 5 | RDRF | Received data ready flag<br>This bit is set by hardware when the content of the RDR register has been transferred to the SCIDR register. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).<br>0: Data are not received<br>1: Received data are ready to be read |

**Table 66. SCICR2 register description (continued)**

| Bit | Bit name | Function |
|---|---|---|
| 3 | TE | Transmitter enable<br>This bit enables the transmitter. It is set and cleared by software.<br>0: Transmitter is disabled<br>1: Transmitter is enabled<br>*Note: During transmission, an '0' pulse on the TE bit ('0' followed by '1') sends a preamble (idle line) after the current word.*<br>*When TE is set there is a 1 bit-time delay before the transmission starts.* |
| 2 | RE | Receiver enable<br>This bit enables the receiver. It is set and cleared by software.<br>0: Receiver is disabled in the SCISR register<br>1: Receiver is enabled and begins searching for a start bit |
| 1 | RWU | Receiver wake up<br>This bit determines if the SCI is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wake up sequence is recognized.<br>0: Receiver in active mode<br>1: Receiver in mute mode<br>*Note: Before selecting mute mode (by setting the RWU bit), the SCI must first receive a data byte, otherwise it cannot function in mute mode with wakeup by idle line detection.*<br>*In address mark detection wake up configuration (WAKE bit = 1) the RWU bit cannot be modified by software while the RDRF bit is set.* |
| 0 | SBK | Send break<br>This bit set is used to send break characters. It is set and cleared by software.<br>0: No break character is transmitted<br>1: Break characters are transmitted<br>*Note: If the SBK bit is set to '1' and then to '0', the transmitter sends a BREAK word at the end of the current word.* |



### Data register (SCIDR)

Contains the received or transmitted data character, depending on whether it is read from or written to.

SCIDR — Reset value: undefined

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

The data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).

**Table 81. LINSCI1 register map and reset values (continued)**

| Addr. (Hex.) | Register name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 40 | SCISR<br>Reset value | TDRE<br>1 | TC<br>1 | RDRF<br>0 | IDLE<br>0 | OR/LHE<br>0 | NF<br>0 | FE<br>0 | PE<br>0 |
| 41 | SCIDR<br>Reset value | DR7<br>- | DR6<br>- | DR5<br>- | DR4<br>- | DR3<br>- | DR2<br>- | DR1<br>- | DR0<br>- |

# 11.6 10-bit A/D converter (ADC)

## 11.6.1 Introduction

The on-chip analog to digital converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to seven multiplexed analog input channels (refer to device pinout description) that allow the peripheral to convert the analog voltage levels from up to seven different sources.

The result of the conversion is stored in a 10-bit data register. The A/D converter is controlled through a control/status register.

## 11.6.2 Main features

- 10-bit conversion
- Up to 7 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in *Figure 69: ADC block diagram on page 164*.

## 11.6.3 Functional description

### Analog power supply

$V_{DDA}$ and $V_{SSA}$ are the high and low level reference voltage pins. In some devices (refer to *Section 2: Pin description*) they are internally connected to the $V_{DD}$ and $V_{SS}$ pins.

Conversion accuracy may therefore be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

**Figure 73. Typical accuracy with RCCR = RCCR0 vs. $V_{DD}$ = 4.5 to 5.5 V and temperature**

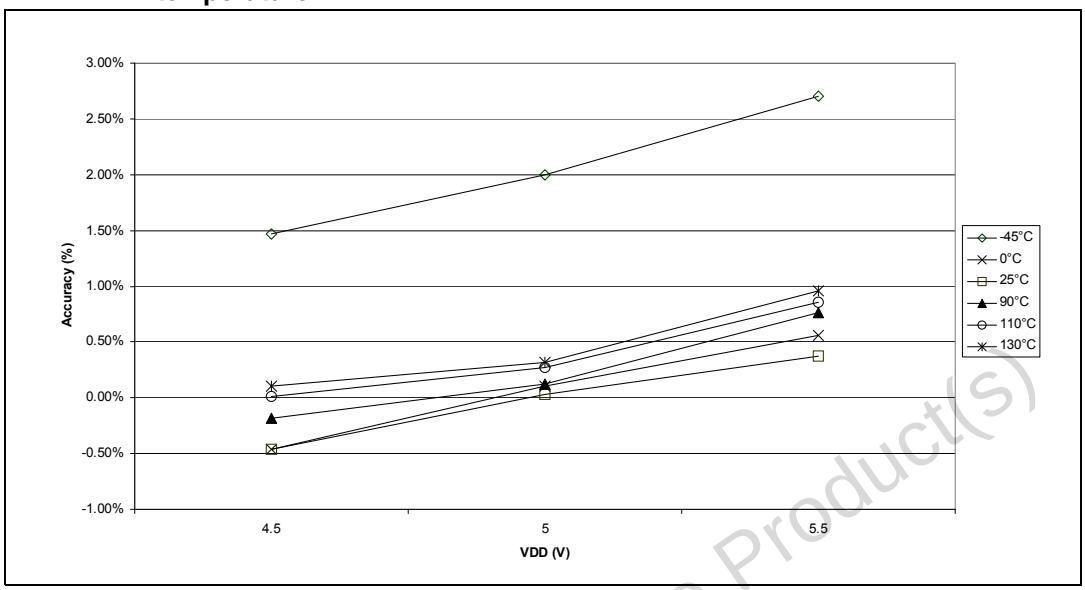



**Figure 74. $f_{RC}$ vs. $V_{DD}$ and temperature for calibrated RCCR0**

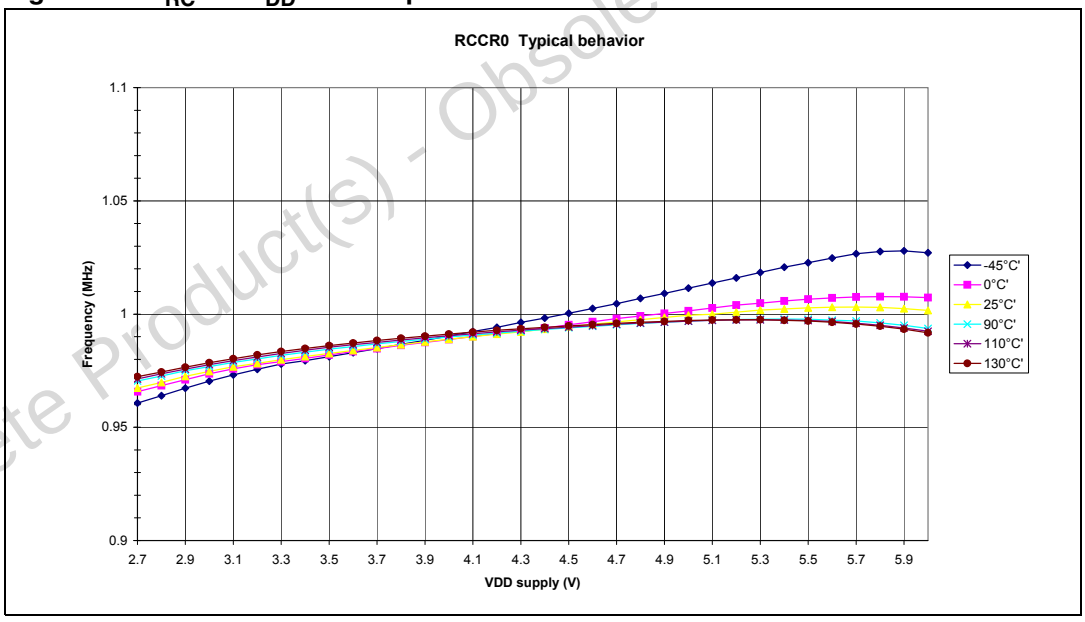





**Table 103. Operating conditions (tested for $T_A$ = -40 to +125 °C) @ $V_{DD}$ = 3.0 to 3.6 V**

| Symbol | Parameter | Conditions | Flash | | | ROM | | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| $f_{RC}$(1) | Internal RC oscillator frequency | RCCR = FF (reset value), $T_A$ = 25 °C, $V_{DD}$ = 3.3 V | | 630 | | | 630 | | kHz |
| | | RCCR = RCCR(2), $T_A$ = -40 to +125 °C, $V_{DD}$ = 3.3 V | 970 | 1000 | 1050 | TBD | 1000 | TBD | |

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

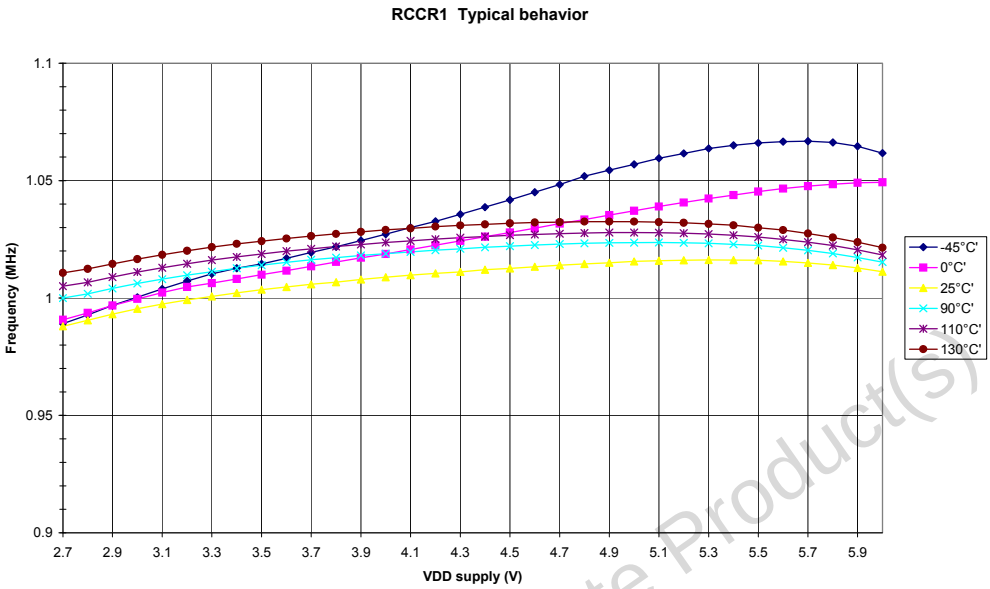
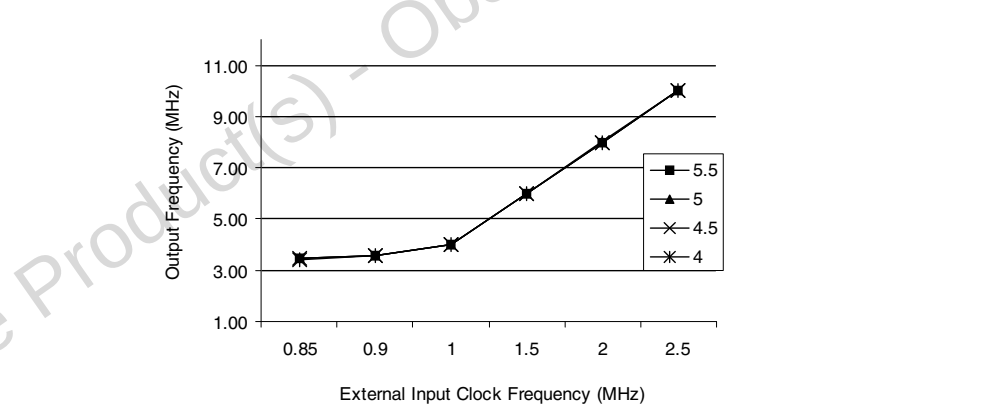



**Figure 77. PLL x 8 output vs. CLKIN frequency**




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) | |

### 15.4.4 Order codes for development and programming tools

*Table 135* below lists the ordering codes for the ST7L3x development and programming tools. For additional ordering codes for spare parts and accessories, refer to the online product selector at www.st.com/mcu.

**Table 135. ST7L3 development and programming tools**

| Supported products | In-circuit debugger, RLink series(1) | | Emulator | | Programming tool | |
|---|---|---|---|---|---|---|
| | Starter kit with demo board | Starter kit without demo board | DVP series | EMU series | In-circuit programmer | ST socket boards and EPBs |
| ST7FL34 | ST7FLITE-SK/RAIS(2)(3) | STX-RLINK(2)(3) | ST7MDT10-DVP3(4) | ST7MDT10-EMU3 | ST7-STICK STX-RLINK (4)(5) | ST7SB10-123(4) |
| ST7FL35 | | | | | | |
| ST7FL38 | | | | | | |
| ST7FL39 | | | | | | |

1. *Available from ST or from Raisonance*
2. *USB connection to PC*
3. Parallel port connection to PC
4. *Add suffix /EU, /UK or /US for the power supply for your region*
5. *Includes connection kit for DIP16/SO16 only. See "How to order an EMU or DVP" in ST product and tool selection guide for connection kit ordering information*