



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIA EXACTAS

SEMINARIO DE PROBLEMAS DE PROGRAMACIÓN DE SISTEMAS EMBEBIDOS.

LABORATORIO #11 – PWM

ALUMNO: MENESES LÓPEZ ARISAI RICARDO

DOCENTE: ALVARADO RODRIGUEZ FRANCISCO JAVIER

20 NOVIEMBRE DEL 2019

MATERIALES

COMPONENTES.

- PROTOBOARD
- CABLE PARA PROTO
- PINZAS DE CORTE / AGARRE
- LED
- FUENTE DE VOLTAJE 5V
- RESISTENCIA 220Ω
- PROGRAMADOR PARA PIC – MASTER PROG.

MICROCONTROLADOR

- PIC16F887

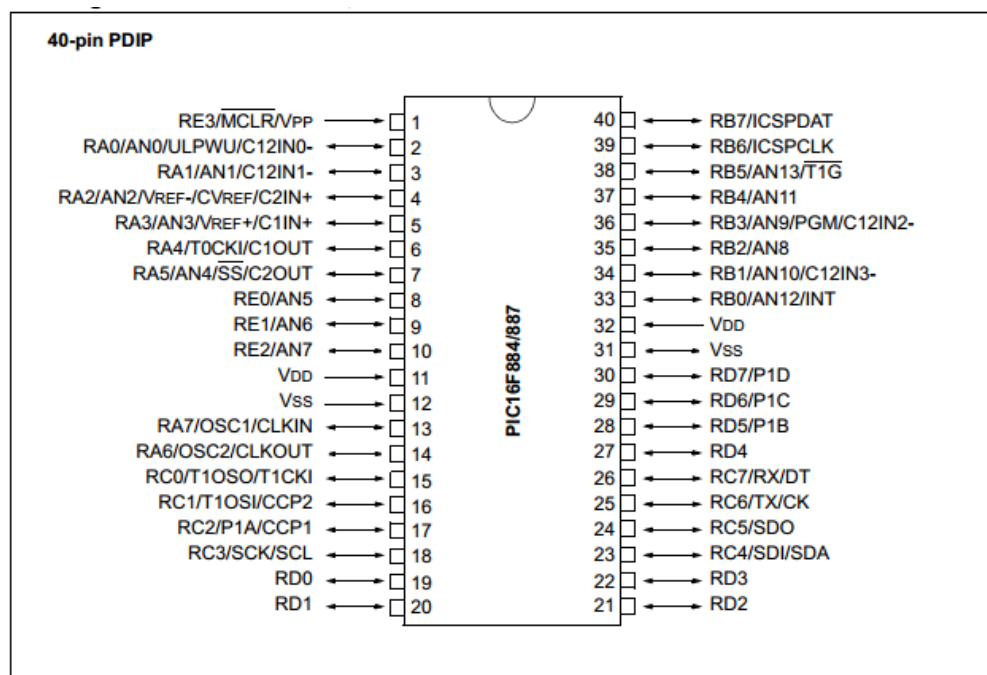
SOFTWARE

- MIKROC PRO FOR PIC
- PROTEUS DESIGN SUITE
- MASTER – PROG+

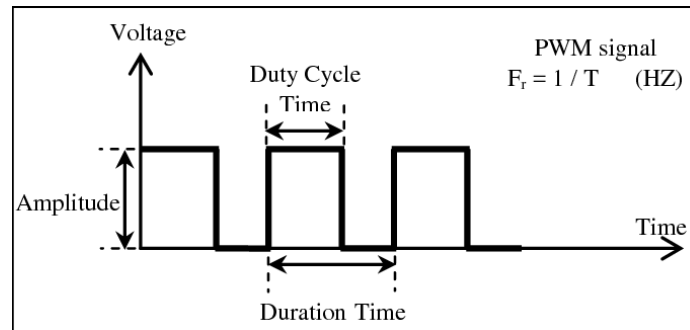
1. OBJETIVO DEL PROYECTO

- La aplicación será sobre un LED, el cual cambiará desde su mínima hasta su máxima intensidad luminosa, en lapsos de $\frac{1}{2}$ por segundo y realizando esta operación de forma cíclica.

2. MARCO TEÓRICO



La modulación por ancho de pulso o PWM es una técnica en la que se modifica el ciclo de trabajo (Duty Cycle) es el ancho relativo de su parte positiva en relación al periodo.



$$D = \frac{\tau}{T}$$

D es el ciclo de trabajo.

τ es el tiempo en que la función está en estado alto.

T es el periodo de la función.

3. DESARROLLO

La información requerida gira en torno a un grupo de elementos:

- Timers
- Comparadores
- Registros
- Latch RS

Los Registros CCPRxL y CCPRxH definen al Duty Cycle, el TMR2 afecta al periodo y al Duty Cycle y PR2 tiene efecto sobre el periodo.

Para definir el periodo de PWM, se tiene la ecuación:

$$PWM = [(PR2) + 1] * (Valor\ del\ pre - escalador\ TMR2) * (Tosc) * 4$$

Donde:

- PR2 – Es el registro del periodo del Timer2 (bits)
- Valor del pre-escalador TMR2 – Es el valor del pres-escalador
- Tosc – Es (1/Fosc).

Definiendo el ancho del pulso (Pulse Width):

$$PW = (CCPRxL: CCPxCON < 5:4 >) * (Valor\ del\ pre - escalador\ TMR2) * Tosc$$

Donde:

- CCPRxL y CCPxCON <5:4> forman un registro de 10 bits.
- CCPRxL almacena los 8 Bits más significativos y CCPxCON los 2 Bits menos significativos en el Bit 5 y 4 respectivamente.

Finalmente, la ecuación que define el Duty Cycle:

$$Duty\ Cycle = \frac{CCPRxL:CCPxCON < 5:4 >}{4(PR2 + 1)}$$

Cálculos

Se buscará un periodo de 256 μ seg. como máximo, tomando en consideración que el valor del pre-escalador es 1:1 y que la frecuencia del oscilador es 4 MHz. Por lo tanto:

$$PR2 = \frac{Periodo\ PWM}{(Valor\ del\ Pre - escalador\ TMR2) * (Tosc) * 4} - 1$$

$$PR2 = \frac{256\ \mu seg}{(1) * \left(\frac{1}{4}\ \mu seg\right) * 4} - 1$$

Determinando el valor de CCPRxL:CCPxCON <5:4> :

$$PW = (CCPRxL:CCPxCON < 5:4 >) * (Valor\ del\ pre - escalador\ TMR2) * Tosc$$

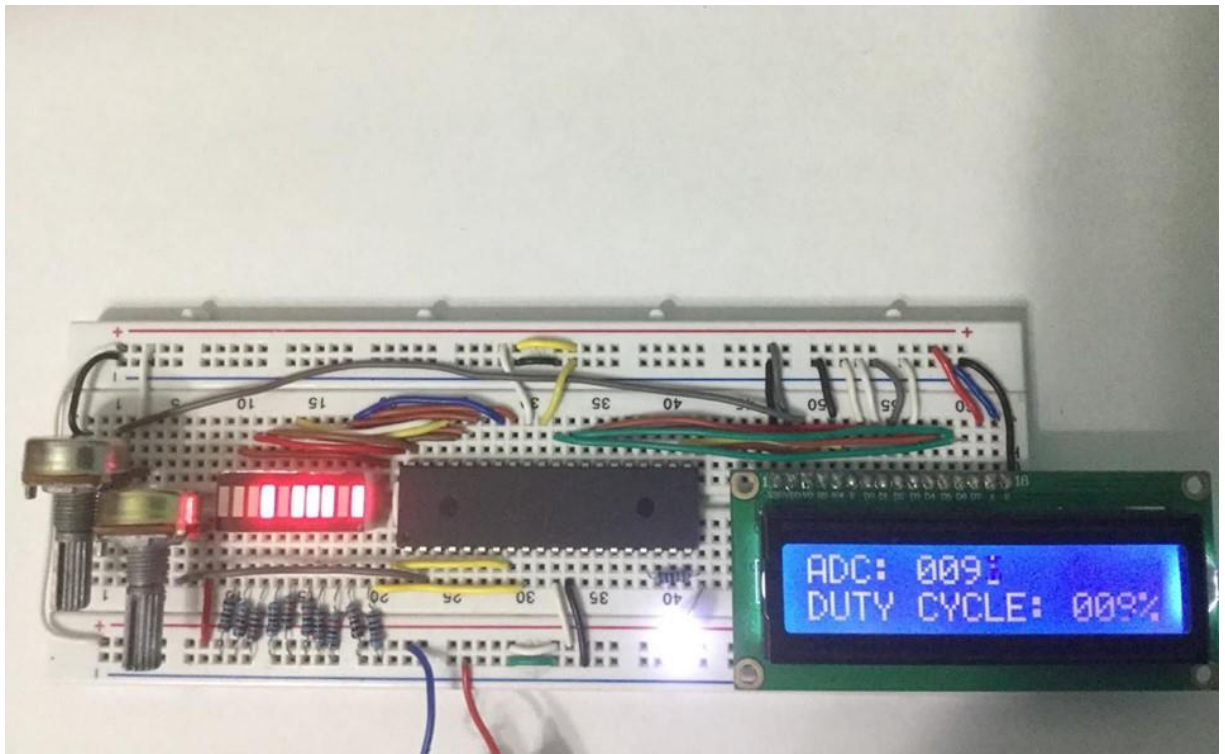
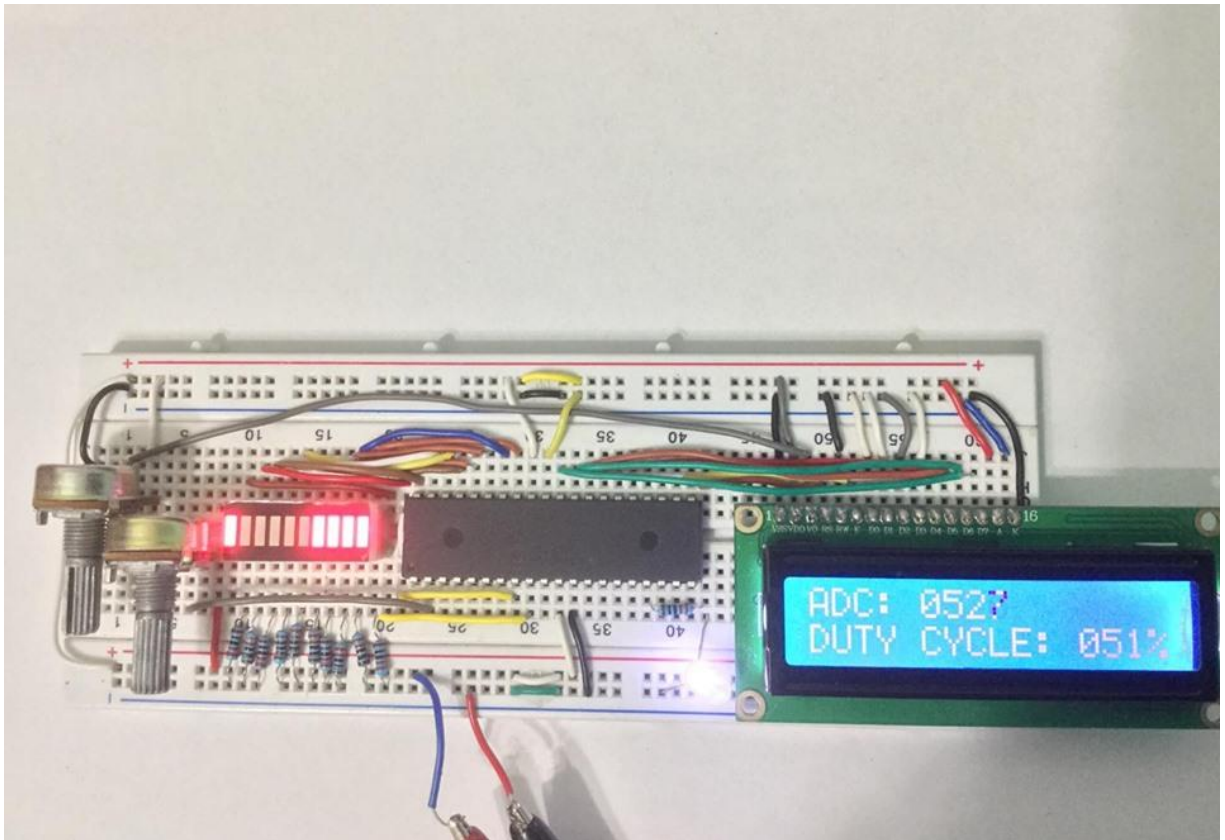
$$(CCPRxL:CCPxCON < 5:4 >) = (256\ \mu seg) * \frac{4}{\mu seg} = 1'24$$

Debe considerarse que se trata de 1024 pasos que equivalen al 100% del periodo; estos pasos irán desde 0 hasta 1023, por lo que debe cargarse 1023 en los registros correspondientes.

4. SIMULACIÓN



5. PROTOBOARD



6. PROGRAMACIÓN

CÓDIGO

```
//Prototipado
void PWM_CFG();           //CONFIGURACIÓN DE LOS REGISTROS DEL PWM
void Ancho_Pulso(int ancho_useg); //ANCHO DE PULSO
void Periodo(int t_useg);  //PERIODO DE LA SEÑAL

void main()
{
    int i;
    PWM_CFG();
    Periodo(256);

    while(1)
    {
        for(i=0;i<=1000;i=i+100)
        { Ancho_Pulso(i); delay_ms(500); }
    }

    void PWM_CFG()
    {
        CCP1CON.P1M1=0; //PWM SINGLE OUTPUT, P1A MODULADA, P1B, P1C Y P1D COMO I/O
        CCP1CON.P1M0=0;
        CCP1CON.CCP1M3=1; //MODO PWM P1A, P1C, P1B, P1D ACTIVAD EN ALTO.
        CCP1CON.CCP1M2=1; //PRE-ESCALA DEL TIMER2 1:1

        PIR1.TMR2IF=0; //LIMPIEZA DE LA BANDERA DE INTERRUPCIÓN DEL TRM2
        T2CON.TMR2ON=1; //ACTIVACIÓN DEL TMR2
        TRISC.RC2=0; //ACTIVACIÓN DE LA SALIDA DE LA TERMINAL
    }

    void Ancho_Pulso(int ancho) //Ancho de Pulso
    {
        //De 0 A 1023 PARTES EQUIVALENTES DE 0 A 100%
        //LA FRECUENCIA EN EL OSCILADOR ES DE 4 MHZ
        //SE CONSIDERA LA PRE-ESCALA DE TMR2 1:1

        CCP1CON|=((ancho&0x0003)<<4);
        CCPR1L=ancho>>2;
    }

    void Periodo(int t_useg) //PERIODO DE LA SEÑAL
    {
        //DE 1 A 256 MICROSEGUNDOS
        //LA FRECUENCIA EN EL OSCILADOR ES DE 4 MHZ
        //SE CONSIDERA LA PRE-ESCALA DEL TMR2 1:1
        PR2=(t_useg-1);
    }
```

La función Ancho_Pulso se encarga de colocar el valor de la variable ancho en los 10 Bits correspondientes, la operación “&” permite enmascarar la variable para obtener los dos bits menos significativos y posteriormente desplazar el resultado 4 posiciones a la izquierda para ubicar estos bits en el Bit 5 y 4 del registro CCP1CON. Para colocar los 8 Bits restantes en el registro CCPR1L se toma la variable “ancho” de 10 Bits y se desplaza dos posiciones a la derecha eliminando dos Bits que ya se ubicaron en el registro anterior.

7. CONCLUSIONES

- En base a CCPRxL:CCPxCON <5:4> va creciendo de 0 a 1023, el ancho de pulso aumentan de 0 a 256 μseg (0% - 100%) en un determinado periodo.
- En base al porcentaje del Duty que va desde 0% a 100% varia el voltaje de salida que se obtiene por los puertos B1 y B2 que corresponden al PWM1 y PWM2, por que la intensidad de corriente varia de acuerdo a dicho porcentaje.

8. BIBLIOGRAFÍA

[1] - J. M. Morán Loza. Programación de Sistemas Embebidos Con Aplicaciones Para El PIC16F8XX. MEXICO: PEARSON, 2014.