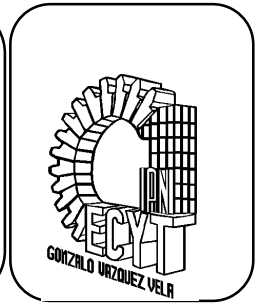


INSTITUTO POLITÉCNICO NACIONAL
Centro de Estudios Científicos y Tecnológicos N° 1
“Gonzalo Vázquez Vela”
Academia de Sistemas Digitales
Prácticas de Micro Electrónica Programable



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Paterno _____ Nombre _____
 N° DE BOLETA: _____ GRUPO: _____

ASIGNATURA: **Micro Electrónica Programable**

HOJA	DE	FECHA			EVALUACION
		DIA	MES	AÑO	

PROFESOR: _____

Práctica 10

Modulación por ancho de pulso (PWM)

Competencias de La Unidad:

- Emplea un microcontrolador para procesamiento de señales analógicas y digitales

Resultado de Aprendizaje Propuesto (RAP):

- Aplica la modulación por ancho de pulso en el control de velocidad de motores de CD y control de posición de servomotores

Objetivos de la Práctica:

1. Utilizar el módulo captura/comparación/PWM del microcontrolador para generar un señal de tipo PWM.
2. Realizar un monitoreo de una variable física, con el fin realizar el control del tiempo de la señal modulada por ancho de pulso.
3. Implementar programas en un circuito basado en microcontrolador, para comprobar su funcionamiento.

Equipo Necesario

Computadora (con el Software MPLAB IDE, IC-PROG o similar, compilador C, Simulador de circuitos electrónicos “Proteus”)
 Programador tipo JDM o similar.

Material Necesario

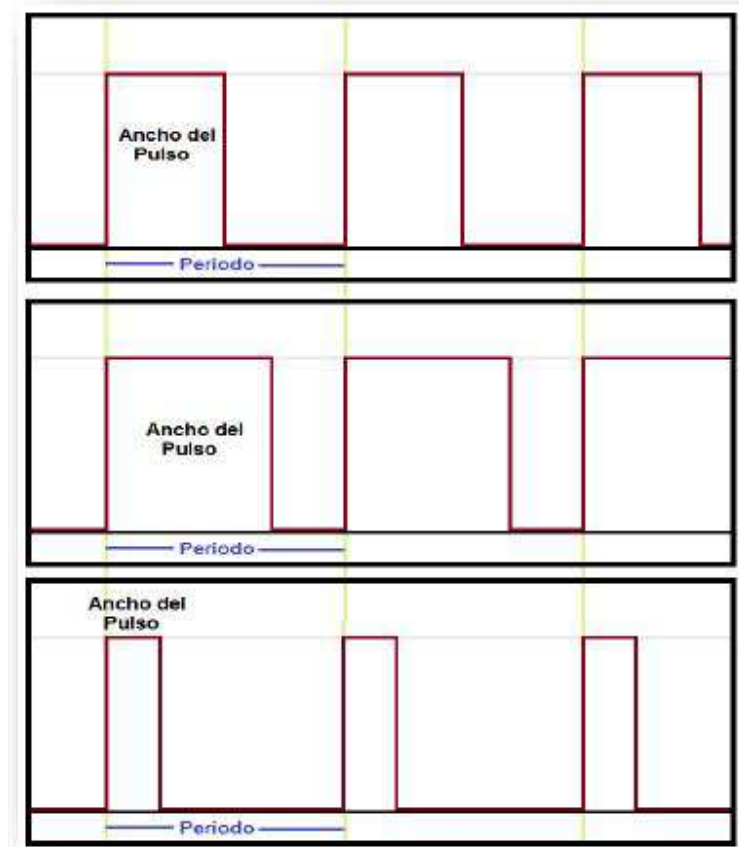
Instrucciones del PIC 16F887 u otro de gama media o alta.
 Hoja de especificaciones del PIC 16F887 u otro de gama media o alta
 Manual de Referencia de CCS

Introducción Teórica

Modulación por ancho de pulso.

Modulación por ancho de pulso o PWM (Pulse-Width Modulation), es una técnica que modifica el ciclo de trabajo o el ancho del pulso de una señal periódica, en este caso representado por una señal cuadrada.

Y uno de los usos del PWM entre muchos otros, es controlar la cantidad de energía, en este caso el voltaje promedio es mayor conforme aumenta el ciclo de trabajo.



Como podemos observar en la imagen el periodo de la señal permanece fijo, por lo tanto, la frecuencia se mantiene y solamente cambia el ciclo de trabajo, en la primera se observa que el ciclo de trabajo es de aproximadamente 50% lo cual nos indica que es el porcentaje de voltaje promedio entregado a la carga.

El PWM se puede utilizar en varias cosas, como el control de la velocidad de motores de DC, la posición de un servomotor, fuentes conmutadas, entre otras cosas más.

En el caso del microcontrolador PIC cuenta con el modulo para poder obtener dicha señal denominado MODULO CCP CAPTURA / COMPARACIÓN / PWM (Modulación de ancho de Pulsos) y en este caso tiene las características:

Los microcontroladores de la familia 16F88X y otras familias disponen de dos módulos de Captura / Comparación y PWM donde cada módulo cuenta con un registro de 16 bits que pueden trabajar como:

- Registro de captura de 16 bit
- Registro de comparación de 16 bits
- Modulación de anchura de pulsos PWM

Los módulos **CCP1** y **CCP2** son idénticos en su modo de funcionamiento, con la excepción del modo de disparo especial

Módulo CCP1:

El registro de Captura / Comparación / PWM (CCPR1) de 16 bits está formado por dos registros de 8 bits: **CCPR1L** (byte, bajo) que ocupa la dirección 15h y el **CCPR1H** (byte alto) que ocupa la posición 16h. El registro de control del **CCP1** es el **CCP1CON** que ocupa la dirección 17h. El modo de disparo especial se genera por la igualdad en la comparación de **CCPR1** con **TMR1** y reestablecerá el **TIMER1** y el **CCPR1**, funciona como un registro de período, capaz de provocar periódicamente interrupciones.

Módulo CCP2:

El registro de Captura / Comparación / PWM (CCPR2) de 16 bits está formado de manera similar que el **CCP1** por dos registros de 8 bits: **CCPR2L** (byte bajo) que ocupa la dirección 1Bh y **CCPR2H** (byte alto) que ocupa la dirección 1Ch. El registro de control del **CCP2** es el **CCP2CON** que se encuentra en la dirección 1Dh. En este caso el modo de disparo especial se genera por igualdad en la comparación con el **TIMER1** provocando que se reestablezca el **TIMER1** y comience una conversión A/D, en el acso del que el convertidor A/D está habilitado. Con lo cual puede realizar la conversiones A/D de forma periódica sin en control del programa de instrucciones.

De lo anterior, el módulo CCP realiza tres funciones básicas basadas en el manejo de los temporizadores

- Comparador: compara el valor del temporizador con el valor de un registro y provocando un acción el PIC.
- Captura obtiene el valor del temporizador en un momento dado, fijado por la acción de una terminal del PIC.
- PWM genera una señal modulada en amplitud de pulsos.

Con el fin de facilitar el manejo del módulo el compilador CCS cuenta con las siguientes funciones para utilizarlo

- Configuración del módulo CCPx

```
setup_ccpx(modo) //x puede ser 1 o 2
```

modo: hace referencia a los bits CCPxM3:CCPxM0 del registro CCPxCON

- Definición del ciclo de trabajo

```
set_pwm_x_duty(valor)
```

valor: dato de 8 o 16 bits que determina el ciclo de trabajo.

ACTIVIDADES TEÓRICAS PREVIAS

Investigar los siguientes:

- Como se determina el periodo de la señal PWM que genera en el módulo.
- Como determina el ciclo de trabajo de la señal PWM generada por el microcontrolador.
- ¿Menciona que características tiene el módulo PWM.
- Investiga las características y diagramas del módulo captura/comparación/ PWM
- ¿Cuales es la resolución del PWM?
- Investiga los modos como se puede configurar el módulo captura/comparación/ PWM
- Investigar como relaciona el TIMER 2 con el modulo PWM y cuáles son las funciones para utilizarlos en conjunto.
- Menciona como se nombran los modos en la función `setup_ccpx(modulo)` de CCS
- Investiga los pasos para poner en marcha el modo PWM

ACTIVIDADES PREVIAS

- Crear un proyecto de nombre `pra10` en la carpeta `c:\MEPIC\practica10` en MPLAB o PIC C Compiler. Los programas de cada ejercicio deben ser guardados con el nombre `practica9X.c` con $X= 1, 2, 3, \dots, A$.
- En el caso de utilizar MPLAB, realizar los siguientes pasos:
 - a. Utilizar Project wizard y seleccionar el compilador de `c`
 - b. Agregar al proyecto los archivos adecuados con extensión `c` y `h`.
 - c. Habilitar Simulador MPLAB SIM y modificar la frecuencia del simulador a 4 Mhz.
 - d. Utilizaremos la herramienta de stopwatch, para obtener la elija Debugger >> Stopwatch.
 - e. Obtener la herramienta de watch, de la siguiente manera View>> watch.
 - f. Y seleccione los registros `PORTA, PORTB, PORTC, PORTD, PORTE, TRISA, TRISB, TRISC, TRISD, TRISE` y `W`
- Si usa PIC C compiler crear el proyecto únicamente.

Nota: La Implementación física de cada circuito, efectuar si el docente lo considera pertinente.

ACTIVIDADES PRÁCTICAS

Parte 1

1. Realizar los siguientes programas que ejemplifican el funcionamiento módulo PWM, simularlos en Proteus mediante el circuito propuesto.

Ejemplo 1

El siguiente programa ejemplifica el uso del módulo PWM (figura 10. 1), generando un señal de diferente valor en cada uno de los canales.

```
#include <16F887.h>
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(clock=4000000)
void main() {
int16 valor,valor2; //variables de 10 bits 0 a 1023
//configuracion del PWM
setup_timer_2(T2_DIV_BY_4,249,1); //define el valor de division del timer 2
setup_ccp1(CCP_PWM); //habilita salida PWM CCP1
setup_ccp2(CCP_PWM); //habilita salida PWM en CCP2
valor=500;
valor2=900; //Por ser una simulacion; el valor maximo asignado es de 900
// debido a que valores mayores no pueden ser visualizados
set_pwm1_duty(valor);
set_pwm2_duty(valor2);
}
```

Observa la señal en el osciloscopio del simulador y obtenga

Señal Canal A (CCP1): Periodo: Tiempo alto: Tiempo Bajo:

Señal Canal B (CCP2): Periodo: Tiempo alto: Tiempo Bajo:

Modifique los valores de las variables valor y valor2; y explique que sucede en la señal en el osciloscopio del simulador y obtenga.

Señal Canal A (CCP1): Periodo: Tiempo alto: Tiempo Bajo:

Señal Canal B (CCP2): Periodo: Tiempo alto: Tiempo Bajo:

Ejemplo 2 Mediante la señal PWM controla la velocidad de un motor de CD (figura10.1).

```
#include <16F887.h>
#device adc=10
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use delay(clock=4000000)
#byte INTCON= 0x0B
#use standard_io(a)
#use standard_io(b)
#use standard_io(c)
long valor; //variables de 10 bits 0 a 1023
int aux;
#INT_TIMER0
void TIMER0_isr(void)
{
delay_us(20);
aux=1;
}
```

```

set_timer0 (150);
}
void main() {
INTCON=0x00;
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256); //Configuración timer0
set_timer0 (150); //Carga del timer0
enable_interrupts(INT_TIMER0); //Habilita interrupción timer0
enable_interrupts(global);
//configura ADC
setup_adc_ports(SAN0); //configura Canal 0 como analógico
// En el caso de requerir todos los canales colocar ALL_ANALOG
setup_adc(ADC_CLOCK_DIV_32); //Fuente de reloj
//configuración del PWM
setup_timer_2(T2_DIV_BY_4,200,1); //define el valor de división del timer 2
//la frecuencia de la señal
setup_ccp1(CCP_PWM); //habilita salida PWM CCP1
//setup_ccp2(CCP_PWM); //habilita salida PWM en CCP2
while(true)
{ aux=0;
set_adc_channel(0); //Habilitación canal0
delay_us(20); // tiempo de retardo para iniciar lectura
valor= read_adc();
set_pwm1_duty(valor);
while(bit_test(aux,2)!=0)
{delay_us(1);
}
}
}
}

```

- **Observa la señal en el osciloscopio del simulador y el motor, varié el valor de del potenciómetro y mencione que sucede, además determine:**

Señal Canal A (CCP1): Periodo:

Tiempo alto:

Tiempo Bajo:

- **Modifique la instrucción:**

setup_timer_2(T2_DIV_BY_4,249,1); **por** setup_timer_2(T2_DIV_BY_16,249,1); **en el programa anterior y mencione que sucede en la señal en el osciloscopio del simulador y en el motor al variar el valor de del potenciómetro, además determine:**

Señal Canal A (CCP1): Periodo:

Tiempo alto:

Tiempo Bajo:

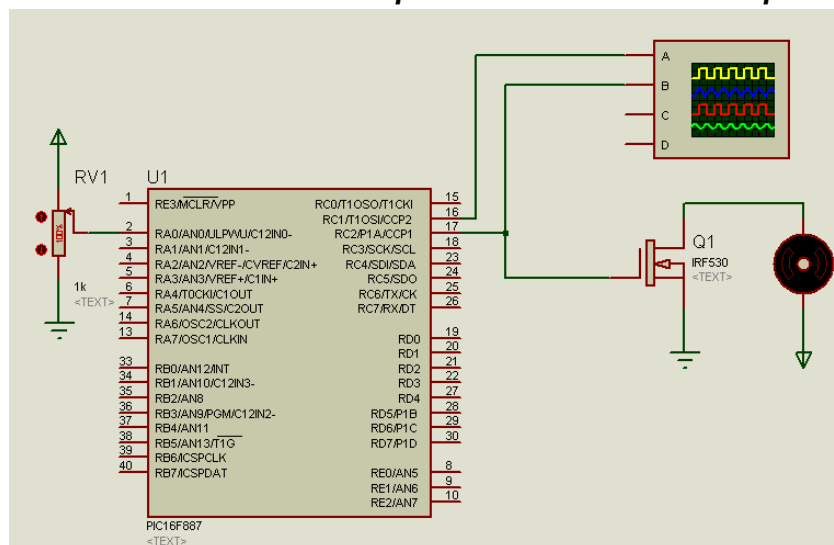


Figura 10.1

Parte 2

Realizar el siguiente programa con su respectiva simulación, que modifique el ancho de pulso de una señal de periodo 1.6ms, mediante 2 interruptores conectados en RB4 (aumenta) y RB5(decrementa), considere que la señal debe iniciar con un ancho de pulso de 5%,(Sugerencia utilice interrupción por cambio de nivel en puerto B) , Proponga el circuito para que maneje la velocidad y sentido de un motor de CD.

2. Conclusiones

A. Realizar conclusiones de manera individual.

3. Cuestionario

- a) ¿Cómo se determina tiempo en alto y bajo en unas tipo PWM?
- b) ¿Cuántas terminales están asociados al Modulo PWM se puede seleccionar en el PIC16f876?
- c) ¿Qué registros están asociado al módulo PWM?
- d) ¿Cuales es la resolución del módulo PWM?

Comentarios Finales

- El alumno entrega un reporte de la práctica, como el profesor lo indique.
- El reporte debe contener el diagrama de flujo o algoritmo (Seudo código) de cada uno de los programas.
- Además, en el reporte deben anexarse las conclusiones y cuestionario contestado.