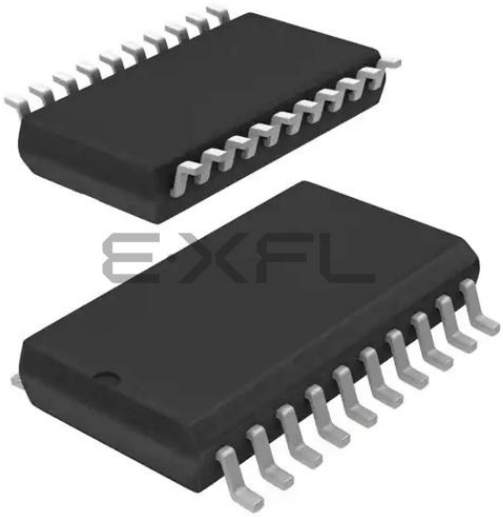


Welcome to E-XFL.COM

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	Not For New Designs
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 x 8)
Program Memory Type	FLASH
EEPROM Size	-
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/st7flite35f2m6

Table of Contents

PIN DESCRIPTION (Cont'd)

Legend / Abbreviations for Table 2:

Type: I = input, O = output, S = supply

In/Output level: C_T= CMOS $0.3V_{DD}/0.7V_{DD}$ with input trigger

Output level: HS = 20mA high sink (on N-buffer only)

Port and control configuration:

- Input: float = floating, wpu = weak pull-up, int = interrupt, ana = analog
- Output: OD = open drain, PP = push-pull

The RESET configuration of each pin is shown in bold which is valid as long as the device is in reset state.

Table 2. Device Pin Description

				Level		Port / Control						Main Function (after reset)	Alternate Function
						Input				Output			
QFN20	SO20/DIP20	Pin Name	Type	Input	Output	float	wpu	int	ana	OD	PP		
19	1	V_{SS} [1]	S									Ground	
20	2	V_{DD} [1]	S									Main power supply	
1	3	$\overline{\text{RESET}}$	I/O	C_T			X			X		Top priority non maskable interrupt (active low)	
2	4	PB0/AIN0/$\overline{\text{SS}}$	I/O	C_T		**X**	ei3		X	X	X	**Port B0**	ADC Analog Input 0 or SPI Slave Select (active low) **Caution:** No negative current injection allowed on this pin. For details, refer to section 13.2.2 on page 132
3	5	PB1/AIN1/SCK	I/O	C_T		**X**			X	X	X	**Port B1**	ADC Analog Input 1 or SPI Serial Clock **Caution:** No negative current injection allowed on this pin. For details, refer to section 13.2.2 on page 132
4	6	PB2/AIN2/ MISO	I/O	C_T		**X**			X	X	X	**Port B2**	ADC Analog Input 2 or SPI Master In/ Slave Out Data
5	7	PB3/AIN3/ MOSI	I/O	C_T		**X**	ei2		X	X	X	**Port B3**	ADC Analog Input 3 or SPI Master Out / Slave In Data
6	8	PB4/AIN4/ CLKIN**	I/O	C_T		**X**	X		X	X	X	**Port B4**	ADC Analog Input 4 or External clock input
7	9	PB5/AIN5	I/O	C_T		**X**	ei2		X	X	X	**Port B5**	ADC Analog Input 5
8	10	PB6/AIN6/RDI	I/O	C_T		**X**			X	X	X	**Port B6**	ADC Analog Input 6 or LINSCI Input
9	11	PA7/TDO	I/O	C_T	HS	**X**	X			X	X	**Port A7**	LINSCI Output

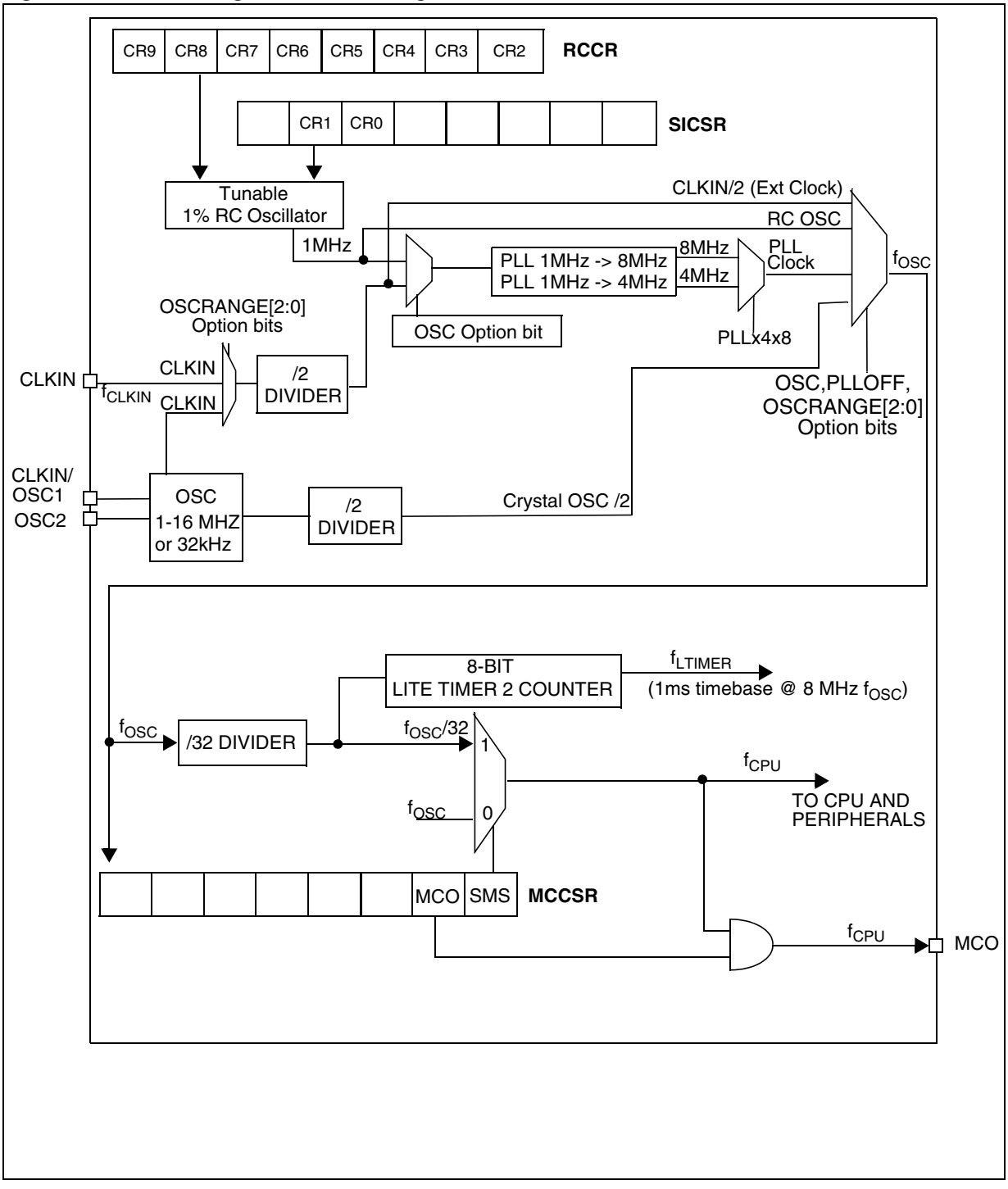
QFN20	SO20/DIP20	Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
						Input				Output			
				Input	Output	float	wpu	int	ana	OD	PP		
10	12	PA6 /MCO/ ICCCLK/ BREAK	I/O	C_T		X	ei1			X	X	**Port A6**	Main Clock Output or In Circuit Communication Clock or External BREAK **Caution:** During normal operation this pin must be pulled- up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset will put it back in input pull-up.
11	13	PA5 /ATPWM3/ ICCDATA	I/O	C_T	HS	**X**				X	X	**Port A5**	Auto-Reload Timer PWM3 or In Circuit Communication Data
12	14	PA4/ATPWM2	I/O	C_T	HS	**X**				X	X	**Port A4**	Auto-Reload Timer PWM2
13	15	PA3/ATPWM1	I/O	C_T	HS	**X**	ei0			X	X	**Port A3**	Auto-Reload Timer PWM1
14	16	PA2/ATPWM0	I/O	C_T	HS	**X**				X	X	**Port A2**	Auto-Reload Timer PWM0
15	17	PA1/ATIC	I/O	C_T	HS	**X**				X	X	**Port A1**	Auto-Reload Timer Input Capture
16	18	PA0/LTIC	I/O	C_T	HS	**X**	X			X	X	**Port A0**	Lite Timer Input Capture
17	19	OSC2	O									Resonator oscillator inverter output	
18	20	OSC1/CLKIN	I									Resonator oscillator inverter input or External clock input	

Notes:

1. It is mandatory to connect all available V_{DD} and V_{DDA} pins to the supply voltage and all V_{SS} and V_{SSA} pins to ground.

2. For input with interrupt possibility "ei_x" defines the associated external interrupt vector which can be assigned to one of the I/O pins using the EISR register. Each interrupt can be either weak pull-up or floating defined through option register OR.

Figure 13. Clock Management Block Diagram



SYSTEM INTEGRITY MANAGEMENT (Cont'd)

7.6.3 Low Power Modes

Mode	Description
WAIT	No effect on SI. AVD interrupts cause the device to exit from Wait mode.
HALT	The SICSR register is frozen. The AVD becomes inactive and the AVD interrupt cannot be used to exit from Halt mode.

7.6.3.1 Interrupts

The AVD interrupt event generates an interrupt if the corresponding Enable Control Bit (AVDIE) is set and the interrupt mask in the CC register is reset (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
AVD event	AVDF	AVDIE	Yes	No

POWER SAVING MODES (Cont'd)

9.4 HALT MODE

The HALT mode is the lowest power consumption mode of the MCU. It is entered by executing the 'HALT' instruction when ACTIVE-HALT is disabled (see section 9.5 on page 42 for more details) and when the AWUEN bit in the AWUCSR register is cleared.

The MCU can exit HALT mode on reception of either a specific interrupt (see Table 6, "Interrupt Mapping," on page 36) or a RESET. When exiting HALT mode by means of a RESET or an interrupt, the oscillator is immediately turned on and the 256 CPU cycle delay is used to stabilize the oscillator. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 25).

When entering HALT mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In HALT mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

The compatibility of Watchdog operation with HALT mode is configured by the "WDGHALT" option bit of the option byte. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET (see section 15.1 on page 161 for more details).

Figure 24. HALT Timing Overview

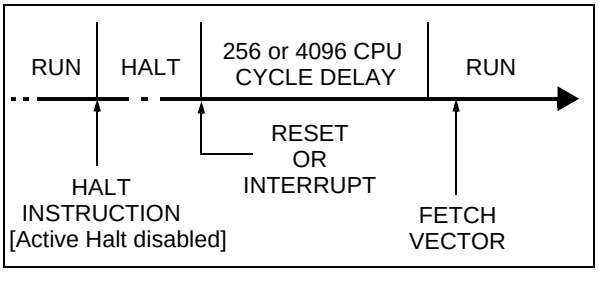
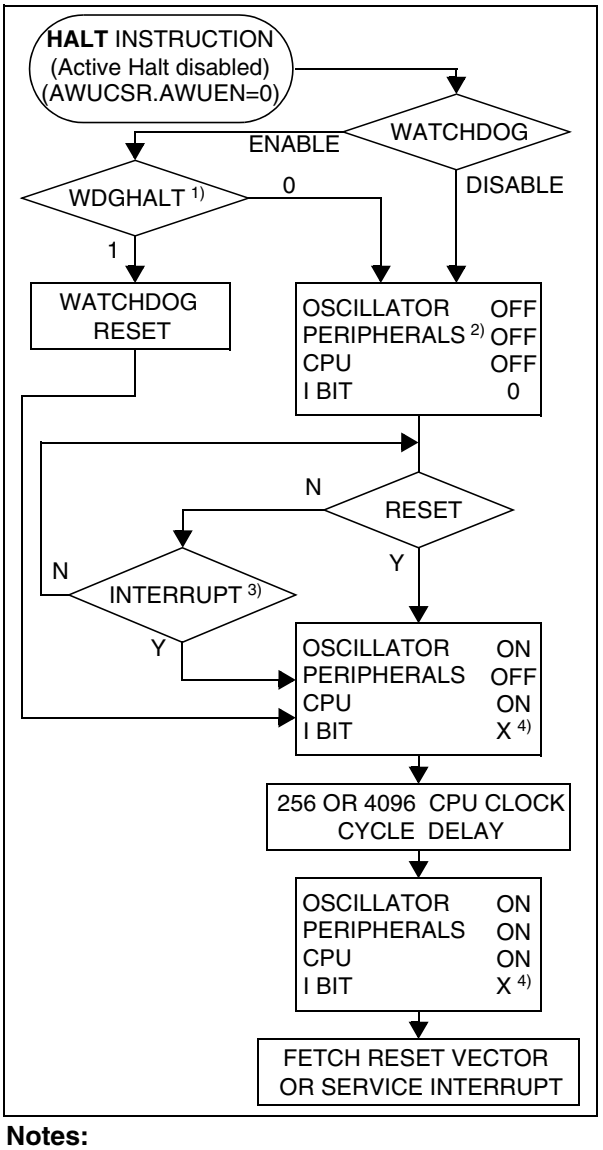


Figure 25. HALT Mode Flow-chart



Notes:

1. WDGHALT is an option bit. See option byte section for more details.

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

3. Only some specific interrupts can exit the MCU from HALT mode (such as external interrupt). Refer to Table 6, "Interrupt Mapping," on page 36 for more details.

4. 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.

POWER SAVING MODES (Cont'd)

Similarities with Halt mode

The following AWUFH mode behaviour is the same as normal Halt mode:

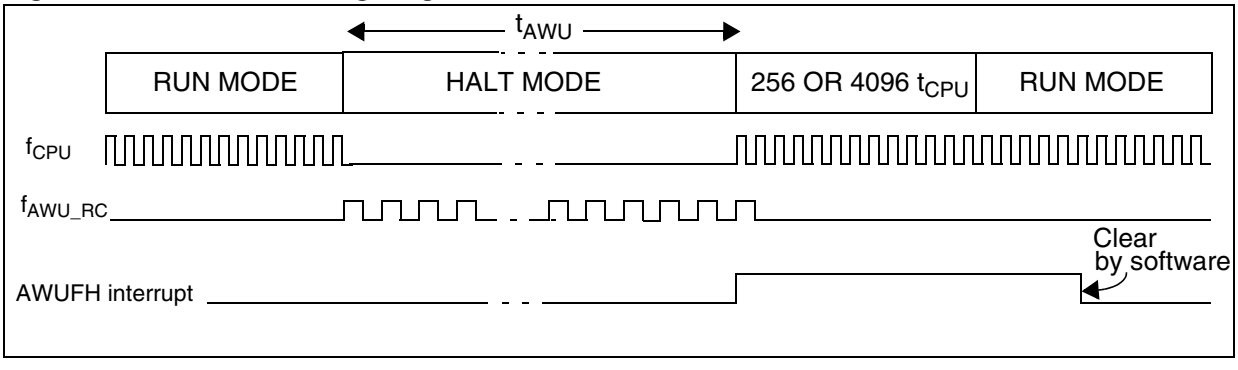
– The MCU can exit AWUFH mode by means of any interrupt with exit from Halt capability or a reset (see Section 9.4 HALT MODE).

– When entering AWUFH mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

– In AWUFH mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. None of the peripherals are clocked except those which get their clock supply from another clock generator (such as an external or auxiliary oscillator like the AWU oscillator).

– The compatibility of Watchdog operation with AWUFH mode is configured by the WDGHALT option bit in the option byte. Depending on this setting, the HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET.

Figure 29. AWUF Halt Timing Diagram



WATCHDOG TIMER (Cont'd)

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is free-running: 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)

11.2.3.3 Input Capture Mode

The 12-bit ATICR register is used to latch the value of the 12-bit free running upcounter CNTR1 after a rising or falling edge is detected on the ATIC pin. When an input capture occurs, the ICF bit is set and the ATICR register contains the value of the free running upcounter. An IC interrupt is generated if the ICIE bit is set. The ICF bit is reset by reading the ATICRH/ATICRL register when the ICF bit is set. The ATICR is a read only register and always contains the free running upcounter value which corresponds to the most recent input capture. Any further input capture is inhibited while the ICF bit is set.

Figure 42. Block Diagram of Input Capture Mode

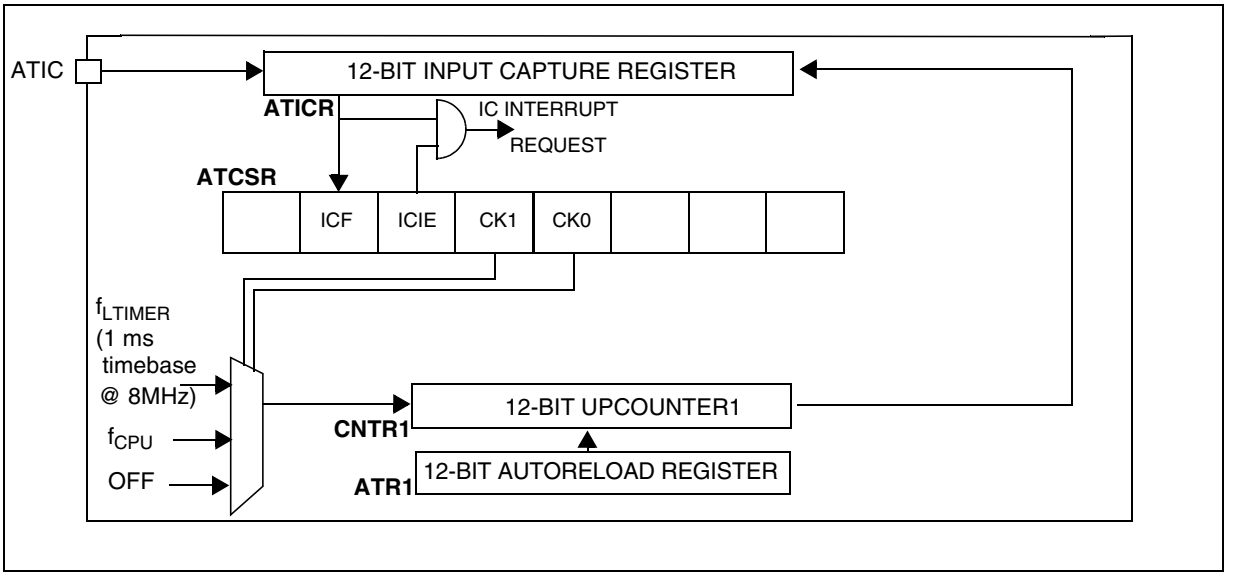
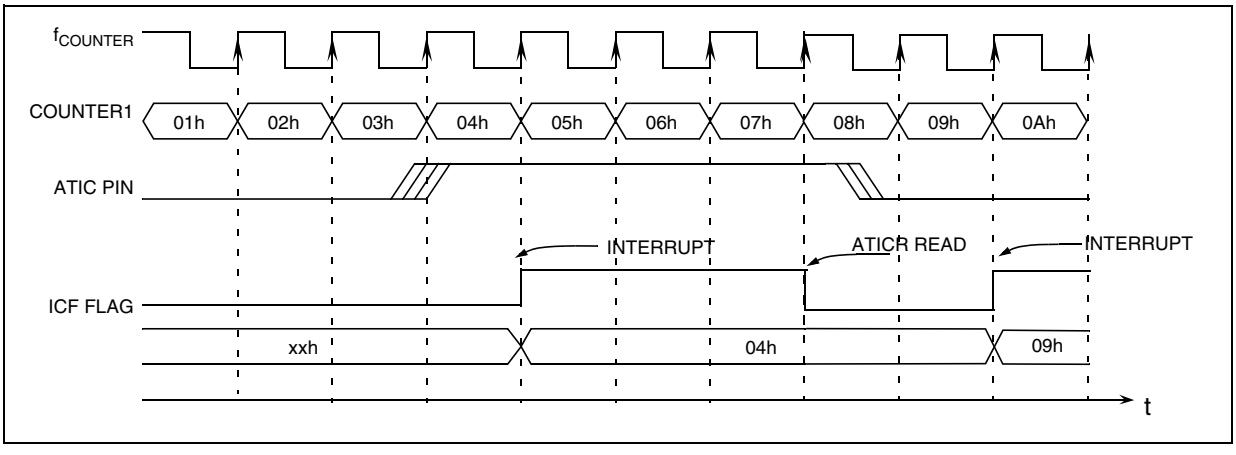


Figure 43. Input Capture timing diagram

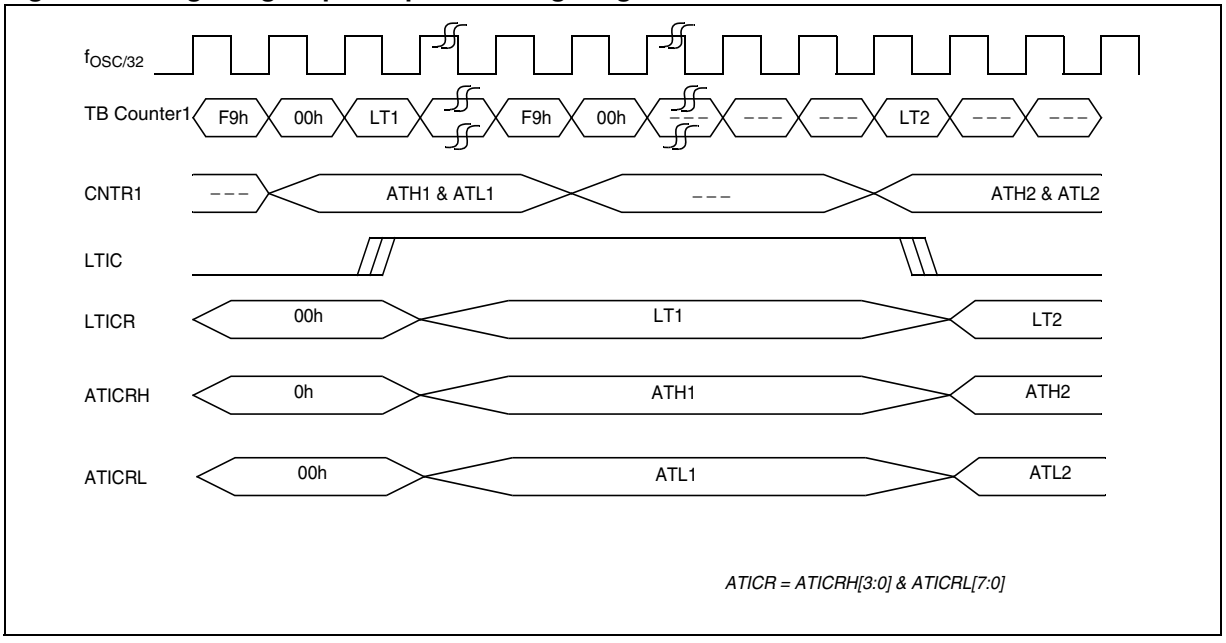


DUAL 12-BIT AUTORELOAD TIMER 3 (Cont'd)

– At the second input capture on the falling edge of the pulse, we assume that the values in the registers are as follows:
LTICR = LT2
ATICRH = ATH2
ATICRL = ATL2
Hence ATICR2 [11:0] = ATH2 & ATL2

Now pulse width P between first capture and second capture will be:
P = decimal (F9 – LT1 + LT2 + 1) * 0.004ms + decimal (ATICR2 - ATICR1 – 1) * 1ms

Figure 45. Long Range Input Capture Timing Diagram



11.2.4 Low Power Modes

Mode	Description
SLOW	The input frequency is divided by 32
WAIT	No effect on AT timer
ACTIVE-HALT	AT timer halted except if CK0=1, CK1=0 and OVFIE=1
HALT	AT timer halted.

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

11.2.5 Interrupts

Interrupt Event [1]	Event Flag	Enable Control Bit	Exit from WAIT	Exit from HALT	Exit from ACTIVE-HALT
Overflow Event	OVF1	OVIE1	Yes	No	Yes[2]
AT3 IC Event	ICF	ICIE	Yes	No	No
CMP Event	CMPFx	CMPIE	Yes	No	No

Note 1: The CMP and AT3 IC events are connected to the same interrupt vector.
The OVF event is mapped on a separate vector (see Interrupts chapter).
They generate an interrupt if the enable bit is set in the ATCSR register and the interrupt mask in the CC register is reset (RIM instruction).

Note 2: Only if CK0=1 and CK1=0 ($f_{COUNTER} = f_{LTIMER}$)

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

11.5.5.6 Receiver Muting and Wake-up Feature

In multiprocessor configurations it is often desirable that only the intended message recipient should actively receive the full message contents, thus reducing redundant SCI service overhead for all non-addressed receivers.

The non-addressed devices may be placed in sleep mode by means of the muting function.

Setting the RWU bit by software puts the SCI in sleep mode:

All the reception status bits can not be set.

All the receive interrupts are inhibited.

A muted receiver may be woken up in one of the following ways:

– by Idle Line detection if the WAKE bit is reset,

– by Address Mark detection if the WAKE bit is set.

Idle Line Detection

Receiver wakes up by Idle Line detection when the Receive line has recognized an Idle Line. Then the RWU bit is reset by hardware but the IDLE bit is not set.

This feature is useful in a multiprocessor system when the first characters of the message determine the address and when each message ends by an idle line: As soon as the line becomes idle, every receivers is waken up and analyse the first characters of the message which indicates the addressed receiver. The receivers which are not addressed set RWU bit to enter in mute mode. Consequently, they will not treat the next characters constituting the next part of the message. At the end of the message, an idle line is sent by the transmitter: this wakes up every receivers which are ready to analyse the addressing characters of the new message.

In such a system, the inter-characters space must be smaller than the idle time.

Address Mark Detection

Receiver wakes up by Address Mark detection when it received a "1" as the most significant bit of a word, thus indicating that the message is an address. The reception of this particular word wakes up the receiver, resets the RWU bit and sets the RDRF bit, which allows the receiver to receive this word normally and to use it as an address word.

This feature is useful in a multiprocessor system when the most significant bit of each character (except for the break character) is reserved for Address Detection. As soon as the receivers received an address character (most significant bit = '1'), the receivers are waken up. The receivers which are not addressed set RWU bit to enter in mute mode. Consequently, they will not treat the next characters constituting the next part of the message.

11.5.5.7 Parity Control

Hardware byte Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the character format defined by the M bit, the possible SCI character formats are as listed in Table 20.

Note: In case of wake-up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Table 20. Character Formats

M bit	PCE bit	Character format
0	0	ǀ SB ǀ 8 bit data ǀ STB ǀ
	1	ǀ SB ǀ 7-bit data ǀ PB ǀ STB ǀ
1	0	ǀ SB ǀ 9-bit data ǀ STB ǀ
	1	ǀ SB ǀ 8-bit data ǀ PB ǀ STB ǀ

Legend: SB = Start Bit, STB = Stop Bit, PB = Parity Bit

Even parity: The parity bit is calculated to obtain an even number of "1s" inside the character made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit will be 0 if even parity is selected (PS bit = 0).

Odd parity: The parity bit is calculated to obtain an odd number of "1s" inside the character made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit will be 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an even number of "1s" if even parity is selected (PS = 0) or an odd number of "1s" if odd parity is selected (PS = 1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PCIE is set in the SCICR1 register.

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

11.5.9.4 LIN Error Detection

LIN Header Error Flag

The LIN Header Error Flag indicates that an invalid LIN Header has been detected.

When a LIN Header Error occurs:

– The LHE flag is set

– An interrupt is generated if the RIE bit is set and the I[1:0] bits are cleared in the CCR register.

If autosynchronization is enabled (LASE bit = 1), this can mean that the LIN Synch Field is corrupted, and that the SCI is in a blocked state (LSF bit is set). The only way to recover is to reset the LSF bit and then to clear the LHE bit.

– The LHE bit is reset by an access to the SCISR register followed by a read of the SCIDR register.

LHE/OVR Error Conditions

When Auto Resynchronization is disabled (LASE bit = 0), the LHE flag detects:

– That the received LIN Synch Field is not equal to 55h.

– That an overrun occurred (as in standard SCI mode)

– Furthermore, if LHDM is set it also detects that a LIN Header Reception Timeout occurred (only if LHDM is set).

When the LIN auto-resynchronization is enabled (LASE bit = 1), the LHE flag detects:

– That the deviation error on the Synch Field is outside the LIN specification which allows up to +/-15.5% of period deviation between the slave and master oscillators.

– A LIN Header Reception Timeout occurred. If $T_{HEADER} > T_{HEADER\_MAX}$ then the LHE flag is set. Refer to Figure 60. (only if LHDM is set to 1)

– An overflow during the Synch Field Measurement, which leads to an overflow of the divider registers. If LHE is set due to this error then the SCI goes into a blocked state (LSF bit is set).

– That an overrun occurred on Fields other than the Synch Field (as in standard SCI mode)

Deviation Error on the Synch Field

The deviation error is checking by comparing the current baud rate (relative to the slave oscillator) with the received LIN Synch Field (relative to the master oscillator). Two checks are performed in parallel:

– The first check is based on a measurement between the first falling edge and the last falling edge of the Synch Field. Let us refer to this period deviation as D:

If the LHE flag is set, it means that:

D > 15.625%

If LHE flag is not set, it means that:

D < 16.40625%

If 15.625% ≤ D < 16.40625%, then the flag can be either set or reset depending on the dephasing between the signal on the RDI line and the CPU clock.

– The second check is based on the measurement of each bit time between both edges of the Synch Field: this checks that each of these bit times is large enough compared to the bit time of the current baud rate.

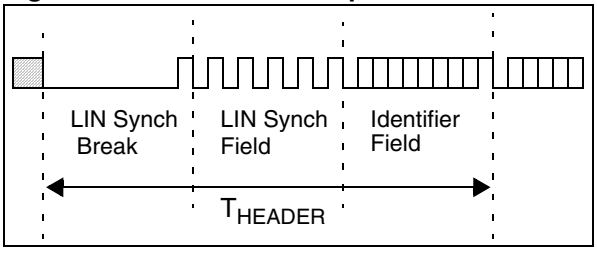
When LHE is set due to this error then the SCI goes into a blocked state (LSF bit is set).

LIN Header Time-out Error

When the LIN Identifier Field Detection Method is used (by configuring LHDM to 1) or when LIN auto-resynchronization is enabled (LASE bit = 1), the LINSCI automatically monitors the T_{HEADER_MAX} condition given by the LIN protocol.

If the entire Header (up to and including the STOP bit of the LIN Identifier Field) is not received within the maximum time limit of 57 bit times then a LIN Header Error is signalled and the LHE bit is set in the SCISR register.

Figure 60. LIN Header Reception Timeout



The time-out counter is enabled at each break detection. It is stopped in the following conditions:
- A LIN Identifier Field has been received
- An LHE error occurred (other than a timeout error).
- A software reset of LSF bit (transition from high to low) occurred during the analysis of the LIN Synch Field or

If LHE bit is set due to this error during the LIN Synchr Field (if LASE bit = 1) then the SCI goes into a blocked state (LSF bit is set).

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

11.5.9.5 LIN Baud Rate

Baud rate programming is done by writing a value in the LPR prescaler or performing an automatic resynchronization as described below.

Automatic Resynchronization

To automatically adjust the baud rate based on measurement of the LIN Synch Field:

– Write the nominal LIN Prescaler value (usually depending on the nominal baud rate) in the LPFR / LPR registers.

– Set the LASE bit to enable the Auto Synchronization Unit.

When Auto Synchronization is enabled, after each LIN Synch Break, the time duration between five falling edges on RDI is sampled on f_{CPU} and the result of this measurement is stored in an internal 15-bit register called SM (not user accessible) (see Figure 61). Then the LDIV value (and its associated LPFR and LPR registers) are automatically updated at the end of the fifth falling edge. During LIN Synch field measurement, the SCI state machine is stopped and no data is transferred to the data register.

11.5.9.6 LIN Slave Baud Rate Generation

In LIN mode, transmission and reception are driven by the LIN baud rate generator

Note: LIN Master mode uses the Extended or Conventional prescaler register to generate the baud rate.

If LINE bit = 1 and LSLV bit = 1 then the Conventional and Extended Baud Rate Generators are disabled: the baud rate for the receiver and transmitter are both set to the same value, depending on the LIN Slave baud rate generator:

$$Tx = Rx = \frac{f_{CPU}}{(16*LDIV)}$$

with:

LDIV is an unsigned fixed point number. The mantissa is coded on 8 bits in the LPR register and the fraction is coded on 4 bits in the LPFR register.

If LASE bit = 1 then LDIV is automatically updated at the end of each LIN Synch Field.

Three registers are used internally to manage the auto-update of the LIN divider (LDIV):

- LDIV_NOM (nominal value written by software at LPR/LPFR addresses)

- LDIV_MEAS (results of the Field Synch measurement)

- LDIV (used to generate the local baud rate)

The control and interactions of these registers, explained in Figure 62 and Figure 63, depend on the LDUM bit setting (LIN Divider Update Method).

Note:

As explained in Figure 62 and Figure 63, LDIV can be updated by two concurrent actions: a transfer from LDIV_MEAS at the end of the LIN Sync Field and a transfer from LDIV_NOM due to a software write of LPR. If both operations occur at the same time, the transfer from LDIV_NOM has priority.

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

Figure 62. LDIV Read / Write Operations When LDUM = 0

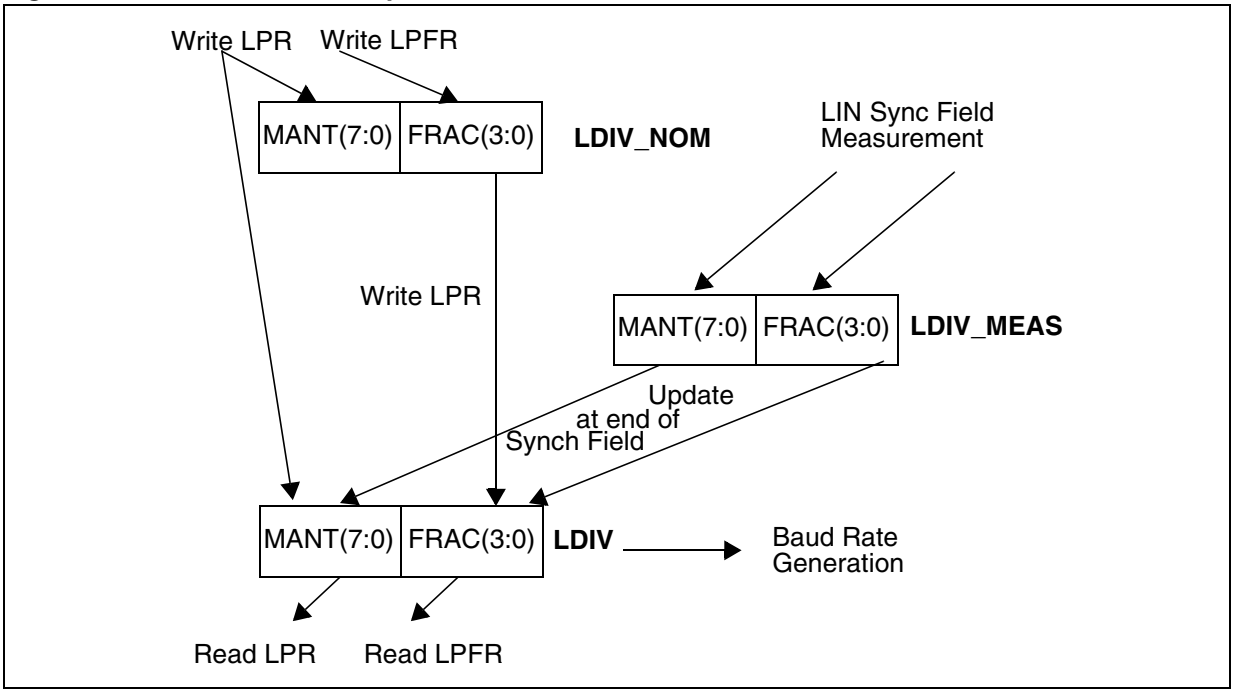
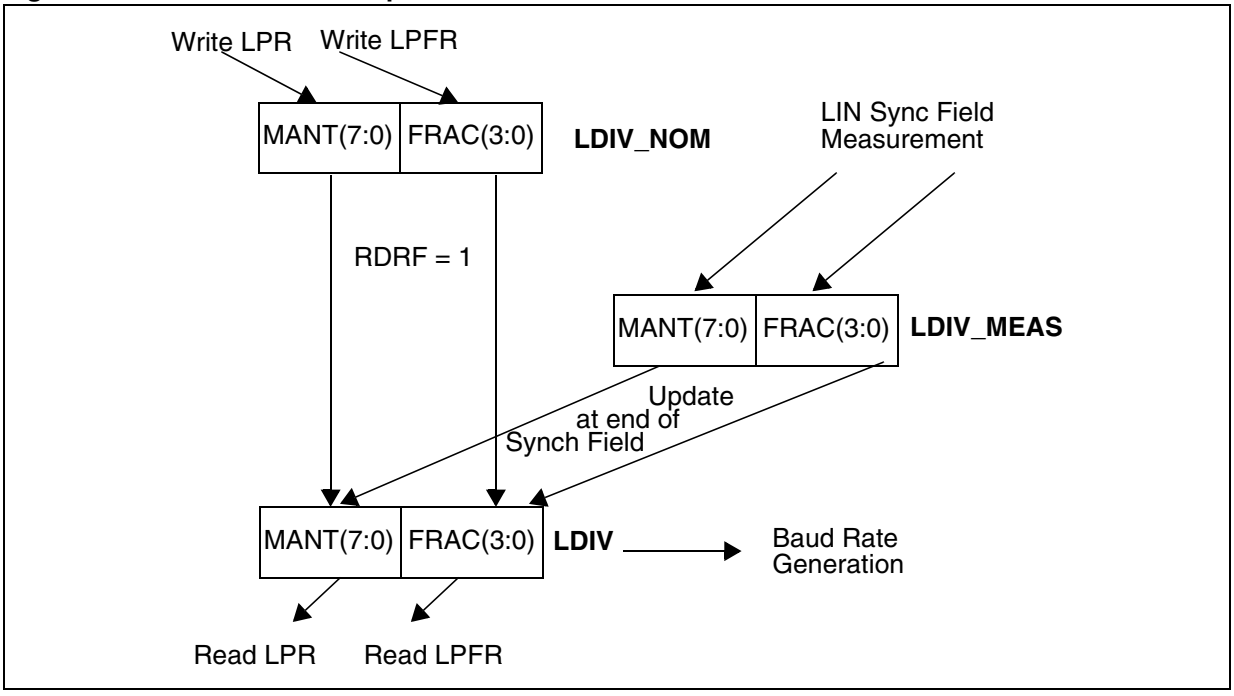


Figure 63. LDIV Read / Write Operations When LDUM = 1



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

11.6.6 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.

Bit 5 = **ADON** *A/D Converter on*
This bit is set and cleared by software.
0: A/D converter is switched off
1: A/D converter is switched on

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

Bit 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
AIN0	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: 0000 0000 (00h)

7							0
D9	D8	D7	D6	D5	D4	D3	D2

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

CONTROL AND DATA REGISTER LOW (ADCDRL)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	SLOW	0	D1	D0

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

Bit 4 = Reserved. Forced by hardware to 0.

Bit 3 = **SLOW** *Slow mode*

This bit is set and cleared by software. It is used together with the SPEED bit 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	x

Bit 2 = Reserved. Forced by hardware to 0.

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

Table 22. ADC Register Map and Reset Values

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

15.2 DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE

Customer code is made up of the FASTROM contents and the list of the selected options (if any). The FASTROM contents are to be sent on diskette, or by electronic means, with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh. The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended on page 164.

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred.

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

Temperature 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 (QFN20)

For a list of available options (e.g. data EEPROM, package) and orderable part numbers or for further information on any aspect of this device, please contact the ST Sales Office nearest to you.

IMPORTANT NOTES (Cont'd)

Impact on application

Software may execute the interrupt routine twice after header reception.

Moreover, in reception mode, as the receiver is no longer in mute mode, an interrupt will be generated on each data byte reception.

Workaround

The problem can be detected in the LINSCI interrupt routine. In case of timeout error (LHE is set and LHLR is loaded with 00h), the software can check the RWU bit in the SCICR2 register. If RWU is cleared, it can be set by software. Refer to Figure 110. Workaround is shown in bold characters.

Figure 110. LINSCI Interrupt routine

```
@interrupt void LINSCI_IT ( void ) /* LINSCI interrupt routine */
{
      /* clear flags */
      SCISR_buffer = SCISR;
      SCIDR_buffer = SCIDR;

      if ( SCISR_buffer & LHE )/* header error ? */
      {
            if (!LHLR)/* header time-out? */
            {
                  if ( !(SCICR2 & RWU) )/* active mode ? */
                  {
                        _asm("sim");/* disable interrupts */
                        SCISR;
                        SCIDR;/* Clear RDRF flag */
                        SCICR2 |= RWU;/* set mute mode */
                        SCISR;
                        SCIDR;/* Clear RDRF flag */
                        SCICR2 |= RWU;/* set mute mode */
                        _asm("rim");/* enable interrupts */
                  }
            }
      }
}
```

Example using Cosmic compiler syntax

17 REVISION HISTORY

Date	Revision	Main changes
29-Jul-05	4	First release on Internet
20-Jul-06	5	Added QFN20 package In Table 3, “Hardware Register Map,” on page 10, replaced h by b for LTCSR1, ATCSR and SICSR reset values section 4.4 on page 13 Modified section 4.4 on page 13 (and modified note 6) and Figure 5 Modified section 5.3 on page 16 (read operation description) Modified note to Figure 8 on page 17 Modified 3rd paragraph and modified text in section 5.5 on page 18 Modified note on clock stability in section 7.1 on page 23 and added one note on internal RC Added illegal opcode references to Section 7.5.1 and Figure 15 on page 28 Modified Figure 18 on page 31 (SICSR register) Modified SICSR reset value in Table 3, “Hardware Register Map,” on page 10, in Figure 18 on page 31 and in section 7.6.4 on page 34 Modified caution in section 8.2 on page 34 Modified External Interrupt Function section in section 10.2.1 on page 47 Modified section 11.4.3.3 on page 82 and added important note Modified section 11.5.5.2 on page 93: changed paragraph “When a character transmission is complete...” Modified values in section 13.2.2 on page 132 Added note 1 and modified note 2 to section 13.3.1.1 on page 134 and to section 13.3.1.2 on page 136 Updated Section 13.3.1.1 and section 13.3.1.2 on page 136 (f_{RC}, ACC_{RC},f_{PLL}) Modified $V_{DD(x4PLL)}$ values in section 13.3.4 on page 139 Added note 2 to section 13.3.2 on page 139 Modified section 13.4.1 on page 140 and added values for V_{DD}=3.3V Added section 13.5.3 on page 144 $V_{ESD(HBM)}$ value updated in section 13.7.3 on page 147, Table in section 13.7.3 on page 147 updated (I_S) Modified note 2 in section 13.8.1 on page 148 Figures in section 13.8.2 on page 150 updated Modified Figure 98 and removed EMC protective circuitry in Figure 99 on page 154 (device works correctly without these components) Modified section 13.10.1 on page 155 ($t_{su(\overline{SS})}$, $t_{v(MO)}$ and $t_{h(MO)}$), added note 4 and added note 1 to several values Modified Figure 101 (CPHA=1) and Figure 102 on page 156 ($t_{v(MO)}$, $t_{h(MO)}$) ADC accuracy data expanded in section 13.11 on page 157 Added ECOPACK information in section 14.1 on page 159 Modified section 14.2 on page 160 and notes Modified section 14.3 on page 161 Added notes to OSC option bit in section 15.1 on page 161 Modified supported part number in Section 15.2 and option list on page 164 Updated section 15.3 on page 165 Added section 15.4 on page 166 Removed LINSCI wrong break duration in section 16 on page 169
21-Sept-06	6	Removed QFN20 package pinout and mechanical data Modified description of CNTR[11:0] bits in section 11.2.6 on page 67 Modified Table 26 on page 163 (QFN20 in grey) Added “External Clock Source” on page 143 and Figure 83 on page 143 Modified option list on page 164