

PWM Library

mikroBasic PRO for PIC Libraries > Hardware Libraries >



PWM Library

CCP module is available with a number of PIC MCUs. mikroBasic PRO for PIC provides library which simplifies using PWM HW Module.



Important : Some MCUs have multiple CCP modules. In order to use the desired CCP library routine, simply change the number **1** in the prototype with the appropriate module number, i.e. `PWM2_Start();`.

Library Routines

- [PWM1_Init](#)
- [PWM1_Set_Duty](#)
- [PWM1_Start](#)
- [PWM1_Stop](#)

PWM1_Init

Prototype	<code>sub procedure PWM1_Init(const freq as longint)</code>
Returns	Nothing.
Description	<p>Initializes the PWM module with duty ratio 0. Parameter <code>freq</code> is a desired PWM frequency in Hz (refer to device data sheet for correct values in respect with F_{osc}).</p> <p>This routine needs to be called before using other functions from PWM Library.</p>
Requires	<p>MCU must have CCP module.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> Note : Calculation of the PWM frequency value is carried out by the compiler, as it would produce a relatively large code if performed on the library level. Therefore, compiler needs to know the value of the parameter in the compile time. That is why this parameter needs to be a constant, and not a variable.</p> </div>
Example	<p>Initialize PWM module at 5KHz:</p> <pre>PWM1_Init(5000)</pre>

PWM1_Set_Duty

Prototype	<code>sub procedure PWM1_Set_Duty(dim duty_ratio as byte)</code>
Returns	Nothing.
Description	<p>Sets PWM duty ratio. Parameter <code>duty</code> takes values from 0 to 255, where 0 is 0%, 127 is 50%, and 255 is 100% duty ratio. Other specific values for duty ratio can be calculated as $(Percent * 255) / 100$.</p>
Requires	<p>MCU must have CCP module. PWM1_Init must be called before using this routine.</p>
Example	<p>Set duty ratio to 75%:</p>

	PWM1_Set_Duty(192)
--	--------------------

PWM1_Start

Prototype	<code>sub procedure PWM1_Start()</code>
Returns	Nothing.
Description	Starts PWM.
Requires	MCU must have CCP module. PWM1_Init must be called before using this routine.
Example	<code>PWM1_Start()</code>

PWM1_Stop

Prototype	<code>sub procedure PWM1_Stop()</code>
Returns	Nothing.
Description	Stops PWM.
Requires	MCU must have CCP module. PWM1_Init must be called before using this routine. PWM1_Start should be called before using this routine, otherwise it will have no effect as the PWM module is not running.
Example	<code>PWM1_Stop()</code>

Library Example

The example changes PWM duty ratio on pin PB3 continually. If LED is connected to PB3, you can observe the gradual change of emitted light.

```

program PWM_Test

dim current_duty, current_duty1, old_duty, old_duty1 as byte

sub procedure InitMain()
    ANSEL = 0           ' Configure AN pins as digital I/O
    ANSELH = 0

    PORTA = 255
    TRISA = 255        ' configure PORTA pins as input
    PORTB = 0          ' set PORTB to 0
    TRISB = 0          ' designate PORTB pins as output
    PORTC = 0          ' set PORTC to 0
    TRISC = 0          ' designate PORTC pins as output
    PWM1_Init(5000)    ' Initialize PWM1 module at 5KHz
    PWM2_Init(5000)    ' Initialize PWM2 module at 5KHz
end sub

main:
    InitMain()
    current_duty = 16  ' initial value for current_duty
    current_duty1 = 16 ' initial value for current_duty1

```

```
PWM1_Start()           ' start PWM1
PWM2_Start()           ' start PWM2
PWM1_Set_Duty(current_duty) ' Set current duty for PWM1
PWM2_Set_Duty(current_duty1) ' Set current duty for PWM2

while (TRUE)           ' endless loop
  if (RA0_bit <> 0) then ' button on RA0 pressed
    Delay_ms(40)
    Inc(current_duty)    ' increment current_duty
    PWM1_Set_Duty(current_duty)
  end if

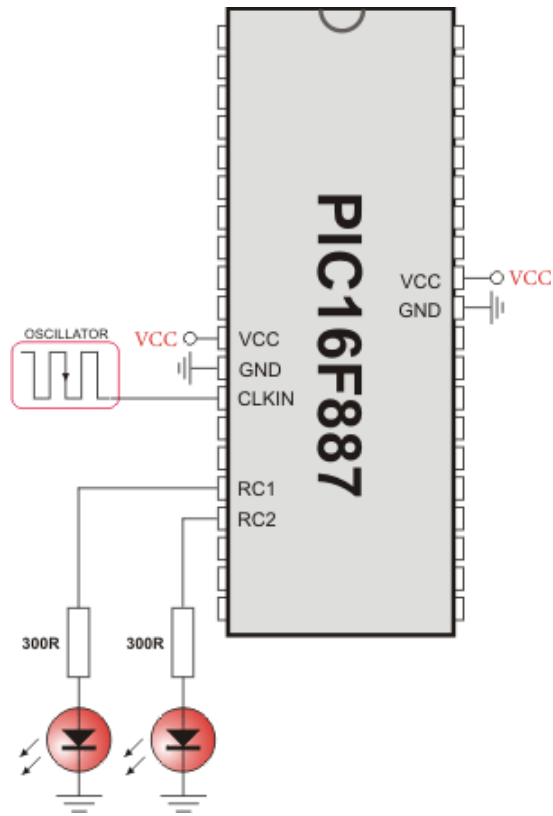
  if (RA1_bit <> 0) then ' button on RA1 pressed
    Delay_ms(40)
    Dec(current_duty)    ' decrement current_duty
    PWM1_Set_Duty(current_duty)
  end if

  if (RA2_bit <> 0) then ' button on RA2 pressed
    Delay_ms(40)
    Inc(current_duty1)   ' increment current_duty1
    PWM2_Set_Duty(current_duty1)
  end if

  if (RA3_bit <> 0) then ' button on RA3 pressed
    Delay_ms(40)
    Dec(current_duty1)   ' decrement current_duty1
    PWM2_Set_Duty(current_duty1)
  end if

  Delay_ms(5)           ' slow down change pace a little
wend
end.
```

HW Connection



PWM demonstration

Copyright (c) 2002-2010 mikroElektronika. All rights reserved.
 What do you think about this topic ? [Send us feedback!](#)