Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321bar9tae

Table 2. Device pin description (continued)

Pin No.		Pin name	Type	Level		Port						Main function (after reset)	Alternate function
						Input				Output			
LQFP64	LQFP44			Input	Output	float	wpu	int	ana	OD	PP		
62	1	PE1/RDI	I/O	C_T		**X**	X			X	X	Port E1	SCI Receive Data In
63	-(1)	PE2	I/O	C_T			**X**			X(6)	X(6)	Port E2	
64	-(2)(1)	PE3	I/O	C_T		**X**	X			X	X	Port E3	

1. On the chip, each I/O port may have up to 8 pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.
2. On the chip, each I/O port may have up to eight pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.
3. It is mandatory to connect all available V_{DD} and V_{AREF} pins to the supply voltage and all V_{SS} and V_{SSA} pins to ground.
4. Not connected in 48 Kbyte and 64 Kbyte ROM devices
5. OSC1 and OSC2 pins connect a crystal/ceramic resonator or an external source to the on-chip oscillator; see *Section 6.4: Multi-oscillator (MO)* and *Section 19.5: Clock and timing characteristics* for more details.
6. Pull-up always activated on PE2; see limitation *Section 22.1.8: Pull-up always active on PE2*.

Legend / Abbreviations for *Table 2*:

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

Input level: A = dedicated analog input

In/Output level: C = CMOS $0.3V_{DD}/0.7V_{DD}$, 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(a), ana = analog
- Output: OD = open-drain(b), PP = push-pull

a. In the interrupt input column, “eiX” defines the associated external interrupt vector. If the weak pull-up column (wpu) is merged with the interrupt column (int), then the I/O configuration is pull-up interrupt input, otherwise the configuration is floating interrupt input.

b. In the open-drain output column, “T” defines a true open-drain I/O (P-Buffer and protection diode to V_{DD} are not implemented). See *Section 9: I/O ports* and *Section 19.8: I/O port pin characteristics* for more details.

Table 19. Interrupt mapping

No.	Source block	Description	Register label	Priority order	Exit from Halt / Active Halt	Address vector
	RESET	Reset	N/A		yes	FFFEh-FFFFh
	TRAP	Software interrupt			no	FFFCh-FFFDh
0	TLI	External top level interrupt	EICR		yes	FFFAh-FFFBh
1	MCC/RTC	Main clock controller time base interrupt	MCCSR	Higher priority	yes	FFF8h-FFF9h
2	ei0	External interrupt port A3..0	N/A		yes	FFF6h-FFF7h
3	ei1	External interrupt port F2..0			yes	FFF4h-FFF5h
4	ei2	External interrupt port B3..0			yes	FFF2h-FFF3h
5	ei3	External interrupt port B7..4			yes	FFF0h-FFF1h
6	Not used					FFEEh-FFEFh
7	SPI	SPI peripheral interrupts	SPICSR		yes(1)	FFECh-FFEDh
8	TIMER A	TIMER A peripheral interrupts	TASR		no	FFEAh-FFEBh
9	TIMER B	TIMER B peripheral interrupts	TBSR		no	FFE8h-FFE9h
10	SCI	SCI peripheral interrupts	SCISR	Lower priority	no	FFE6h-FFE7h
11	AVD	Auxiliary voltage detector interrupt	SICSR		no	FFE4h-FFE5h
12	I2C	I2C peripheral interrupts	(see peripheral)		no	FFE2h-FFE3h
13	PWM ART	PWM ART interrupt	ARTCSR		yes(2)	FFE0h-FFE1h

1. Exit from HALT possible when SPI is in slave mode.
2. Exit from HALT possible when PWM ART is in external clock mode.

Table 21. Interrupt sensitivity - ei2 (port B3..0)

IS11	IS10	External interrupt sensitivity	
		IPB bit = 0	IPB bit = 1
0	0	Falling edge and low level	Rising edge and high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

Table 22. Interrupt sensitivity - ei3 (port B7..4)

IS11	IS10	External interrupt sensitivity
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Table 23. Interrupt sensitivity - ei0 (port A3..0)

IS21	IS20	External interrupt sensitivity	
		IPA bit = 0	IPA bit = 1
0	0	Falling edge and low level	Rising edge and high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

Table 24. Interrupt sensitivity - ei1 (port F2..0)

IS21	IS20	External interrupt sensitivity
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Table 33. I/O port register map and reset values (continued)

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Reset value of all I/O port registers		0	0	0	0	0	0	0	0
0003h	PBDR	MSB							LSB
0004h	PBDDR								
0005h	PBOR								
0006h	PCDR	MSB							LSB
0007h	PCDDR								
0008h	PCOR								
0009h	PDDR	MSB							LSB
000Ah	PDDDR								
000Bh	PDOR								
000Ch	PEDR	MSB							LSB
000Dh	PEDDR								
000Eh	PEOR								
000Fh	PFDR	MSB							LSB
0010h	PFDDR								
0011h	PFOR								

Related documentation

SPI Communication between ST7 and EEPROM (AN 970)

S/W implementation of I2C bus master (AN1045)

Software LCD driver (AN1048)

Table 43. Main clock controller register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
002Bh	SICSR Reset value	AVDS 0	AVDIE 0	AVDF 0	LVDRF x	 0	 0	 0	WDGRF x
002Ch	MCCSR Reset value	MCO 0	CP1 0	CP0 0	SMS 0	TB1 0	TB0 0	OIE 0	OIF 0
002Dh	MCCBCR Reset value	 0	 0	 0	 0	 0	 0	BC1 0	BC0 0

13.3.3 Input capture

In this section, the index, *i*, may be 1 or 2 because there are two input capture functions in the 16-bit timer.

The two 16-bit input capture registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition is detected on the ICAP*i* pin (see *Figure 46*).

	MS Byte	LS Byte
ICiR	IC*i*HR	IC*i*LR

IC*i*R register is a read-only register.

The active transition is software programmable through the IEDG*i* bit of Control Registers (CR*i*).

Timing resolution is one count of the free running counter: (f_{CPU}/CC[1:0]).

Procedure:

To use the input capture function select the following in the CR2 register:

- Select the timer clock (CC[1:0]) (see *Table 60: Timer clock selection*).
- Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

And select the following in the CR1 register:

- Set the ICIE bit to generate an interrupt after an input capture coming from either the ICAP1 pin or the ICAP2 pin
- Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

When an input capture occurs:

- ICF*i* bit is set.
- The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see *Figure 47*).
- A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the input capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set
2. An access (read or write) to the IC*i*LR register

Note: 1 After reading the ICiHR register, transfer of input capture data is inhibited and ICFi will never be set until the ICiLR register is also read.

2 The ICiR register contains the free running counter value which corresponds to the most recent input capture.

3 The two input capture functions can be used together even if the timer also uses the two output compare functions.

4 In One pulse Mode and PWM mode only Input Capture 2 can be used.

5 *The alternate inputs (ICAP1 and ICAP2) are always directly connected to the timer. So any transitions on these pins activates the input capture function.*

6 *Moreover if one of the ICAPi pins is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.*

7 *This can be avoided if the input capture function i is disabled by reading the ICiHR (see note 1).*

8 *The TOF bit can be used with interrupt generation in order to measure events that go beyond the timer range (FFFFh).*

Figure 46. Input capture block diagram

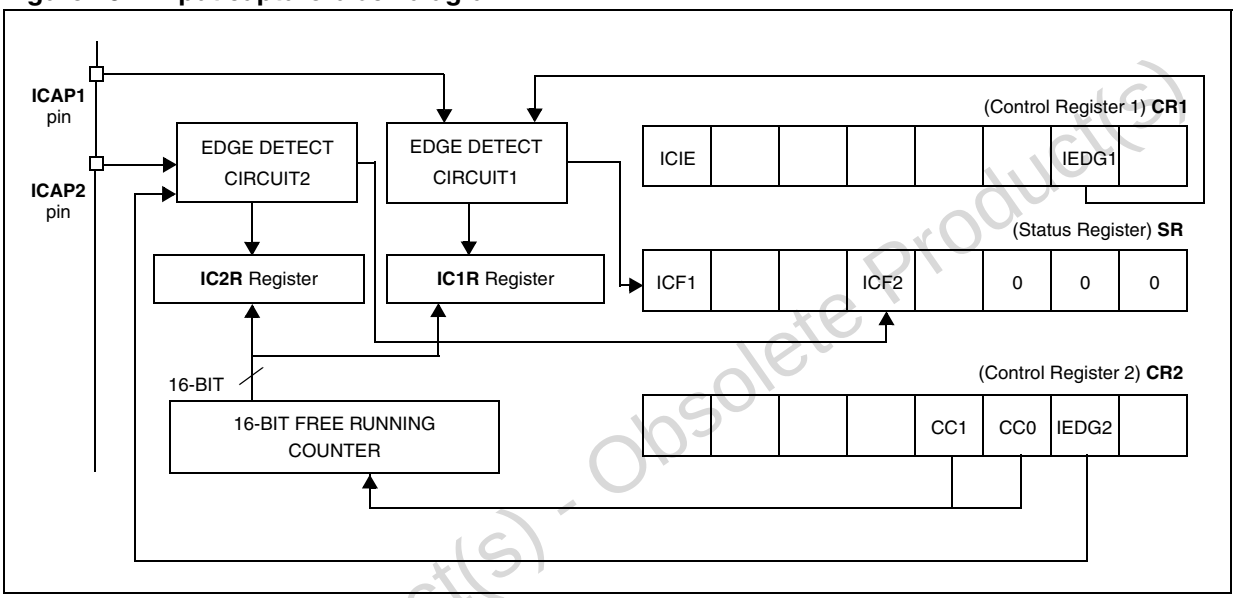
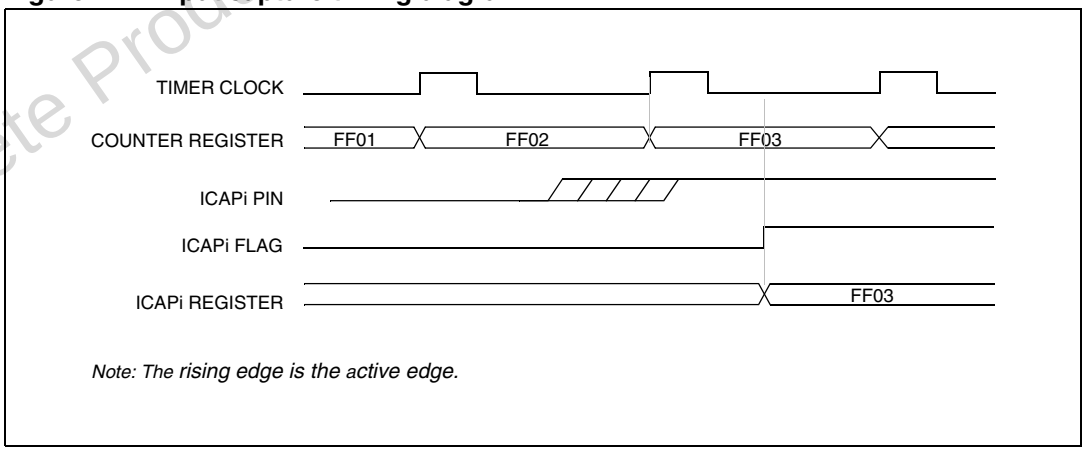


Figure 47. Input capture timing diagram



Note: The rising edge is the active edge.

13.7.9 Output compare 2 low register (OC2LR)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

OC2LR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MSB							LSB
RW	RW	RW	RW	RW	RW	RW	RW

13.7.10 Counter high register (CHR)

This is an 8-bit register that contains the high part of the counter value.

CHR Reset value: 1111 1111 (FFh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

13.7.11 Counter low register (CLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.

CLR Reset value: 1111 1100 (FCh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

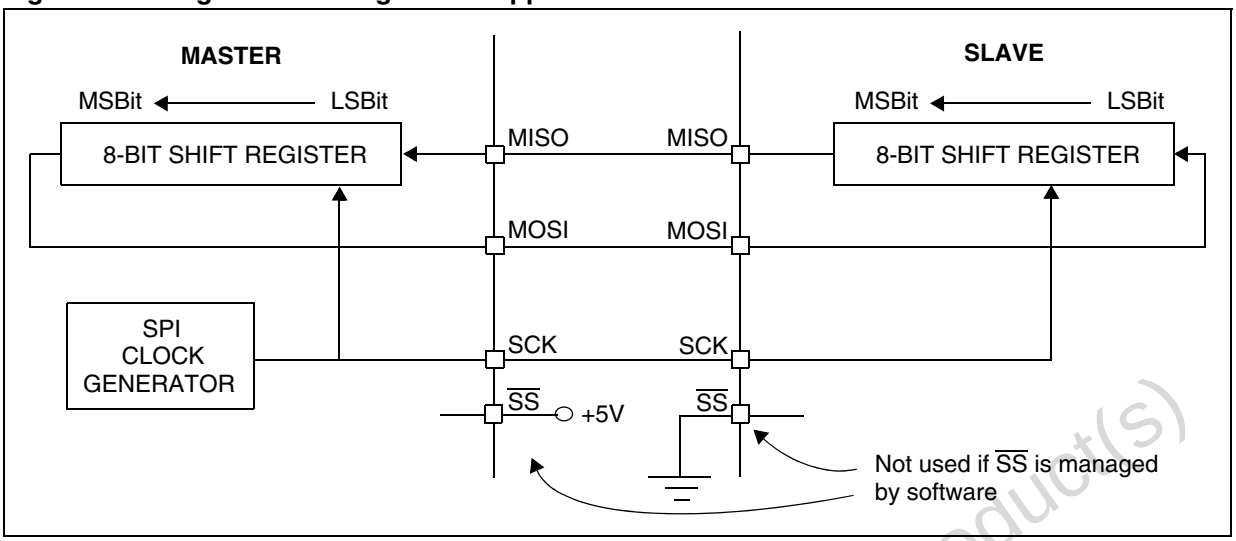
13.7.12 Alternate counter high register (ACHR)

This is an 8-bit register that contains the high part of the counter value.

ACHR Reset value: 1111 1111 (FFh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Figure 56. Single master/single slave application



14.3.2 Slave select management

As an alternative to using the $\overline{SS}$ pin to control the Slave Select signal, the application can choose to manage the Slave Select signal by software. This is configured by the SSM bit in the SPICSR register (see *Figure 58*)

In software management, the external $\overline{SS}$ pin is free for other application uses and the internal $\overline{SS}$ signal level is driven by writing to the SSI bit in the SPICSR register.

In Master mode

- $\overline{SS}$ internal must be held high continuously

In Slave mode

There are two cases depending on the data/clock timing relationship (see *Figure 57*):

If CPHA = 1 (data latched on 2nd clock edge):

- $\overline{SS}$ internal must be held low during the entire transmission. This implies that in single slave applications the $\overline{SS}$ pin either can be tied to V_{SS}, or made free for standard I/O by managing the $\overline{SS}$ function by software (SSM = 1 and SSI = 0 in the in the SPICSR register)

If CPHA = 0 (data latched on 1st clock edge):

- $\overline{SS}$ internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If $\overline{SS}$ is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see *Write collision error (WCOL) on page 129*).

Table 65. SPICR register description (continued)

Bit	Name	Function
1:0	SPR[1:0]	*Serial Clock Frequency* These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode. *Note: These 2 bits have no effect in slave mode.*

Table 66. SPI master mode SCK frequency

Serial clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

14.8.2 Control/status register (SPICSR)

SPICSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SPIF	WCOL	OVR	MODF	Reserved	SOD	SSM	SSI
RO	RO	RO	RO	-	RW	RW	RW

Table 67. SPICSR register description

Bit	Name	Function
7	SPIF	*Serial Peripheral Data Transfer Flag* 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. While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.
6	WCOL	*Write Collision status* 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 60*). 0: No write collision occurred. 1: A write collision has been detected.
5	OVR	*SPI Overrun error* 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 *Overrun condition (OVR) on page 129*). An interrupt is generated if SPIE = 1 in SPICR register. The OVR bit is cleared by software reading the SPICSR register. 0: No overrun error 1: Overrun error detected

15.4.2 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 62*).

Procedure

1. Select the M bit to define the word length.
2. Select the desired baud rate using the SCIBRR and the SCIETPR registers.
3. Set the TE bit to assign the TDO pin to the alternate function and to send an idle frame as first transmission.
4. 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 bit is 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 frame 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 bit is 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 frame length depends on the M bit (see *Figure 63*).

As long as the SBK bit is set, the SCI send break frames to the TDO pin. After clearing this

Conventional baud rate generation

The baud rate for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(16*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(16*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits)

TR = 1, 2, 4, 8, 16, 32, 64,128 (see SCT[2:0] bits)

RR = 1, 2, 4, 8, 16, 32, 64,128 (see SCR[2:0] bits)

All these bits are in the SCIBRR register.

Example: If f_{CPU} is 8 MHz (normal mode) and if PR = 13 and TR = RR = 1, the transmit and receive baud rates are 38400 baud.

Note: The baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.

Extended baud rate generation

The extended prescaler option provides a very fine tuning of the baud rate, using a 255 value prescaler, whereas the conventional baud rate generator retains industry standard software compatibility.

The extended baud rate generator block diagram is described in the *Figure 64*.

The output clock rate sent to the transmitter or to the receiver is the output from the 16 divider divided by a factor ranging from 1 to 255 set in the SCIERPR or the SCIETPR register.

Note: The extended prescaler is activated by setting the SCIETPR or SCIERPR register to a value other than zero. The baud rates are calculated as follows:

$$Tx = \frac{f_{CPU}}{16*ETPR*(PR*TR)} \qquad Rx = \frac{f_{CPU}}{16*ERPR*(PR*RR)}$$

with:

ETPR = 1,..,255 (see SCIETPR register)

ERPR = 1,..,255 (see SCIERPR register)

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 cannot be set.
- All the receive interrupts are inhibited.

A muted receiver may be awakened by one of the following two ways:

- by Idle Line detection if the WAKE bit is reset
- by Address Mark detection if the WAKE bit is set

Table 71. SCI interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Idle Line Detected	IDLE	ILIE	Yes	No
Parity Error	PE	PIE	Yes	No

15.7 SCI registers

15.7.1 Status register (SCISR)

SCISR Reset value: 1100 0000 (C0h)

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE
RO	RO	RO	RO	RO	RO	RO	RO

Table 72. SCISR register description

Bit	Name	Function
7	TDRE	*Transmit data register empty* This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE bit = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Data is not transferred to the shift register 1: Data is transferred to the shift register *Note: Data is not transferred to the shift register unless the TDRE bit is cleared.*
6	TC	*Transmission complete* This bit is set by hardware when transmission of a frame containing Data is complete. An interrupt is generated if TCIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Transmission is not complete 1: Transmission is complete *Note: TC is not set after the transmission of a Preamble or a Break.*
5	RDRF	*Received data ready flag* This bit is set by hardware when the content of the RDR register has been transferred to the SCIDR register. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: Data is not received 1: Received data is ready to be read

Table 76. SCIERPR register description

Bit	Name	Function
7:0	ERPR[7:0]	*8-bit Extended Receive Prescaler Register* The extended baud rate generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see *Figure 64*) is divided by the binary factor set in the SCIERPR register (in the range 1 to 255). The extended baud rate generator is not used after a reset.

15.7.7 Extended transmit prescaler division register (SCIETPR)

This register allows setting of the external prescaler rate division factor for the transmit circuit.

SCIETPR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ETPR[7:0]							
RW							

Table 77. SCIETPR register description

Bit	Name	Function
7:0	ETPR[7:0]	*8-bit Extended Transmit Prescaler Register* The extended baud rate generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see *Figure 64*) is divided by the binary factor set in the SCIETPR register (in the range 1 to 255). The extended baud rate generator is not used after a reset.

Table 78. Baud rate selection

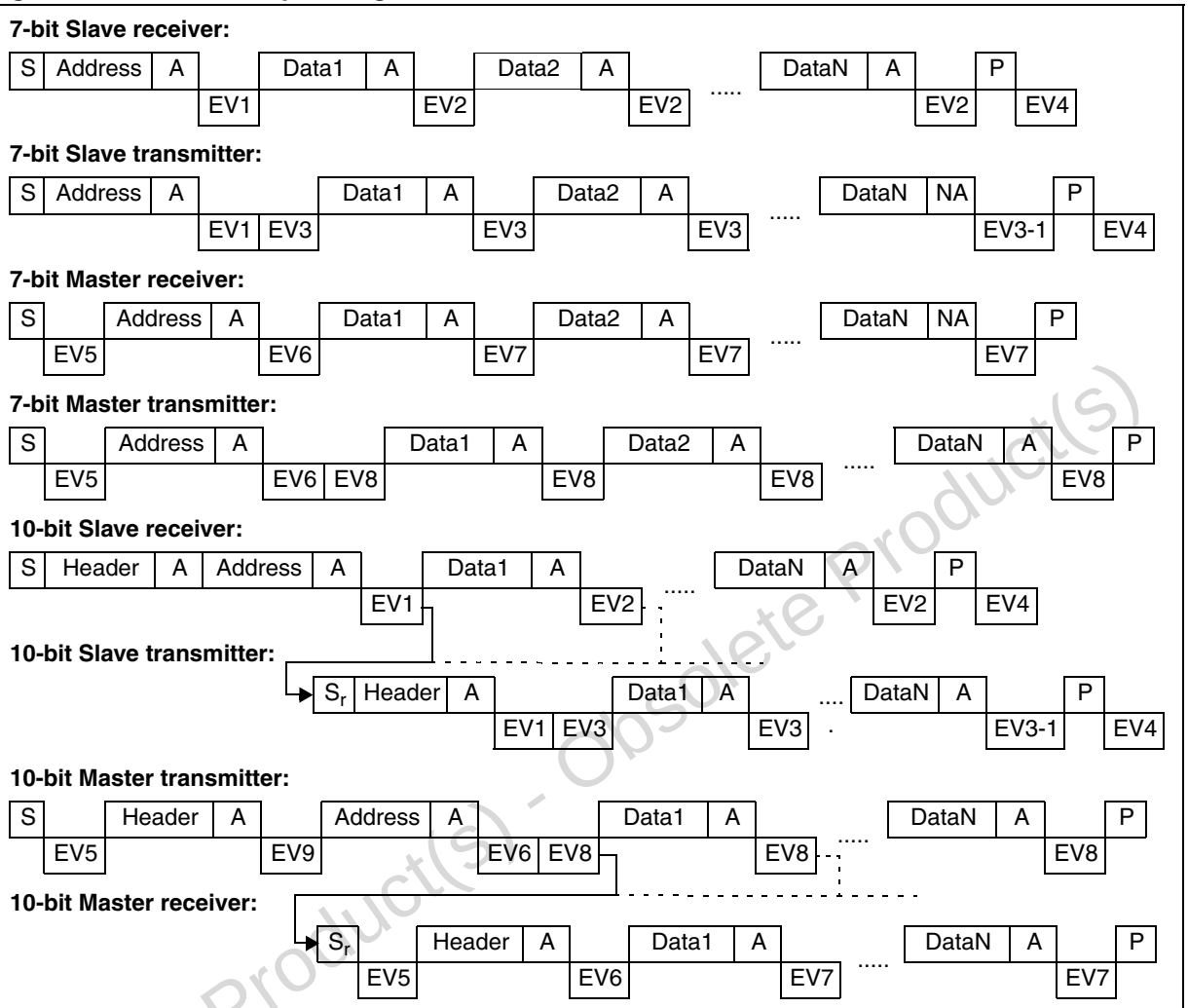
Symbol	Parameter	Conditions			Standard	Baud rate	Unit
		f_{CPU}	Accuracy versus standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8 MHz	~0.16%	Conventional mode TR (or RR) = 128, PR = 13 TR (or RR) = 32, PR = 13 TR (or RR) = 16, PR = 13 TR (or RR) = 8, PR = 13 TR (or RR) = 4, PR = 13 TR (or RR) = 16, PR = 3 TR (or RR) = 2, PR = 13 TR (or RR) = 1, PR = 13	300 1200 2400 4800 9600 10400 19200 38400	~300.48 ~1201.92 ~2403.84 ~4807.69 ~9615.38 ~10416.67 ~19230.77 ~38461.54	Hz
			~0.79%	Extended mode ETPR (or ERPR) = 35, TR (or RR) = 1, PR = 1	14400	~14285.71	

Error cases

- **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and BERR bits are set by hardware with an interrupt if ITE is set.
  Note that BERR will not be set if an error is detected during the first or second pulse of each 9-bit transaction:
  - ***Single Master Mode***
    If a Start or Stop is issued during the first or second pulse of a 9-bit transaction, the BERR flag will not be set and transfer will continue however the BUSY flag will be reset. To work around this, slave devices should issue a NACK when they receive a misplaced Start or Stop. The reception of a NACK or BUSY by the master in the middle of communication makes it possible to re-initiate transmission.
  - ***Multimaster Mode***
    Normally the BERR bit would be set whenever unauthorized transmission takes place while transfer is already in progress. However, an issue will arise if an external master generates an unauthorized Start or Stop while the I^2C master is on the first or second pulse of a 9-bit transaction. It is possible to work around this by polling the BUSY bit during I^2C master mode transmission. The resetting of the BUSY bit can then be handled in a similar manner as the BERR flag being set.
- **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the Start or Stop bit. The AF bit is cleared by reading the I2CSR2 register. However, if read before the completion of the transmission, the AF flag will be set again, thus possibly generating a new interrupt. Software must ensure either that the SCL line is back at 0 before reading the SR2 register, or be able to correctly handle a second interrupt during the 9th pulse of a transmitted byte.
- **ARLO:** Detection of an arbitration lost condition.
  In this case the ARLO bit is set by hardware (with an interrupt if the ITE bit is set and the interface goes automatically back to slave mode (the M/SL bit is cleared).

Note: In all these cases, the SCL line is not held low; however, the SDA line can remain low due to possible '0' bits transmitted last. It is then necessary to release both lines by software.

Figure 68. Transfer sequencing



Legend:

S = Start, Sr = Repeated Start, P = Stop, A = Acknowledge, NA = Non-acknowledge, EVx = Event (with interrupt if ITE = 1)

EV1: EVF = 1, ADSL = 1, cleared by reading SR1 register.

EV2: EVF = 1, BTF = 1, cleared by reading SR1 register followed by reading DR register.

EV3: EVF = 1, BTF = 1, cleared by reading SR1 register followed by writing DR register.

EV3-1: EVF = 1, AF = 1, BTF = 1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP = 1, STOP = 0) or by writing DR register (DR = FFh). ***Note:*** *If lines are released by STOP = 1, STOP = 0, the subsequent EV4 is not seen.*

EV4: EVF = 1, STOPF = 1, cleared by reading SR2 register.

EV5: EVF = 1, SB = 1, cleared by reading SR1 register followed by writing DR register.

EV6: EVF = 1, cleared by reading SR1 register followed by writing CR register (for example PE = 1).

EV7: EVF = 1, BTF = 1, cleared by reading SR1 register followed by reading DR register.

EV8: EVF = 1, BTF = 1, cleared by reading SR1 register followed by writing DR register.

EV9: EVF = 1, ADD10 = 1, cleared by reading SR1 register followed by writing DR register.

16.7 Register description

16.7.1 I²C control register (CR)

CR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved		PE	ENGC	START	ACK	STOP	ITE
-		RW	RW	RW	RW	RW	RW

Table 82. CR register description

Bit	Name	Function
7:6	-	Reserved. Forced to 0 by hardware.
5	PE	*Peripheral enable* This bit is set and cleared by software. 0: Peripheral disabled 1: Master/Slave capability *Notes:* *- When PE = 0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released while PE = 0* *- When PE = 1, the corresponding I/O pins are selected by hardware as alternate functions.* *To enable the I²C interface, write the CR register **TWICE** with PE = 1 as the first write only activates the interface (only PE is set).*
4	ENGC	*Enable General Call* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0). The 00h General Call address is acknowledged (01h ignored). 0: General Call disabled 1: General Call enabled *Note: In accordance with the I2C standard, when GCAL addressing is enabled, an I2C slave can only receive data. It will not transmit data to the master.*
3	START	*Generation of a Start condition* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0) or when the Start condition is sent (with interrupt generation if ITE = 1). **In Master mode** 0: No start generation 1: Repeated start generation **In Slave mode** 0: No start generation 1: Start generation when the bus is free
2	ACK	*Acknowledge enable* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0). 0: No acknowledge returned 1: Acknowledge returned after an address byte or a data byte is received

Table 99. Instructions supporting direct, indexed, indirect, and indirect indexed addressing modes (continued)

Type	Instruction	Function
Short instructions only	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 Operations
	SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
	SWAP	Swap Nibbles
	CALL, JP	Call or Jump subroutine

18.1.7 Relative (direct, indirect)

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

Table 100. Available relative direct/indirect instructions

Instruction	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

Relative (direct)

The offset is following the opcode.

Relative (indirect)

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

18.2 Instruction groups

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Table 101. Instruction groups

Group	Instructions							
Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					

19.3.4 External voltage detector (EVD) thresholds

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A.

Table 109. External voltage detector (EVD) thresholds

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(EVD)}$	1⇒0 AVDF flag toggle threshold (V_{DD} rise)(1)		1.15	1.26	1.35	V
$V_{IT-(EVD)}$	0⇒1 AVDF flag toggle threshold (V_{DD} fall)(1)		1.1	1.2	1.3	
$V_{hys(EVD)}$	EVD voltage threshold hysteresis	$V_{IT+(EVD)}$-$V_{IT-(EVD)}$		200		mV

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

19.9 Control pin characteristics

19.9.1 Asynchronous $\overline{\text{RESET}}$ pin

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A unless otherwise specified.

Table 127. Asynchronous $\overline{\text{RESET}}$ pin characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage(1)				$0.3xV_{DD}$	V
V_{IH}	Input high level voltage(1)		$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis(2)			2.5		
V_{OL}	Output low level voltage(3)	V_{DD} = 5V, I_{IO} = +2mA		0.2	0.5	
I_{IO}	Input current on $\overline{\text{RESET}}$ pin			2		mA
R_{ON}	Weak pull-up equivalent resistor		20	30	120	kΩ
$t_{w(RSTL)out}$	Generated reset pulse duration	Stretch applied on external pulse	0		42(4)	µs
		Internal reset sources	20	30	42(4)	
$t_{h(RSTL)in}$	External reset pulse hold time(5)		2.5			
$t_{g(RSTL)in}$	Filtered glitch duration(6)			200		ns

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels.
3. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 19.2.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
4. Data guaranteed by design, not tested in production.
5. To guarantee the reset of the device, a minimum pulse has to be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on the $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.
6. The reset network (the resistor and two capacitors) protects the device against parasitic resets, especially in noisy environments.