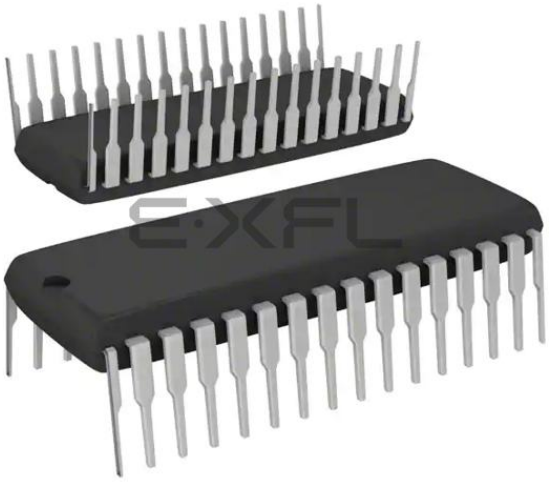


Welcome to E-XFL.COM

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, USB
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	32-SDIP (0.400", 10.16mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f63bk6b1

List of tables

Figure 2. 48-pin LQFP pinout

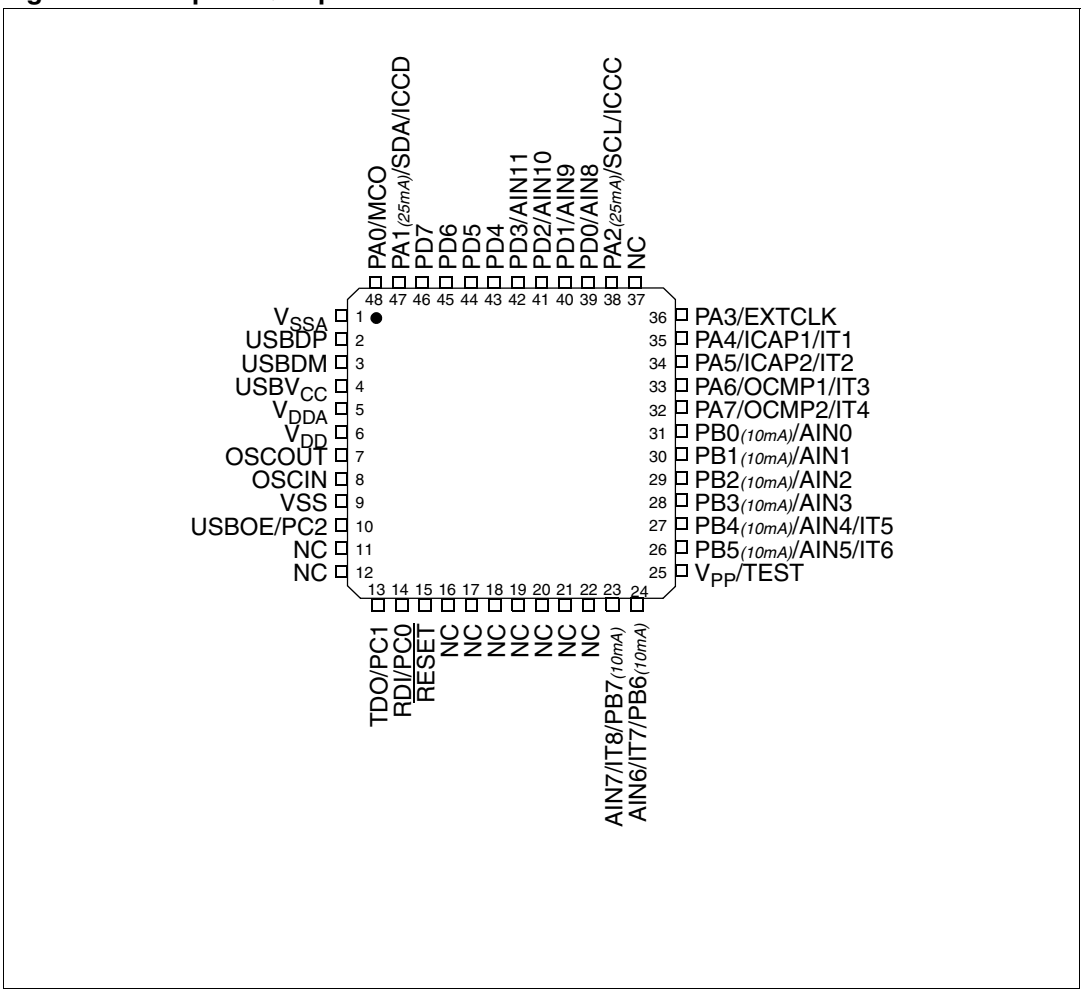


Table 3. Device pin description (QFN40, LQFP48, SO34 and SDIP32) (continued)

Pin n°				Pin name	Type	Level		Port /control						Main function (after reset)	Alternate function
								Input				Output			
SDIP32	SO34	QFN40	LQFP48			Input	Output	float	wpu	int	ana	OD	PP		
15	16	23	29	PB2/AIN2	I/O	CT	10mA	X			X		X	Port B2	ADC analog input 2
16	17	24	30	PB1/AIN1	I/O	CT	10mA	X			X		X	Port B1	ADC analog input 1
17	18	25	31	PB0/AIN0	I/O	CT	10mA	X			X		X	Port B0	ADC analog input 0
18	19	26	32	PA7/OCMP2/IT4	I/O	CT			X	X			X	Port A7	Timer output Compare 2
19	20	27	33	PA6/OCMP1/IT3	I/O	CT			X	X			X	Port A6	Timer output Compare 1
20	21	28	34	PA5/ICAP2/IT2	I/O	CT			X	X			X	Port A5	Timer input Capture 2
21	22	29	35	PA4/ICAP1/IT1	I/O	CT			X	X			X	Port A4	Timer input Capture 1
22	23	30	36	PA3/EXTCLK	I/O	CT			X				X	Port A3	Timer External clock
23	24	31	38	PA2/SCL/ICCCLK	I/O	C_T	25mA	X				T		Port A2	I²C serial clock, ICC clock
-	-	32	39	PD0(1)/AIN8	I/O	C_T		X			X		X	Port D0	ADC analog input 8
-	-	33	40	PD1(1)/AIN9	I/O	C_T		X			X		X	Port D1	ADC analog input 9
-	-	34	41	PD2(1)/AIN10	I/O	C_T		X			X		X	Port D2	ADC analog input 10
-	-	35	42	PD3(1)/AIN11	I/O	C_T		X			X		X	Port D3	ADC analog input 11
-	-	36	43	PD4(1)	I/O	C_T			X				X	Port D4	
-	-	37	44	PD5(1)	I/O	C_T			X				X	Port D5	
-	-	38	45	PD6(1)	I/O	C_T			X				X	Port D6	
-	-	39	46	PD7(1)	I/O	C_T			X				X	Port D7	
-	25	-	-	NC	--									Not connected	
24	26	-	-	NC	--									Not connected	
25	27	-	-	NC	--									Not connected	
26	28	40	47	PA1/SDA/ICCDATA	I/O	CT	25mA	X				T		Port A1	I²C serial data, ICC data
27	29	1	48	PA0/MCO	I/O	CT				X			X	Port A0	Main clock output
28	30	2	1	V_{SSA}	S									Analog ground	
29	31	3	2	USBDP	I/O									USB bidirectional data (data +)	
30	32	4	3	USBDM	I/O									USB bidirectional data (data -)	

Table 6. Hardware register memory map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0025h	USB	USBPIDR	USB PID register	x0h	Read only
0026h		USBDMAR	USB DMA address register	xxh	R/W
0027h		USBIDR	USB Interrupt/DMA register	x0h	R/W
0028h		USBISTR	USB Interrupt Status register	00h	R/W
0029h		USBIMR	USB Interrupt Mask register	00h	R/W
002Ah		USBCTLR	USB Control register	06h	R/W
002Bh		USBDADDR	USB Device Address register	00h	R/W
002Ch		USBEP0RA	USB Endpoint 0 register A	0000 xxxxb	R/W
002Dh		USBEP0RB	USB Endpoint 0 register B	80h	R/W
002Eh		USBEP1RA	USB Endpoint 1 register A	0000 xxxxb	R/W
002Fh		USBEP1RB	USB Endpoint 1 register B	0000 xxxxb	R/W
0030h		USBEP2RA	USB Endpoint 2 register A	0000 xxxxb	R/W
0031h		USBEP2RB	USB Endpoint 2 register B	0000 xxxxb	R/W
0032h to 0036h	Reserved (5 bytes)				
0032h 0036h	Reserved (5 Bytes)				
0037h	Flash	FCSR	Flash Control /Status register	00h	R/W
0038h	Reserved (1 byte)				
0039h	I²C	I2CDR	I²C Data register	00h	R/W
003Ah			Reserved	-	
003Bh		I2COAR	I²C (7 Bits) Slave Address register	00h	R/W
003Ch		I2CCCR	I²C Clock Control register	00h	R/W
003Dh		I2CSR2	I²C 2nd Status register	00h	Read only
003Eh		I2CSR1	I²C 1st Status register	00h	Read only
003Fh		I2CCR	I²C Control register	00h	R/W

Figure 12. Low voltage detector functional diagram

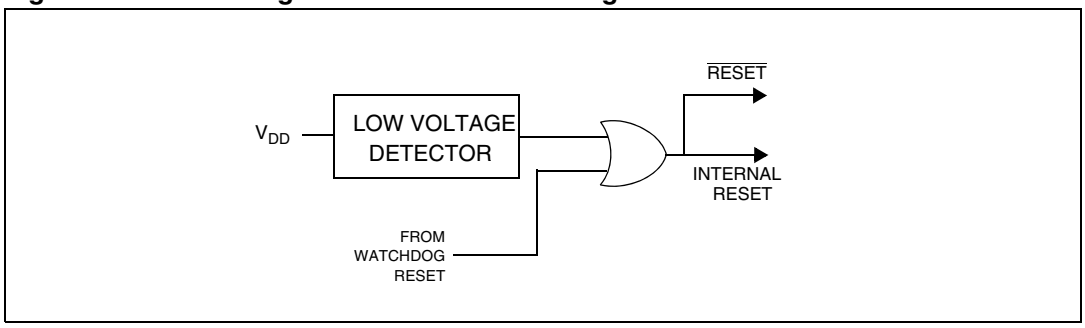
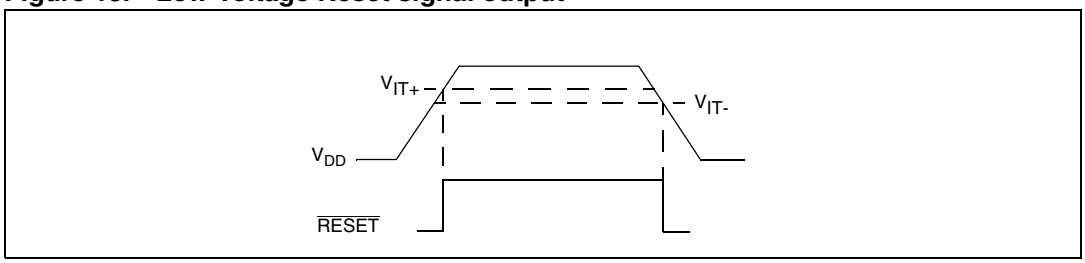
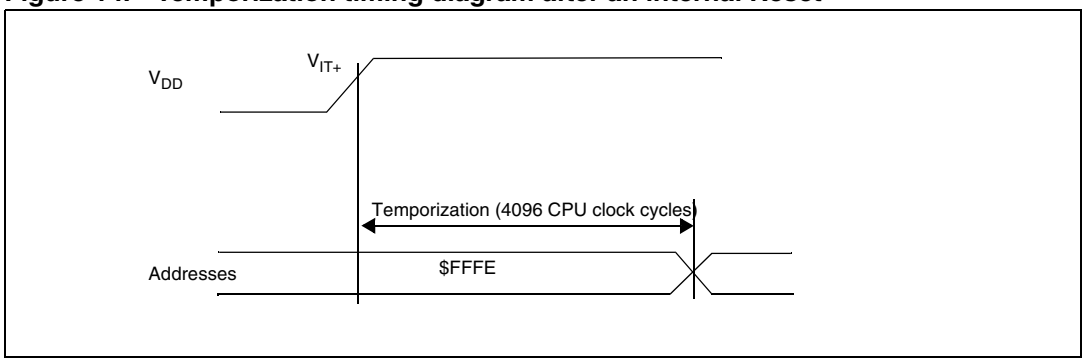


Figure 13. Low Voltage Reset signal output



1. Hysteresis (V_{IT+}-V_{IT-}) = V_{hys}

Figure 14. Temporization timing diagram after an internal Reset



7 Interrupts

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in *Table 9* and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in *Figure 19*.

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

When an interrupt has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- The I bit of the CC register is set to prevent additional interrupts.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to *Table 9* for vector addresses).

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

Note: As a consequence of the IRET instruction, the I bit will be cleared and the main program will resume.

Priority management

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

In the case several interrupts are simultaneously pending, a hardware priority defines which one will be serviced first (see *Table 9*).

Non-maskable software interrupts

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on *Figure 19*.

Interrupts and low power mode

All interrupts allow the processor to leave the Wait low power mode. Only external and specific mentioned interrupts allow the processor to leave the Halt low power mode (refer to the “Exit from HALT“ column in *Table 9*).

External interrupts

The pins ITi/PAk and ITj/PBk (i=1,2; j= 5,6; k=4,5) can generate an interrupt when a rising edge occurs on this pin. Conversely, the ITl/PAn and ITm/PBn pins (l=3,4; m= 7,8; n=6,7) can generate an interrupt when a falling edge occurs on this pin.

Interrupt generation will occur if it is enabled with the ITiE bit (i=1 to 8) in the ITRFRE register and if the I bit of the CC is reset.

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

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

Figure 32. Input Capture block diagram

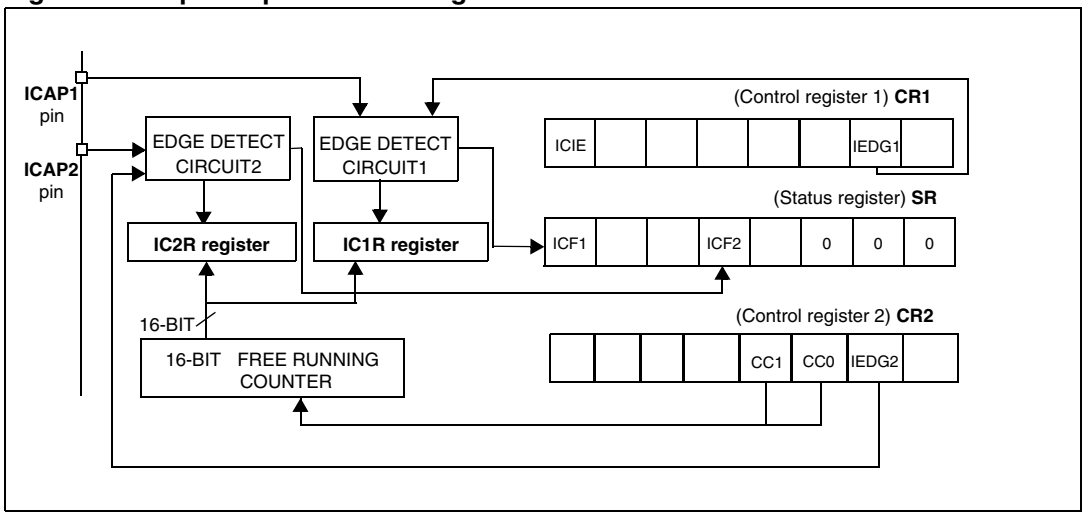
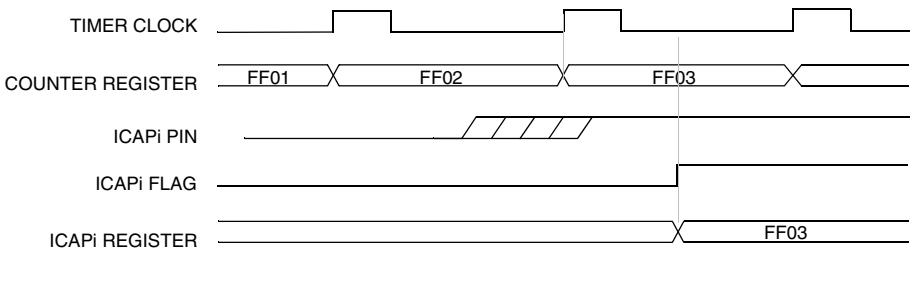


Figure 33. Input Capture timing diagram



1. The rising edge is the active edge.

One Pulse mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

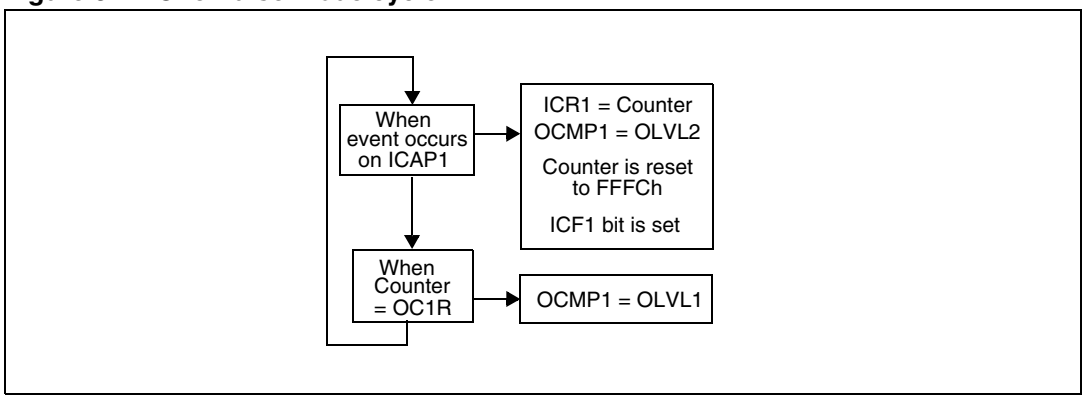
The One Pulse mode uses the input Capture1 function and the output Compare1 function.

Procedure

To use One Pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in the opposite column).
2. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
   - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
3. Select the following in the CR2 register:
   - Set the OC1E bit, the OCMP1 pin is then dedicated to the output Compare 1 function.
   - Set the OPM bit.
   - Select the timer clock CC[1:0] (see *Table 24*).

Figure 37. One Pulse mode cycle

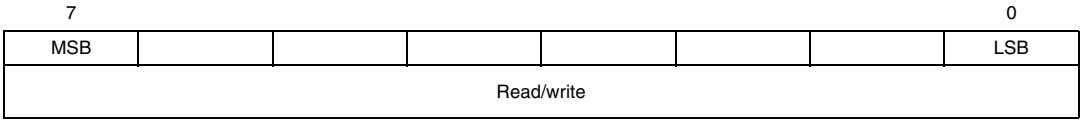


Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

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

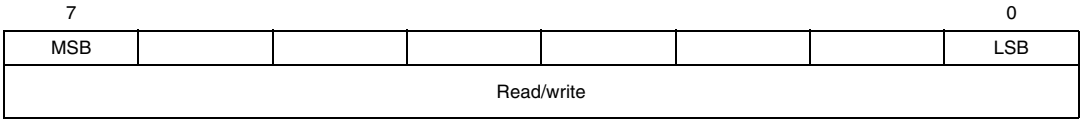
1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.



Output Compare 2 High register (OC2HR)

Reset value: 1000 0000 (80h)

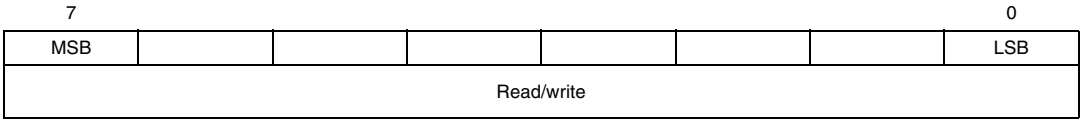
This is an 8-bit register that contains the high part of the value to be compared to the CHR register.



Output Compare 2 Low register (OC2LR)

Reset value: 0000 0000 (00h)

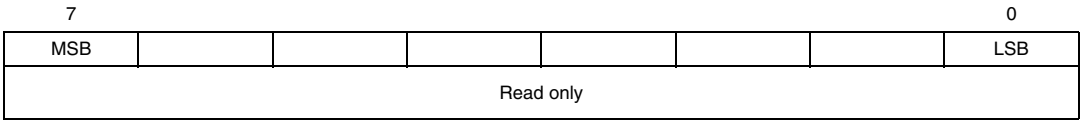
This is an 8-bit register that contains the low part of the value to be compared to the CLR register.



Counter High register (CHR)

Reset value: 1111 1111 (FFh)

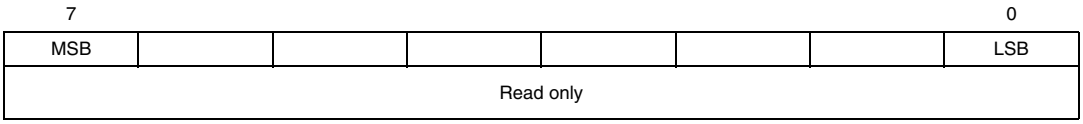
This is an 8-bit register that contains the high part of the counter value.



Counter Low register (CLR)

Reset value: 1111 1100 (FCh)

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.



Alternate Counter High register (ACHR)

Reset value: 1111 1111 (FFh)

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

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 41*).

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 a 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 CC 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 or after the break frame) the TC bit is set and an interrupt is generated if the TCIE is set and the I bit is cleared in the CC 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 42*).

clock occurs just before the pin value changes, the samples would then be out of sync by ~4 µs. This means the entire bit length must be at least 40 µs (36 µs for the 10th sample + 4 µs for synchronization with the internal sampling clock).

Clock deviation causes

The causes which contribute to the total deviation are:

- D_{TRA}: Deviation due to transmitter error (Local oscillator error of the transmitter or the transmitter is transmitting at a different baud rate).
- D_{QUANT}: Error due to the baud rate quantisation of the receiver.
- D_{REC}: Deviation of the local oscillator of the receiver: This deviation can occur during the reception of one complete SCI message assuming that the deviation has been compensated at the beginning of the message.
- D_{TCL}: Deviation due to the transmission line (generally due to the transceivers)

All the deviations of the system should be added and compared to the SCI clock tolerance:

$D_{TRA} + D_{QUANT} + D_{REC} + D_{TCL} < 3.75\%$

Noise error causes

See also description of Noise error in *Section* .

Start bit

The noise flag (NF) is set during start bit reception if one of the following conditions occurs:

1. A valid falling edge is not detected. A falling edge is considered to be valid if the 3 consecutive samples before the falling edge occurs are detected as '1' and, after the falling edge occurs, during the sampling of the 16 samples, if one of the samples numbered 3, 5 or 7 is detected as a “1”.
2. During sampling of the 16 samples, if one of the samples numbered 8, 9 or 10 is detected as a “1”.

Therefore, a valid Start Bit must satisfy both the above conditions to prevent the Noise Flag getting set.

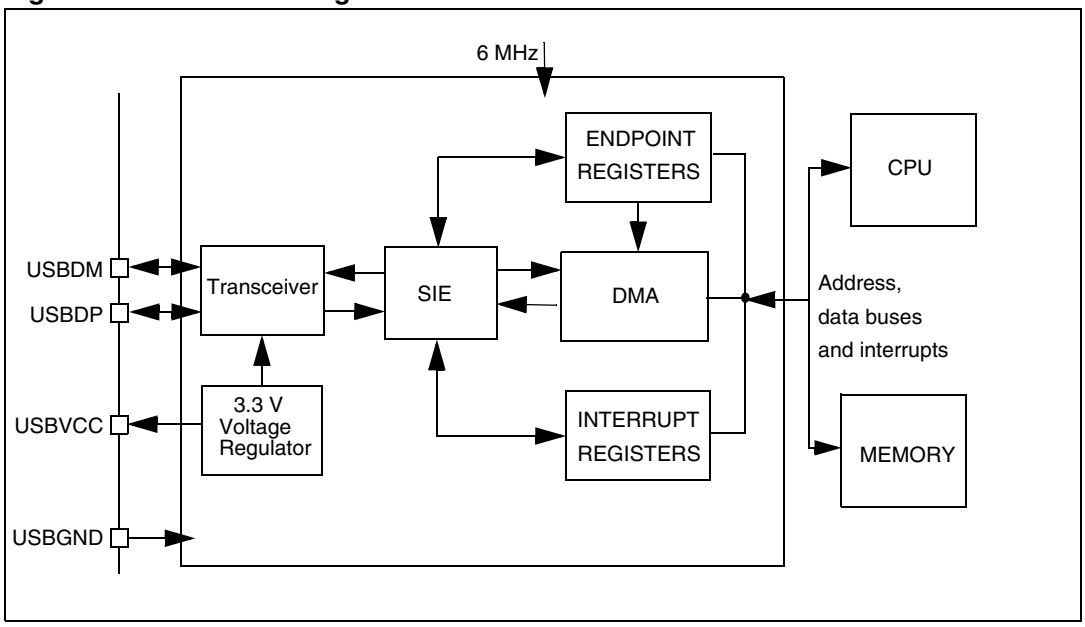
Data bits

The noise flag (NF) is set during normal data bit reception if the following condition occurs:

- During the sampling of 16 samples, if all three samples numbered 8, 9 and10 are not the same. The majority of the 8th, 9th and 10th samples is considered as the bit value.

Therefore, a valid Data Bit must have samples 8, 9 and 10 at the same value to prevent the Noise Flag getting set.

Figure 44. USB block diagram



11.4.4 Register description

DMA Address register (DMAR)

Reset value: undefined

7							0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
Read.write							

[7:0] **DA[15:8]** *DMA address bits 15-8.*

Software must write the start address of the DMA memory area whose most significant bits are given by DA15-DA6. The remaining 6 address bits are set by hardware. See the description of the IDR register and *Figure 45*.

Control register (CTLR)

Reset value: 0000 0110 (06h)

7							0
0	0	0	0	RESUME	PDWN	FSUSP	FRES
Read/write							

[7:4] Reserved. Forced by hardware to 0.

3 **RESUME** *Resume*.
This bit is set by software to wakeup the Host when the ST7 is in suspend mode.
0: Resume signal not forced
1: Resume signal forced on the USB bus.
Software should clear this bit after the appropriate delay.

2 **PDWN** *Power down*.
This bit is set by software to turn off the 3.3 V on-chip voltage regulator that supplies the external pull-up resistor and the transceiver.
0: Voltage regulator on
1: Voltage regulator off
Note: After turning on the voltage regulator, software should allow at least 3 µs for stabilization of the power supply before using the USB interface.

1 **FSUSP** *Force suspend mode*.
This bit is set by software to enter Suspend mode. The ST7 should also be halted allowing at least 600 ns before issuing the HALT instruction.
0: Suspend mode inactive
1: Suspend mode active
When the hardware detects USB activity, it resets this bit (it can also be reset by software).

0 **FRES** *Force reset.*
This bit is set by software to force a reset of the USB interface, just as if a RESET sequence came from the USB.
0: Reset not forced
1: USB interface reset forced.
The USB is held in RESET state until software clears this bit, at which point a “USB-RESET” interrupt will be generated if enabled.

Device Address register (DADDR)

Reset value: 0000 0000 (00h)

7							0
0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
Read.write							

7 Reserved. Forced by hardware to 0.

[6:0] **ADD[6:0]** *Device address, 7 bits.*
Software must write into this register the address sent by the host during enumeration.
Note: This register is also reset when a USB reset is received from the USB bus or forced through bit FRES in the CTLR register.

Endpoint n register A (EPnRA)

These registers (EP0RA, EP1RA and EP2RA) are used for controlling data transmission. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RA register are not available on some devices (see device feature list and register map).

Reset value: 0000 xxxx (0xh)

7							0
ST_ OUT	DTOG _TX	STAT _TX1	STAT _TX0	TBC3	TBC2	TBC1	TBC0
Read.write							

7 **ST_OUT** *Status out.*
This bit is set by software to indicate that a status out packet is expected: in this case, all nonzero OUT data transfers on the endpoint are STALLed instead of being ACKed. When ST_OUT is reset, OUT transactions can have any number of bytes, as needed.

6 **DTOG_TX** *Data Toggle, for transmission transfers.*
It contains the required value of the toggle bit (0=DATA0, 1=DATA1) for the next transmitted data packet. This bit is set by hardware at the reception of a SETUP PID. DTOG_TX toggles only when the transmitter has received the ACK signal from the USB host. DTOG_TX and also DTOG_RX (see EPnRB) are normally updated by hardware, at the receipt of a relevant PID. They can be also written by software.

[5:4] **STAT_TX[1:0]** *Status bits, for transmission transfers.*
These bits contain the information about the endpoint status, which are listed in *Table 34*.

These bits are written by software. Hardware sets the STAT_TX bits to NAK when a correct transfer has occurred (CTR=1) related to a IN or SETUP transaction addressed to this endpoint; this allows the software to prepare the next set of data to be transmitted.

[3:0] **TBC[3:0]** *Transmit byte count for Endpoint n.*
Before transmission, after filling the transmit buffer, software must write in the TBC field the transmit packet size expressed in bytes (in the range 0-8).

Caution: Any value outside the range 0-8 willinduce undesired effects (such as continuous data transmission).

Table 34. STAT_TX bit definition

STAT_TX1	STAT_TX0	Meaning
0	0	**DISABLED:** transmission transfers cannot be executed.
0	1	**STALL**: the endpoint is stalled and all transmission requests result in a STALL handshake.
1	0	**NAK**: the endpoint is naked and all transmission requests result in a NAK handshake.
1	1	**VALID**: this endpoint is enabled for transmission.

Setting the START bit while the BUSY bit is cleared causes the interface to switch to Master mode (M/SL bit set) and generates a Start condition.

Once the Start condition is sent, the EVF and SB bits are set by hardware with an interrupt if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address byte, **holding the SCL line low** (see *Figure 48* Transfer sequencing EV5).

Slave address transmission

Then the slave address byte is sent to the SDA line via the internal shift register.

After completion of this transfer (and acknowledge from the slave if the ACK bit is set), the EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see *Figure 48* Transfer sequencing EV6).

Next the master must enter Receiver or Transmitter mode.

Master receiver

Following the address transmission and after the SR1 and CR registers have been accessed, the master receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- Acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see *Figure 48* Transfer sequencing EV7).

To close the communication: before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

Master transmitter

Following the address transmission and after SR1 register has been read, the master sends bytes from the DR register to the SDA line via the internal shift register.

The master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see *Figure 48* Transfer sequencing EV8).

When the acknowledge bit is received, the interface sets, EVF and BTF bits with an interrupt if the ITE bit is set.

To close the communication: after writing the last byte to the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

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

Software procedure

Refer to the control/status register (CSR) and data register (DR) in *Section 11.6.6* for the bit definitions and to *Figure 51* for the timings.

ADC configuration

The total duration of the A/D conversion is 12 ADC clock periods ($1/f_{ADC}=4/f_{CPU}$).

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

In the CSR register:

- Select the CH[3:0] bits to assign the analog channel to be converted.

ADC conversion

In the CSR register:

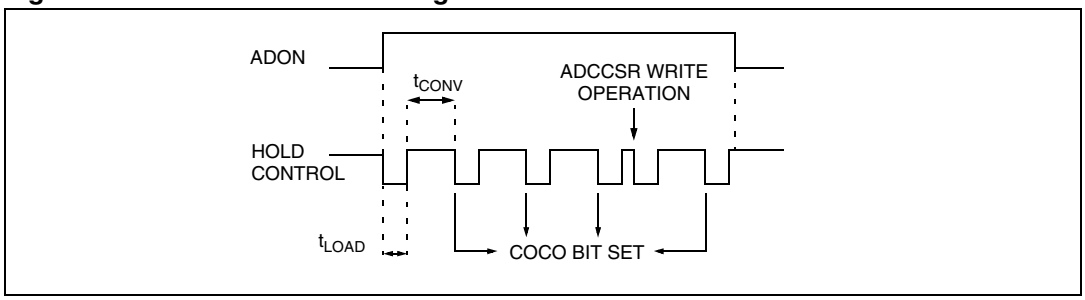
- Set the ADON bit to enable the A/D converter and to start the first conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

- The COCO bit is set by hardware.
- No interrupt is generated.
- The result is in the DR register and remains valid until the next conversion has ended.

A write to the CSR register (with ADON set) aborts the current conversion, resets the COCO bit and starts a new conversion.

Figure 51. ADC conversion timings



11.6.4 Low power modes

Table 44. Low power modes

Mode	Description
WAIT	No effect on A/D Converter
HALT	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilization time before accurate conversions can be performed.

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

Figure 64. V_{OL} high sink V_{DD}=5 V

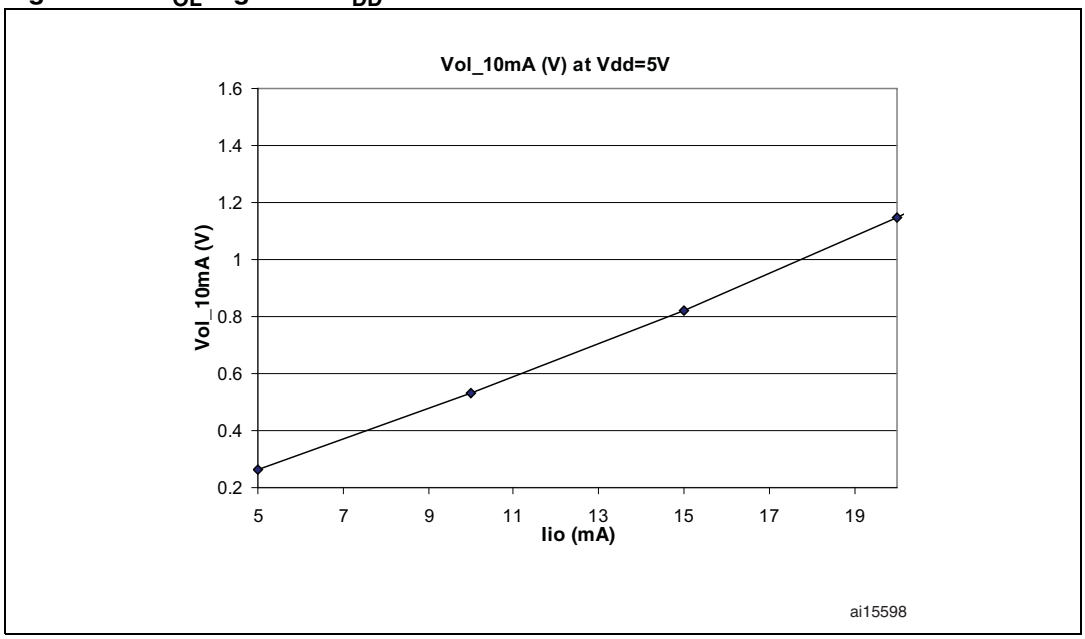


Figure 65. V_{OL} very high sink V_{DD}=5 V