PWM Tips and Tricks

Pulse-Width Modulation (PWM) Tips and Tricks

Introduction

The 16-bit PWM module features built-in comparison functionality and dual outputs that are individually controllable. The justification and duty cycle of each PWM output can be controlled individually, which can be advantageous for many different types of PWM applications. This document describes some of the basic configuration modes that this PWM module can be configured to operate with.

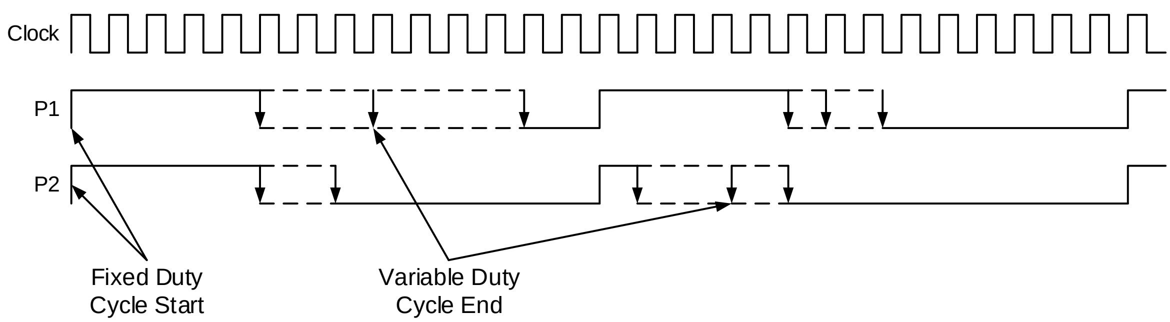
Table of Contents

1. Left-Aligned Mode

In Left-Aligned mode, the active part of the PWM signal is at the start of the period. As an example, if the PWM module has been configured to have a left-aligned output with a 75% duty cycle, the output will be active for the first 75% of the period and then transition to inactive for the remaining 25% of the period.

To configure the PWM module to generate a left-aligned output, the MODE bits of the PWM Configuration registers are used to select the output alignment mode and the P1 or P2 parameter registers are used to determine the resulting period of the output signal. The code example below shows an initialization routine that can be used to generate two left-aligned PWM outputs. The P1 output is a left-aligned PWM output with a 75% duty cycle and the P2 output shows a left-aligned PWM output with a 50% duty cycle. The figure below shows the left-aligned PWM output signals side-by-side, along with the corresponding PWM clock.

Figure 1-1. Left-Aligned Mode



Example 1-1. PWM Left-Aligned Initialization

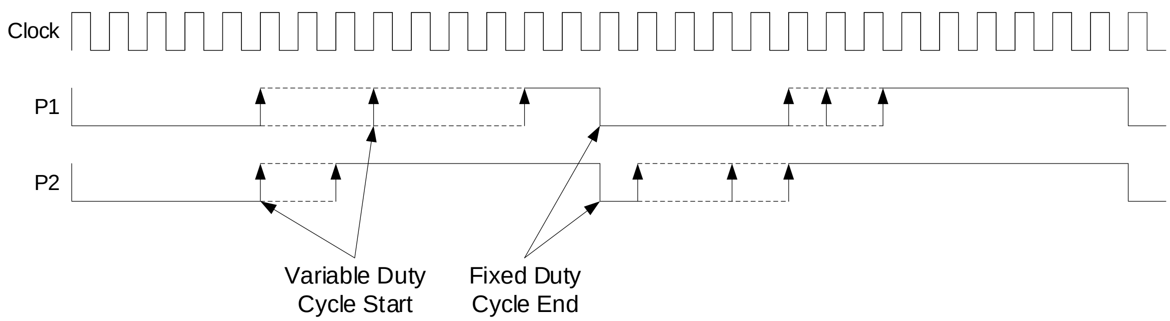
```
void PWM1_Initialize(void)
{
    PWM1ERS = 0x00;     // PWMERS External Reset Disabled;
    PWM1CLK = 0x02;     // PWMCLK FOSC;
    PWM1LDS = 0x00;     // PWMLDS Autoload disabled;
    PWM1PRL = 0x0F;     // PWMPRL 15;
    PWM1PRH = 0x00;     // PWMPRH 0;
    PWM1CPRE = 0x00;    // PWMCPRE No prescale;
    PWM1PIPOS = 0x00;   // PWMPIPOS No postscale;
    PWM1GIR = 0x00;     // PWMS1P2IF PWM2 output match did not occur;
                        // PWMS1P1IF PWM1 output match did not occur;
    PWM1GIE = 0x00;     // PWMS1P2IE disabled; PWMS1P1IE disabled;
    PWM1S1CFG = 0x00;   // PWMPOL2 disabled; PWMPOL1 disabled; PWMPPEN
                        // disabled; PWMMODE PWMOUT1,PWMOUT2 in left
                        // aligned mode
    PWM1S1P1L = 0x0C;   // PWMS1P1L 12;
    PWM1S1P1H = 0x00;   // PWMS1P1H 0;
    PWM1S1P2L = 0x08;   // PWMS1P2L 8;
    PWM1S1P2H = 0x00;   // PWMS1P2H 0;
    PWM2CON = 0x80;     // PWMEN enabled; PWMLD disabled; PWMERSPOL
                        // disabled; PWMERSNOW disabled;
}
```

2. Right-Aligned Mode

In Right-Aligned mode, the active part of the PWM signal is at the end of the period. As an example, If the PWM module has been configured to have a right-aligned output with a 75% duty cycle, the output will be inactive for the first 25% of the period and then transition to active for the remaining 75% of the period.

To configure the PWM module to generate a right-aligned output, the MODE bits of the PWM Configuration registers are used to select the output alignment mode and the P1 or P2 parameter registers are used to determine the resulting period of the output signal. The code example below shows an initialization routine that can be used to generate two right-aligned PWM outputs. The P1 output is a right-aligned PWM output with a 75% duty cycle and the P2 output shows a right-aligned PWM output with a 50% duty cycle. The figure below shows the right-aligned PWM output signals side-by-side, along with the corresponding PWM clock.

Figure 2-1. Right-Aligned Mode



Example 2-1. PWM Right Align Initialization

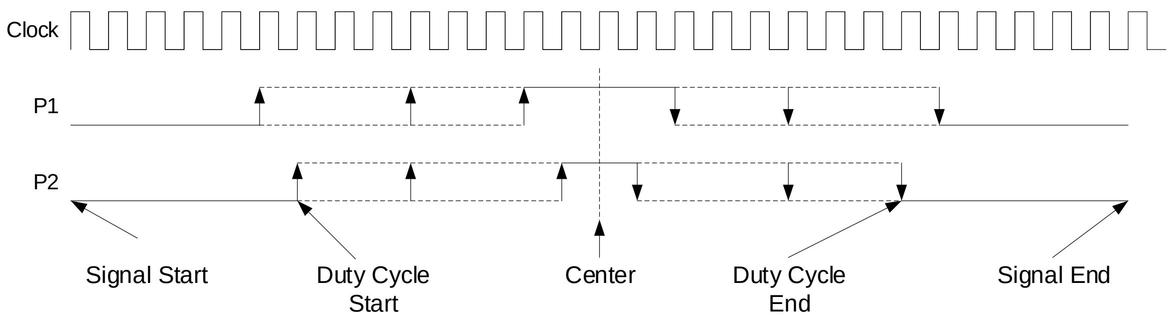
```
void PWM1_Initialize_RightAlign(void)
{
    PWM1ERS = 0x00;          // External Reset Disabled;
    PWM1CLK = 0x02;          // FOSC;
    PWM1LDS = 0x00;          // Autoload disabled;
    PWM1PRL = 0x0F;          // PWMPRL 15;
    PWM1PRH = 0x00;          // PWMPRH 0;
    PWM1CPRE = 0xF9;         // Prescale by 250;
    PWM1PIPOS = 0x00;        // PWMPIPOS No postscale;
    PWM1GIR = 0x00;          // PWMS1P2IF PWM2 output match did not occur;
                             // PWMS1P1IF PWM1 output match did not occur;
    PWM1S1CFG = 0x01;        // PWMPOL2 disabled; PWMPOL1 disabled; PWMPPEN
                             // disabled; PWMMODE PWMOUT1 and PWMOUT2 in
                             // right-aligned mode;
    PWM1S1P1L = 0x0C;        // PWMS1P1L 12;
    PWM1S1P1H = 0x00;        // PWMS1P1H 0;
    PWM1S1P2L = 0x08;        // PWMS1P2L 8;
    PWM1S1P2H = 0x00;        // PWMS1P2H 0;
    PWM1GIRbits.S1P1IF = 0;  // Clear PWM1 slice 1, output 1 interrupt flag
    PWM1GIRbits.S1P2IF = 0;  // Clear PWM1 slice 1, output 2 interrupt flag
    PWM1CON = 0x80;          // PWMEN enabled; PWMLD disabled; PWMERSPOL
                             // disabled; PWMERSNOW disabled;
}
```

3. Center-Aligned Mode

In Center-Aligned mode, the active part of the PWM signal is in the middle of the signal. A typical use case is using PWM to drive LEDs to prevent sudden current draw changes when all of the LEDs switch to active at once. In this mode, the effective frequency is half of the configured value. As an example, if the PWM module has been configured to have a center-aligned output with a 50% duty cycle, the output grows symmetrically from the center for an active time of 50% and an inactive time of 25% on each side.

To configure the PWM module to generate a center-aligned output, the MODE bits of the PWM Configuration registers are used to select the output alignment and the P1/P2 parameter registers are used to determine the resulting period of the output signal. The code example below shows an initialization routine that can be used to generate a center-aligned PWM output. The P1 output is a center-aligned PWM output with a 75% duty cycle and the P2 output shows a center-aligned PWM output with a 50% duty cycle. The figure below shows the center-aligned PWM output signals side-by-side, along with the corresponding PWM clock.

Figure 3-1. Center-Aligned Mode



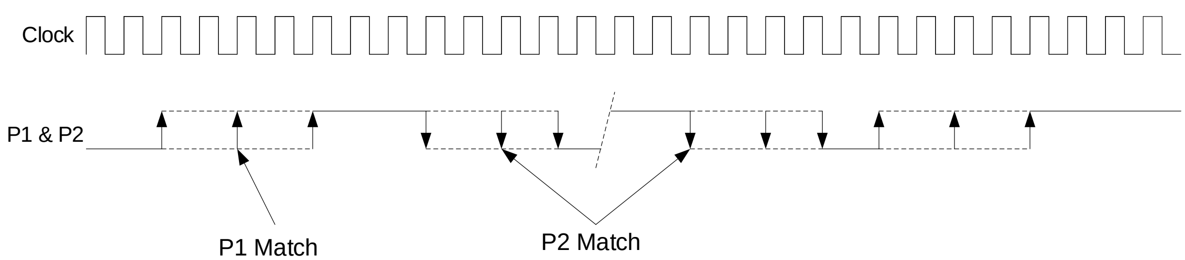
Example 3-1. PWM Center-Aligned Initialization

```
void PWM1_Initialize(void)
{
    PWM1ERS   = 0x00;   //PWMERS External Reset Disabled;
    PWM1CLK   = 0x02;   //PWMCLK FOSC;
    PWM1LDS   = 0x00;   //PWMLDS Autoload disabled;
    PWM1PRL   = 0xFF;   //PWMPRL 255;
    PWM1PRH   = 0x7C;   //PWMPRH 124;
    PWM1CPRE  = 0x00;   //PWMCPRE No prescale;
    PWM1PIPOS = 0x00;   //PWMPIPOS No postscale;
    PWM1GIR   = 0x00;   //PWMS1P2IF PWM2 output match did not occur;
                        //PWMS1P1IF PWM1 output match did not occur;
    PWM1S1CFG = 0x02;   //PWMPOL2 disabled; PWMPOL1 disabled; PWMPPEN
                        //disabled; PWMMODE PWMOUT1 and PWMOUT2 in center
                        //aligned mode;
    PWM1S1P1L = 0xC0;   //PWMS1P1L 192;
    PWM1S1P1H = 0x5D;   //PWMS1P1H 93;
    PWM1S1P2L = 0x80;   //PWMS1P2L 128;
    PWM1S1P2H = 0x3E;   //PWMS1P2H 62;
    PWM1CON   = 0x80;   //PWMEN enabled; PWMLD disabled; PWMERSPOL disabled;
                        //PWMERSNOW disabled;
}
```

4. Variable-Aligned Mode

In Variable-Aligned mode, the active part of the PWM signal can be at any part of the PWM period. A variable-aligned PWM is commonly used in power supply and motor control applications where a fixed pulse width is needed, but the effective ON time of the MOSFETs needs to be controlled. In this mode, when the PWM timer and parameter P1 match, the output transitions to the Active state. When the PWM timer and parameter P2 match, the output transitions to the Inactive state. Adjusting the P1 and P2 values will move the transition edges. Both the PWM outputs are identical in this mode. As an example, if P1 is 20% of the period, and P2 is 50% of the period, the output will be low for the first 20% of the period, high for the next 30%, and return to low for the remaining 50%.

Figure 4-1. Variable-Aligned Mode



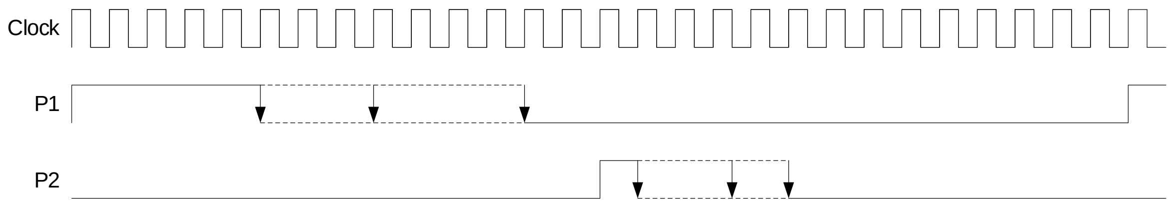
Example 4-1. PWM Variable-Aligned Initialization

```
void PWM1_Initialize_V_Aligned(void)
{
    PWM1ERS = 0x00;   //PWMERS External Reset Disabled;
    PWM1CLK = 0x02;   //PWMCLK FOSC;
    PWM1LDS = 0x00;   //PWMLDS Autoload disabled;
    PWM1PRL = 0x9F;   //PWMPRL 159;
    PWM1PRH = 0x0F;   //PWMPRH 15;
    PWM1CPRE = 0x00;  //PWMCPRE No prescale;
    PWM1PIPOS = 0x00; //PWMPIPOS No postscale;
    PWM1GIR = 0x00;   //PWMS1P2IF PWM2 output match did not occur;
                      //PWMS1P1IF PWM1 output match did not occur;
    PWM1S1CFG = 0x03; //PWMPOL2 disabled; PWMPOL1 disabled; PWMPPEN
                      //disabled; PWMOUT1 and PWMOUT2 in variable
                      //aligned mode;
    PWM1S1P1L = 0xF0; //PWMS1P1L = 240;
    PWM1S1P1H = 0x09; //PWMS1P1H  = 9;
    PWM1S1P2L = 0xA0; //PWMS1P2L = 160;
    PWM1S1P2H = 0x0E; //PWMS1P2H = 14;
    PWM1CON = 0x80;   //PWMEN enabled; PWMLD disabled; PWMERSPOL
                      //disabled; PWMERSNOW disabled;
}
```


5. Push-Pull Mode

The Push-Pull mode generates two PWM outputs alternately on the P1 and P2 channels. The duty cycle of the two outputs are still independently controllable. This mode works with Left-Aligned and Right-Aligned modes. Typically, this mode is used to generate an AC drive for either a motor or transformer primary drive using a full-bridge driver.

Figure 5-1. Push-Pull Mode



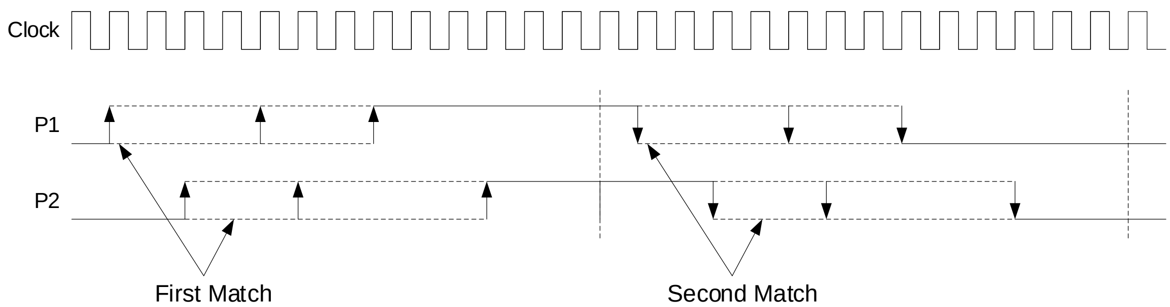
Example 5-1. PWM Push-Pull Initialization

```
void PWM1_Initialize(void)
{
    PWM1ERS = 0x00;         //PWMERS External Reset Disabled;
    PWM1CLK = 0x02;         //PWMCLK FOSC;
    PWM1LDS = 0x00;         //PWMLDS Autoload disabled;
    PWM1PRL = 0x1F;         //PWMPRL 31;
    PWM1PRH = 0x00;         //PWMPRH 0;
    PWM1CPRE = 0x00;        //PWMCPRE No prescale;
    PWM1PIPOS = 0x00;       //PWMPIPOS No postscale;
    PWM1GIR = 0x00;         //PWMS1P2IF PWM2, PWMS1P1IF PWM1 no match
    PWM1S1CFG = 0x0A;       //PWMPOL2 disabled; PWMPOL1 disabled;
                            //PWMPPEN disabled; PWMMODE PWMOUT1 and
                            //PWMOUT2 in center-aligned mode;
    PWM1S1P1L = 0x10;       //PWMS1P1L 16;
    PWM1S1P1H = 0x00;       //PWMS1P1H 0;
    PWM1S1P2L = 0x10;       //PWMS1P2L 16;
    PWM1S1P2H = 0x00;       //PWMS1P2H 0;
    PWM1GIRbits.S1P1IF = 0; //Clear PWM1 slice 1, output 1 interrupt flag
    PWM1GIRbits.S1P2IF = 0; //Clear PWM1 slice 1, output 2 interrupt flag
    PWM1CON = 0x80;         //PWMEN enabled; PWMLD disabled;
                            //PWMERSPOL disabled; PWMERSNOW disabled;
}
```


6. Toggle on Compare

This output mode toggles the PWM slice output when the duty cycle compare match fires. The result is two 50% duty cycle outputs, with double the PWM period, and offset from each other by the difference in the duty cycle settings. If both duty cycle registers are set to the same value, both outputs will rise and fall together. If the duty cycle registers are set with a 20 μs difference, the two 50% duty cycle outputs will be offset from each other by 20 μs. This output mode allows the users to generate 50% duty cycle clock signals with a programmable phase difference. The following code generates a 10 kHz output with a 30 μs offset.

Figure 6-1. Toggle Mode



Example 6-1. PWM Toggle Initialization

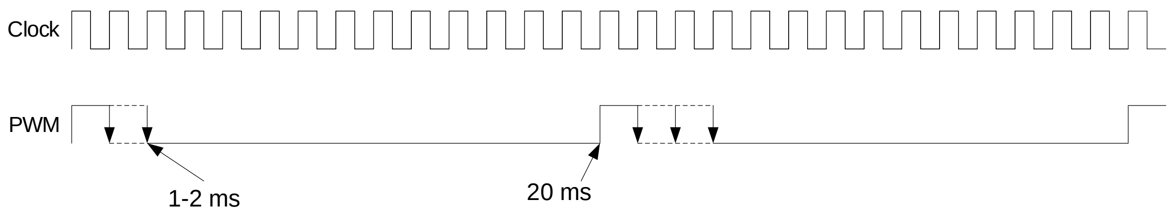
```
void PWM1_Initialize(void)
{

    PWM1ERS = 0x00;     //PWMERS External Reset Disabled;
    PWM1CLK = 0x03;     //PWMCLK HFINTOSC;
    PWM1LDS = 0x00;     //PWMLDS Autoload disabled;
    PWM1PRL = 0xC7;     //PWMPRL 199;
    PWM1PRH = 0x00;     //PWMPRH 0;
    PWM1CPRE = 0x00;    //PWMCPRE No prescale;
    PWM1PIPOS = 0x00;   //PWMPIPOS No postscale;
    PWM1GIR = 0x00;     //PWMS1P2IF PWM2 output no match; PWMS1P1IF PWM1 output
                        //no match;
    PWM1GIE = 0x00;     //PWMS1P2IE disabled; PWMS1P1IE disabled;
    PWM1S1CFG = 0x05;   //PWMPOL2 disabled; PWMPOL1 disabled; PWMPPEN disabled;
                        //PWMMODE Toggle PWMOUT1 and PWMOUT2 on compare match;
    PWM1S1P1L = 0x27;   //PWMS1P1L 39;
    PWM1S1P1H = 0x00;   //PWMS1P1H 0;
    PWM1S1P2L = 0x9F;   //PWMS1P2L 159;
    PWM1S1P2H = 0x00;   //PWMS1P2H 0;
    PWM1CON = 0x80;     //PWMEN enabled; PWMLD disabled; PWMERSPOL disabled;
}
```

7. Servo Control

The 16-bit PWM module is an excellent choice to be used as a servo motor controller, which requires minimal software intervention. Servo motors usually have three-pin connectors consisting of V_{DD}, V_{SS} and a Control Pin (PWM Output from the microcontroller). The typical PWM output signal used to drive servo motors generates a pulse every 20 ms that lasts anywhere from 1 ms to 2 ms. This equates to a PWM frequency of 50 Hz with a varying duty cycle of 5% to 10%, depending upon the desired position of the output shaft. A pulse width of 1 ms typically corresponds to the -90° position of the servo output shaft, and a 2 ms pulse will cause the output shaft to move to the 90° position.

To use the PWM module to drive a servo motor, the PWM frequency should be configured to be 50 Hz by selecting the appropriate PWM clock source and clock prescaler value. The user should then generate a left-aligned PWM output with a duty cycle of 5% to 10% to generate a pulse between 1 ms and 2 ms by using the corresponding PR register to change the duty cycle of the signal. The Example 1 below shows how the 16-bit PWM module can be initialized to drive a servo motor, and Example 2 includes a code snippet that can be used to sweep the servo motor output shaft between 0° and 180°. The figure below illustrates the PWM output signals that can be used to drive a servo motor between 0° and 180°.

Figure 7-1. PWM Output Signals



Example 7-1. PWM Servo Motor Control Initialization

```
void PWM1_Initialize(void)
{
    PWM1ERS = 0x00;            //PWMERS External Reset Disabled;
    PWM1CLK = 0x03;            //PWMCLK HFINTOSC;
    PWM1LDS = 0x00;            //PWMLDS Autoload disabled;
    PWM1PRL = 0x3F;            //PWMPRL 63;
    PWM1PRH = 0x9C;            //PWMPRH 156;
    PWM1CPRE = 0x01;           //PWMCPRE Prescale by 2;
    PWM1PIPOS = 0x00;          //PWMPIPOS No postscale;
    PWM1GIR = 0x00;            //PWMS1P2IF PWM2 output match did not occur;
                               //PWMS1P1IF PWM1 output match did not occur;
    PWM1GIE = 0x00;            //PWMS1P2IE disabled; PWMS1P1IE disabled;
    PWM1S1CFG = 0x00;          //PWMPOL2 disabled; PWMPOL1 disabled;
                               //PWMPPEN disabled; PWMMODE PWMOUT1 and PWMOUT2
                               //in left-aligned mode;
    PWM1S1P1L = 0xA0;          //PWMS1P1L 160;
    PWM1S1P1H = 0x0F;          //PWMS1P1H 15;
    PWM1S1P2L = 0x80;          //PWMS1P2L 128;
    PWM1S1P2H = 0x0C;          //PWMS1P2H 12;
    PIR4bits.PWM1PIF = 0;      //Clear PWM1 period interrupt flag
    PIR4bits.PWM1IF = 0;       //Clear PWM1 interrupt flag
    PWM1GIRbits.S1P1IF = 0;    //Clear PWM1 slice 1, output 1 interrupt flag
    PWM1GIRbits.S1P2IF = 0;    //Clear PWM1 slice 1, output 2 interrupt flag
    PIE4bits.PWM1IE = 0;       //PWM1 interrupt enable bit
    PIE4bits.PWM1PIE = 0;      //PWM1 period interrupt enable bit
    PWM1CON = 0x80;            //PWMEN enabled; PWMLD disabled; PWMERSPOL
                               //disabled; PWMERSNOW disabled;
}
```

Example 7-2. Servo Motor 5% to 10%

```
while (1)
{
   PWM1_SetSlice1Output1DutyCycleRegister(0x07D0);    //PWM duty cycle at 5%
   PWM1CONbits.LD = 1;                                //PWM Reload Period &
                                                      //Duty Cycle Registers
                                                      //Bit
   __delay_ms(1000);
   PWM1_SetSlice1Output1DutyCycleRegister(0x0FA0);    //PWM duty cycle at 10%
   PWM1CONbits.LD = 1;                                //PWM Reload Period &
                                                      //Duty Cycle Registers
                                                      //Bit
   __delay_ms(1000);
}
```

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with

Trademarks

ISBN: 978-1-5224-5425-3

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
http://www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-72884388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820