



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	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	16MHz
Connectivity	LINbusSCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	15
Program Memory Size	8KB (8K × 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 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/st7flite39f2m6

Email: info@E-XFL.COM

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

FLASH PROGRAM MEMORY (Cont'd)

4.4 ICC INTERFACE

ICP needs a minimum of 4 and up to 6 pins to be connected to the programming tool. These pins are:

- RESET: device reset
- V_{SS}: device power supply ground

Figure 5. Typical ICC Interface

- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- CLKIN/PB4: main clock input for external source
- V_{DD}: application board power supply (optional, see Note 3)



Notes:

1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented if another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values. 2. During the IC<u>P sessi</u>on, the programming tool must control the RESET pin. This can lead to con-

flicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (push pull output or pull-up resistor<1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R>1K or a reset management IC with open drain output and pull-up resistor>1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.

3. The use of Pin 7 of the ICC connector depends on the Programming Tool architecture. This pin

must be connected when using most ST Programming Tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

4. Pin 9 must be connected to the PB4 pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability must have OSC2 grounded in this case.

5. With any programming tool, while the ICP option is disabled, the external clock must be provided on PB4.

6. In 38-pulse ICC mode, the internal RC oscillator is forced as a clock source, regardless of the selection in the option byte. For ST7LITE30 devices which do not support the internal RC oscillator, the "option byte disabled" mode must be used (35-pulse ICC mode entry, clock provided by the tool). Caution: During normal operation ICCCLK pin must be pulled- up, internally or externally (external pull-up of 10k mandatory in noisy environentering ment). This avoids ICC mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset puts it back in input pull-up.

FLASH PROGRAM MEMORY (Cont'd)

4.5 Memory Protection

There are two different types of memory protection: Read Out Protection and Write/Erase Protection which can be applied individually.

4.5.1 Read out Protection

Readout protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller. Both program and data E^2 memory are protected.

In flash devices, this protection is removed by reprogramming the option. In this case, both program and data E^2 memory are automatically erased and the device can be reprogrammed.

 Read-out protection selection is enabled and removed through the FMP_R bit in the option byte.

4.5.2 Flash Write/Erase Protection

Write/erase protection, when set, makes it impossible to both overwrite and erase program memory. It does not apply to E^2 data. Its purpose is to provide advanced security to applications and prevent any change being made to the memory content.

Warning: Once set, Write/erase protection can never be removed. A write-protected flash device is no longer reprogrammable.

Write/erase protection is enabled through the FMP_W bit in the option byte.

4.6 Related Documentation

For details on Flash programming and ICC protocol, refer to the ST7 Flash Programming Reference Manual and to the ST7 ICC Protocol Reference Manual.

4.7 Register Description

FLASH CONTROL/STATUS REGISTER (FCSR) Read/Write

Reset Value: 000 0000 (00h) 1st RASS Key: 0101 0110 (56h) 2nd RASS Key: 1010 1110 (AEh)

7							0
0	0	0	0	0	OPT	LAT	PGM

Note: This register is reserved for programming using ICP, IAP or other programming methods. It controls the XFlash programming and erasing operations.

When an EPB or another programming tool is used (in socket or ICP mode), the RASS keys are sent automatically.



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

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).



SYSTEM INTEGRITY MANAGEMENT (Cont'd)

7.6.4 Register Description

SYSTEM INTEGRITY (SI) CONTROL/STATUS REGISTER (SICSR)

Read/Write

Reset Value: 0110 0xx0 (6xh)

7							0
0	CR1	CR0	WDG RF	LOCKED	LVDRF	AVDF	AVDIE

Bit 7 = Reserved, must be kept cleared.

Bits 6:5 = **CR[1:0]** *RC* Oscillator Frequency Adjustment bits

These bits, as well as CR[9:2] bits in the RCCR register must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. Refer to section 7.3 on page 24

Bit 4 = WDGRF Watchdog reset flag

This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (by reading SICSR register) or an LVD Reset (to ensure a stable cleared state of the WDGRF flag when CPU starts).

Combined with the LVDRF flag information, the flag description is given by the following table.

RESET Sources	LVDRF	WDGRF
External RESET pin	0	0
Watchdog	0	1
LVD	1	Х

Bit 3 = LOCKED PLL Locked Flag

This bit is set by hardware. It is cleared only by a power-on reset. It is set automatically when the PLL reaches its operating frequency.

0: PLL not locked

1: PLL locked

Bit 2 = LVDRF LVD reset flag

This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (by reading). When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.

Bit 1 = AVDF Voltage Detector flag

This read-only bit is set and cleared by hardware. If the AVDIE bit is set, an interrupt request is generated when the AVDF bit is set. Refer to Figure 19 and to Section 7.6.2.1 for additional details. 0: V_{DD} over AVD threshold

1: V_{DD} under AVD threshold

Bit 0 = **AVDIE** Voltage Detector interrupt enable

This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag is set. The pending interrupt information is automatically cleared when software enters the AVD interrupt routine.

0: AVD interrupt disabled 1: AVD interrupt enabled

Application notes

The LVDRF flag is not cleared when another RE-SET type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure.

In this case, a watchdog reset can be detected by software while an external reset can not.



INTERRUPTS (Cont'd)

Bit 3:2 = ei1[1:0] *ei1 pin selection* These bits are written by software. They select the Port A I/O pin used for the ei1 external interrupt according to the table below.

External Interrupt I/O pin selection

ei11	ei10	I/O Pin
0	0	No interrupt*
0	1	PA4
1	0	PA5
1	1	PA6

* Reset State

Bit 1:0 = **ei0[1:0]** *ei0 pin selection*

These bits are written by software. They select the

Port A I/O pin used for the ei0 external interrupt according to the table below.

External Interrupt I/O pin selection

ei01	ei00	I/O Pin
0	0	No Interrupt*
0	1	PA1
1	0	PA2
1	1	PA3

* Reset State

Bits 1:0 = Reserved.



11 ON-CHIP PERIPHERALS

11.1 WATCHDOG TIMER (WDG)

11.1.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

11.1.2 Main Features

- Programmable free-running downcounter (64 increments of 16000 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero

- Optional reset on HALT instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte

11.1.3 Functional Description

The counter value stored in the CR register (bits T[6:0]), is decremented every 16000 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically $30\mu s$.



Figure 33. Watchdog Block Diagram

WATCHDOG TIMER (Cont'd)

57

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is freerunning: it counts down even if the watchdog is disabled. The value to be stored in the CR register must be between FFh and C0h (see Table 14 .Watchdog Timing):

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.

Table 14.Watchdog Timing

f _{CPU} = 8MHz								
WDG Counter Code	min [ms]	max [ms]						
C0h	1	2						
FFh	127	128						

Notes: The timing variation shown in Table 14 is due to the unknown status of the prescaler when writing to the CR register.

11.1.4 Hardware Watchdog Option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the CR is not used.

Refer to the Option Byte description in section 15.1 on page 161.

11.1.4.1 Using Halt Mode with the WDG (WDGHALT option)

If Halt mode with Watchdog is enabled by option byte (No watchdog reset on HALT instruction), it is recommended before executing the HALT instruction to refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.

DUAL 12-BIT AUTORELOAD TIMER 3 (Cont'd)

Dead Time Generation

A dead time can be inserted between PWM0 and PWM1 using the DTGR register. This is required for half-bridge driving where PWM signals must not be overlapped. The non-overlapping PWM0/ PWM1 signals are generated through a programmable dead time by setting the DTE bit.

Dead time value = DT[6:0] x Tcounter1

DTGR[7:0] is buffered inside so as to avoid deforming the current PWM cycle. The DTGR effect will take place only after an overflow.

Figure 39. Dead Time Generation

Notes:

1. Dead time is generated only when DTE=1 and DT[6:0] \neq 0. If DTE is set and DT[6:0]=0, PWM output signals will be at their reset state.

2. Half Bridge driving is possible only if polarities of PWM0 and PWM1 are not inverted, i.e. if OP0 and OP1 are not set. If polarity is inverted, overlapping PWM0/PWM1 signals will be generated.



In the above example, when the DTE bit is set:

- PWM goes low at DCR0 match and goes high at ATR1+Tdt
- PWM1 goes high at DCR0+Tdt and goes low at ATR match.

With this programmable delay (Tdt), the PWM0 and PWM1 signals which are generated are not overlapped.



SERIAL PERIPHERAL INTERFACE (cont'd)

SPI CONTROL/STATUS REGISTER (SPICSR)

Read/Write (some bits Read Only) Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF** Serial Peripheral Data Transfer Flag (Read only)

This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register).

- 0: Data transfer is in progress or the flag has been cleared.
- 1: Data transfer between the device and an external device has been completed.

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

Bit 6 = **WCOL** Write Collision status (Read only)

This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 53).

0: No write collision occurred

1: A write collision has been detected

Bit 5 = **OVR** SPI Overrun error (Read only)

This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 11.4.5.2). An interrupt is generated if SPIE = 1 in the SPICR register. The OVR bit is cleared by software reading the SPICSR register. 0: No overrun error

1: Overrun error detected

Bit 4 = **MODF** Mode Fault flag (Read only)

This bit is set by hardware when the \overline{SS} pin is pulled low in master mode (see Section 11.4.5.1 Master Mode Fault (MODF)). An SPI interrupt can be generated if SPIE = 1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF = 1 followed by a write to the SPICR register).

0: No master mode fault detected

1: A fault in master mode has been detected

Bit 3 = Reserved, must be kept cleared.

Bit 2 = SOD SPI Output Disable

This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode) 0: SPI output enabled (if SPE = 1) 1: SPI output disabled

Bit 1 = **SSM** *SS Management*

This bit is set and cleared by software. When set, it disables the alternate function of the SPI SS pin and uses the SSI bit value instead. See Section 11.4.3.2 Slave Select Management.

- 0: Hardware management (SS managed by external pin)
- 1: Software management (internal SS signal controlled by SSI bit. External SS pin free for general-purpose I/O)

Bit 0 = **SSI** *SS* Internal Mode

This bit is set and cleared by software. It <u>acts</u> as a 'chip select' by controlling the level of the SS slave select signal when the SSM bit is set.

0: Slave selected

1: Slave deselected

SPI DATA I/O REGISTER (SPIDR)

Read/Write

Reset Value: Undefined

'							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

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

Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see Figure 48).



Δ

LINSCITM SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)

11.5.5.2 Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. When the M bit is set, word length is 9 bits and the 9th bit (the MSB) has to be stored in the T8 bit in the SCICR1 register.

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 55).

Procedure

- Select the M bit to define the word length.
- Select the desired baud rate using the SCIBRR and the SCIETPR registers.
- Set the TE bit to send a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones (Idle Line) as first transmission.
- Access the SCISR register and write the data to send in the SCIDR register (this sequence clears the TDRE bit). Repeat this sequence for each data to be transmitted.

Clearing the TDRE bit is always performed by the following software sequence:

- 1. An access to the SCISR register
- 2. A write to the SCIDR register

The TDRE bit is set by hardware and it indicates:

- The TDR register is empty.
- The data transfer is beginning.
- The next data can be written in the SCIDR register without overwriting the previous data.

This flag generates an interrupt if the TIE bit is set and the I[1:0] bits are cleared in the CCR register.

When a transmission is taking place, a write instruction to the SCIDR register stores the data in the TDR register and which is copied in the shift register at the end of the current transmission. When no transmission is taking place, a write instruction to the SCIDR register places the data directly in the shift register, the data transmission starts, and the TDRE bit is immediately set.

When a character transmission is complete (after the stop bit) the TC bit is set and an interrupt is generated if the TCIE is set and the I[1:0] bits are cleared in the CCR register.

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

1. An access to the SCISR register

2. A write to the SCIDR register

Note: The TDRE and TC bits are cleared by the same software sequence.

Break Characters

Setting the SBK bit loads the shift register with a break character. The break character length depends on the M bit (see Figure 56).

As long as the SBK bit is set, the SCI sends break characters to the TDO pin. After clearing this bit by software, the SCI inserts a logic 1 bit at the end of the last break character to guarantee the recognition of the start bit of the next character.

Idle Line

Setting the TE bit drives the SCI to send a preamble of 10 (M = 0) or 11 (M = 1) consecutive '1's (idle line) before the first character.

In this case, clearing and then setting the TE bit during a transmission sends a preamble (idle line) after the current word. Note that the preamble duration (10 or 11 consecutive '1's depending on the M bit) does not take into account the stop bit of the previous character.

Note: Resetting and setting the TE bit causes the data in the TDR register to be lost. Therefore the best time to toggle the TE bit is when the TDRE bit is set, that is, before writing the next byte in the SCIDR.

LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)



LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)

EXTENDED RECEIVE PRESCALER DIVISION REGISTER (SCIERPR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ERPR							
7	6	5	4	3	2	1	0

Bits 7:0 = **ERPR[7:0]** 8-bit Extended Receive Prescaler Register

The extended Baud Rate Generator is activated when a value other than 00h is stored in this register. The clock frequency from the 16 divider (see Figure 57) is divided by the binary factor set in the SCIERPR register (in the range 1 to 255).

The extended baud rate generator is not active after a reset.

EXTENDED TRANSMIT PRESCALER DIVISION REGISTER (SCIETPR)

Read/Write

Reset Value:0000 0000 (00h)

7							0
ETPR							
7	6	5	4	3	2	1	0

Bits 7:0 = **ETPR[7:0]** 8-bit Extended Transmit Prescaler Register

The extended Baud Rate Generator is activated when a value other than 00h is stored in this register. The clock frequency from the 16 divider (see Figure 57) is divided by the binary factor set in the SCIETPR register (in the range 1 to 255).

The extended baud rate generator is not active after a reset.

Note: In LIN slave mode, the Conventional and Extended Baud Rate Generators are disabled.



LINSCITM SERIAL COMMUNICATION INTERFACE (LIN Mode) (cont'd)

11.5.9.3 LIN Reception

In LIN mode the reception of a byte is the same as in SCI mode but the LINSCI has features for handling the LIN Header automatically (identifier detection) or semiautomatically (Synch Break detection) depending on the LIN Header detection mode. The detection mode is selected by the LHDM bit in the SCICR3.

Additionally, an automatic resynchronization feature can be activated to compensate for any clock deviation, for more details please refer to Section 11.5.9.5 LIN Baud Rate.

LIN Header Handling by a Slave

Depending on the LIN Header detection method the LINSCI will signal the detection of a LIN Header after the LIN Synch Break or after the Identifier has been successfully received.

Note:

It is recommended to combine the Header detection function with Mute mode. Putting the LINSCI in Mute mode allows the detection of Headers only and prevents the reception of any other characters.

This mode can be used to wait for the next Header without being interrupted by the data bytes of the current message in case this message is not relevant for the application.

Synch Break Detection (LHDM = 0):

When a LIN Synch Break is received:

- The RDRF bit in the SCISR register is set. It indicates that the content of the shift register is transferred to the SCIDR register, a value of 0x00 is expected for a Break.
- The LHDF flag in the SCICR3 register indicates that a LIN Synch Break Field has been detected.
- An interrupt is generated if the LHIE bit in the SCICR3 register is set and the I[1:0] bits are cleared in the CCR register.
- Then the LIN Synch Field is received and measured.
 - If automatic resynchronization is enabled (LA-SE bit = 1), the LIN Synch Field is not transferred to the shift register: There is no need to clear the RDRF bit.
 - If automatic resynchronization is disabled (LA-SE bit = 0), the LIN Synch Field is received as a normal character and transferred to the SCIDR register and RDRF is set.

Note:

In LIN slave mode, the FE bit detects all frame error which does not correspond to a break.

Identifier Detection (LHDM = 1):

This case is the same as the previous one except that the LHDF and the RDRF flags are set only after the entire header has been received (this is true whether automatic resynchronization is enabled or not). This indicates that the LIN Identifier is available in the SCIDR register.

Notes:

During LIN Synch Field measurement, the SCI state machine is switched off: No characters are transferred to the data register.

LIN Slave parity

In LIN Slave mode (LINE and LSLV bits are set) LIN parity checking can be enabled by setting the PCE bit.

In this case, the parity bits of the LIN Identifier Field are checked. The identifier character is recognized as the third received character after a break character (included):



The bits involved are the two MSB positions (7th and 8th bits if M = 0; 8th and 9th bits if M = 0) of the identifier character. The check is performed as specified by the LIN specification:



LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Master/Slave) (Cont'd)

Addr.	Pagiotar Nama	7	6	E	4	2	_	-	0
(Hex.)	Register Name		0	5	4	3	2		U
40	SCISR	TDRE	TC	RDRF	IDLE	OR/LHE	NF	FE	PE
40	Reset Value	1	1	0	0	0	0	0	0
41	SCIDR	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
41	Reset Value	-	-	-	-	-	-	-	-
	SCIBRR	SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0
42	LPR (LIN Slave Mode)	LPR7	LPR6	LPR5	LPR4	LPR3	LPR2	LPR1	LPR0
	Reset Value	0	0	0	0	0	0	0	0
12	SCICR1	R8	T8	SCID	М	WAKE	PCE	PS	PIE
43	Reset Value	x	0	0	0	0	0	0	0
11	SCICR2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
44	Reset Value	0	0	0	0	0	0	0	0
45	SCICR3	NP	LINE	LSLV	LASE	LHDM	LHIE	LHDF	LSF
45	Reset Value	0	0	0	0	0	0	0	0
	SCIERPR	ERPR7	ERPR6	ERPR5	ERPR4	ERPR3	ERPR2	ERPR1	ERPR0
46	LHLR (LIN Slave Mode)	LHL7	LHL6	LHL5	LHL4	LHL3	LHL2	LHL1	LHL0
	Reset Value	0	0	0	0	0	0	0	0
	SCITPR	ETPR7	ETPR6	ETPR5	ETPR4	ETPR3	ETPR2	ETPR1	ETPR0
47	LPFR (LIN Slave Mode)	LDUM	0	0	0	LPFR3	LPFR2	LPFR1	LPFR0
	Reset Value	0	0	0	0	0	0	0	0

Table 21. LINSCI1 Register Map and Reset Values



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

11.6.3.2 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.

11.6.3.3 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:

- Select the CS[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 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.

11.6.3.4 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.

11.6.4 Low Power Modes

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.
	After wakeup from Halt mode, the A/D
HALT	Converter requires a stabilization time
	t _{STAB} (see Electrical Characteristics)
	before accurate conversions can be
	performed.

11.6.5 Interrupts

None.

ST7 ADDRESSING MODES (cont'd)

12.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two submodes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 24. InstructionsSupportingDirect,Indexed,IndirectandIndirectIndexedAddressingModesIndirectIndexed

Long and Short Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Addition/subtrac- tion operations
BCP	Bit Compare

Short Instructions Only	Function				
CLR	Clear				
INC, DEC	Increment/Decrement				
TNZ	Test Negative or Zero				
CPL, NEG	1 or 2 Complement				
BSET, BRES	Bit Operations				
BTJT, BTJF	Bit Test and Jump Opera- tions				
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations				
SWAP	Swap Nibbles				
CALL, JP	Call or Jump subroutine				

57

12.1.7 Relative Mode (Direct, Indirect)

This addressing mode is used to modify the PC register value by adding an 8-bit signed offset to it.

Available Relative Direct/ Indirect Instructions	Function				
JRxx	Conditional Jump				
CALLR	Call Relative				

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset follows the opcode.

Relative (Indirect)

The offset is defined in memory, of which the address follows the opcode.

INSTRUCTION GROUPS (cont'd)

57

Mnemo	Description	Function/Example	Dst	Src	н	Ι	Ν	Z	С
ADC	Add with Carry	A = A + M + C	А	М	Н		Ν	Z	С
ADD	Addition	A = A + M	А	М	Н		Ν	Z	С
AND	Logical And	A = A . M	А	М			Ν	Z	
BCP	Bit compare A, Memory	tst (A . M)	А	М			Ν	Z	
BRES	Bit Reset	bres Byte, #3	М						
BSET	Bit Set	bset Byte, #3	М						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	М						С
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	М						С
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
СР	Arithmetic Compare	tst(Reg - M)	reg	М			Ν	Z	С
CPL	One Complement	A = FFH-A	reg, M				Ν	Z	1
DEC	Decrement	dec Y	reg, M				Ν	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			Н	Ι	Ν	Z	С
INC	Increment	inc X	reg, M				Ν	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if $(C + Z = 0)$	Unsigned >							

OPERATING CONDITIONS (Cont'd)

13.3.1.2 Devices with tested for T_A = -40 to +125°C @ V_{DD} = 3.0 to 3.6V

Symbol	Parameter	Conditions	Min	Тур	Max	Unit
£ 1)	Internal RC oscillator fre-	RCCR = FF (reset value), T_A =25°C, V_{DD} = 3.3V		630		kU-
IRC /	quency	RCCR=RCCR1 ²⁾ ,T _A =25°C,V _{DD} = 3.3V	995	1000	1005	КПZ
	Accuracy of Internal RC	T _A =25°C, V _{DD} =3.0 to 3.6V	-1		+1	
ACC _{RC}	oscillator when calibrated	T_A =-40 to +85°C, V_{DD} =3.0 to 3.6V	-3		+3	%
	with RCCR=RCCR1 ²⁾³⁾	T _A =-40 to +125°C, V _{DD} =3.0 to 3.6V	-3		+3	
I _{DD(RC)}	RC oscillator current con- sumption	T _A =25°C,V _{DD} =3.3V		500 ³⁾⁴⁾		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			1		MHz
t _{LOCK}	PLL Lock time ⁷⁾			2		ms
t _{STAB}	PLL Stabilization time ⁷⁾			4		ms
100	v4 PLL Acoursov	$f_{RC} = 1MHz@T_A=25°C, V_{DD}=2.7 \text{ to } 3.3V$		0.1 ⁶⁾		%
ACCPLL	X4 FLL Accuracy	$f_{RC} = 1MHz@T_A = -40 \text{ to } +125^{\circ}C, V_{DD} = 3.3V$		0.1 ⁶⁾		%
t _{w(JIT)}	PLL jitter period	f _{RC} = 1MHz		8 ⁵⁾		kHz
JIT _{PLL}	PLL jitter (∆f _{CPU} /f _{CPU})			1 ⁵⁾		%
I _{DD(PLL)}	PLL current consumption	T _A =25°C		450 ³⁾		μA

Notes:

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.
- $\ensuremath{\textbf{3.}}$ Data based on characterization results, not tested in production
- 4. Measurement made with RC calibrated at 1MHz.

5. Guaranteed by design.

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

7. After the LOCKED bit is set ACC_{PLL} is max. 10% until t_{STAB} has elapsed. See Figure 12 on page 24.





Figure 107. 20-Pin Plastic Dual In-Line Package, 300-mil Width

14.2 THERMAL CHARACTERISTICS

Symbol	Ratings		Value	Unit
B	Package thermal resistance	DIP20	63	°C/M
hthJA	(junction to ambient)	SO20	76	0/10
T _{Jmax}	Maximum junction temperature ¹⁾		150	°C
P _{Dmax}	Maximum neuror discinction 2	DIP20	400	m\//
		SO20	330	11100

Notes:

1. The maximum chip-junction temperature is based on technology characteristics.

2. The maximum power dissipation is obtained from the formula $P_D = (T_J - T_A) / R_{thJA}$. The power dissipation of an application is defined by the user with the formula: $P_D = P_{INT} + P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD} \times V_{DD}$) and P_{PORT} is the port power dissipation depending on the ports used in the application.



15.2 DEVICE ORDERING INFORMAT ION AND TRANSFER OF CUSTOMER CODE

Customer code is made up of the FASTROM con-ing the correctly completed OPTION LIST appendtents and the list of the selected options (if any) onpage 164

The FASTROM contents are to be sent on dis-Refer to application note AN1635 for information kette, or by electronic means, with the S19 hexan the counter listing returned by ST after code decimal file generated by the development tool. All as been transferred.

unused bytes must be set to FFh. The selected options are communicated to STMicroelectronics us the STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Figure 108. Ordering information scheme

Example:	ST7	F	LITE3x	F	2	U	3	TR	
Family ST7 Microcontroller Family									
Memory type F: Flash P: FASTROM									
Sub-family LITE30, LITE35 or LITE39									
No. of pins F = 20									
Memory size 2 = 8K									
Package B = DIP M = SO U= QFN									
lemperature range 6 = -40 °C to 85 °C 3 = -40 °C to 125 °C]		
Shipping Option TR = Tape & Reel packing Blank = Tube (DIP20 or SO20)	or Tray (C	2FN2C))						
For a list of available options further information on any asp	(e.g. data ect of this	EEPR d ev ec	OM, packaç asje contact	ge) an t the S	d ord	erable es Off	e part fice ne	: numb earest	ers or to you

5/