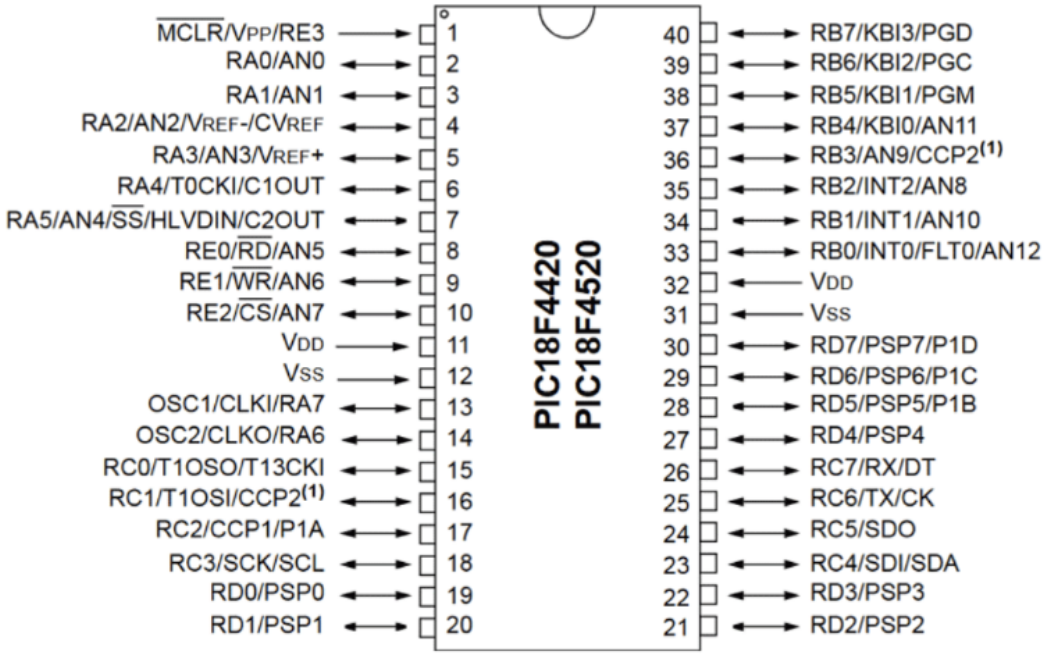
Lab08 CCP Module

Lab08 CCP Module

Each Capture/Compare/PWM module is associated with a control register (generically, CCPxCON) and a data register (CCPRx).

- PIC18F4520 Datasheet：
  https://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf
  (https://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf)



Timer

TABLE 15-1: CCP MODE – TIMER RESOURCE

CCP/ECCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

REGISTER 14-1: T3CON: TIMER3 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6,3 **T3CCP<2:1>:** Timer3 and Timer1 to CCPx Enable bits
1x = Timer3 is the capture/compare clock source for the CCP modules
01 = Timer3 is the capture/compare clock source for CCP2;
Timer1 is the capture/compare clock source for CCP1
00 = Timer1 is the capture/compare clock source for the CCP modules

bit 5-4 **T3CKPS<1:0>**: Timer3 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 2 **T3SYNC:** Timer3 External Clock Input Synchronization Control bit
(Not usable if the device clock comes from Timer1/Timer3.)
When TMR3CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

Mode Selection

REGISTER 16-1: CCP1CON: ECCP CONTROL REGISTER (40/44-PIN DEVICES)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **P1M<1:0>:** Enhanced PWM Output Configuration bits

If CCP1M3:CCP1M2 = 00, 01, 10:

xx = P1A assigned as capture/compare input/output; P1B, P1C, P1D assigned as port pins

If CCP1M3:CCP1M2 = 11:

00 = Single output, P1A modulated; P1B, P1C, P1D assigned as port pins
01 = Full-bridge output forward, P1D modulated; P1A active; P1B, P1C inactive
10 = Half-bridge output, P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins
11 = Full-bridge output reverse, P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B<1:0>:** PWM Duty Cycle bit 1 and bit 0

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

bit 3-0 **CCP1M<3:0>:** Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCP module)
0001 = Reserved
0010 = Compare mode, toggle output on match
0011 = Capture mode
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, initialize CCP1 pin low; set output on compare match (set CCP1IF)
1001 = Compare mode, initialize CCP1 pin high; clear output on compare match (set CCP1IF)
1010 = Compare mode, generate software interrupt only, CCP1 pin reverts to I/O state
1011 = Compare mode, trigger special event (ECCP resets TMR1 or TMR3, sets CCP1IF bit)
1100 = PWM mode, P1A, P1C active-high; P1B, P1D active-high
1101 = PWM mode, P1A, P1C active-high; P1B, P1D active-low
1110 = PWM mode, P1A, P1C active-low; P1B, P1D active-high
1111 = PWM mode, P1A, P1C active-low; P1B, P1D active-low

Capture mode

In capture mode, we can get the value in the timer if there is an input. That is, we can measure the arrival time of an event.

CCP Pin Configuration

In Capture mode, the appropriate CCPx pin should be configured as an **input** by setting the corresponding **TRIS** direction bit.

Event definition

We capture 16-bits value of the TMR1 or TMR3 register when an event occurs on the corresponding CCPx pin.

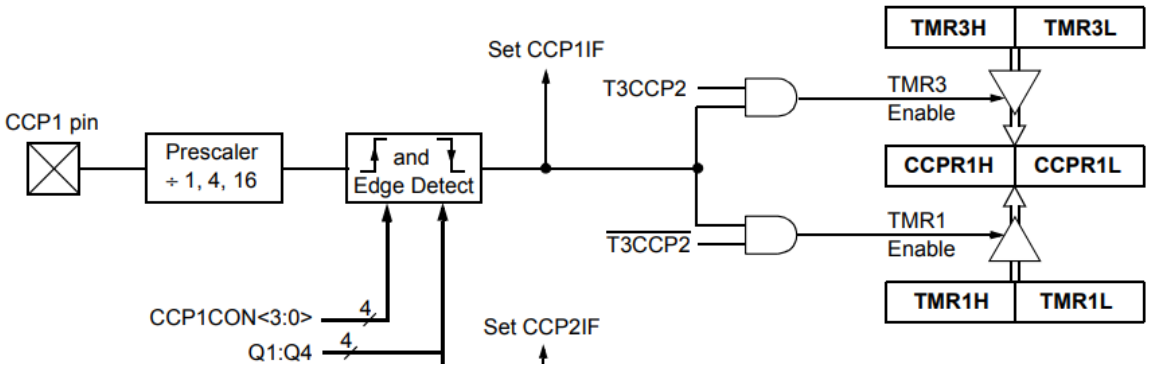
An event is defined as one of the following:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

Interrupt

When a capture is made, the interrupt request flag bit, CCPxIF, is set; it must be cleared in software. If another capture occurs before the value in register CCPRx is read, the old captured value is overwritten by the new captured value.

Capture Mode Operation Block Diagram



- T3CON is used for selecting the timers of the CCP modules.
- When the rising or falling edge is detected at CCP1 pin, the interupt flag CCP1F is set. It should be cleared through software.
- When the interupt occur, the value of the timer will be copied into register CCPR1.

Compare mode

When a match occurs, the CCPx pin can be:

- driven high
- driven low
- toggled (high-to-low or low-to-high)
- remain unchanged (that is, reflects the state of the I/O latch)

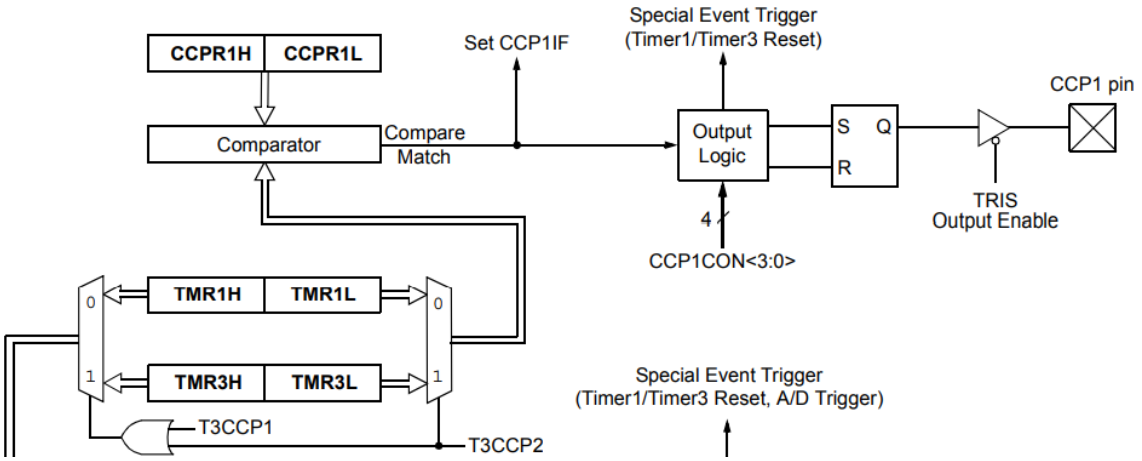
CCP Pin Configuration

In Compare mode, the appropriate CCPx pin should also be configured as an **output** by setting the corresponding **TRIS** direction bit.

Interrupt

The action on the pin is based on the value of the modeselect bits (CCPxM<3:0>). At the same time, the interrupt flag bit, CCPxIF, is set.

Compare Mode Operation Block Diagram

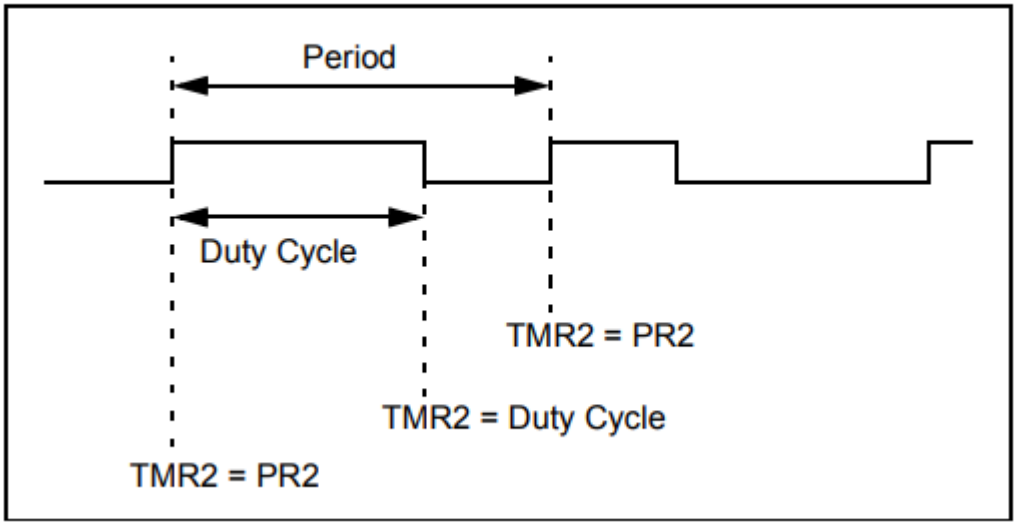


- T3CON is used for selecting the timers of the CCP modules.
- The value of register CCPR1 is constanly compared with the timer.
- When these values are the same, the CCP1IF is set and the output of CCP1 pin will be changed.

PWM mode

PWM is called Pulse-width modulation. The output voltage is determined by the rate between the length of **duty cycle** and **period**. We use PWM mode in various field, such as Breathing light or Servomotor.

FIGURE 15-4: PWM OUTPUT



PWM Period

- **The PWM period is specified by writing to the PR2register**

- $$\text{PWM Period } = [(\text{PR2}) + 1] \times 4 \times Tosc \times \text{ (TMR2 Prescale Value)}$$

  PWM frequency is defined as $\frac{1}{[\text{ PWM period }]}$.

  When TMR2 is equal to PR2, the following three events occur on the next increment cycle :

  – TMR2 is cleared
  – The CCPx pin is set (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
  – The PWM duty cycle is latched from CCPRxL into CCPRxH
  The Timer2 postscalers are not used in the determination of the PWM frequency.

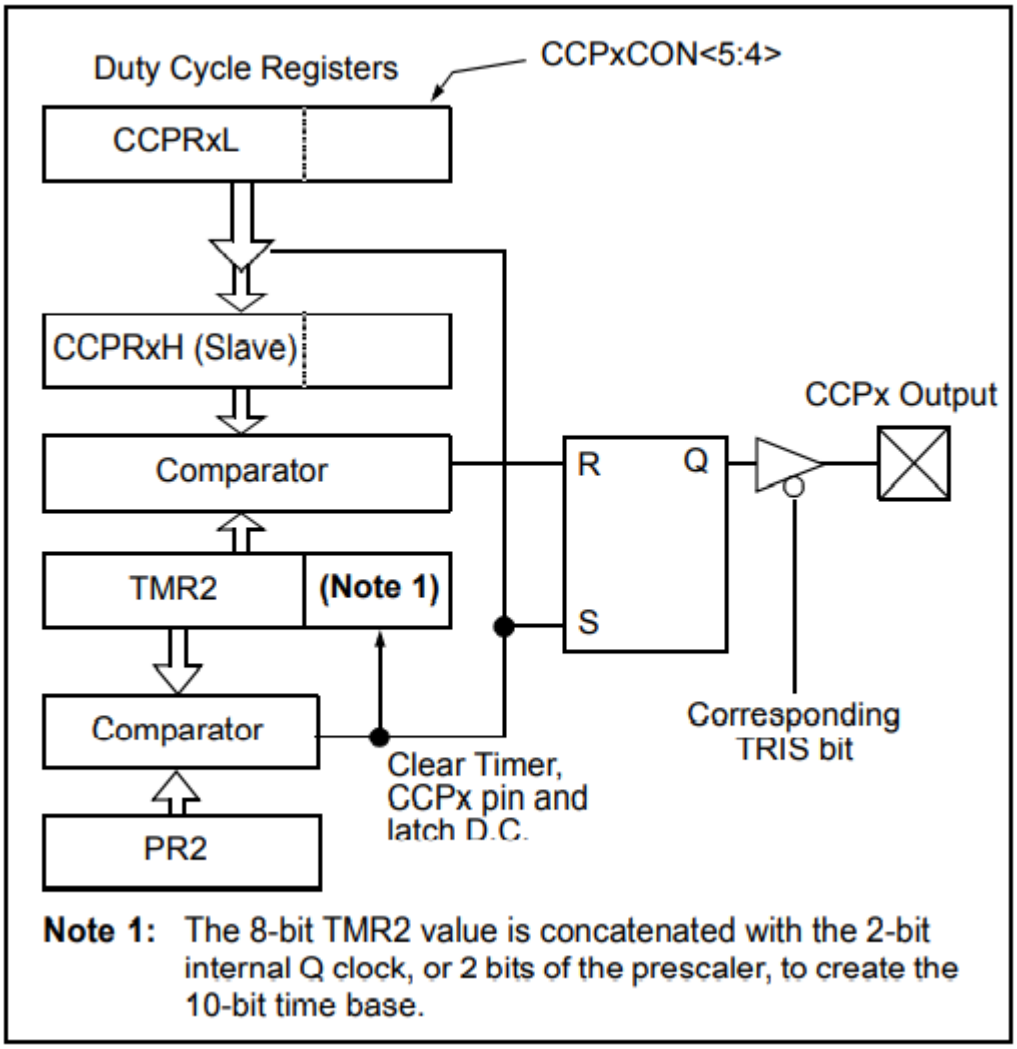
PWM Duty Cycle

- **The PWM duty cycle is specified by writing to the CCPRxL register and to the CCPxCON<5:4> bits. Up to 10-bit resolution is available.**

  $$\text{PWM Duty Cycle} = (\text{CCPRXL:CCPXCON<5:4>}) \times Tosc \times \text{ (TMR2 Prescale Value)}$$

  CCPRxL and CCPxCON<5:4> can **be written to at anytime** , but the duty cycle value is not latched into CCPRxH until after a match between PR2 and TMR2occurs (i.e., the period is complete). In PWM mode, **CCPRxH is a read-only register.**

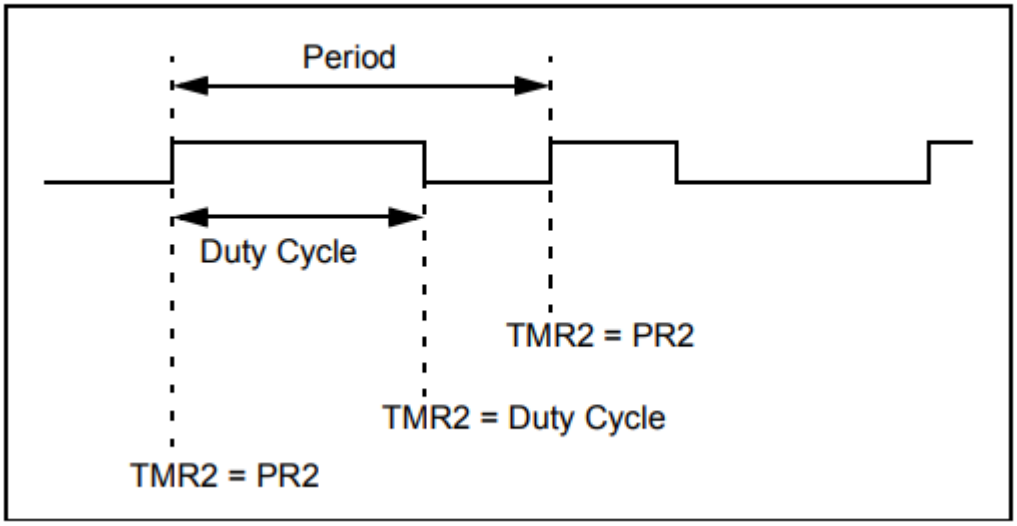
How PIC18 Run PWM mode



- Load the period value (in PR2) and duty cycle (in CCPRxL and CCPxCON<5:4>) and initiate the CCPx pin as an Output.
- Move the value from CCPRxL and CCPxCON<5:4> to CCPRxH.
- Configure T2CON and TMR2, set TMR2 to 0, and start the Timer2. Set the output to high.
  (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
- TMR2 is compared with CCPRxH. If they are the same, the output pin goes low.

- When TMR2 and PR2 are the same, run the whole process again.

FIGURE 15-4: PWM OUTPUT



Setup for PWM Operation

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCPx pin an output by clearing the appropriate TRIS bit. e.g. TRISC for CCP1.
4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.

5. Configure the CCPx module for PWM operation.

REGISTER 16-1: CCP1CON: ECCP CONTROL REGISTER (40/44-PIN DEVICES)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **P1M<1:0>:** Enhanced PWM Output Configuration bits

If CCP1M3:CCP1M2 = 00, 01, 10:

xx = P1A assigned as capture/compare input/output; P1B, P1C, P1D assigned as port pins

If CCP1M3:CCP1M2 = 11:

00 = Single output, P1A modulated; P1B, P1C, P1D assigned as port pins
01 = Full-bridge output forward, P1D modulated; P1A active; P1B, P1C inactive
10 = Half-bridge output, P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins
11 = Full-bridge output reverse, P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B<1:0>:** PWM Duty Cycle bit 1 and bit 0

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

bit 3-0 **CCP1M<3:0>:** Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCP module)
0001 = Reserved
0010 = Compare mode, toggle output on match
0011 = Capture mode
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, initialize CCP1 pin low; set output on compare match (set CCP1IF)
1001 = Compare mode, initialize CCP1 pin high; clear output on compare match (set CCP1IF)
1010 = Compare mode, generate software interrupt only, CCP1 pin reverts to I/O state
1011 = Compare mode, trigger special event (ECCP resets TMR1 or TMR3, sets CCP1IF bit)
1100 = PWM mode, P1A, P1C active-high; P1B, P1D active-high
1101 = PWM mode, P1A, P1C active-high; P1B, P1D active-low
1110 = PWM mode, P1A, P1C active-low; P1B, P1D active-high
1111 = PWM mode, P1A, P1C active-low; P1B, P1D active-low

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

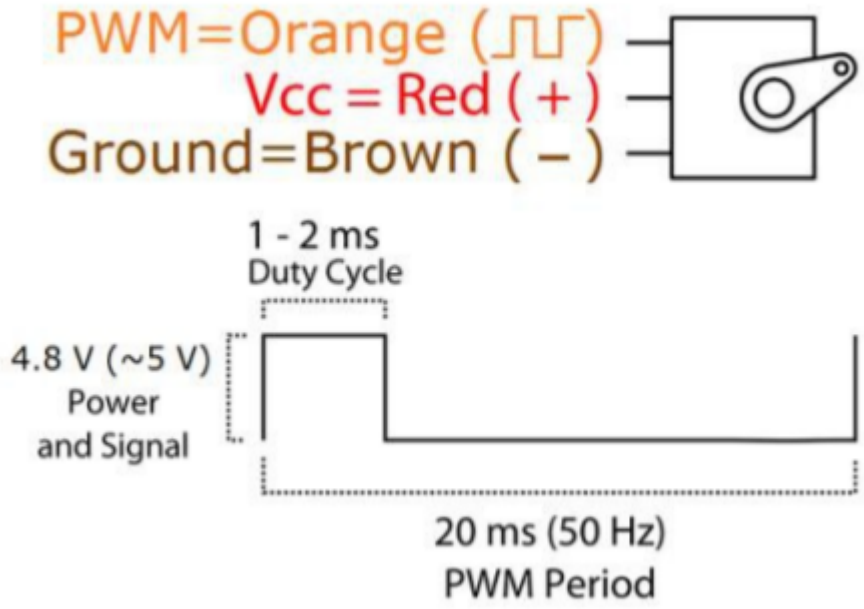
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

Tower Pro SG90 specs

- Modulation: Analog
- Torque: 4.8V
- Speed: 0.12 sec/60°
- Motor Type: 3-pole
- Gear Type: Plastic
- Rotation/Support: Bushing
- Pulse Width: 500 ~ 2400 μs (-90° ~ 90°, 1450 μs → 0°)
- Connector Type: JR



The motor is fragile. Please use it carefully.

Demo Code

The following code will rotate the servomotor into 0 degree.

```c
#include <xc.h>
#include <pic18f4520.h>

#pragma config OSC = INTIO67    // Oscillator Selection bits
#pragma config WDT = OFF        // Watchdog Timer Enable bit
#pragma config PWRT = OFF       // Power-up Enable bit
#pragma config BOREN = ON       // Brown-out Reset Enable bit
#pragma config PBADEN = OFF     // Watchdog Timer Enable bit
#pragma config LVP = OFF        // Low Voltage (single -supply) In-Circut
#pragma config CPD = OFF        // Data EEPROM  Memory Code Protection bi

void main(void)
{
    // Timer2 -> On, prescaler -> 4
    T2CONbits.TMR2ON = 0b1;
    T2CONbits.T2CKPS = 0b01;

    // Internal Oscillator Frequency, Fosc = 125 kHz, Tosc = 8 µs
    OSCCONbits.IRCF = 0b001;

    // PWM mode, P1A, P1C active-high; P1B, P1D active-high
    CCP1CONbits.CCP1M = 0b1100;

    // CCP1/RC2 -> Output
    TRISC = 0;
    LATC = 0;

    // Set up PR2, CCP to decide PWM period and Duty Cycle
    /**
     * PWM period
     * = (PR2 + 1) * 4 * Tosc * (TMR2 prescaler)
     * = (0x9b + 1) * 4 * 8µs * 4
     * = 0.019968s ~= 20ms
     */
    PR2 = 0x9b;

    /**
     * Duty cycle
     * = (CCPR1L:CCP1CON<5:4>) * Tosc * (TMR2 prescaler)
     * = (0x0b*4 + 0b01) * 8µs * 4
     * = 0.00144s ~= 1450µs
     */
    CCPR1L = 0x0b;
    CCP1CONbits.DC1B = 0b01;

    while(1);
    return;
}
```