



MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



Práctica 3. Saltos y Subrutinas con un Microcontrolador.

1. Objetivos de aprendizaje.

- a. Objetivo general.
2. Que el alumno obtenga conocimientos sobre el control de flujo de un programa para realizar saltos y subrutinas, utilizando el repertorio de instrucciones del microcontrolador.
- b. Objetivos específicos.
3. Aplicar la teoría del microcontrolador para ejecutar saltos condicionados y no condicionados usando registros de propósito general.
4. Diseñar el programa empleando diagramas de flujo.
5. Simular el sistema electrónico usando el programa diseñado.
6. Implementar el circuito físico basado en los puntos anteriores.

2. Introducción.

En esta práctica se estudia una serie de instrucciones que controlan el flujo de información dentro del programa. Una subrutina se describe como un conjunto de instrucciones que se encuentran separadas del programa principal y se pueden utilizar en varias ocasiones y existen diferentes tipos de subrutinas.

Existen diferentes tipos de saltos, pero en su mayoría producen una baja en el rendimiento del procesador ya que el procesador tiene que esperar a que se resuelva el salto, algunos pueden presentarse como retardos o como la ejecución de dos posibles caminos.

Existen saltos hacia atrás llamados en su mayoría *bucles* y saltos hacia delante corresponden más a estructuras *if-then-else*, son empleados en sistemas



MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



automatizados, donde se requiere que existan respuestas ante ciertas condiciones externas que modifican el estado del microcontrolador y normalmente se manifiesta al exterior usando actuadores.

3. Equipo y material.

- Microcontrolador con al menos dos pines disponibles (uno para entrada y otro para salida de datos).
- Reloj de cuarzo (oscilador).
- Capacitores.
- Botón pulsador (para reinicio del microcontrolador).
- Interruptores (uno al menos para la entrada).
- Resistencias.
- Luces indicadoras (una al menos para la salida).
- Protoboard y cables.
- Fuente de alimentación.

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 o salidas. Adicionalmente se solicita que genere un programa donde 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, oscilador, fuente de



MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



alimentación, 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 se cumplan los valores de entrada en comparación a los de salida.

7. Aplicaciones.

El control de flujo de programa se usa para ejecutar un sin fin de algoritmos de control, tanto en proyectos electrónicos como en mecatrónicos. Los saltos condicionados e incondicionados se usan para subrutinas, manejo de información, respuestas automáticas, entre otras.

8. Bibliografía.

- Enrique Palacios Municio. Fernando Remiro Domínguez, Lucas J. López Pérez. (2004). Microcontrolador PIC16F84 Desarrollo de proyectos. México: Alfaomega.
- Ramón Pallas Areny. (2007). Microcontroladores: fundamentos y aplicaciones con PIC. México: Alfaomega.
- Microchip® (2009), PIC16F887 enero del 2009, de Microchip Sitio web: <http://ww1.microchip.com/downloads/en/DeviceDoc/41291F.pdf>.



MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



- Rito Mijarez Castro. (2014). Electrónica. PATRIA.
<https://books.google.com.mx/books?id=S-jhBAAAQBAJ&pg=PA254&lpg=PA254&dq=#v=onepage&q&f=false>.

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. Interruptores: deslizables de ocho palancas
5. Luces indicadoras: barra de LEDs
6. Botón de reinicio: botón pulsador
7. Resistencias: 16 de 330 Ω y 17 de 1K Ω
8. Tarjeta de desarrollo: Protoboard
9. Cables de colores: preferente de calibre 22
10. Fuente de poder: 5V CD
11. Computadora: Programas de simulación y compilación
12. Grabador de microcontroladores.

En esta práctica se estudia una serie de instrucciones que controlan el flujo de información dentro del programa. En primer lugar, se estudia la instrucción de *salto incondicionado*, para lo cual se usa la instrucción *GOTO* que lleva de una línea del programa a otra direccionada por una etiqueta o bien se le asigna un valor al contador del programa, para ello se emplea el símbolo \$. Asimismo, existe la opción de hacer subrutinas dentro de los programas para reducir código repetitivo y se usan las instrucciones *CALL / RETURN*, que son para llamar una subrutina y regresar al programa principal, respectivamente. Ver la Tabla 3.1.

Tabla 3.1. Ejemplos de instrucciones de salto incondicionado, las flechas indican el flujo del programa.

Etiquetas	Instrucciones/operandos	Comentarios
Inicio	BSF PORTD, 0	enciende pin cero del Puerto D
	CALL Subrutina	llama a la subrutina
	GOTO Inicio	salta a Inicio
...	...	otras instrucciones
Subrutina	BCF PORTD, 0	apaga pin cero del Puerto D



MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



	RETURN	Regresa de la subrutina a la instrucción posterior inmediata de donde fue llamada la subrutina.
--	--------	---

Existe el salto condicionado, que se usa para ejecutar una sección del código u otro, dependiendo de condiciones internas (banderas) o externas (sensores) del microcontrolador. Las instrucciones para el PIC son: BTFSS, BTFSC, INCFSZ y DECFSZ.

Tabla 3.2. Ejemplos de salto condicionado, hay que considerar que se tiene un sensor en el pin 7 del Puerto B.

Etiquetas	Instrucciones/operandos	Comentarios
Inicio	BTFSS PORTB, 7	pregunta si el pin siete del puerto D es uno.
	GOTO Inicio	NO, el sensor asociado al PB7 está apagado, salta a Inicio.
	GOTO Continua	SI, el sensor asociado al PB7 está encendido, salta a Continua.
	...	Otras instrucciones.
Continua	...	Otras instrucciones.

*NOTA: la flecha continua es para el caso de que el sensor está encendido y la discontinua es cuando está apagado.

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, (Asignar	Salida	BIT	PIN	Registro de configuración asociado, (Asignar
---------	-----	-----	--	--------	-----	-----	--

MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica

			"unos" lógicos).				"ceros" lógicos).
PORTB	0	33	TRISB	PORTD	0	19	TRISD
					1	20	
					2	21	
					3	22	
					4	27	
					5	28	
					6	29	
					7	30	

Tabla 3.3. Registros de entradas y salidas (versión 1), Fuente(s): Construcción propia, 2018.

El siguiente programa sirve para rotar un bit de un puerto ya sea de derecha a izquierda o por el contrario de izquierda a derecha mediante una selección realizada en un pin del puerto B. El resultado de la operación se despliega en el Puerto D a través de 8 luces, en este caso con una barra de LED (Tabla 3.3). La Figura 3.1 muestra el diagrama de flujo. A continuación, se muestra una breve descripción del programa diseñado.

1. Inicio.
2. Para configurar los puertos como entradas o como salidas con ayuda de los registros relacionados indicados por el fabricante. (Tabla 3.3).
3. El valor binario del Puerto B de entrada se lee y se selecciona el sentido de la rotación, la cual se despliega en el Puerto D.
4. Repite.

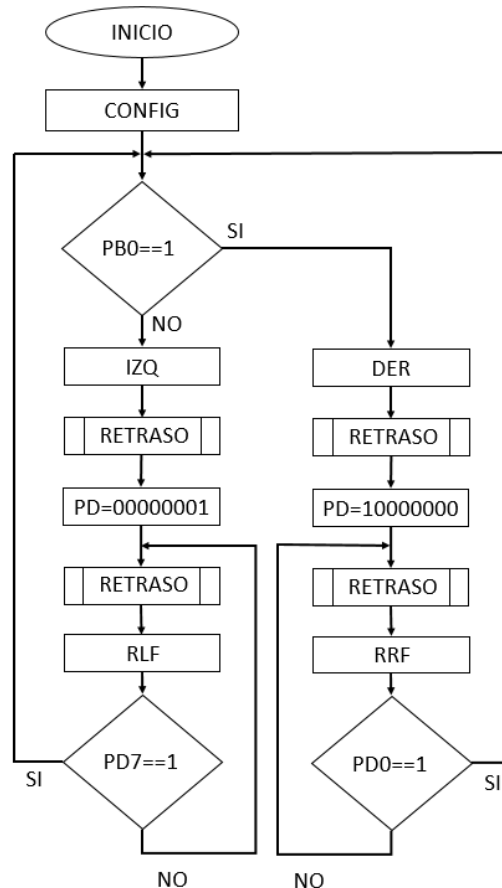


Figura 3.1. Diagrama de flujo del ejemplo de selección de rotación Izquierda-Derecha de un puerto (versión 1) Fuente(s): Construcción propia, 2018.

b. Actividad II.

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 3.2 se diseñó en el entorno de desarrollo de MPLAB para este documento.



MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



```
#include <pl6F887.inc>

__CONFIG __CONFIG1, _MCLRE_OFF & _WDI_OFF & _LVP_OFF & _CPD_OFF & _CP_OFF & _BOR_OFF & _FWRTE_ON & _FCMEN_OFF & _IESO_OFF & _DEBUG_OFF & _HS_OSC

DELAY1 EQU 0x20
DELAY2 EQU 0x21
DELAY3 EQU 0x22

ORG 0x00

SETUP
    BSF STATUS,RP0 ; Accedemos al Banco 1
    BCF STATUS,RP1
    CLRF TRISD ; Volvemos salidas al puerto D
    MOVLW 0xFF
    MOVWF TRISE ; Volvemos entradas al puerto B
    BSF STATUS,RP1 ; Accedemos al Banco 3
    CLRF ANSELH ; Volvemos digitales los puertos del ANSELH
    BCF STATUS,RP0 ; Accedemos al Banco 0
    BCF STATUS,RP1
    CLRF PORTD ; Limpiamos el Puerto D

INICIO
    BTFSS PORTB,0 ; Leemos el bit 0 del puerto B y preguntamos si esta en alto
    IZQ ; Salta a la etiqueta IZQ
    GOTO DER ; Salta a la etiqueta DER

IZQ
    CALL RETRASO ; Llama a la etiqueta Retraso
    MOVLW B'00000001'
    MOVWF PORTD ; Le asigna el valor binario 00000001 al puerto D

LOOPL
    CALL RETRASO ; Llama a la etiqueta retraso
    RLF PORTD,f ; Aplica un registro de corrimiento hacia la izquierda al Puerto D
    BTFSS PORTD,7 ; Leemos el bit 7 del puerto D y preguntamos si esta en alto
    GOOPL ; Salta a la etiqueta LOOPL
    GOTO INICIO ; Salta a la etiqueta INICIO

DER
    CALL RETRASO ; Llama a la etiqueta retraso
    MOVLW B'10000000'
    MOVWF PORTD ; Le asigna el valor binario 10000000 al puerto D

LOOPR
    CALL RETRASO ; Llama a la etiqueta retraso
    RRF PORTD,f ; Aplica un registro de corrimiento hacia la derecha al Puerto D
    BTFSS PORTD,0 ; Leemos el bit 0 del puerto B y preguntamos si esta en alto
    GOOPR ; Salta a la etiqueta LOOPR
    GOTO INICIO ; Salta a la etiqueta INICIO

RETRASO
    MOVLW .100
    MOVWF DELAY1 ; Le asigna un valor de 100 al Delay 1
    MOVLW .50
    MOVWF DELAY2 ; Le asigna el valor de 50 al Delay 2
    MOVLW .13
    MOVWF DELAY3 ; Le asigna el valor de 13 al Delay 3

DELAY_LOOP
    DECFSZ DELAY1,f ; Decrementa el registro y pregunta si ha llegado a cero, en caso
    ; positivo salta una instruccion
    GOTO DELAY_LOOP ; Salta a la instruccion DELAY_LOOP
    DECFSZ DELAY2,f ; Decrementa el registro y pregunta si ha llegado a cero, en caso
    ; positivo salta una instruccion
    GOTO DELAY_LOOP ; Salta a la instruccion DELAY_LOOP
    DECFSZ DELAY3,f ; Decrementa el registro y pregunta si ha llegado a cero, en caso
    ; positivo salta una instruccion
    GOTO DELAY_LOOP ; Salta a la instruccion DELAY_LOOP
    RETURN
END
```

Figura 3.2. El programa en ensamblador (versión 1) Fuente(s): Construcción propia, 2018.

Además, en la Figura 3.3 se muestra una captura de pantalla del funcionamiento del PIC en el ambiente de simulación Proteus.

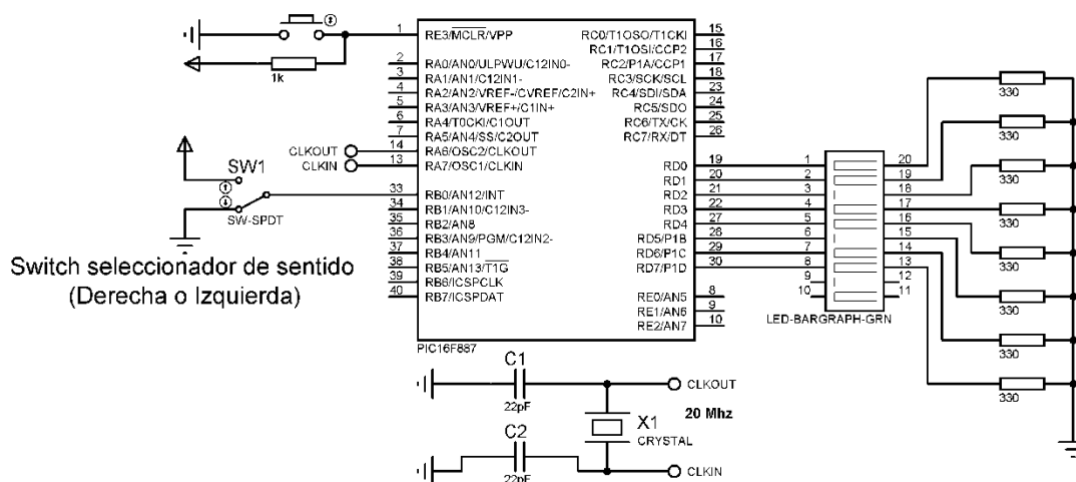


Figura 3.3. Circuito de la simulación, se observa que las salidas de la derecha reflejan el sentido de rotación que se ingresa por el puerto asociado al interruptor de la izquierda (versión 1), Fuente(s): Construcción propia, 2018.

c. Actividad III.

Alambrado del circuito.

Se muestra en la figura 3.4 un circuito armado de la solución con los componentes sugeridos.

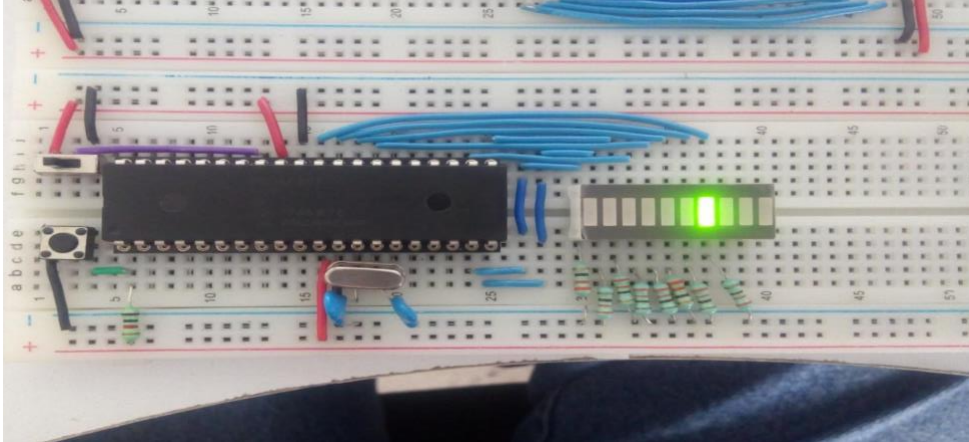


Figura 3.4. El circuito real de la práctica (versión 1) Fuente(s): Construcción propia, 2018.

Resultados / Conclusión.

Se configuraron dos puertos de un microcontrolador, uno como entradas y el otro como salidas, se ingresó la información y se selecciona el sentido de la rotación, la cual se despliega en las salidas del microcontrolador por el puerto D.

Ejercicio adicional: Registro de Corrimiento Izquierda-Derecha Automático.

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, (Asignar "unos" lógicos).	Salida	BIT	PIN	Registro de configuración asociado, (Asignar "ceros" lógicos).

MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica

PORTB	0	33	TRISB	PORTD	0	19	TRISD
					1	20	
					2	21	
					3	22	
					4	27	
					5	28	
					6	29	
					7	30	

Tabla 3.4. Registros de entradas y salidas (versión 1) Fuente(s): Construcción propia, 2018.

El siguiente programa sirve para rotar un bit de un puerto de derecha a izquierda y posteriormente de izquierda a derecha de forma automática. El resultado de la operación se despliega en el Puerto D a través de 8 luces, en este caso con una barra de LED (Tabla 3.4). La Figura 3.5 muestra el diagrama de flujo. A continuación, se muestra una breve descripción del programa diseñado.

1. Inicio.
2. Para configurar los puertos como entradas o como salidas con ayuda de los registros relacionados indicados por el fabricante. (Ver Tabla 3.4).
3. La rotación automática se despliega en el Puerto D.
4. Repite.

MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica

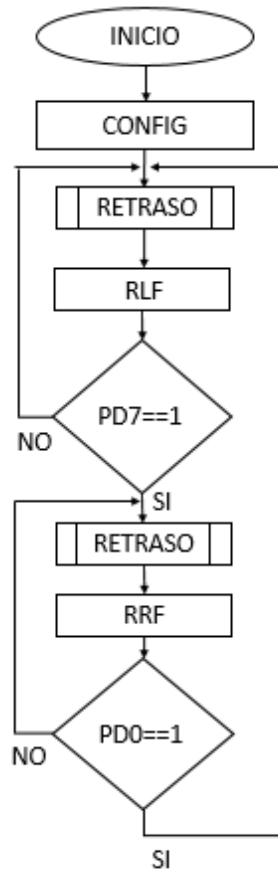


Figura 3.5. Circuito de la simulación, se observa que las salidas de la derecha reflejan el sentido de rotación automático de izquierda a derecha y de derecha a izquierda (versión 1) Fuente(s): Construcción propia, 2018.

b. Actividad II.

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 3.6 se diseñó en el entorno de desarrollo de MPLAB para este documento.



MANUAL DE PRÁCTICAS DE MICROCONTROLADORES

Área: Electrónica



```
#include <p16F887.inc>
    _CONFIG _CONFIG1, 0x20F2
    _CONFIG _CONFIG2, 0x3EFF

DELAY1 EQU 0x20
DELAY2 EQU 0x21
DELAY3 EQU 0x22

ORG 0x00

SETUP
    BSF STATUS,RP0 ; Accedemos al Banco 1
    BCF STATUS,RP1
    CLRF TRISD ; Volvemos salidas al puerto D
    MOVLW 0xFF
    MOVWF TRISB ; Volvemos entradas al puerto B
    BSF STATUS,RP1 ; Accedemos al Banco 3
    CLRF ANSELH ; Volvemos digitales los puertos del ANSELH
    BCF STATUS,RP0 ; Accedemos al Banco 0
    BCF STATUS,RP1
    CLRF PORTD ; Limpiamos el Puerto D
    BSF PORTD,0

LOOPL
    CALL RETRASO ; Llama a la etiqueta Retraso
    RLF PORTD,f ; Aplica un registro de corrimiento hacia la izquierda al Puerto D
    BTFSS PORTD,7 ; Leemos el bit 7 del puerto D y preguntamos si esta en alto
    GOTO LOOPL ; Salta a la etiqueta LOOPL
    GOTO LOOPR ; Salta a la etiqueta LOOPR

LOOPR
    CALL RETRASO ; Llama a la etiqueta Retraso
    RRF PORTD,f ; Aplica un registro de corrimiento hacia la derecha al Puerto D
    BTFSS PORTD,0 ; Leemos el bit 7 del puerto D y preguntamos si esta en alto
    GOTO LOOPR ; Salta a la etiqueta LOOPR
    GOTO LOOPL ; Salta a la etiqueta LOOPL

RETRASO
    MOVLW .100
    MOVWF DELAY1 ; Le asigna un valor de 100 al Delay 1
    MOVLW .50
    MOVWF DELAY2 ; Le asigna el valor de 50 al Delay 2
    MOVLW .13
    MOVWF DELAY3 ; Le asigna el valor de 13 al Delay 3

DELAY_LOOP
    DECFSZ DELAY1,f ; Decrementa el registro y pregunta si ha llegado a cero, en caso
    ; positivo salta una instruccion
    GOTO DELAY_LOOP ; Salta a la instruccion DELAY_LOOP
    DECFSZ DELAY2,f ; Decrementa el registro y pregunta si ha llegado a cero, en caso
    ; positivo salta una instruccion
    GOTO DELAY_LOOP ; Salta a la instruccion DELAY_LOOP
    DECFSZ DELAY3,f ; Decrementa el registro y pregunta si ha llegado a cero, en caso
    ; positivo salta una instruccion
    GOTO DELAY_LOOP ; Salta a la instruccion DELAY_LOOP
    RETURN

END
```

Figura 3.6. El programa en ensamblador (versión 1) Fuente(s): Construcción propia, 2018.

Además, en la Figura 3.7 se muestra una captura de pantalla del funcionamiento del PIC en el ambiente de simulación Proteus.

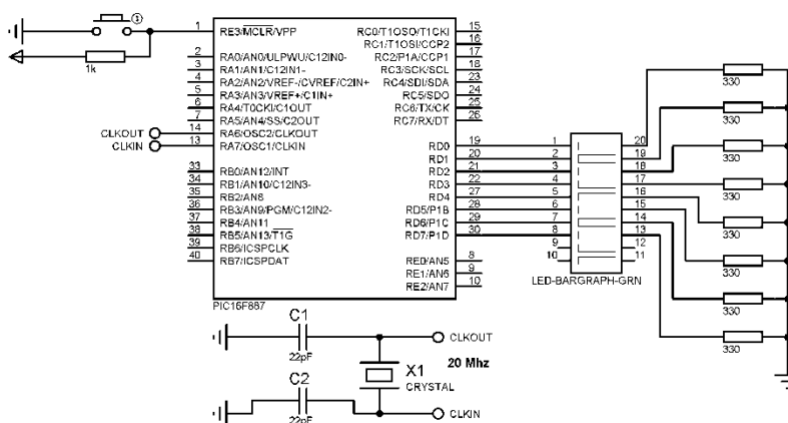


Figura 3.7. Circuito de la simulación, se observa que las salidas de la derecha reflejan la rotación automática solicitada (versión 1) Fuente(s): Construcción propia, 2018.

c. Actividad III.

Alambrado del circuito.

Se muestra en la figura 3.8 un circuito armado de la solución con los componentes sugeridos.

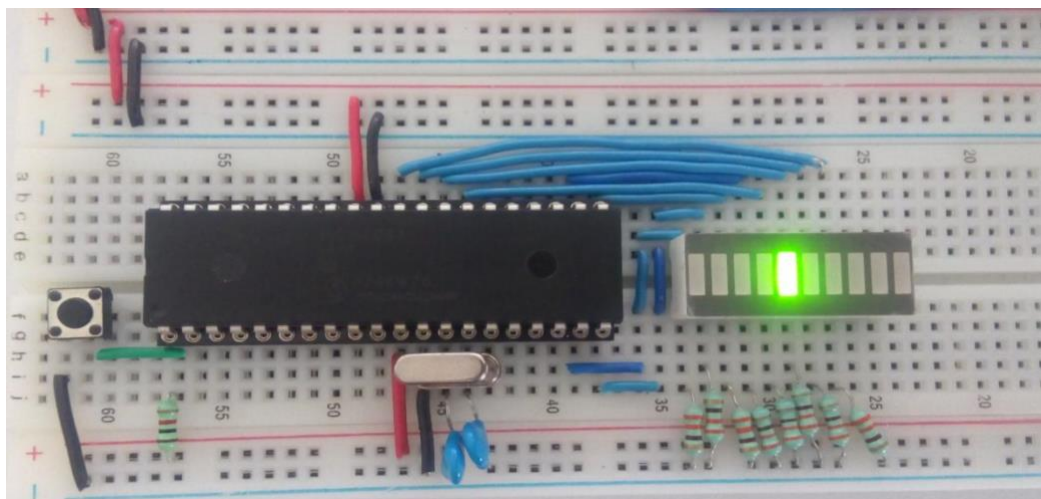


Figura 3.8. El circuito real de la práctica (versión 1) Fuente(s): Construcción propia, 2018.



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.

