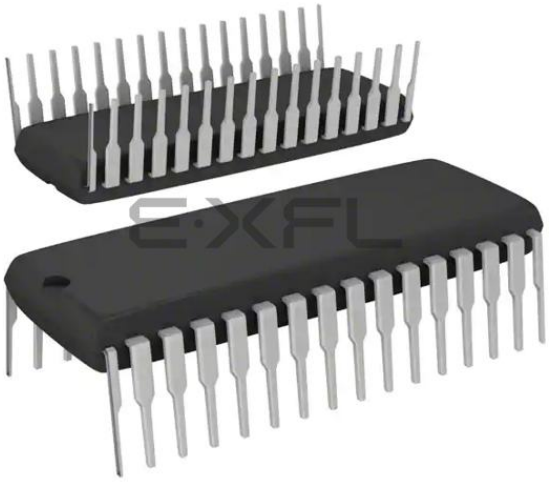


Welcome to E-XFL.COM

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	Through Hole
Package / Case	32-SDIP (0.400", 10.16mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7foxk2b6

Table 4. Hardware register map(1) (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0055h	16-bit Timer(4)	TACR2	Timer A Control register 2	00h	R/W
0056h		TACR1	Timer A Control register 1	00h	R/W
0057h		TACSR	Timer A Control/status register	00h	Read Only
0058h		TAICHR1	Timer A Input capture 1 high register	xxh	Read Only
0059h		TAICLR1	Timer A Input capture 1 low register	xxh	Read Only
005Ah		TAOCHR1	Timer A Output compare 1 high register	80h	R/W
005Bh		TAOCLR1	Timer A Output compare 1 low register	00h	R/W
005Ch		TACHR	Timer A Output counter high register	FFh	Read Only
005Dh		TACLR	Timer A Output counter low register	FCh	Read Only
005Eh		TAACHR	Timer A Alternate counter high register	FFh	Read Only
005Fh		TAACLR	Timer A Alternate counter low register	FCh	Read Only
0060h		TAICHR2	Timer A Input capture 2 high register	xxh	Read Only
0061h		TAICLR2	Timer A Input capture 2 low register	xxh	Read Only
0062h		TAOCHR2	Timer A Output compare 2 high register	80h	R/W
0063h		TAOCLR2	Timer A Output compare 2 low register	00h	R/W
0064h	I2C	I2CCR	I^2C Control register	00h	R/W
0065h		I2CSR1	I^2C Status register 1	00h	Read only
0066h		I2CSR2	I^2C Status register 2	00h	Read only
0067h		I2CCCR	I^2C Clock Control register	00h	R/W
0068h		I2COAR1	I^2C Own Address register 1	00h	R/W
0069h		I2COAR2	I^2C Own Address register 2	40h	R/W
006Ah		I2CDR	I^2C Data register	00h	R/W
0070h	SPI(4)	SPIDR	SPI Data register	0xh	R/W
0071h		SPICR	SPI Control register	00h	R/W
0072h		SPISR	SPI Status register	xxh	R/W

1. Legend: x=undefined, R/W=read/write.
2. Reset status is 03h for ST7FOXK2 and 00h for ST7FOXF1 and ST7FOXK1
3. For a description of the Debug Module registers, see ICC protocol reference manual.
4. Available on ST7FOXK2 only

4 Flash programmable memory

4.1 Introduction

The ST7 single voltage extended Flash (XFlash) is a non-volatile memory that can be electrically erased and programmed either on a byte-by-byte basis or up to 32 bytes in parallel.

The XFlash devices can be programmed off-board (plugged in a programming tool) or on-board using In-Circuit Programming or In-Application Programming.

The array matrix organization allows each sector to be erased and reprogrammed without affecting other sectors.

4.2 Main features

- ICP (In-Circuit Programming)
- IAP (In-Application Programming)
- ICT (In-Circuit Testing) for downloading and executing user application test patterns in RAM
- Sector 0 size configurable by option byte
- Read-out and write protection

4.3 Programming modes

The ST7 can be programmed in three different ways:

- Insertion in a programming tool. In this mode, Flash sectors 0 and 1, option byte row can be programmed or erased.
- In-Circuit Programming. In this mode, Flash sectors 0 and 1, option byte row can be programmed or erased without removing the device from the application board.
- In-Application Programming. In this mode, sector 1 can be programmed or erased without removing the device from the application board and while the application is running.

4.3.1 In-Circuit Programming (ICP)

ICP uses a protocol called ICC (In-Circuit Communication) which allows an ST7 plugged on a printed circuit board (PCB) to communicate with an external programming device connected via cable. ICP is performed in three steps:

Switch the ST7 to ICC mode (In-Circuit Communications). This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the $\overline{\text{RESET}}$ pin is pulled low. When the ST7 enters ICC mode, it fetches a specific Reset vector which points to the ST7 System Memory containing the ICC protocol routine. This routine enables the ST7 to receive bytes from the ICC interface.

- Download ICP Driver code in RAM from the ICCDATA pin
- Execute ICP Driver code in RAM to program the Flash memory

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 and to manipulate data.

5.3.2 Index registers (X and Y)

In indexed addressing modes, these 8-bit registers are used to create either effective addresses or 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 (not pushed to and popped from the stack).

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 register (CC)

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

Reset value: 111x 1xxx

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

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

Arithmetic management bits

Bit 4 = **H** *Half carry bit*

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. 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.

6.4 System Integrity management (SI)

The System Integrity Management block contains the Low voltage Detector (LVD).

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Section 11.2.1 on page 184 for further details.

6.4.1 Low Voltage Detector (LVD)

The Low Voltage Detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a $V_{IT-(LVD)}$ reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The $V_{IT-(LVD)}$ reference value for a voltage drop is lower than the $V_{IT+(LVD)}$ reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD Reset circuitry generates a reset when V_{DD} is below:

- $V_{IT+(LVD)}$ when V_{DD} is rising
- $V_{IT-(LVD)}$ when V_{DD} is falling

The LVD function is illustrated in *Figure 16*.

The voltage threshold can be enabled/disabled by option byte. See *Section 13.1 on page 211*.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above $V_{IT-(LVD)}$, the MCU can only be in two modes:

- Under full software control
- In static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

During a Low Voltage Detector Reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

Note: Use of LVD with capacitive power supply: with this type of power supply, if power cuts occur in the application, it is recommended to pull V_{DD} down to 0 V to ensure optimum restart conditions. Refer to circuit example in Figure 89 on page 207 and note 4.

The LVD is an optional function which can be selected by option byte. See Section 13.1 on page 211.

It allows the device to be used without any external RESET circuitry.

If the LVD is disabled, an external circuitry must be used to ensure a proper power-on reset.

It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from Reset, to ensure the application functions properly.

Caution: If an LVD reset occurs after a watchdog reset has occurred, the LVD will take priority and will clear the watchdog flag.

7.5.3 External Interrupt Control register (EICR)

Reset value: 0000 0000 (00h)

7							0
0	0	IS21	IS20	IS11	IS10	IS01	IS00
Read/write							

Bits 7:6 = Reserved, must be kept cleared.

Bits 5:4 = **IS2[1:0]** *ei2 sensitivity bits*

These bits define the interrupt sensitivity for ei2 (Port C) according to *Table 19*.

Bits 3:2 = **IS1[1:0]** *ei1 sensitivity bits*

These bits define the interrupt sensitivity for ei1 (Port B) according to *Table 19*.

Bits 1:0 = **IS0[1:0]** *ei0 sensitivity bits*

These bits define the interrupt sensitivity for ei0 (Port A) according to *Table 19*.

Note: 1 These 8 bits can be written only when the I bit in the CC register is set.

2 Changing the sensitivity of a particular external interrupt clears this pending interrupt. This can be used to clear unwanted pending interrupts. Refer to Section : External interrupt function.

Table 19. Interrupt sensitivity bits

ISx1	ISx0	External interrupt sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

8.4.1 Active-halt mode

Active-Halt mode is the lowest power consumption mode of the MCU with a real-time clock available. It is entered by executing the 'HALT' instruction when active halt mode is enabled.

The MCU can exit Active-Halt mode on reception of a Lite timer/ AT timer interrupt or a Reset.

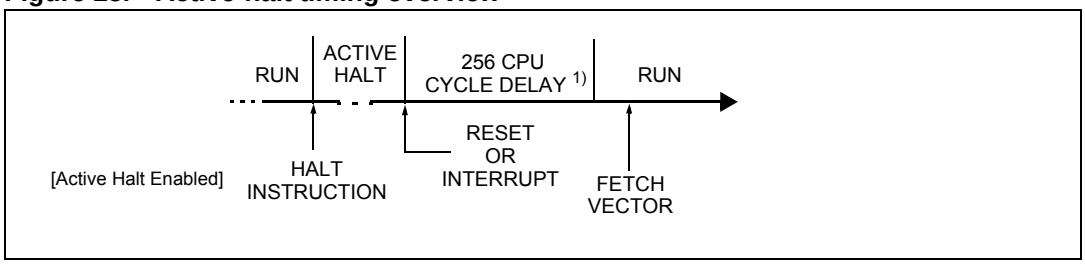
- When exiting Active-Halt mode by means of a Reset, a 256 CPU cycle delay occurs. After the start up delay, the CPU resumes operation by fetching the Reset vector which woke it up (see *Figure 26*).
- When exiting Active-Halt mode by means of an interrupt, the CPU immediately resumes operation by servicing the interrupt vector which woke it up (see *Figure 26*).

When entering Active-Halt mode, the I bit in the CC register is cleared to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In Active-Halt mode, only the main oscillator and the selected timer counter (LT/AT) are running to keep a wakeup time base. All other peripherals are not clocked except those which get their clock supply from another clock generator (such as external or auxiliary oscillator).

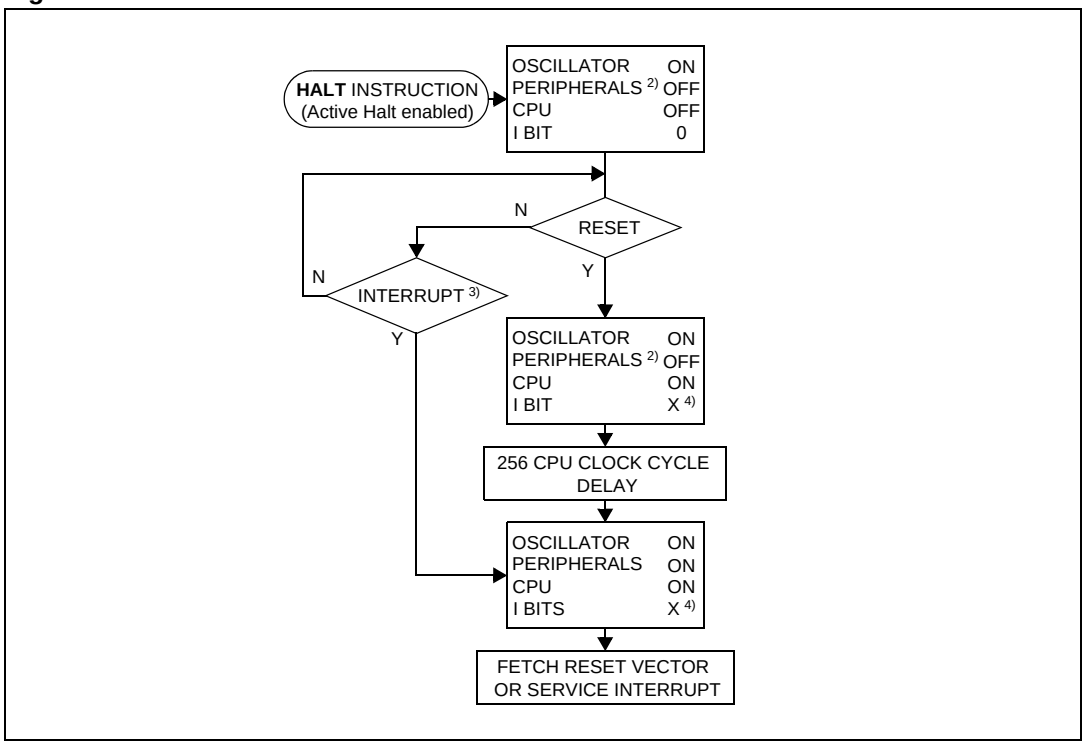
Caution: As soon as Active-Halt is enabled, executing a HALT instruction while the Watchdog is active does not generate a Reset if the WDGHALT bit is reset.
This means that the device cannot spend more than a defined delay in this power saving mode.

Figure 25. Active-halt timing overview



1. This delay occurs only if the MCU exits Active-Halt mode by means of a RESET.

Figure 26. Active-halt mode flowchart



1. This delay occurs only if the MCU exits Active-Halt mode by means of a RESET.
2. Peripherals clocked with an external clock source can still be active.
3. Only the Lite timer RTC and AT timer interrupts can exit the MCU from Active-Halt mode.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I bit of the CC register is set during the interrupt routine and cleared when the CC register is popped.

8.4.2 Halt mode

The Halt mode is the lowest power consumption mode of the MCU. It is entered by executing the HALT instruction when active halt mode is disabled.

The MCU can exit Halt mode on reception of either a specific interrupt (see *Table 17: ST7FOXF1/ST7FOXK1 Interrupt mapping*) or a Reset. When exiting Halt mode by means of a Reset or an interrupt, the main oscillator is immediately turned on and the 256 CPU cycle delay is used to stabilize it. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the Reset vector which woke it up (see *Figure 28*).

When entering Halt mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes immediately.

In Halt mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

The compatibility of Watchdog operation with Halt mode is configured by the “WDGHALT” option bit of the option byte. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog Reset (see *Section 13.1: Option bytes* for more details).

10.2 Dual 12-bit autoreload timer

10.2.1 Introduction

The 12-bit Autoreload timer can be used for general-purpose timing functions. It is based on one or two free-running 12-bit upcounters with an Input Capture register and four PWM output channels. There are 7 external pins:

- Four PWM outputs
- ATIC/LTIC pins for the Input Capture function
- BREAK pins for forcing a break condition on the PWM outputs

10.2.2 Main features

- Single Timer or Dual Timer mode with two 12-bit upcounters (CNTR1/CNTR2) and two 12-bit autoreload registers (ATR1/ATR2)
- Maskable overflow interrupts
- PWM mode
  - Generation of four independent PWMx signals
  - Dead time generation for Half bridge driving mode with programmable dead time
  - Frequency 2 kHz - 4 MHz (@ 8 MHz f_{CPU})
  - Programmable duty-cycles
  - Polarity control
  - Programmable output modes
- Output Compare mode
- Input Capture mode
  - 12-bit Input Capture register (ATICR)
  - Triggered by rising and falling edges
  - Maskable IC interrupt
  - Long range Input Capture
- Internal/External Break control
- Flexible Clock control
- One Pulse mode on PWM2/3
- Force update

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*.

Bit 0 = **CMPIE** *Compare Interrupt Enable bit*

This bit is read/write by software and cleared by hardware after a reset. it can be used to mask the interrupt generated when any of the cmpfx bit is set.

0: Output Compare Interrupt Disabled.

1: Output Compare Interrupt Enabled.

Counter register 1 High (CNTR1H)

Reset value: 0000 0000 (00h)

15							8
0	0	0	0	CNTR1_11	CNTR1_10	CNTR1_9	CNTR1_8
Read only							

Counter register 1 Low (CNTR1L)

Reset value: 0000 0000 (00h)

7							0
CNTR1_7	CNTR1_6	CNTR1_5	CNTR1_4	CNTR1_3	CNTR1_2	CNTR1_1	CNTR1_0
Read only							

Bits 15:12 = Reserved

Bits 11:0 = **CNTR1[11:0]** *Counter value*

This 12-bit register is read by software and cleared by hardware after a reset. The counter CNTR1 increments continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations. As there is no latch, it is recommended to read LSB first. In this case, CNTR1H can be incremented between the two read operations and to have an accurate result when $f_{timer}=f_{CPU}$, special care must be taken when CNTR1L values close to FFh are read.

When a counter overflow occurs, the counter restarts from the value specified in the ATR1 register.

Input Capture register Low (ATICRL)

Reset value: 0000 0000 (00h)

7							0
ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0
Read only							

Bits 11:0 = **ICR[11:0]** *Input Capture Data*.

This is a 12-bit register which is readable by software and cleared by hardware after a reset. The ATICR register contains captured the value of the 12-bit CNTR1 register when a rising or falling edge occurs on the ATIC or LTIC pin (depending on ICS). Capture will only be performed when the ICF flag is cleared.

Break Enable register (BREAKEN)

Reset value: 0000 0011 (03h)

7							0
0	0	0	0	0	0	BREN2	BREN1
Read/write							

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

Bit 1 = **BREN2** *Break Enable for Counter 2 bit*

This bit is read/write by software. It enables the break functionality for Counter2 if BA bit is set in BREAKCR. It controls PWM2/3 if ENCNTR2 bit is set.

0: No Break applied for CNTR2

1: Break applied for CNTR2

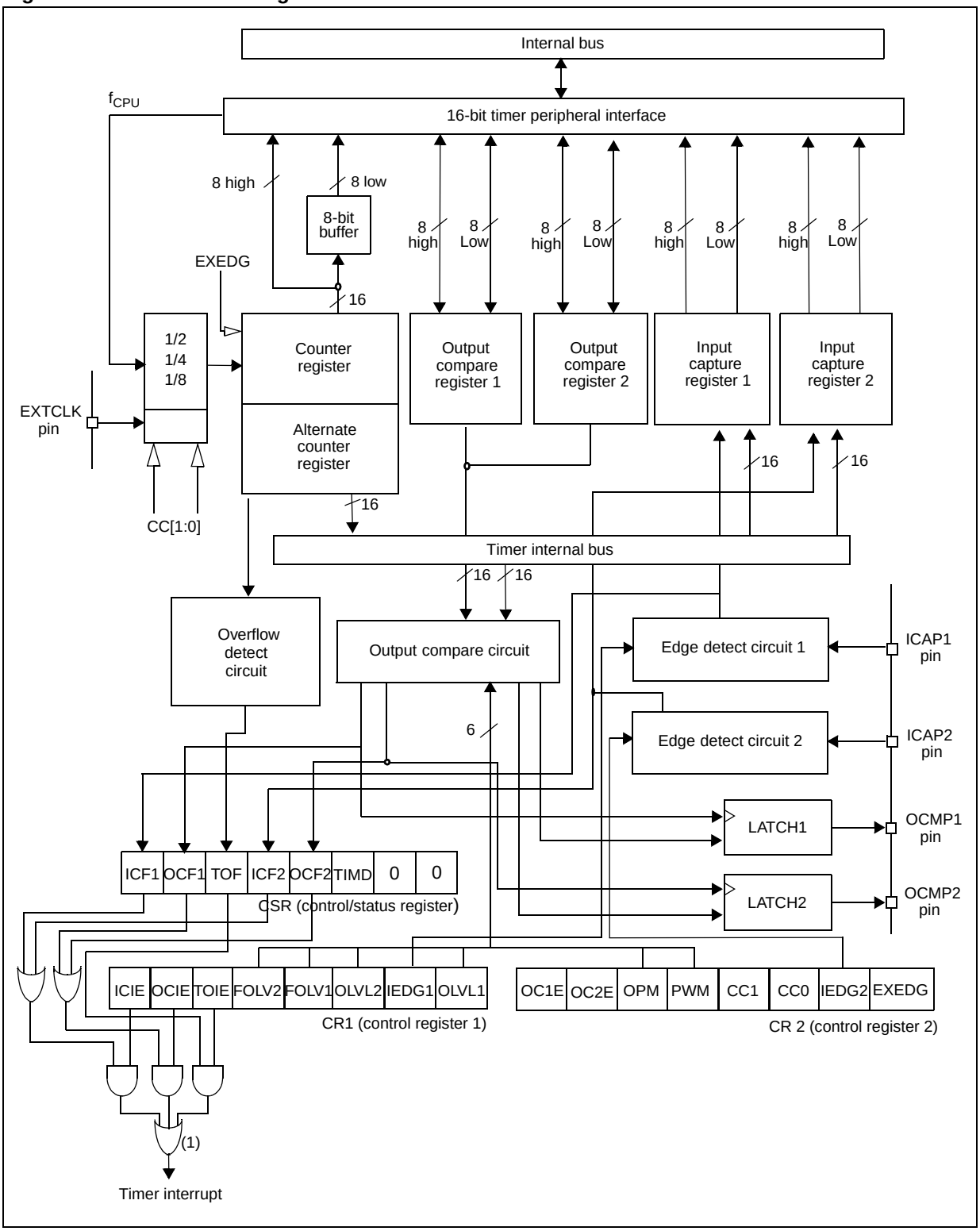
Bit 0 = **BREN1** *Break Enable for Counter 1 bit*

This bit is read/write by software. It enables the break functionality for Counter1. If BA bit is set, it controls PWM0/1 by default, and controls PWM2/3 also if ENCNTR2 bit is reset.

0: No Break applied for CNTR1

1: Break applied for CNTR1

Figure 55. Timer block diagram



1. If IC, OC and TO interrupt requests have separate vectors then the last OR is not present (see *Table 17: ST7FOXF1/ST7FOXK1 Interrupt mapping*)

the user toggle 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 in order to measure event that go beyond the timer range (FFFFh).

Figure 60. Input capture block diagram

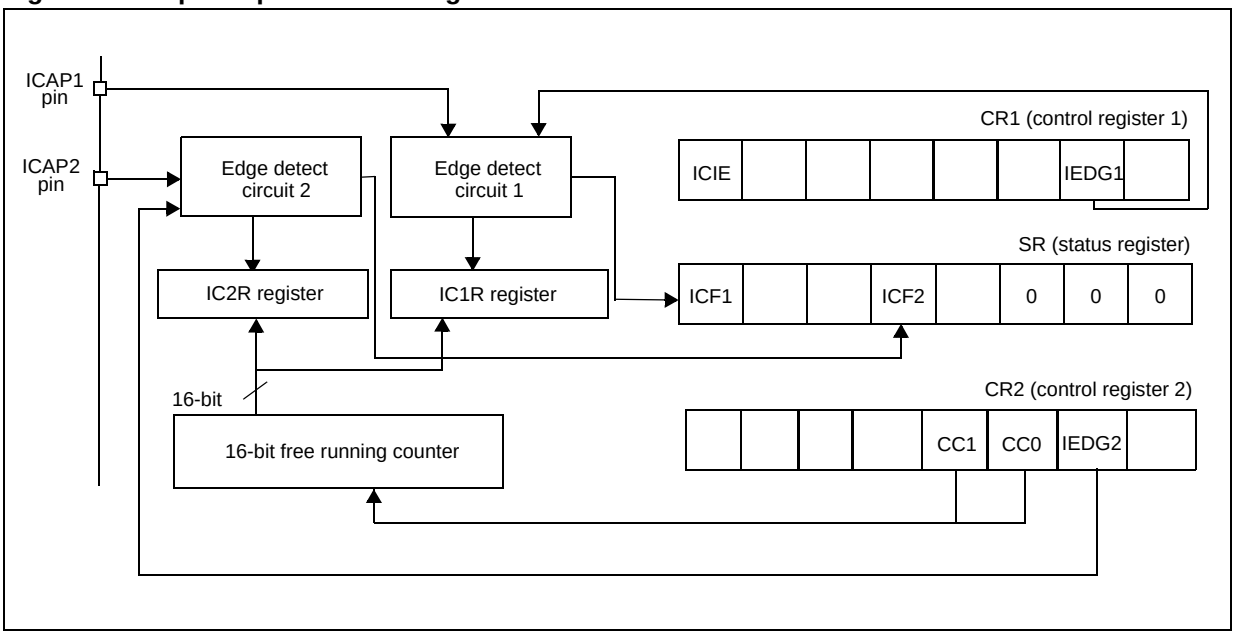
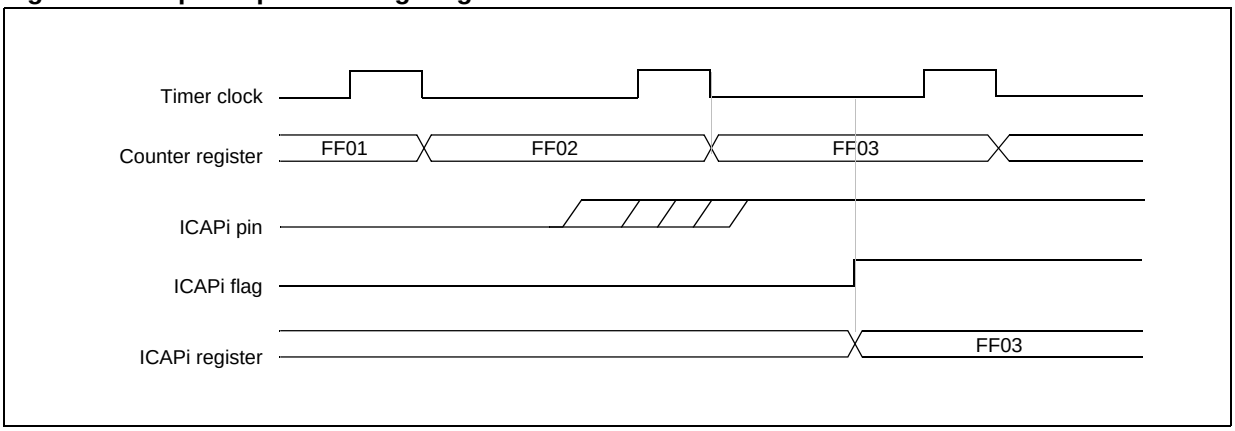


Figure 61. Input capture timing diagram



1. The active edge is the rising edge.
2. The time between an event on the ICAPi pin and the appearance of the corresponding flag is from 2 to 3 CPU clock cycles. This depends on the moment when the ICAP event happens relative to the timer clock.

Bit 4 = **ICF2** *Input capture flag 2*

0: No input capture (reset value)

1: An input capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 = **OCF2** *Output compare flag 2*

0: No match (reset value)

1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2 = **TIMD** *Timer disable*

This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disables the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed while it is disabled.

0: Timer enabled

1: Timer prescaler, counter and outputs disabled

Bits 1:0 = Reserved, must be kept cleared

Timer A Input capture 1 high register (TAIC1HR)

Reset value: undefined

This is an 8-bit read-only register that contains the high part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB
Read Only							

Timer A Input capture 1 low register (TAIC1LR)

Reset value: undefined

This is an 8-bit read-only register that contains the low part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB
Read Only							

Table 49. I^2C register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0064h	**I2CCR** Reset Value	0	0	PE 0	ENGC 0	START 0	ACK 0	STOP 0	ITE 0
0065h	**I2CSR1** Reset Value	EVF 0	ADD10 0	TRA 0	BUSY 0	BTF 0	ADSL 0	M/SL 0	SB 0
0066h	**I2CSR2** Reset Value	0	0	0	AF 0	STOPF 0	ARLO 0	BERR 0	GCAL 0
0067h	**I2CCCR** Reset Value	FM/SM 0	CC6 0	CC5 0	CC4 0	CC3 0	CC2 0	CC1 0	CC0 0
0068h	**I2COAR1** Reset Value	ADD7 0	ADD6 0	ADD5 0	ADD4 0	ADD3 0	ADD2 0	ADD1 0	ADD0 0
0069h	**I2COAR2** Reset Value	FR1 0	FR0 1	0	0	0	ADD9 0	ADD8 0	0
006Ah	**I2CDR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0

Figure 75. Generic $\overline{SS}$ timing diagram

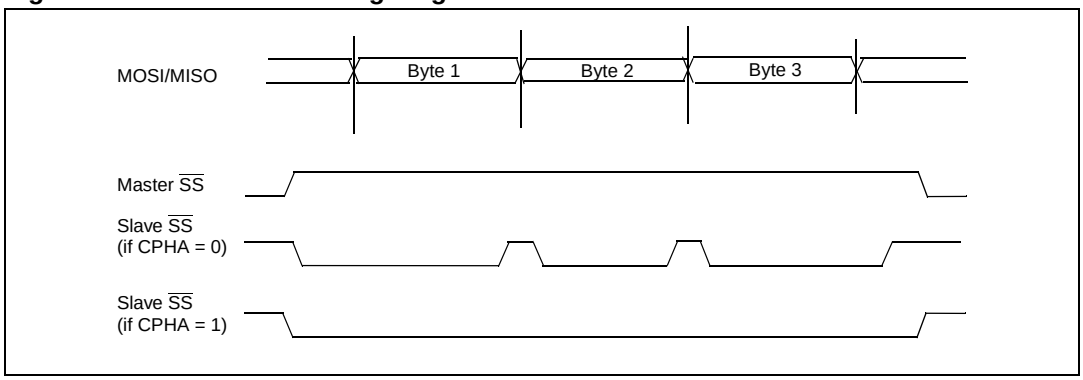
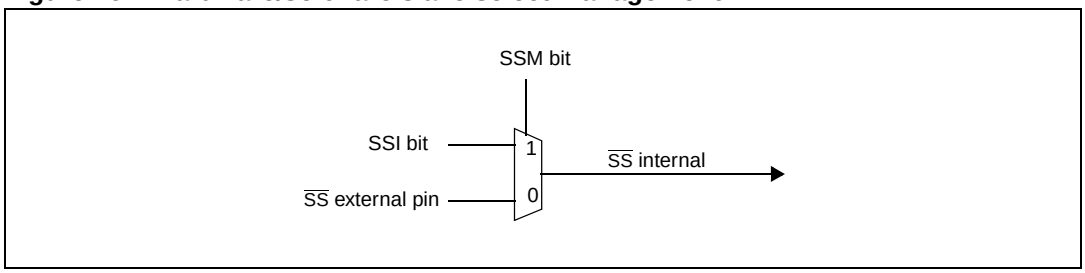


Figure 76. Hardware/software slave select management



Master mode operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

How to operate the SPI in master mode

To operate the SPI in master mode, perform the following steps in order:

1. Write to the SPICR register:
   - Select the clock frequency by configuring the SPR[2:0] bits.
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. *Figure 77* shows the four possible configurations.

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

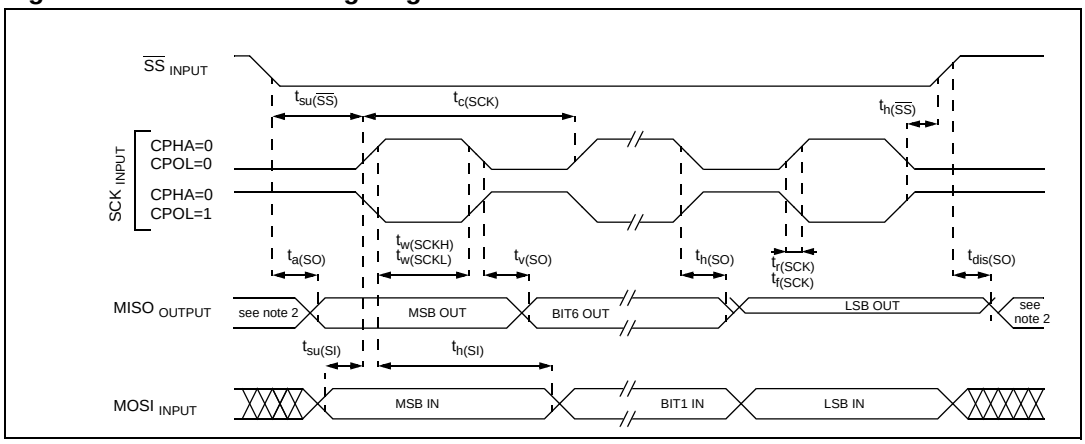
2. Write to the SPICSR register:
   - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the $\overline{SS}$ pin high for the complete byte transmit sequence.
3. Write to the SPICR register:
   - Set the MSTR and SPE bits

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

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

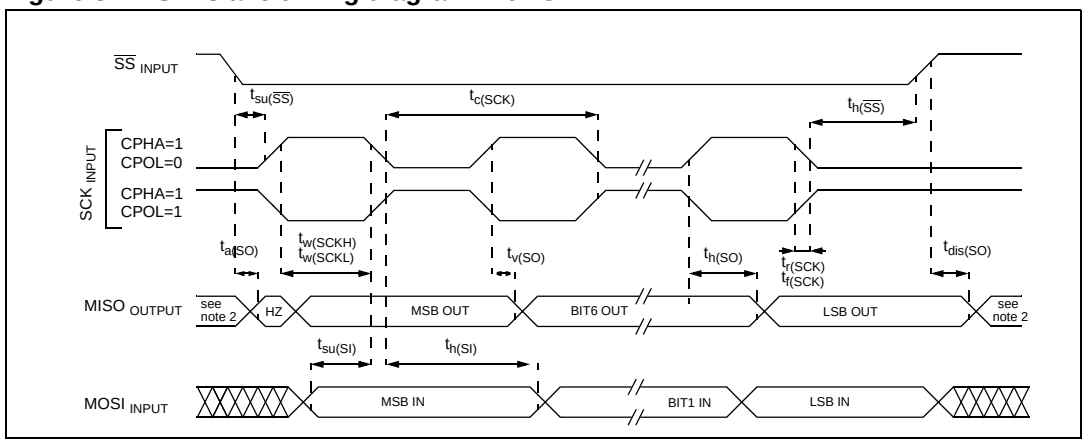
The transmit sequence begins when software writes a byte in the SPIDR register.

Figure 83. SPI slave timing diagram with CPHA=0



1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

Figure 84. SPI slave timing diagram with CPHA=1



1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

12.9.2 Output driving current

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

Table 88. Output driving current characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} (1)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time	V_{DD} = 5 V	I_{IO}=+5 mA, $T_A \leq$ 85°C		1.0	V
			I_{IO}=+2mA, $T_A \leq$ 85°C		0.4	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time		I_{IO}=+20mA,$T_A \leq$ 85°C		1.3	
			I_{IO}=+8mA$T_A \leq$ 85°C		0.75	
V_{OH} (2)	Output high level voltage for an I/O pin when 4 pins are sourced at same time		I_{IO}=-5mA,$T_A \leq$ 85°C	V_{DD}-1.5		
			I_{IO}=-2mA$T_A \leq$ 85°C	V_{DD}-0.8		

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section Table 67.* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
2. The I_{IO} current sourced must always respect the absolute maximum rating specified in *Section Table 67.* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}.

Table 96. ST7 application notes (continued)

Identification	Description
AN1103	Improved B-EMF detection for low speed, low voltage with ST72141
AN1150	Benchmark ST72 vs PC16
AN1151	Performance comparison between ST72254 & PC16F876
AN1278	LIN (Local Interconnect Network) solutions
Product migration	
AN1131	Migrating applications from ST72511/311/214/124 to ST72521/321/324
AN1322	Migrating an application from ST7263 Rev.B to ST7263B
AN1365	Guidelines for migrating ST72C254 applications to ST72F264
AN1604	How to use ST7MDT1-TRAIN with ST72F264
AN2200	Guidelines for migrating ST7LITE1x applications to ST7FLITE1xB
Product optimization	
AN 982	Using ST7 with ceramic resonator
AN1014	How to minimize the ST7 power consumption
AN1015	Software techniques for improving microcontroller EMC performance
AN1040	Monitoring the Vbus signal for USB self-powered devices
AN1070	ST7 checksum self-checking capability
AN1181	Electrostatic discharge sensitive measurement
AN1324	Calibrating the RC oscillator of the ST7FLITE0 MCU using the mains
AN1502	Emulated data EEPROM with ST7 HD Flash memory
AN1529	Extending the current & voltage capability on the ST7265 V_{DDF} supply
AN1530	Accurate timebase for low-cost ST7 applications with internal RC oscillator
AN1605	Using an active RC to wake up the ST7LITE0 from power saving mode
AN1636	Understanding and minimizing ADC conversion errors
AN1828	PIR (passive infrared) detector using the ST7FLITE05/09/SUPERLITE
AN1946	Sensorless BLDC motor control and BEMF sampling methods with ST7MC
AN1953	PFC for ST7MC starter kit
AN1971	ST7LITE0 microcontrolled ballast
Programming and tools	
AN 978	ST7 Visual DeVELOP software key debugging features
AN 983	Key features of the Cosmic ST7 C-compiler package
AN 985	Executing code In ST7 RAM
AN 986	Using the indirect addressing mode with ST7
AN 987	ST7 serial test controller programming
AN 988	Starting with ST7 assembly tool chain

Figure 96. 32-pin plastic dual in-line package, shrink 400-mil width, package outline

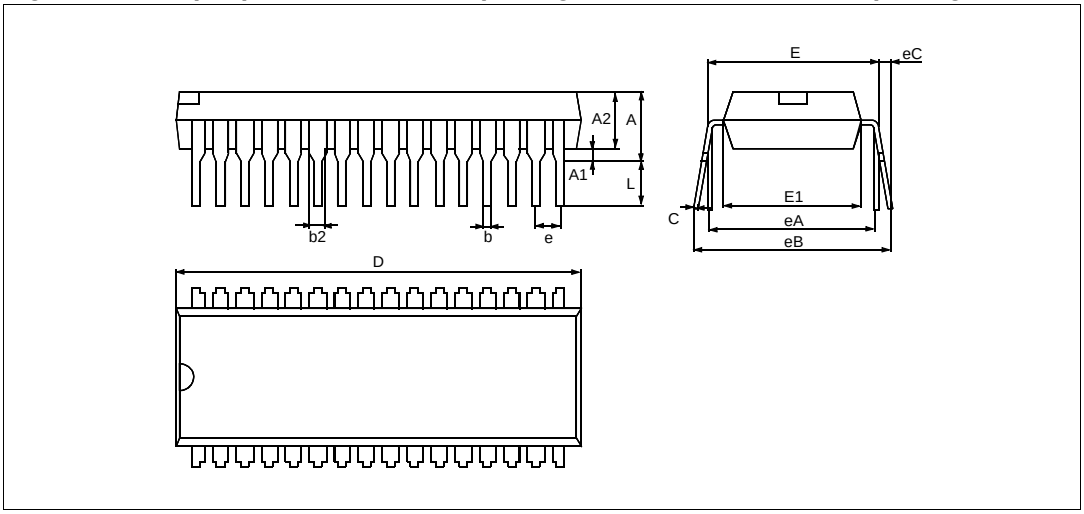


Table 99. 32-pin plastic dual in-line package, shrink 400-mil width, mechanical data

Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A	3.56	3.76	5.08	0.1402	0.1480	0.2000
A1	0.51			0.0201		
A2	3.05	3.56	4.57	0.1201	0.1402	0.1799
b	0.36	0.46	0.58	0.0142	0.0181	0.0228
b1	0.76	1.02	1.40	0.0299	0.0402	0.0551
C	0.20	0.25	0.36	0.0079	0.0098	0.0142
D	27.43		28.45	1.0799		1.1201
E	9.91	10.41	11.05	0.3902	0.4098	0.4350
E1	7.62	8.89	9.40	0.3000	0.3500	0.3701
e		1.78			0.0701	
eA		10.16			0.4000	
eB			12.70			0.5000
eC			1.40			0.0551
L	2.54	3.05	3.81	0.1000	0.1201	0.1500
	Number of pins					
N	32					

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