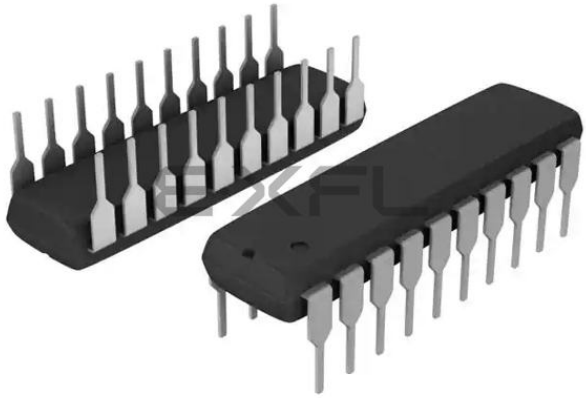


**Welcome to E-XFL.COM**

## 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 | 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) | 2.4V ~ 5.5V |
| Data Converters | A/D 7x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 20-DIP (0.300", 7.62mm) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/st7flite29f1b6 |

# List of figures

**Table 3. Hardware register map(1) (continued)**

| Address | Block | Register label | Register name | Reset status | Remarks |
|---|---|---|---|---|---|
| 0037h | ITC | EICR | External Interrupt Control Register | 00h | R/W |
| 0038h | MCC | MCCSR | Main Clock Control/Status Register | 00h | R/W |
| 0039h<br>003Ah | Clock and Reset | RCCR<br>SICSR | RC oscillator Control Register<br>System Integrity Control/Status Register | FFh<br>0000 0XX0h | R/W<br>R/W |
| 003Bh | Reserved area (1 byte) | | | | |
| 003Ch | ITC | EISR | External Interrupt Selection Register | 0Ch | R/W |
| 003Dh to 0048h | Reserved area (12 bytes) | | | | |
| 0049h<br>004Ah | AWU | AWUPR<br>AWUCSR | AWU Prescaler Register<br>AWU Control/Status Register | FFh<br>00h | R/W<br>R/W |
| 004Bh<br>004Ch<br>004Dh<br>004Eh<br>004Fh<br>0050h | DM(4) | DMCR<br>DMSR<br>DMBK1H<br>DMBK1L<br>DMBK2H<br>DMBK2L | DM Control Register<br>DM Status Register<br>DM Breakpoint Register 1 High<br>DM Breakpoint Register 1 Low<br>DM Breakpoint Register 2 High<br>DM Breakpoint Register 2 Low | 00h<br>00h<br>00h<br>00h<br>00h<br>00h | R/W<br>R/W<br>R/W<br>R/W<br>R/W<br>R/W |
| 0051h to 007Fh | Reserved area (47 bytes) | | | | |

1. Legend: x = undefined, R/W = read/write.
2. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.
3. The bits associated with unavailable pins must always keep their reset value.
4. For a description of the Debug Module registers, see ICC reference manual.

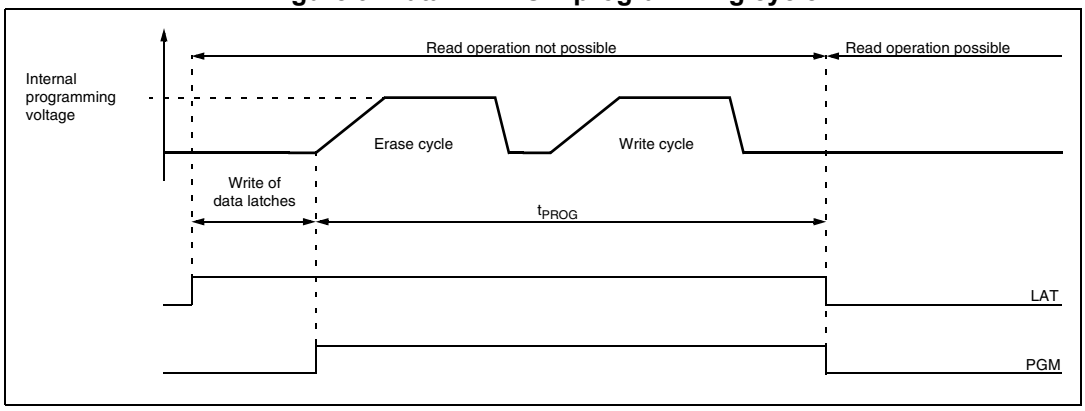
## 5.6 Data EEPROM Read-out protection

The Read-out protection is enabled through an option bit (see *Section 15.1: Option bytes*).

When this option is selected, the programs and data stored in the EEPROM memory are protected against Read-out (including a re-write protection). In Flash devices, when this protection is removed by reprogramming the Option Byte, the entire Program memory and EEPROM is first automatically erased.

*Note: Both Program Memory and DATA EEPROM are protected using the same option bit.*

**Figure 9. Data EEPROM programming cycle**




## 5.7 Register description

### EEPROM Control/Status register (EECSR)

Read/Write

Reset Value: 0000 0000 (00h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | E2LAT | E2PGM |

- Bits 7:2 = Reserved, forced by hardware to 0.
- Bit 1 = **E2LAT** *Latch Access Transfer*
  This bit is set by software. It is cleared by hardware at the end of the programming cycle. It can only be cleared by software if the E2PGM bit is cleared.
  0: Read mode
  1: Write mode
- Bit 0 = **E2PGM** *Programming control and status*
  This bit is set by software to begin the programming cycle. At the end of the programming cycle, this bit is cleared by hardware.
  0: Programming finished or not yet started
  1: Programming cycle is in progress

*Note: If the E2PGM bit is cleared during the programming cycle, the memory data is not guaranteed.*

### Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

### Index registers (X and Y)

In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. The cross-assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

### Program counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (program counter high which is the MSB).

### Condition code register (CC)

Read/Write

Reset value: 111x1xxx

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | H | I | N | Z | C |

The 8-bit Condition Code register contains the interrupt masks and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

- Bit 4 = **H** *Half carry*

  This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.
  0: No half carry has occurred
  1: A half carry has occurred
  This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

- Bit 3 = **I** *Interrupt mask*

  This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.
  0: Interrupts are enabled
  1: Interrupts are disabled

  This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

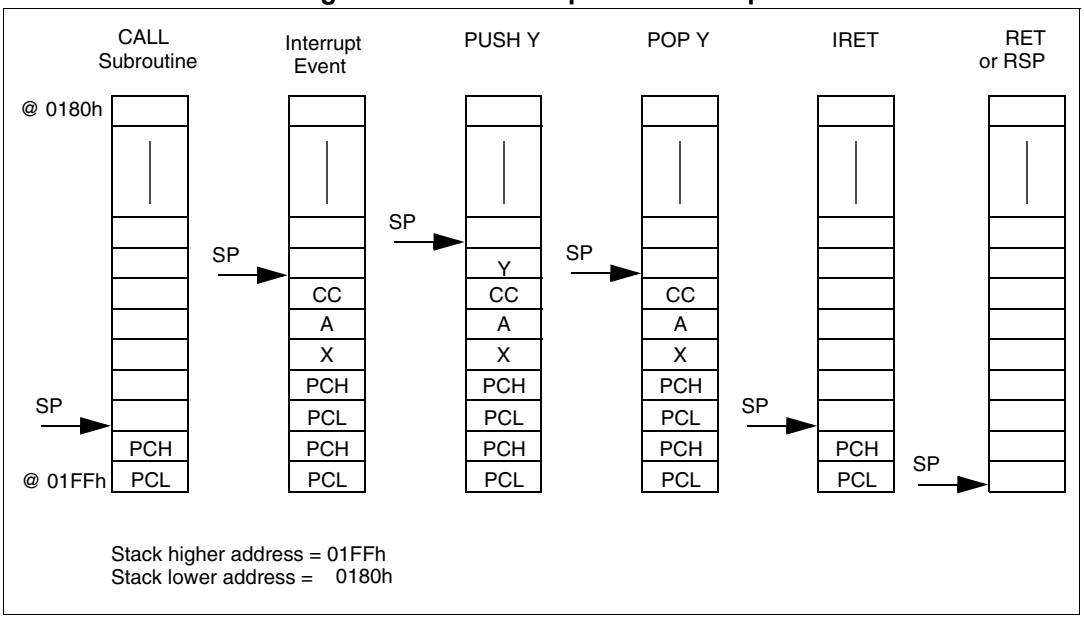
*Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptible because the I bit is set by hardware at the start of the routine and reset by the IRET instruction at the end of the routine. If the I bit is cleared*

pointed to by the SP. Then the other registers are stored in the next locations as shown in *Figure 11*:

- When an interrupt is received, the SP is decremented and the context is pushed on the stack.
- On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

**Figure 11. Stack manipulation example**



### 10.2.3 Alternate functions

Many ST7s I/Os have one or more alternate functions. These may include output signals from, or input signals to, on-chip peripherals. The Device Pin Description table 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 will increase 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.

**Figure 31. I/O port general block diagram**

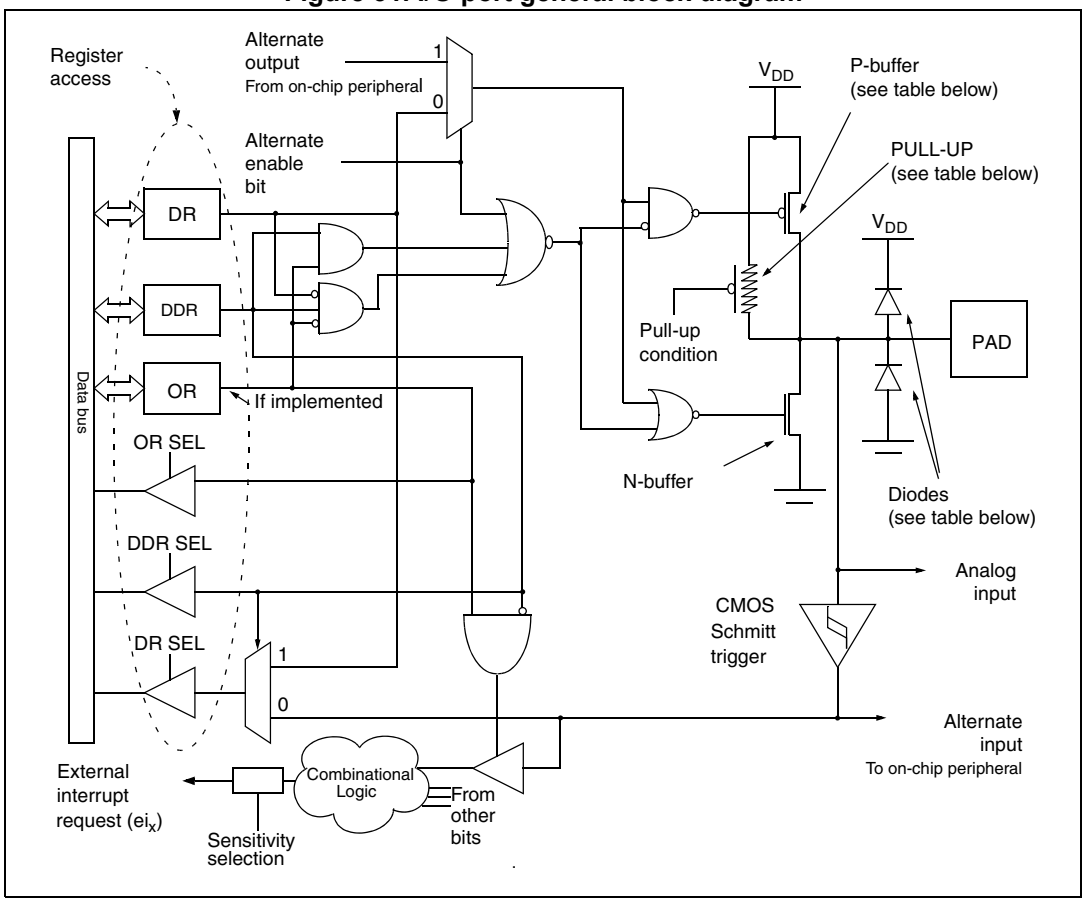


**Figure 36. PWM function**

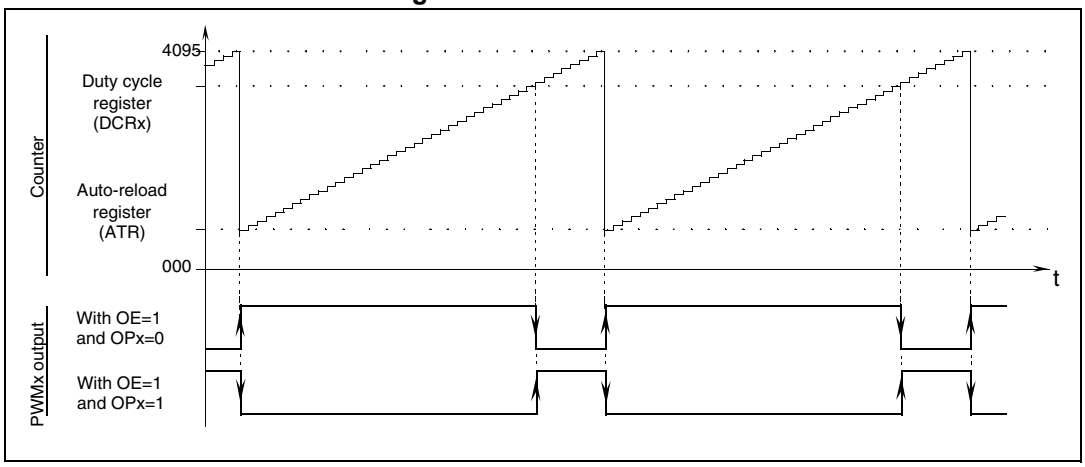
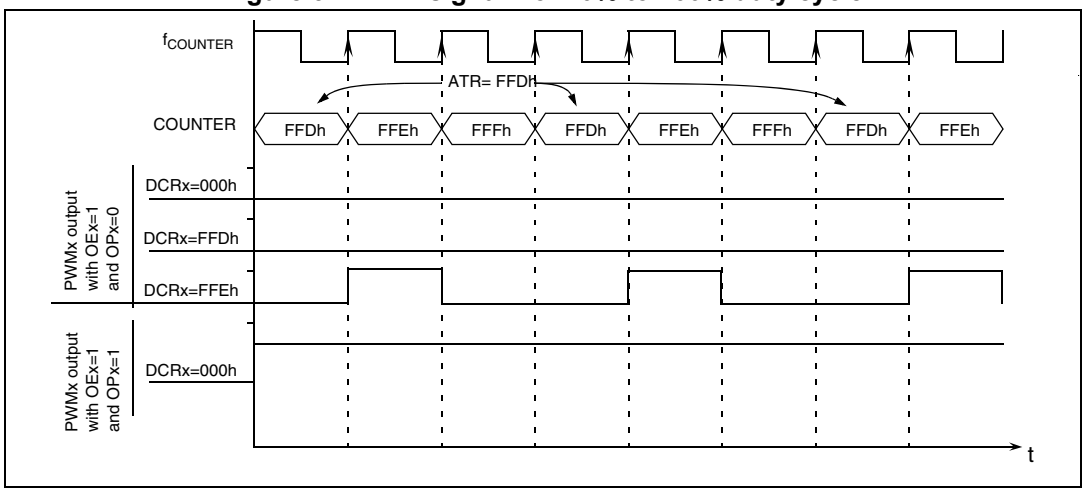



**Figure 37. PWM signal from 0% to 100% duty cycle**




## Output compare mode

To use this function, load a 12-bit value in the DCRxH and DCRxL registers.

When the 12-bit upcounter (CNTR) reaches the value stored in the DCRxH and DCRxL registers, the CMPF bit in the PWMxCSR register is set and an interrupt request is generated if the CMPIE bit is set.

*Note: The output compare function is only available for DCRx values other than 0 (reset value).*

## Break function

The break function is used to perform an emergency shutdown of the power converter.

The break function is activated by the external BREAK pin (active low). In order to use the BREAK pin it must be previously enabled by software setting the BPEN bit in the BREAKCR register.

When a low level is detected on the BREAK pin, the BA bit is set and the break function is activated.

0: CMPF interrupt disabled.
1: CMPF interrupt enabled.

### Counter register high (CNTRH)

Read only

Reset value: 0000 0000 (000h)

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | CNTR11 | CNTR10 | CNTR9 | CNTR8 |

### Counter register low (CNTRL)

Read only

Reset value: 0000 0000 (000h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| CNTR7 | CNTR6 | CNTR5 | CNTR4 | CNTR3 | CNTR2 | CNTR1 | CNTR0 |

- Bits 15:12 = Reserved
- Bits 11:0 = **CNTR[11:0]** *Counter value*
  This 12-bit register is read by software and cleared by hardware after a reset. The counter is incremented continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations, LSB first. When a counter overflow occurs, the counter restarts from the value specified in the ATR register.

### Autoreload register (ATRH)

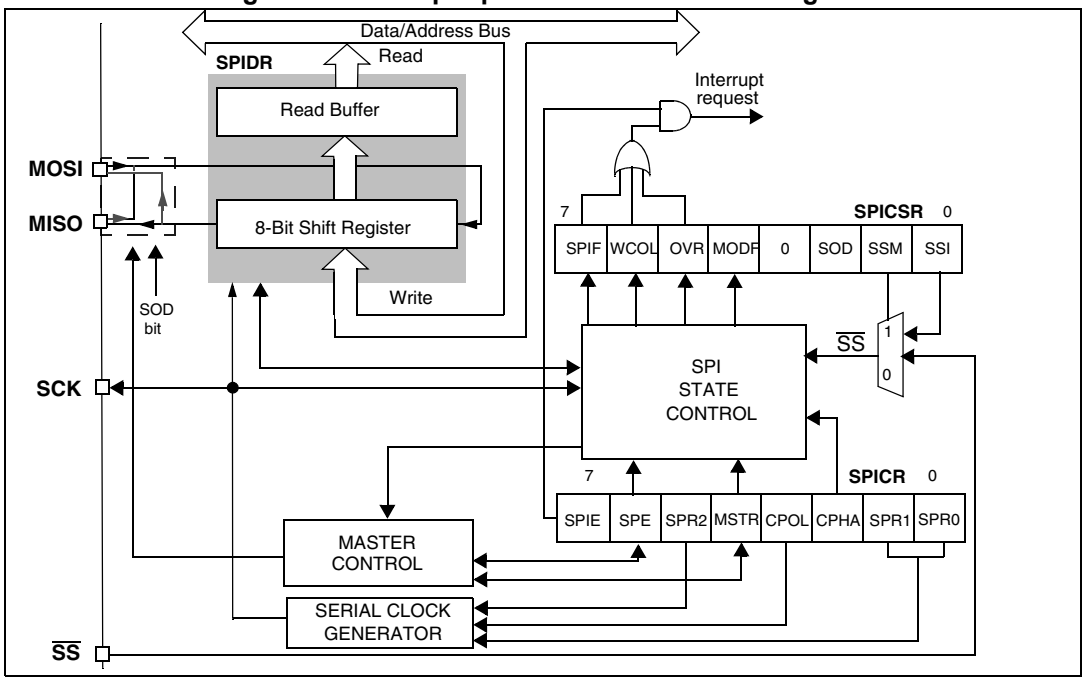
Read / Write

Reset Value: 0000 0000 (00h)

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | ATR11 | ATR10 | ATR9 | ATR8 |

- Bits 15:12 = Reserved
- Bits 11:0 = **ATR[11:0]** *Counter value*
  This 12-bit register is read by software and cleared by hardware after a reset. The counter is incremented continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations, LSB first. When a counter overflow occurs, the counter restarts from the value specified in the ATR register.

**Figure 42. Serial peripheral interface block diagram**




## Functional description

A basic example of interconnections between a single master and a single slave is illustrated in *Figure 43: Single master/ single slave application*.

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

The communication is always initiated by the master. When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex communication with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

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

Four possible data/clock timing relationships may be chosen (see *Figure 46: Data clock timing diagram*) but master and slave must be programmed with the same timing mode.

1. The MODF bit is set and an SPI interrupt request is generated if the SPIE bit is set.
2. The SPE bit is reset. This blocks all output from the Device and disables the SPI peripheral.
3. The MSTR bit is reset, thus forcing the Device into slave mode.

Clearing the MODF bit is done through a software sequence:

1. A read access to the SPICSR register while the MODF bit is set.
2. A write to the SPICR register.

*Note: To avoid any conflicts in an application with multiple slaves, the $\overline{SS}$ pin must be pulled high during the MODF bit clearing sequence. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.*

Hardware does not allow the user to set the SPE and MSTR bits while the MODF bit is set except in the MODF bit clearing sequence.

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

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

### Overrun condition (OVR)

An overrun condition occurs, when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previously transmitted byte.

When an overrun occurs:

– The OVR bit is set and an interrupt request is generated if the SPIE bit is set.

In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read to the SPIDR register returns this byte. All other bytes are lost.

The OVR bit is cleared by reading the SPICSR register.

### Write collision error (WCOL)

A write collision occurs when the software tries to write to the SPIDR register while a data transfer is taking place with an external device. When this happens, the transfer continues uninterrupted; and the software write will be unsuccessful.

Write collisions can occur both in master and slave mode. See also *Slave select management on page 93*.

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

The WCOL bit in the SPICSR register is set if a write collision occurs. No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see *Figure 47: Clearing the WCOL bit (write collision flag) software sequence*).

1. All power ($V_{DD}$) and ground ($V_{SS}$) lines must always be connected to the external supply.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if $V_{IN}$ maximum is respected. If $V_{IN}$ maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding $V_{IN}$ maximum must always be respected.
3. Negative injection disturbs the analog performance of the device. In particular, it induces leakage currents throughout the device including the analog inputs. To avoid undesirable effects on the analog functions, care must be taken:
   - Analog input pins must have a negative injection less than 0.8 mA (assuming that the impedance of the analog voltage is lower than the specified limits)
   - Pure digital pins must have a negative injection less than 1.6mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.
4. No negative current injection allowed on PB0 and PB1 pins.
5. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterization with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

**Table 59. Thermal characteristics**

| Symbol | Ratings | Value | Unit |
|---|---|---|---|
| $T_{STG}$ | Storage temperature range | -65 to +150 | °C |
| $T_J$ | Maximum junction temperature(1) | – | – |

1. (see *Table 90: Thermal characteristics*)

**Figure 54. RC Osc Freq vs $V_{DD}$ (calibrated with RCCR0: 5V@ 25°C)**

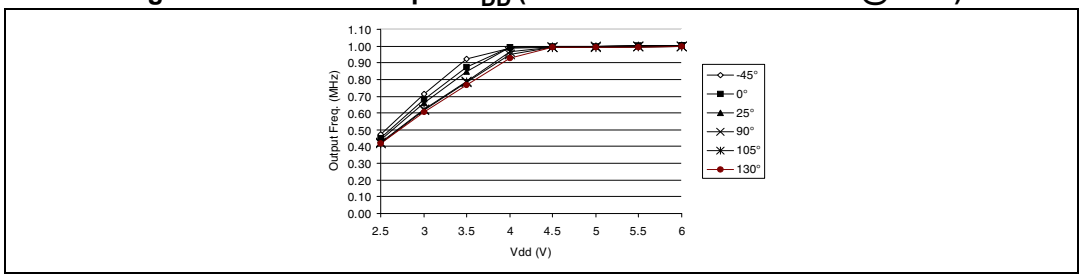



**Figure 55. Typical RC oscillator Accuracy vs temperature @ $V_{DD}$=5V (calibrated with RCCR0: 5V @ 25°C)**

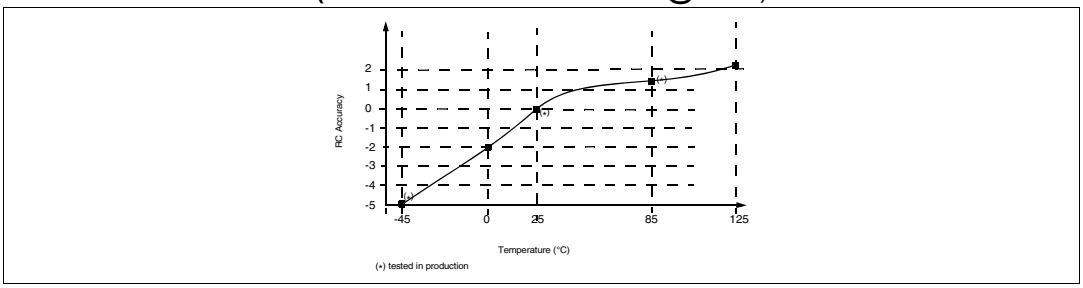



**Figure 56. RC Osc Freq vs $V_{DD}$ and RCCR Value**

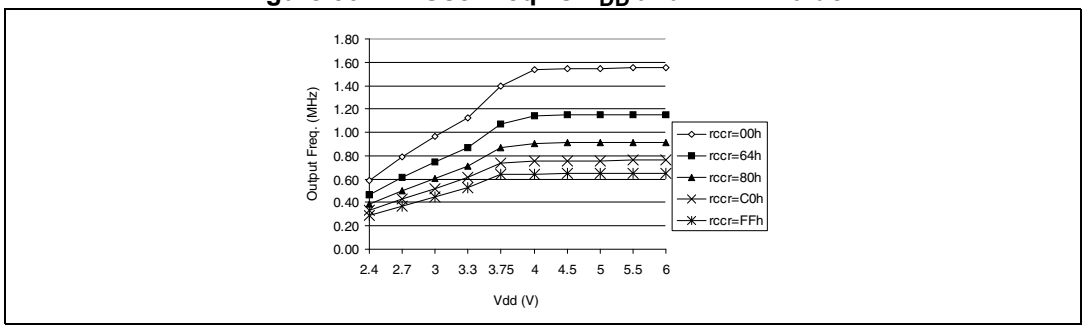
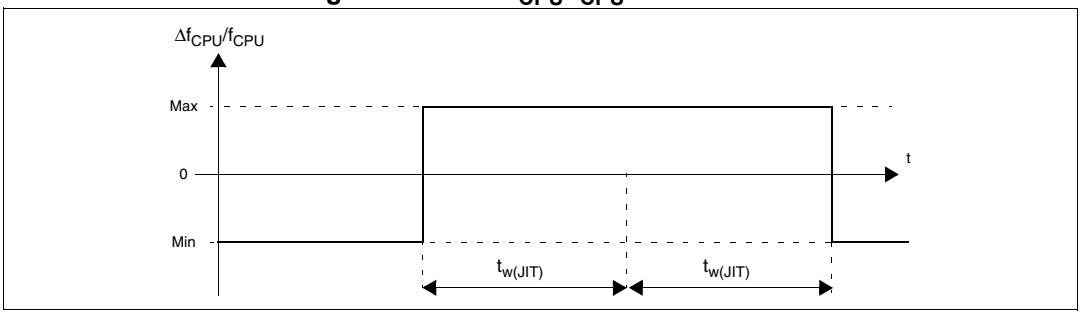
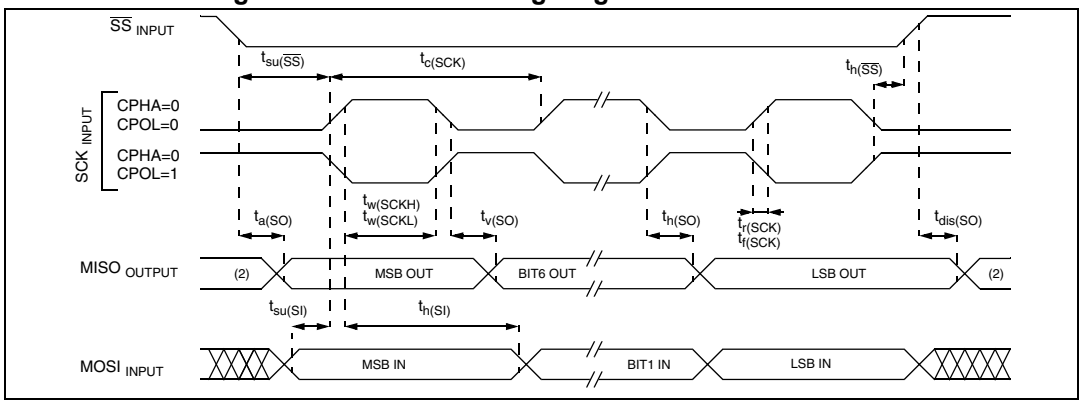



**Figure 57. PLL $\Delta f_{CPU}/f_{CPU}$ versus time**



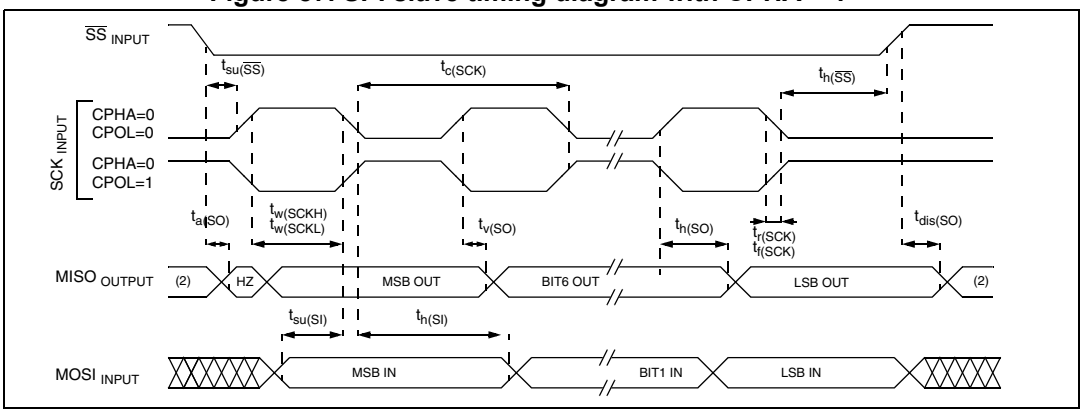
3. Depends on $f_{CPU}$. For example, if $f_{CPU}$ = 8MHz, then $T_{CPU}$ = 1/ $f_{CPU}$ = 125ns and $t_{SU(\overline{SS})}$ = 550ns

**Figure 86. SPI slave timing diagram with CPHA = 0(1)**



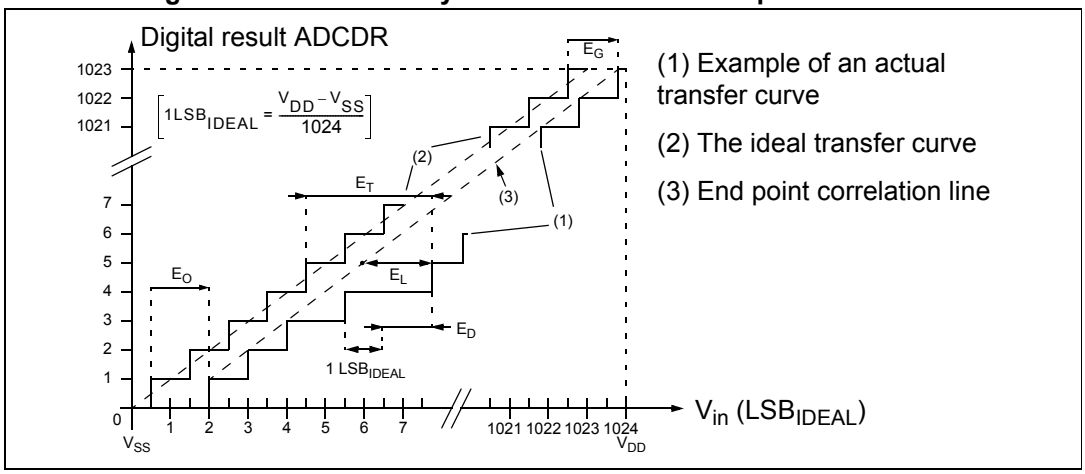

1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

**Figure 87. SPI slave timing diagram with CPHA = 1(1)**



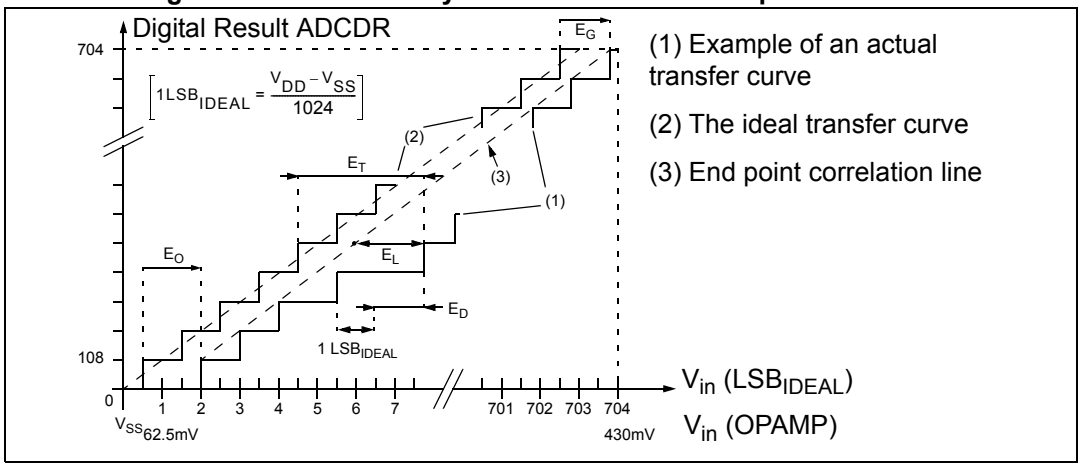

1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

**Figure 90. ADC accuracy characteristics with amplifier disabled**




- **$E_T$**=Total Unadjusted Error: maximum deviation between the actual and the ideal transfer curves.
- **$E_O$**=Offset Error: deviation between the first actual transition and the first ideal one.
- **$E_G$**=Gain Error: deviation between the last ideal transition and the last actual one.
- **$E_D$**=Differential Linearity Error: maximum deviation between actual steps and the ideal one.
- **$E_L$**=Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

**Figure 91. ADC accuracy characteristics with amplifier enabled**




*Note: When the AMPSEL bit in the ADCDRL register is set, it is mandatory that $f_{ADC}$ be less than or equal to 2 MHz (if $f_{CPU}$=8MHz, then SPEED=0, SLOW=1).*

- **$E_T$**=Total Unadjusted Error: maximum deviation between the actual and the ideal transfer curves.
- **$E_O$**=Offset Error: deviation between the first actual transition and the first ideal one.
- **$E_G$**=Gain Error: deviation between the last ideal transition and the last actual one.
- **$E_D$**=Differential Linearity Error: maximum deviation between actual steps and the ideal one.
- **$E_L$**=Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

### 15.1.2 Option byte 1

- OPT7 = **PLLx4x8** *PLL Factor selection.*
  0: PLLx4
  1: PLLx8
- OPT6 = **PLLOFF** *PLL disable*
  0: PLL enabled
  1: PLL disabled (by-passed)
- OPT5 = **PLL32OFF** *32MHz PLL disable*
  0: PLL32 enabled
  1: PLL32 disabled (by-passed)
- OPT4 = **OSC** *RC Oscillator selection*
  0: RC oscillator on
  1: RC oscillator off

*Note: 1% RC oscillator available on ST7LITE25 and ST7LITE29 devices only*

*If the RC oscillator is selected, then to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100nF, between the $V_{DD}$ and $V_{SS}$ pins as close as possible to the ST7 device.*

- OPT3:2 = **LVD[1:0]** *Low voltage detection selection*
  These option bits enable the LVD block with a selected threshold as shown in *Table 95*.

**Table 95. LVD threshold configuration**

| Configuration | LVD1 | LVD0 |
|---|---|---|
| LVD Off | 1 | 1 |
| Highest Voltage Threshold (4.1V) | 1 | 0 |
| Medium Voltage Threshold (3.5V) | 0 | 1 |
| Lowest Voltage Threshold (2.8V) | 0 | 0 |

- OPT1 = **WDG SW** *Hardware or Software Watchdog*
  This option bit selects the watchdog type.
  0: Hardware (watchdog always enabled)
  1: Software (watchdog to be enabled by software)
- OPT0 = **WDG HALT** *Watchdog Reset on HALT*
  This option bit determines if a RESET is generated when entering HALT mode while the Watchdog is active.
  0: No Reset generation when entering HALT mode
  1: Reset generation when entering HALT mode

**Table 97. Supported part numbers**

| Part number | Program memory (Bytes) | RAM (Bytes) | Data EEPROM (Bytes) | Temp. range | Package |
|---|---|---|---|---|---|
| ST7FLITE20F2B6 | 8K Flash | 384 | – | -40 °C to +85 °C | DIP20 |
| ST7FLITE20F2M6 | | | | | SO20 |
| ST7FLITE25F2B6 | | | – | | DIP20 |
| ST7FLITE25F2M6 | | | | | SO20 |
| ST7FLITE29F2B6 | | | 256 | | DIP20 |
| ST7FLITE29F2M6 | | | | | SO20 |
| ST7FLITE29F2M7 | | | | -40 °C to +105 °C | SO20 |
| ST7PLITE20F2B6 | 8K FASTROM | 384 | – | -40 °C to +85 °C | DIP20 |
| ST7PLITE20F2M6 | | | | | SO20 |
| ST7PLITE25F2B6 | | | – | | DIP20 |
| ST7PLITE25F2M6 | | | | | SO20 |
| ST7PLITE29F2B6 | | | 256 | | DIP20 |
| ST7PLITE29F2M6 | | | | | SO20 |

*Note: Contact ST sales office for product availability.*

## 15.3 Development tools

STMicroelectronics offers a range of hardware and software development tools for the ST7 microcontroller family. Full details of tools available for the ST7 from third party manufacturers can be obtained from the STMicroelectronics Internet site: http//www.st.com.

Tools from these manufacturers include C compliers, evaluation tools, emulators and programmers.

### Emulators

Two types of emulators are available from ST for the ST7LITE2 family (refer to *Table 99*):

- **ST7 DVP3** entry-level emulator offers a flexible and modular debugging and programming solution. SO20 packages need a specific connection kit.
- **ST7 EMU3** high-end emulator is delivered with everything (probes, TEB, adapters etc.) needed to start emulating the ST7LITE2. To configure it to emulate other ST7 subfamily devices, the active probe for the ST7EMU3 can be changed and the ST7EMU3 probe is designed for easy interchange of TEBs (Target Emulation Board).

### In-circuit debugging kit

Two configurations are available from ST:

- ST7FLIT2-IND/USB: Low-cost In-Circuit Debugging kit from Softec Microsystems. Includes STX-InDART/USB board (USB port) and a specific demo board for ST7FLITE29 (DIP16) (a promotion package of 15 STFLIT2-IND/USB can be ordered with the following order code: STFLIT2-IND/15)
- STxF-INDART/USB (a promotion package of 15 STxF-INDART/USB can be ordered with the following order code: STxF-INDART)

### Flash programming tools

- ST7-STICK ST7 In-circuit Communication Kit, a complete software/hardware package for programming ST7 Flash devices. It connects to a host PC parallel port and to the target board or socket board via ST7 ICC connector.
- ICC Socket Boards provide an easy to use and flexible means of programming ST7 Flash devices. They can be connected to any tool that supports the ST7 ICC interface, such as ST7 EMU3, ST7-DVP3, inDART, ST7-STICK, or many third-party development tools.

### Evaluation boards

One evaluation tool is available from ST:

- ST7FLIT2-COS/COM: STReal time starter kit from Cosmic software.