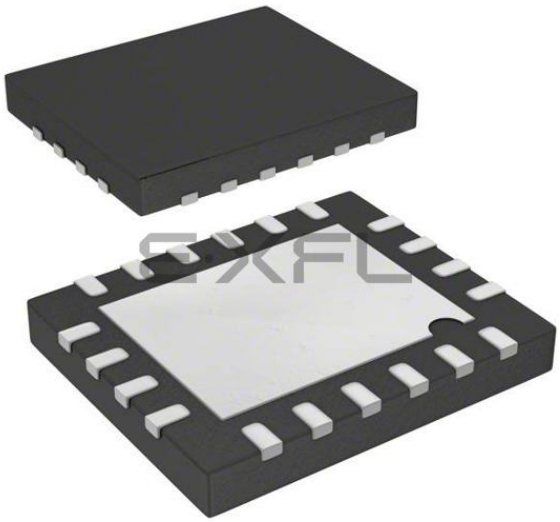


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	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	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-VQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fli39f2u6tr

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.

DATA EEPROM (Cont'd)

5.4 POWER SAVING MODES

Wait mode

The DATA EEPROM can enter WAIT mode on execution of the WFI instruction of the microcontroller or when the microcontroller enters Active-HALT mode.The DATA EEPROM will immediately enter this mode if there is no programming in progress, otherwise the DATA EEPROM will finish the cycle and then enter WAIT mode.

Active-Halt mode

Refer to Wait mode.

Halt mode

The DATA EEPROM immediately enters HALT mode if the microcontroller executes the HALT instruction. Therefore the EEPROM will stop the function in progress, and data may be corrupted.

5.5 ACCESS ERROR HANDLING

If a read access occurs while E2LAT=1, then the data bus will not be driven.

If a write access occurs while E2LAT=0, then the data on the bus will not be latched.

If a programming cycle is interrupted (by RESET action), the integrity of the data in memory is not guaranteed.

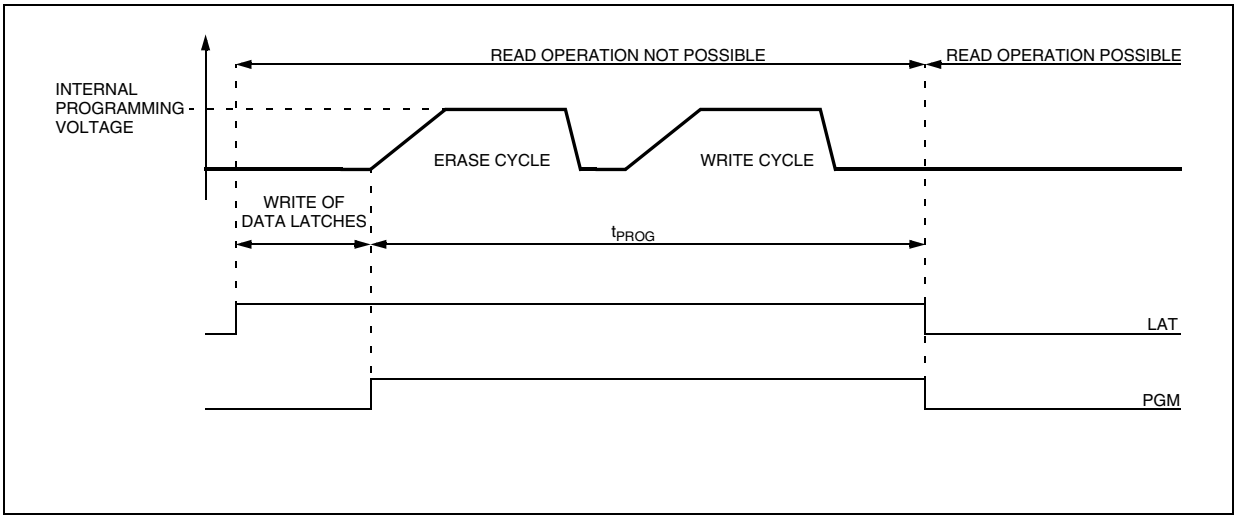
5.6 Data EEPROM Read-out Protection

The read-out protection is enabled through an option bit (see section 15.1 on page 161).

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

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

Figure 9. Data EEPROM Programming Cycle



DATA EEPROM (Cont'd)

5.7 REGISTER DESCRIPTION

EEPROM CONTROL/STATUS REGISTER (EECSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	E2LAT	E2PGM

Bits 7:2 = Reserved, forced by hardware to 0.

Bit 1 = **E2LAT** *Latch Access Transfer*
This bit is set by software. It is cleared by hardware at the end of the programming cycle. It can only be cleared by software if the E2PGM bit is cleared.
0: Read mode
1: Write mode

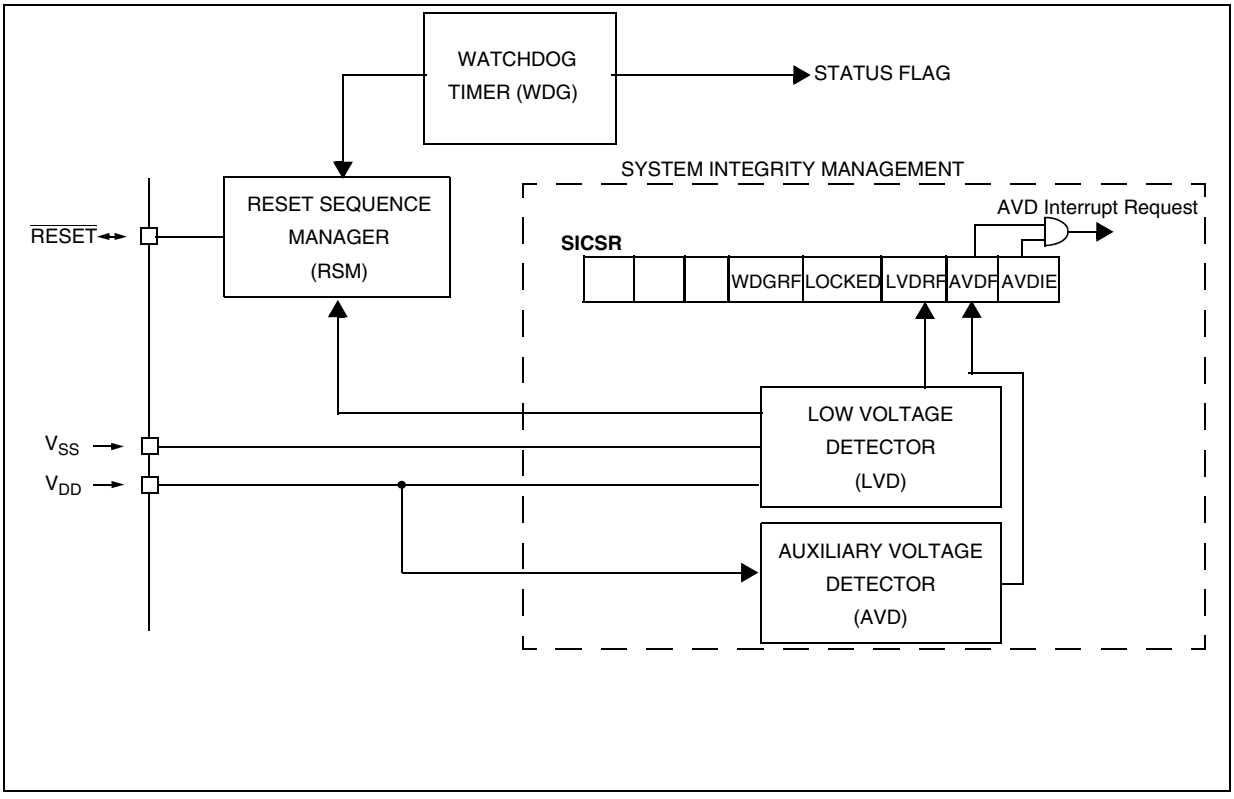
Bit 0 = **E2PGM** *Programming control and status*
This bit is set by software to begin the programming cycle. At the end of the programming cycle, this bit is cleared by hardware.
0: Programming finished or not yet started
1: Programming cycle is in progress

Note: if the E2PGM bit is cleared during the programming cycle, the memory data is not guaranteed

Table 4. DATA EEPROM Register Map and Reset Values

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

Figure 18. Reset and Supply Management Block Diagram



SYSTEM INTEGRITY MANAGEMENT (Cont'd)

7.6.2 Auxiliary Voltage Detector (AVD)

The Voltage Detector function (AVD) is based on an analog comparison between a $V_{IT-(AVD)}$ and $V_{IT+(AVD)}$ reference value and the V_{DD} main supply voltage (V_{AVD}). The $V_{IT-(AVD)}$ reference value for falling voltage is lower than the $V_{IT+(AVD)}$ reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real time status bit (AVDF) in the SICSR register. This bit is read only.

Caution: The AVD functions only if the LVD is enabled through the option byte.

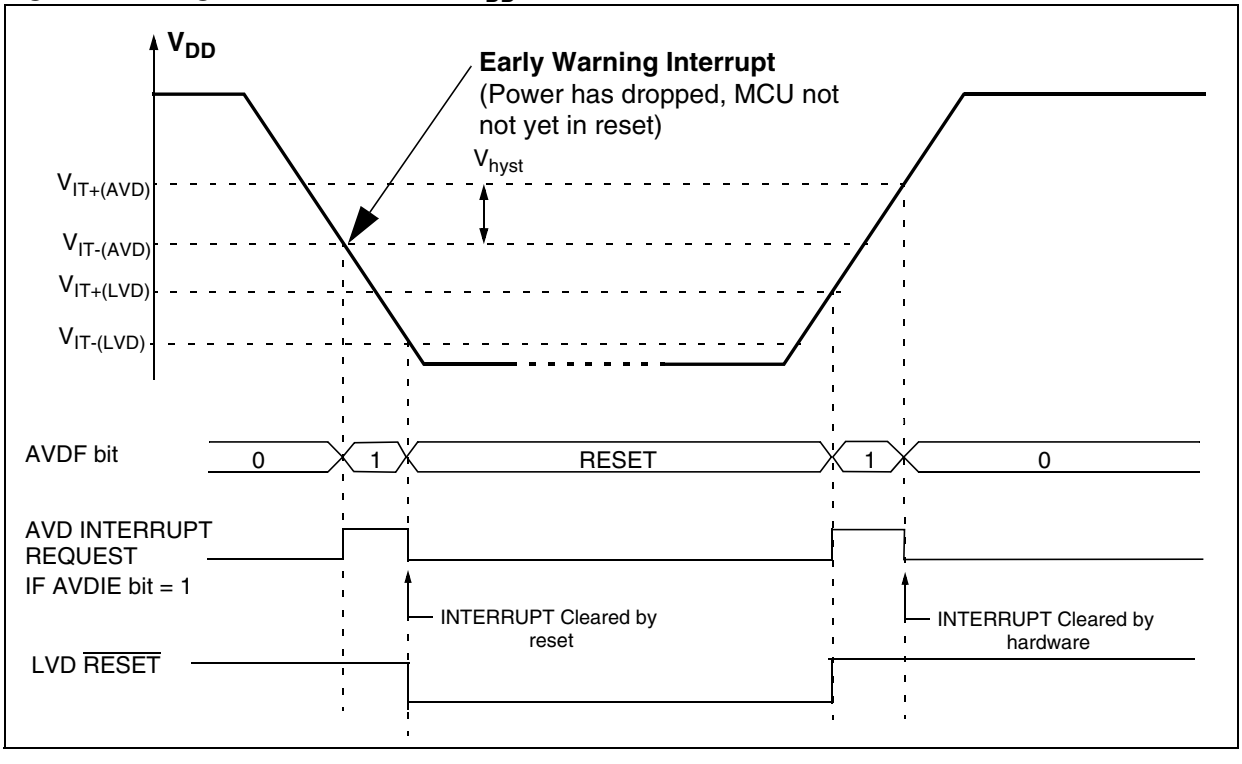
7.6.2.1 Monitoring the V_{DD} Main Supply

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see section 15.1 on page 161).

If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the $V_{IT+(LVD)}$ or $V_{IT-(AVD)}$ threshold (AVDF bit is set).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See Figure 19.

Figure 19. Using the AVD to Monitor V_{DD}



POWER SAVING MODES (Cont'd)

Similarities with Halt mode

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

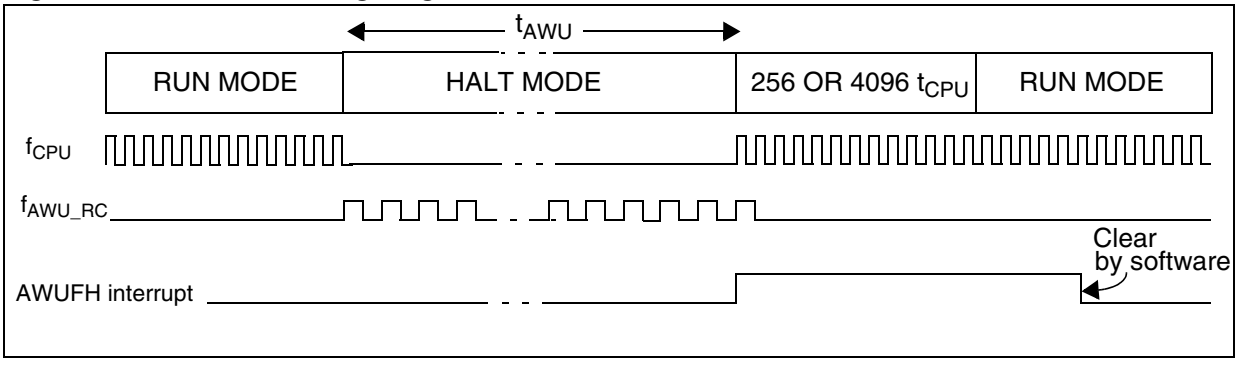
– The MCU can exit AWUFH mode by means of any interrupt with exit from Halt capability or a reset (see Section 9.4 HALT MODE).

– 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.

LITE TIMER (Cont'd)

11.3.4 Low Power Modes

Mode	Description
SLOW	No effect on Lite timer (this peripheral is driven directly by $f_{OSC}/32$)
WAIT	No effect on Lite timer
ACTIVE-HALT	No effect on Lite timer
HALT	Lite timer stops counting

11.3.5 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Active Halt	Exit from Halt
Timebase 1 Event	TB1F	TB1IE	Yes	Yes	No
Timebase 2 Event	TB2F	TB2IE	Yes	No	No
IC Event	ICF	ICIE	Yes	No	No

Note: The TBxF and ICF interrupt events are connected to separate interrupt vectors (see Interrupts chapter).

They generate an interrupt if the enable bit is set in the LTCSR1 or LTCSR2 register and the interrupt mask in the CC register is reset (RIM instruction).

11.3.6 Register Description

LITE TIMER CONTROL/STATUS REGISTER 2 (LTCSR2)
Read / Write
Reset Value: 0x00 0000 (x0h)

7							0
0	0	0	0	0	0	TB2IE	TB2F

Bits 7:2 = Reserved, must be kept cleared.

Bit 1 = **TB2IE** *Timebase 2 Interrupt enable*.
This bit is set and cleared by software.
0: Timebase (TB2) interrupt disabled
1: Timebase (TB2) interrupt enabled

Bit 0 = **TB2F** *Timebase 2 Interrupt Flag*.
This bit is set by hardware and cleared by software reading the LTCSR2 register. Writing to this bit has no effect.
0: No Counter 2 overflow
1: A Counter 2 overflow has occurred

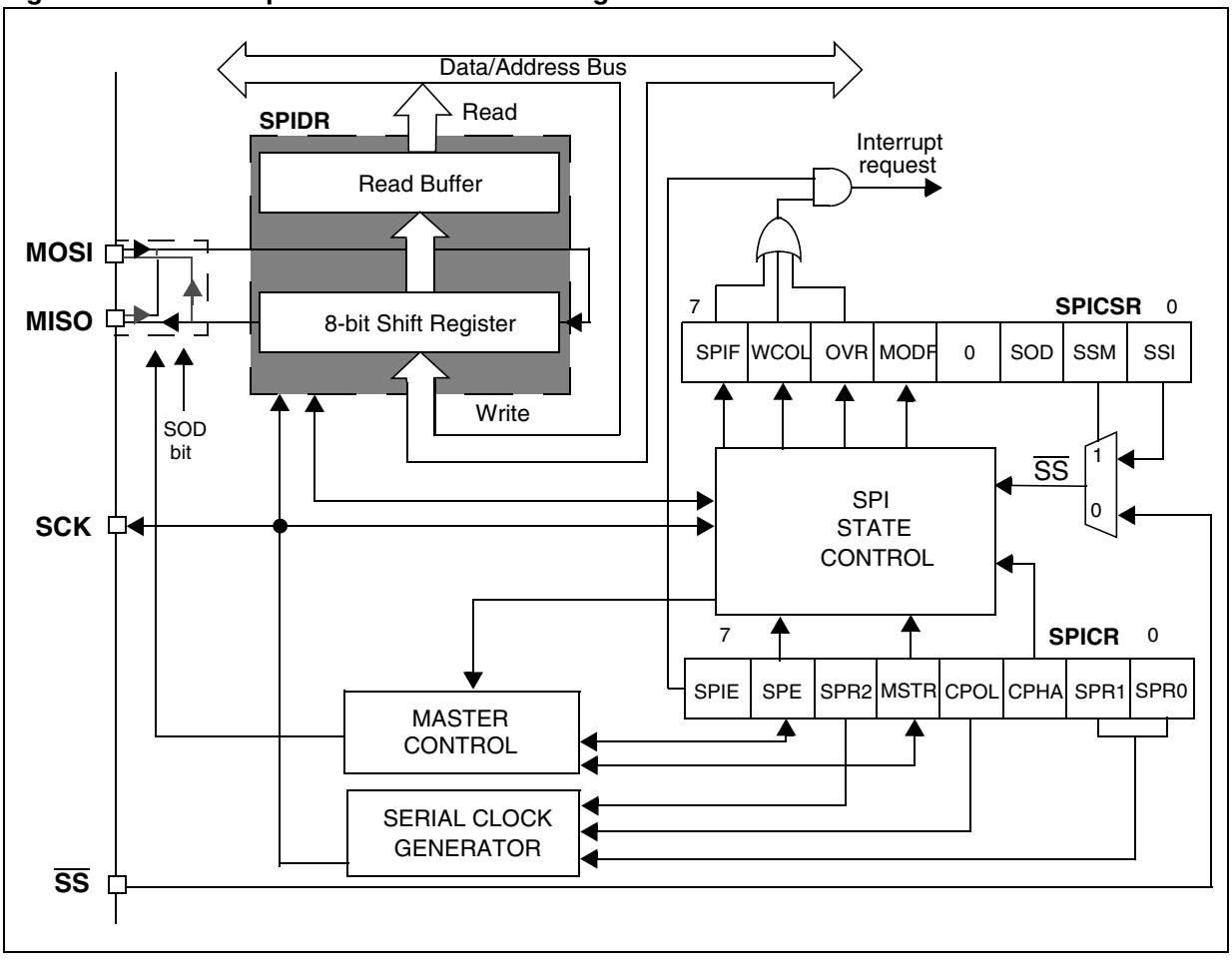
LITE TIMER AUTORELOAD REGISTER (LTARR)
Read / Write
Reset Value: 0000 0000 (00h)

7							0
AR7	AR7	AR7	AR7	AR3	AR2	AR1	AR0

Bits 7:0 = **AR[7:0]** *Counter 2 Reload Value.*
These bits register is read/write by software. The LTARR value is automatically loaded into Counter 2 (LTCNTR) when an overflow occurs.

SERIAL PERIPHERAL INTERFACE (SPI) (cont'd)

Figure 48. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (cont'd)

11.4.3.1 Functional Description

A basic example of interconnections between a single master and a single slave is illustrated in Figure 49.

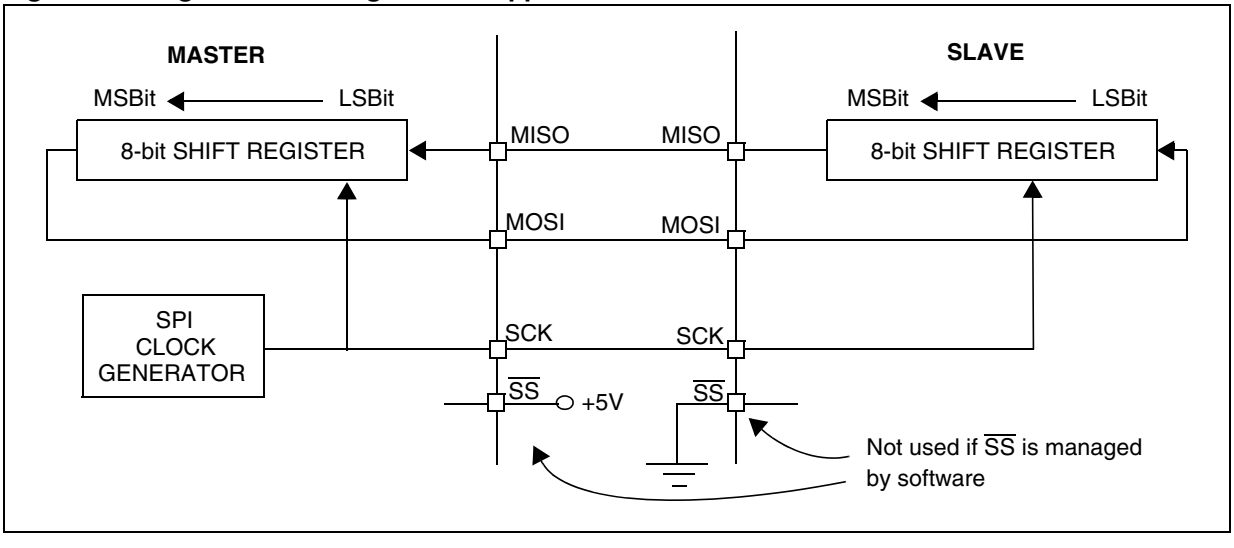
The MOSI pins are connected together and the MISO pins are connected together. In this way data is transferred serially between master and slave (most significant bit first).

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

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

Four possible data/clock timing relationships may be chosen (see Figure 52 on page 83) but master and slave must be programmed with the same timing mode.

Figure 49. Single Master/ Single Slave Application



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 52 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
     Note: 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 52).
     Note: The slave must have the same CPOL and CPHA settings as the master.
   - Manage the $\overline{SS}$ pin as described in Section 11.4.3.2 and Figure 50. 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 11.4.5.2).

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** *$\overline{SS}$ Management*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI $\overline{SS}$ pin and uses the SSI bit value instead. See Section 11.4.3.2 Slave Select Management.
0: Hardware management ($\overline{SS}$ managed by external pin)
1: Software management (internal $\overline{SS}$ signal controlled by SSI bit. External $\overline{SS}$ pin free for general-purpose I/O)

Bit 0 = **SSI** *$\overline{SS}$ Internal Mode*
This bit is set and cleared by software. It acts as a 'chip select' by controlling the level of the $\overline{SS}$ 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

7							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).

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

BAUD RATE REGISTER (SCIBRR)

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

7 0

SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

Note: When LIN slave mode is disabled, the SCIBRR register controls the conventional baud rate generator.

Bits 7:6 = **SCP[1:0]** *First SCI Prescaler*
These 2 prescaling bits allow several standard clock division ranges:

PR Prescaling factor	SCP1	SCP0
1	0	0
3		1
4	1	0
13		1

Bits 5:3 = **SCT[2:0]** *SCI Transmitter rate divisor*
These 3 bits, in conjunction with the SCP1 and SCP0 bits define the total division applied to the bus clock to yield the transmit rate clock in conventional Baud Rate Generator mode.

TR dividing factor	SCT2	SCT1	SCT0
1	0	0	0
2			1
4		1	0
8			1
16	1	0	0
32			1
64		1	0
128			1

Bits 2:0 = **SCR[2:0]** *SCI Receiver rate divider*
These 3 bits, in conjunction with the SCP[1:0] bits define the total division applied to the bus clock to yield the receive rate clock in conventional Baud Rate Generator mode.

RR dividing factor	SCR2	SCR1	SCR0
1	0	0	0
2			1
4		1	0
8			1
16	1	0	0
32			1
64		1	0
128			1

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	ERPR 6	ERPR 5	ERPR 4	ERPR 3	ERPR 2	ERPR 1	ERPR 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	ETPR 6	ETPR 5	ETPR 4	ETPR 3	ETPR 2	ETPR 1	ETPR 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.

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

If LHE bit is set due to this error during Fields other than LIN Synch Field or if LASE bit is reset then the current received Header is discarded and the SCI searches for a new Break Field.

Note on LIN Header Time-out Limit

According to the LIN specification, the maximum length of a LIN Header which does not cause a timeout is equal to 1.4 * (34 + 1) = 49 T_{BIT_MASTER}.

T_{BIT_MASTER} refers to the master baud rate.

When checking this timeout, the slave node is desynchronized for the reception of the LIN Break and Synch fields. Consequently, a margin must be allowed, taking into account the worst case: This occurs when the LIN identifier lasts exactly 10 T_{BIT_MASTER} periods. In this case, the LIN Break and Synch fields last 49 - 10 = 39T_{BIT_MASTER} periods.

Assuming the slave measures these first 39 bits with a desynchronized clock of 15.5%. This leads to a maximum allowed Header Length of:

39 x (1/0.845) T_{BIT_MASTER} + 10T_{BIT_MASTER}

= 56.15 T_{BIT_SLAVE}

A margin is provided so that the time-out occurs when the header length is greater than 57 T_{BIT_SLAVE} periods. If it is less than or equal to 57 T_{BIT_SLAVE} periods, then no timeout occurs.

LIN Header Length

Even if no timeout occurs on the LIN Header, it is possible to have access to the effective LIN header Length (T_{HEADER}) through the LHL register. This allows monitoring at software level the T_{FRAME_MAX} condition given by the LIN protocol.

This feature is only available when LHDM bit = 1 or when LASE bit = 1.

Mute Mode and Errors

In mute mode when LHDM bit = 1, if an LHE error occurs during the analysis of the LIN Synch Field or if a LIN Header Time-out occurs then the LHE bit is set but it does not wake up from mute mode. In this case, the current header analysis is discarded. If needed, the software has to reset LSF bit. Then the SCI searches for a new LIN header.

In mute mode, if a framing error occurs on a data (which is not a break), it is discarded and the FE bit is not set.

When LHDM bit = 1, any LIN header which respects the following conditions causes a wake-up from mute mode:

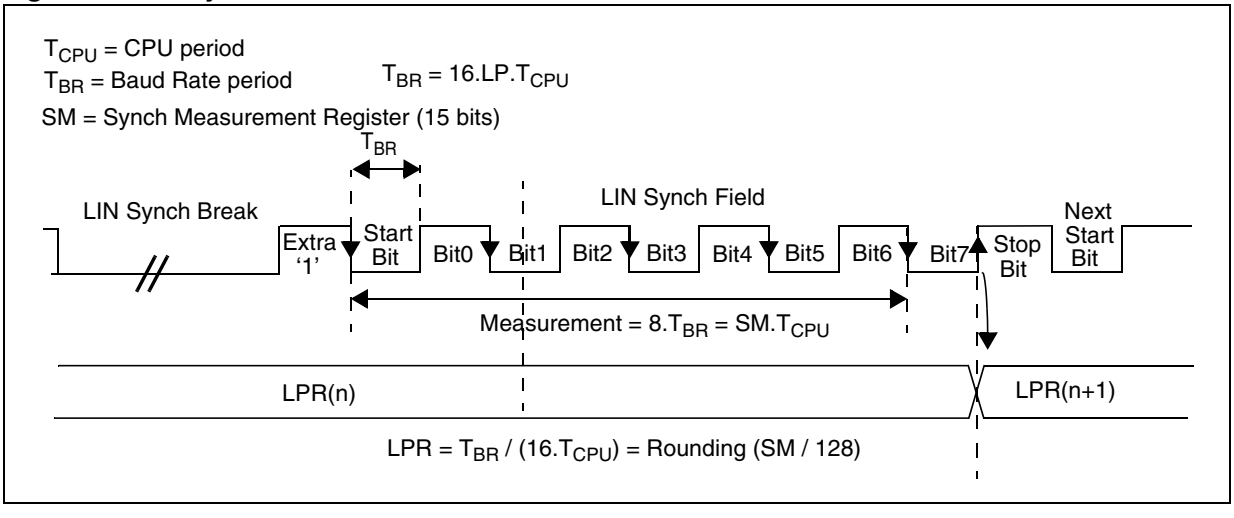
- A valid LIN Break Field (at least 11 dominant bits followed by a recessive bit)

- A valid LIN Synch Field (without deviation error)

- A LIN Identifier Field without framing error. Note that a LIN parity error on the LIN Identifier Field does not prevent wake-up from mute mode.

- No LIN Header Time-out should occur during Header reception.

Figure 61. LIN Synch Field Measurement



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)

SCICR2 register is set, the LHDM bit selects the Wake-Up method (replacing the WAKE bit).
0: LIN Synch Break Detection Method
1: LIN Identifier Field Detection Method

Bit 2 = **LHIE** *LIN Header Interrupt Enable*
This bit is set and cleared by software. It is only usable in LIN Slave mode.
0: LIN Header Interrupt is inhibited.
1: An SCI interrupt is generated whenever LHDF = 1.

Bit 1 = **LHDF** *LIN Header Detection Flag*
This bit is set by hardware when a LIN Header is detected and cleared by a software sequence (an access to the SCISR register followed by a read of the SCICR3 register). It is only usable in LIN Slave mode.
0: No LIN Header detected.
1: LIN Header detected.

Notes: The header detection method depends on the LHDM bit:

– If LHDM = 0, a header is detected as a LIN Synch Break.

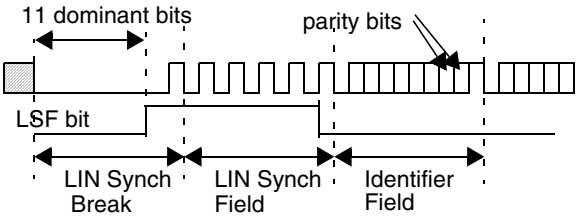
– If LHDM = 1, a header is detected as a LIN Identifier, meaning that a LIN Synch Break Field + a LIN Synch Field + a LIN Identifier Field have been consecutively received.

Bit 0 = **LSF** *LIN Synch Field State*

This bit indicates that the LIN Synch Field is being analyzed. It is only used in LIN Slave mode. In Auto Synchronization Mode (LASE bit = 1), when the SCI is in the LIN Synch Field State it waits or counts the falling edges on the RDI line.

It is set by hardware as soon as a LIN Synch Break is detected and cleared by hardware when the LIN Synch Field analysis is finished (see Figure 65). This bit can also be cleared by software to exit LIN Synch State and return to idle mode.
0: The current character is not the LIN Synch Field
1: LIN Synch Field State (LIN Synch Field undergoing analysis)

Figure 65. LSF Bit Set and Clear



LIN DIVIDER REGISTERS

LDIV is coded using the two registers LPR and LPFR. In LIN Slave mode, the LPR register is accessible at the address of the SCIBRR register and the LPFR register is accessible at the address of the SCIETPR register.

LIN PRESCALER REGISTER (LPR)

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

7							0
LPR7	LPR6	LPR5	LPR4	LPR3	LPR2	LPR1	LPR0

LPR[7:0] *LIN Prescaler (mantissa of LDIV)*

These 8 bits define the value of the mantissa of the LIN Divider (LDIV):

LPR[7:0]	Rounded Mantissa (LDIV)
00h	SCI clock disabled
01h	1
...	...
FEh	254
FFh	255

Caution: LPR and LPFR registers have different meanings when reading or writing to them. Consequently bit manipulation instructions (BRES or BSET) should never be used to modify the LPR[7:0] bits, or the LPFR[3:0] bits.

13.6 MEMORY CHARACTERISTICS

T_A = -40°C to 125°C, unless otherwise specified

13.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode [1]	HALT mode (or RESET)	1.6			V

13.6.2 FLASH Program Memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating voltage for Flash write/erase	Refer to operating range of V_{DD} with T_A, section 13.3.1 on page 133	2.7		5.5	V
t_{prog}	Programming time for 1~32 bytes [2]	T_A=–40 to +125°C		5	10	ms
	Programming time for 1.5 kBytes	T_A=+25°C		0.24	0.48	s
t_{RET}	Data retention [4]	T_A=+55°C[3]	20			years
N_{RW}	Write erase cycles	T_A=+25°C	10K			cycles
I_{DD}	Supply current	Read / Write / Erase modes f_{CPU} = 8MHz, V_{DD} = 5.5V			2.6 [6]	mA
		No Read/No Write Mode			100	µA
		Power down mode / HALT		0	0.1	µA

13.6.3 EEPROM Data Memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating voltage for EEPROM write/erase	Refer to operating range of V_{DD} with T_A, section 13.3.1 on page 133	2.7		5.5	V
t_{prog}	Programming time for 1~32 bytes	T_A=–40 to +125°C		5	10	ms
t_{ret}	Data retention [4]	T_A=+55°C [3]	20			years
N_{RW}	Write erase cycles	T_A=+25°C	300K			cycles

Notes:

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in HALT mode or under RESET) or in hardware registers (only in HALT mode). Guaranteed by construction, not tested in production.

2. Up to 32 bytes can be programmed at a time.

3. The data retention time increases when the T_A decreases.

4. Data based on reliability test results and monitored in production.

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

6. Guaranteed by Design. Not tested in production.

OPTION BYTES (Cont'd)

	OPTION BYTE 0								OPTION BYTE 1							
	7							0	7							0
	AWU CK	OSCRANGE 2:0			SEC1	SEC0	FMPR	FMPW	PLL x4x8	PLL OFF	Res.	OSC	LVD 1:0		WDG SW	WDG HALT
Default Value	1	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1

OPTION BYTE 1

OPT 7 = **PLLx4x8** *PLL Factor Selection*.
0: PLLx4
1: PLLx8

OPT 6 = **PLLOFF** *PLL Disable*
This option bit enables or disables the PLL.
0: PLL enabled
1: PLL disabled (bypassed)

OPT 5 = Reserved. Must always be set to 1.

OPT 4 = **OSC** *RC Oscillator Selection*
This option bit enables to select the internal RC Oscillator.
0: RC Oscillator on
1: RC Oscillator off

Notes:

– RC oscillator available on ST7LITE35 and ST7LITE39 devices only

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

OPT 3:2 = **LVD[1:0]** *Low Voltage Selection*
These option bits enable the voltage detection block (LVD and AVD) with a selected threshold to the LVD and AVD.

Configuration	VD1	VD0
LVD Off	1	1
Highest Voltage Threshold	1	0
Medium Voltage Threshold	0	1
Lowest Voltage Threshold	0	0

OPT 1 = **WDGSW** *Hardware or Software Watchdog*
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

OPT 0 = **WDG HALT** *Watchdog Reset on Halt*
0: No reset generation when entering HALT mode
1: Reset generation when entering HALT mode

07-Nov-06	7	Removed note “negative injection not allowed on PB0 and PB1 pins” (Table 2 on page 7 and section 13.2.2 on page 132) Added QFN20 package pinout (with new QFN20 mechanical data): Figure 2 on page 6 and Figure 105 on page 159 Modified section 15.3 on page 165 Modified option list on page 164
10-May-07	8	Added note 1 to Table 2 on page 7 Added caution “negative injection not allowed on PB0 and PB1 pins” (Table 2 on page 7 and section 13.2.2 on page 132) Modified section 11.6.3.3 on page 122 and added section 11.6.3.4 on page 122 Modified EOC bit description in section 11.6.6 on page 123 Added caution to section 7.5.1 on page 27 Modified note 1 in section 7.1 on page 23 Modified LTCSR1 reset value in section 11.3.6 on page 75 Modified part numbers for QFN20 package in Table 26 on page 163 and in option list on page 164 Modified section 15.1 on page 161 (added note to OSCRANGE option bits)
16-Nov-2007	9	Title of the document changed Modified section 7.6.4 on page 34 Soldering information section removed Modified “PACKAGE MECHANICAL DATA” on page 159 (values in inches are rounded to 4 decimal digits instead of 3 decimal digits) Modified section 13.7 on page 146 (removed references to DLU) Modified “DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE” on page 163 and option list on page 164