



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica

## Práctica 9



### Control de Motores de DC (PWM) y CAD. Programados en Lenguaje C

#### Agradecimiento

Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME  
PE110618

#### 1. Objetivos de aprendizaje

- a. Objetivo general.
  - El objetivo de esta práctica es implementar un sistema que permita controlar el sentido de giro y velocidad de dos motores de corriente directa, mediante la programación en lenguaje C.
  
- b. Objetivos específicos.
  - Conocer el funcionamiento de un motor de corriente continua.



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES



Área: Electrónica

- Analizar y configurar el funcionamiento PWM del microcontrolador PIC16F887, para el control de movimiento de los motores.

## 2. Introducción

Existen una gran variedad de tamaños, formas y potencias de motores DC, pero todos se basan en el mismo principio para su funcionamiento.

Accionar un motor DC es tan fácil como alimentar sus terminales con un voltaje y tierra. Para invertir su dirección basta con invertir la alimentación.

A diferencia de los motores a pasos y los servos motores, los motores DC no pueden ser colocados en una posición específica ya que estos están diseñados para girar a la máxima velocidad y el sentido que la alimentación que se les aplicó.

Si queremos controlar su velocidad, se debe controlar la frecuencia con que alimentamos las terminales del motor; es decir, sólo estará alimentado por lapsos de tiempo los cuales controlarán su velocidad, es decir se está controlando la velocidad por PWM (Modulación por Ancho de Pulso). De tal forma que entre más grandes (hablamos de milisegundos) sean esos lapsos, menor será su velocidad.

### L293D

El circuito L293 es útil porque las tensiones de la salida de los circuitos TTL son de 5 V y no son suficientes para excitar un motor DC o PAP (paso a paso), por lo tanto, es necesario ocupar un integrado como el L293D que está diseñado para proporcionar corrientes bidireccionales de hasta 600 mA y voltajes de 4,5 V a 36 V, de esta manera podremos controlar la dirección y velocidad de motores DC y manejar motores PAP.

En el diagrama (9.1) se muestran ejemplos de cómo conectar un motor DC donde:

Los pines 1 y 9 activan o desactivan las salidas del circuito.

Los pines 2, 7, 10 y 15 son las entradas de control para los motores.

Los pines 3, 6, 11 y 14 son las salidas de control para los motores.

Los pines 4, 5, 12 y 13 deberán conectarse a tierra en todo momento.

El pin 8 es la alimentación de CI y debe tener 5 V.

Y por último el pin 16 es el voltaje de alimentación de los motores que puede ser de 4.5 a 36 V.

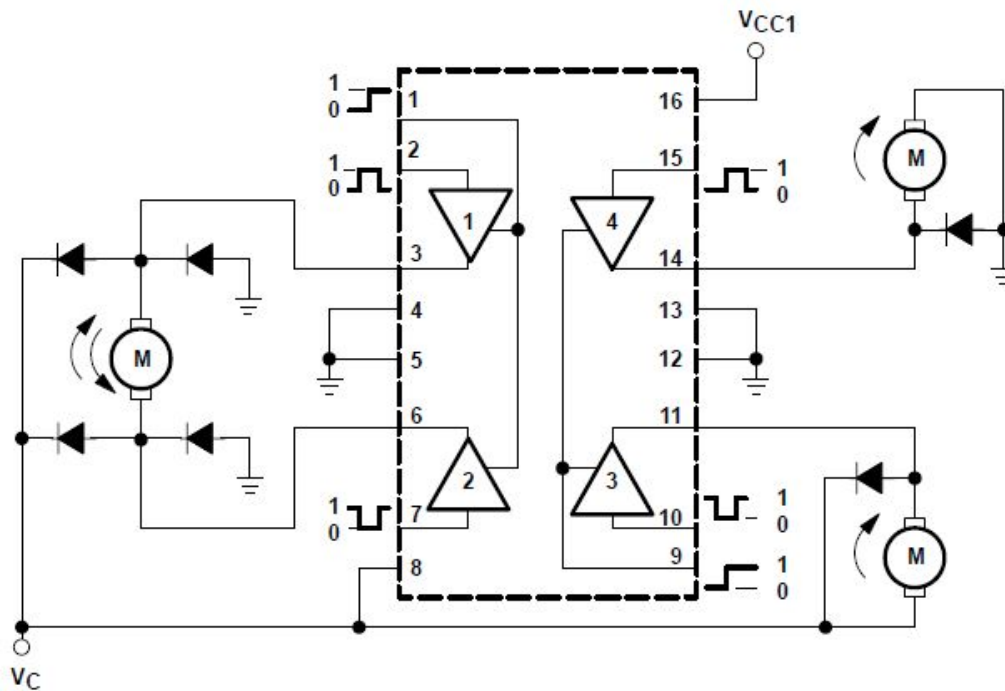


Diagrama 9.1 Fuentes (robots-argentina.com) *Conexionado para un motor con giro en ambos sentidos (lado izquierdo) y con motores con giro en sentido único en dos salidas (lado derecho) (5 nov 2018)*

## TMR2

El PIC16F887 cuenta con tres temporizadores para manejar eficientemente las operaciones que involucren tiempo y conteos. Estos se denominan TMR0, TMR1, TMR2.

En esta práctica utilizaremos en TMR2 el cual es un temporizador de 8 bits, que se puede leer y escribir, y que también puede realizar funciones para el SSP



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



(*Synchronous Serial Port*) y con los módulos de captura y comparación. Ocupan la dirección 11H del mapa de registros específicos.

La señal de reloj que utiliza el TMR2 es interna y se denomina  $F_{osc}/4$ , y antes de ser aplicada pasa por un *prescaler* de frecuencias con rangos de 1:1, 1:4 y 1:16, pasando por los 16 valores posibles.

Para controlar el funcionamiento del TMR2 se utiliza el registro T2CON, que ocupa la dirección 12H del mapa de registros específicos y cuya distribución se muestra a continuación (9.2).

### REGISTER 7-1: 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
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

<b>Legend:</b>			
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

Imagen 9.2 Fuentes (manual referido) T2CON: Timer2 registro de control

Los bits 1 y 0 (T2CKPS1:T2CKPS0) sirven para seleccionar el rango de división del *prescaler* de impulsos.

T2CKPS1	T2CKPS0	Rango
0	0	1:1
0	1	1:4
1	x	1:16

El bit 2 (TMR2ON) enciende y apaga el TMR2

1 = TMR2 ON

0=TMR2 OFF

Los siguientes bits del 6 al 3 (TOUTPS3-TOUTPS0) determinan el rango por el que se divide la frecuencia, el *postscaler*.



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



TOUTPS3-TUOTPS0	Rango
0000	1:1
0001	1:2
0010	1:3
...	...
1111	1:16

Resumen de los bits de los registros que se utilizan en el manejo y programación de TMR2 en la imagen (9.3).

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PR2	Timer2 Module Period Register								1111 1111	1111 1111
TMR2	Holding Register for the 8-bit TMR2 Register								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for Timer2 module.

## Imagen 9.3 Fuentes (manual referido) Registros asociados con el timer2

El señalizador de desbordamiento de TMR2 es el bit 1 (TMR2IF) de registro PIR1, que ocupa la dirección 0Ch.

El *prescaler* y el *postscaler* se ponen a 0 al escribir en el TMR2, al escribir el T2CON o con un *reset*. Sin embargo, cuando se escribe el T2CON no se borra el TMR2, sólo cuando se produce un *reset*.

El TMR2 tiene asociado un registro de período PR2, que se encuentra en la dirección 92h. Cuando el valor de conteo de TMR2 coincide con el valor del PR2 se genera un impulso de salida EQ y se resetea el TMR2. Estos impulsos en EQ pueden ser divididos por el *postscaler* antes de activar el señalizador TMR2IF.

Este temporizador también tiene la capacidad de producir una petición de interrupción si se pone a 1 el bit TMR2IE de habilitación, que reside en el registro PIE1 (8Ch). Además, con la programación adecuada, los impulsos de salida del TMR2 (antes del *postscaler*) pueden actuar como base de tiempo para el módulo SSP.



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES



Área: Electrónica

## EL TMR2 en C

La configuración del módulo TMR2 en el compilador de C se realiza con la función.

*setup\_timer\_2 (modo, periodo, postscaler);*

Dónde:

Modo afectará los bits 2, 1 y 0 del registro T2CON. En el cual podemos utilizar la tabla 9.4

Dato para "Modo"	Bit 2 del T2CON TMR2 (1=on, 0=off)	Bits 1 y 0 T2CON Valores del <i>prescaler</i>	Valor enviado al T2CON
T2_DISABLED	1	00	00000000 00h
T2_DIV_BY_1	1	00	00000100 04h
T2_DIV_BY_4	1	01	00000101 05h
T2_DIV_BY_16	1	10	00000110 06h

Tabla 9.4 Fuentes (creación propia) Configuración del registro T2CON (2018)

Periodo es un valor entre 0-255 (8 bits) para el registro PR2.

*Postscaler* es el valor para el *postscaler* que puede tomar los valores de 1-16 y afecta los bits 6-3 del registro T2CON.

Las funciones que nos permiten leer y escribir el módulo TMR2 son:

*valor = get\_timer2();*

*set\_timer2 (valor);* //Donde valor en un número de 0-255 (8 bits).

El PIC16F887 dispone de dos módulos CCP, llamados CPP1 Y CCP2 que son idénticos excepto por la modalidad de "disparo especial". Estos módulos pueden realizar tres funciones principales: Modo Captura, Modo Comparación y Modo Modulación de Anchura de Pulso (PWM).



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



En esta práctica utilizaremos el modo PWM con el módulo CCP2 que tiene como registros de trabajo a CCPR2H (1Ch) y CCPR2L (1Bh) y como registros de control a CCP2CON (1Dh). Las parejas de registros son las encargadas de modular la anchura del impulso. Los bits DC2B1 y DC2B0 actúan como los dos bits de menos peso en modo PWM, estando ubicados los otros ocho en el registro CCPR2L. Figura 9.5

REGISTER CCP2CON: CCP2 CONTROL REGISTER							
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
bit 7							bit 0

<b>Legend:</b>			
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 **Unimplemented:** Read as '0'
- bit 5-4 **DC2B<1:0>:** PWM Duty Cycle Least Significant bits  
Capture mode:  
 Unused.  
Compare mode:  
 Unused.  
PWM mode:  
 These bits are the two LSBs of the PWM duty cycle. The eight MSBs are found in CCPR2L.
- bit 3-0 **CCP2M<3:0>:** CCP2 Mode Select bits  
 0000 = Capture/Compare/PWM off (resets CCP2 module)  
 0001 = Unused (reserved)  
 0010 = Unused (reserved)  
 0011 = Unused (reserved)  
 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, set output on match (CCP2IF bit is set)  
 1001 = Compare mode, clear output on match (CCP2IF bit is set)  
 1010 = Compare mode, generate software interrupt on match (CCP2IF bit is set, CCP2 pin is unaffected)  
 1011 = Compare mode, trigger special event (CCP2IF bit is set, TMR1 is reset and A/D conversion is started if the ADC module is enabled. CCP2 pin is unaffected.)  
 11xx = PWM mode.

Figura 9.5 Fuentes (manual referido) CCP2CON: CCP2 registros de control

## Modo PWM (Pulse Width Modulation)

Un modulador por ancho de pulso (PWM) es un dispositivo que puede usarse para controlar la velocidad en motores DC. Los motores DC pequeños y medianos de imán permanente, son controlados exitosamente con transistores de conmutación por ancho de pulso.



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



Un circuito PWM arroja como resultado una onda cuadrada con ciclo variable de ON y Off, variando en el tiempo del 0 al 100 %. De esta manera, una cantidad variable de potencia es transferida a la carga.

La principal ventaja de un circuito de PWM sobre un controlador que se base en la variación lineal de la potencia suministrada a una carga mediante cambio resistivo es la eficiencia. A una señal de control del 50%, el PWM usará cerca del 50% de la potencia total, de la cual casi toda será transferida a la carga. En un controlador tipo resistivo, de un 50% de potencia que se quiera transferir a la carga se estima que le puede llegar cerca de un 71%. El otro 21% se pierde en forma de calor.

La principal desventaja de los circuitos de PWM es la posibilidad de que exista interferencia por radiofrecuencia (RFI). El RFI puede minimizarse ubicando el controlador cerca de la carga y utilizando un filtrado de la fuente de alimentación. Este circuito posee una pequeña protección contra RFI y produce una mínima interferencia

### 3. Equipo y material

- Equipo de cómputo
- Microcontrolador con al menos dos pines disponibles analógicos, y un puerto de salida.
- Reloj de cuarzo (oscilador).
- Capacitores.
- Botón pulsador (para reiniciar el microcontrolador).
- Resistencias.
- Luces indicadoras (cuatro al menos para las salidas).
- Protoboard y cables.
- Fuente de alimentación.
- Dos Motores DC.
- Controlador de motor L293
- Multímetro
- Osciloscopio

### 4. Metodología.

Se solicita al alumno que entienda la teoría sobre los registros asociados a la configuración de los pines de los puertos en un microcontrolador para emplearlos como entradas analógicas, salidas digitales y configuraciones de los registros para el PWM. Adicionalmente se solicita que genere un programa donde





# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



se lea información del medio externo, se procese la información en forma digital dentro del microcontrolador y posteriormente se despliegue la información relacionada con las entradas.

## 5. Desarrollo

### a. Actividad I.

**Diseño del programa.** Se recomienda el uso de una tabla de entradas y salidas, definir las variables internas que necesite el programa (en su caso), generar y discutir diagramas de flujo. Se recomienda el uso de computadora y los programas necesarios para la compilación del programa en el lenguaje seleccionado. Además, se debe consultar la hoja de especificaciones del microcontrolador.

### b. Actividad II

**Diseño y simulación del circuito electrónico.** Se consulta la hoja de especificaciones del microcontrolador para conectar los dispositivos de entrada y salida, así como los elementos de soporte, por ejemplo, el oscilador, la fuente de alimentación, el botón de reinicio (*reset*), por mencionar algunos. Se recomienda el uso de computadora y los programas necesarios para la simulación del circuito, empleando el programa diseñado en la actividad I.

### c. Actividad III

**Alambrado del circuito.** Interconectar los elementos seleccionados siguiendo el diagrama electrónico en una tarjeta *protoboard* (placa que posee unos orificios conectados eléctricamente entre sí siguiendo un patrón horizontal o vertical. Es empleada para realizar pruebas de circuitos electrónicos), previamente se debe grabar el código generado para el microcontrolador (archivo \*.HEX), en un grabador, finalmente verificar que en el circuito no haya cortocircuitos antes de energizar el sistema.

## 6. Resultados

Para que el usuario de este manual pueda ver resultados, es necesario definir qué acción realizará la salida ante la entrada. Se verificará que el funcionamiento de los motores DC sea el adecuado dependiendo de la configuración del PWM, además de que cumpla con lo solicitado en la práctica.



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



## 7. Aplicaciones

El control de los motores DC puede ser utilizado en una gran cantidad de equipos y dispositivos electrónicos; además de que permite reafirmar conocimientos adquiridos en prácticas anteriores.

## 8. Bibliografía

- ANGULO J. (2000). Microcontroladores PIC. España. Paraninfo.
- GONZALEZ J. (1992). Introducción a los Microcontroladores. España. Graw-Hill.
- MYKE P. (2001). Programming and Customizing the PIC Microcontroller. McGraw-Hill. México.
- PALACIOS E. (2006). Microcontrolador PIC16F84, Desarrollo de proyectos México: Alfaomega Ra-Ma.
- TAFANERA A. (2000). Teoría y diseño con Microcontroladores PIC. México. Autores Editores.
- TORRES P. (1994). Microprocesadores y Microcontroladores Aplicados a la Industria. Madrid. Paraninfo.
- Microchip® (2019), PIC16F887 enero del 2019, de Microchip Sitio web: <http://ww1.microchip.com/downloads/en/DeviceDoc/30292D.pdf>

## 9. Posible Solución:

Lista de material sugerido

1. Microcontrolador: PIC16F887 o similar
2. Oscilador: Reloj de cuarzo de 20 MHz
3. Capacitores: Dos capacitores de 22pF
4. Luces indicadoras: Barra de LEDs



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES



Área: Electrónica

5. Botón de reinicio: Botón pulsador
6. Resistencias: 16 de 330  $\Omega$  y 17 de 1K $\Omega$
7. Resistencia variable: 2 Potenciómetros de 10K $\Omega$ .
8. Protoboard
9. Cables de colores: Calibre 22
10. Fuente de poder: 5V CD
11. Computadora: Programas de simulación y compilación
12. Grabador de microcontroladores.
13. Dos Motores DC
14. Controlador de motor: Puente H L293D

## DESARROLLO

a. Actividad I.

### Diseño del programa y circuito electrónico.

Generando una tabla de entradas y salidas para las conexiones al microcontrolador. Se consultó la hoja de especificaciones.

Entrada	BIT	PIN	Registro de configuración asociado, (Asignación de entradas analógicas)	Salida	BIT	PIN	Registro de configuración asociado, (Asignar "ceros" lógicos)
PORTA	0	2	TRISA	PORTC	0	15	TRISC
					1	16	
					2	17	
	3	18					
	4	23					
	5	24					
	1	3					

Tabla 9.6. Registros de entradas y salidas (versión 1) Fuente(s): Construcción propia, 2018.

El siguiente programa se configura el PWM para la manipulación de velocidad de dos motores DC en sentido horario y antihorario con la ayuda de un

puente H. La Figura 9.7 muestra el diagrama de flujo. A continuación, se muestra una breve descripción del programa diseñado.

1. Inicio.
2. Para configurar los puertos de entrada y de salidas con ayuda de los registros relacionados indicados por el fabricante. (Ver Tabla 9.1).
3. En el puerto C observamos los valores binarios utilizados para que los motores DC tenga un sentido de giro horario con una velocidad variable mediante el uso de potenciómetros.
4. Repite.



Figura 9.7. Diagrama de flujo del ejemplo de uso PWM para el control de un motor DC con velocidad de giro regulable (versión 1) Fuente(s): Construcción propia, 2018.

## b. Actividad II



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



## **Simulación del circuito.**

Se recomienda el uso de computadora y los programas necesarios para la simulación y la compilación del programa en lenguaje ensamblador.

El código mostrado en la Figura 9.2 se diseñó en el entorno de desarrollo de MPLAB para este documento.



# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



```
#include <16F887.h>
#define ADC=8
#define fuses XT,NOWDT,NOPROTECT,NOLVF, NOMCLR, NOCPD, NOBROWNOUT, NODEBUG, NOLVF, NOIESC, NOFCMEN
#define use delay(clock=4000000)

void main()
{
    //Bloque de configuracion
    int8 valor;
    setup_ccp1(CCP_PWM);           // Se configura el modulo CCP1 en modo PWM
    setup_ccp2(CCP_PWM);           // Se configura el modulo CCP2 en modo PWM

    setup_timer_2(T2_DIV_BY_16, 250, 1);

    set_pwm1_duty(0);               // Se fija el ciclo de trabajo a 0% en CCP1 y CCP2
    set_pwm2_duty(0);

    setup_adc_ports(ALL_ANALOG);    // Configuracion del ADC:
    setup_adc(ADC_CLOCK_INTERNAL); // Activar todas las entradas y fuente de reloj interna

    output_high(PIN_C0);            // Fijar sentido de M1
    output_low(PIN_C3);

    // Bloque principal
    while( TRUE )
    {
        // Control de M1 - Control de velocidad -
        set_adc_channel( 0 );        // Seleccionar AN0 como entrada al ADC
        delay_ms(10);
        valor = read_adc();          // Iniciar lectura del ADC
        set_pwm1_duty(valor);        // Fijar ciclo de trabajo

        // Control de M2 - Control de velocidad y sentido -
        set_adc_channel( 1 );        // Seleccionar AN1 como entrada al ADC
        delay_ms(10);
        valor = read_adc();          // Iniciar lectura del ADC

        if(valor > 127)
        {
            valor = (valor-128)*2;   // Calcular ciclo de trabajo
            set_pwm2_duty(valor);     // Fijar ciclo de trabajo
            output_high(PIN_C4);      // Girar en sentido horario
            output_low(PIN_C5);
        }
        else
        {
            valor+=2;                // Calcular ciclo de trabajo
            valor = 254 - valor;
            set_pwm2_duty(valor);     // Fijar ciclo de trabajo
            output_low(PIN_C4);       // Girar en sentido antihorario
            output_high(PIN_C5);
        }
    }
}
```

Figura 9.8. El programa en lenguaje C (versión 1) Fuente(s): Construcción propia, 2018.

Además, en la Figura 9.9 se muestra una captura de pantalla del funcionamiento del PIC en el ambiente de simulación Proteus.

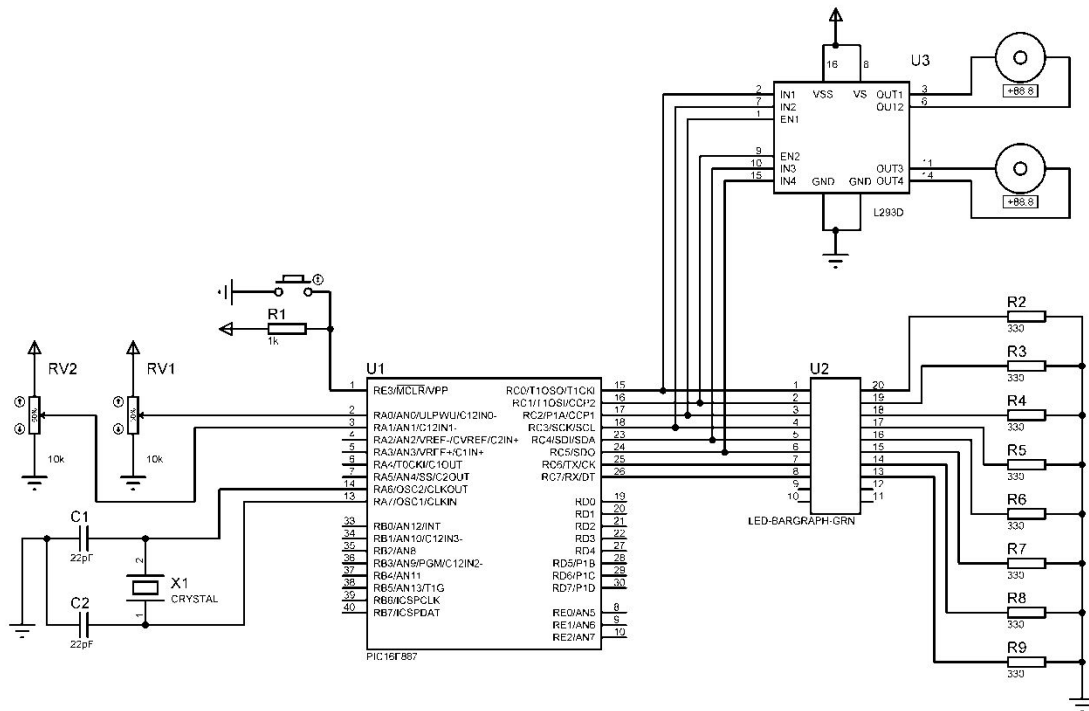


Figura 9.9. Circuito de la simulación, se observa que las salidas de la derecha muestran el movimiento de los motorreductores con una velocidad variable debido al valor analógico del potenciómetro (versión 1) Fuente(s): Construcción propia, 2018.

c. Actividad III  
**Alambrado del circuito.**

Se muestra en las figuras 9.10 un circuito armado de la solución con los componentes sugeridos.

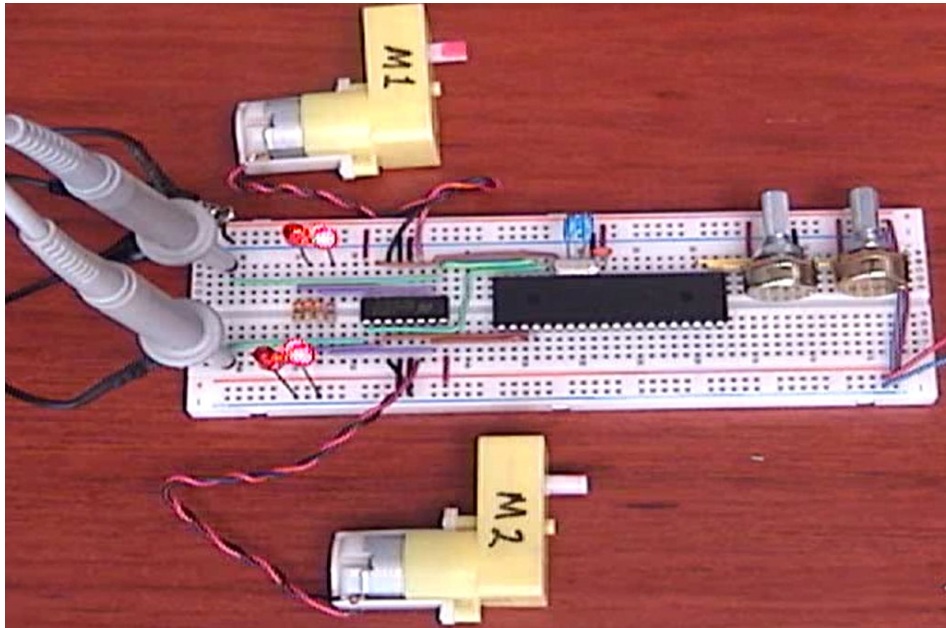


Figura 9.10. El circuito real de la práctica (versión 1) Fuente(s): Construcción propia, 2018.

### Resultados / Conclusión

Se implementó un sistema que permitió el control del sentido de giro y velocidad de dos motores de corriente directa, la última mediante la configuración del PWM del microcontrolador PIC16F887 en lenguaje de alto nivel.





# MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



## 10 AGRADECIMIENTOS

- Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE110618.
- Trabajo realizado con el apoyo de la Facultad de Estudios Superiores Aragón.

