

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Microcomputadoras

Proyecto 02: Convertidor A/D, Volmetro Digital

Profesor: Rubén Anaya García

Alumnos:

- Murrieta Villegas Alfonso
- Reza Chavarría, Sergio Gabriel
- Valdespino Mendieta Joaquín

Grupo: 01

Semestre: 2021-2

Práctica 02: Volmetro Digital

Objetivo

Utilizando como entrada un potenciómetro, cuya entrada variará de 0 a 5 Volts, un selector con 2 switches y como salida el LCD para desplegar la opción deseada, de acuerdo con la siguiente tabla

Selección	Despliegue de conversión A/D en formato	Ejemplo para entrada de 5 V
0	Decimal	255
1	Hexadecimal	0XFF
2	Binario	1111 1111
3	Voltaje	5.00V (Usar 3 dígitos)

Desarrollo

Para el desarrollo del proyecto se designaron el uso de los 5 puertos disponibles del PIC16F877A. El puerto A es utilizado para usar el convertidor en el canal 0, para esto solo fue necesario la configuración de A0 fuera utilizado para el convertidor, los demás puertos serán utilizados como digitales. El puerto E se utilizó para la selección de opciones de formato.

Los puertos de salida se utilizarán de la siguiente manera. El puerto B es utilizado para el manejo de la configuración envió de datos y control junto con el habilitador del LCD. El puerto C será utilizado para el envío de datos y control al display. Y el puerto D es para obtener la información de salida del conversor, esto para corroboración.

Projecto 02
Alumnos:
Muriela Villegas Alfonso
Reza Chavarria Sergio Gabriel
Valdespino Mendez Joaquin

CONVERTIDOR A/D (CANAL 0, PUERTO A)

U1

COMPONENTES DEL LCD (PUERTO B)

LCD1
LMH48

DATOS Y CONTROL DEL LCD (PUERTO C)

SELECCION (PUERTO E)

PIC16F877A

LED7
LED6
LED5
LED4
LED3
LED2
LED1
LED0

R4
R1
R5
R6
R7
R8
R9
R10

DSW1
DIPSW_2

Conversión a hexadecimal

Para la visualización en hexadecimal igual se obtendrá una copia para su manejo. Para el proceso se realizó la obtención de la parte hexadecimal mayor a partir del recorrimiento a la derecha y de la limpieza del bit de acarreo. Una vez realizado el proceso 4 veces se obtiene la parte superior en los 4 primeros bits y con esto se guarda en un registro de la memoria. Este proceso se realiza para la obtención de la parte hexadecimal menor, con un recorrimiento a la izquierda con la limpieza del bit de acarreo y un intercambio de bits al finalizar. Igual el resultado se guardará por separado.

Ambos registros serán mandados a imprimir y para diferencias se imprimirá el carácter 'H'.

Conversión a Binario

El proceso obtendrá una copia de la información del puerto C. Una vez obtenida la información se realizará un recorrimiento a la izquierda y dependiendo del estado del bit de acarreo. Dependiendo del estado del bit se imprimirá un 0 o 1 y se revisará si ya se revisaron los 8 bits correspondientes. Al finalizar se imprimirá el carácter 'B'.

Conversión a Voltaje

Se obtendrá a partir de ciclos el valor de unidad del Voltaje, esto a partir de la disminución repetida del valor de 0X33. Esto se repetirá hasta que el valor sea menor a 0. Para la obtención de décimas y centésimas se realizará un ciclo similar, antes de esto el valor obtenido del primer ciclo se multiplicará por 9. Después de esto se dividirá a través del valor original. Con el procedimiento se obtendrán los decimales.

Al finalizar la obtención de la información, se imprimirá el valor de la siguiente manera "unidad"+"."+"+"decimal"+"centesimal".

Código

```
;=====
;Proyecto 02: Voltmetro
;Alumnos
;  Murrieta Villegas Alfonso
;  Reza Chavarria Sergio Gabriel
;  Valdespino Mendieta Joaquín
;=====

        PROCESSOR    16f877
        INCLUDE      <p16f877.inc>

valor    EQU    H'20'
valor1   EQU    H'21'
valor2   EQU    H'22'

FIN      EQU    H'23'
INFO     EQU    H'24'
ESPACIOS EQU    0X20

NUM_HEX  EQU          H'30'
COPIA_HEX EQU        H'31'

CENTENAS EQU          H'32' ;Registro para centenas
DECENAS  EQU          H'33' ;Registro de decenas
UNIDADES EQU          H'34' ;Registro de unidades

TOP_H    EQU          H'36'
SUB_H    EQU          H'37'

VOLT_NUM EQU          H'38'
VOLT_UNIDAD EQU        H'40'
VOLT_DECIMA EQU        H'41'
VOLT_CENTENA EQU        H'42'

AUX      EQU    H'50'
CONT     EQU    H'51'

        ORG 0
        GOTO INICIO
        ORG 5

INICIO

;=====
;Configuración Convertidor A/DCanal 0

        CLRF    PORTA
        BSF     STATUS,RP0 ;Cambio al Banco 1
        BCF     STATUS,RP1

        MOVLW   0X0E          ;Solo el canal 0 será analógico
                                ;los demas puertos son digitales
        MOVWF   ADCON1

        MOVLW   0X3f          ;Configura el puerto A como entrada (canal 0)
        MOVWF   TRISA

        BCF     STATUS,RP0 ;Regresa al Banco 0
```

```

    MOVLW B'11000001' ;Configuración del registro analógico
                                ;Se configura el canal 0->
                                ;Frecuencia del reloj:11
                                ;CHS2-0:000
                                ;GO/DONE:0 Termina la conversión
                                ; -:0
                                ;adon:1 enciende el convertidor

    MOVWF ADCON0                ;Asigna la conf. al adcon0

;=====
;Puertos

    BSF          STATUS,RP0      ;Cambio al banco 1
    BCF          STATUS,RP1

    MOVLW 0X00
    MOVWF TRISB                  ;PUERTO B COMO SALIDA
    MOVWF TRISC                  ;PUERTO C COMO SALIDA
    MOVWF TRISD                  ;PUERTO D COMO SALIDA

    MOVLW 0X07
    MOVWF TRISE                  ;PUERTO E COMO ENTRADA

    BCF          STATUS,RP0      ;CAMBIO BANCO 0
                                ;LIMPIEZA DE PUERTOS

    CLRF PORTB
    CLRF PORTC
    CLRF PORTD
    CLRF PORTE

    CALL INICIA_LCD

;=====
;CONVERSION
;=====
LECTURA:
    BSF ADCON0,2 ;Enciende el proceso de conversión
    CALL RETARDO

ESPERA:
    BTFSC ADCON0,2;Si está prendido el convertidor
    GOTO ESPERA
    MOVF ADRESH,W ;Registro de los resultados en la parte alta
    MOVWF PORTD   ;Lanza adresh al puerto D
    GOTO SELECT

RETARDO
    MOVLW 0X20      ;Rutina que genera un DELAY
    MOVWF valor1

RE_ONE
    DECFSZ valor1
    GOTO RE_ONE
    RETURN

;=====
;SELECCIÓN POR TECLADO
;=====
SELECT
    MOVLW H'00'

```

```
XORWF PORTE,W
BTFSC STATUS,Z
GOTO DECIMAL
```

```
MOVLW H'01'
XORWF PORTE,W
BTFSC STATUS,Z
GOTO HEXA
```

```
MOVLW H'02'
XORWF PORTE,W
BTFSC STATUS,Z
GOTO BINARIO
```

```
MOVLW H'03'
XORWF PORTE,W
BTFSC STATUS,Z
GOTO VOLTAJE
```

```
GOTO LECTURA
```

```
;=====
;ESTADOS DE SALIDA
;=====
```

```
;*****
;CONVERSIÓN DECIMAL
;*****
DECIMAL
```

```
MOVLW 0X80
CALL COMANDO
```

```
CLRF AUX
MOVF ADRESH,W
MOVWF NUM_HEX
MOVWF COPIA_HEX
```

```
CALL OBT_DECIMAL
CALL IMPR_CENT
CALL IMPR_DEC
CALL IMPR_UNI
```

```
MOVLW A' '
CALL DATOS
```

```
MOVLW A'D'
CALL DATOS
```

;Espaciado del texto

```
MOVLW 0X09
MOVWF CONT
CALL ESPACIADO
```

```
MOVLW 0XC0
CALL COMANDO
```

```
MOVLW 0X0F
MOVWF CONT
```

```

CALL    ESPACIADO

GOTO    LECTURA
;*****
;CONVERSIÓN HEXADECIMAL
;*****
HEXA

    MOVLW 0X80
    CALL  COMANDO

    CLRF  AUX
    MOVF  ADRESH,W
    MOVWF NUM_HEX
    MOVWF COPIA_HEX

    CALL  OBT_HEXA

    CALL  IMPR_TOP_H
    CALL  IMPR_SUB_H

    MOVLW A' '
    CALL  DATOS

    MOVLW A'H'
    CALL  DATOS

;Espaciado del texto

    MOVLW 0X0A
    MOVWF CONT
    CALL  ESPACIADO

    MOVLW 0XC0
    CALL  COMANDO

    MOVLW 0X10
    MOVWF CONT
    CALL  ESPACIADO

GOTO    LECTURA

;*****
;CONVERSIÓN BINARIA
;*****
BINARIO

    MOVLW 0X80
    CALL  COMANDO

    CLRF  AUX
    MOVF  ADRESH,W
    MOVWF NUM_HEX
    MOVWF COPIA_HEX

    MOVLW 0X08
    MOVWF CONT

    CALL  OBT_BIN

    MOVLW A' '
    CALL  DATOS

```



```
MOVLW A'B'
CALL  DATOS
```

;Espaciado del texto

```
MOVLW 0X05
MOVWF CONT
CALL  ESPACIADO
```

```
MOVLW 0XC0
CALL  COMANDO
```

```
MOVLW 0X10
MOVWF CONT
CALL  ESPACIADO
```

```
GOTO  LECTURA
```

```
;*****
;VOLTAJE
;*****
VOLTAJE
```

```
MOVLW 0X80
CALL  COMANDO
```

```
CLRF  AUX
MOVF  ADRESH,W
MOVWF VOLT_NUM
```

```
CALL  VOLT
```

```
CALL  IMPR_VOLT_UNI
```

```
MOVLW A'.'
CALL  DATOS
```

```
CALL  IMPR_VOLT_DEC
CALL  IMPR_VOLT_CENT
```

```
MOVLW A' '
CALL  DATOS
```

```
MOVLW A'V'
CALL  DATOS
```

;Espaciado del

texto

```
MOVLW 0X0F
MOVWF CONT
CALL  ESPACIADO
```

```
MOVLW 0XC0
CALL  COMANDO
```

```
MOVLW 0X10
MOVWF CONT
CALL  ESPACIADO
```

```
GOTO  LECTURA
```

GOTO LECTURA

```
;=====
;CONVERSIONES
;=====
```

```
;*****
;DECIMAL
;*****
```

OBT_DECIMAL

```
    MOVLW 0X64          ;W=100=0X64
    CALL DIVISION
    MOVF  AUX,W          ;W=AUX
    MOVWF CENTENAS      ;CENTENAS=AUX
```

OBT_DEC

```
    MOVLW 0X0A          ;W=10=0X0A
    CLRF  AUX            ;AUX=0X00
    CALL DIVISION
    MOVF  AUX,W          ;W=AUX
    MOVWF DECENAS       ;DECENAS=W
```

OBT_UNI

```
    MOVLW 0X01          ;W=01=0X01
    CLRF  AUX            ;AUX=0X00
    CALL DIVISION
    MOVF  AUX,W          ;W=NUM_HEX
    MOVWF UNIDADES      ;UNIDADES=W
    RETURN
```

DIVISION
decenas ;División general para obtener centenas y

```
    SUBWF NUM_HEX,F      ;NUM_HEX=NUM_HEX-W
    BTFSS STATUS,C       ;Revisar si existe Bit de acarreo
    GOTO  AJUSTE         ;Si no hay bit ir a ajuste
    INCF  AUX            ;Si hay bit de acarreo
                        ;aumento a la cantidad de
```

iteraciones realizadas

```
    GOTO DIVISION      ;Repetir DIVISION
```

AJUSTE

```
    ;Ajuste para resta que no completa
    ;centena o decena
```

```
    ADDWF NUM_HEX      ;Regresa una iteración errónea
    RETURN
```

IMPR_CENT

;IMPRESIÓN DE CENTENAS

```
    MOVF  CENTENAS,W
    MOVWF AUX
    GOTO  IMPR
```

IMPR_DEC

;IMPRESIÓN DE DECENAS

```
    MOVF  DECENAS,W
    MOVWF AUX
    GOTO  IMPR
```

IMPR_UNI

;IMPRESIÓN DE UNIDADES

```
    MOVF  UNIDADES,W
    MOVWF AUX
    GOTO  IMPR
```

```
;*****
```

```
;HEXADECIMAL
```

```
;*****
```

OBT_HEX

```

        MOVLW 0X04
        MOVWF CONT
OBT_TOP_H                                ;OBTENER TOP DE NUM HEXADECIMAL
        RRF          COPIA_HEX,F
        BCF          STATUS,C
        DECFSZ CONT
        GOTO  OBT_TOP_H
        MOVF  COPIA_HEX,W
        MOVWF TOP_H

        MOVLW 0X04
        MOVWF CONT
        MOVF  NUM_HEX,W
        MOVWF COPIA_HEX

OBT_SUB_H                                ;OBTENER SUB DE NUM HEXADECIMAL
        RLF          COPIA_HEX,F
        BCF          STATUS,C
        DECFSZ CONT
        GOTO  OBT_SUB_H
        SWAPF COPIA_HEX
        MOVF  COPIA_HEX,W
        MOVWF SUB_H
        RETURN

IMPR_TOP_H                                ;IMPRESIÓN PARTE SUPERIOR HEX
        MOVF  TOP_H,W
        MOVWF AUX
        GOTO  IMPR

IMPR_SUB_H                                ;IMPRESIÓN PARTE INFERIOR HEX
        MOVF  SUB_H,W
        MOVWF AUX
        GOTO  IMPR

;*****
;BINARIO
;*****
OBT_BIN
        RLF          COPIA_HEX
        BTFSS  STATUS,C
        GOTO  CASE_0
        GOTO  CASE_1

CASE_0                                ;IMPRESIÓN DE 0
        CALL  ZERO
        GOTO  LOOP_BIN

CASE_1                                ;IMPRESIÓN DE 1
        CALL  ONE
        GOTO  LOOP_BIN

LOOP_BIN
        DECFSZ CONT
        GOTO  OBT_BIN
        RETURN

;*****
;VOLTAJE
;*****
VOLT
        CLRF VOLT_UNIDAD

```

```

    CLR F VOLT_DECIMA
    CLR F VOLT_CENTENA

    MOVLW 0X33

VOLT_UNI_OBT                                ;OBTENCIÓN DE UNIDAD DE VOLTS
    SUBWF VOLT_NUM
    BTFSS STATUS,C
    GOTO VOLT_DEC
    INCF VOLT_UNIDAD
    GOTO VOLT_UNI_OBT

VOLT_DEC                                    ;Ajuste al contador y multiplicador
    CALL AJUSTE_VOLT
    CALL MULT_VOLT

VOLT_DEC_OBT                                ;OBTENCIÓN DE DECIMAL
    SUBWF VOLT_NUM                            ;DIVISIÓN DE VALOR
    BTFSS STATUS,C
    GOTO VOLT_DEC_2_OBT                        ;SEGUNDA PARTE
    INCF VOLT_DECIMA                            ;AUMENTO DEL VALOR DECIMAL
    GOTO VOLT_DEC_OBT

VOLT_DEC_2_OBT
    BTFSS AUX,0                                ;REVISA SI EL VALOR 0 ES 1
    GOTO VOLT_CENT                            ;IR A CENTESIMAS
    DECF AUX                                    ;DECREMENTO DEL AUX
    INCF VOLT_DECIMA                            ;AUMENTO DEL VALOR DECIMAL
    GOTO VOLT_DEC_OBT                        ;REGRESA A LA PRIMERA PARTE

VOLT_CENT                                    ;Ajuste de contador y multiplicador
    CALL AJUSTE_VOLT
    CALL MULT_VOLT

VOLT_CENT_OBT                                ;OBTENCIÓN CENTÉSIMAS
    SUBWF VOLT_NUM                            ;DIVISIÓN
    BTFSS STATUS,C
    GOTO VOLT_CENT_2_OBT                        ;SEGUNDA PARTE
    INCF VOLT_CENTENA                            ;INCREMENTO CENTESIMA
    GOTO VOLT_CENT_OBT                        ;REPETIR

VOLT_CENT_2_OBT
    BTFSS AUX,0                                ;REVISA V0 EN 1
    RETURN                                    ;REGRESA DE LA SUBROUTINA
    DECF AUX                                    ;AUX--
    INCF VOLT_CENTENA                            ;INCREMENTO CENTESIMA
    GOTO VOLT_CENT_OBT                        ;REGRESAR A LA PRIMERA PARTE

AJUSTE_VOLT                                ;AJUSTE
    PARA EL CONTADOR
    ADDWF VOLT_NUM
    MOVLW H'09'                                ;CONTADOR=09
    MOVWF CONT
    MOVF VOLT_NUM,W
    RETURN

MULT_VOLT                                    ;MULTIFICADOR
    ADDWF VOLT_NUM
    BTFSC STATUS,C

```

```

    INCF    AUX                                ; CUENTA LA
CANTIDAD DE BITS DE ACARREO OBTENIDOS
    DECFSZ  CONT
    GOTO    MULT_VOLT
    MOVLW   H'33'
    RETURN

```

```

IMPR_VOLT_UNI                                ; ENVIO DE
UNIDADES PARA IMPRESIÓN
    MOVF    VOLT_UNIDAD,W
    MOVWF   AUX
    GOTO    IMPR

```

```

IMPR_VOLT_DEC                                ; ENCIO DE DECIMAS
PARA IMPRESIÓN
    MOVF    VOLT_DECIMA,W
    MOVWF   AUX
    GOTO    IMPR

```

```

IMPR_VOLT_CENT                                ; ENVIO DE
CENTESIMAS PARA IMPRESIÓN
    MOVF    VOLT_CENTENA,W
    MOVWF   AUX
    GOTO    IMPR

```

```

;=====
;IMPRESIÓN DE NÚMEROS
;=====
ESPACIADO
    MOVLW   A' '
    CALL    DATOS
    DECFSZ  CONT
    GOTO    ESPACIADO
    RETURN

```

```

IMPR                                ; IMPRESIÓN DE CARACTERES CORRESPONDIENTE
                                ; COMPARACIONES ENTRE LOS VALORES

```

```

    MOVLW   0X00
    XORWF   AUX,W
    BTFSC   STATUS,Z
    GOTO    ZERO

```

```

    MOVLW   0X01
    XORWF   AUX,W
    BTFSC   STATUS,Z
    GOTO    ONE

```

```

    MOVLW   0X02
    XORWF   AUX,W
    BTFSC   STATUS,Z
    GOTO    TWO

```

```

    MOVLW   0X03
    XORWF   AUX,W
    BTFSC   STATUS,Z
    GOTO    THREE

```

```

    MOVLW   0X04
    XORWF   AUX,W
    BTFSC   STATUS,Z
    GOTO    FOUR

```

```

    MOVLW    0X05
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     FIVE

    MOVLW    0X06
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     SIX

    MOVLW    0X07
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     SEVEN

    MOVLW    0X08
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     EIGHT

    MOVLW    0X09
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     NINE

    MOVLW    0X0A
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     LET_A

    MOVLW    0X0B
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     LET_B

    MOVLW    0X0C
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     LET_C

    MOVLW    0X0D
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     LET_D

    MOVLW    0X0E
    XORWF    AUX,W
    BTFSC    STATUS,Z
    GOTO     LET_E

    GOTO     LET_F

;IMPRESIÓN DE LOS DATOS

ZERO
    MOVLW    A'0'
    CALL     DATOS
    RETURN

ONE
    MOVLW    A'1'
    CALL     DATOS
    RETURN

```

```
TWO
    MOVLW  A'2'
    CALL   DATOS
    RETURN

THREE
    MOVLW  A'3'
    CALL   DATOS
    RETURN

FOUR
    MOVLW  A'4'
    CALL   DATOS
    RETURN

FIVE
    MOVLW  A'5'
    CALL   DATOS
    RETURN

SIX
    MOVLW  A'6'
    CALL   DATOS
    RETURN

SEVEN
    MOVLW  A'7'
    CALL   DATOS
    RETURN

EIGHT
    MOVLW  A'8'
    CALL   DATOS
    RETURN

NINE
    MOVLW  A'9'
    CALL   DATOS
    RETURN

LET_A
    MOVLW  A'A'
    CALL   DATOS
    RETURN

LET_B
    MOVLW  A'B'
    CALL   DATOS
    RETURN

LET_C
    MOVLW  A'C'
    CALL   DATOS
    RETURN

LET_D
    MOVLW  A'D'
    CALL   DATOS
    RETURN

LET_E
    MOVLW  A'E'
    CALL   DATOS
    RETURN

LET_F
    MOVLW  A'F'
    CALL   DATOS
    RETURN
```

```

;=====
;CONFIGURACIÓN INICIAL DE DISPLAY LCD
;=====
INICIA_LCD
    MOVLW 0X30          ;Tamaño de dato (8 bits)
    CALL  COMANDO
    CALL  RETARDO_100
    MOVLW 0X30          ;Tamaño de dato (8 bits)
    CALL  COMANDO
    CALL  RETARDO_100
    MOVLW 0X38          ;Tamaño de dato (8 bits) con el uso de las 2 líneas del
display
    CALL  COMANDO
    MOVLW 0X0C          ;Encendido del display
    CALL  COMANDO
    MOVLW 0X01          ;Limpieza del display
    CALL  COMANDO
    MOVLW 0X06          ;Posición del cursor e incremento de este para la impresión
                        ;Desplazamiento a la derecha
    CALL  COMANDO
    MOVLW 0X02          ;Posición del cursor al inicio
    CALL  COMANDO
    RETURN

                        ;E=          Habilitador
                        ;R/S=0 Control
                        ;R/S=1 Datos

COMANDO
    MOVWF PORTC
    CALL  RETARDO_200
    MOVLW H'02'
    MOVWF PORTB          ;E=1 y RS=0
    CALL  RETARDO_200
    MOVLW H'00'
    MOVWF PORTB          ;E=0 y RS=0
    CALL  RETARDO_200
    CALL  RETARDO_200
    RETURN

;=====
;ENVIO DE DATOS AL DISPLAY
;=====
DATOS                                ;ENVIO DE DATOS AL PUERTO B
    MOVWF PORTC
    CALL  RETARDO_200
    MOVLW H'03'          ;E=1 y RS=1
    MOVWF PORTB
    CALL  RETARDO_200
    MOVLW H'01'          ;E=0 y RS=1
    MOVWF PORTB
    CALL  RETARDO_200
    CALL  RETARDO_200
    RETURN

;=====
;RETARDOS UTILIZADOS
;=====
RETARDO_200                        ;RETARDO DE 200 MS
    MOVLW 0X02
    MOVWF valor1
LOOP

```



```

        MOVLW  D'164'
        MOVFW  valor2
LOOP1
        DECFSZ valor2
        GOTO   LOOP1
        DECFSZ valor1
        GOTO   LOOP
        RETURN

```

```

RETARDO_100                                ;RETARDO DE 100 MS
        MOVLW  0X03
        MOVWF  valor
TRES
        MOVLW  0XFF
        MOVWF  valor1
DOS
        MOVLW  0XFF
        MOVWF  valor2
UNO
        DECFSZ valor2
        GOTO   UNO
        DECFSZ valor1
        GOTO   DOS
        DECFSZ valor
        GOTO   TRES
        RETURN

```

```

        END

```

Diagramas de Flujo

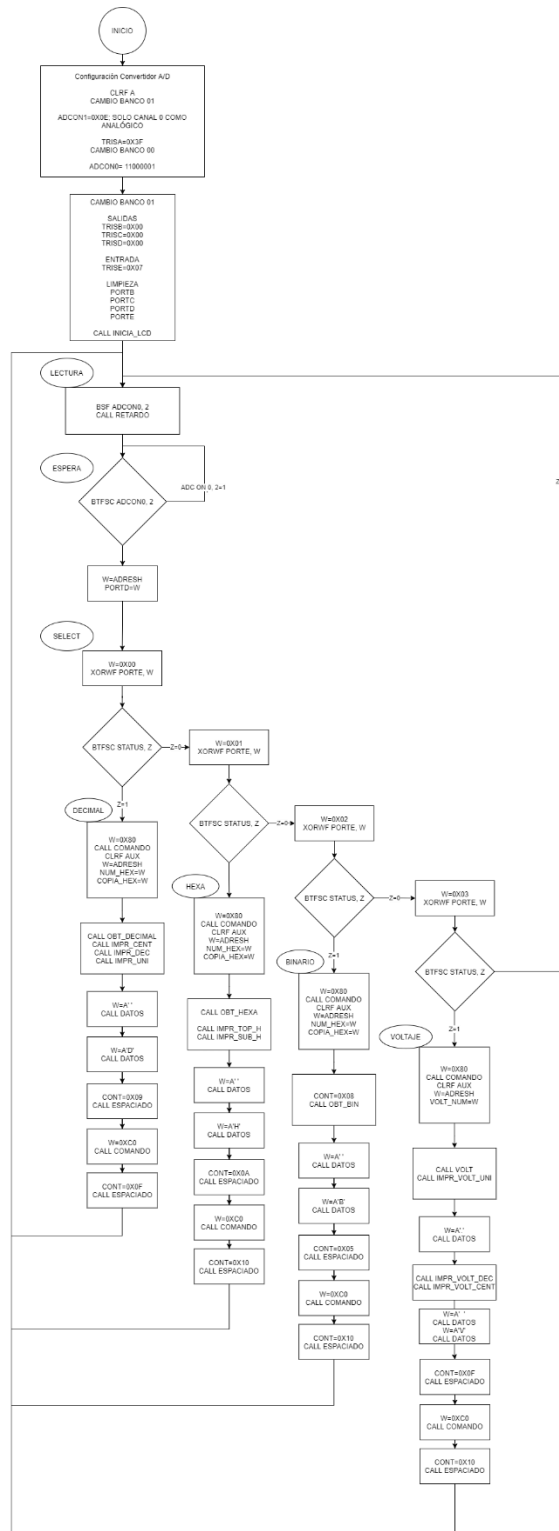


Diagrama 1: Proyecto base, configuración inicial, inicialización Convertidor A/D y segmento de selección

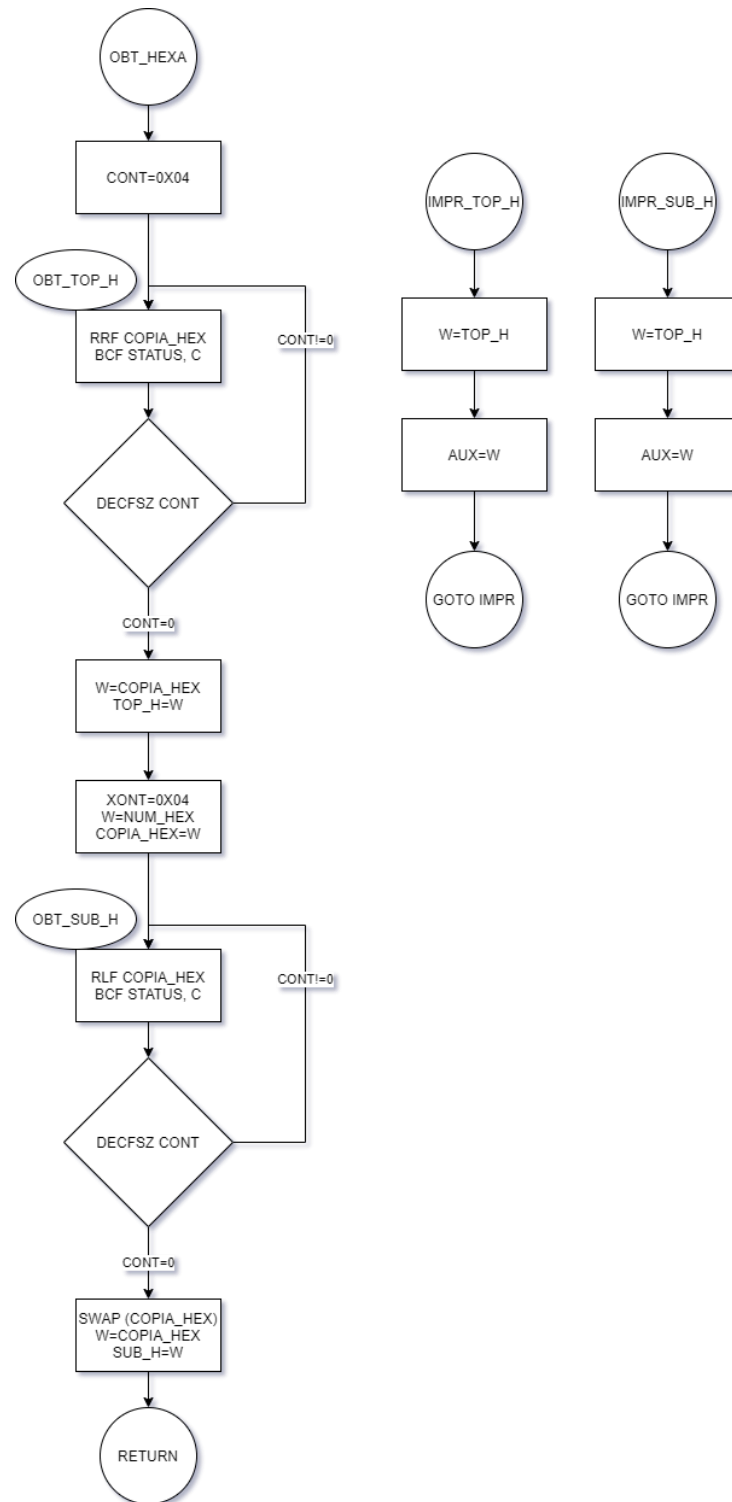


Diagrama 4: Conversión Hexadecimal

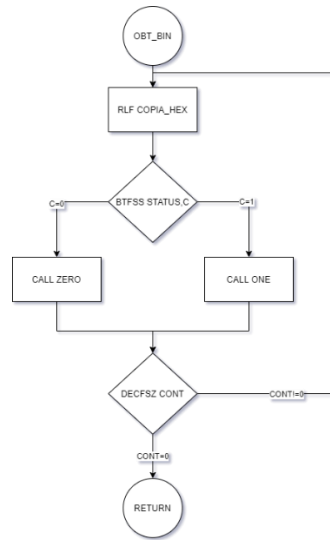


Diagrama 5: Conversión Binaria

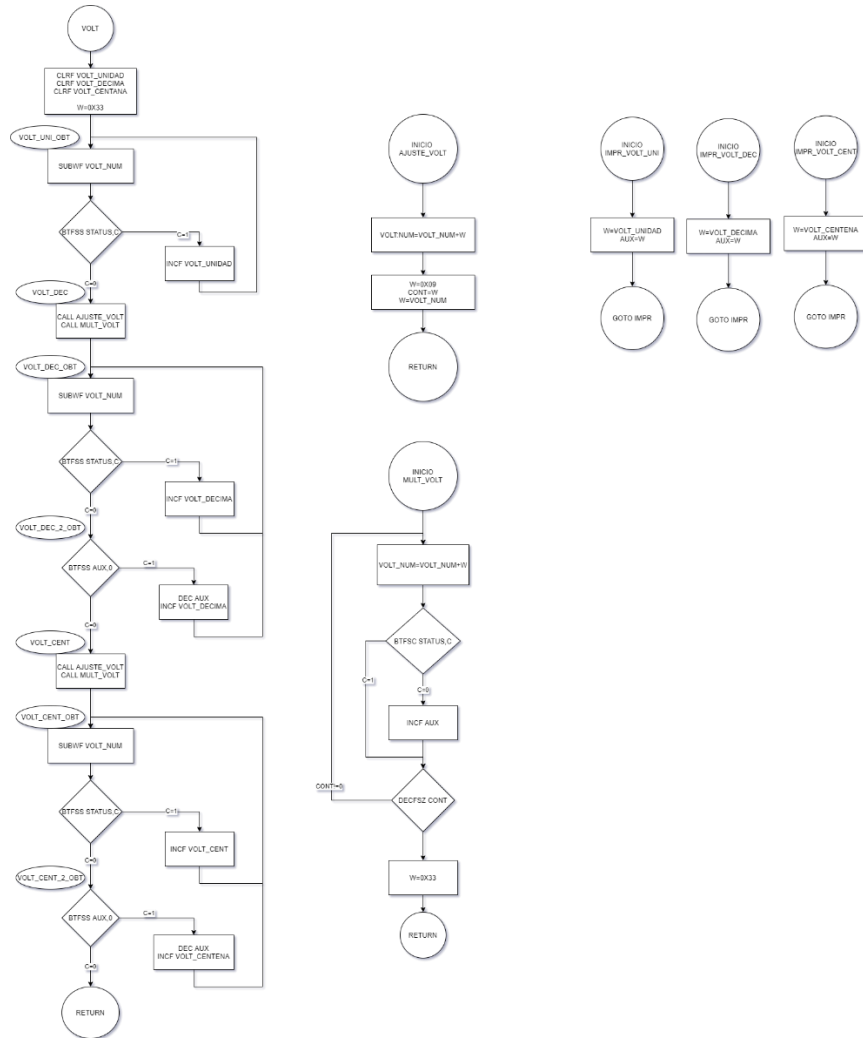


Diagrama 6: Conversión a Volts

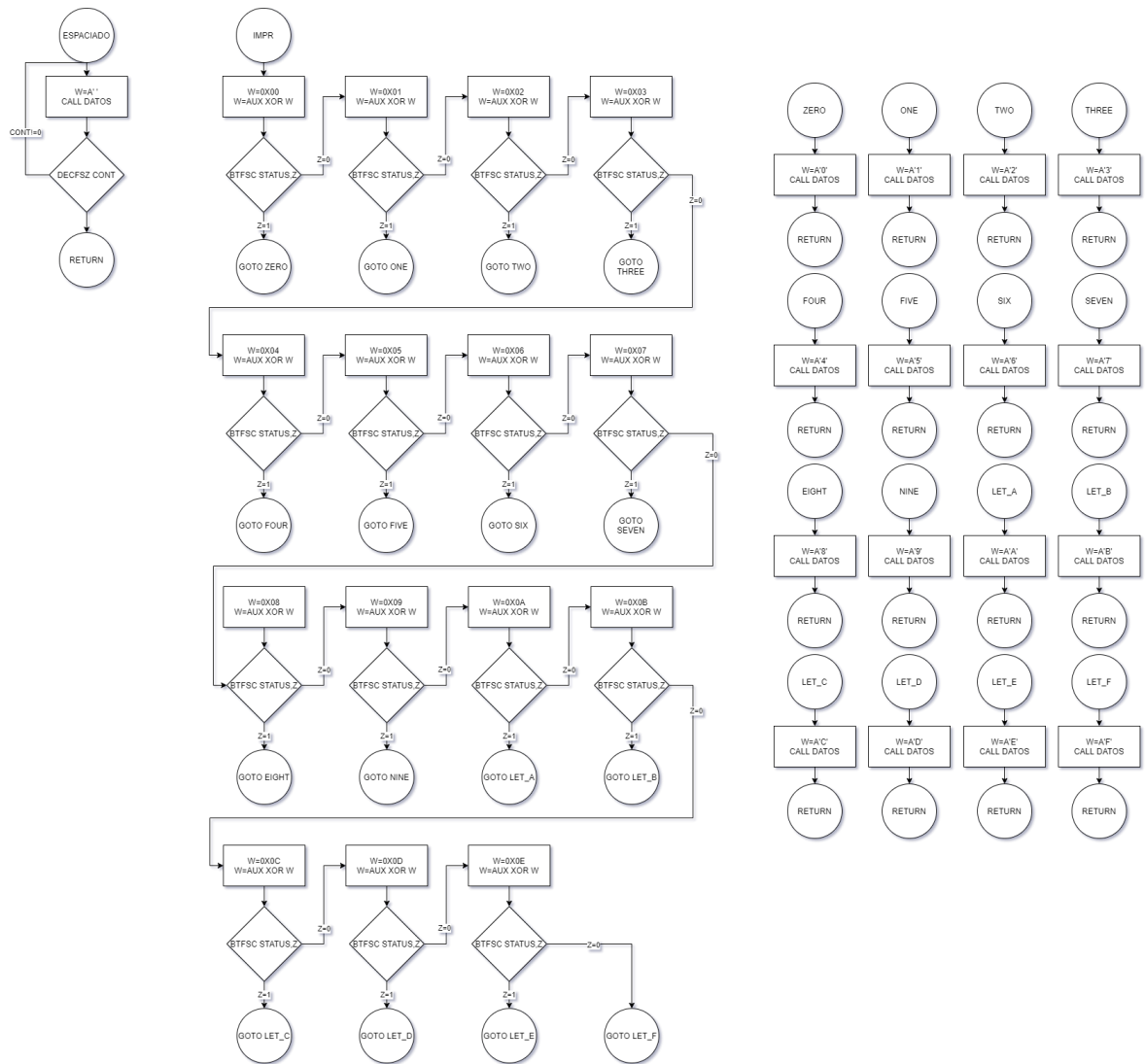
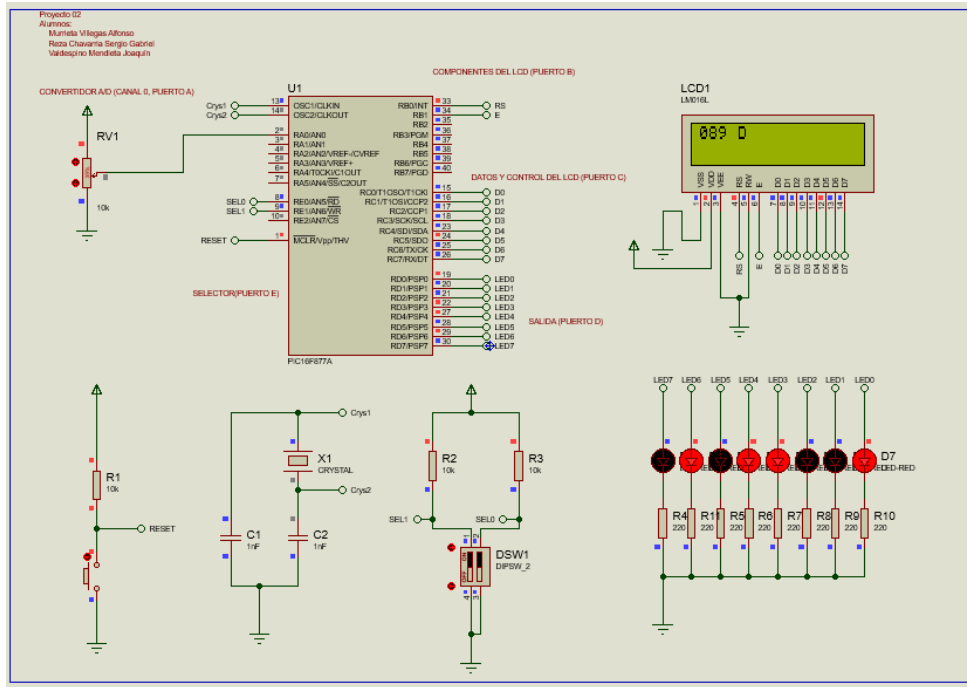
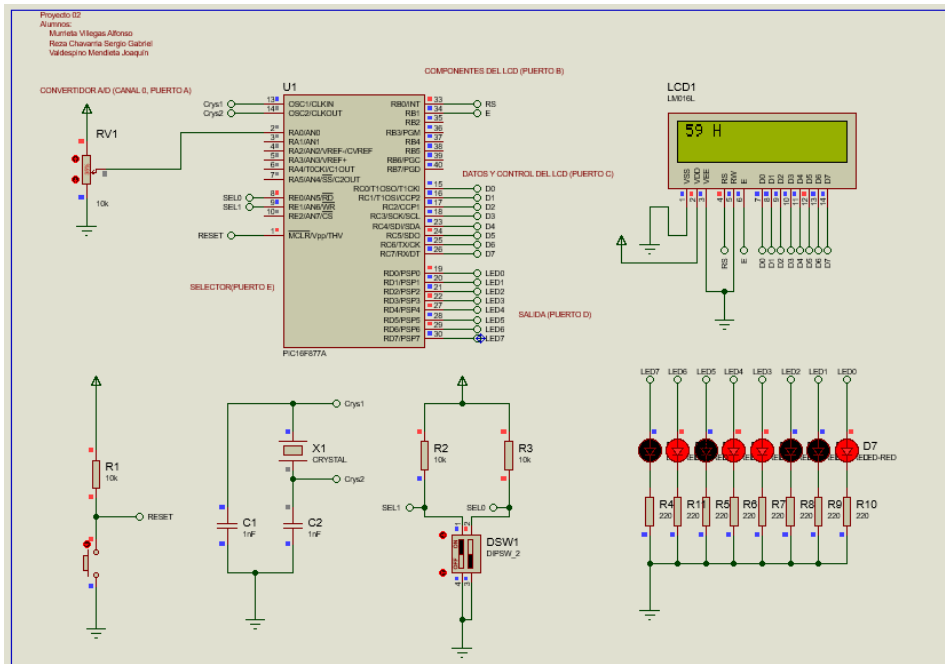


Diagrama 7: Espaciado e Impresión de datos

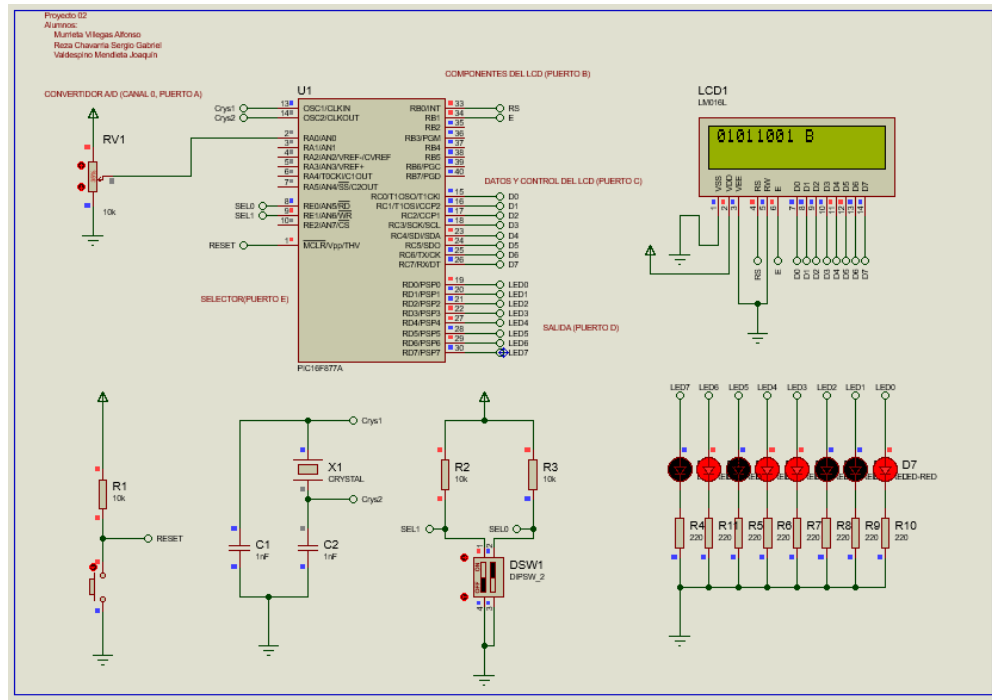
Pruebas



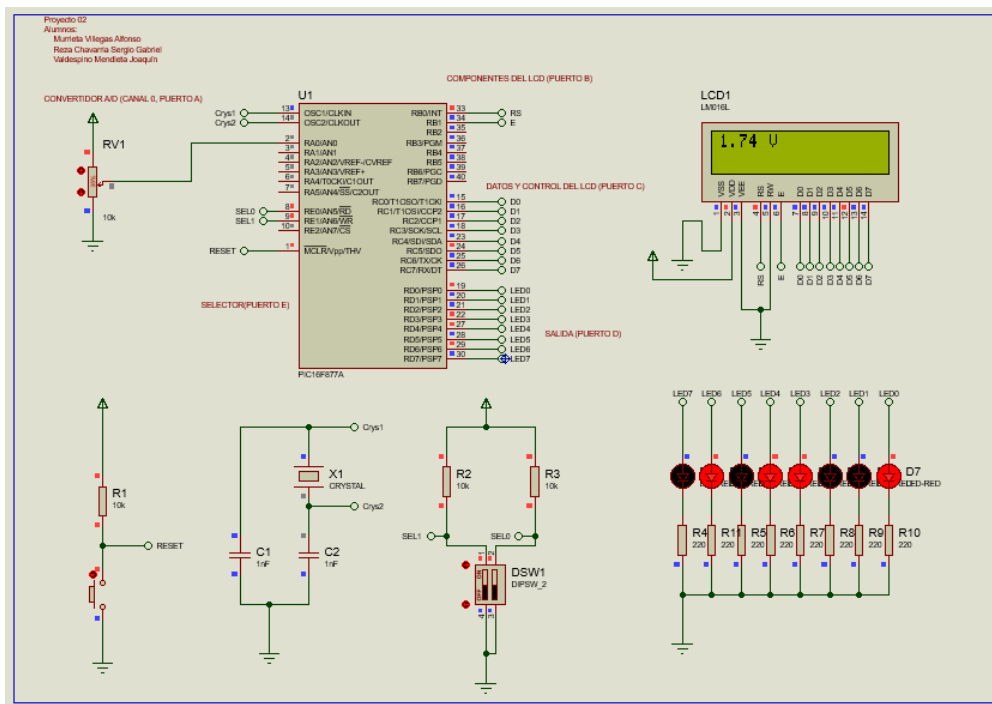
Pruebas 1: Conversión decimal (Potenciómetro 35%)



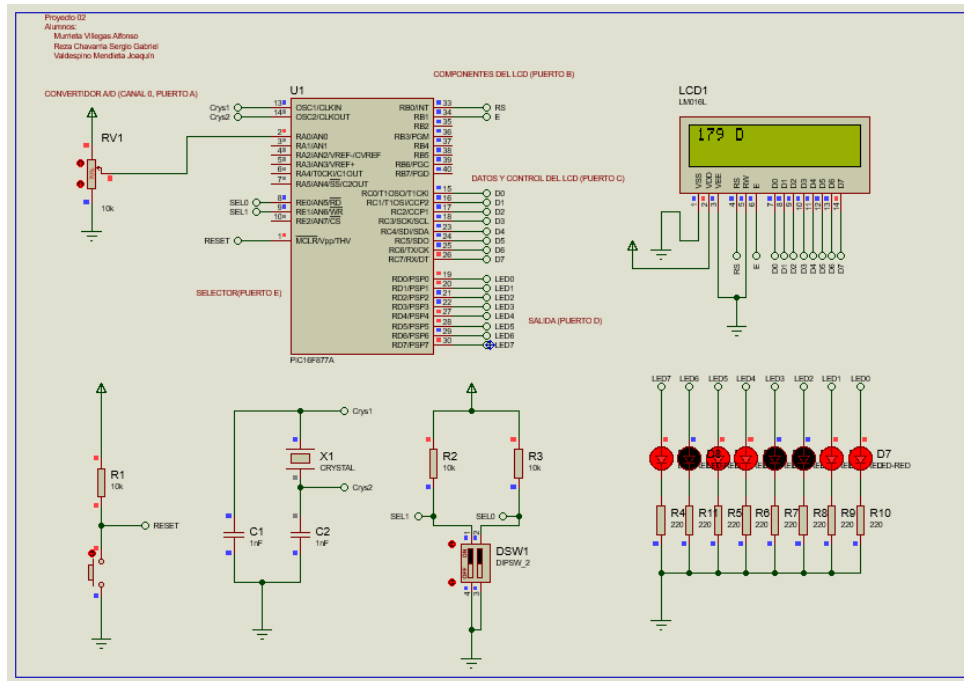
Pruebas 2: Conversión Hexadecimal (Potenciómetro 35%)



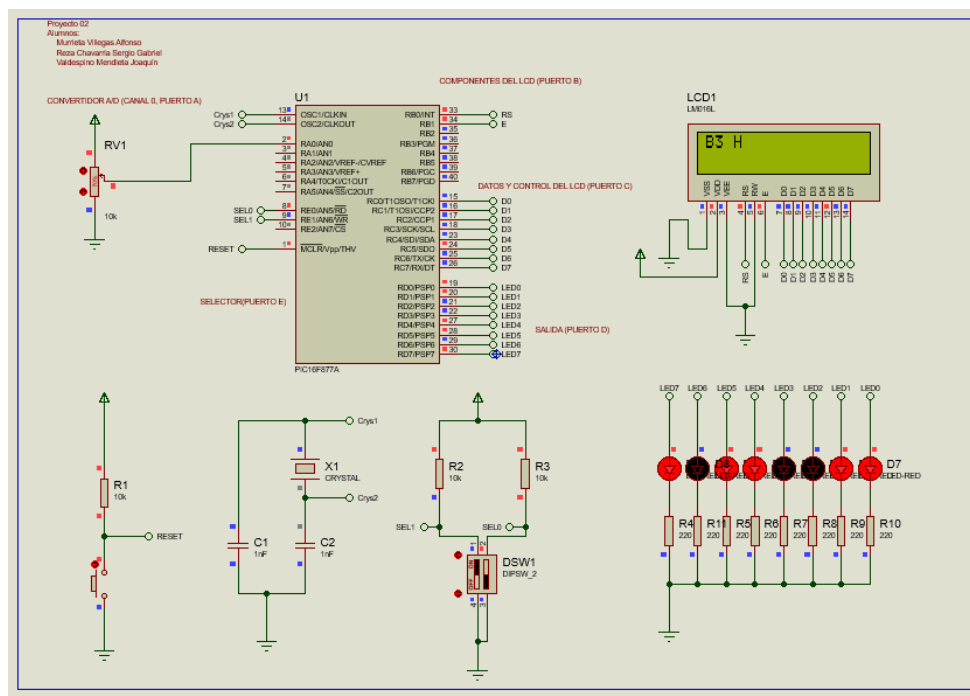
Pruebas 3: Conversión Binaria (Potenciómetro 35%)



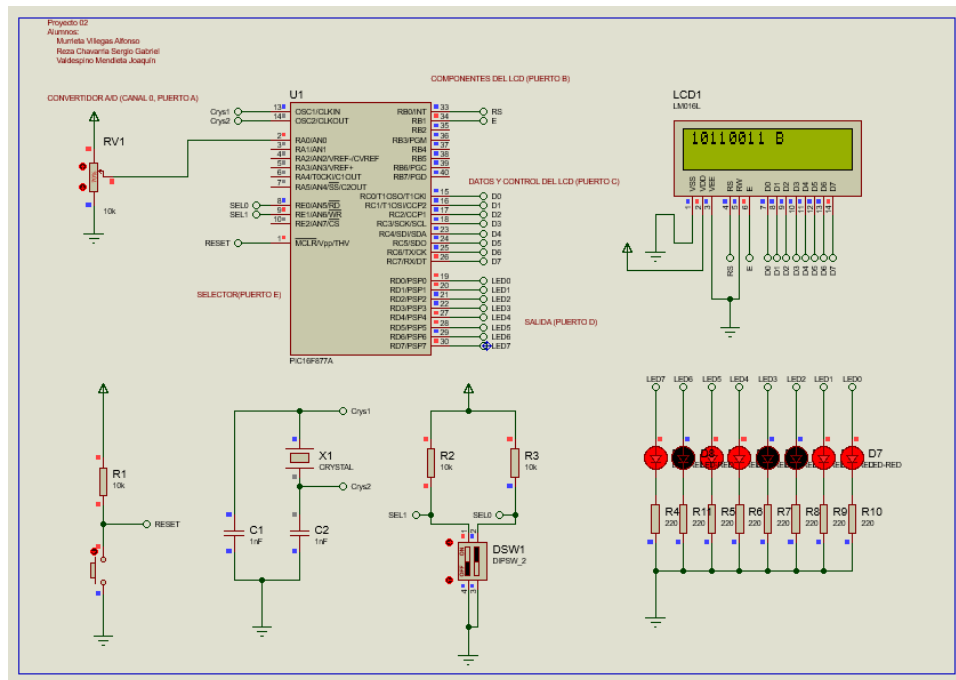
Pruebas 4: Conversión a Voltaje (Potenciómetro 35%)



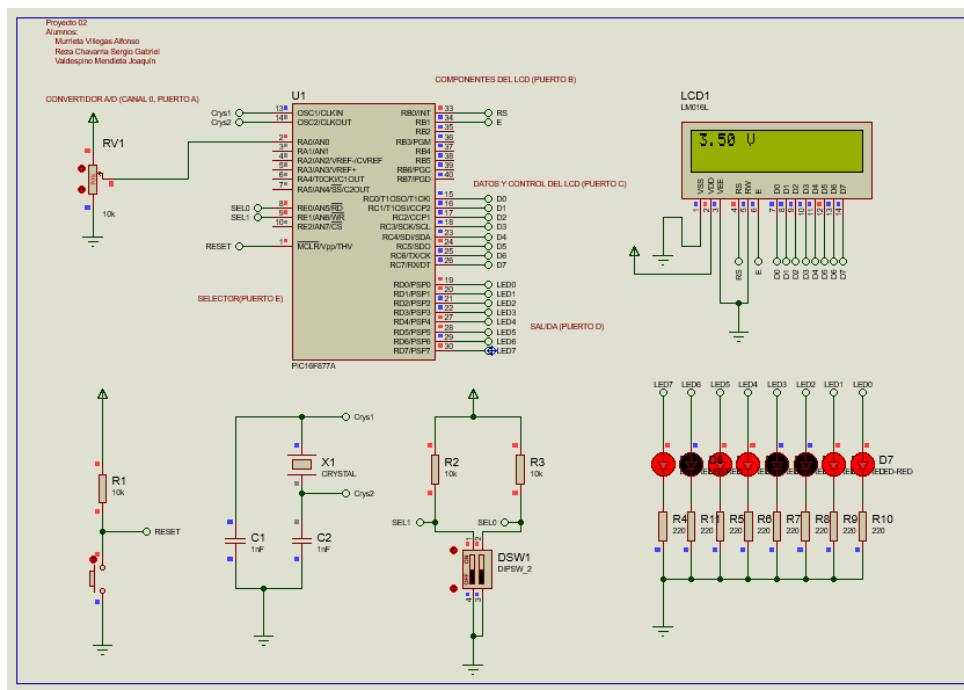
Pruebas 5: Conversión Decimal (Potenciómetro 70%)



Pruebas 6: Conversión Hexadecimal (Potenciómetro 70%)



Pruebas 7: Conversión Binario (Potenciómetro 70%)



Pruebas 8: Conversión a Voltaje (Potenciómetro 70%)

Conclusiones

Murrieta Villegas Alfonso

En el presente proyecto aprendimos y aplicamos el uso del conversor analógico - digital o ADC con el objetivo de profundizar nuestras habilidades de programación en ensamblador como nuestra previa experiencia con electrónica digital.

Por último, con base a proyectos y prácticas previas es como partimos a realizar un proyecto con una mayor complejidad respecto al uso de subrutinas y sentencias dedicadas exclusivamente a desplegar información específica en una pantalla

Reza Chavarria Sergio Gabriel

A partir del uso del convertidor se puede utilizar la información del exterior, de diferentes tipos de fuentes, para poder llevarlo a su análisis, compresión y uso de estos en nuestros proyectos.

Además, se pudo adaptar las rutinas realizadas en el proyecto anterior, de forma modular, para utilizarlos de una manera similar, pero con nuevos propósitos. Esto da la ventaja de apoyarse de material o conocimiento previo.

Valdespino Mendieta Joaquin

En el presente proyecto podemos observar y aplicar los conocimientos de otras áreas, como circuitos para poder entender y procesar la información de entidades externas, de fuentes diversas, en este caso en la elaboración de un voltámetro, desde bajo nivel, con esto podemos concluir que el saber cómo programar los microcontroladores, podemos desarrollar aplicativos, útiles para otras áreas, como el caso de electrónica.