



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Not For New Designs
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/st7flit15bf0m3

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

DATA EEPROM (Cont'd)

5.7 REGISTER DESCRIPTION

EEPROM CONTROL/STATUS REGISTER (EEC-SR)

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

<u>(</u>ح)

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

Table 3. DATA EEPROM Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0030h	EECSR Reset Value	0	0	0	0	0	0	E2LAT 0	E2PGM 0

CPU REGISTERS (cont'd) CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	н	Ι	Ν	Z	С

The 8-bit Condition Code register contains the interrupt mask 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.

57

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 by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = N Negative

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7th bit of the result.

0: The result of the last operation is positive or null.

1: The result of the last operation is negative (that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = Z Zero

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

- 0: The result of the last operation is different from zero.
- 1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = C Carry/borrow

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation. 0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

RESET SEQUENCE MANAGER (Cont'd)

The RESET pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to follow the guidelines mentioned in the electrical characteristics section.

7.5.3 External Power-On RESET

If the LVD is disabled by option byte, to start up the microcontroller correctly, the user must ensure by means of an external reset circuit that the reset signal is held low until V_{DD} is over the minimum level specified for the selected f_{OSC} frequency.

A proper reset signal for a slow rising V_{DD} supply can generally be provided by an external RC network connected to the RESET pin.

7.5.4 Internal Low Voltage Detector (LVD) RESET

Two different RESET sequences caused by the internal LVD circuitry can be distinguished:

- Power-On RESET
- Voltage Drop RESET

The device $\overline{\text{RESET}}$ pin acts as an output that is pulled low when $V_{DD}{<}V_{IT{+}}$ (rising edge) or $V_{DD}{<}V_{IT{-}}$ (falling edge) as shown in Figure 17.

The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

7.5.5 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 17.

Starting from the Watchdog counter underflow, the device RESET pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.



Figure 17. RESET Sequences

8 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: Maskable hardware interrupts as listed in the "interrupt mapping" table and a nonmaskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 1.

The maskable interrupts must be enabled by clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

Note: After reset, all interrupts are disabled.

When an interrupt has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- The I bit of the CC register is set to prevent additional interrupts.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to the Interrupt Mapping table for vector addresses).

The interrupt service routine should finish with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I bit is cleared and the main program resumes.

Priority Management

By default, a servicing interrupt cannot be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case when several interrupts are simultaneously pending, an hardware priority defines which one will be serviced first (see the Interrupt Mapping table).

Interrupts and Low Power Mode

All interrupts allow the processor to leave the WAIT low power mode. Only external and specifically mentioned interrupts allow the processor to leave the HALT low power mode (refer to the "Exit from HALT" column in the Interrupt Mapping table).

8.1 NON MASKABLE SOFTWARE INTERRUPT

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It is serviced according to the flowchart in Figure 1.

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 the I/O ports section), 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 I bit of the CC register is cleared.
- The corresponding enable bit is set in the control register.

If any 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.

POWER SAVING MODES (Cont'd)

Figure 27. ACTIVE-HALT Timing Overview



Figure 28. ACTIVE-HALT Mode Flow-chart



Notes:

1. This delay occurs only if the MCU exits ACTIVE-HALT mode by means of a RESET.

2. Peripherals clocked with an external clock source can still be active.

3. Only the RTC1 interrupt and some specific interrupts can exit the MCU from ACTIVE-HALT mode. Refer to Table 5, "Interrupt Mapping," on page 37 for more details.

Before servicing an interrupt, the CC register is pushed on the stack. The I bit of the CC register is set during the interrupt routine and cleared when the CC register is popped.

9.6 AUTO WAKE UP FROM HALT MODE

Auto Wake Up From Halt (AWUFH) mode is similar to Halt mode with the addition of a specific internal RC oscillator for wake-up (Auto Wake Up 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 realtime 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 (fAWU 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 or 4096 cycle delay is used to stabilize it. After this start-up 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 fAWU BC 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 fAWU BC to the input capture of the 12-bit Auto-Reload timer, allowing the fAWU RC to be measured using the main oscillator clock as a reference timebase.

0

POWER SAVING MODES (Cont'd)

9.6.0.1 Register Description

AWUFH CONTROL/STATUS REGISTER (AWUCSR)

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

7							0
0	0	0	0	0	AWU F	AWU M	AWU EN

Bits 7:3 = Reserved.

Bit 1= AWUF Auto Wake Up Flag

This bit is set by hardware when the AWU module generates an interrupt and cleared by software on reading AWUCSR. Writing to this bit does not change its value.

0: No AWU interrupt occurred

1: AWU interrupt occurred

Bit 1= **AWUM** Auto Wake Up Measurement

This bit enables the AWU RC oscillator and connects its output to the input capture of the 12-bit Auto-Reload timer. This allows the timer to be used to measure the AWU RC oscillator dispersion and then compensate this dispersion by providing the right value in the AWUPRE register.

0: Measurement disabled

1: Measurement enabled

5/

Bit 0 = **AWUEN** Auto Wake Up From Halt Enabled This bit enables the Auto Wake Up From Halt feature: once HALT mode is entered, the AWUFH wakes up the microcontroller after a time delay dependent on the AWU prescaler value. It is set and cleared by software.

- 0: AWUFH (Auto Wake Up From Halt) mode disabled
- 1: AWUFH (Auto Wake Up From Halt) mode enabled

AWUFH PRESCALER REGISTER (AWUPR) Read/Write

Table 7.	AWU F	Register	Мар	and	Reset	Values
----------	-------	----------	-----	-----	-------	--------

7	
1	

| AWU |
|-----|-----|-----|-----|-----|-----|-----|-----|
| PR7 | PR6 | PR5 | PR4 | PR3 | PR2 | PR1 | PR0 |

Bits 7:0= **AWUPR[7:0]** Auto Wake Up Prescaler These 8 bits define the AWUPR Dividing factor (as explained below:

AWUPR[7:0]	Dividing factor
00h	Forbidden
01h	1
FEh	254
FFh	255

In AWU mode, the period that the MCU stays in Halt Mode (t_{AWU} in Figure 30 on page 45) is defined by

$$^{t}AWU = 64 \times AWUPR \times \frac{1}{f_{AWURC}} + ^{t}RCSTRT$$

This prescaler register can be programmed to modify the time that the MCU stays in Halt mode before waking up automatically.

Note: If 00h is written to AWUPR, depending on the product, an interrupt is generated immediately after a HALT instruction, or the AWUPR remains unchanged.

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0049h	AWUPR Reset Value	AWUPR7 1	AWUPR6 1	AWUPR5 1	AWUPR4 1	AWUPR3 1	AWUPR2 1	AWUPR1 1	AWUPR0 1
004Ah	AWUCSR Reset Value	0	0	0	0	0	AWUF	AWUM	AWUEN

DUAL 12-BIT AUTORELOAD TIMER 4 (Cont'd)

Long Input Capture

Pulses that last more than $8\mu s$ can be measured with an accuracy of $4\mu s$ if $f_{OSC} = 8$ MHz in the following conditions:

- The 12-bit AT4 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 AT4 Timer capture.

Figure 45. Long Range Input Capture Block Diagram

- 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 AT4 Timer to get a 20-bit input capture value. Refer to Figure 11.



Notes:

1. Since the input capture flags (ICF) for both timers (AT4 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.

2. 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 as follows:

- First, reset both ICIE bits.
- Then set the ICS bit.
- Reset both ICF bits.

- And then set the ICIE bit of desired interrupt.

3. How to compute a pulse length with long input capture feature.

As both timers are used, computing a pulse length is not straight-forward. The procedure is as follows:

 At the first input capture on the rising edge of the pulse, we assume that values in the registers are as follows:

LTICR = LT1ATICRH = ATH1 ATICRL = ATL1

Hence ATICR1 [11:0] = ATH1 & ATL1

Refer to Figure 12.

DUAL 12-BIT AUTORELOAD TIMER 4 (Cont'd)

11.2.6 Register Description

TIMER CONTROL STATUS REGISTER (ATCSR) Read / Write

Reset Value: 0x00 0000 (x0h)

7							0
0	ICF	ICIE	CK1	CK0	OVF1	OVFIE1	CMPIE

Bit 7 = Reserved.

Bit 6 = **ICF** Input Capture Flag.

This bit is set by hardware and cleared by software by reading the ATICR register (a read access to ATICRH or ATICRL will clear this flag). Writing to this bit does not change the bit value.

0: No input capture

1: An input capture has occurred

Bit 5 = **ICIE** *IC* Interrupt Enable. This bit is set and cleared by software. 0: Input capture interrupt disabled 1: Input capture interrupt enabled

Bits 4:3 = **CK[1:0]** Counter Clock Selection.

These bits are set and cleared by software and cleared by hardware after a reset. They select the clock frequency of the counter.

Counter Clock Selection	CK1	СКО
OFF	0	0
32 MHz	1	1
f _{LTIMER} (1 ms timebase @ 8 MHz)	0	1
fcpu	1	0

Bit 2 = OVF1 Overflow Flag.

This bit is set by hardware and cleared by software by reading the ATCSR register. It indicates the transition of the counter1 CNTR1 from FFFh to ATR1 value.

0: No counter overflow occurred

1: Counter overflow occurred

Bit 1 = **OVFIE1** Overflow Interrupt Enable.

This bit is read/write by software and cleared by hardware after a reset.

0: Overflow interrupt disabled.

1: Overflow interrupt enabled.

Bit 0 = **CMPIE** Compare Interrupt Enable.

This bit is read/write by software and cleared by hardware after a reset. It can be used to mask the interrupt generated when any of the CMPFx bit is set.

0: Output compare interrupt disabled.

1: Output Compare interrupt enabled.

COUNTER REGISTER 1 HIGH (CNTR1H)

Read only

Reset Value: 0000 0000 (00h)

15							8
0	0	0	0	CNTR1_ 11	CNTR1_ 10	CNTR1_ 9	CNTR1_ 8

COUNTER REGISTER 1 LOW (CNTR1L)

Read only

7

Reset Value: 0000 0000 (00h)

							-
CNTR1							
7	6	5	4	3	2	1	0

Bits 15:12 = Reserved.

Bits 11:0 = CNTR1[11:0] Counter Value.

This 12-bit register is read by software and cleared by hardware after a reset. The counter CNTR1 increments 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. As there is no latch, it is recommended to read LSB first. In this case, CNTR1H can be incremented between the two read operations and to have an accurate result when $f_{timer}=f_{CPU}$, special care must be taken when CNTR1L values close to FFh are read.

When a counter overflow occurs, the counter restarts from the value specified in the ATR1 register.



0

ST7LITE1xB

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
21	ATCSR2	FORCE2	FORCE1	ICS	OVFIE2	OVF2	ENCNTR2	TRAN2	TRAN1
	Reset Value	0	0	0	0	0	0	1	1
22	BREAKCR	BRSEL	BREDGE	BA	BPEN	PWM3	PWM2	PWM1	PWM0
	Reset Value	0	0	0	0	0	0	0	0
23	ATR2H Reset Value	0	0	0	0	ATR11 0	ATR10 0	ATR9 0	ATR8 0
24	ATR2L	ATR7	ATR6	ATR5	ATR4	ATR3	ATR2	ATR1	ATR0
	Reset Value	0	0	0	0	0	0	0	0
25	DTGR	DTE	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	Reset Value	0	0	0	0	0	0	0	0
26	BREAKEN Reset Value	0	0	0	0	0	0	BREN2 1	BREN1 1

SERIAL PERIPHERAL INTERFACE (cont'd)

11.4.3.3 Master Mode Operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

How to operate the SPI in master mode

To operate the SPI in master mode, perform the following steps in order:

- 1. Write to the SPICR register:
 - Select the clock frequency by configuring the SPR[2:0] bits.
 - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. Figure 5 shows the four possible configurations.
 Note: The slave must have the same CPOL and CPHA settings as the master.
- 2. Write to the SPICSR register:
 - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the SS pin high for the complete byte transmit sequence.
- 3. Write to the SPICR register:
 - Set the MSTR and SPE bits
 <u>Note</u>: MSTR and SPE bits remain set only if SS is high).

Important note: if the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account.

The transmit sequence begins when software writes a byte in the SPIDR register.

11.4.3.4 Master Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware.
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

- 1. An access to the SPICSR register while the SPIF bit is set
- 2. A read to the SPIDR register

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

11.4.3.5 Slave Mode Operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

- 1. Write to the SPICSR register to perform the following actions:
 - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see Figure 5).
 Note: The slave must have the same CPOL and CPHA settings as the master.
 - Manage the \overline{SS} pin as described in Section 0.1.3.2 and Figure 3. If CPHA = 1 \overline{SS} must be held low continuously. If CPHA = 0 \overline{SS} must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
- 2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

11.4.3.6 Slave Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware.
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

- 1. An access to the SPICSR register while the SPIF bit is set
- 2. A write or a read to the SPIDR register

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

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see Section 0.1.5.2).



10-BIT A/D CONVERTER (ADC) (Cont'd)

11.5.3.2 Input Voltage Amplifier

The input voltage can be amplified by a factor of 8 by enabling the AMPSEL bit in the ADCDRL register.

When the amplifier is enabled, the input range is 0V to $V_{\mbox{\scriptsize DD}}/8.$

For example, if $V_{DD} = 5V$, then the ADC can convert voltages in the range 0V to 430mV with an ideal resolution of 0.6mV (equivalent to 13-bit resolution with reference to a V_{SS} to V_{DD} range).

For more details, refer to the Electrical characteristics section.

Note: The amplifier is switched on by the ADON bit in the ADCCSR register, so no additional startup time is required when the amplifier is selected by the AMPSEL bit.

11.5.3.3 Digital A/D Conversion Result

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than V_{DDA} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and AD-CDRL registers. The accuracy of the conversion is described in the Electrical Characteristics Section.

 R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

11.5.3.4 A/D Conversion

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the «I/O ports» chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register:

57/

 Select the CH[2:0] bits to assign the analog channel to convert.

ADC Conversion mode

In the ADCCSR register:

Set the ADON bit to enable the A/D converter and to start the conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

- The EOC bit is set by hardware.
- The result is in the ADCDR registers.

A read to the ADCDRH or a write to any bit of the ADCCSR register resets the EOC bit.

To read the 10 bits, perform the following steps:

- 1. Poll the EOC bit
- 2. Read ADCDRL
- 3. Read ADCDRH. This clears EOC automatically.

To read only 8 bits, perform the following steps:

- 1. Poll EOC bit
- 2. Read ADCDRH. This clears EOC automatically.

11.5.3.5 Changing the conversion channel

The application can change channels during conversion.

When software modifies the CH[2:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

11.5.4 Low Power Modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

Mode	Description
WAIT	No effect on A/D Converter
	A/D Converter disabled.
HALT	After wakeup from Halt mode, the A/D Converter requires a stabilization time t_{STAB} (see Electrical Characteristics) before accurate conversions can be performed.

11.5.5 Interrupts

None.

10-BIT A/D CONVERTER (ADC) (Cont'd)

11.5.6 Register Description

CONTROL/STATUS REGISTER (ADCCSR)

Read/Write (Except bit 7 read only)

Reset Value: 0000 0000 (00h)

7							0
EOC	SPEED	ADON	0	0	CH2	CH1	CH0

Bit 7 = **EOC** End of Conversion

This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register. 0: Conversion is not complete

1: Conversion complete

Bit 6 = SPEED ADC clock selection

This bit is set and cleared by software. It is used together with the SLOW bit to configure the ADC clock speed. Refer to the table in the SLOW bit description (ADCDRL register).

Bit 5 = ADON A/D Converter on

This bit is set and cleared by software. 0: A/D converter and amplifier are switched off

1: A/D converter and amplifier are switched on

Bits 4:3 = **Reserved.** Must be kept cleared.

Bits 2:0 = CH[2:0] Channel Selection

These bits are set and cleared by software. They select the analog input to convert.

Channel Pin*	CH2	CH1	CH0
AINO	0	0	0
AIN1	0	0	1
AIN2	0	1	0
AIN3	0	1	1
AIN4	1	0	0
AIN5	1	0	1
AIN6	1	1	0

*The number of channels is device dependent. Refer to the device pinout description.

DATA REGISTER HIGH (ADCDRH)

Read Only

Reset Value: xxxx xxxx (xxh)

7							0
D9	D8	D7	D6	D5	D4	D3	D2

Bits 7:0 = D[9:2] MSB of Analog Converted Value

AMP CONTROL/DATA REGISTER LOW (AD-CDRL)

Read/Write

Reset Value: 0000 00xx (0xh)

7							0
0	0	0	AMP CAL	SLOW	AMP- SEL	D1	D0

Bits 7:5 = Reserved. Forced by hardware to 0.

Bit 4 = AMPCAL Amplifier Calibration Bit

This bit is set and cleared by software. It is advised to use this bit to calibrate the ADC when amplifier is ON. Setting this bit internally connects amplifier input to 0V. Hence, corresponding ADC output can be used in software to eliminate amplifier-offset error.

0: Calibration off

1: Calibration on. (The input voltage of the amplifier is set to 0V)

Bit 3 = SLOW Slow mode

This bit is set and cleared by software. It is used together with the SPEED bit in the ADCCSR register to configure the ADC clock speed as shown on the table below.

f _{ADC}	SLOW	SPEED
f _{CPU} /2	0	0
f _{CPU}	0	1
f _{CPU} /4	1	х

Note: max f_{ADC} allowed = 4MHz (see section 13.11 on page 139)

10-BIT A/D CONVERTER (ADC) (Cont'd)

Bit 2 = **AMPSEL** Amplifier Selection Bit This bit is set and cleared by software. 0: Amplifier is not selected 1: Amplifier is selected

Note: When AMPSEL=1 it is mandatory that $\rm f_{ADC}$ be less than or equal to 2 MHz.

Bits 1:0 = D[1:0] LSB of Analog Converted Value

	Table 18	ADC	Register	Map a	nd Rese	t Values
--	----------	-----	----------	-------	---------	----------

57

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0034h	ADCCSR Reset Value	EOC 0	SPEED 0	ADON 0	0 0	0 0	CH2 0	CH1 0	CH0 0
0035h	ADCDRH	D9	D8	D7	D6	D5	D4	D3	D2
	Reset Value	x	x	x	x	x	x	x	x
0036h	ADCDRL	0	0	0	AMPCAL	SLOW	AMPSEL	D1	D0
	Reset Value	0	0	0	0	0	0	x	x

OPERATING CONDITIONS (Cont'd)

13.3.5.2 Devices with "'6" or "3" order code suffix (tested for $T_A = -40$ to +125°C) @ $V_{DD} = 3.0$ to 3.6V

Symbol	Parameter	Conditions	Min	Тур	Мах	Unit
£	Internal RC oscillator fre-	RCCR = FF (reset value), T _A =25°C, V _{DD} = 3.3V		700		
IRC	quency ¹⁾	RCCR=RCCR1 ²⁾ , T _A =25°C, V _{DD} = 3.3V	992	1000	1008	КПZ
		T _A =25°C,V _{DD} =3.3V	-0.8		+0.8	%
	Accuracy of Internal RC oscillator when calibrated with RCCR=RCCR1 ²⁾	T _A =25°C,V _{DD} =3.0 to 3.6V ³⁾			+1	%
ACC	Accuracy of Internal RC	T _A =25 to +85°C,V _{DD} =3.3V	-3		+3	%
ACCRC	with RCCR=RCCR1 ²⁾	T _A =25 to +85°C,V _{DD} =3.0 to 3.6V ³⁾	-3.5		+3.5	%
ACC _{RC}		T _A =25 to +125°C,V _{DD} =3.0 to 3.6V ³⁾	-5		+6.5	%
		T_A =-40 to +25°C, V_{DD} =3.0 to 3.6V ³)	-3.5		+4	%
I _{DD(RC)}	RC oscillator current con- sumption	T _A =25°C,V _{DD} =3.3V		400 ³⁾		μA
t _{su(RC)}	RC oscillator setup time	T _A =25°C,V _{DD} =3.3V			10 ²⁾	μs
f _{PLL}	x4 PLL input clock			0.7 ³⁾		MHz
t _{LOCK}	PLL Lock time ⁵⁾			2		ms
t _{STAB}	PLL Stabilization time ⁵⁾			4		ms
ACC		$f_{RC} = 1MHz@T_A=25°C, V_{DD}=2.7 \text{ to } 3.3V$		0.1 ⁴⁾		%
ACCPLL	X4 FLL Accuracy	$f_{RC} = 1MHz@T_A=40 \text{ to } +85^{\circ}C, V_{DD}=3.3V$		0.1 ⁴⁾		%
t _{w(JIT)}	PLL jitter period ⁶⁾	f _{RC} = 1MHz		120		μs
JIT _{PLL}	PLL jitter (∆f _{CPU} /f _{CPU})			1 ⁷⁾		%
I _{DD(PLL)}	PLL current consumption	T _A =25°C		190 ³⁾		μA

Notes:

ل حک

1. If the RC oscillator clock is selected, 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.

2. See "INTERNAL RC OSCILLATOR ADJUSTMENT" on page 23.

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

4. Averaged over a 4ms period. After the LOCKED bit is set, a period of t_{STAB} is required to reach ACC_{PLL} accuracy

5. After the LOCKED bit is set ACC_{PLL} is max. 10% until t_{STAB} has elapsed. See Figure 13 on page 24.

6. This period is the PLL servoing period. During this period, the frequency remains unchanged.

7. Guaranteed by design.

13.7 EMC CHARACTERISTICS

Susceptibility tests are performed on a sample basis during product characterization.

13.7.1 Functional EMS (Electro Magnetic Susceptibility)

Based on a simple running application on the product (toggling 2 LEDs through I/O ports), the product is stressed by two electro magnetic events until a failure occurs (indicated by the LEDs).

- ESD: Electro-Static Discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- FTB: A Burst of Fast Transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed. The test results are given in the table below based on the EMS levels and classes defined in application note AN1709.

13.7.1.1 Designing hardened software to avoid noise problems

EMC characterization and optimization are performed at component level with a typical application environment and simplified MCU software. It should be noted that good EMC performance is highly dependent on the user application and the software in particular.

Therefore it is recommended that the user applies EMC software optimization and prequalification tests in relation with the EMC level requested for his application.

Software recommendations:

The software flowchart must include the management of runaway conditions such as:

- Corrupted program counter
- Unexpected reset
- Critical Data corruption (control registers...)

Prequalification trials:

Most of the common failures (unexpected reset and program counter corruption) can be reproduced by manually forcing a low state on the RE-SET pin or the Oscillator pins for 1 second.

To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behaviour is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

Symbol	Parameter	Conditions	Level/ Class
V _{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD} =5V, T_A =+25°C, f_{OSC} =8MHz conforms to IEC 1000-4-2	2B
V _{FFTB}	Fast transient voltage burst limits to be applied through 100pF on V_{DD} and V_{SS} pins to induce a functional disturbance	V _{DD} =5V, T _A =+25°C, f _{OSC} =8MHz conforms to IEC 1000-4-4	3B

13.7.2 Electro Magnetic Interference (EMI)

Based on a simple application running on the product (toggling 2 LEDs through the I/O ports), the product is monitored in terms of emission. This emission test is in line with the norm SAE J 1752/3 which specifies the board and the loading of each pin.

Symbol	Parameter	Conditions	Monitored	Max vs. [Unit		
Cymbol	i arameter	Conditions	Frequency Band	8/4MHz	16/8MHz		
S _{EMI}		V _{DD} =5V, T _A =+25°C, SO20 package, conforming to SAE J 1752/3	0.1MHz to 30MHz	15	21		
	Peak level		30MHz to 130MHz	22	29	dBµV	
			130MHz to 1GHz	17	22		
		3 1 1 1	SAE EMI Level	3.5	3.5	-	

Note:

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



I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 87. Typical V_{OL} at V_{DD} =2.7V (High-sink)



Figure 88. Typical V_{OL} at V_{DD}=3.3V (High-sink)



Figure 89. Typical V_{OL} at V_{DD}=5V (High-sink)



Figure 90. Typical V_{OL} vs. V_{DD} (standard I/Os)











57

PACKAGE CHARACTERISTICS (Cont'd)

Figure 114. 16-Pin Plastic Small Outline Package, 300-mil Width



Figure 115. 20-Pin Plastic Dual In-Line Package, 300-mil Width

57/



PACKAGE CHARACTERISTICS (Cont'd)

Figure 116. 20-Pin Plastic Small Outline Package, 300-mil Width



Figure 117. 20-Lead Very thin Fine pitch Quad Flat No-Lead Package



15 DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in user programmable versions (FLASH).

15.1 OPTION BYTES

The two option bytes allow the hardware configuration of the microcontroller to be selected.

The option bytes can be accessed only in programming mode (for example using a standard ST7 programming tool).

OPTION BYTE 0

5/

OPT7 = Reserved, must always be 1.

OPT6 = **PKG** *Package selection* 0: 16-pin package 1: 20-pin package

OPT5:4 = **CLKSEL** *Clock Source Selection*

When the internal RC oscillator is not selected (Option OSC=1), these option bits select the clock source: resonator oscillator or external clock

Clock Sou	urce	Port C	CLKSEL		
Resonator		Ext. Osc Disabled/ Port C Enabled	0	0	
Ext.	on PB4	Ext. Osc Enabled/	0	1	
Clock source: CLKIN	on PC0	Port C Disabled	1	1	
Reserve	ed		1	0	

Note: When the internal RC oscillator is selected, the CLKSEL option bits must be kept at their default value in order to select the 256 clock cycle delay (see Section 7.5).

ST7FLITE1xB devices are shipped to customers with a default program memory content (FFh). This implies that FLASH devices have to be configured by the customer using the Option Bytes.

OPT3:2 = SEC[1:0] Sector 0 size definition

These option bits indicate the size of sector 0 according to the following table.

Sector 0 Size	SEC1	SEC0
0.5k	0	0
1k	0	1
2k	1	0
4k	1	1

OPT1 = **FMP_R** Read-out protection

Readout protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Erasing the option bytes when the FMP_R option is selected will cause the whole memory to be erased first and the device can be reprogrammed. Refer to the ST7 Flash Programming Reference Manual and section 4.5 on page 14 for more details

0: Read-out protection off

1: Read-out protection on

OPT0 = FMP_W FLASH write protection

This option indicates if the FLASH program memory is write protected.

Warning: When this option is selected, the program memory (and the option bit itself) can never be erased or programmed again.

0: Write protection off

1: Write protection on

	OPTION BYTE 0								OPTION BYTE 1								
	7							0	7						0		
	Res.	PKG	CLK	SEL	SEC1	SEC0	FMP R	FMP W	PLL x4x8	PLL OFF	PLL32 OFF	osc	LVD1	LVD0	WDG SW	WDG HALT	
Default Value	1	1	1	1	0	1	0	0	1	1	1	0	1	1	1	1	

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2008 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com