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	MARC4
Core Size	4-Bit
Speed	8MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	4KB (4K x 8)
Program Memory Type	EEPROM
EEPROM Size	-
RAM Size	256 x 4
Voltage - Supply (Vcc/Vdd)	2.2V ~ 6.2V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-BSOP (0.295", 7.50mm Width)
Supplier Device Package	44-SSO
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/t48c510-ilq

Table of Contents

Table of Contents (continued)

Table 2 Interrupt priority table

Interrupt	Priority	EEPROM Address	Maskable	Interrupt Opcode
INT0	lowest	040h	Yes	C8h (SCALL 040h)
INT1	|	080h	Yes	D0h (SCALL 080h)
INT2	|	0C0h	Yes	D8h (SCALL 0C0h)
INT3	|	100h	Yes	E8h (SCALL 100h)
INT4	|	140h	Yes	E8h (SCALL 140h)
INT5	|	180h	Yes	F0h (SCALL 180h)
INT6	↓	1C0h	Yes	F8h (SCALL 1C0h)
INT7	highest	1E0h	Yes	FCh (SCALL 1E0h)

1.3.1 Hardware Interrupts

Table 3 Hardware interrupts

Interrupt	Possible Interrupt Priorities								RST	Interrupt Mask		Function
Source	0	1	2	3	4	5	6	7		Register	Bit	
NRST external									X	–	–	low level active
Watchdog									#	–	–	1/2 – 2 sec. time out
Port A coded reset									#	–	–	level any inputs
Port A monitor		*		*		*		*		PAIPR	3	any edge, any input
Port B monitor		*		*		*		*		PBIPR	3	any edge, any input
Port 60 external		*		*		*		*		P6CR	1,0	any edge
Port 61 external	*		*		*		*			P6CR	3,2	any edge
Interval timer INTA		*				*				ITIPR	0	1 of 8 frequencies (1 – 128 Hz)
Interval timer INTB			*				*			ITIPR	1	1 of 8 frequencies (8 – 8192 Hz)
Timer 0		*		*		*		*		T0CR	0	overflow/compare/ end measurement
Timer 1	*		*		*		*			T1CR	0	compare

X = hardwired (neither optional or software configurable)
# = configurable option (see “Hardware Options”)
* = software configurable (see “Peripheral Modules” section for further details)

In the T48C510, there are eleven hardware interrupt sources which can be programmed to occupy a variety of priority levels. With the exception of the reset sources (RST), each source can be individually masked by mask bits in the corresponding control registers. An overview of the possible hardware configurations is shown in table 3.

1.3.2 Software Interrupts

The program can generate interrupts using the software interrupt instruction (SWI) which is supported in qFORTH by predefined macros named SWI0...SWI7. The software triggered interrupt operates in exactly the same way as any hardware triggered interrupt.

The SWI instruction takes the top two elements from the expression stack and writes the corresponding bits via the I/O bus to the interrupt pending register. Thus, by using the SWI instruction, interrupts can be re-prioritized or lower priority processes scheduled for later execution.

maintained stable to within a tolerance of ± 10% over the full operating temperature and voltage range.

For example: A $SYSCL_{max}$ frequency of 2 MHz, can be obtained by connecting a resistor R_{ext} = 150 kΩ (see figures 14, 52, 53 and 54).

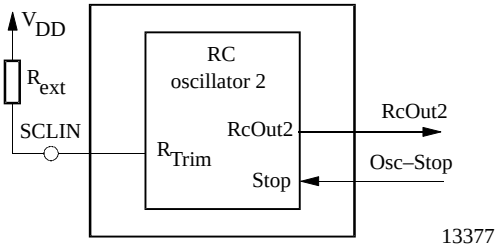

Figure 14. RC-oscillator 2

4-MHz Oscillator

The integrated system clock oscillator requires an external crystal or ceramic resonator connected between the OSCIN and OSCOUT pins to establish oscillation. All the necessary oscillator circuitry, with the exception of the actual crystal, resonator and the optional C3 and C4 are integrated on-chip.

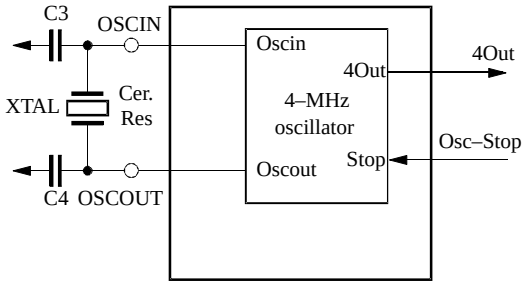

Figure 15. System clock oscillator

32-kHz Oscillator

Some applications require accurate long-term time keeping without putting excessive demands on the CPU or alternatively low resolution computing power. In this case, the on-chip ultra low power 32-kHz crystal oscillator can be used to generate both the SUBCL and/or the SYSCL. In this mode, power consumption can be significantly reduced. The 32-kHz crystal oscillator will remain operating (not stopped) during any CPU power-down/SLEEP mode.

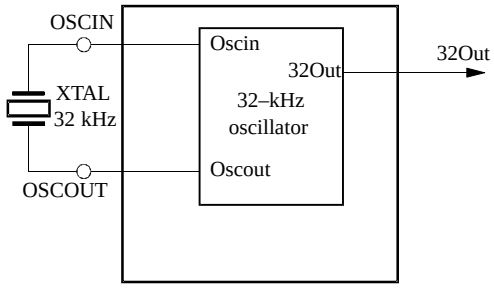

Figure 16. 32-kHz crystal oscillator

Quartz Oscillator Configuration

If the customer's application necessitates the use of a quartz crystal clock source and this requires capacitive trimming, the trimming capacitors are not integrated into the MTP unlike the M44C510E and should therefore be connected externally as descrete components between the respective Quartz Crystal terminals (OSCIN, OSCOUT) and VSS.

2 Peripheral Modules

2.1 Addressing Peripherals

Accessing the peripheral modules takes place via the I/O bus (see figure 18). The IN or OUT instructions allow direct addressing of up to 16 I/O modules. A dual register addressing scheme has been adopted which addresses the “primary register” directly. To address the “auxiliary register”, the access must be switched with an “auxiliary switching module”. Thus, a single IN (or OUT) to the module address will read (or write) into the module primary register. Accessing the auxiliary register is performed with the same instruction preceded by writing the module address into the auxiliary switching module. Byte-wide registers are accessed by multiple IN (or OUT) instructions. Extended addressing is used for more complex peripheral modules, with a larger number of registers. In this case, a bank of up to 16 subport registers are indirectly addressed with the subport address being initially written into the auxiliary register. Please refer to the ’HARDC510.SCR’ hardware interface file as a programming guideline.

Port Data Direction Register (PxDDR)

Auxiliary register address: 'Port address'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
PxDDR	**PxDDR3**	**PxDDR2**	**PxDDR1**	**PxDDR0**	**Reset value: 1111b**

Table 9 Port Data Direction Register (PxDDR)

Code: 3 2 1 0	Function
x x x 1	BPx0 in input mode
x x x 0	BPx0 in output mode
x x 1 x	BPx1 in input mode
x x 0 x	BPx1 in output mode
x 1 x x	BPx2 in input mode
x 0 x x	BPx2 in output mode
1 x x x	BPx3 in input mode
0 x x x	BPx3 in output mode

2.2.1 Bidirectional Port 0 and Port 1

In this port type, the data direction register is not independently software programmable because the direction of the complete port is switched automatically when an I/O instruction occurs (see figure 19). The port can be switched to output mode with an OUT instruction and to input with an IN instruction. The data written to a port will be stored in the output data latches and appears immediately at the port pin following the OUT instruction. After RESET, all output latches are set to '1' and the ports are switched to input mode. An IN instruction reads the condition of the associated pins.

Note: Care must be taken when switching these bidirectional ports from output to input. The capacitive pin loading at this port, in conjunction with the high resistance pull-ups, may cause the CPU to read the contents of the output data register rather than the external input state. This can be avoided by using either of the following programming techniques:

- Use two IN instructions and DROP the first data nibble. The first IN switches the port from output to input and the DROP removes the first invalid nibble. The second IN reads the valid pin state.
- Use an OUT instruction followed by an IN instruction. With the OUT instruction, the capacitive load is charged or discharged depending on the optional pull-up /pull-down configuration. Write a "1" for pins with pull-up resistors, and a "0" for pins with pull-down resistors.

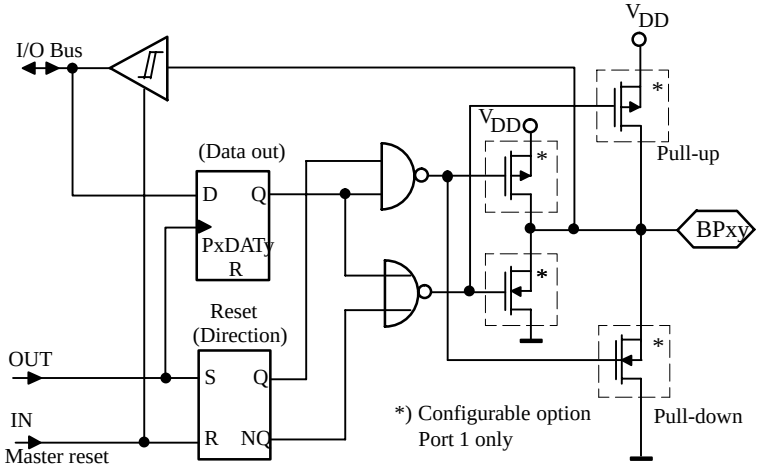

Figure 19. Bidirectional Ports 0 and 1

2.2.2 Bidirectional Port 5, Port 7 and Port C

All bidirectional ports except Port 0 and Port 1, include a bitwise-programmable Data Direction Register (PxDDR) which allows the individual programming of each port bit as input or output. It also enables the reading of the pin condition in output mode.

The bidirectional Ports 5, 7 and C as well as Port A and Port B are equipped with the same standard I/O logic. However, Port 5, Port 7 and Port C include standard CMOS input stages, whereas Port A, Port B and all other digital signal pins have Schmitt-trigger inputs. Port 5 and Port 7 have high current output drive capability for up to 20 mA @ 5 V. Whereby the instantaneous sum of the output currents should not exceed 100 mA.

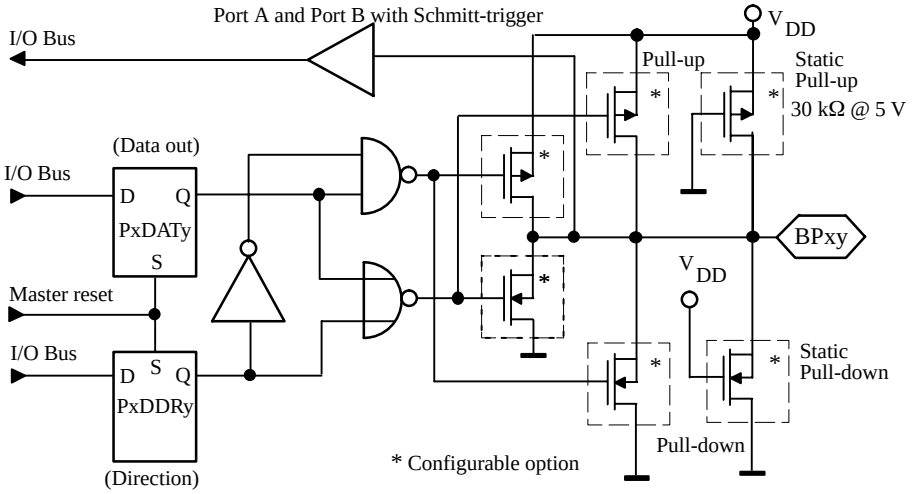

Figure 20. Bidirectional Ports 5, 7, A, B and C

2.2.3 Bidirectional Port A and Port B with Port Monitor Function

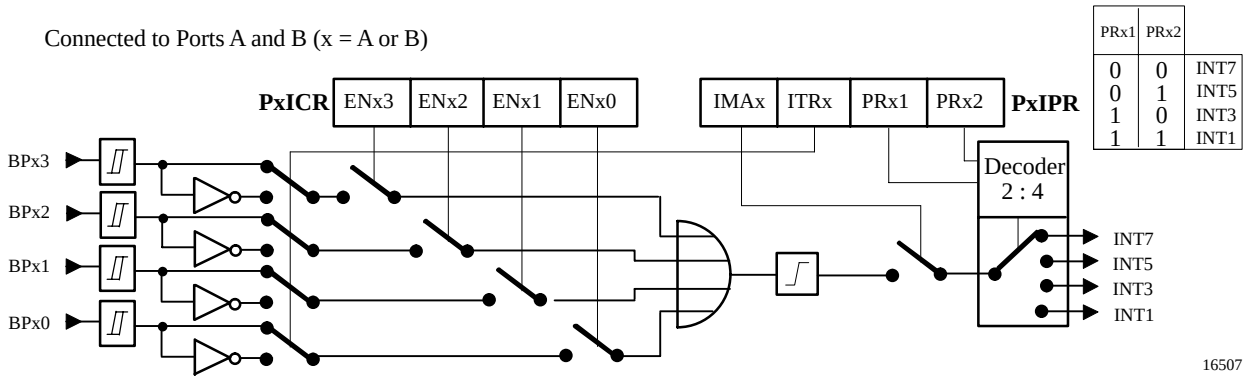

Figure 21. Port monitor module of Port A and Port B

In addition to the standard I/O functions described in section 2.2.2, both Port A (BPA3 – BPA0) and Port B (BPB3 – BPB0) are equipped with Schmitt-trigger inputs and a port monitor module. This module is connected across all four port pins (see figure 21) and is intended for monitoring those pins selected by control bits Enx3 – Enx0 and generating an interrupt when the first pin leaves a preselected logical default idle state. This state is defined by control bit ITRx . Transitions on other pins will only cause an interrupt if the other pins have first returned to the idle state. This, for example is useful for interrupt initiated port scanning without the power consuming task of continuously polling for port activity.

Using the Port Interrupt Control Register (PxICR), pins can be individually selected. A non-selected pin cannot generate an interrupt. The Port Interrupt Priority Register (PxIPR) allows masking of each interrupt, definition of

2.3.1 Interval Timer Registers

The Interval Timer Frequency Select Register (ITFSR) is I/O mapped to the primary address register of the prescaler/ interval timer address ('F'hex) and the Interval Timer Interrupt Priority Register (ITIPR) to the corresponding auxiliary register. The interrupt masks MIA and MIB enable interrupt masking of INTA and INTB respectively. Each interrupt source can be programmed with PRA and PRB to one of two interrupt priority levels. Disabling both interrupts resets the interval timer.

Interval Timer Interrupt Priority Register (ITIPR)

Auxiliary register address (write only): 'F'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
ITIPR	**PRB**	**PRA**	**MIB**	**MIA**	**Reset value: 1111b**

PRB – Priority select Interval Timer Interrupt INTB
PRA – Priority select Interval Timer Interrupt INTA
MIB – Mask Interval Timer Interrupt INTB
MIA – Mask Interval Timer Interrupt INTA

Table 13 Interval Timer Interrupt Priority Register (ITIPR)

Code 3 2 1 0	Function
x x 1 1	Reset prescaler and halt
x x x 1	Interrupt A disabled
x x x 0	Interrupt A enabled
x x 1 x	Interrupt B disabled
x x 0 x	Interrupt B enabled
x 1 x x	Interrupt A => priority 1
x 0 x x	Interrupt A => priority 5
1 x x x	Interrupt B => priority 2
0 x x x	Interrupt B => priority 6

Interval Timer Frequency Select Register (ITFSR)

Primary register address (write only): 'F'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
ITFSR	**FS3**	**FS2**	**FS1**	**FS0**	**Reset value: 1111b**

FS3 ... 0 – Frequency select code

Table 14 Interval Timer Frequency Select Register (ITFSR)

Code 3 2 1 0	Function	SUBCL divide by	SUBCL = 32 kHz		Code 3 2 1 0	Function	SUBCL divide by	SUBCL = 32 kHz
0 0 0 0	INTA	2^{15}	Select 1 Hz		1 0 0 0	INTB	2^{12}	Select 8 Hz
0 0 0 1		2^{14}	Select 2 Hz		1 0 0 1		2^{11}	Select 16 Hz
0 0 1 0		2^{13}	Select 4 Hz		1 0 1 0		2^{9}	Select 64 Hz
0 0 1 1		2^{12}	Select 8 Hz		1 0 1 1		2^{7}	Select 256 Hz
0 1 0 0		2^{11}	Select 16 Hz		1 1 0 0		2^{5}	Select 1024 Hz
0 1 0 1		2^{10}	Select 32 Hz		1 1 0 1		2^{4}	Select 2048 Hz
0 1 1 0		2^{9}	Select 64 Hz		1 1 1 0		2^{3}	Select 4096 Hz
0 1 1 1		2^{8}	Select 128 Hz		1 1 1 1		2^{2}	Select 8192 Hz

The control bit FS3 determines whether the INTA or the INTB buffer register is loaded with the select code (FS2–FS0). This allows independent programming of interval times for INTA and INTB.

2.4 Watchdog Timer

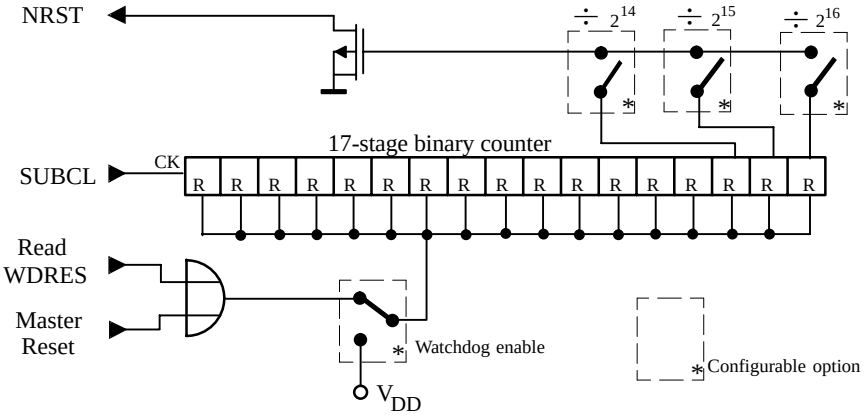

Figure 27. Watchdog timer

The watchdog timer is a 17-stage binary divider clocked by SUBCL generated within the clock module (see figures 11 and 27). It can only be enabled as a configurable option whereby it must be periodically reset from the application program. The program cannot disable the watchdog. If the CPU find itself for an extended length of time in SLEEP mode or in a section of program that includes no watchdog reset, then the watchdog will overflow, thus forcing the NRST pin low. This initiates a master reset. The timeout period can be set to 0.5, 1 or 2 seconds (if SUBCL = 32 kHz) by using a configurable option.

To reset the watchdog, the program must perform an IN-instruction on the address CWD (’3’hex). No relevant data is received. The operation is therefore normally followed by a DROP to flush the data from the stack.

2.5 Timer/Counter Module (TCM)

The TCM consists of two timer/counter blocks (Timer 0 and Timer 1) which can be used separately, or together as a single 16-bit counter/timer (see figures 28 and 30). Each timer can be supplied by various internal or external clock sources. These can be selected and divided under program control using the Timer/Counter Control Register (TCCR), the Timer 0 Control Register (T0CR) and the Timer 1 Control Register (T1CR). Capture and compare registers (T0CA,T1CA,T0CP and T1CP) not only allow event counting, but also the generation of various timed output waveforms including programmable frequencies, modulated melody tones, Pulse Width Modulated (PWM) and Pulse Density Modulated (PDM) output signals. When in one of these signal generation modes, the capture register acts as timer shadow register, the current timer state is freezed whenever read by the CPU. The Timer 0 is further equipped for performing a variety of time measurement operations. In this mode the capture register is used together with the gating logic for performing asynchronous, externally triggered snapshot measurements. These measurements include single input pulse width and period measurements and also dual input phase and positional measurement. The mode configuration is set in the Timer 0 and Timer 1 Mode Registers (T0MO and T1MO).

Each timer represents a single maskable interrupt source (T0INT and T1INT), the priority of which can be configured under program control. A Timer 0 interrupt can be caused by any of three conditions (overflow, compare or end-of-measurement). The associated status register (T0SR) differentiates between these. A status register is not necessary in the Timer 1 as an interrupt is caused only on a compare condition.

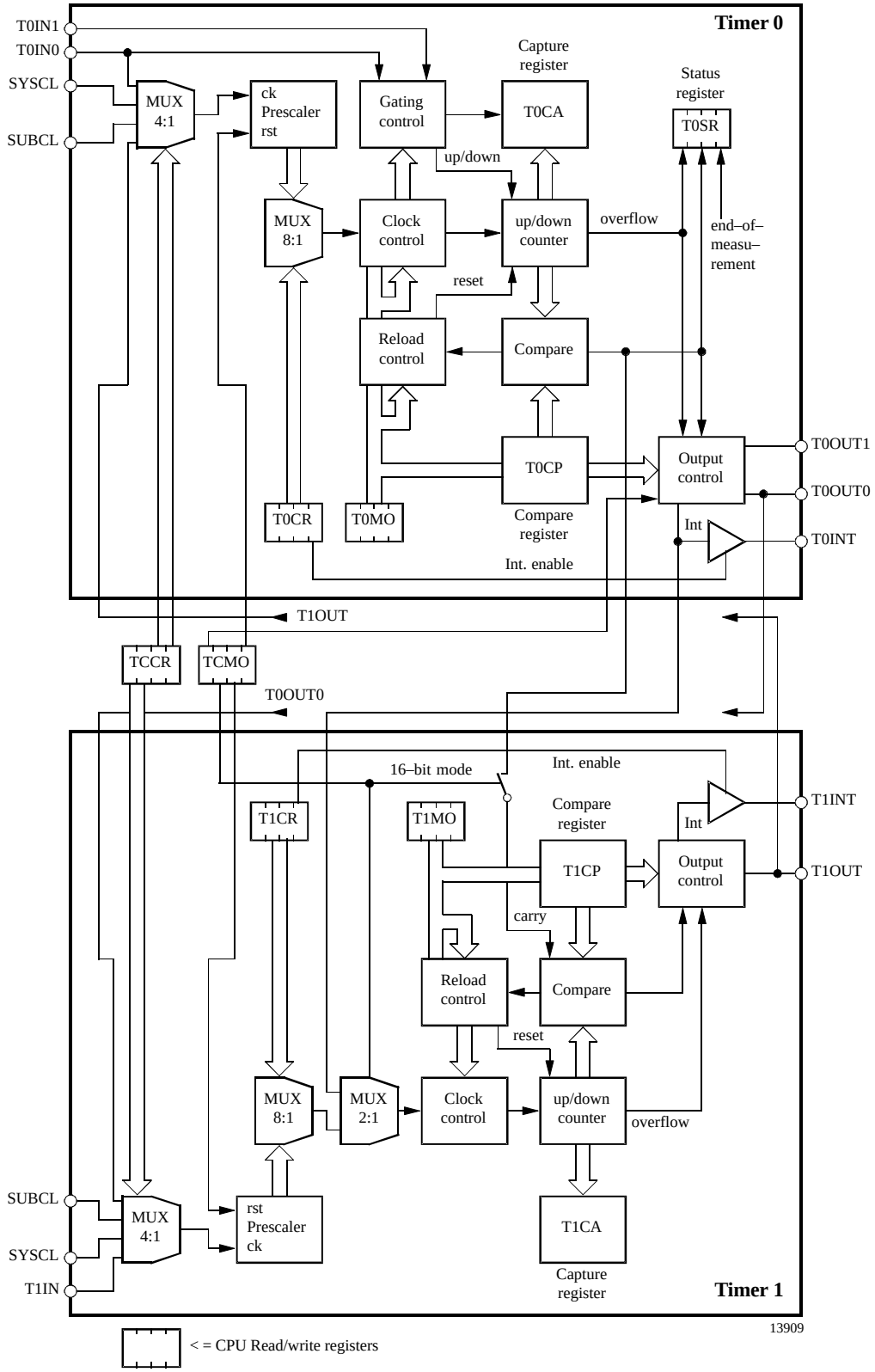

Figure 28. Timer/counter module

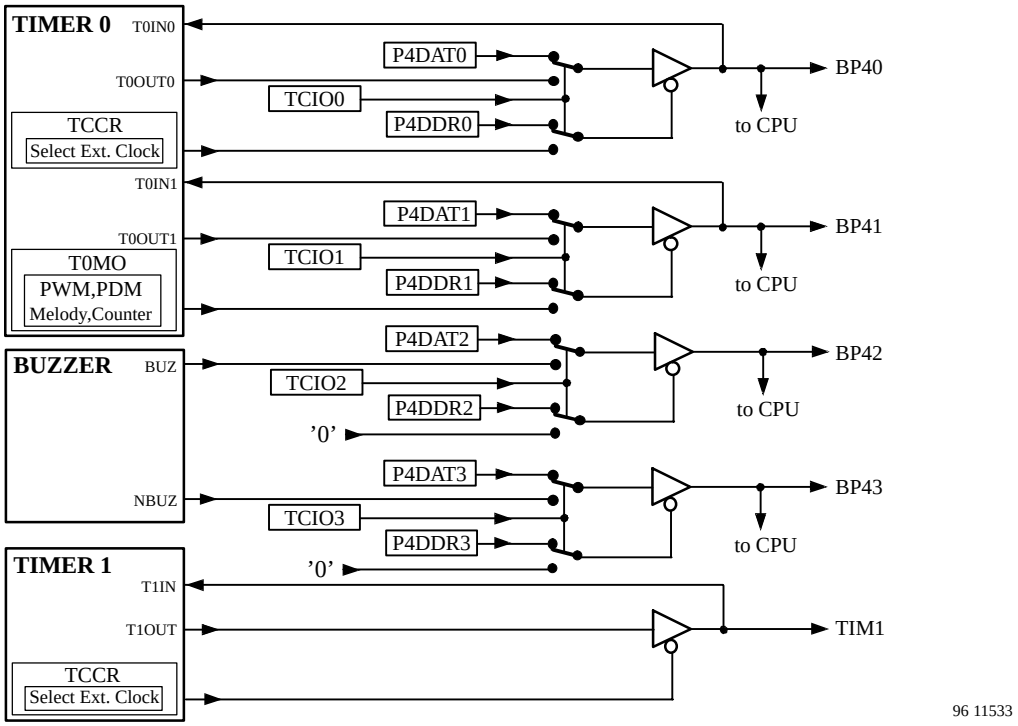

Figure 29. Timer/counter and buzzer external interface

Timer/Counter Mode Register (TCMO)

Subport address (indirect write access): '4'hex of Port address '9'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
TCMO	**T0NINV**	**TC8**	**T1RST**	**T0RST**	**Reset value: 1111b**

T0NINV – Timer 0 output (BP41) appears non-inverted at BP40

TC8 – Timer/Counter in 8-/16-bit mode

T1STP – Timer 1 Stop/Run

T0STP – Timer 0 Stop/Run

Table 18 Timer/Counter Mode Register (TCMO)

Code 3 2 1 0	Function
x x x 0	Timer 0 running
x x x 1	Timer 0 halted
x x 0 x	Timer 1 running
x x 1 x	Timer 1 halted
x 0 x x	Timer/counter in 16-bit mode
x 1 x x	Timer/counter in 8-bit mode
0 x x x	Inverted output BP41 appears on BP40 (BP40 = NOT BP41)
1 x x x	Non-inverted output BP41 appears on BP40 (BP40 = BP41)

Timer 0 Mode Register (T0MO)

Subport address (indirect write access): ’0’hex of Port adress ’9’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T0MO	**T0MO3**	**T0MO2**	**T0MO1**	**T0MO0**	**Reset value: 1111b**

T0MO3 ... 0 – Timer 0 Mode Code

Table 19 Timer 0 Mode Register (T0MO)

Code 3 2 1 0	Function	Assuming TCIOR1=TCIOR0=low		Interrupt set / T0SR affected		
		BP40 (*3)	BP41	cmp	ofl	eom
0 0 0 0	reserved			–	–	–
0 0 0 1	reserved			–	–	–
0 0 1 0	Modulated melody mode	Envelope (out)	Tone (out)	y/y	y/y	n/n
0 0 1 1	Melody mode	Tone (out)	Tone (out)	y/y	y/y	n/n
0 1 0 0	Counter-auto reload (50% duty cycle)	Toggle (out) /Clock (in)	Toggle (out)	y/y	y/y	n/n
0 1 0 1	Counter-free running (50% duty cycle)	Toggle (out) /Clock (in)	Toggle (out)	n/y	y/y	n/n
0 1 1 0	Pulse density modulation	PDM (out) /Clock (in)	PDM (out)	n/y	y/y	n/n
0 1 1 1	Pulse width modulation	PWM (out) /Clock (in)	PWM (out)	n/y	y/y	n/n
1 0 0 0	Phase measurement	Signal 1 (in)	Signal 2 (in)	n/n	y/y	y/y
1 0 0 1	Position measurement	Signal 1 (in)	Signal 2 (in)	(*1)	(*2)	n/n
1 0 1 0	Low pulse width measurement	Clock (in)	Signal (in)	n/y	y/y	y/y
1 0 1 1	High pulse width measurement	Clock (in)	Signal (in)	n/y	y/y	y/y
1 1 0 0	Counter- auto reload (strobe)	Strobe (out) /Clock (in)	Strobe (out)	y/y	y/y	n/y
1 1 0 1	Counter-free running (strobe)	Strobe (out) /Clock (in)	Strobe (out)	n/y	y/y	n/y
1 1 1 0	Period measurement (rising edge)	Clock (in)	Signal (in)	n/y	y/y	y/y
1 1 1 1	Period measurement (falling edge)	Clock (in)	Signal (in)	n/y	y/y	y/y

*1 **Note:** The compare interrupt/status flag can only be set when counting up.

*2 **Note:** The overflow interrupt/status flag is set on both an overflow or an underflow.

*3 **Note:** The BP40 signals can be inverted if T0NINV=0 (TCMO register)

Timer 0 Interrupt Status Register (T0SR)

Auxiliary register address (read access): ’9’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T0SR	**not used**	**T0EOM**	**T0OFL**	**T0CMP**	**Reset value: x000b**

Note: The status register is reset automatically when read and also when Timer 0 is reset.

T0EOM– Timer 0 End Of Measurement status flag

T0OFL – Timer 0 OverFLow status flag

T0CMP – Timer 0 CoMPare status flag

Timer 0 Free Running Counter Modes (Strobe and 50% Duty Cycle)

In the free running counter mode, Timer 0 can be used as an event counter for summing external event pulses on BP40, or as a timer with an internal time-based clock. When enabled, the counter will count up generating an output signal on BP41 whenever the counter contents match the compare register (see figure 31). This signal can appear either as a strobe pulse or as a simple toggling of the output state (50% duty cycle) depending on the timer mode. Interrupts (if not masked) are generated every 256 clocks on the overflow condition. The current counter state can be read at any time by reading the capture register,. The compare register has no effect on the counter cycle time and will not influence interrupts.

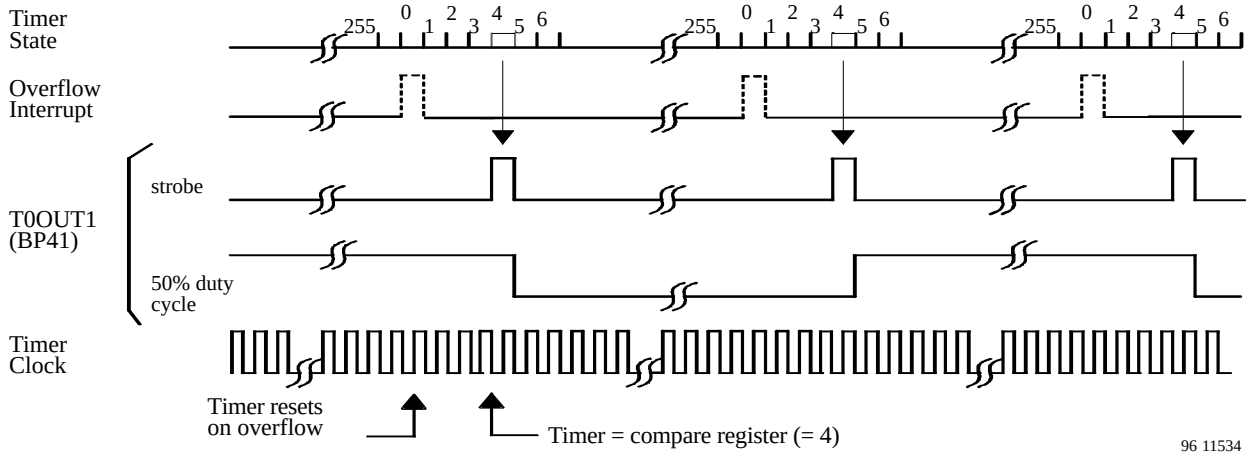

Figure 31. Timer 0 free running counter mode

Timer 0 Counter Reload Modes (Strobe and 50% Duty Cycle)

As in the free running mode, the counter can also be clocked from either an external signal on BP40 or from an internal clock source. In this mode, the counter repetition period is completely defined by the contents of the compare register (T0CP) (see figure 32). The counter counts up with the selected clock frequency. When it reaches the value held in the compare register, the counter then returns to the zero state. At the same time, depending on the selected timer mode, the BP41 either toggles or generates a strobe pulse. If the Timer 0 interrupt is unmasked, a compare interrupt is also generated.

The resultant output frequency $f_{OUT} = f_{IN}/2*(n+1)$ where
n = compare value (n = 1 – 255).

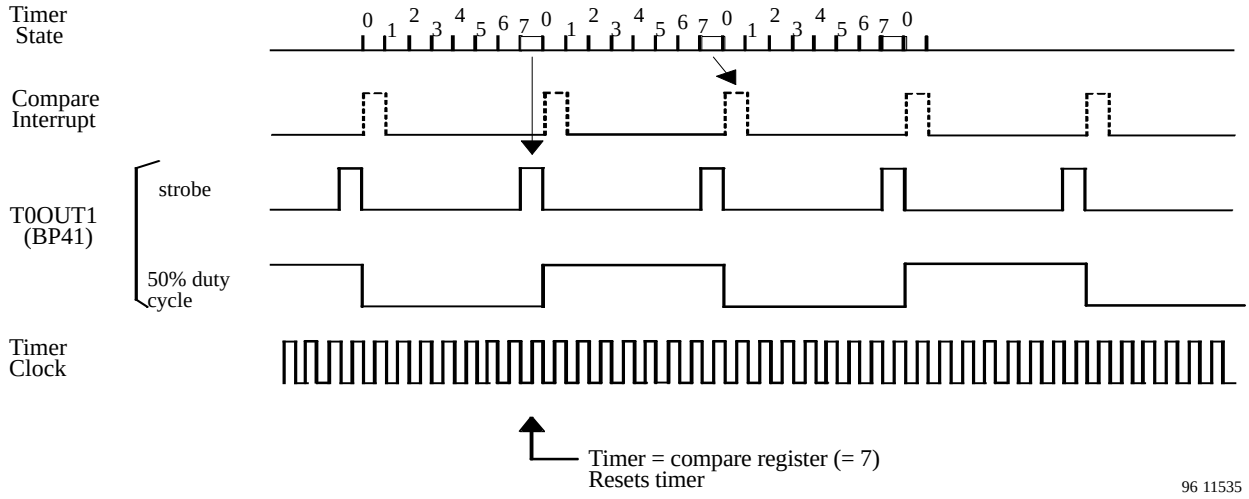

Figure 32. Timer 0 counter reload mode

Period Measurement Modes (Rising and Falling Edge)

During the period measurement mode, the counter counts the number of either internal or external clocks in one period of the BP41 input signal (see figure 37). Dependent on the mode chosen, this will be from rising edge to the next rising edge or conversely, falling edge to the following falling edge. On the trigger edge, the counter state is loaded into the capture register and subsequently reset. The measured value remains in the capture register until overwritten by the following measured value. Interrupts can be generated by either an overflow condition or an end-of-measurement (eom) event. An 'eom' event signals the CPU that a new measured value is present in the capture register and can be read, if required.

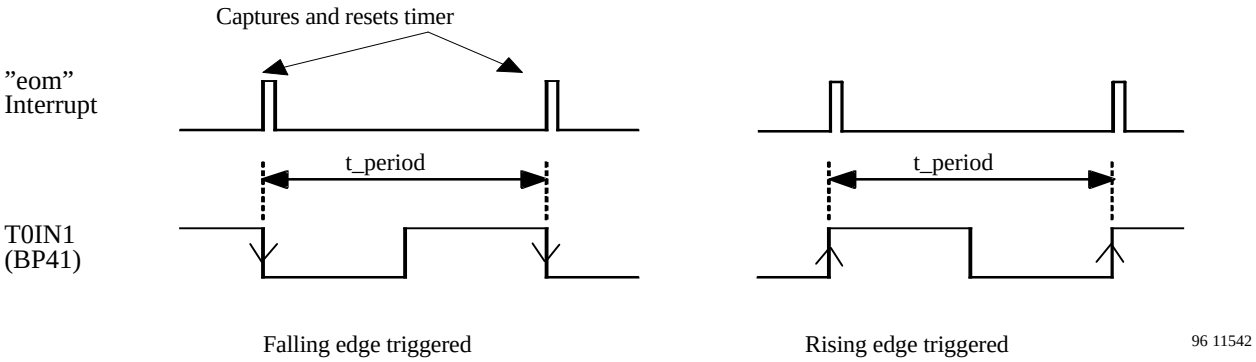

Figure 37. Period measurement

Pulse Width Measurement Modes (High and Low)

In this mode, the selected clock source is gated to the counter for the duration of each input pulse received on BP41 (see figure 38). Whether the measurement takes place during the high or low phase depends on the selected mode. At the end of each pulse, the counter state is loaded into the capture register and subsequently reset. Interrupts can be generated by either an overflow condition or an end-of-measurement (eom) event. An 'eom' event signals the CPU that a new measured value is present in the capture register can be read, if required.

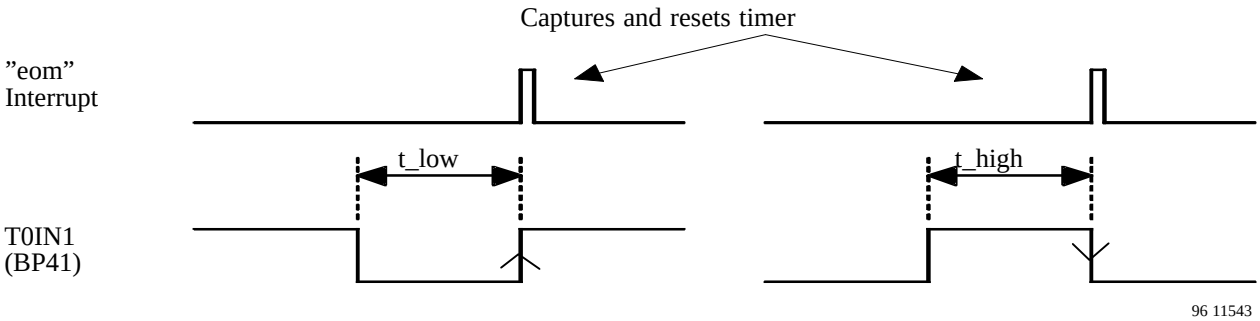

Figure 38. Pulse width measurement

Phase Measurement Mode

This mode allows the Timer 0 to measure the phase misalignment between two 1:1 mark space ratio input signals connected to the BP40 and BP41 pins (see figure 39). The counter clock is gated with the phase misalignment period (tp), during which time the counter increments with the selected clock frequency. This misalignment period is defined as the period during which BP40 is high and BP41 is low. Capturing and resetting of the counter always takes place on the rising edge of BP41. The measured value remains in the capture register until overwritten by the next measurement. Interrupts can be generated by either an overflow condition or an end-of-measurement ('eom') event. An 'eom' event signals the CPU that a new measured value is present in the capture register and can be read, if required.

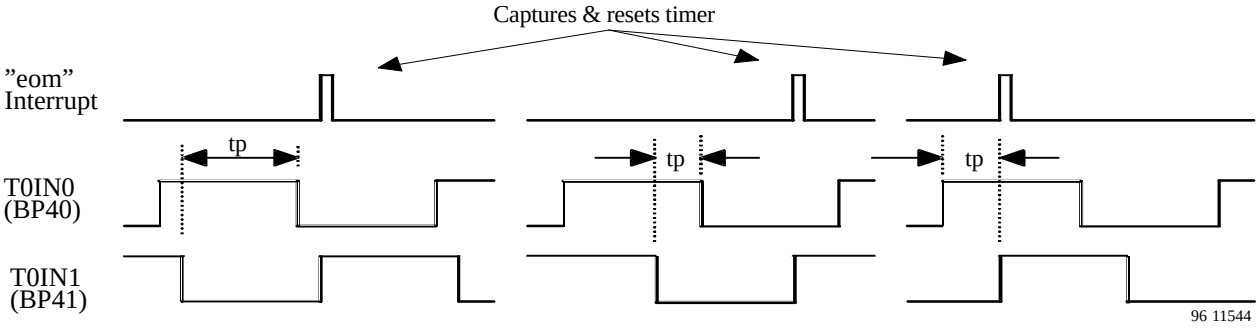

Figure 39. Phase measurement

Position Measurement Mode

This mode is intended for the evaluation of positional sensors with biphase output signals. Figure 40 illustrates a typical positional sensor system which delivers both incremental positional stepping signals and also directional information. The direction can be deduced from the relative phase of the two signals. Therefore if BP40 is high on the rising edge of BP41, the moving mask travels to the left and if it is low then it travels to the right. The direction (left/right) information is used to set the direction of the up/down counter which enables the BP40 pulses to be counted. Assuming that the system has been reset on a reference position, the counter will always hold the absolute current position of the moving mask. This can be read by the CPU if necessary. This mode is the only one in which the counter is allowed to decrement. Therefore, in this case it is possible for both an underflow or an overflow to occur. The overflow interrupt (if unmasked) will trigger on either of these conditions while the compare interrupt on the other hand will only trigger if the counter is counting upwards. To differentiate between an overflow or underflow, the compare value can be set to '0' hex, for example. An overflow would then set both the overflow and compare status flags while an underflow sets the overflow status flag only.

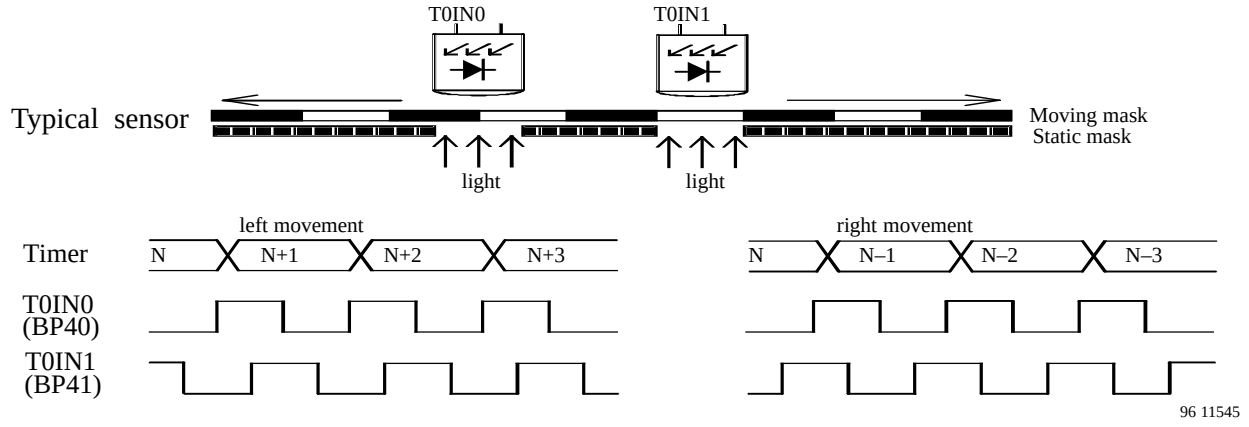

Figure 40. Position measurement mode

2.5.4 Timer 1 Modes

The Timer 1 is aimed at performing event counting and timing functions (see figure 28). It has, unlike the Timer 0, no gated clock or externally triggered capture modes. The counter counts up with an internal or external clock, depending on the state of the Timer 1 Control Register (T1CR) and the Timer/Counter Clock Control Register (TCCR) and generates a compare interrupt whenever the counter matches the Timer 1 compare register. This is the only Timer 1 interrupt source. Masking can be performed using the mask bit in the Timer 1 Control Register (T1CR) and priority can be defined in the Timer/Counter Interrupt Priority Register (TCIP). The TIM1 pin is used by the Timer 1 either as clock/event input or timer output. I/O control of the Timer 1 pin TIM1 is controlled entirely by the hardware, therefore if the TIM1 is selected as an external clock or event source (in the TCCR), there can be no Timer 1 signal output. In this case, the timer would be used solely to generate interrupts.

In autostop operation, the Timer 1 will halt both itself and Timer 0 whenever the Timer 1 compare value is reached. This feature can be used for example to generate an exact burst of pulses. Both timers will remain stopped until restarted. Restarting is performed in the normal way by setting the appropriate control bits in the Timer/Counter Mode Register (TCM0).

Timer 1 Mode Register (T1MO)

Subport address (indirect write address): ’2’hex of Port address ’9‘hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T1MO	**T1MO3**	**T1MO2**	**T1MO1**	**T1MO0**	**Reset value: 1111b**

T1MO3 ... 0 – Timer 1 Mode Control

Table 22 Timer 1 Mode Register (T1MO)

Code 3 2 1 0	Function	Compare Interrupt
x x 0 0	Counter free running (50% duty cycle)	yes
x x 0 1	Counter auto reload (50% duty cycle)	yes
x x 1 0	Pulse width modulation	yes
x x 1 1	Counter auto-reload (strobe output)	yes
x 0 x x	Increment on falling edge of clock	–
x 1 x x	Increment on rising edge of clock	–
1 x x x	Normal operation (no autostop)	yes
0 x x x	Autostop operation (Timer 1 stops Timer 2)	yes

Timer 1 Control Register (T1CR)

The T1CR is responsible for the predivision of the selected Timer 1 input clock (see TCCR). It can be divided or used directly as clock for the up counter. Bit 0 is the mask bit for the Timer 1 interrupt.

Subport address (indirect write access): ’3’hex of Port adress ’9‘hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T1CR	**T1FS3**	**T1FS2**	**T1FS1**	**T1IM**	**Reset value: 1111b**

T1FS3 ... 1 – Timer 1 Prescaler Division Factor Code

T1IM – Timer 1 Interrupt Mask

Buzzer Control Register (BZCR)

Subport address (indirect write access): 'A'hex of Port adress '9'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
BZCR	**BZFS2**	**BZFS1**	**BZOP**	**BZOF**	**Reset value: 1111b**

BZFS2, BZFS2 – Buzzer Frequency Select code

BZOP – Buzzer Output Stop State

BZOF – Buzzer off/on

Table 24 Buzzer Control Register (BZCR)

Code 3 2 1 0	Function
x x x 0	Buzzer on
x x x 1	Buzzer off
x x 0 x	Buzzer output stop state: BP42 = BP43 = low
x x 1 x	Buzzer output stop state: BP42 = BP43 = high
0 0 x x	Buzzer frequency: 32 kHz (= SUBCL)
0 1 x x	Buzzer frequency: 8 kHz (= SUBCL / 4)
1 0 x x	Buzzer frequency: 4 kHz (= SUBCL / 8)
1 1 x x	Buzzer frequency: 2 kHz (= SUBCL / 16)

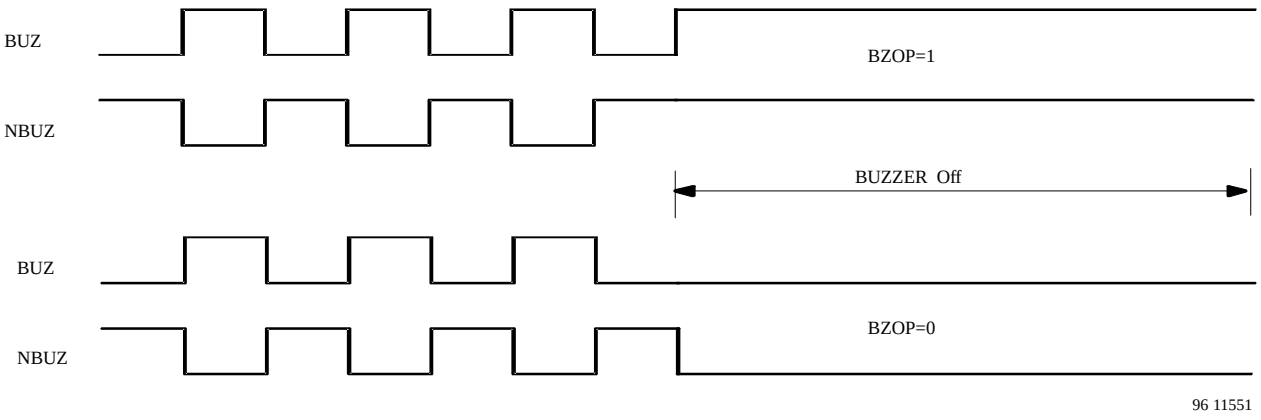

Figure 45. Buzzer waveform

other on board components either present an excessive power supply load or be unable to withstand the ICP 5-Volt supply voltage, then the VDD power line should also be jumpered (J2).

During the programming operation all ports are set into input mode, with the previously programmed pullup/pull-downs transitors deactivated.

In normal operational mode, the PM pin is strapped to ground and the Port 0 reverts to a port function as described in section 2.2.1.

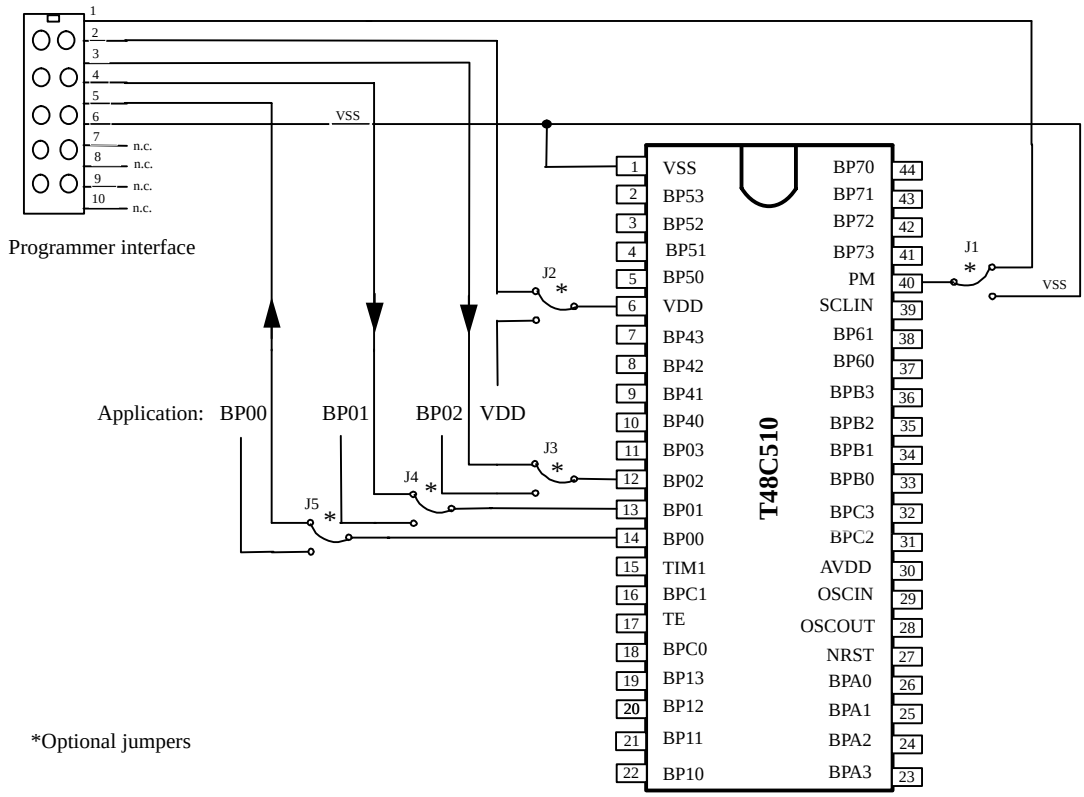

Figure 47. In-system programming

2.8 Noise Considerations

When designing the microcontroller based application, several factors should be taken into consideration to increase noise immunity and reduce electromagnetic emission (EME). Many such potential problems can be avoided by careful layout of the printed circuit board (PCB). The PCB contains many parasitic components which at first sight are not apparent. PCB tracks can act as antennas or as coupling capacitors. Long stretches of parallel tracks and long high frequency signal lines should thus be avoided wherever possible to minimise the chance of picking up or transmitting unwanted signals.

2.8.1 Noise Immunity

The following guidelines will increase system noise immunity:

- Unconnected inputs should not be left open. If port pins are not required then it is recommended to set pullup or pulldown option on these pins.
- Special care should be taken when laying out the PCB that interrupt, reset and clock signal lines are kept short and are carefully shielded or have sufficient spacing from other on board noise generating sources.
- A quartz crystal should always be directly located immediately next to the microcontroller crystal oscillator terminals (OSCIN and OSCOUT), the connections being always very short. This avoids not only signal coupling onto the clock source but can also reduces EME.

5 Hardware Options

The following list shows all the T48C510 hardware options that can be programmed into the configuration EEPROM. SPD –> strong static pull-down, SPU –> strong static pull-up

Port 0 **Output** Standard drive

BP00 CMOS Pull-up

BP01 CMOS Pull-up

BP02 CMOS Pull-up

BP03 CMOS Pull-up

Port 1 **Output** Standard drive

BP10 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P]

BP11 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P]

BP12 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P]

BP13 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P]

Port 4 **Output** Standard drive

BP40 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP41 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP42 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP43 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

Port 5 **Output** Standard drive

BP50 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP51 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP52 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP53 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

Port 7 **Output** Standard drive

BP70 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP71 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP72 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

BP73 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (30 kΩ)
❑ SPD (30 kΩ)

Port 6 **Output** Standard drive

BP60 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (4 kΩ)
❑ SPD (4 kΩ)

BP61 ❑ CMOS ❑ Pull-up
❑ Open drain [N] ❑ Pull-down
❑ Open drain [P] ❑ SPU (4 kΩ)
❑ SPD (4 kΩ)

Ozone Depleting Substances Policy Statement

It is the policy of **Atmel Germany GmbH** to

1. Meet all present and future national and international statutory requirements.

2. Regularly and continuously improve the performance of our products, processes, distribution and operating systems with respect to their impact on the health and safety of our employees and the public, as well as their impact on the environment.

It is particular concern to control or eliminate releases of those substances into the atmosphere which are known as ozone depleting substances (ODSs).

The Montreal Protocol (1987) and its London Amendments (1990) intend to severely restrict the use of ODSs and forbid their use within the next ten years. Various national and international initiatives are pressing for an earlier ban on these substances.

Atmel Germany GmbH has been able to use its policy of continuous improvements to eliminate the use of ODSs listed in the following documents.

1. Annex A, B and list of transitional substances of the Montreal Protocol and the London Amendments respectively

2. Class I and II ozone depleting substances in the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA

3. Council Decision 88/540/EEC and 91/690/EEC Annex A, B and C (transitional substances) respectively.

Atmel Germany GmbH can certify that our semiconductors are not manufactured with ozone depleting substances and do not contain such substances.

Data sheets can also be retrieved from the Internet: http://www.atmel–wm.com

Atmel Germany GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany
Telephone: 49 (0)7131 67 2594, Fax number: 49 (0)7131 67 2423