

# Proton+ PIC18 Timer Macros

## Table Of Contents

<b>Introduction .....</b>	<b>2</b>
<b>Version vs. Devices .....</b>	<b>2</b>
<b>CloseTimer .....</b>	<b>3</b>
CloseTimer0 .....	3
CloseTimer1 .....	3
CloseTimer2 .....	3
CloseTimer3 .....	3
CloseTimer4 .....	3
CloseTimer5 .....	3
<b>OpenTimer .....</b>	<b>4</b>
OpenTimer0 .....	4
OpenTimer1 (V1, V2, V3, V4, V5) .....	5
OpenTimer1 (V6) .....	6
OpenTimer2 .....	8
OpenTimer3 (V2, V4) .....	9
OpenTimer3 (V6) .....	10
OpenTimer4 .....	12
OpenTimer5 .....	13
<b>ReadTimer .....</b>	<b>14</b>
ReadTimer0 .....	14
ReadTimer1 .....	14
ReadTimer2 .....	14
ReadTimer3 .....	14
ReadTimer4 .....	15
ReadTimer5 .....	15
<b>WriteTimer .....</b>	<b>16</b>
WriteTimer0 .....	16
WriteTimer1 .....	16
WriteTimer2 .....	16
WriteTimer3 .....	17
WriteTimer4 .....	17
WriteTimer5 .....	17
<b>SetTmrCCPSrc .....</b>	<b>18</b>
SetTmrCCPSrc .....	18
<b>Example Use of the Timer Macros: .....</b>	<b>19</b>
<b>T3_OSC1EN_ON .....</b>	<b>20</b>
<b>T3_OSC1EN_OFF .....</b>	<b>20</b>

# Proton+ PIC18 Timer Macros

## Introduction

The timer peripherals are supported with the following macros. The macros are a mixture of compiler types and preprocessor types, and can be found in "Includes\Sources\Timers.inc"

## Timer Macros

Macro	Description
<b>CloseTimerx</b>	Disable timer <b>x</b> .
<b>OpenTimerx</b>	Configure and enable timer <b>x</b> .
<b>ReadTimerx</b>	Read the value of timer <b>x</b> .
<b>WriteTimerx</b>	Write a value into timer <b>x</b> .
<b>SetTmrCCPSrc</b>	Configure the timer as a clock source to CCP module.

Based on the availability of on-chip timer peripherals, all PIC 18 devices are divided into following different timer versions. So, before calling the library macros care has to be taken to know the version of the configured device and to use the appropriate macros.

Below is the table to find the timer version for the configured device:

## Version vs. Devices

Version name	Device number
TMR_V1	18F1230, 18F1330
TMR_V2	18F242, 18F252, 18F442, 18F452, 18F248, 18F258, 18F448, 18F458, 18F2439, 18F2539, 18F4439, 18F4539, 18F1220, 18F1320, 18F2220, 18F2320, 18F4220, 18F4320, 18F2420, 18F2520, 18F4420, 18F4520, 18F2423, 18F2523, 18F4423, 18F4523, 18F2455, 18F2550, 18F4455, 18F4550, 18F2480, 18F2580, 18F4480, 18F4580, 18F2410, 18F2510, 18F2515, 18F2610, 18F4410, 18F4510, 18F4515, 18F4610, 18F2525, 18F2620, 18F4525, 18F4620, 18F2585, 18F2680, 18F4585, 18F4680, 18F2682, 18F2685, 18F4682, 18F4685, 18F2221, 18F2321, 18F4221, 18F4321, 18F6310, 18F6410, 18F8310, 18F8410, 18F6390, 18F6490, 18F8390, 18F8490, 18F6585, 18F6680, 18F8585, 18F8680, 18F63J11, 18F64J11, 18F65J11, 18F83J11, 18F84J11, 18F85J11, 18F63J90, 18F64J90, 18F65J90, 18F83J90, 18F84J90, 18F85J90, 18F23K20, 18F24K20, 18F25K20, 18F43K20, 18F44K20, 18F45K20, 18F13K50, 18LF13K50, 18F14K50, 18LF14K50, 18F13K22, 18F14K22, 18LF13K22, 18LF14K22, 18F66J90, 18F67J90, 18F86J90, 18F87J90
TMR_V3	18F2450, 18F4450, 18F24J10, 18F25J10, 18F44J10, 18F45J10
TMR_V4	18F6620, 18F6720, 18F8620, 18F8720, 18F6520, 18F8520, 18F6527, 18F6622, 18F6627, 18F6722, 18F8527, 18F8622, 18F8627, 18F8722, 18F6525, 18F6621, 18F8525, 18F8621, 18F65J10, 18F65J15, 18F66J10, 18F66J15, 18F67J10, 18F85J10, 18F85J15, 18F86J10, 18F86J15, 18F87J10, 18F66J60, 18F66J65, 18F67J60, 18F86J60, 18F86J65, 18F87J60, 18F96J60, 18F96J65, 18F97J60, 18F66J11, 18F66J16, 18F67J11, 18F86J11, 18F86J16, 18F87J11, 18F65J50, 18F66J50, 18F66J55, 18F67J50, 18F85J50, 18F86J50, 18F86J55, 18F87J50
TMR_V5	18F2331, 18F2431, 18F4331, 18F4431
TMR_V6	18F24J11, 18F25J11, 18F26J11, 18F44J11, 18F45J11, 18F46J11, 18F24J50, 18F25J50, 18F26J50, 18F44J50, 18F45J50, 18F46J50, 18LF24J11, 18LF25J11, 18LF26J11, 18LF44J11, 18LF45J11, 18LF46J11, 18LF24J50, 18LF25J50, 18LF26J50, 18LF44J50, 18LF45J50, 18LF46J50

# Proton+ PIC18 Timer Macros

## CloseTimer

### CloseTimer0

For **TMR\_V1**, **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Disable the specified timer.  
**Include:** `Timers.inc`  
**Prototype:** `CloseTimer0`  
**Remarks:** This macro disables the interrupt and the specified timer.

### CloseTimer1

For **TMR\_V1**, **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Disable the specified timer.  
**Include:** `Timers.inc`  
**Prototype:** `CloseTimer1`  
**Remarks:** This macro disables the interrupt and the specified timer.

### CloseTimer2

For **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Disable the specified timer.  
**Include:** `Timers.inc`  
**Prototype:** `CloseTimer2`  
**Remarks:** This macro disables the interrupt and the specified timer.

### CloseTimer3

For **TMR\_V2**, **TMR\_V4** and **TMR\_V6**

**Macro:** Disable the specified timer.  
**Include:** `Timers.inc`  
**Prototype:** `CloseTimer3`  
**Remarks:** This macro disables the interrupt and the specified timer.

### CloseTimer4

For **TMR\_V4** and **TMR\_V6**

**Macro:** Disable the specified timer.  
**Include:** `Timers.inc`  
**Prototype:** `CloseTimer4`  
**Remarks:** This macro disables the interrupt and the specified timer.

### CloseTimer5

For **TMR\_V5**

**Macro:** Disable the specified timer.  
**Include:** `Timers.inc`  
**Prototype:** `CloseTimer5`  
**Remarks:** This macro disables the interrupt and the specified timer.

# Proton+ PIC18 Timer Macros

## OpenTimer

### OpenTimer0

For TMR\_V1, TMR\_V2, TMR\_V3, TMR\_V4, TMR\_V5 and TMR\_V6

**Macro:** Configure and enable timer0.

**Include:** `Timers.inc`

**Prototype:** `OpenTimer0(pConfig)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file TimerDefs.inc.

#### Enable Timer0 Interrupt:

<code>TIMER_INT_ON</code>	Interrupt enabled
<code>TIMER_INT_OFF</code>	Interrupt disabled

#### Timer Width:

<code>T0_8BIT</code>	8-bit mode
<code>T0_16BIT</code>	16-bit mode

#### Clock Source:

<code>T0_SOURCE_EXT</code>	External clock source (I/O pin)
<code>T0_SOURCE_INT</code>	Internal clock source (TOSC)

#### External Clock Trigger (for `T0_SOURCE_EXT`):

<code>T0_EDGE_FALL</code>	External clock on falling edge
<code>T0_EDGE_RISE</code>	External clock on rising edge

#### Prescale Value:

<code>T0_PS_1_1</code>	1:1 prescale
<code>T0_PS_1_2</code>	1:2 prescale
<code>T0_PS_1_4</code>	1:4 prescale
<code>T0_PS_1_8</code>	1:8 prescale
<code>T0_PS_1_16</code>	1:16 prescale
<code>T0_PS_1_32</code>	1:32 prescale
<code>T0_PS_1_64</code>	1:64 prescale
<code>T0_PS_1_128</code>	1:128 prescale
<code>T0_PS_1_256</code>	1:256 prescale

**Remarks:** This macro configures timer0 according to the options specified and then enables it.

**Code Example:** `OpenTimer0(TIMER_INT_OFF & T0_8BIT & T0_SOURCE_INT & T0_PS_1_32)`

# Proton+ PIC18 Timer Macros

## OpenTimer1 (V1, V2, V3, V4, V5)

For TMR\_V1, TMR\_V2, TMR\_V3, TMR\_V4 and TMR\_V5

**Macro:** Configure and enable timer1.

**Include:** `Timers.inc`

**Prototype:** `OpenTimer1(pConfig)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file TimerDefs.inc.

### Enable Timer1 Interrupt:

<code>TIMER_INT_ON</code>	Interrupt enabled
<code>TIMER_INT_OFF</code>	Interrupt disabled

### Timer Width:

<code>T1_8BIT_RW</code>	8-bit mode
<code>T1_16BIT_RW</code>	16-bit mode

### Clock Source:

<code>T1_SOURCE_EXT</code>	External clock source (I/O pin)
<code>T1_SOURCE_INT</code>	Internal clock source (TOSC)

### Prescaler:

<code>T1_PS_1_1</code>	1:1 prescale
<code>T1_PS_1_2</code>	1:2 prescale
<code>T1_PS_1_4</code>	1:4 prescale
<code>T1_PS_1_8</code>	1:8 prescale

### Oscillator Use:

<code>T1_OSC1EN_ON</code>	Enable Timer1 oscillator
<code>T1_OSC1EN_OFF</code>	Disable Timer1 oscillator

### Synchronize Clock Input:

<code>T1_SYNC_EXT_ON</code>	Sync external clock input
<code>T1_SYNC_EXT_OFF</code>	Don't sync external clock input

**Remarks:** This macro configures timer1 according to the options specified and then enables it.

**Code Example:** `OpenTimer1(TIMER_INT_ON & T1_8BIT_RW & T1_SOURCE_EXT & T1_PS_1_1)`

# Proton+ PIC18 Timer Macros

## OpenTimer1 (V6)

For TMR\_V6

**Macro:** Configure and enable timer1.  
**Include:** `Timers.inc`  
**Prototype:** `OpenTimer1(pConfig , pConfig1)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file TimerDefs.inc.

### Enable Timer1 Interrupt:

<code>TIMER_INT_ON</code>	Interrupt enabled
<code>TIMER_INT_OFF</code>	Interrupt disabled

### Clock Source:

<code>T1_SOURCE_PINOSC</code>	Ext clock or Crystal oscillator
<code>T1_SOURCE_CAPOSC</code>	Capacitive sensing oscillator
<code>T1_SOURCE_FOSC_4</code>	Instruction cycle (FOSC/4)
<code>T1_SOURCE_FOSC</code>	System Clock (FOSC)

### Prescaler:

<code>T1_PS_1_1</code>	1:1 prescale
<code>T1_PS_1_2</code>	1:2 prescale
<code>T1_PS_1_4</code>	1:4 prescale
<code>T1_PS_1_8</code>	1:8 prescale

### Oscillator Use:

<code>T1_OSC1EN_ON</code>	Enable Timer1 oscillator
<code>T1_OSC1EN_OFF</code>	Disable Timer1 oscillator

### Synchronize Clock Input:

<code>T1_SYNC_EXT_ON</code>	Sync external clock input
<code>T1_SYNC_EXT_OFF</code>	Don't sync external clock input

### Timer Width:

<code>T1_8BIT_RW</code>	8-bit mode
<code>T1_16BIT_RW</code>	16-bit mode

# Proton+ PIC18 Timer Macros

## ***PConfig1:***

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file TimerDefs.inc.

### **Timer1 Gate Enable**

TIMER_GATE_ON	Counting is controlled by Timer1 gate
TIMER_GATE_OFF	Timer is always counting

### **Timer1 Gate Polarity**

TIMER_GATE_POL_HI	Gate is active-high
TIMER_GATE_POL_LO	Gate is active-low

### **Timer1 Gate Toggle Mode**

TIMER_GATE_TOGGLE_ON	Gate Toggle mode enabled
TIMER_GATE_TOGGLE_OFF	Gate Toggle mode disabled

### **Timer1 Gate One Shot Enable**

TIMER_GATE_1SHOT_ON	Gate one shot is enabled
TIMER_GATE_1SHOT_OFF	Gate one shot is disabled

### **Timer1 Gate Source Select**

TIMER_GATE_SRC_T1GPIN	Timer1 gate pin
TIMER_GATE_SRC_T0	Timer0 overflow output
TIMER_GATE_SRC_T2	Timer2 match PR2 output

### **Enable Timer1 Gate Interrupt:**

TIMER_GATE_INT_OFF	Interrupts disabled
TIMER_GATE_INT_ON	Interrupts enabled

**Remarks:** This macro configures timer1 according to the options specified and then enables it.

**Code Example:** `OpenTimer1(T1_SOURCE_PINOSC & T1_PS_1_1 & T1_8BIT_RW, _  
TIMER_GATE_ON & TIMER_GATE_TOGGLE_ON)`

# Proton+ PIC18 Timer Macros

## OpenTimer2

For TMR\_V2, TMR\_V3, TMR\_V4, TMR\_V5 and TMR\_V6

**Macro:** Configure and enable timer2.

**Include:** `Timers.inc`

**Prototype:** `OpenTimer2(pConfig)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file TimerDefs.inc.

### Enable Timer2 Interrupt:

<code>TIMER_INT_ON</code>	Interrupt enabled
<code>TIMER_INT_OFF</code>	Interrupt disabled

### Prescale Value:

<code>T2_PS_1_1</code>	1:1 prescale
<code>T2_PS_1_4</code>	1:4 prescale
<code>T2_PS_1_16</code>	1:16 prescale

### Postscale Value:

<code>T2_POST_1_1</code>	1:1 postscale
<code>T2_POST_1_2</code>	1:2 postscale
<code>T2_POST_1_3</code>	1:3 postscale
<code>T2_POST_1_4</code>	1:4 postscale
<code>T2_POST_1_5</code>	1:5 postscale
<code>T2_POST_1_6</code>	1:6 postscale
<code>T2_POST_1_7</code>	1:7 postscale
<code>T2_POST_1_8</code>	1:8 postscale
<code>T2_POST_1_9</code>	1:9 postscale
<code>T2_POST_1_10</code>	1:10 postscale
<code>T2_POST_1_11</code>	1:11 postscale
<code>T2_POST_1_12</code>	1:12 postscale
<code>T2_POST_1_13</code>	1:13 postscale
<code>T2_POST_1_14</code>	1:14 postscale
<code>T2_POST_1_15</code>	1:15 postscale
<code>T2_POST_1_16</code>	1:16 postscale

**Remarks:** This macro configures timer2 according to the options specified and then enables it.

**Code Example:** `OpenTimer2(TIMER_INT_OFF & T2_PS_1_1 & T2_POST_1_8 )`



# Proton+ PIC18 Timer Macros

## OpenTimer3 (V2, V4)

For TMR\_V2 and TMR\_V4

**Macro:** Configure and enable timer3.

**Include:** `Timers.inc`

**Prototype:** `OpenTimer3(pConfig)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file `TimerDefs.inc`.

### Enable Timer3 Interrupt:

<code>TIMER_INT_ON</code>	Interrupt enabled
<code>TIMER_INT_OFF</code>	Interrupt disabled

### Timer Width:

<code>T3_8BIT_RW</code>	8-bit mode
<code>T3_16BIT_RW</code>	16-bit mode

### Clock Source:

<code>T3_SOURCE_EXT</code>	External clock source (I/O pin)
<code>T3_SOURCE_INT</code>	Internal clock source (TOSC)

### Prescale Value:

<code>T3_PS_1_1</code>	1:1 prescale
<code>T3_PS_1_2</code>	1:2 prescale
<code>T3_PS_1_4</code>	1:4 prescale
<code>T3_PS_1_8</code>	1:8 prescale

### Synchronize Clock Input:

<code>T3_SYNC_EXT_ON</code>	Sync external clock input
<code>T3_SYNC_EXT_OFF</code>	Don't sync external clock input

**Remarks:** This macro configures timer3 according to the options specified and then enables it.

**Code Example:** `OpenTimer3(T3_8BIT_RW & T3_SOURCE_EXT & T3_PS_1_1 & T3_SYNC_EXT_OFF)`

# Proton+ PIC18 Timer Macros

## OpenTimer3 (V6)

For TMR\_V6

**Macro:** Configure and enable timer3.

**Include:** `Timers.inc`

**Prototype:** `OpenTimer3(pConfig , pConfig1)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file TimerDefs.inc.

### Enable Timer3 Interrupt:

<code>TIMER_INT_ON</code>	Interrupt enabled
<code>TIMER_INT_OFF</code>	Interrupt disabled

### Clock Source:

<code>T3_SOURCE_PINOSC</code>	Ext clock or Crystal oscillator
<code>T3_SOURCE_CAPOSC</code>	Capacitive sensing oscillator
<code>T3_SOURCE_FOSC_4</code>	Instruction cycle (FOSC/4)
<code>T3_SOURCE_FOSC</code>	System Clock (FOSC)

### Prescaler:

<code>T3_PS_1_1</code>	1:1 prescale
<code>T3_PS_1_2</code>	1:2 prescale
<code>T3_PS_1_4</code>	1:4 prescale
<code>T3_PS_1_8</code>	1:8 prescale

### Oscillator Use:

<code>T3_OSC1EN_ON</code>	Enable Timer1 oscillator
<code>T3_OSC1EN_OFF</code>	Disable Timer1 oscillator

### Synchronize Clock Input:

<code>T3_SYNC_EXT_ON</code>	Sync external clock input
<code>T3_SYNC_EXT_OFF</code>	Don't sync external clock input

### Timer Width:

<code>T3_8BIT_RW</code>	8-bit mode
<code>T3_16BIT_RW</code>	16-bit mode

# Proton+ PIC18 Timer Macros

## *PConfig1:*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file TimerDefs.inc.

### Timer3 Gate Enable

TIMER\_GATE\_ON    counting is controlled by Timer3 gate  
TIMER\_GATE\_OFF   Timer is always counting

### Timer3 Gate Polarity

TIMER\_GATE\_POL\_HI            Gate    is active-high  
TIMER\_GATE\_POL\_LO            Gate is active-low

### Timer3 Gate Toggle Mode

TIMER\_GATE\_TOGGLE\_ON        Gate Toggle mode enabled  
TIMER\_GATE\_TOGGLE\_OFF       Gate Toggle mode disabled

### Timer3 Gate One Shot Enable

TIMER\_GATE\_1SHOT\_ON         Gate one shot is enabled  
TIMER\_GATE\_1SHOT\_OFF        Gate one shot is disabled

### Timer3 Gate Source Select

TIMER\_GATE\_SRC\_T1GPIN       Timer1 gate pin  
TIMER\_GATE\_SRC\_T0            Timer0 overflow output  
TIMER\_GATE\_SRC\_T2            Timer2 match PR2    output

### Enable Timer3 Gate Interrupt:

TIMER\_GATE\_INT\_OFF          Interrupts disabled  
TIMER\_GATE\_INT\_ON            Interrupts enabled

**Remarks:**            This macro configures timer3 according to the options specified and then enables it.

**Code Example:** `OpenTimer3(TIMER_INT_ON & T3_PS_1_1 & T3_8BIT_RW, _  
                                 TIMER_GATE_ON & TIMER_GATE_POL_HI)`

# Proton+ PIC18 Timer Macros

## OpenTimer4

For TMR\_V4 and TMR\_V6

**Macro:** Configure and enable timer4.

**Include:** `Timers.inc`

**Prototype:** `OpenTimer4(pConfig)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file `TimerDefs.inc`.

### Enable Timer4 Interrupt:

`TIMER_INT_ON` Interrupt enabled

`TIMER_INT_OFF` Interrupt disabled

### Prescale Value:

`T4_PS_1_1` 1:1 prescale

`T4_PS_1_4` 1:4 prescale

`T4_PS_1_16` 1:16 prescale

### Postscale Value:

`T4_POST_1_1` 1:1 postscale

`T4_POST_1_2` 1:2 postscale

`T4_POST_1_3` 1:3 postscale

`T4_POST_1_4` 1:4 postscale

`T4_POST_1_5` 1:5 postscale

`T4_POST_1_6` 1:6 postscale

`T4_POST_1_7` 1:7 postscale

`T4_POST_1_8` 1:8 postscale

`T4_POST_1_9` 1:9 postscale

`T4_POST_1_10` 1:10 postscale

`T4_POST_1_11` 1:11 postscale

`T4_POST_1_12` 1:12 postscale

`T4_POST_1_13` 1:13 postscale

`T4_POST_1_14` 1:14 postscale

`T4_POST_1_15` 1:15 postscale

`T4_POST_1_16` 1:16 postscale

**Remarks:** This macro configures timer4 according to the options specified and then enables it.

**Code Example:** `OpenTimer4(TIMER_INT_OFF & T4_PS_1_1 & T4_POST_1_8)`

# Proton+ PIC18 Timer Macros

## OpenTimer5

For TMR\_V5

**Macro:** Configure and enable timer5.

**Include:** `Timers.inc`

**Prototype:** `OpenTimer5(pConfig, pT5pr)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file `TimerDefs.inc`.

*pT5pr*

pT5pr value will be loaded in to low and high byte of the Timer5 Period Register (i.e. `PR5L = pT5pr : PR5H = (pT5pr >> 8)`)

### Enable Timer5 Interrupt:

`TIMER_INT_ON` Interrupt enabled

`TIMER_INT_OFF` Interrupt disabled

### Timer5 Sleep Enable:

`T5_SLP_EN` Enable during sleep

`T5_SLP_DIS` Disable during sleep

### Special Event Reset:

`T5_SP_EVNT_REN` Special event reset enable

`T5_SP_EVNT_RDIS` Special event reset disable

### Timer Mode:

`T5_MD_SINGL_SHOT` Single-shot mode

`T5_MD_CONT_COUNT` Continuous count mode

### Prescale Value:

`T5_PS_1_1` 1:1 prescale

`T5_PS_1_2` 1:2 prescale

`T5_PS_1_4` 1:4 prescale

`T5_PS_1_8` 1:8 prescale

### Synchronize Clock Input:

`T5_EX_CLK_SYNC` Sync external clock input

`T5_EX_CLK_NOSYNC` Don't sync external clock input

### Clock Source:

`T5_CLK_EXTRN` External clock source

`T5_CLK_INT` Internal clock source

**Remarks:** This macro configures timer5 according to the options specified, enables it, and then writes the value in to Timer5 period register.

**Code Example:** `OpenTimer5(T5_MD_CONT_COUNT & T5_PS_1_1 & T5_CLK_INT , $C350)`

# Proton+ PIC18 Timer Macros

## ReadTimer

### ReadTimer0

For **TMR\_V1**, **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Reads the value of the timer0.

**Include:** `Timers.inc`

**Prototype:** `Var = ReadTimer0`

**Remarks:** This macro reads the value of the timer0 register.

`Timer0: TMR0L,TMR0H`

**Return Value:** The current value of the timer.

### ReadTimer1

For **TMR\_V1**, **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Reads the value of the timer1.

**Include:** `Timers.inc`

**Prototype:** `Var = ReadTimer1`

**Remarks:** This macro reads the value of the timer1 register.

`Timer1: TMR1L,TMR1H`

**Return Value:** The current value of the timer.

### ReadTimer2

For **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Reads the value of the timer2.

**Include:** `Timers.inc`

**Prototype:** `Var = ReadTimer2`

**Remarks:** This macro reads the value of the timer2 register.

`Timer2: TMR2`

**Return Value:** The current value of the timer.

### ReadTimer3

For **TMR\_V2**, **TMR\_V4** and **TMR\_V6**

**Macro:** Reads the value of the timer3.

**Include:** `Timers.inc`

**Prototype:** `Var = ReadTimer3`

**Remarks:** This macro reads the value of the timer3 register.

`Timer3: TMR3L,TMR3H`

**Return Value:** The current value of the timer.

## Proton+ PIC18 Timer Macros

### ReadTimer4

For **TMR\_V4** and **TMR\_V6**

**Macro:** Reads the value of the timer4.

**Include:** `Timers.inc`

**Prototype:** `Var = ReadTimer4`

**Remarks:** This macro reads the value of the timer4 register.

`Timer4: TMR4`

**Return Value:** The current value of the timer.

### ReadTimer5

For **TMR\_V5**

**Macro:** Reads the value of the timer5.

**Include:** `Timers.inc`

**Prototype:** `Var = ReadTimer5`

**Remarks:** This macro reads the value of the timer4 register.

`Timer5: TMR5L,TMR5H`

**Return Value:** The current value of the timer.

# Proton+ PIC18 Timer Macros

## WriteTimer

### WriteTimer0

For **TMR\_V1**, **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Write a value into the timer0.

**Include:** `Timers.inc`

**Prototype:** `WriteTimer0(pTimer)`

**Arguments:** *pTimer*

The value that will be loaded into the specified timer.

**Remarks:** This macro writes a value to the respective timer register:

Timer0: `TMR0L, TMR0H`

**Code Example:** `WriteTimer0(10000)`

### WriteTimer1

For **TMR\_V1**, **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Write a value into the timer1.

**Include:** `Timers.inc`

**Prototype:** `WriteTimer1(pTimer)`

**Arguments:** *pTimer*

The value that will be loaded into the specified timer.

**Remarks:** This macro writes a value to the respective timer register:

Timer1: `TMR1L, TMR1H`

**Code Example:** `WriteTimer1(10000)`

### WriteTimer2

For **TMR\_V2**, **TMR\_V3**, **TMR\_V4**, **TMR\_V5** and **TMR\_V6**

**Macro:** Write a value into the timer2.

**Include:** `Timers.inc`

**Prototype:** `WriteTimer2(pTimer)`

**Arguments:** *pTimer*

The value that will be loaded into the specified timer.

**Remarks:** This macro writes a value to the respective timer register:

Timer2: `TMR2`

**Code Example:** `WriteTimer2(100)`



# Proton+ PIC18 Timer Macros

## WriteTimer3

For **TMR\_V2**, **TMR\_V4** and **TMR\_V6**

**Macro:** Write a value into the timer3.

**Include:** `Timers.inc`

**Prototype:** `WriteTimer3(pTimer)`

**Arguments:** *pTimer*

The value that will be loaded into the specified timer.

**Remarks:** This macro writes a value to the respective timer register:

Timer3: TMR3L, TMR3H

**Code Example:** `WriteTimer3(10000)`

## WriteTimer4

For **TMR\_V4** and **TMR\_V6**

**Macro:** Write a value into the timer4.

**Include:** `Timers.inc`

**Prototype:** `WriteTimer4(pTimer)`

**Arguments:** *pTimer*

The value that will be loaded into the specified timer.

**Remarks:** This macro writes a value to the respective timer register:

Timer4: TMR4

**Code Example:** `WriteTimer0(100)`

## WriteTimer5

For **TMR\_V5**

**Macro:** Write a value into the timer5.

**Include:** `Timers.inc`

**Prototype:** `WriteTimer5(pTimer)`

**Arguments:** *pTimer*

The value that will be loaded into the specified timer.

**Remarks:** This macro writes a value to the respective timer register:

Timer5: TMR5L, TMR5H

**Code Example:** `WriteTimer5(1000)`

# Proton+ PIC18 Timer Macros

## SetTmrCCPSrc

### SetTmrCCPSrc

For **TMR\_V2**, **TMR\_V4** and **TMR\_V6**

**Macro:** Enable timer source to CCP module.

**Include:** `Timers.inc`

**Prototype:** `SetTmrCCPSrc(pConfig)`

**Arguments:** *pConfig*

A bitmask that is created by performing either a bitwise AND operation ('&'), which is user configurable, with a value from each of the categories listed below. These values are defined in the file `TimerDefs.inc`.

#### **TMR\_V2:**

<code>T3_SOURCE_CCP</code>	Timer3 source for both CCP's
<code>T1_CCP1_T3_CCP2</code>	Timer1 source for CCP1 and Timer3 source for CCP2
<code>T1_SOURCE_CCP</code>	Timer1 source for both CCP's

#### **TMR\_V4:**

<code>T34_SOURCE_CCP</code>	Timer3 and Timer4 are sources for all CCP's
<code>T12_CCP12_T34_CCP345</code>	Timer1 and Timer2 are sources for CCP1 and CCP2 and Timer3 and Timer4 are sources for CCP3 through CCP5
<code>T12_CCP1_T34_CCP2345</code>	Timer1 and Timer2 are sources for CCP1 and Timer3 and Timer4 are sources for CCP2 through CCP5
<code>T12_SOURCE_CCP</code>	Timer1 and Timer2 are sources for all CCP's

#### **TMR\_V6:**

<code>T34_SOURCE_CCP12</code>	Timer1 and 2 source for both CCP's
<code>T12_CCP1_T34_CCP2</code>	Timer1 and Timer2 are sources for CCP1 and Timer3 and Timer4 are sources CCP2
<code>T12_SOURCE_CCP</code>	Timer1 and Timer2 are sources for all CCP's

**Remarks:** This macro configures the timer as a clock source for the CCP module.

**Code Example:** `SetTmrCCPSrc(T34_SOURCE_CCP12)`

# Proton+ PIC18 Timer Macros

## Example Use of the Timer Macros:

```
'
' Timer Macro Demo
'
' This can be simulated using the "PIC18_ALCD_VHB" circuit within the ISIS simulator
'
  Include "Proton18_4.Inc"          ' Use the Proton Development board for the demo
  Optimiser_Level = 3
  Dead_Code_Remove = On

  Include "Timers.Inc"             ' Load the Timer Macros into the program

  Dim Result As Word

' Configure Timer0
  OpenTimer0(TIMER_INT_OFF & T0_SOURCE_INT & T0_PS_1_32 & T0_16BIT)

  HRSOut "Press a Key\r"
  While 1 = 1
    While InKey = 16 : Wend        ' Wait for a Keypress on the keypad
    Result = ReadTimer0            ' Read Timer0
    WriteTimer0(0)                 ' Reset Timer0
    HRSOut "Timer0 Value = ", Dec Result,13 ' Display the value of Timer0
    While InKey <> 16 : Wend        ' Wait for the key to released
    DelayMS 50
  Wend
  CloseTimer0                     ' Close modules
  Stop
```

## Proton+ PIC18 Timer Macros

### T3\_OSC1EN\_ON

For **TMR\_V1**, **TMR\_V2** and **TMR\_V4**

**Macro:** Enables Oscillator as external source

**Include:** `Timers.inc`

**Prototype:** **T3\_OSC1EN\_ON**

**Remarks:** This Macro enables the oscillator associated with Timer1 as source of external clock input for Timer3.

### T3\_OSC1EN\_OFF

For **TMR\_V1**, **TMR\_V2** and **TMR\_V4**

**Macro:** Disables Oscillator as external source

**Include:** `Timers.inc`

**Prototype:** **T3\_OSC1EN\_OFF**

**Remarks:** This Macro disables the oscillator associated with Timer1 and selects the signal on pin T13CKI as source of external clock input for Timer3.