FACULTAD DE INGENIERÍA, UNAM

LABORATORIO DE MICROCOMPUTADORASS

SEMESTRE 2023-2

GRUPO 11

PREVIO PRÁCTICA 6

CONVERTIDOR ANALÓGICO/DIGITAL

NOMBRE DEL ALUMNO:

ARRIAGA MEJÍA JOSÉ CARLOS

PROFESOR

ING. ROMAN V. OSORIO COMPARAN

FECHA DE ENTREGA: **14 DE ABRIL DE 2023** **CALIFICACION**

Objetivo

Familiarizar al alumno con el uso y aplicación del Convertidor Analógico/Digital de un microcontrolador.

Nota: Debido a que en el puerto D se encuentran conectados los led y ambos displays, en el ejercicio 1 y 2 la salida se marcaran en ambas salidas.

1.- Empleando el canal de su elección del convertido A/D, realizar un programa en el cuál, de acuerdo a una entrada analógica que se ingrese por este canal, se represente el resultado de la conversión en un puerto paralelo utilizar el arreglo de leds para ver la salida, como se muestra en la figura 6.1.

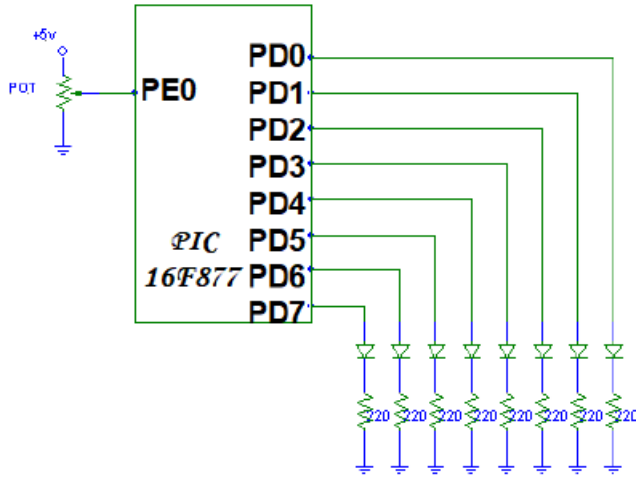


Figura 6.1 Circuito con lectura de una señal analógica

Aplicamos el algoritmo que viene en el manual de prácticas.

El algoritmo a emplear para el uso del convertidor A/D, con resolución de 8 bits:

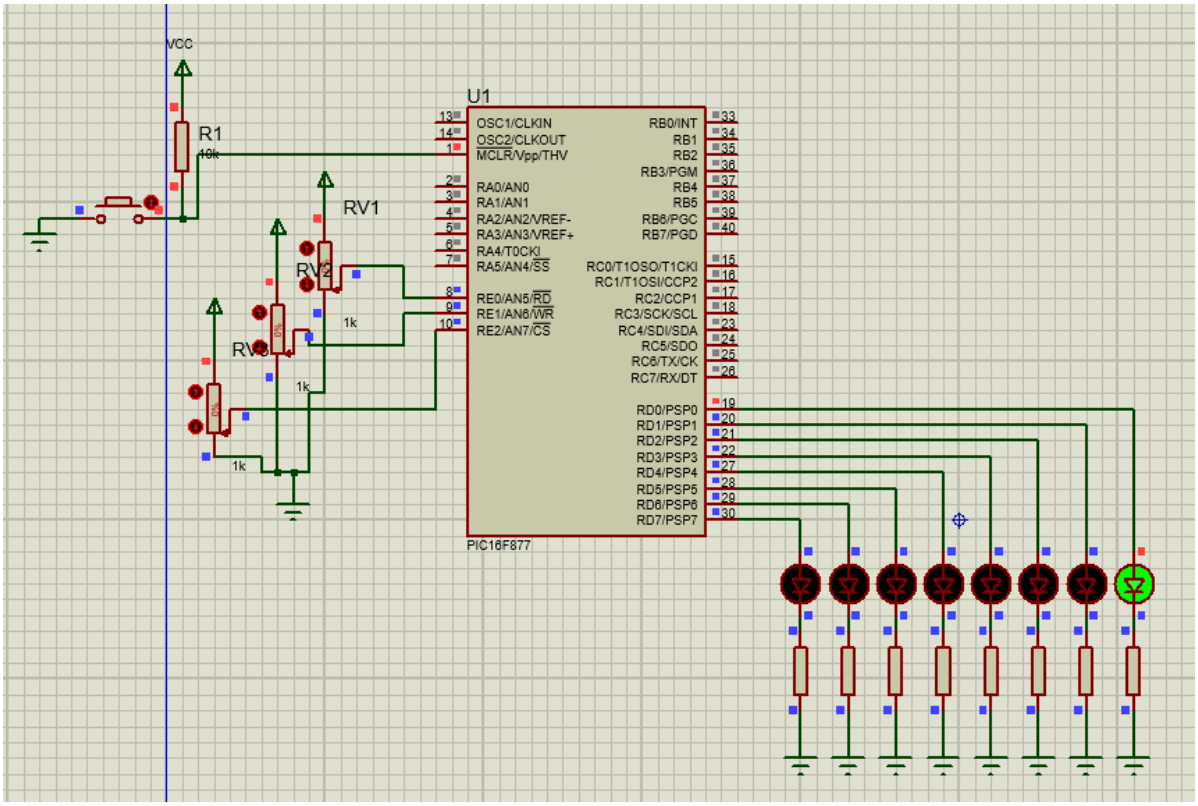
1. Ubicado en el banco cero, limpiar el puerto A, usando CLRF PORTA.
2. Cambiar al banco uno.
3. Configurar el puerto A como entradas analógicas, escribir 00H al registro ADCON1.
4. Regresar al banco 0.
5. Realizar la configuración de la fuente de reloj, el canal de entrada y prender al convertidor A/D, en el registro ADCON0.
6. Iniciar la conversión colocando un ‘1’ a la bandera GO/DONE#.
7. Generar un tiempo de retardo de 20 microsegundos.
8. Esperar a que GO/DONE# sea igual a cero, lo que indica que ha concluido el proceso de conversión.
9. Lee el resultado de la conversión del registro ADRESH.

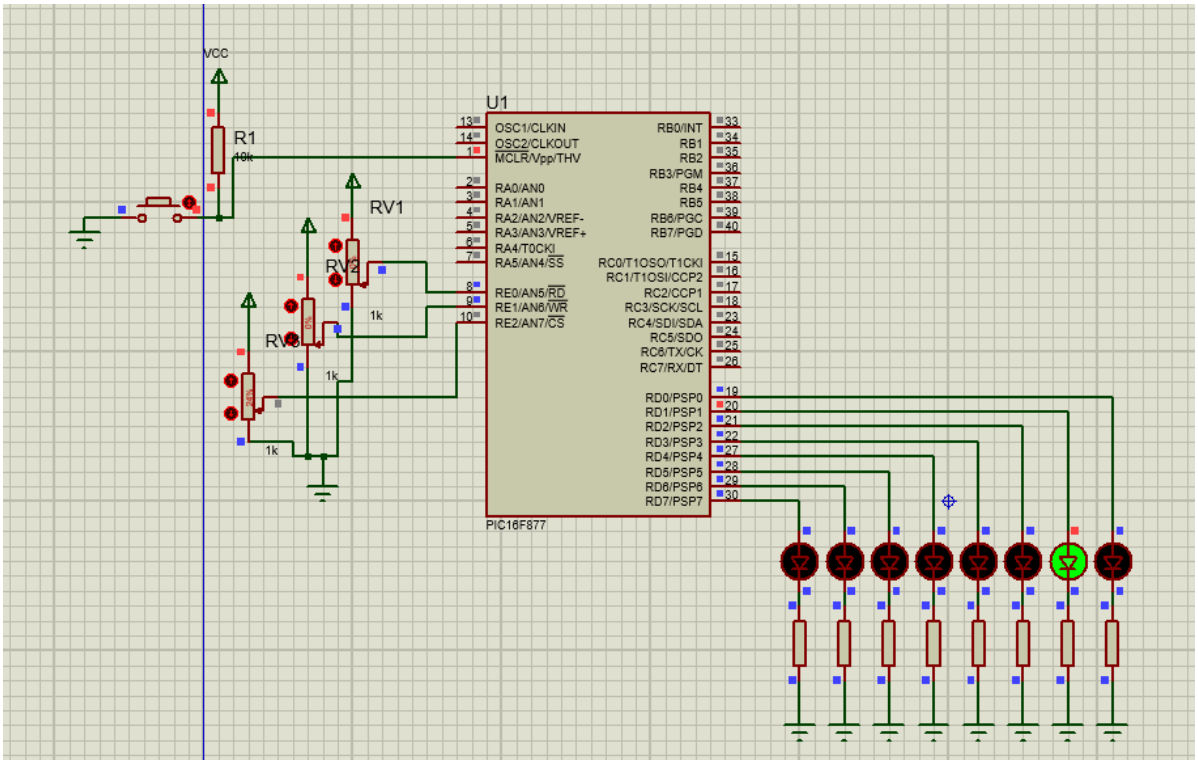
```
processor 16f877A
#include <p16f877a.inc>
    valor1 equ h'21'
    valor2 equ h'22'
    valor3 equ h'23'
    cte1 equ .002
    cte2 equ .200
    cte3 equ .82
    ORG 0
    GOTO INICIO
    ORG 5
INICIO:
    CLRF PORTE ; Limpia PORTE
    BSF STATUS,RP0 ; Cambia a banco 1
    MOVLW 00H ; Define puertos A como analógico
    MOVWF ADCON1
    CLRF TRISD ; Limpia TRISD para que sea salidas
    BCF STATUS,RP0 ; Cambia al banco 0
    MOVLW B'11111001'   ; Cargamos 11xxxxxx freq,xx111xxx canales, xxxxx00x estado conversion go/done,
                        ; xxxxxxx1 prender convertidor
    MOVWF ADCON0
CICLO:
    BSF ADCON0,GO ; inicia conversión A/D
    CALL RETARDO_20US ; espera un retardo
```

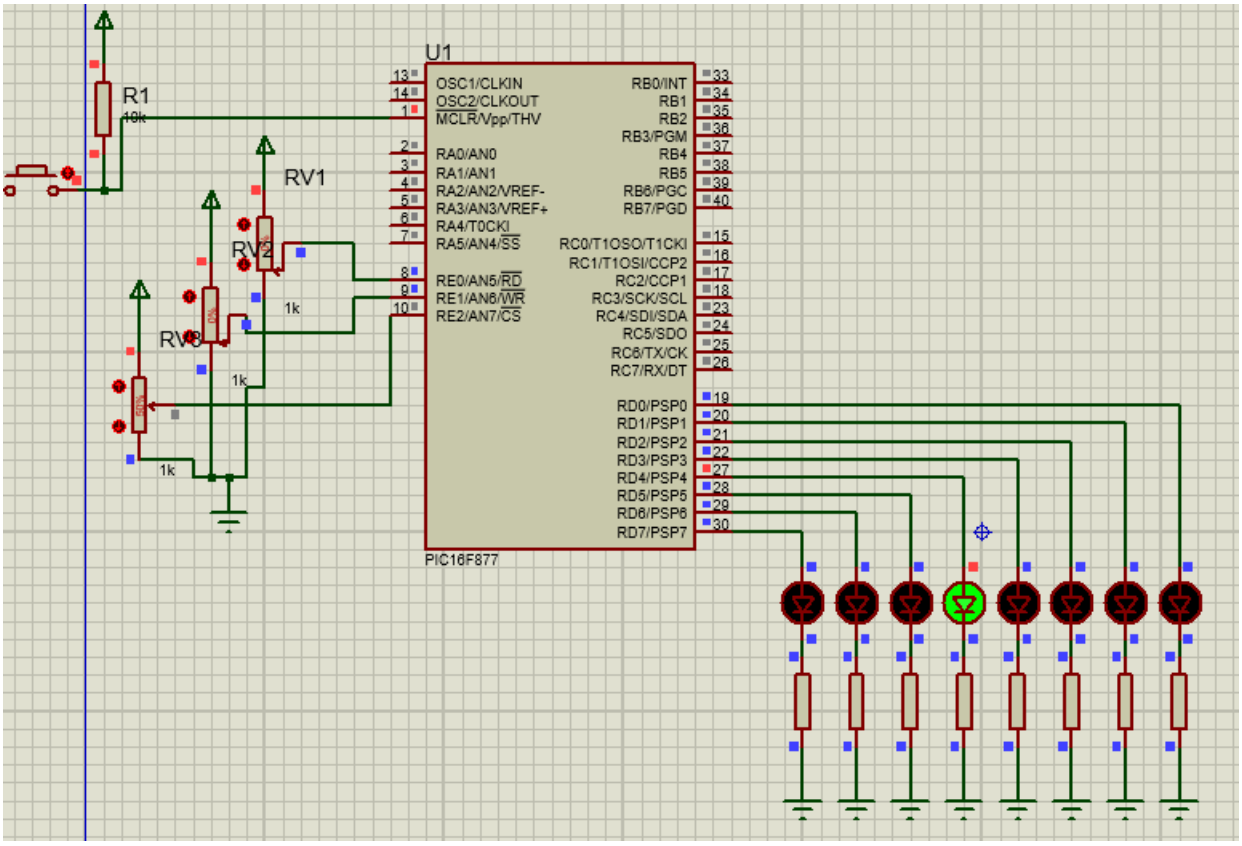
Definimos rangos para que se prenda cada led, en este caso dividí los 255 que caben en 8 bits, entre los mismo 8 para tener un rango lo más equitativo. Se verifica a en que rango esta la entrada y se manda a la salida correspondiente.

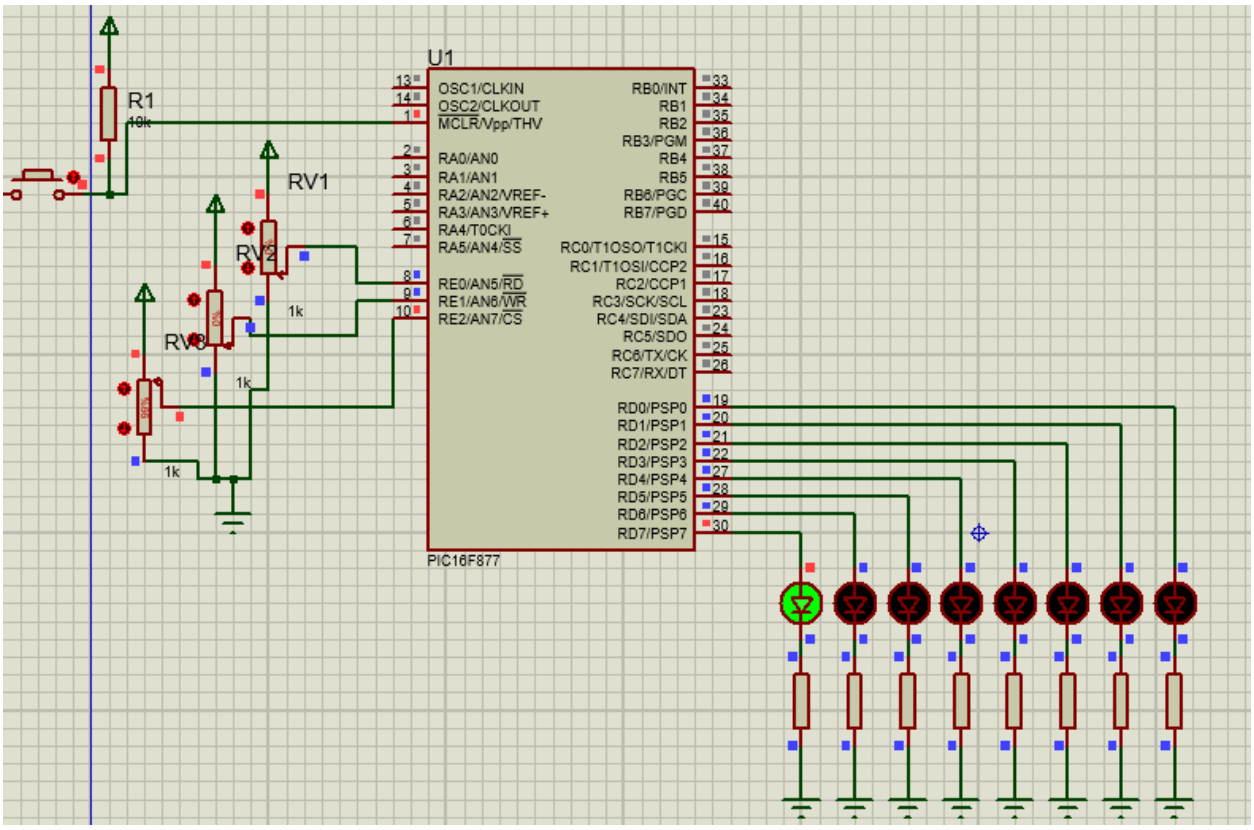
```
ESPERA:
    BTFSC ADCON0,GO ; checa si acabó (vale 0)
    GOTO ESPERA ; se queda en ciclo esperando}
    MOVF ADRESH,W ; lee el resultado
    MOVLW B'00000000' ; 0
    SUBWF ADRESH,W ; W = ADRESH - 0
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO CERO ; es negativo, entonces es cero
    MOVLW B'00100000' ;
    SUBWF ADRESH,W ; W = ADRESH - 32
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO UNO ; es negativo, entonces es uno
    MOVLW B'01000000' ;
    SUBWF ADRESH,W ; W = ADRESH - 64
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO DOS ; es negativo, entonces es dos
    MOVLW B'01100000' ;
    SUBWF ADRESH,W ; W = ADRESH - 96
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO TRES ; es negativo, entonces es tres
    MOVLW B'10000000' ;
    SUBWF ADRESH,W ; W = ADRESH - 128
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO CUATRO ; es negativo, entonces es cuatro
    MOVLW B'10100000' ;
    SUBWF ADRESH,W ; W = ADRESH - 160
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO CINCO
    MOVLW B'11000000' ;
    SUBWF ADRESH,W ; W = ADRESH - 192
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO SEIS
    MOVLW B'11100000' ;
    SUBWF ADRESH,W ; W = ADRESH - 224
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO SIETE
    MOVLW B'11111111' ;
    SUBWF ADRESH,W ; W = ADRESH - 255
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO OCHO
```

A pesar de que hay 3 potenciómetros solamente estamos usando uno, el de hasta la derecha. Podemos ver como al ir subiendo va recorriendo los leds.






U1
R1
10k
RV1
RV2
RV3
1k
1k
1k
OSC1/CLKIN
OSC2/CLKOUT
MCLR/Vpp/THV
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877


U1
R1
10k
RV1
RV2
RV3
1k
1k
1k
OSC1/CLKIN
OSC2/CLKOUT
MCLR/Vpp/THV
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877

Estas son las salidas correspondientes a cada rango.

```
CERO:;num de led encendido
    MOVLW B'00000000' ; W = valor de salida = 0
    GOTO CARGA
UNO:
    MOVLW B'00000001' ; W = valor de salida = 1
    GOTO CARGA
DOS:
    MOVLW B'00000010' ; W = valor de salida = 2
    GOTO CARGA
TRES:
    MOVLW B'00000100' ; W = valor de salida = 3
    GOTO CARGA
CUATRO:
    MOVLW B'00010000' ; W = valor de salida = 4
    GOTO CARGA
CINCO:
    MOVLW B'00010000' ; W = valor de salida = 5
    GOTO CARGA
SEIS:
    MOVLW B'00100000' ; W = valor de salida = 6
    GOTO CARGA
SIETE:
    MOVLW B'01000000' ; W = valor de salida = 7
    GOTO CARGA
OCHO:
    MOVLW B'10000000' ; W = valor de salida = 8
    GOTO CARGA
CARGA:
    MOVWF PORTD ; se carga la salida del programa
    GOTO CICLO
```

Subrutina para generar un retardo de 20us

```
RETARDO_20US ; subrutina de retardo
    MOVLW cte1
    MOVWF valor1
tres
    MOVLW cte2
    MOVWF valor2
dos
    MOVLW cte3
    MOVWF valor3
uno
    DECFSZ valor3
    GOTO uno
    DECFSZ valor2
    GOTO dos
    DECFSZ valor1
    GOTO tres
    RETURN
END
```

2.- Utilizando el circuito anterior, realizar un programa que indique el rango en el cual se encuentra el voltaje a la entrada del convertidor canal seleccionado. Mostrar el valor en un display de 7 segmentos.

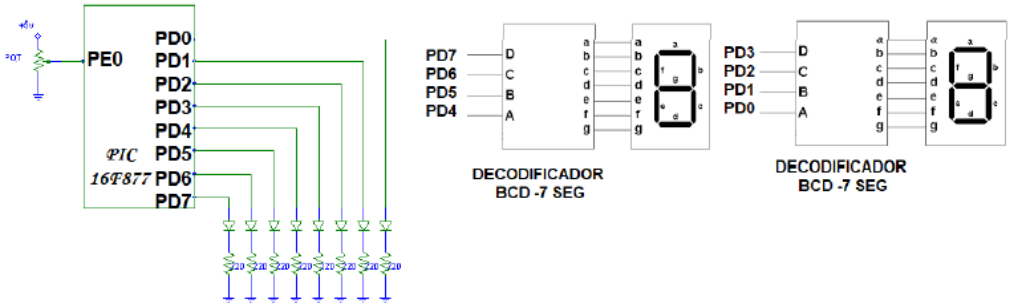


Figura 6.2 Circuito actividad 2

Entrada Analógica Ve	Salida
0 – 0.99 V	0
1.0 – 1.99 V	1
2.0 – 2.99 V	2
3.0 – 3.99 V	3
4.00 – 4.80 V	4
4.80 – 5.00 V	5

Tabla 6.1
Donde Vcc = 5 volts

Utilizamos el mismo algoritmo que en el ejercicio anterior.

```
processor 16f877A
#include <p16f877a.inc>
    valor1 equ h'21'
    valor2 equ h'22'
    valor3 equ h'23'
    cte1 equ .002
    cte2 equ .200
    cte3 equ .82
    ORG 0
    GOTO INICIO
    ORG 5
INICIO:
    CLRF PORTE ; Limpia PORTE
    BSF STATUS,RP0 ; Cambia a banco 1
    MOVLW 00H ; Define puertos A como analógico
    MOVWF ADCON1
    CLRF TRISD ; Limpia TRISD para que sea salidas
    BCF STATUS,RP0 ; Cambia al banco 0
    MOVLW B'11111001'   ; Cargamos 11xxxxxx freq,xx111xxx canales, xxxxx00x estado conversion go/done,
                        ; xxxxxxx1 prender convertidor
    MOVWF ADCON0
CICLO:
    BSF ADCON0,GO ; inicia conversión A/D
    CALL RETARDO_20US ; espera un retardo
```

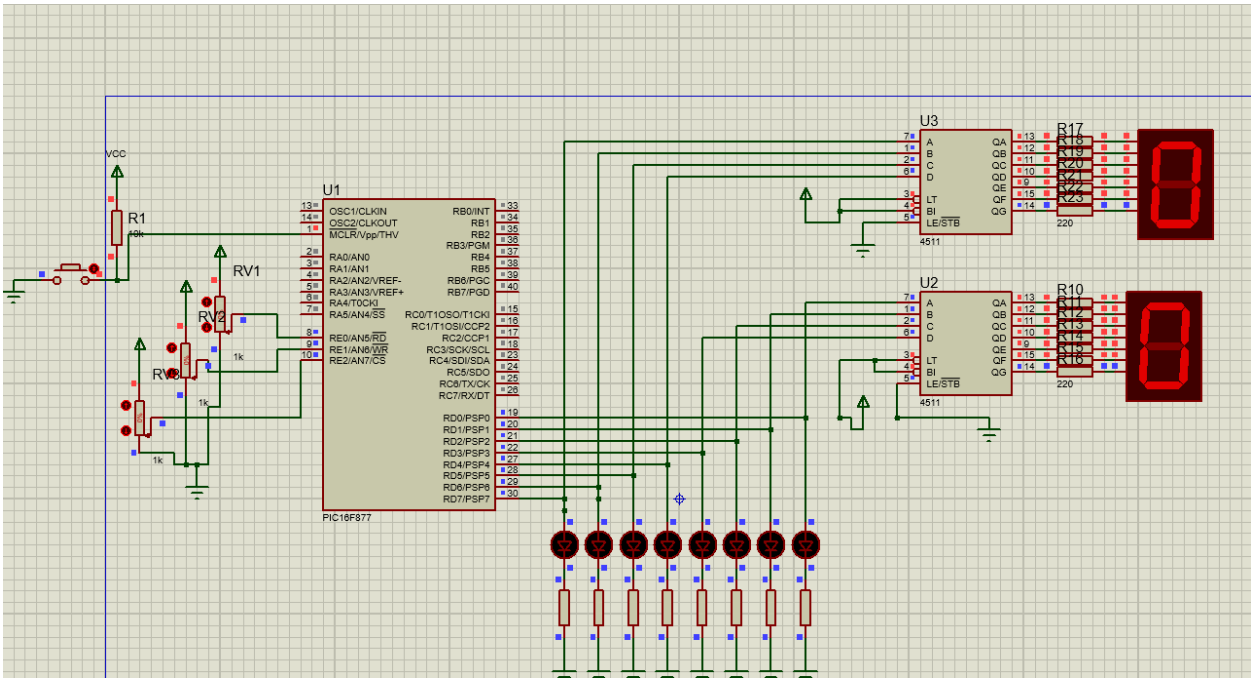
Al igual que en el anterior dividimos entre 5 para sacar los valores de la tabla y para el ultimo renglón se saca el equivalente de los .20V.

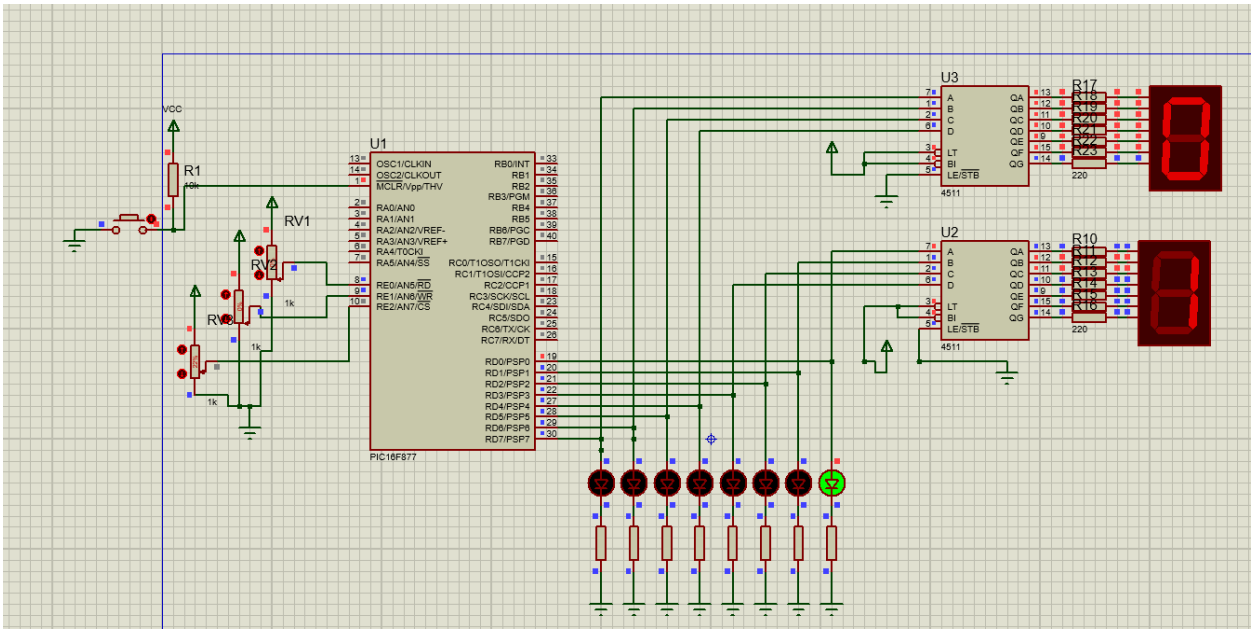
```
ESPERA:
    BTFSC ADCON0,GO ; checa si acabó (vale 0)
    GOTO ESPERA ; se queda en ciclo esperando}
    MOVF ADRESH,W ; lee el resultado
    MOVLW B'00110011' ; 51
    SUBWF ADRESH,W ; W = ADRESH - 51
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO CERO ; es negativo, entonces es cero
    MOVLW B'01100111' ; 103
    SUBWF ADRESH,W ; W = ADRESH - 103
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO UNO ; es negativo, entonces es uno
    MOVLW B'10011011' ; 155
    SUBWF ADRESH,W ; W = ADRESH - 155
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO DOS ; es negativo, entonces es dos
    MOVLW B'11001111' ; 207
    SUBWF ADRESH,W ; W = ADRESH - 207
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO TRES ; es negativo, entonces es tres
    MOVLW B'11111001' ; 249
    SUBWF ADRESH,W ; W = ADRESH - 249
    BTFSS STATUS,C ; si es positivo siguiente comparacion
    GOTO CUATRO ; es negativo, entonces es cuatro
    GOTO CINCO
```

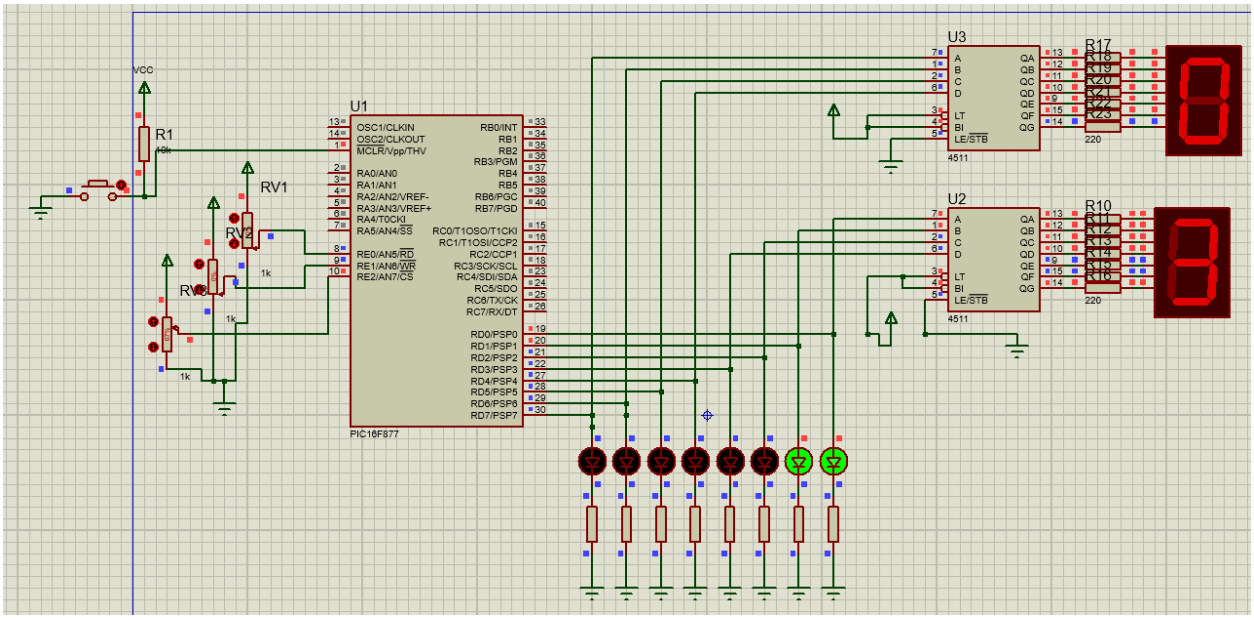
Las salidas y subrutina de retraso

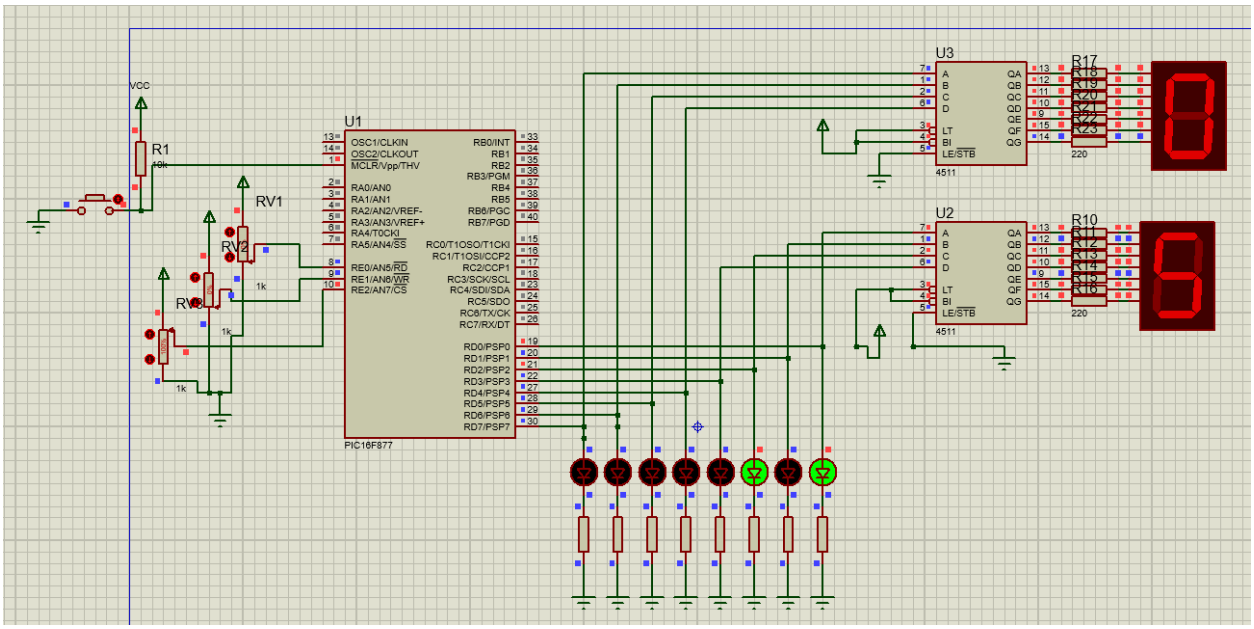
```
CERO:
    MOVLW B'00000000' ; W = valor de salida = 0
    GOTO CARGA
UNO:
    MOVLW B'00000001' ; W = valor de salida = 1
    GOTO CARGA
DOS:
    MOVLW B'00000010' ; W = valor de salida = 2
    GOTO CARGA
TRES:
    MOVLW B'00000011' ; W = valor de salida = 3
    GOTO CARGA
CUATRO:
    MOVLW B'00000100' ; W = valor de salida = 4
    GOTO CARGA
CINCO:
    MOVLW B'00000101' ; W = valor de salida = 5
    GOTO CARGA
CARGA:
    MOVWF PORTD ; se carga la salida del programa
    GOTO CICLO
; 0 - 51 = 0 --> '00110011'
; 52 - 103 = 1 --> '01100111'
; 104 - 155 = 2 --> '10011011'
; 156 - 207 = 3 --> '11001111'
; 208 - 249 = 4 --> '11111001'
; 250 - 255 = 5 --> '11111111'
```

```
; 250 - 255 = 5 --> 11111111
RETARDO_20US ; subrutina de retardo
    MOVLW cte1
    MOVWF valor1
tres
    MOVLW cte2
    MOVWF valor2
dos
    MOVLW cte3
    MOVWF valor3
uno
    DECFSZ valor3
    GOTO uno
    DECFSZ valor2
    GOTO dos
    DECFSZ valor1
    GOTO tres
    RETURN
END
```


VCC
R1
10k
RV1
RV2
RV3
1k
U1
OSC1/CLKIN
OSC2/CLKOUT
MCLR/Vpp/THV
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877
U3
U2
A
B
C
D
LT
BI
LE/STB
QA
QB
QC
QD
QE
QF
QG
4511
R17
R18
R19
R20
R21
R22
R23
R10
R11
R12
R13
R14
R15
R16
220


VCC
R1
10k
RV1
RV2
RV3
1k
U1
OSC1/CLKIN
OSC2/CLKOUT
MCLR/Vpp/THV
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877
U3
U2
A
B
C
D
LT
BI
LE/STB
QA
QB
QC
QD
QE
QF
QG
4511
R17
R18
R19
R20
R21
R22
R23
R10
R11
R12
R13
R14
R15
R16
220





3.- Realizar un programa, de manera que identifique cuál de tres señales analógicas que ingresan al convertidor A/D es mayor que las otras dos; representar el resultado de acuerdo al contenido de la tabla 6.2.

Señal	PD2	PD1	PD0
Ve1>Ve2 y Ve3	0	0	1
Ve2>Ve1 y Ve3	0	1	1
Ve3>Ve1 y Ve2	1	1	1

Tabla 6.2

Si v1 es mayor en el display sale 1, si v2 es mayor sale 3 y si v3 es mayor sale 7

Circuito empleado para este ejercicio.

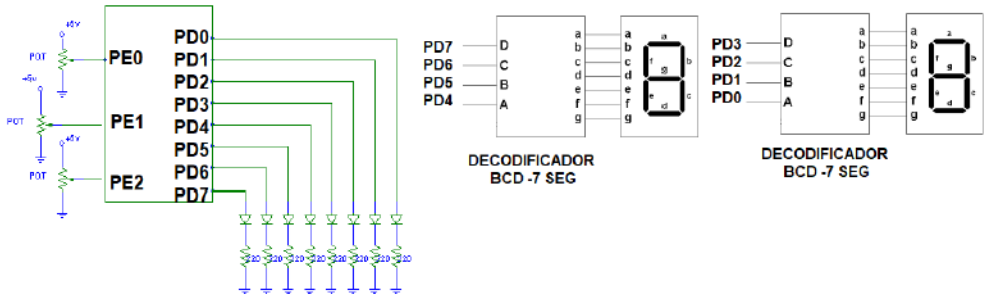


Figura 6.3 Tres señales analógicas

Usamos el mismo algoritmo, solamente que usando los tres potenciómetros para poder guardar 3 variables.

```
processor 16f877A
#include <p16f877a.inc>
    valor1 equ h'21'
    valor2 equ h'22'
    valor3 equ h'23'
    cte1 equ .002
    cte2 equ .200
    cte3 equ .82
    ve1 equ h'24'
    ve2 equ h'25'
    ve3 equ h'26'
    ORG 0
    GOTO INICIO
    ORG 5
INICIO:
    CLRF PORTE ; Limpia PORTE
    BSF STATUS,RP0 ; Cambia a banco 1
    MOVLW 00H ; Define puertos E como analógico
    MOVWF ADCON1
    CLRF TRISD ; Limpia TRISD para que sea salidas
    BCF STATUS,RP0 ; Cambia al banco 0
CICLO:
    MOVLW B'11101001' ; Cargamos freq, canales, go/done, prender conv
    MOVWF ADCON0
    BSF ADCON0,GO ; inicia conversión A/D canal 5
    CALL RETARDO_20US ; espera un retardo
ESPERA1:
    BTFSC ADCON0,GO ; checa si acabó (vale 0)
    GOTO ESPERA1 ; se queda en ciclo esperando
    MOVF ADRESH,W ; lee el resultado
    MOVWF ve1 ; guarda ve1
    MOVLW B'11110001' ; Cargamos 11 freq, 110 canales, 00 algo, 1 prender conv
    MOVWF ADCON0
    BSF ADCON0,GO ; inicia conversión A/D canal 6
    CALL RETARDO_20US ; espera un retardo
ESPERA2:
    BTFSC ADCON0,GO ; checa si acabó (vale 0)
    GOTO ESPERA2 ; se queda en ciclo esperando
    MOVF ADRESH,W ; lee el resultado
    MOVWF ve2 ; guarda ve2
    MOVLW B'11111001' ; Cargamos freq, canales, go/done, prender conv
    MOVWF ADCON0
    BSF ADCON0,GO ; inicia conversión A/D canal 7
    CALL RETARDO_20US ; espera un retardo
```

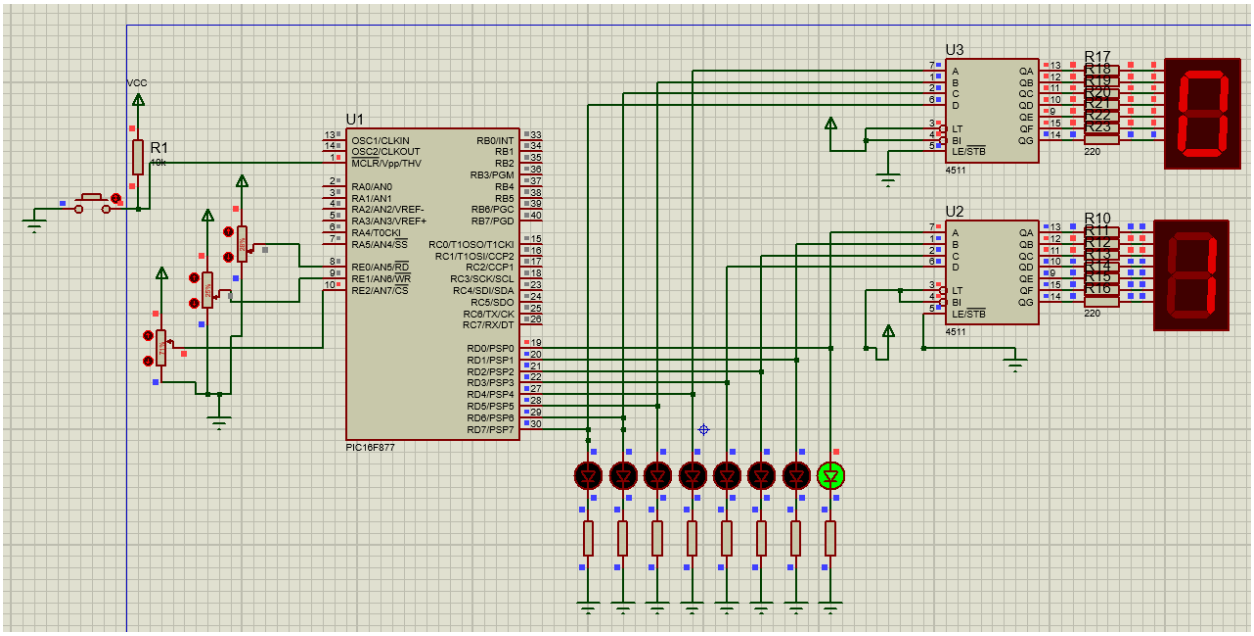
Comienza las comparaciones para ver cual valor es más grande y se manda a la salida correspondiente.

```
ESPERA3:
    BTFSC ADCON0,GO ; checa si acabó (vale 0)
    GOTO ESPERA3 ; se queda en ciclo esperando
    MOVF ADRESH,W ; lee el resultado
    MOVWF ve3 ; guarda ve3
    MOVF ve3,W ; W = ve3
    BCF STATUS,C
    SUBWF ve2,W ; W = ve2 - ve3
    BTFSS STATUS,C ; si ve2 es mayor C prende
    GOTO VE3MAYOR ; si ve3 es mayor entra aquí (otra comparación)
    MOVF ve2,W ; W = ve2
    BCF STATUS,C
    SUBWF ve1,W ; W = ve1 - ve2
    BTFSS STATUS,C ; si ve1 es mayor C prende
    GOTO RES2 ; si ve2 es mayor va a RES2
    GOTO RES1 ; si ve1 es mayor va a RES1
VE3MAYOR:
    MOVF ve3,W ; W = ve3
    BCF STATUS,C
    SUBWF ve1,W ; W = ve1 - ve3
    BTFSS STATUS,C ; si ve1 es mayor C prende
    GOTO RES3 ; si ve3 es mayor va a RES3
    GOTO RES1 ; si ve1 es mayor va a RES1
RES1:
    MOVLW B'00000111' ; W = 1 mayor = salida 111
    GOTO CARGA
RES2:
    MOVLW B'00000011' ; W = 2 mayor = salida 011
    GOTO CARGA
RES3:
    MOVLW B'00000001' ; W = 3 mayor = salida 001
    GOTO CARGA
CARGA:
    MOVWF PORTD ; carga la salida
    GOTO CICLO
```

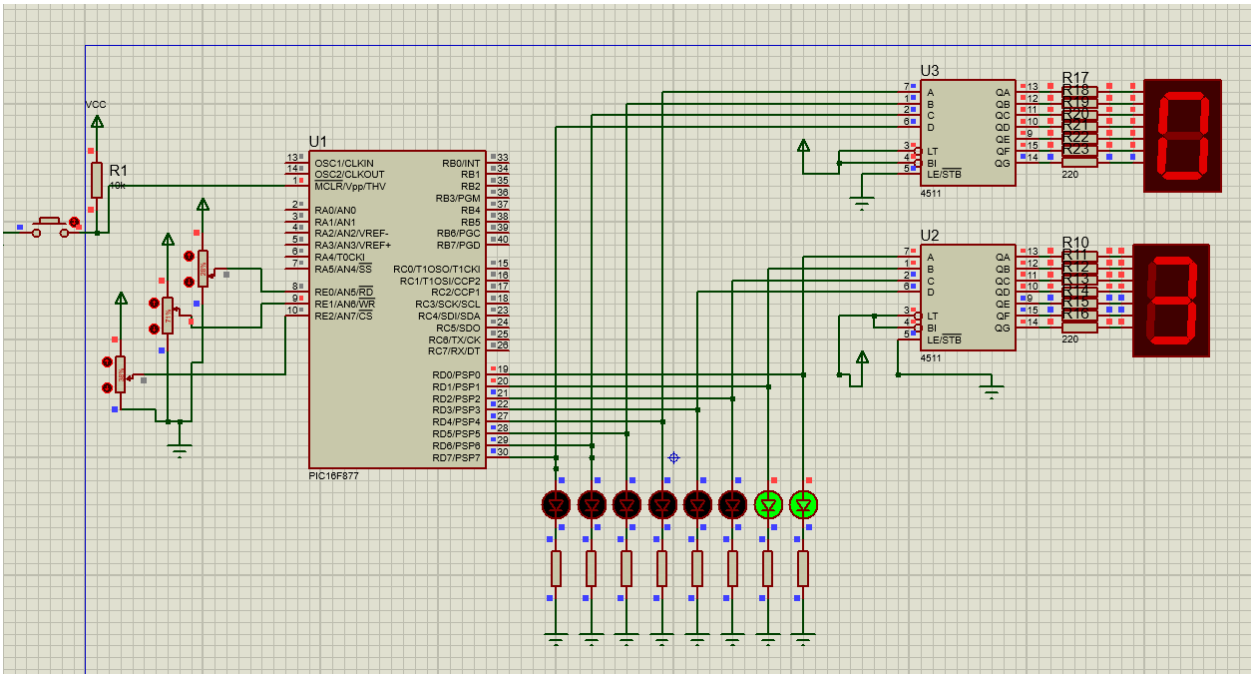
Subrutina de retardo

```
RETARDO_20US ; subrutina de retardo
    MOVLW cte1
    MOVWF valor1
tres
    MOVLW cte2
    MOVWF valor2
dos
    MOVLW cte3
    MOVWF valor3
uno
    DECFSZ valor3
    GOTO uno
    DECFSZ valor2
    GOTO dos
    DECFSZ valor1
    GOTO tres
    RETURN
END
```

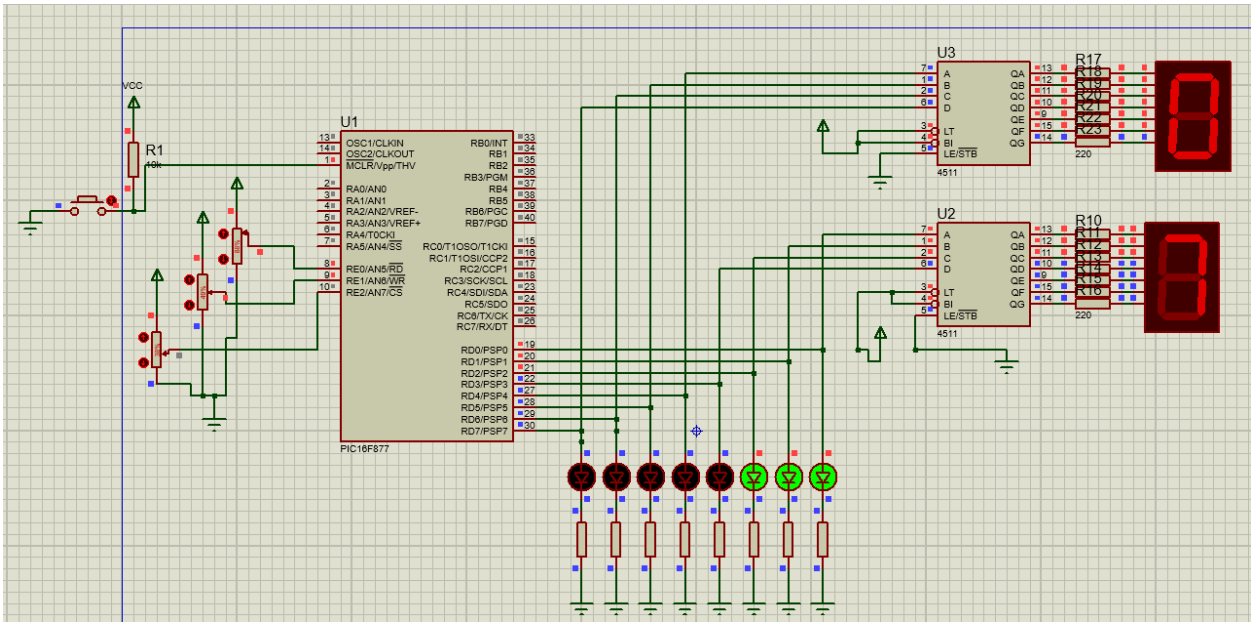
V1 es mayor



V2 es mayor



V3 es mayor


VCC
R1
10k
U1
OSC1/CLKIN
OSC2/CLKOUT
MCLR/Vpp/THV
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877
U3
A
B
C
D
LT
BI
LE/STB
QA
QB
QC
QD
QE
QF
QG
4511
R17
R18
R19
R20
R21
R22
R23
220
U2
4511
R10
R11
R12
R13
R14
R15
R16
220