

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Proyecto 3: Convertidor analógico digital

Semestre: 2020-2

Materia: Microcomputadoras

Grupo: 1

Profesor: Rubén Anaya García

Alumnos:

Garciarebollo Rojas Diego Iñaki

Vega López Amós Manuel

Descripción del proyecto:

El proyecto tiene el objetivo de utilizar el convertidor analógico digital disponible en el PIC16F877A para poder transformar una señal analógica modificable en tiempo real gracias a un potenciómetro lineal, de 0 [v] hasta 5[v], y obtener su equivalente digital, gracias al uso del registro ADRESH, en 4 diferentes formatos:

-Decimal

-Hexadecimal

-Binario

-Voltaje real

Cabe aclarar que los primeros tres dependen de la conversión directa del registro para su formato de salida, mientras que el ultimo muestra el valor real del voltaje en nuestro sistema decimal base 10.

Para esto creamos la configuración para que por el PORTA se recibiera la señal analógica, mientras que por el PORTB se colocara el dipswitch con el cual se modificaría el formato de salida, así mismo el LCD estaría conectado a través de PORTC y PORTD, debido a su uso de las líneas necesarias para recibir comandos y lo necesario para que funcione.

```
inicio                                ; Etiqueta de inicio de programa
    CLRF PORTA                        ; Limpiamos PORTA
    CLRF PORTB                        ; Limpiamos PORTB
    CLRF PORTC                        ; Limpiamos PORTC
    CLRF PORTD                        ; Limpiamos PORTD
    BSF STATUS,RP0                   ; Ponemos un 1 en el bit 5 de STATUS (RP0)
    BCF STATUS,RP1                   ; Ponemos un 0 en el bit 6 de STATUS (RP1) para cambiar de
banco 0 al 1
    MOVLW H'40'                      ; W <- h'40'
    MOVWF ADCON1                     ; ADCON1 <- (W)
    MOVLW H'3F'                      ; W <- h'3F'
    MOVWF TRISA                      ; TRISA <- (W) configuramos PORTA como entrada
    MOVLW H'FF'                      ; W <- h'FF' B'11111111' We need inputs for the dipswitch
    MOVWF TRISB                      ; TRISB <- (W) configuramos PORTB como entrada
    MOVLW H'F8'                      ; W <- h'F8' B'11111000' WE NEEED THRRE OUTPUTS
    MOVWF TRISC                      ; TRISC <- (W) configuramos PORTC como salida
    MOVLW H'00'                      ; W <- h'00'
    MOVWF TRISD                      ; TRISD <- (W) configuramos PORTD como salida
    BCF STATUS,RP0                   ; regresar al banco 0 poniendo el bit 5 de STATUS (RP0) en 0
    MOVLW 0x81
    MOVWF ADCON0

    CALL Inicia_LCD                  ; Se llama a la subrutina que inicializa el LCD
Comportamiento:
    CLRF numero
    CLRF RegAux3
    CLRF RegAux4
    MOVLW 0x01
    call LCD_Comando
    MOVLW 0x02
    call LCD_Comando
```

```

BSF ADCON0,2          ; EMPEZAMOS LA CONVERSION A/D
CALL Retardo_20_micro
BTFSC ADCON0,2        ; CHECAMOS SI SE TERMINO LA CONVERSION A/D
GOTO Comportamiento   ; SE REGRESA EN CASO DE QUE NO SE HAYA ACABADO
MOVF ADRESH,W         ; PASAMOS ADRESH A W PARA TENER EL VALOR DE LA CONVERSION
MOVWF RegAux2         ; RegAux2=AdresH

```

Con esto configurado, así como los valores necesarios para el funcionamiento del convertidor analógico digital, procedimos a aplicar un direccionamiento indexado aplicando una máscara, esto encargándose de acceder a cada parte del código correspondiente a cada una de las 4 posibilidades antes mencionadas.

```

loop:
MOVWF PORTB,W         ; W <- (PORTA) leer entrada en PORTA
ANDLW H'03'          ; Se realiza un AND logico con H'03' (Mascara de bits)
ADDWF PCL,F           ; Se agrega al PC el valor resultante de aplicar la mascara
goto Decimal          ; Mostrar el voltaje en Decimal          PORTB: 00
goto Hexadecimal      ; Mostrar el voltaje en Hexadecimal      PORTB: 01
goto Binario          ; Mostrar el voltaje en binario          PORTB: 10
goto Voltaje          ; Mostrar el voltaje real                PORTB: 11

```

Una vez realizado esto, se imprimiría dependiendo la opción, el resultado dado.

Para la opción 1 a 3 depende de la conversión directa obtenida del registro ADRESH, solamente transformándola al formato deseado, sin alterar el valor base, por lo que los resultados en decimal van de 0 a 255, en hexadecimal de 00 a FF, y en binario de 00000000 a 11111111.

Para decimal ocupamos una comprobación para determinar si el resultado tiene valores mayores a 100, 10 y 1 mediante una resta, y a partir de ello y con uso del comando swap para ordenar el resultado debido a que se usa dos registros, podemos obtener un resultado de 0 a 255.

Para hexadecimal es un poco mas complejo, ya que debemos realizar una comprobación para ver si el resultado dado alcanza a estar en parte alta (0001 000) o parte baja (0000 0001), y a partir de ello realizar una comprobación mas para ver si cada parte es mayor a 9, en caso de serlo imprimimos la letra correspondiente de A a F, y en caso de no serlo imprimimos simplemente un dígito.

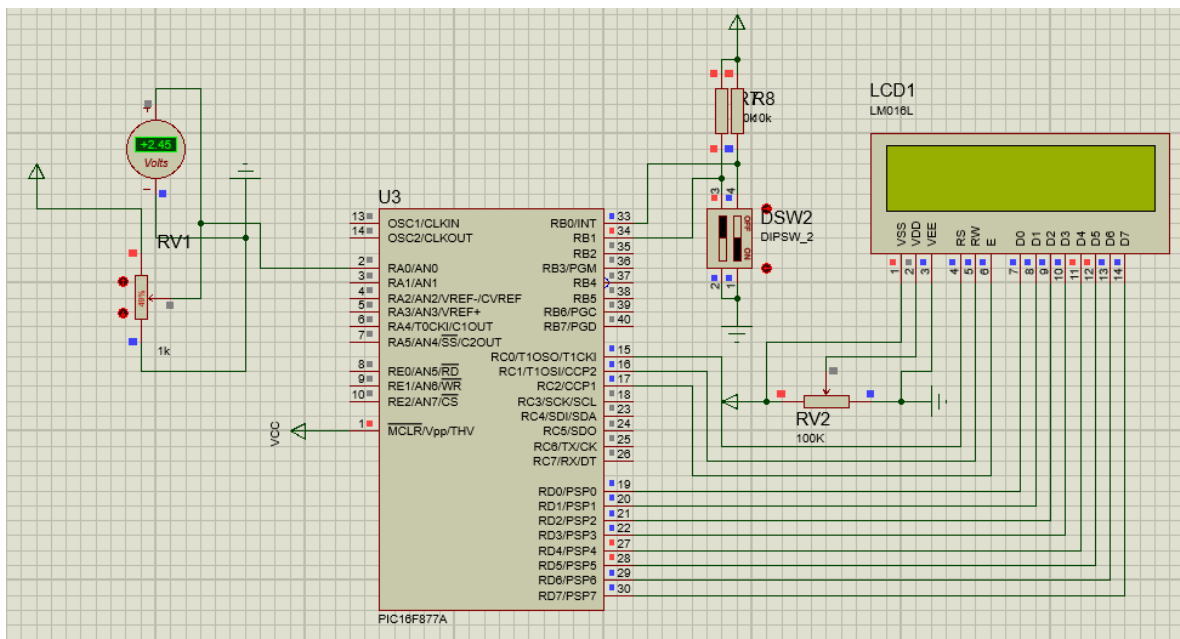
Para el binario es mas simple el proceso y hacemos una comprobación del dígito correspondiente a través del uso de una resta y de un registro auxiliar, con el cual comprobamos si el dígito es un 0 o un 1, y a partir de ello imprimir y rotar hacia la derecha la variable que contiene el resultado de la conversión.

Cabe aclarar que debido a pérdidas propias de la naturaleza del PIC y de la forma en que se realiza la conversión, en la opción 4, el resultado no es perfectamente exacto en algunos casos puntuales, pero eso como ya se mencionó, es debido a la propia naturaleza de la conversión.

Componentes utilizados

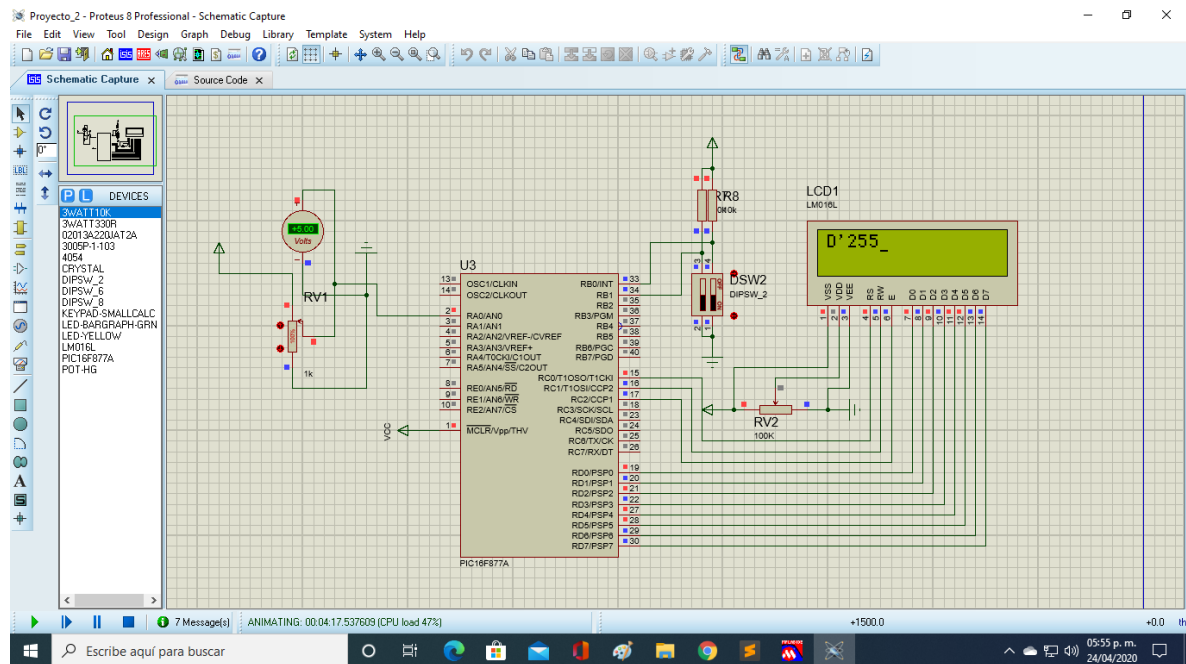
- PIC16F877A
- Potenciómetro lineal
- Multímetro
- Dipswitch de dos posiciones
- Potenciómetro
- LCD 16X2
- 2 resistencias de 10k Ohms
- 2 fuentes de voltaje
- Protoboard

Circuito

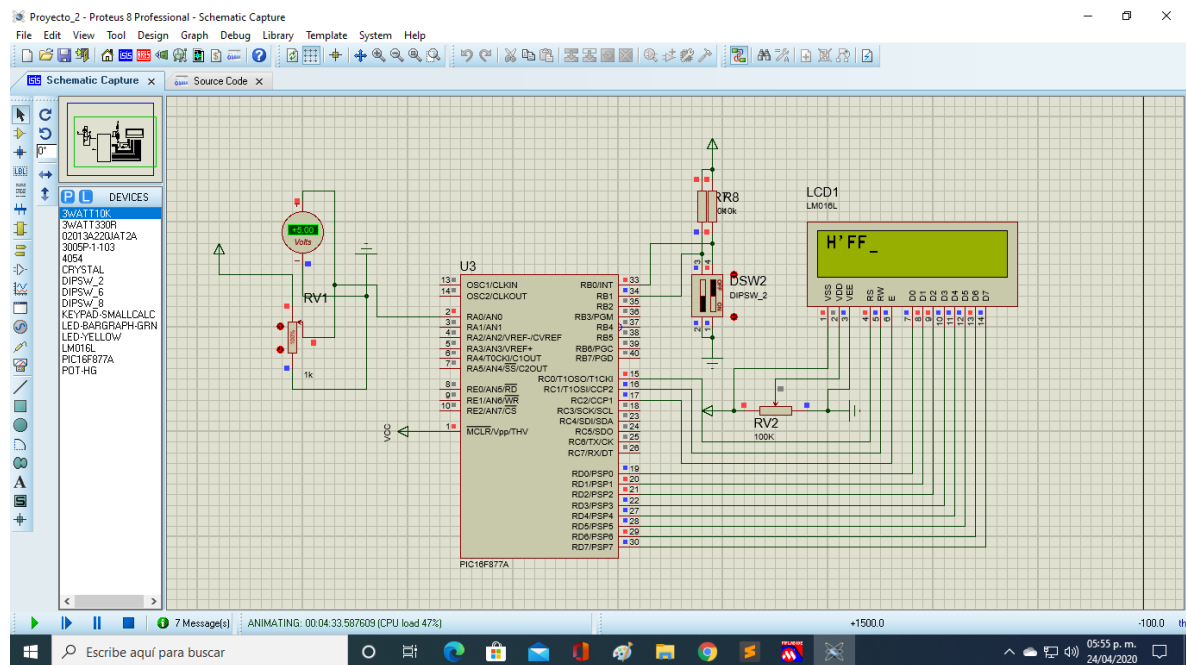


Capturas

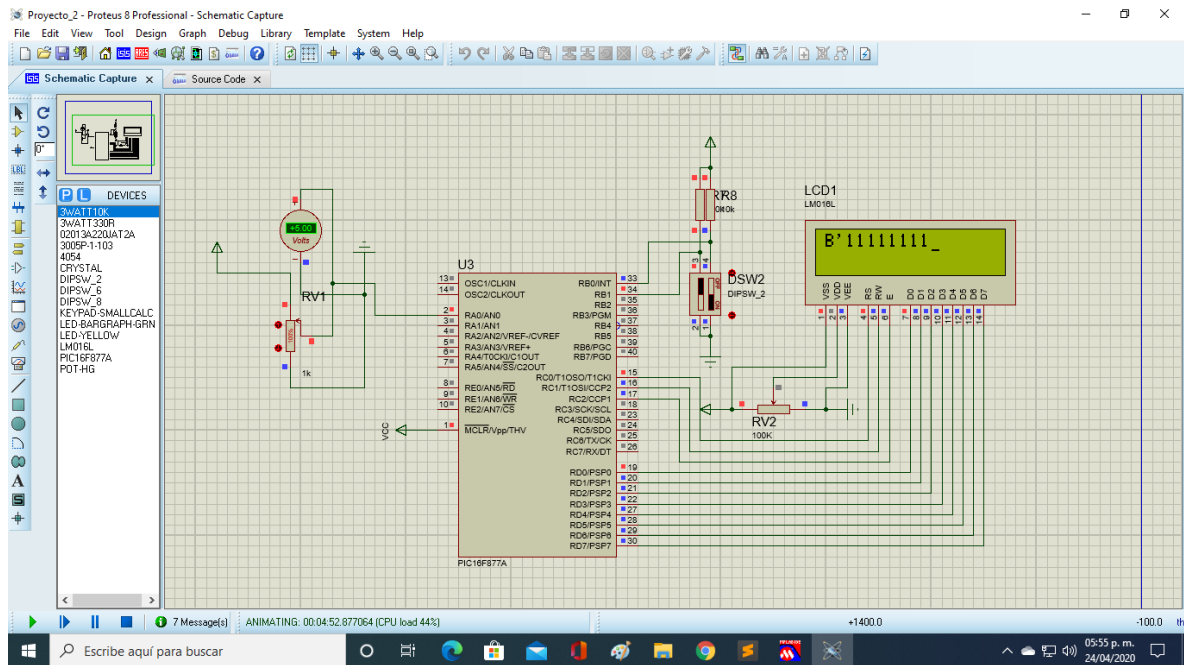
Opción 1: Decimal



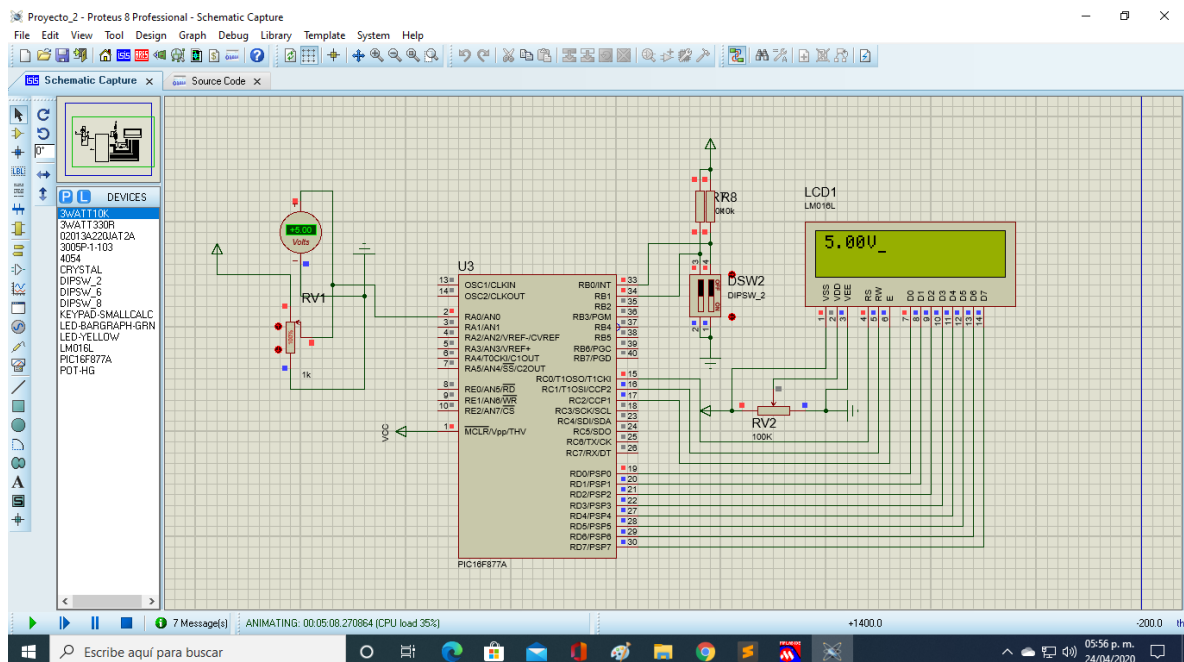
Opción 2: Hexadecimal



Opción 3: Binario



Opción 4: Voltaje real en decimal



Conclusión

Con este proyecto pudimos reforzar los conocimientos adquiridos sobre el tema de convertidor analógico digital, el cual nos permite tener acceso a mas herramientas de las cuales podremos disponer según lo necesitemos. En este caso, por ejemplo, creamos una especie de multímetro, que dependiendo del voltaje de entrada y de la selección de salida, era la forma en que obteníamos el equivalente digital de la señal analógica. Este es solo un ejemplo de todo lo que podemos crear gracias a esta función del PIC, lo que nos abre un abanico de posibilidades acerca de sus aplicaciones dentro de proyectos futuros.