



FACULTAD DE CIENCIAS EXACTAS,  
FÍSICAS Y NATURALES

ELECTRÓNICA DIGITAL 2

CONTROL DE PROXIMIDAD  
MEDIANTE USO DE SENSOR  
ULTRASÓNICO HC-SR04

*Zimmel Ceccón, Ezequiel José*  
email [ezequielzimmel@gmail.com](mailto:ezequielzimmel@gmail.com)

*Docente: Ing. Martín Ayarde*

ÚLTIMA REVISIÓN: 17 DE JUNIO DE 2018

# INFORME TRABAJO FINAL

## ÍNDICE

|  |    |
|--|----|
| 1. Descripción general                 | 3  |
| 1.1. Introducción                      | 3  |
| 1.2. Elementos empleados               | 3  |
| 2. Desarrollo                          | 4  |
| 2.1. Requerimientos y Especificaciones | 4  |
| 2.2. Grabación del PIC                 | 14 |
| 2.3. Implementación                    | 15 |

## ABSTRACT

Hoy en día existen numerosos dispositivos que utilizan ultrasonidos, tanto en el área industrial, como en la medicina y muchas otras aplicaciones no tan conocidas. En este proyecto vamos a explorar las posibilidades del sensor HC-SR04, un sensor ultrasónico de distancia de bajo costo, usado comúnmente en robots, pero no necesariamente limitado a ese uso. En el circuito de prueba del proyecto utilizaremos un microcontrolador PIC16F887 que medirá el tiempo del pulso ECHO. Después calcularemos la distancia y se mostrará a través de displays 7-SEG. A su vez, la misma información será transmitida por serie a un ordenador.

El documento desarrolla todas las instancias necesarias para lograr implementar un proyecto empleando microcontroladores PIC. Las instancias del documento abarcan desde el detallado de los elementos a emplear, pasando por la grabación del microcontrolador, hasta la muestra del proyecto montado y funcionando sobre una protoboard.

## 1 DESCRIPCIÓN GENERAL

### 1.1 Introducción

Los sensores ultrasónicos modernos para la medición de distancias, son de bajo costo y permiten su acoplamiento a sistemas microcontroladores. Estos sensores cuentan con una pequeña bocina emisora del sonido y un micrófono receptor de la señal ultrasónica rebotada por el objeto del cual se desea medir la distancia.

El principio de funcionamiento es el siguiente: a través de la bocina se envían trenes de pulsos de ultrasonido a una frecuencia 40Khz. Esta señal rebota contra el objeto y regresa al sensor, siendo detectada por el micrófono (Figura 1). La distancia se calcula considerando el tiempo (en microsegundos) del ancho del pulso de la señal Echo, y dividida por 58 para obtener el valor en centímetros.

$$\text{Distancia[cm]} = \frac{\text{Ancho del pulso Echo}[\mu\text{s}]}{58} \quad (1)$$

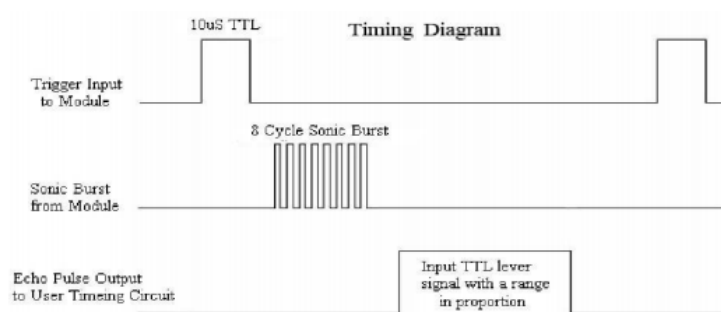


Figura 1: Funcionamiento

El sensor **HC-SR04**, con capacidad de medir distancias entre 2 cms y 4 metros, maneja 2 señales de control conectadas al microcontrolador, en este caso el PIC16F887. La primera es el 'Trigger', que es un pulso con 10µs (mínimo) de duración que dispara la emisión de 8 pulsos de ultrasonido a través de la bocina, como se muestra en la Figura 1. La segunda señal, llamada 'Echo', genera un pulso cuya duración es igual al tiempo de respuesta de la señal ultrasónica rebotada por el objeto a medir.

### 1.2 Elementos empleados

Para el desarrollo del proyecto se precisa el uso de:

- Programador PIC y software PICkit con soporte para el microcontrolador.
- Microcontrolador **PIC16F887**.
- Protoboard(s).
- 1 Buzzer piezoelectrico. [Hoja de datos](#).
- 1 sensor ultrasónico HC-SR04. [Hoja de datos](#).
- 3 pulsador Switch.
- Resistencias varias.
- 2 diplay 7-SEG x4 Anodo Común. [Hoja de datos](#).
- 6 transistores PNP de baja señal. [Hoja de datos](#).
- 1 LED RGB. [Hoja de datos](#).
- 1 módulo USB-UART CP2102. [Hoja de datos](#).

## 2 DESARROLLO

En esta sección se describirá lo que se espera que el sistema haga (Requerimientos), y como fueron implementados dichos requerimientos (Especificaciones). A su vez, se detallarán los cálculos de los elementos pasivos del circuito, y se abordan los pasos necesarios para la grabación del microcontrolador.

### 2.1 Requerimientos y Especificaciones

A partir del siguiente diagrama funcional, se desarrollan las correspondientes subsecciones.

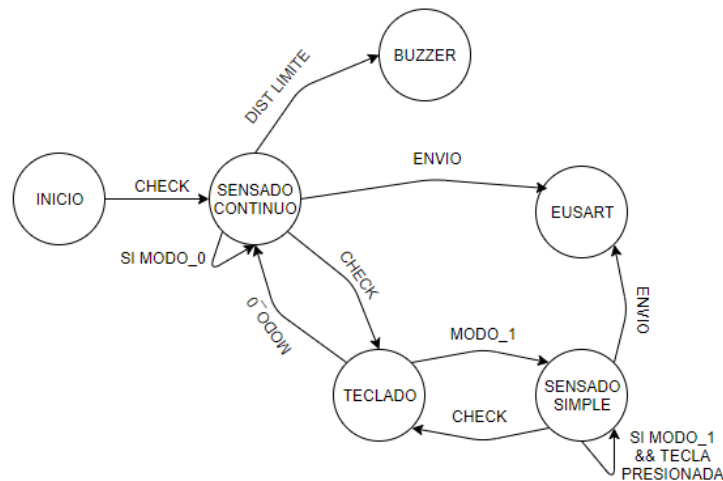


Figura 2: Funcionamiento

#### 2.1.1 Sensado de distancia

Para la adquisición de la distancia se emplea la ecuación 1, mediante el uso de interrupciones por RB0 y TMR0.

Lo primero que se hace es realizar la configuración del registro INTCON (Interrupciones) y OPTION\_REG (Flanco de interrupción de RB0 y PS).

```

MOVLW    B'11010000'
MOVWF    OPTION_REG                ;RBo por flanco ascendente. PS 1:2
MOVLW    B'10010000'
MOVWF    INTCON                    ;GIE (ON), ToIE (OFF), INTE (ON)
  
```

Se elige interrumpir RB0 por flanco ascendente para detectar el frente del pulso ECHO. A partir de esta interrupción, se inicializa el TMR0 y se habilita dicha interrupción, a los fines de contabilizar los centímetros de distancia al objetivo. Por otro lado, el flanco de interrupción de RB0 se cambia a flanco descendente, para detectar la cola del pulso ECHO, y a partir de dicho instante deshabilitar la interrupción por TMR0.

Las interrupciones son gestionadas por el vector de interrupción, en donde se chequean las banderas de las interrupciones y se redirecciona a la subrutina de atención correspondiente.

```

INT_VECTOR
    ;Guardamos el contexto
    MOVLW    W_TEMP
    SWAPF    STATUS,W
    MOVWF    S_TEMP
  
```

```

;Chequeamos banderas de interrupcion
BTFSC    INTCON,INTF
CALL     RBo_INT
BTFSC    INTCON,ToIF
CALL     TMRo_INT
;Recuperamos el contexto
SWAPF    S_TEMP,W
MOVF     STATUS
SWAPF    W_TEMP,F
SWAPF    W_TEMP,W
RETFIE

```

La subrutina de RBo chequea el estado actual del Pin de entrada al microcontrolador desde el Pin ECHO del sensor. Si el estado es Alto (1 lógico), inicializa el TMR0, habilita su interrupción y cambia su flanco de interrupción pasando de flanco ascendente a descendente. Por el contrario, si el estado es Bajo (0 lógico), deshabilita el TMR0 y cambia su flanco de interrupción de descendente a ascendente.

```

RBo_INT
    BCF     INTCON,INTF
    BTFSS    ECO
    GOTO    DETENER_TIMER
;Cargo al TMRo para interrupcion de 58us
MOVLW    .198
MOVF     TMRo
;Habilitamos (inicia) el TMRo
BSF      INTCON,ToIE
;Reseteamos los registros asociados a los displays
CLRF     UNIDADES
CLRF     DECENAS
CLRF     CENTENAS
;Activacion de RBo por flanco descendiente
BANKSEL  OPTION_REG
BCF      OPTION_REG,6
BANKSEL  PORTA
RETURN

DETENER_TIMER
;Se detiene al TMRo
BCF      INTCON,ToIE
;Se levanta la bandera de adquisicion de distancia
BSF      ECO_DONE
;Activacion de RBo por flanco ascendente
BANKSEL  OPTION_REG
BSF      OPTION_REG,6
BANKSEL  PORTA
RETURN

```

Con cada interrupción del TMR0 se incrementan, según correspondan, los registros asociados al almacenaje de la distancia sensada. El valor de carga del TMR0 garantiza que se produzcan interrupciones cada 58μs. El valor se obtiene a partir del siguiente cálculo:

$$58\mu s = \frac{4}{F_{osc}} * (256 - \text{Carga}) * PS = \frac{4}{8\text{MHz}} * (256 - 198) * 2$$

*Nota: Los cálculos anteriores son teóricos, en la práctica se ajusto el valor de Carga para mayor exactitud en la medición.*

```

TMRo_INT
    BCF      INTCON ,ToIF

```

```

MOVLW    .198
MOVWF    TMRo
INCF     UNIDADES

MOVLW    0x0A
SUBWF    UNIDADES,W
BTFSS    STATUS,Z
RETURN

CLRF     UNIDADES
INCF     DECENAS

MOVLW    0x0A
SUBWF    DECENAS,W
BTFSS    STATUS,Z
RETURN

CLRF     DECENAS
INCF     CENTENAS

MOVLW    0x0A
SUBWF    CENTENAS,w
BTFSS    STATUS,Z
RETURN

CLRF     CENTENAS
RETURN

```

### 2.1.2 Seteo de distancia máxima

El seteo de la distancia máxima permite al usuario definir la distancia a partir de la cual el sistema dejará de activar la alerta visual y sonora de proximidad. Esto es llevado a cabo por la rutina de chequeo de botones, y en función de que botón es accionado los registros que almacenan la distancia máxima son incrementado o disminuidos.

```

CHECK_BOTON
    BTFSC    PORTD, RD5
    GOTO     RD_5
    BTFSC    PORTD, RD6
    GOTO     RD_6
    BTFSC    PORTD, RD7
    GOTO     RD_7
    RETURN

RD_5
    BTFSC    PORTD, RD5
    GOTO     RD_5
    BSF      SIN_SHOT
    RETURN

RD_6
    BTFSC    PORTD, RD6
    GOTO     RD_6
    INCF     MD_UNI, F
    BCF      DEC_FLAG, 0
    MOVLW    .10
    XORWF    MD_UNI,W
    BTFSS    STATUS,Z
    RETURN

    CLRF     MD_UNI
    INCF     MD_DEC, F
    BCF      DEC_FLAG, 1
    MOVLW    .10

```

```

XORWF MD_DEC,W
BTFSS STATUS,Z
RETURN
CLRF MD_DEC
INCF MD_CEN,F
BCF DEC_FLAG,2
MOVLW .10
XORWF MD_CEN,W
BTFSS STATUS,Z
RETURN
CLRF CENTENAS
RETURN

```

RD\_7

```

BTFSC PORTD,RD7
GOTO RD_7
BTFSC DEC_FLAG,0
RETURN
MOVF MD_UNI,W
XORLW .0
BTFSC STATUS,Z
GOTO CHECK_DEC
DECF MD_UNI,F
RETURN

```

CHECK\_DEC

```

BTFSC DEC_FLAG,1
GOTO CLR_MD_UNI
MOVLW .9
MOVWF MD_UNI
DECF MD_DEC,F
MOVLW 0xFF
XORWF MD_DEC,W
BTFSS STATUS,Z
RETURN

```

```

BTFSC DEC_FLAG,2
GOTO CLR_MD_DEC
MOVLW .9
MOVWF MD_DEC
DECF MD_CEN,F
MOVLW 0xFF
XORWF MD_CEN,W
BTFSS STATUS,Z
RETURN

```

CLR\_MD\_CEN

```

BSF DEC_FLAG,2
MOVLW 0x00
MOVWF MD_CEN
MOVWF MD_DEC
MOVWF MD_UNI
RETURN

```

CLR\_MD\_DEC

```

BSF DEC_FLAG,1
MOVLW 0x00
MOVWF MD_UNI

```

```

        MOVWF MD_DEC
        RETURN
CLR_MD_UNI
        BSF    DEC_FLAG,0
        MOVLW  0x00
        MOVWF  MD_UNI
        RETURN

```

### 2.1.3 *Despliegue de datos por displays 7-SEG*

La multiplexación de los displays 7-SEG, permite mostrar la distancia sensada y la distancia máxima configurada por el usuario. Los valores almacenados en los registros de distancia se encuentran en BCD por lo que, mediante una tabla de conversión, se obtiene el correspondiente valor 7-SEG. A su vez, la subrutina de RETARDO garantiza que el tiempo que permanece encendido cada display, simule que están todos encendidos.

```

MOSTRAR
        MOVLW  b'00111000'
        MOVWF  PORTB
        MOVLW  UNIDADES
        MOVWF  FSR
        MOVF   INDF,W
        CALL   CONVIERTE
        BCF    PORTB,5
        MOVWF  PORTA
        CALL   RETARDO
        BSF    PORTB,5
        INCF   FSR
        MOVF   INDF,W
        CALL   CONVIERTE
        BCF    PORTB,4
        MOVWF  PORTA
        CALL   RETARDO
        BSF    PORTB,4
        INCF   FSR
        MOVF   INDF,W
        CALL   CONVIERTE
        BCF    PORTB,3
        MOVWF  PORTA
        CALL   RETARDO
        BSF    PORTB,3
        INCF   FSR

MOSTRAR_MD
        MOVF   INDF,W
        CALL   CONVIERTE
        BCF    PORTC,5
        MOVWF  PORTA
        CALL   RETARDO
        BSF    PORTC,5
        INCF   FSR
        MOVF   INDF,W
        CALL   CONVIERTE
        BCF    PORTC,4
        MOVWF  PORTA
        CALL   RETARDO
        BSF    PORTC,4

```



```

INCF    FSR
MOVF    INDF,W
CALL    CONVIERTE
BCF     PORTC,3
MOVWF   PORTA
CALL    RETARDO
BSF     PORTC,3
RETURN

```

#### 2.1.4 *Cambio entre modos de uso*

Cuando se preciona la tecla correspondiente (RD5), se pone en 1 la bandera del 'Modo Simple', que será chequeada en el bloque de rutina principal, accediendo a la rutina SINGLE\_SHOT. En esta rutina se habilitan 2 botones, el primero permite hacer una medición de distancia, y el otro pone en 0 la bandera del 'Modo Simple', recupera el estado previo a este modo y continua con la ejecución continua del sistema.

```

SINGLE_SHOT
    MOVF    MD_UNI,W
    MOVWF   MD_UNI_C
    MOVF    MD_DEC,W
    MOVWF   MD_DEC_C
    MOVF    MD_CEN,W
    MOVWF   MD_CEN_C

ESPERA
    MOVLW   .10
    MOVWF   MD_UNI
    MOVWF   MD_DEC
    MOVWF   MD_CEN
    MOVLW   MD_UNI
    MOVWF   FSR
    CALL    MOSTRAR_MD
    BTFSC   PORTD,RD7
    CALL    DISPARO_
    BTFSS   PORTD,RD5
    GOTO    ESPERA

SALIR_
    BTFSC   PORTD,RD5
    GOTO    SALIR_
    MOVLW   UNIDADES
    MOVWF   FSR
    MOVF    MD_UNI_C,W
    MOVWF   MD_UNI
    MOVF    MD_CEN_C,W
    MOVWF   MD_CEN
    MOVF    MD_DEC_C,W
    MOVWF   MD_DEC
    BCF     SIN_SHOT
    RETURN

DISPARO_
    CALL    SENSADO
    BTFSC   PORTD,RD7
    CALL    MOSTRAR
    BTFSC   PORTD,RD7
    GOTO    $-3

```

CALL     ENVIAR  
RETURN

### 2.1.5 *Aviso de proximidad al objeto sentido*

Se realiza la comparación, en binario, de la distancia sensada y el valor de distancia máximo ingresado por el usuario. Si la distancia sensada es mayor al máximo, el sistema desactiva todo tipo de alerta (OFF\_); en caso contrario, procede de la siguiente manera:

- Calcula el Rango de distancia, haciendo la diferencia entre el valor máximo ingresado por el usuario y un valor mínimo prefijado (en nuestro caso 5).
- A partir del Rango y con un valor prefijado de cantidad de intervalos (en nuestro caso 5), calcula el intervalo menor (representando unidades de distancia [cm]), a partir del cual, mediante suma de si mismo, obtiene los restante puntos de corte. Ej: Si tenemos que la distancia máxima es 50[cm], tendríamos que  $Rango = (50 - 5) = 45[cm]$ , luego el intervalo mínimo es 9[cm], quedando los puntos de corte en [9, 18, 27, 36, 45].
- Con los puntos de cortes ya definidos, se procede a comprar la distancia sensada con cada nivel de corte. Si la distancia es mayor al primer nivel de corte (según ejemplo 9[cm]), se procede a comprar con el segundo nivel de corte (según ejemplo 18[cm]), continuando hasta el último punto de corte. Siguiendo con el ejemplo, en caso de que la distancia sensada sea mayor a 18[cm] pero menor a 27[cm], el sistema accede a una subrutina asociada a ese intervalo, en donde se setean los valores de RETARDO y se activa la correspondiente alerta visual.

#### COMPARAR\_DISTANCIA

```
CALL    BCD2BIN
MOVF    BCD_CONV,W
SUBWF   MD_BCD_C,W
BTFSS   STATUS,C
GOTO    OFF_

MOVLW   .5
SUBWF   MD_BCD_C,W
MOVWF   RANGO
MOVWF   TEMP
INCF    INTERVALO,F
MOVLW   .5
SUBWF   TEMP,F
BTFSC   STATUS,C
GOTO    $-4
DECF    INTERVALO,F
MOVF    INTERVALO,W
ADDLW   .8
MOVWF   TEMP

MOVLW   .8
SUBWF   BCD_CONV,W
BTFSS   STATUS,C
GOTO    NIV_5

MOVF    BCD_CONV,W
SUBWF   TEMP,W
```

```

    BTFSC    STATUS,C
    GOTO     NIV_4
    MOVF     INTERVALO,W
    ADDWF    TEMP,F

    MOVF     BCD_CONV,W
    SUBWF    TEMP,W
    BTFSC    STATUS,C
    GOTO     NIV_3
    MOVF     INTERVALO,W
    ADDWF    TEMP,F

    MOVF     BCD_CONV,W
    SUBWF    TEMP,W
    BTFSC    STATUS,C
    GOTO     NIV_2
    MOVF     INTERVALO,W
    ADDWF    TEMP,F

    MOVF     BCD_CONV,W
    SUBWF    TEMP,W
    BTFSC    STATUS,C
    GOTO     NIV_1
    MOVF     INTERVALO,W
    ADDWF    TEMP,F

    MOVF     BCD_CONV,W
    SUBWF    TEMP,W
    BTFSC    STATUS,C
    GOTO     NIV_0

OFF_
    BCF      BUZZER
    BCF      LED_B
    BCF      LED_R
    BCF      LED_G
    RETURN

NIV_5
    BANKSEL  PORTC
    BSF      BUZZER
    BSF      LED_R
    RETURN

NIV_4
    BANKSEL  PORTC
    BSF      LED_R
    BCF      LED_G
    BCF      LED_B
    MOVLW    .10
    MOVWF    SUP
    GOTO     CARGAR_RES

NIV_3
    BANKSEL  PORTC
    BSF      LED_G
    BCF      LED_B
    BCF      LED_R

```

```

        MOVLW    .30
        MOVWF    SUP
        GOTO     CARGAR_RES

NIV_2
        BANKSEL PORTC
        BSF      LED_G
        BCF      LED_B
        BCF      LED_R
        MOVLW    .50
        MOVWF    SUP
        GOTO     CARGAR_RES

NIV_1
        BANKSEL PORTC
        BSF      LED_B
        BCF      LED_R
        BCF      LED_G
        MOVLW    .70
        MOVWF    SUP
        GOTO     CARGAR_RES

NIV_0
        BANKSEL PORTC
        BSF      LED_B
        BCF      LED_R
        BCF      LED_G
        MOVLW    .90
        MOVWF    SUP
        GOTO     CARGAR_RES

CARGAR_RES
        MOVLW    .26
        MOVWF    MED
        MOVLW    .69
        MOVWF    INT
        CLRF     TEMP
        CLRF     INTERVALO
        BTFSC    BUZZER
        GOTO     SWITCH
        BSF      BUZZER
        CALL     RETARDO
        CALL     RETARDO
        RETURN

SWITCH
        MOVLW    0xF8
        ANDWF    PORTC
        BCF      BUZZER
        CLRF     TEMP
        CLRF     INTERVALO
        CALL     RETARDO_BUZ
        RETURN

```

### 2.1.6 Envío de datos a PC mediante comunicación serie

Lo primero es configurar el periférico para la comunicación. Se realiza una comunicación de 8bits, asíncrona y de alta velocidad (registro TXSTA). Luego se carga el registro SPBRG con un valor que permita, en nuestro caso, un Baud Rate de 9600 considerando que la frecuencia del oscilador está configurada a 8MHz. Dicho va-

lor puede obtenerse mediante tablas suministradas por el fabricante o mediante la siguiente ecuación:

$$X = \frac{F_{osc}}{16 * \text{BaudRateDeseado}} - 1 = \frac{8\text{MHz}}{16 * 9600} - 1 = 51$$

;9600 baudios con Fosc 8MHz

MOVLW .51

MOVWF SPBRG

MOVLW B'00100100'

MOVWF TXSTA

;Cambiar de banco

;BRG 8bits

BCF BAUDCTL, BRG16

Realizada la configuración, solo resta llamar a la subrutina ENVIAR, para transmitir los datos desde el microcontrolador hacia la PC. Durante la transmisión se hace una espera activa hasta que el registro TXREG se vacíe, garantizando que la totalidad de los bits fueron transmitidos.

ENVIAR\_DATO\_TX

BANKSEL TXREG

MOVWF TXREG

BSF STATUS, RPO

;Chequeo TSR, espero a que se vacíe (0)

BTFSS TXSTA, TRMT

GOTO \$-1

BCF STATUS, RPO

RETURN

2.2 Grabación del PIC

Para realizar la programación se uso la herramienta *PICKit 2* y se mapearon los pines entre el programador y el PIC16F887 conforme a la siguiente imagen:

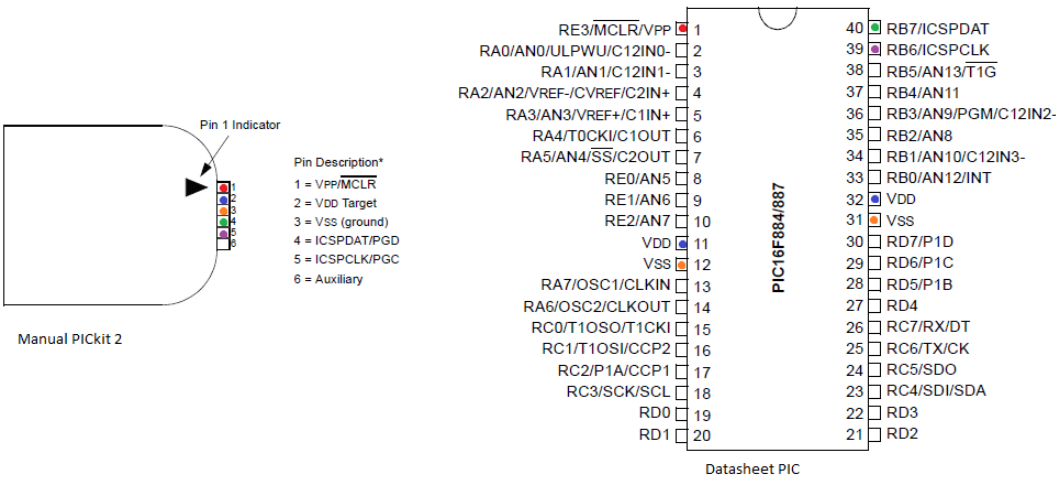


Figura 3: Mapeo de pines.

Para alimentar al PIC hay que usar solo un par de pines  $V_{DD}$  y  $V_{SS}$ .

Una vez conectado el programador y ya en la herramienta del PICKit, elegimos el dispositivo(PIC16F887) y seleccionamos la opción de importación del archivo .hex y grabación.

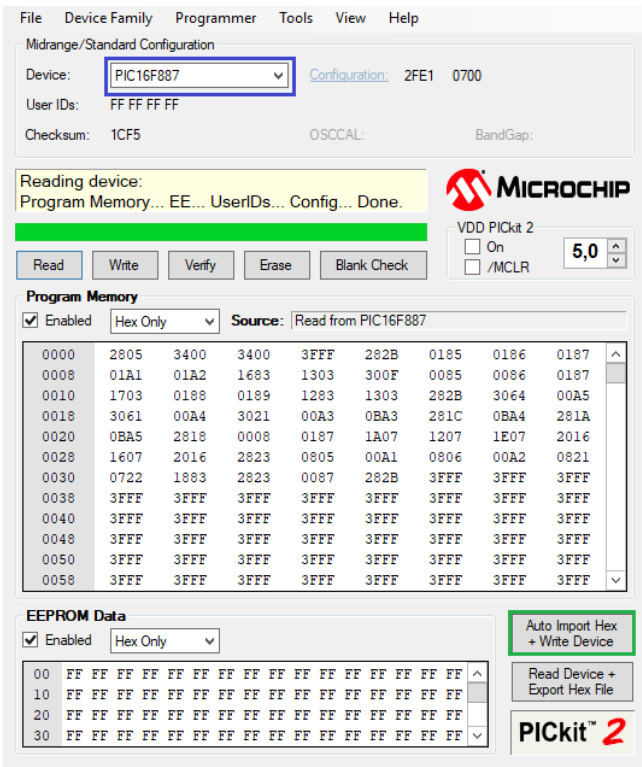


Figura 4: Grabación del microcontrolador.

## 2.3 Implementación

### 2.3.1 Simulación

A los fines de probar si el código desarrollado cumple con los requerimientos, se implemento todo el circuito empleando la herramienta *Proteus*. Con el documento se adjunta el archivo .pdsprj del simulador.

La siguiente figura muestra como se disponen e interconectan los diversos elementos que conforman el circuito.

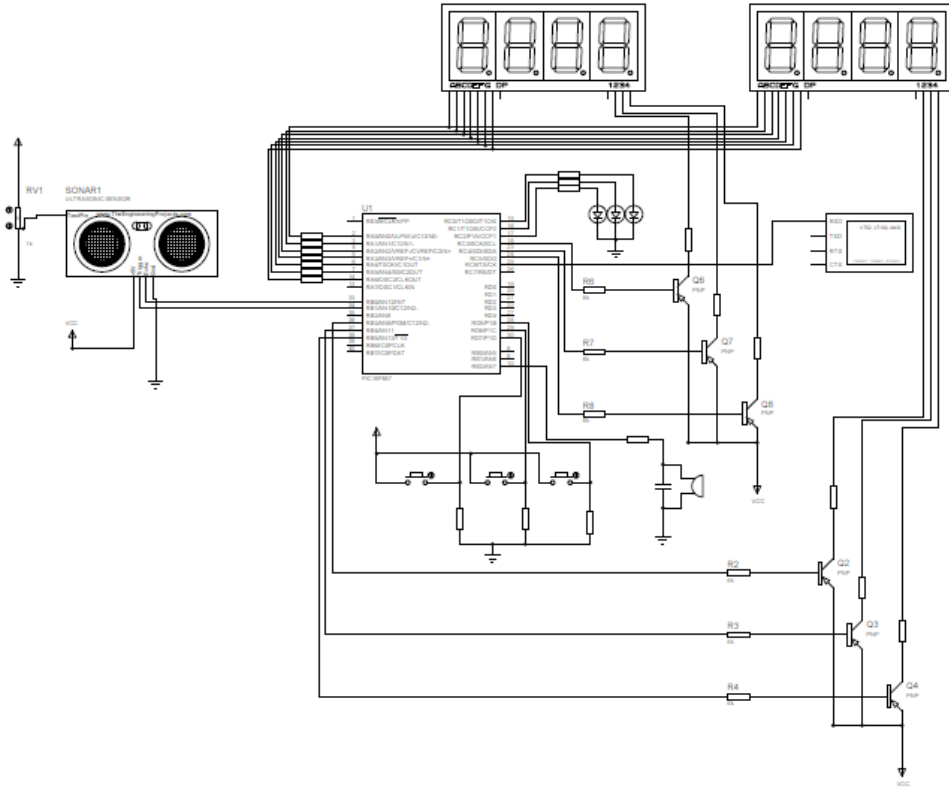


Figura 5: Esquemático del circuito.

*Nota: Las resistencias conectadas en los colectores de los transistores, son unicamente a los fines que la simulación en Proteus funcione. En el montaje físico dichas resistencias no se utilizan.*

En esta etapa lo que resta es el cálculo del valor de la resistencia limitadora de corriente asociada a cada LED ( $R_C$ ) y el valor de la resistencia conectada en la base de cada transistor ( $R_B$ ).

A partir de la Figura 6 y aplicando la ley de Kirchhoff de la tensión, obtenemos las siguientes ecuaciones:

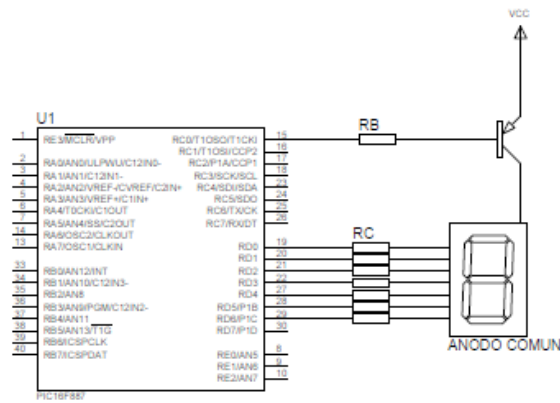
$$V_{CC} = V_{ce} + V_{diodo} + I_c * R_c \quad (2)$$

Luego  $R_c$  es:

$$R_c = \frac{V_{CC} - V_{ce} - V_{diodo}}{I_c} = \frac{5,9V - 0,2V - 2V}{10mA} = 370\Omega$$

Si tomamos como valor de  $h_{FE}$  100, el valor de  $R_b$  lo calculamos a partir de la siguiente ecuación:

$$V_{CC} = V_{be} + I_b * R_b \quad (3)$$





### 2.3.2 Montaje

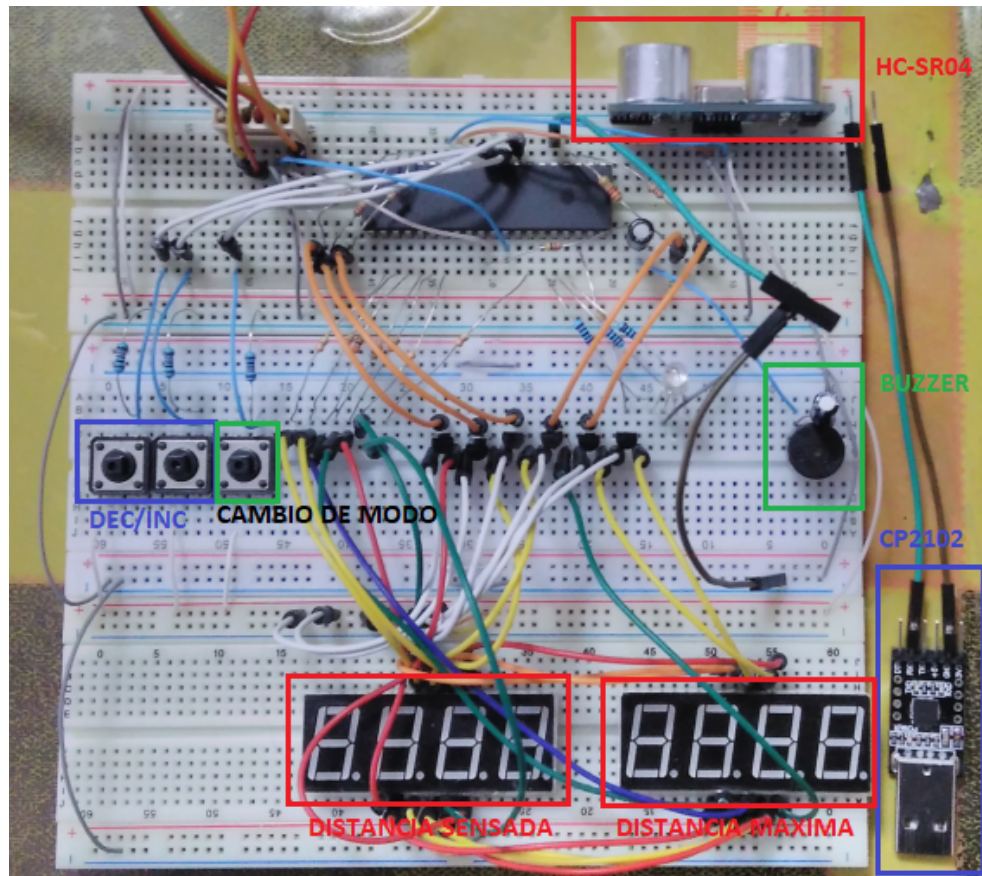


Figura 7: Montaje del circuito.

Junto con los archivos .asm, .hex, .pdsprj y el presente documento, se adjunta un video del proyecto funcionando.