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	CANbus, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K 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/st72f521r7ta

3 Register and memory map

As shown in *Figure 4*, the MCU is capable of addressing 64 Kbytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, up to 2 Kbytes of RAM and up to 60 Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

The highest address bytes contain the user reset and interrupt vectors.

IMPORTANT: Memory locations marked as “Reserved” must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Related documentation

*Executing Code in ST7 RAM (*AN 985)

Figure 4. Memory map

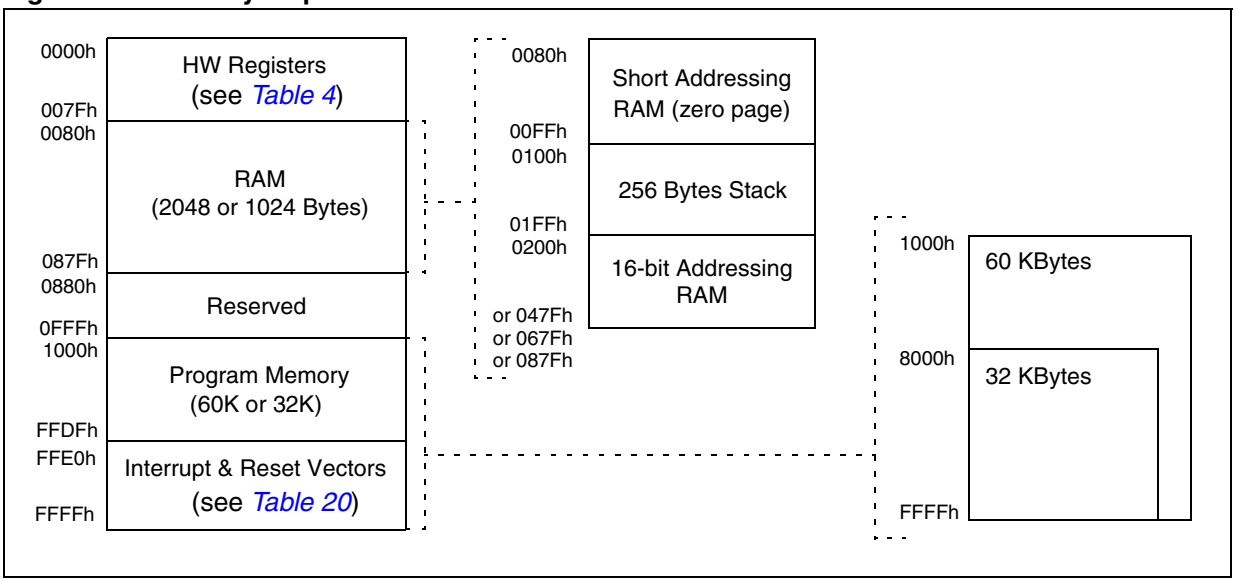


Table 4. Hardware register map

Address	Block	Register label	Register name	Reset status	Remarks
0000h	Port A	PADR	Port A Data Register	00h(1)	R/W
0001h		PADDR	Port A Data Direction Register	00h	R/W
0002h		PAOR	Port A Option Register	00h	R/W
0003h	Port B	PBDR	Port B Data Register	00h(1)	R/W
0004h		PBDDR	Port B Data Direction Register	00h	R/W
0005h		PBOR	Port B Option Register	00h	R/W
0006h	Port C	PCDR	Port C Data Register	00h(1)	R/W
0007h		PCDDR	Port C Data Direction Register	00h	R/W
0008h		PCOR	Port C Option Register	00h	R/W
0009h	Port D	PDDR	Port D Data Register	00h(1)	R/W
000Ah		PDDDR	Port D Data Direction Register	00h	R/W
000Bh		PDOR	Port D Option Register	00h	R/W

5.3.1 Accumulator (A)

The accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations as well as data manipulations.

5.3.2 Index registers (X and Y)

These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation (the Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures.

5.3.3 Program counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

5.3.4 Condition code (CC) register

The 8-bit condition code register contains the interrupt masks and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

CC Reset value: 111x1xxx

7	6	5	4	3	2	1	0
1	1	I1	H	I0	N	Z	C
		RW	RW	RW	RW	RW	RW

Table 7. Arithmetic management bits

Bit	Name	Function
4	H	*Half carry* This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions. 0: No half carry has occurred. 1: A half carry has occurred. This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.
2	N	*Negative* This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the result 7th bit. 0: The result of the last operation is positive or null. 1: The result of the last operation is negative (that is, the most significant bit is a logic 1). This bit is accessed by the JRMI and JRPL instructions.

6.6.5 System Integrity (SI) Control/Status register (SICSR)

SICSR Reset value: 000x 000x (00h)

7	6	5	4	3	2	1	0
AVDS	AVDIE	AVDF	LVDRF	Reserved			WDGRF
RW	RW	RW	RW	-			RW

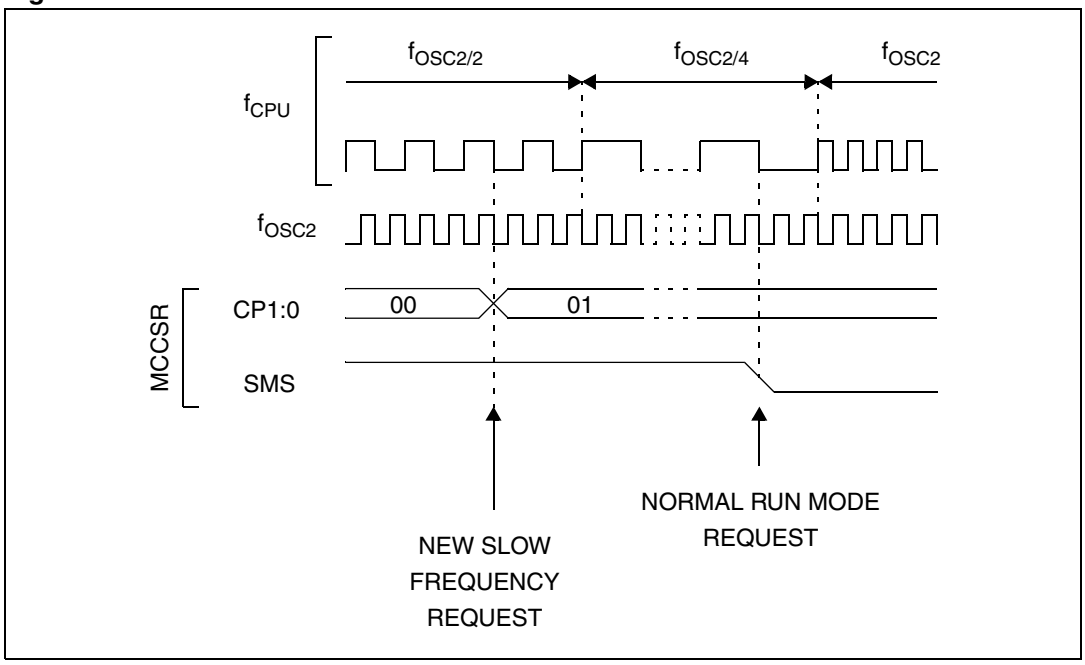
Table 13. SICSR description

Bit	Name	Function
7	AVDS	*Voltage Detection selection* This bit is set and cleared by software. Voltage Detection is available only if the LVD is enabled by option byte. 0: Voltage detection on V_{DD} supply 1: Voltage detection on EVD pin
6	AVDIE	*Voltage Detector interrupt enable* This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag changes (toggles). The pending interrupt information is automatically cleared when software enters the AVD interrupt routine. 0: AVD interrupt disabled 1: AVD interrupt enabled
5	AVDF	*Voltage Detector flag* This read-only bit is set and cleared by hardware. If the AVDIE bit is set, an interrupt request is generated when the AVDF bit changes value. Refer to *Figure 15* and to *Monitoring the VDD main supply on page 47* for additional details. 0: V_{DD} or V_{EVD} over $V_{IT+(AVD)}$ threshold 1: V_{DD} or V_{EVD} under $V_{IT-(AVD)}$ threshold
4	LVDRF	*LVD reset flag* This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (writing zero). See *Table 14: Reset source flags* for more details. When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.
3:1	-	Reserved, must be kept cleared.
0	WDGRF	*Watchdog reset flag* This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (writing zero) or an LVD Reset (to ensure a stable cleared state of the WDGRF flag when CPU starts). Combined with the LVDRF flag information, the flag description is given in *Table 14*.

Table 14. Reset source flags

Reset sources	LVDRF	WDGRF
External $\overline{\text{RESET}}$ pin	0	0
Watchdog	0	1
LVD	1	X

Figure 23. Slow mode clock transitions



8.3 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the 'WFI' instruction.

All peripherals remain active. During Wait mode, the I[1:0] bits of the CC register are forced to '10', to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in Wait mode until an interrupt or RESET occurs, whereupon the Program Counter branches to the starting address of the interrupt or Reset service routine.

The MCU will remain in Wait mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to the following *Figure 24*.

11 Main clock controller with real-time clock and beeper (MCC/RTC)

11.1 Introduction

The Main Clock Controller consists of three different functions:

- a programmable CPU clock prescaler
- a clock-out signal to supply external devices
- a real-time clock timer with interrupt capability

Each function can be used independently and simultaneously.

11.2 Programmable CPU clock prescaler

The programmable CPU clock prescaler supplies the clock for the ST7 CPU and its internal peripherals. It manages Slow power saving mode (see *Section 8.2: Slow mode on page 65* for more details).

The prescaler selects the f_{CPU} main clock frequency and is controlled by three bits in the MCCSR register: CP[1:0] and SMS.

11.3 Clock-out capability

The clock-out capability is an alternate function of an I/O port pin that outputs a f_{CPU} clock to drive external devices. It is controlled by the MCO bit in the MCCSR register.

Caution: When selected, the clock out pin suspends the clock during Active Halt mode.

11.4 Real-time clock timer (RTC)

The counter of the real-time clock timer allows an interrupt to be generated based on an accurate real-time clock. Four different time bases depending directly on f_{OSC2} are available. The whole functionality is controlled by four bits of the MCCSR register: TB[1:0], OIE and OIF.

When the RTC interrupt is enabled (OIE bit set), the ST7 enters Active Halt mode when the HALT instruction is executed. See *Section 8.4: Active Halt and Halt modes on page 68* for more details.

11.5 Beeper

The beep function is controlled by the MCCBCR register. It can output three selectable frequencies on the BEEP pin (I/O port alternate function).

Figure 37. PWM auto-reload timer function

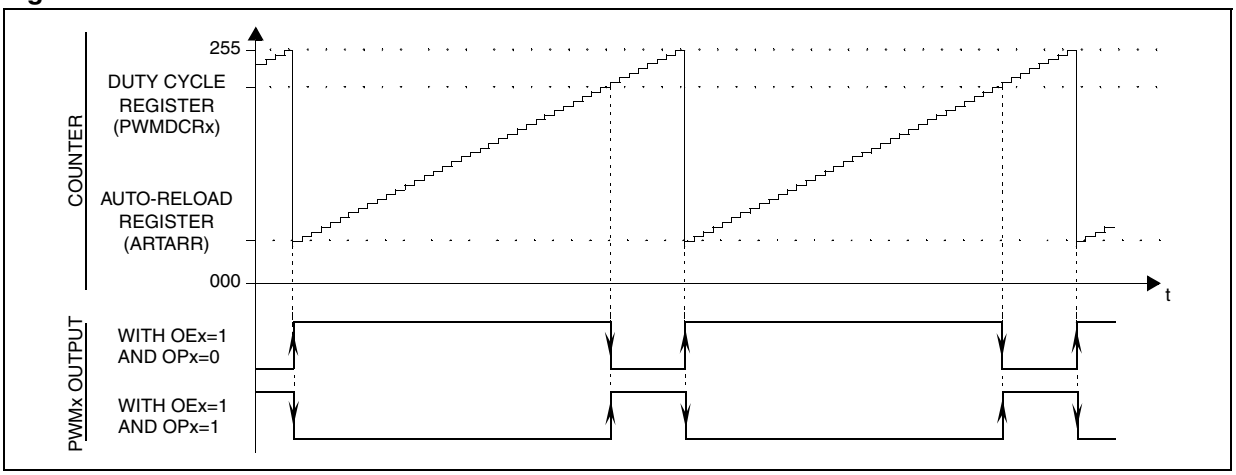
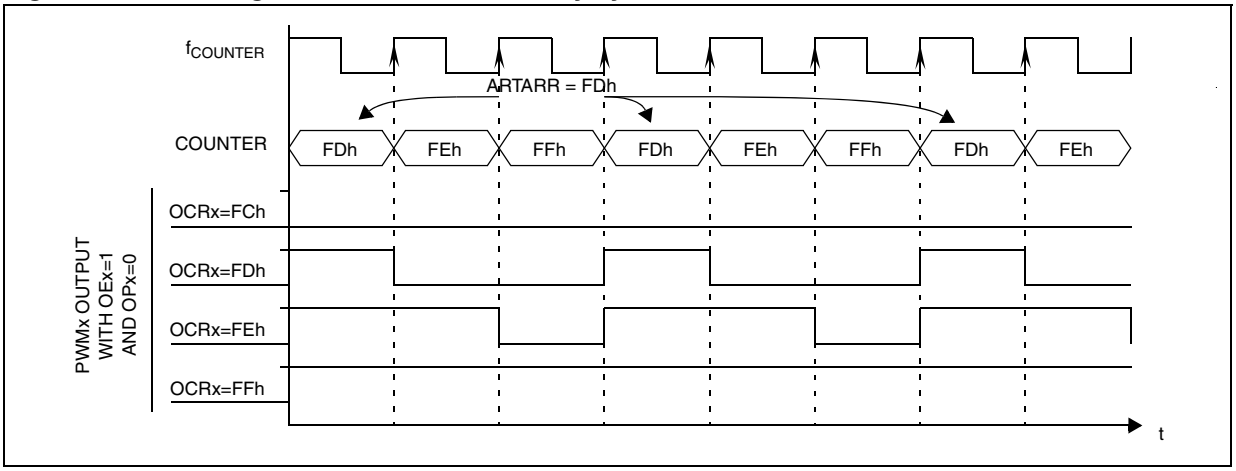


Figure 38. PWM signal from 0% to 100% duty cycle



12.2.6 Output compare and time base interrupt

On overflow, the OVF flag of the ARTCSR register is set and an overflow interrupt request is generated if the overflow interrupt enable bit, OIE, in the ARTCSR register, is set. The OVF flag must be reset by the user software. This interrupt can be used as a time base in the application.

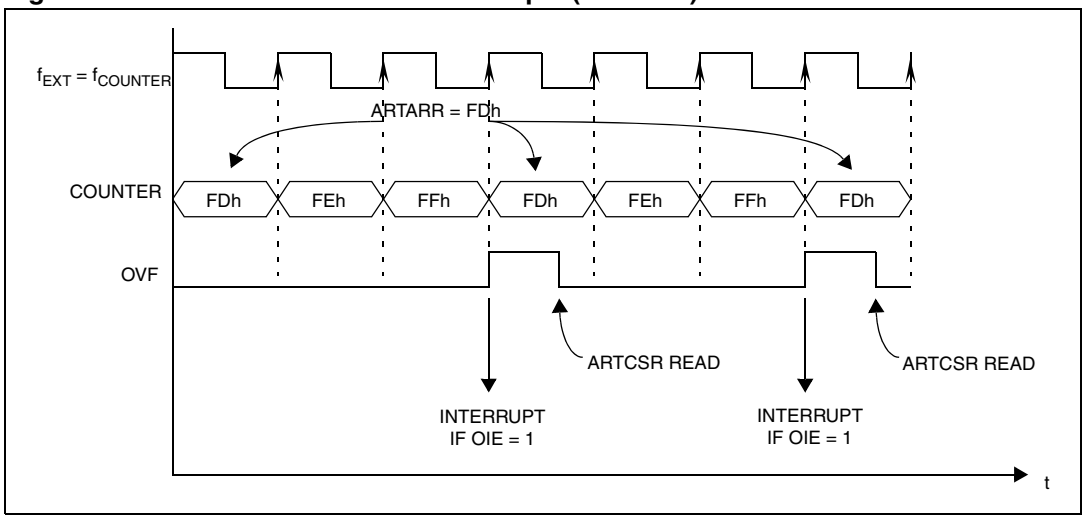
12.2.7 External clock and event detector mode

Using the f_{EXT} external prescaler input clock, the auto-reload timer can be used as an external clock event detector. In this mode, the ARTARR register is used to select the n_{EVENT} number of events to be counted before setting the OVF flag.

$$n_{EVENT} = 256 - ARTARR$$

Caution: The external clock function is not available in Halt mode. If Halt mode is used in the application, prior to executing the HALT instruction, the counter must be disabled by clearing the TCE bit in the ARTCSR register to avoid spurious counter increments.

Figure 39. External event detector example (3 counts)



12.2.8 Input capture function

This mode allows the measurement of external signal pulse widths through ARTICRx registers.

Each input capture can generate an interrupt independently on a selected input signal transition. This event is flagged by a set of the corresponding CFx bits of the Input Capture Control/Status register (ARTICCSR).

These input capture interrupts are enabled through the CIEx bits of the ARTICCSR register.

The active transition (falling or rising edge) is software programmable through the CSx bits of the ARTICCSR register.

The read only input capture registers (ARTICRx) are used to latch the auto-reload counter value when a transition is detected on the ARTICx pin (CFx bit set in ARTICCSR register). After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

Note: After a capture detection, data transfer in the ARTICRx register is inhibited until it is read (clearing the CFx bit).

The timer interrupt remains pending while the CFx flag is set when the interrupt is enabled (CIEx bit set). This means that the ARTICRx register has to be read at each capture event to clear the CFx flag.

The timing resolution is given by auto-reload counter cycle time ($1/f_{COUNTER}$).

Note: During Halt mode, if both the input capture and the external clock are enabled, the ARTICRx register value is not guaranteed if the input capture pin and the external clock change simultaneously.

13.3 Functional description

13.3.1 Counter

The main block of the Programmable Timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high and low.

Counter Register (CR)

- Counter High Register (CHR) is the most significant byte (MS Byte)
- Counter Low Register (CLR) is the least significant byte (LS Byte)

Alternate Counter Register (ACR)

- Alternate Counter High Register (ACHR) is the most significant byte (MS Byte)
- Alternate Counter Low Register (ACLR) is the least significant byte (LS Byte)

These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register (SR) (see note at the end of paragraph entitled *16-bit read sequence*).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.

Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in One Pulse mode and PWM mode.

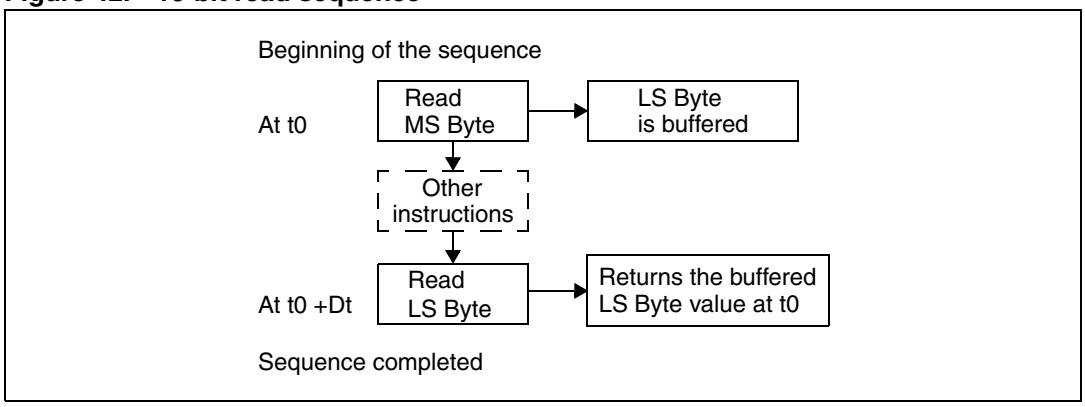
The timer clock depends on the clock control bits of the CR2 register, as illustrated in *Table 61: Timer clock selection*. The value in the counter register repeats every 131072, 262144 or 524288 CPU clock cycles depending on the CC[1:0] bits.

The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

16-bit read sequence

The 16-bit read sequence (from either the Counter Register or the Alternate Counter Register) is illustrated in *Figure 42*.

Figure 42. 16-bit read sequence



The user must read the MS Byte first; the LS Byte value is then buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MS Byte several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LS Byte of the count value at the time of the read.

Whatever timer mode is used (input capture, output compare, one pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFFFh to 0000h, after which

- the TOF bit of the SR register is set
- a timer interrupt is generated if
  - the TOIE bit of the CR1 register is set and
  - the I bit of the CC register is cleared

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done in two steps:

1. Reading the SR register while the TOF bit is set
2. An access (read or write) to the CLR register

Note: The TOF bit is not cleared by accesses to ACLR register. The advantage of accessing the ACLR register rather than the CLR register is that it allows simultaneous use of the overflow function and reading the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by Wait mode.

In Halt mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a Reset).

The OC1R register value required for a specific timing application can be calculated using the following formula:

$$OC\mathit{i}R\ value = \frac{t * f_{CPU} - 5}{PRESC}$$

Where:

t = Pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits; see *Table 61: Timer clock selection*)

If the timer clock is an external clock the formula is:

$$OC\mathit{i}R = t * f_{EXT} - 5$$

Where:

t = Pulse period (in seconds)

f_{EXT} = External clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see *Figure 52*).

Note: 1 The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.

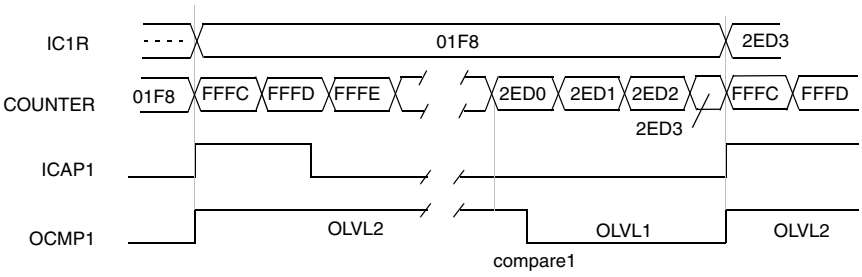
2 When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

3 If OLVL1 = OLVL2 a continuous signal will be seen on the OCMP1 pin.

4 The ICAP1 pin cannot be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.

5 When one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

Figure 52. One pulse mode timing example



Note: IEDG1 = 1, OC1R = 2ED0h, OLVL1 = 0, OLVL2 = 1

Figure 57. Generic $\overline{SS}$ timing diagram

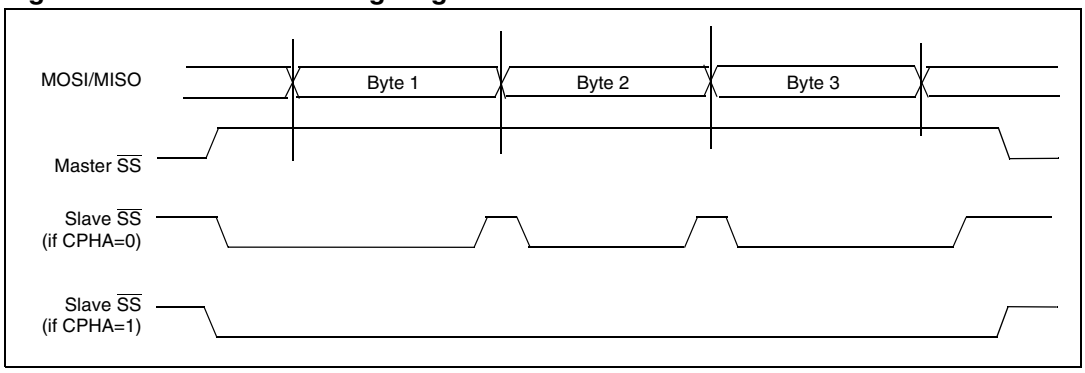
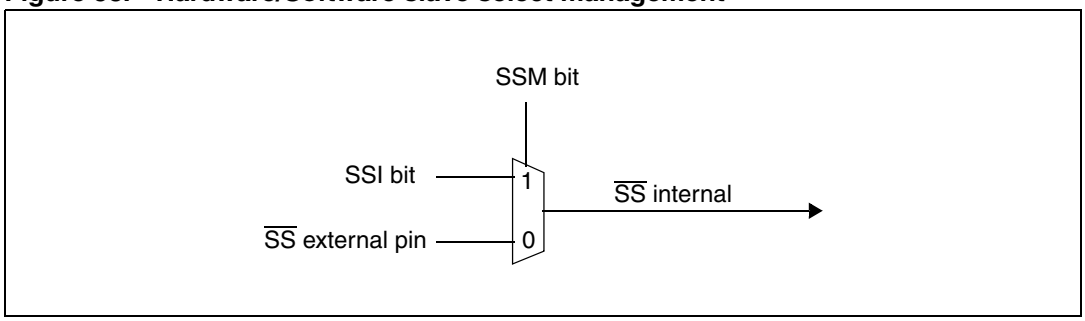


Figure 58. Hardware/Software slave select management



14.3.3 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:
   a) Select the clock frequency by configuring the SPR[2:0] bits.
   b) Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. *Figure 59* shows the four possible configurations.

Note: The slave must have the same CPOL and CPHA settings as the master.

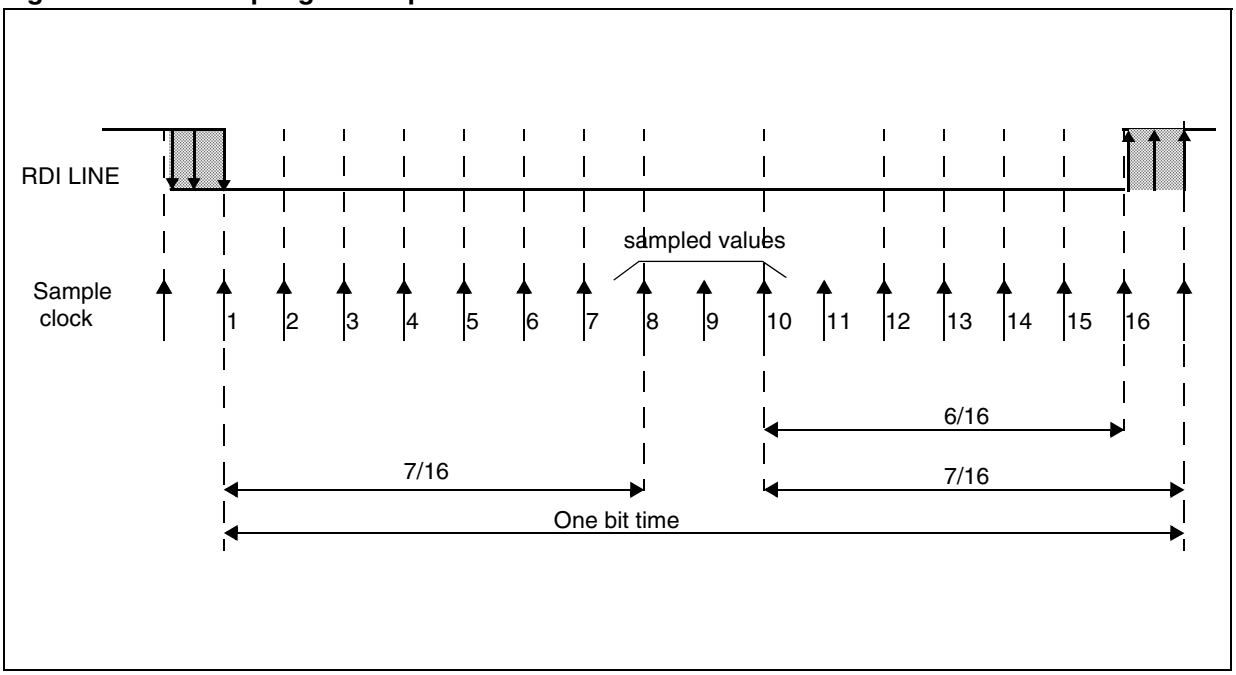
2. Write to the SPICSR register:
   Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the $\overline{SS}$ pin high for the complete byte transmit sequence.
3. Write to the SPICR register:
   Set the MSTR and SPE bits

Note: MSTR and SPE bits remain set only if $\overline{SS}$ is high).

IMPORTANT: If the SPICSR register is not written first, the SPICR register setting (MSTR bit) may not be taken into account.

The transmit sequence begins when software writes a byte in the SPIDR register.

Figure 65. Bit sampling in reception mode



15.5 Low power modes

Table 71. Effect of low power modes on SCI

Mode	Effect
Wait	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
Halt	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

15.6 Interrupts

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 72. SCI interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE	Yes	No
Received Data Ready to be Read	RDRF	RIE	Yes	No
Overrun Error Detected	OR		Yes	No

Table 83. CR register description (continued)

Bit	Name	Function
1	STOP	*Generation of a Stop condition* This bit is set and cleared by software. It is also cleared by hardware in master mode. *Note: This bit is not cleared when the interface is disabled (PE = 0).* **In Master mode** 0: No stop generation 1: Stop generation after the current byte transfer or after the current Start condition is sent. The STOP bit is cleared by hardware when the Stop condition is sent. **In Slave mode** 0: No stop generation 1: Release the SCL and SDA lines after the current byte transfer (BTF = 1). In this mode the STOP bit has to be cleared by software.
0	ITE	*Interrupt enable* This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE = 0). 0: Interrupts disabled 1: Interrupts enabled Refer to *Figure 69* and *Table 82* for the relationship between the events and the interrupt. SCL is held low when the ADD10, SB, BTF or ADSL flags or an EV6 event (see *Figure 68*) is detected.

16.7.2 I²C status register 1 (SR1)

SR1 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
EVF	ADD10	TRA	BUSY	BTF	ADSL	M/SL	SB
RO	RO	RO	RO	RO	RO	RO	RO

Table 84. SR1 register description

Bit	Name	Function
7	EVF	*Event flag* This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in *Figure 68*. It is also cleared by hardware when the interface is disabled (PE = 0). 0: No event 1: One of the following events has occurred: - BTF = 1 (Byte received or transmitted) - ADSL = 1 (Address matched in Slave mode while ACK = 1) - SB = 1 (Start condition generated in Master mode) - AF = 1 (No acknowledge received after byte transmission) - STOPF = 1 (Stop condition detected in Slave mode) - ARLO = 1 (Arbitration lost in Master mode) - BERR = 1 (Bus error, misplaced Start or Stop condition detected) - ADD10 = 1 (Master has sent header byte) - Address byte successfully transmitted in Master mode

Table 88. OAR1 register description

Bit	Name	Function	
		7-bit addressing mode	10-bit addressing mode
7:1	ADD[7:1]	*Interface address* These bits define the I^2C bus address of the interface. They are not cleared when the interface is disabled (PE = 0).	Not applicable
0	ADD0	*Address direction bit* This bit is 'don't care', the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE = 0). Address 01h is always ignored.	
7:0	ADD[7:0]	Not applicable	*Interface address* These are the least significant bits of the I^2C bus address of the interface. They are not cleared when the interface is disabled (PE = 0).

16.7.7 I^2C own address register (OAR2)

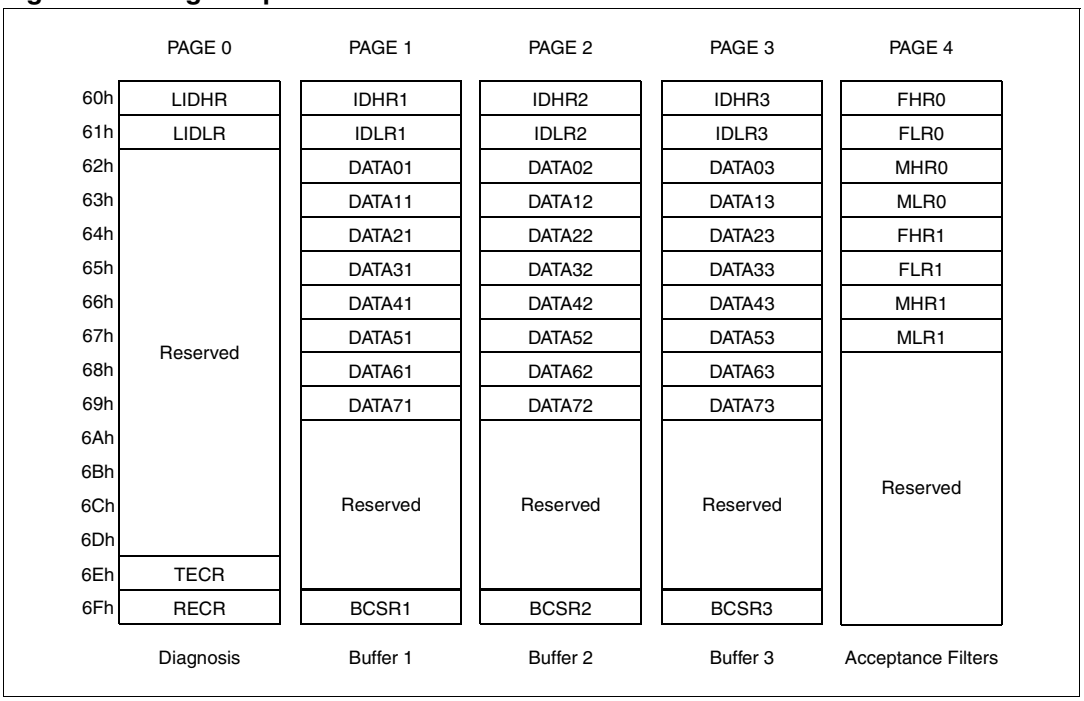
OAR2 Reset value: 0100 0000 (40h)

7	6	5	4	3	2	1	0
FR[1:0]		Reserved			ADD[9:8]		Reserved
RW		-			RW		-

Table 89. OAR2 register description

Bit	Name	Function
7:6	FR[1:0]	*Frequency bits* These bits are set by software only when the interface is disabled (PE = 0). To configure the interface to I^2C specified delays, select the value corresponding to the CPU frequency f_{CPU}. 00: f_{CPU} < 6 MHz 01: f_{CPU} = 6 to 8 MHz
5:3	-	Reserved
2:1	ADD[9:8]	*Interface address* These are the most significant bits of the I^2C bus address of the interface (10-bit mode only). They are not cleared when the interface is disabled (PE = 0).
0	-	Reserved

Figure 76. Page maps



20.7 EMC (electromagnetic compatibility) characteristics

Susceptibility tests are performed on a sample basis during product characterization.

20.7.1 Functional EMS (electromagnetic susceptibility)

Based on a simple running application on the product (toggling two LEDs through I/O ports), the product is stressed by two electromagnetic events until a failure occurs (indicated by the LEDs).

- **ESD**: Electrostatic discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- **FTB**: A burst of fast transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100pF capacitor until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed. The test results given in *Table 142* below are based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

EMC characterization and optimization are performed at component level with a typical application environment and simplified MCU software. It should be noted that good EMC performance is highly dependent on the user application and the software in particular.

Therefore it is recommended that the user applies EMC software optimization and prequalification tests in relation with the EMC level requested for his application.

Software recommendations

The software flowchart must include the management of runaway conditions such as:

- Corrupted program counter
- Unexpected reset
- Critical Data corruption (control registers...)

Prequalification trials

Most of the common failures (unexpected reset and program counter corruption) can be reproduced by manually forcing a low state on the $\overline{\text{RESET}}$ pin or the oscillator pins for 1 second.

To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behavior is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

20.11.2 I²C - inter IC control interface

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

Refer to *Section 20.8: I/O port pin characteristics* for more details on the input/output alternate function characteristics (SDAI and SCLI). The ST7 I2C interface meets the requirements of the standard I2C communication protocol described in the following table.

Table 153. I²C control interface characteristics

Symbol	Parameter	Standard mode I²C		Fast mode I²C(1)		Unit
		Min(2)	Max(2)	Min(2)	Max(2)	
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		µs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		
$t_{su(SDA)}$	SDA setup time	250		100		ns
$t_{h(SDA)}$	SDA data hold time	0(3)		0(4)	900(3)	
$t_{r(SDA)}$ $t_{r(SCL)}$	SDA and SCL rise time		1000	20+0.1C_b	300	
$t_{f(SDA)}$ $t_{f(SCL)}$	SDA and SCL fall time		300			
$t_{h(STA)}$	START condition hold time	4.0		0.6		µs
$t_{su(STA)}$	Repeated START condition setup time	4.7				
$t_{su(STO)}$	STOP condition setup time	4.0				
$t_{w(STO:STA)}$	STOP to START condition time (bus free)	4.7		1.3		
C_b	Capacitive load for each bus line		400		400	pF

1. At 4 MHz f_{CPU}, maximum I²C speed (400 kHz) is not achievable. In this case, maximum I²C speed will be approximately 260 kHz.
2. Data based on standard I²C protocol requirement, not tested in production.
3. The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal.
4. The device must internally provide a hold time of at least 300ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.

20.11.3 CAN - Controller area network interface

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified. Refer to *Chapter 9: I/O ports* for more details on the input/output alternate function characteristics (CANTX and CANRX).

Table 155. CAN characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{p(RX:TX)}$	CAN controller propagation time(1)				60	ns

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

20.12 10-bit ADC characteristics

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

Table 156. 10-bit ADC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{ADC}	ADC clock frequency		0.4		2	MHz
V_{AREF}	Analog reference voltage	$0.7*V_{DD} \le V_{AREF} \le V_{DD}$	3.8		V_{DD}	V
V_{AIN}	Conversion voltage range		V_{SSA}		V_{AREF}	
I_{lkg}	Positive input leakage current for analog input(1)	$-40°C \le T_A \le 85°C$ range			±250	nA
		Other T_A ranges			±1	µA
R_{AIN}	External input impedance(2)				See *Figure 112* and *Figure 113*	kΩ
C_{AIN}	External capacitor on analog input					pF
f_{AIN}	Variation frequency of analog input signal					Hz
C_{ADC}	Internal sample and hold capacitor			12		pF
t_{ADC}	Conversion time (Sample + Hold) f_{CPU} = 8 MHz, speed = 0, f_{ADC} = 2 MHz		7.5			µs
t_{ADC}	No. of sample capacitor loading cycles		4			$1/f_{ADC}$
	No. of hold conversion cycles		11			

1. Injecting negative current on adjacent pins may result in increased leakage currents. Software filtering of the converted analog value is recommended.
2. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.

Table 164. Option byte 1 bit description (continued)

Bit	Name	Function
OPT5:4	OSCTYPE[1:0]	Oscillator type These option bits select the ST7 main clock source type. 00: Clock source = Resonator oscillator 01: Reserved 10: Clock source = Internal RC oscillator 11: Clock source = External source
OPT3:1	OSCRANGE[2:0]	Oscillator range When the resonator oscillator type is selected, these option bits select the resonator oscillator current source corresponding to the frequency range of the used resonator. Otherwise, these bits are used to select the normal operating frequency range (see *Table 166: Oscillator frequency range selection (OPT3:1)*).
OPT0	PLLOFF	PLL activation This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL must not be used with the internal RC oscillator or with external clock source. The PLL is guaranteed only with an input frequency between 2 and 4 MHz. 0: PLL x2 enabled 1: PLL x2 disabled **Caution**: The PLL can be enabled only if the “OSCRANGE” (OPT3:1) bits are configured to “MP - 2~4 MHz”. Otherwise, the device functionality is not guaranteed.

Table 165. Package selection (OPT7)

Version	Selected package	PKG1	PKG0
M	LQFP80	1	1
(A)R	LQFP64	1	0

Note: On the chip, each I/O port has 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.

Table 166. Oscillator frequency range selection (OPT3:1)

Typical frequency range		OSCRANGE		
		2	1	0
LP	1~2 MHz	0	0	0
MP	2~4 MHz	0	0	1
MS	4~8 MHz	0	1	0
HS	8~16 MHz	0	1	1

22.2 ROM device ordering information and transfer of customer code

Customer code is made up of the ROM/FASTROM contents and the list of the selected options (if any). The ROM/FASTROM contents are to be sent on diskette, or by electronic means, with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

Complete the appended *ST72521-Auto Microcontroller FASTROM/ROM Option List on page 264* to communicate the selected options to STMicroelectronics and check for regular updates of the option list on the ST website or ask your ST representative.

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

The following *Figure 122* and *Figure 123* serve as guides for ordering. The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Caution: The Readout Protection binary value is inverted between ROM and Flash products. The option byte checksum will differ between ROM and Flash.