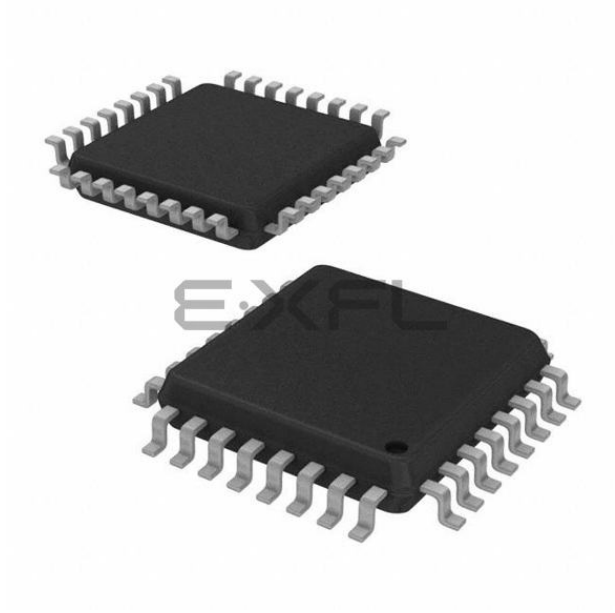


Welcome to E-XFL.COM

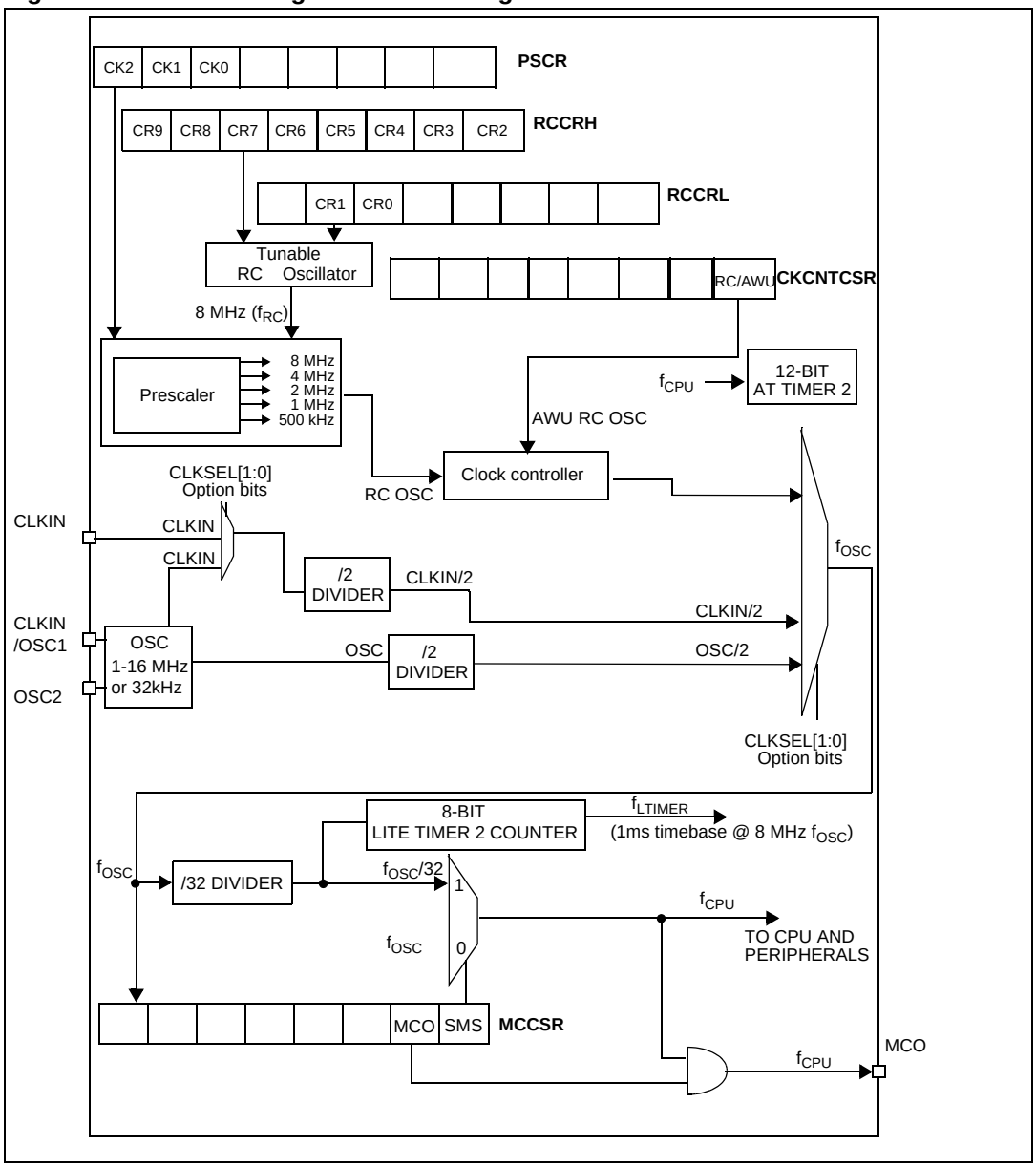
What is "Embedded - Microcontrollers"?

"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, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7foxk2t6

Figure 12. Clock management block diagram



6.2 Multi-oscillator (MO)

The main clock of the ST7 can be generated by four different source types coming from the multi-oscillator block (1 to 16 MHz):

- An external source
- 5 different configurations for crystal or ceramic resonator oscillators
- An internal high frequency RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configurations are shown in *Table 8*. Refer to the electrical characteristics section for more details.

Bit 2 = **RC_FLAG** *RC Selection bit*

This bit is set and cleared by hardware.

0: No switch from RC to AWU requested

1: RC clock activated and temporization completed

Bit 1 = Reserved, must be kept cleared.

Bit 0 = **RC/AWU** *RC/AWU Selection bit*

0: RC enabled

1: AWU enabled (default value)

Table 12. Clock register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0035h	RCC_CSR	- 0	- 0	- 0	- 0	- 0	- 0	RCCLAT 0	RCCPGM 0
003Ah	MCCSR Reset Value	- 0	- 0	- 0	- 0	- 0	- 0	MCO 0	SMS 0
003Bh	RCCRH Reset Value	CR9 1	CR8 1	CR7 1	CR6 1	CR5 1	CR4 1	CR3 1	CR2 1
003Ch	RCCRL Reset Value	- 0	CR1 1	CR0 1	WDGRF 0	- 0	LVDRF x	- 0	- 0
003Dh	PSCR Reset Value	CK2 0	CK1 0	CK0 0	- 0	- 0	- 0	- 0 or 1	- 0 or 1
0051h	CKCNTCSR Reset Value	- 0	- 0	- 0	- 0	AWU_ FLAG 1	RC_FLAG 0	- 0	RC/AWU 1

7.5 Description of interrupt registers

7.5.1 CPU CC register interrupt bits

Reset value: 111x 1010(xAh)

7							0
1	1	I1	H	I0	N	Z	C
Read/write							

Bits 5, 3 = **I1, I0** *Software Interrupt Priority bits*

These two bits indicate the current interrupt software priority (see *Table 14*).

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (ISPRx).

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see *Table 16: Dedicated interrupt instruction set*).

TRAP and RESET events can interrupt a level 3 program.

Table 14. Setting the interrupt software priority

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low ↓ High	1	0
Level 1		0	1
Level 2			0
Level 3 (= interrupt disable*)		1	1

7.5.2 Interrupt software priority registers (ISPRx)

All ISPRx register bits are read/write except bit 7:4 of **ISPR3** which are read only.

Reset value: 1111 1111 (FFh)

	7							0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
ISPR3	1	1	1	1	1	1	I1_12	I0_12

ISPRx registers contain the interrupt software priority of each interrupt vector. Each interrupt vector (except RESET and TRAP) has corresponding bits in these registers to define its software priority. This correspondence is shown in *Table 15*.

Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.

Figure 35. Single timer mode (ENCNTR2=0)

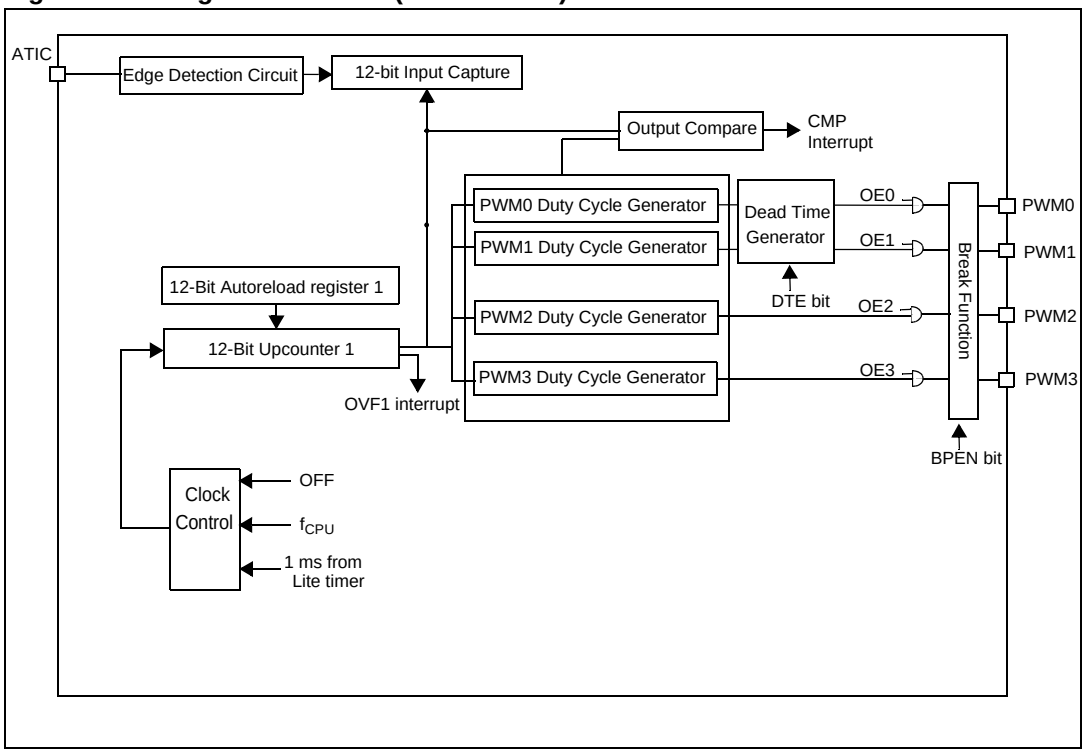
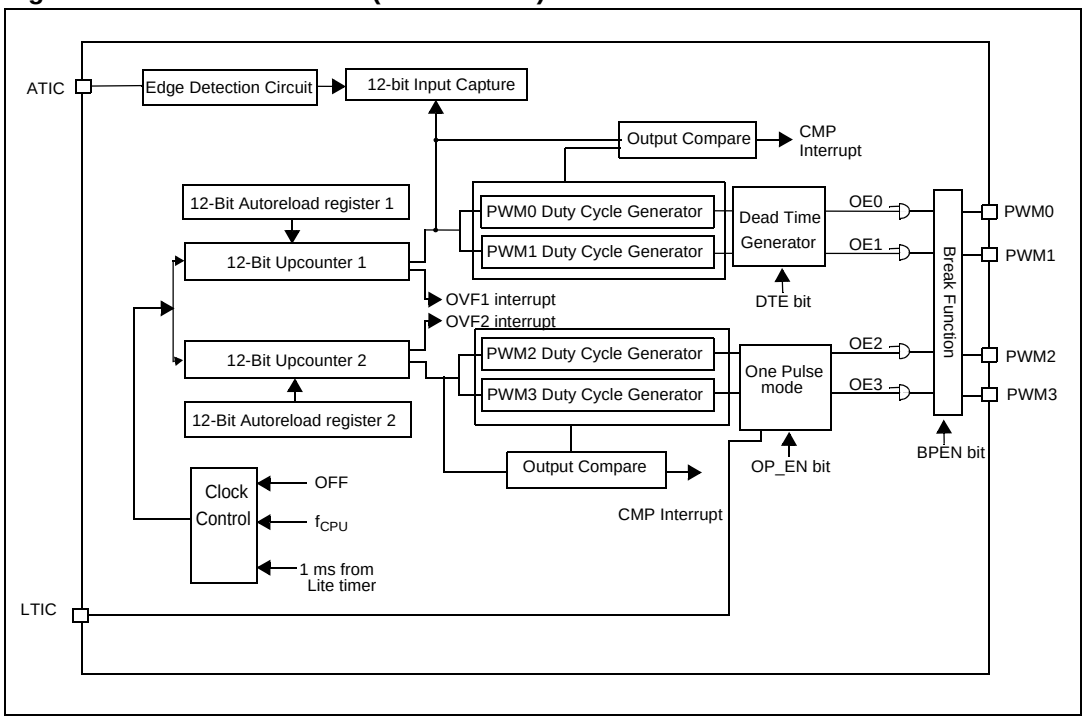


Figure 36. Dual timer mode (ENCNTR2=1)



10.2.3 Functional description

PWM mode

This mode allows up to four Pulse Width Modulated signals to be generated on the PWMx output pins.

- **PWM frequency**

  The four PWM signals can have the same frequency (f_{PWM}) or can have two different frequencies. This is selected by the ENCNTR2 bit which enables Single Timer or Dual Timer mode (see *Figure 35* and *Figure 36*). The frequency is controlled by the counter period and the ATR register value. In Dual Timer mode, PWM2 and PWM3 can be generated with a different frequency controlled by CNTR2 and ATR2.

  $$f_{PWM} = f_{COUNTER}/(4096 - ATR)$$

  Following the above formula, if $f_{COUNTER}$ equals 4 MHz, the maximum value of f_{PWM} is 2 MHz (ATR register value = 4094), and the minimum value is 1 kHz (ATR register value = 0).

  The maximum value of ATR is 4094 because it must be lower than the DC4R value which must be 4095 in this case.

- **Duty cycle**

  The duty cycle is selected by programming the DCRx registers. These are preload registers. The DCRx values are transferred in Active duty cycle registers after an overflow event if the corresponding transfer bit (TRANx bit) is set.

  The TRAN1 bit controls the PWMx outputs driven by counter 1 and the TRAN2 bit controls the PWMx outputs driven by counter 2.

  PWM generation and output compare are done by comparing these active DCRx values with the counter.

  The maximum available resolution for the PWMx duty cycle is:

  $$Resolution = 1/(4096 - ATR)$$

  where ATR is equal to 0. With this maximum resolution, 0% and 100% duty cycle can be obtained by changing the polarity.

  At reset, the counter starts counting from 0.

  When a upcounter overflow occurs (OVF event), the preloaded Duty cycle values are transferred to the active Duty Cycle registers and the PWMx signals are set to a high level. When the upcounter matches the active DCRx value the PWMx signals are set to a low level. To obtain a signal on a PWMx pin, the contents of the corresponding active DCRx register must be greater than the contents of the ATR register.

  The maximum value of ATR is 4094 because it must be lower than the DCR value which must be 4095 in this case.

- **Polarity inversion**

  The polarity bits can be used to invert any of the four output signals. The inversion is synchronized with the counter overflow if the corresponding transfer bit in the ATCSR2 register is set (reset value). See *Figure 37*.

Timer Control register 2 (ATCSR2)

Reset value: 0000 0011 (03h)

7							0
FORCE2	FORCE1	ICS	OVFIE2	OVF2	ENCNTR2	TRAN2	TRAN1
Read/write							

Bit 7 = **FORCE2** *Force Counter 2 Overflow bit*

This bit is read/set by software. When set, it loads FFFh in the CNTR2 register. It is reset by hardware one CPU clock cycle after counter 2 overflow has occurred.

0 : No effect on CNTR2

1 : Loads FFFh in CNTR2

Note: This bit must not be reset by software

Bit 6 = **FORCE1** *Force Counter 1 Overflow bit*

This bit is read/set by software. When set, it loads FFFh in CNTR1 register. It is reset by hardware one CPU clock cycle after counter 1 overflow has occurred.

0 : No effect on CNTR1

1 : Loads FFFh in CNTR1

Note: This bit must not be reset by software

Bit 5 = **ICS** *Input Capture Shorted bit*

This bit is read/write by software. It allows the ATtimer CNTR1 to use the LTIC pin for long Input Capture.

0 : ATIC for CNTR1 Input Capture

1 : LTIC for CNTR1 Input Capture

Bit 4 = **OVFIE2** *Overflow interrupt 2 enable bit*

This bit is read/write by software and controls the overflow interrupt of counter2.

0: Overflow interrupt disabled.

1: Overflow interrupt enabled.

Bit 3 = **OVF2** *Overflow flag*

This bit is set by hardware and cleared by software by reading the ATCSR2 register. It indicates the transition of the counter2 from FFFh to ATR2 value.

0: No counter overflow occurred

1: Counter overflow occurred

Bit 2 = **ENCNTR2** *Enable counter2* for PWM2/3

This bit is read/write by software and switches the PWM2/3 operation to the CNTR2 counter. If this bit is set, PWM2/3 will be generated using CNTR2.

0: PWM2/3 is generated using CNTR1.

1: PWM2/3 is generated using CNTR2.

Note: Counter 2 gets frozen when the ENCNTR2 bit is reset. When ENCNTR2 is set again, the counter will restart from the last value.

10.3.5 Interrupts

Table 40. Description of interrupt events

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

The TBxF and ICF interrupt events are connected to separate interrupt vectors (see *Section 7: Interrupts*).

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

10.3.6 Register description

Lite Timer Control/Status register 2 (LTCSR2)

Reset value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	TB2IE	TB2F
Read / Write							

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

Bit 1 = **TB2IE** *Timebase 2 Interrupt enable bit*

This bit is set and cleared by software.

0: Timebase (TB2) interrupt disabled

1: Timebase (TB2) interrupt enabled

Bit 0 = **TB2F** *Timebase 2 Interrupt flag*

This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect.

0: No Counter 2 overflow

1: A Counter 2 overflow has occurred

Counter register (CR)

- Counter high register (CHR) is the most significant byte (MSB).
- Counter low register (CLR) is the least significant byte (LSB).

Alternate counter register (ACR)

- Alternate counter high register (ACHR) is the MSB.
- Alternate counter low register (ACLR) is the LSB.

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*16-bit read sequence (from either the counter register or the alternate counter register) on page 120*).

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 37*. The value in the counter register repeats every 131 072, 262 144 or 524 288 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.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

Figure 57. Counter timing diagram, internal clock divided by 2

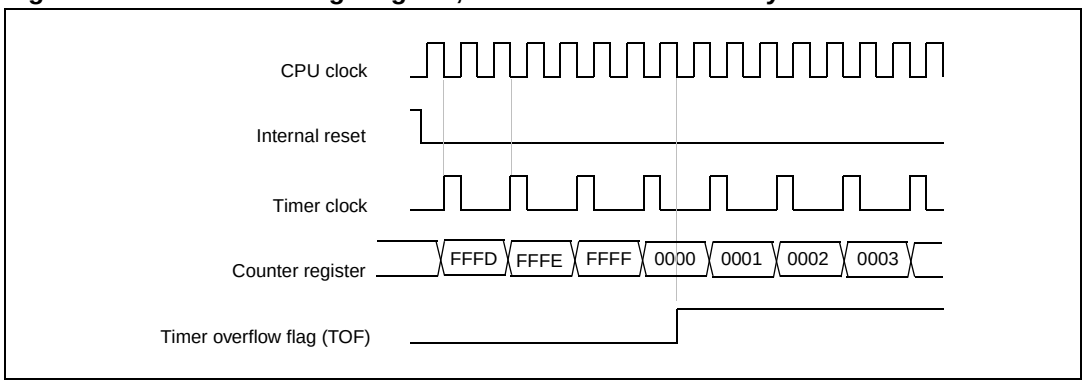


Figure 58. Counter timing diagram, internal clock divided by 4

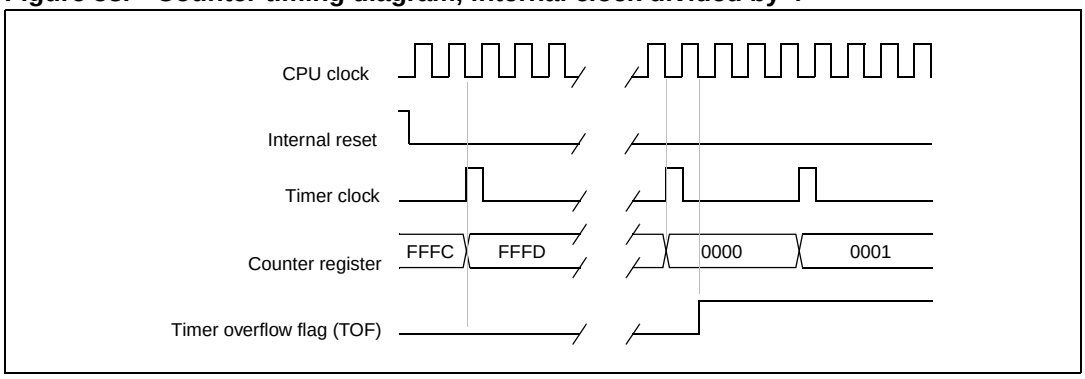
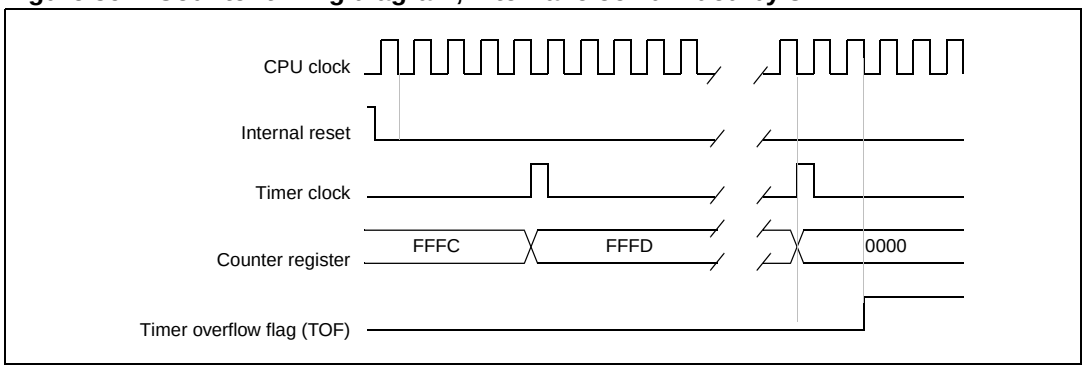


Figure 59. Counter timing diagram, internal clock divided by 8



Note: The device is in reset state when the internal reset signal is high, when it is low the device is running.

Master mode

To switch from default Slave mode to Master mode a Start condition generation is needed.

Start condition

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.

The master then waits for a read of the SR1 register followed by a write in the DR register with the Slave address, **holding the SCL line low** (see *Figure 71* Transfer sequencing EV5).

Slave address transmission

1. The slave address is then sent to the SDA line via the internal shift register.
   - In 7-bit addressing mode, one address byte is sent.
   - In 10-bit addressing mode, sending the first byte including the header sequence causes the following event. The EVF bit is set by hardware with interrupt generation if the ITE bit is set.
2. The master then waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see *Figure 71* Transfer sequencing EV9).
3. Then the second address byte is sent by the interface.
4. 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.
5. 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 71* Transfer sequencing EV6).
6. Next the master must enter Receiver or Transmitter mode.

Note: In 10-bit addressing mode, to switch the master to Receiver mode, software must generate a repeated Start condition and resend the header sequence with the least significant bit set (11110xx1).

Master receiver

Following the address transmission and after 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:

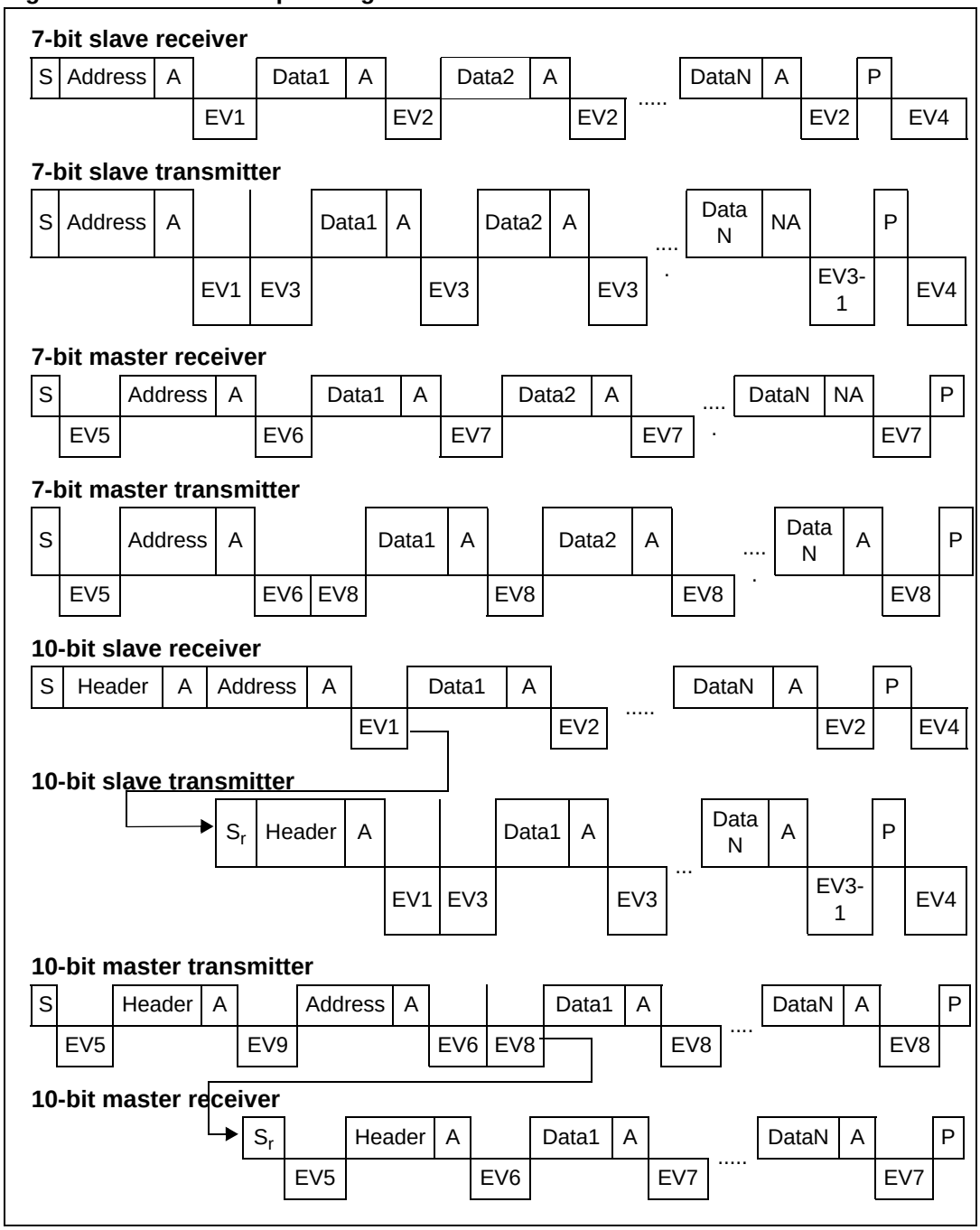
- Acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

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 71* 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.

Figure 71. Transfer sequencing



1. S=Start, S_r = Repeated Start, P=Stop, A=Acknowledge, NA=Non-acknowledge, EVx=Event (with interrupt if ITE=1).
2. **EV1:** EVF=1, ADSL=1, cleared by reading SR1 register.
3. **EV2:** EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.
4. **EV3:** EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.
5. **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). If lines are released by STOP=1, STOP=0, the

Bit 1 = **STOP** *Generation of a Stop condition bit*

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.

Bit 0 = **ITE** *Interrupt Enable bit*

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 72* 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 71*) is detected.

I²C Own Address register (I2COAR1)

Reset value: 0000 0000 (00h)

7							0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
Read / Write							

- In 7-bit addressing mode

  Bits 7:1 = **ADD[7:1]** *Interface address*. These bits define the I²C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

  Bit 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).

Note: Address 01h is always ignored.

- In 10-bit addressing mode

  Bits 7:0 = **ADD[7:0]** *Interface address*. These are the least significant bits of the I²C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

I²C Own Address register (I2COAR2)

Reset value: 0100 0000 (40h)

7							0
FR1	FR0	0	0	0	ADD9	ADD8	0
Read / Write							

Bits 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 I²C specified delays select the value corresponding to the microcontroller frequency f_{CPU}.

Table 48. Configuration of I²C delay times

f_{CPU}	FR1	FR0
< 6 MHz	0	0
6 to 8 MHz	0	1

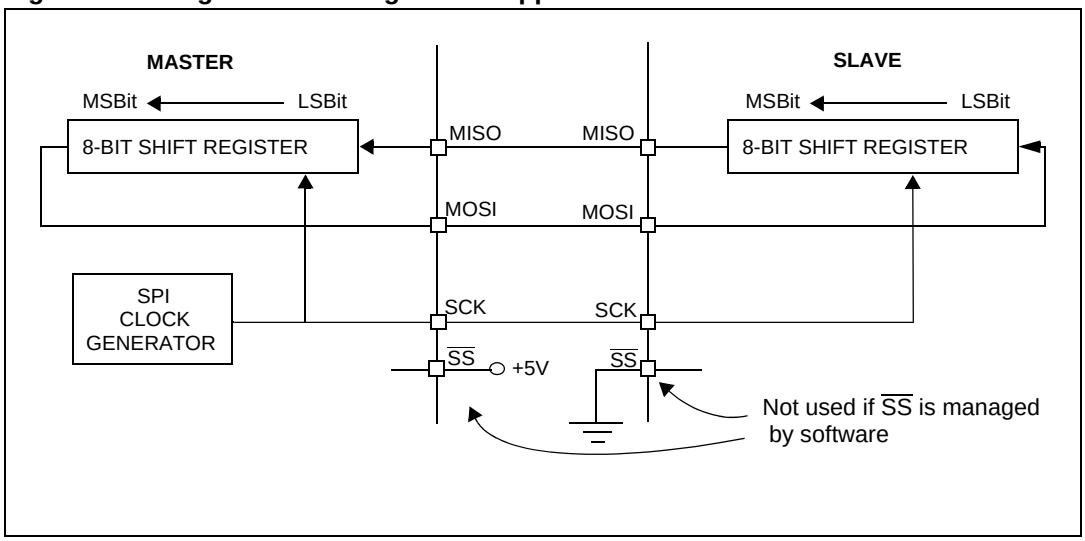
Bits 5:3 = Reserved

Bits 2:1 = **ADD[9:8]** *Interface address*

These are the most significant bits of the I²C bus address of the interface (10-bit mode only). They are not cleared when the interface is disabled (PE=0).

Bit 0 = Reserved.

Figure 74. Single master/ single slave application



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 76*).

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 75*):

If CPHA = 1 (data latched on second 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 SPICSR register)

If CPHA = 0 (data latched on first 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 *Section : Write collision error (WCOL)*).

SPI control/status register (SPICSR)

Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI
Read / Write (some bits Read only)							

Bit 7 = **SPIF** *Serial Peripheral Data Transfer Flag (Read only)*.

This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register).

0: Data transfer is in progress or the flag has been cleared.

1: Data transfer between the device and an external device has been completed.

While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Bit 6 = **WCOL** *Write Collision status (Read only)*.

This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 75).

0: No write collision occurred

1: A write collision has been detected

Bit 5 = **OVR** *SPI Overrun error (Read only)*.

This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section Overrun condition (OVR)). An interrupt is generated if SPIE = 1 in the SPICR register. The OVR bit is cleared by software reading the SPICSR register.

0: No overrun error

1: Overrun error detected

Bit 4 = **MODF** *Mode Fault flag (Read only)*.

This bit is set by hardware when the SS pin is pulled low in master mode (see Section Master mode fault (MODF)). An SPI interrupt can be generated if SPIE = 1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF = 1 followed by a write to the SPICR register).

0: No master mode fault detected

1: A fault in master mode has been detected

Bit 3 = Reserved, must be kept cleared.

Bit 2 = **SOD** *SPI Output Disable*.

This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode)

0: SPI output enabled (if SPE = 1)

1: SPI output disabled

Data register High (ADCDRH)

Reset value: xxxx xxxx (xxh)

7							0
D9	D8	D7	D6	D5	D4	D3	D2
Read only							

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

ADC Control/data register Low (ADCDRL)

Reset value: 0000 00xx (0xh)

7							0
0	0	0	0	SLOW	0	D1	D0
Read/write							

Bits 7:4 = Reserved. Forced by hardware to 0.

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

This bit is set and cleared by software. It is used together with the SPEED bit in the ADCCSR register to configure the ADC clock speed as shown on the table below.

Table 56. Configuring the ADC clock speed

f_{ADC}(1)	SLOW	SPEED
$f_{CPU}/2$	0	0
f_{CPU}	0	1
$f_{CPU}/4$	1	x

1. The maximum allowed value of f_{ADC} is 4 MHz (see *Section 12.11 on page 209*)

Bit 2 = Reserved. Forced by hardware to 0.

Bits 1:0 = **D[1:0]** *LSB of Analog Converted value*

Table 57. ADC register mapping and reset values

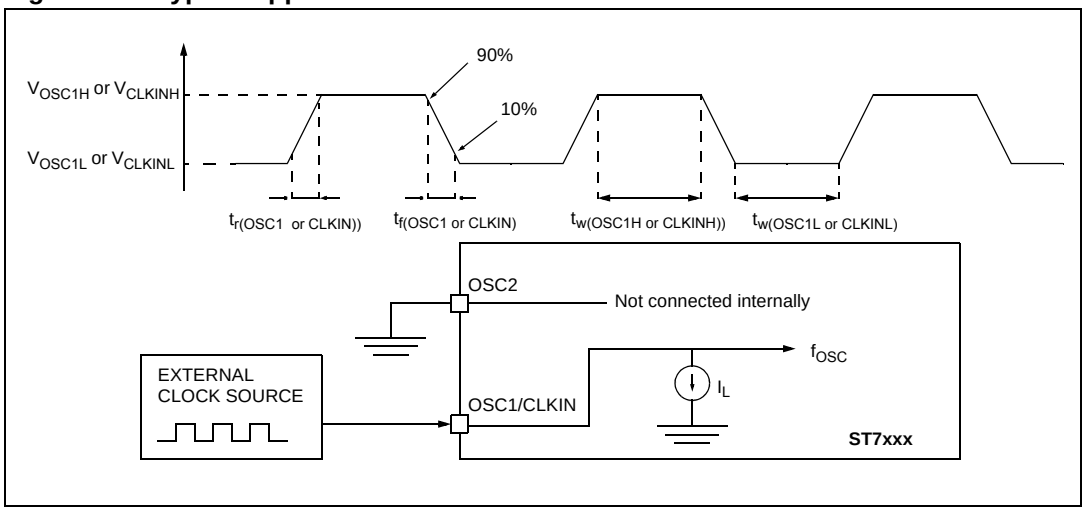
Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0036h	**ADCCSR** Reset Value	EOC 0	SPEED 0	ADON 0	0 0	CH3 0	CH2 0	CH1 0	CH0 0
0037h	**ADCDRH** Reset Value	D9 x	D8 x	D7 x	D6 x	D5 x	D4 x	D3 x	D2 x
0038h	**ADCDRL** Reset Value	0 0	0 0	0 0	0	SLOW 0	0	D1 x	D0 x

Table 78. External clock source characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSC1H} or V_{CLKIN_H}	OSC1/CLKIN input pin high level voltage	see *Figure 86*	$0.7xV_{DD}$		V_{DD}	V
V_{OSC1L} or V_{CLKIN_L}	OSC1/CLKIN input pin low level voltage		V_{SS}		$0.3xV_{DD}$	
$t_{w(OSC1H)}$ or $t_{w(CLKINH)}$ $t_{w(OSC1L)}$ or $t_{w(CLKINL)}$	OSC1/CLKIN high or low time(1)		15			ns
$t_{r(OSC1)}$ or $t_{r(CLKIN)}$ $t_{f(OSC1)}$ or $t_{f(CLKIN)}$	OSC1/CLKIN rise or fall time(1)				15	
I_L	OSCx/CLKIN Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$			±1	µA

1. Data based on design simulation and/or technology characteristics, not tested in production.

Figure 86. Typical application with an external clock source



12.6.1 Auto wakeup from Halt oscillator (AWU)

Table 79. AWU from Halt characteristics

Symbol	Parameter(1)	Conditions	Min	Typ	Max	Unit
f_{AWU}	AWU Oscillator Frequency		16	32	64	kHz
t_{RCSRT}	AWU Oscillator startup time				50	µs

1. Guaranteed by Design. Not tested in production.

13 Device configuration and ordering information

This device is available for production in user programmable version (Flash).

ST7FOX XFlash devices are shipped to customers with a default program memory content (FFh).

13.1 Option bytes

The two option bytes allow the hardware configuration of the microcontroller to be selected. The option bytes can be accessed only in programming mode (for example using a standard ST7 programming tool).

13.1.1 Option byte 1

Bits 7:6 = **CKSEL[1:0]** *Start-up clock selection.*

These bits are used to select the startup frequency. By default, the internal RC is selected.

Table 92. Startup clock selection

Configuration	CKSEL1	CKSEL0
Internal RC as Startup Clock	0	0
AWU RC as a Startup Clock	0	1
External crystal/ceramic resonator	1	0
External Clock	1	1

Bits 5:3 = Reserved, must always be 1.

Bit 2 = **LVD** *Low Voltage Detection selection.*

This option bit enables the low voltage detection block (LVD).

0: LVD on
1: LVD off (default value)

Bit 1 = **WDG SW** *Hardware or software watchdog*

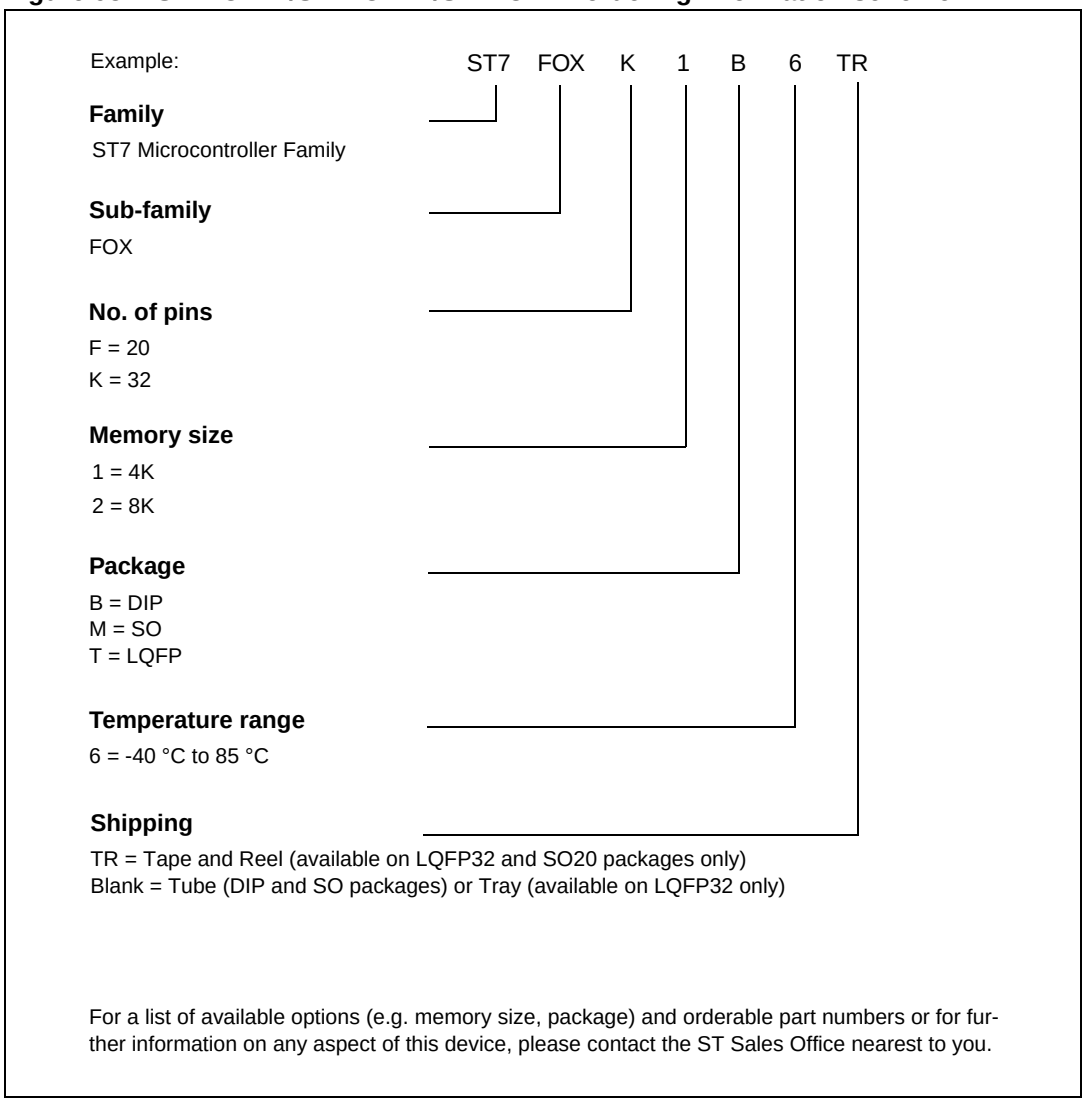
This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

Bit 0 = **WDG HALT** *Watchdog Reset on Halt*

This option bit determines if a Reset is generated when entering Halt mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

13.2 Device ordering information

Figure 93. ST7FOXF1/ST7FOXK1/ST7FOXK2 ordering information scheme



ST7FOX failure analysis service

For ST7FOX family devices, STMicroelectronics agrees to accept return of defective parts subject to the FAR (Failure Analysis Report) procedure only if the customer reject rate exceeds 0.35 % for each delivered batch.

A batch is identified with a single trace code located on the top side marking.

Figure 95. 20-pin plastic dual in-line package, 300-mil width, package outline

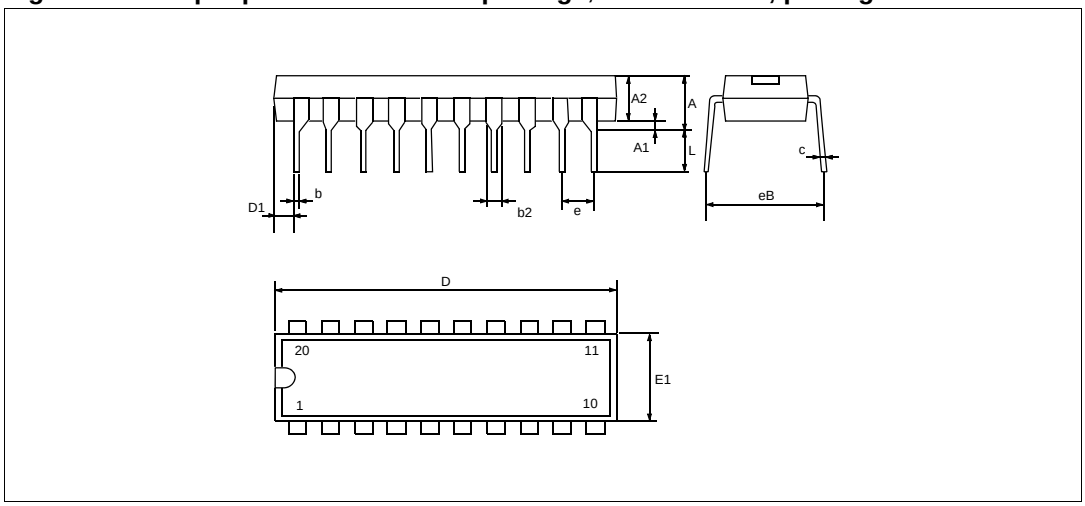


Table 98. 20-pin plastic dual in-line package, 300-mil width, mechanical data

Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A			5.33			0.2098
A1	0.38			0.0150		
A2	2.92	3.30	4.95	0.1150	0.1299	0.1949
b	0.36	0.46	0.56	0.0142	0.0181	0.0220
b2	1.14	1.52	1.78	0.0449	0.0598	0.0701
c	0.20	0.25	0.36	0.0079	0.0098	0.0142
D	24.89	26.16	26.92	0.9799	1.0299	1.0598
D1	0.13			0.0051		
e		2.54			0.1000	
eB			10.92			0.4299
E1	6.10	6.35	7.11	0.2402	0.2500	0.2799
L	2.92	3.30	3.81	0.1150	0.1299	0.1500
	Number of Pins					
N	20					

1. Values in inches are converted from mm and rounded to 4 decimal digits.