

DIRECCIÓN GENERAL DE EDUCACIÓN TECNOLÓGICA INDUSTRIAL





"Francisco I. Madero" Carrera: Mecatrónica

Reporte de la Actividad 23. ADC Convertidor de Analógico a Digital

NOMBRE: Orlando Contreras Reyes

NL: 6

Título: ADC Convertidor de Analógico a Digital

a) El diagrama electrónico. Realízalo a mano. ADC Volmetro LM041L SS GH U1 RV1 OSC1/CLKI MCLR/VPP RC0/T10S0/T1CK RC1/T10SI/CCP2A RC2/CCP1 RC3/SCK/SCL RA1/AN1 RC4/SDI/SDA RC5/SDO RC6/TX/CK RA2/AN2/VREF-RA3/AN3/VREF+ S & S D6 D5 RA4/T0CKI RC7/RX/DT RA5/AN4/SS/LVDIN RA6/OSC2/CLKO RD0/PSP0

RD1/PSP1 RD2/PSP2

RD3/PSP3 RD4/PSP4

RD5/PSP5

RD6/PSP6 RD7/PSP7

RE0/RD/AN5 RE1/WR/AN6 RE2/CS/AN7

RB0/INT0

RB3/CCP2B

RB4 RB5/PGM

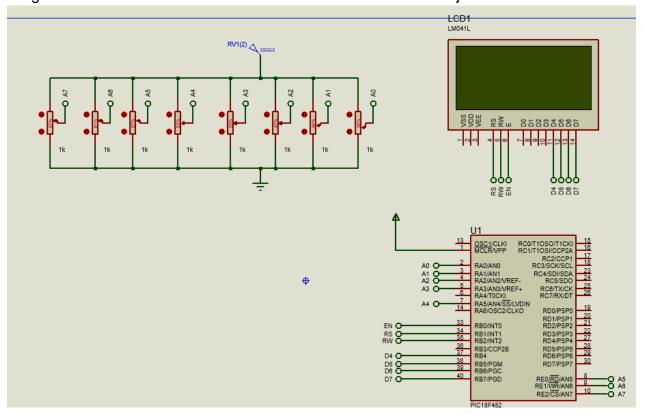
RB6/PGC RB7/PGD

PIC18F452

```
b) Código en CCS SIN LIBRERÍA
   //----- MAIN LIBRARY -----
                                                     //--Inicio--
                                                     void main(){
   #include <18F452.h>
   //----- ADC CONFIGURATION ------
                                                          //Set Outputs
   #device adc=10
                                                          setup_adc_ports(AN0);//
   //----- FUSES CONFIGURATION ------
                                                     setup_adc(ADC_CLOCK_INTERNAL);//
   NOWDT, HS, PUT, NOPROTECT, NOBR
                                                     Set the speed of clock
   OWNOUT, NOLVP, NOCPD
                                                          lcd init();//initialize the lcd
   #use delay(clock=4MHz)
                                                          lcd_gotoxy(1,1);
   //----- EXT LIBRARIES -----
                                                            printf(lcd_putc,"Orlando");
                                                           lcd_gotoxy(1,2);
   #include <lcd420.c>
   //----- SET OUTPUTS -----
                                                            printf(lcd_putc,"Contreras");
   //--Ports-
   //--Var--
                                                          //Infinite Loop
   int16 var_adc;
                                                          while(true){
   float var_analog;
```

```
set_adc_channel(0);
                               //Enable
                                                           lcd_gotoxy(17,2);//row 2
                                                           printf(lcd_putc,"Volt=
Channel 0
                                                     %01.2fV",var analog);
       delay_us(20);
                                                           delay_ms(100);
     var_adc=read_adc();//save
                                     the
value of adc on "var adc"
                                                           }//end while
     var_analog=5.0*var_adc/1024.0;
                                                     }//end main
     lcd_gotoxy(17,1);//row 1
printf(lcd_putc,"ADC=%4ld",var_adc);
```

- c) Explicación del código
- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz, se declara la resolución del ADC que será de 10 bits, se incluye la librería LCD420.C porque se usará un LCD de 4x16.
- 3) Se declara como entero la variable del adc (Var_adc) y como flotante el voltaje o variable analógica (var_analog)-
- 4) En el void main se configuran los pines analógicos a utilizar en este caso se utilizará el pin A0 además de configurar la velocidad como la del reloj interno y se inicializa la lcd. Se imprime el mensaje Orlando en la primer fila y en la segunda de imprime Contreras
- 5) En el bucle infinito se configura el canal en el que vamos a trabajar como canal 0 (Esto es porque utilizaremos el Pin 0 analógico) se guarda en una variable (Var_adc) el valor obtenido del pin analógico. La variable Var_Adc es multiplicada por 5.0 (Voltaje total) y dividida por 1024 (Bits de resolución) y es guardado en la variable analógica (Var_Analog).
- 6) Finalmente en la tercer fila se imprime el valor obtenido del pin analógico y en la 4ta se imprime el valor Analógico que se obtuvo al ser multiplicado por 5 y dividido entre 1024
- d) Diagrama Electronico Todos los canales Valor ADC/Valor Voltaje



e) Código en CCS Valor ADC

```
//----- MAIN LIBRARY ------
                                                           set_adc_channel(adc);
#include <18F452.h>
                                                           delay_us(20);
                                                           array[adc]=read_adc();
//----- ADC CONFIGURATION ------
#device adc=10
//----- FUSES CONFIGURATION ------
                                                         lcd gotoxy(1,1);
#fuses
                                                    printf(lcd_putc,"A1=%4ld",array[0]);
NOWDT,HS,PUT,NOPROTECT,NOBR
OWNOUT, NOLVP, NOCPD
                                                         lcd_gotoxy(9,1);
#use delay(clock=4MHz)
//----- EXT LIBRARIES -----
                                                    printf(lcd_putc,"A2=%4ld",array[1]);
#include <lcd420.c>
                                                         lcd_gotoxy(1,2);
//----- SET OUTPUTS -----
//--Ports-
                                                    printf(lcd_putc,"A3=%4ld",array[2]);
//--Var--
                                                         lcd_gotoxy(9,2);
int16 array[7],adc;
//--Inicio--
                                                    printf(lcd_putc,"A4=%4ld",array[3]);
                                                         lcd_gotoxy(17,1);
void main(){
                                                    printf(lcd_putc,"A5=%4ld",array[4]);
     //Set Outputs
                                                         lcd_gotoxy(25,1);
setup_adc_ports(ALL_ANALOG);//Set
                                                    printf(lcd_putc,"A6=%4ld",array[5]);
the ports
                                                         lcd_gotoxy(17,2);
setup_adc(ADC_CLOCK_INTERNAL);//
Set the speed of clock
                                                    printf(lcd_putc,"A7=%4ld",array[6]);
     lcd_init();//initialize the lcd
                                                         lcd_gotoxy(25,2);
     //Infinite Loop
                                                    printf(lcd_putc,"A8=%4ld",array[7]);
     while(true){
                                                         }//end while
     for(adc=0;adc<8;adc++){
                                                    }//end main
```

- f) Explicación del codigo Valor ADC
- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz, se declara la resolución del ADC que será de 10 bits, se incluye la librería LCD420.C porque se usará un LCD de 4x16.
- 3) Se declara como entero la variable del adc (Var_adc) y el array de 8 posiciones en el que se guardaran los valores obtenidos en cada pin.
- 4) En el void main se configuran los pines analógicos a utilizar en este caso se utilizarán todos los pines, además de configurar la velocidad como la del reloj interno y se inicializa la lcd.

- 5) En el bucle infinito introducimos un for que servirá para repetir la misma instrucción para todos los pines, la instrucción que repetiremos será configurar el canal en el que trabajaremos como la variable que incrementaremos después de cada vuelta (adc) para después guardarla en el array en la posición de la variable (adc). Esta operación se repetirá 7 veces
- 6) Cuando salga de la sentencia imprimirá en la LCD A1/A2/A3/A4/A5/A6/A7/A8 = y aquí pondremos el nombre del array en la posición correspondiente (para pin 0 posición 0 para pin 1 posición 1...).

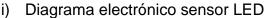
```
g) Codigo en CCS Valor Voltaje
   //----- MAIN LIBRARY -----
                                                              set_adc_channel(adc);
   #include <18F452.h>
                                                              delay_us(20);
   //----- ADC CONFIGURATION ------
                                                              array[adc]=read_adc();
   #device adc=10
   //----- FUSES CONFIGURATION ------
                                                       array2[analog]=5.0*array[adc]/1024.0;
   #fuses
   NOWDT, HS, PUT, NOPROTECT, NOBR
   OWNOUT, NOLVP, NOCPD
                                                            lcd_gotoxy(1,1);
   #use delay(clock=4MHz)
   //----- EXT LIBRARIES -----
                                                       printf(lcd_putc,"1=%01.2fV",array2[0]);
   #include <lcd420.c>
                                                            lcd_gotoxy(10,1);
   //----- SET OUTPUTS -----
                                                       printf(lcd putc,"2=%01.2fV",array2[1]);
   //--Ports-
                                                            lcd_gotoxy(1,2);
   //--Var--
   int16 array[7];
                                                       printf(lcd_putc,"3=%01.2fV",array2[2]);
   float array2[7];
                                                            lcd_gotoxy(10,2);
   int16 analog,adc;
   //--Inicio--
                                                       printf(lcd_putc,"4=%01.2fV",array2[3]);
                                                            lcd qotoxy(17,1);
   void main(){
                                                       printf(lcd_putc,"5=%01.2fV",array2[4]);
        //Set Outputs
                                                            lcd gotoxy(26,1);
   setup_adc_ports(ALL_ANALOG);//Set
   the ports
                                                       printf(lcd putc,"6=%01.2fV",array2[5]);
                                                            lcd_gotoxy(17,2);
   setup_adc(ADC_CLOCK_INTERNAL);//
   Set the speed of clock
                                                       printf(lcd_putc,"7=%01.2fV",array2[6]);
        lcd_init();//initialize the lcd
                                                            lcd_gotoxy(26,2);
                                                       printf(lcd_putc,"8=%01.2fV",array2[7]);
        //Infinite Loop
        while(true){
                                                            }//end while
                                                       }//end main
   for(adc=0,analog=0;adc<7;adc++,analo
```

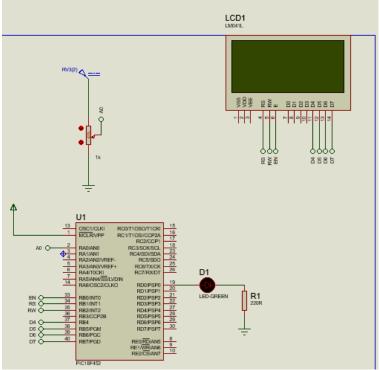
h) Explicación del código Valor Voltaje

 $q++){$

- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz, se declara la resolución del ADC que será de 10 bits, se incluye la librería LCD420.C porque se usará un LCD de 4x16.
- 3) Se declara como entero la variable del adc (Var_adc) y el array de 8 posiciones en el que se guardaran los valores obtenidos en cada pin.

- 4) En el void main se configuran los pines analógicos a utilizar en este caso se utilizarán todos los pines, además de configurar la velocidad como la del reloj interno y se inicializa la lcd.
- 5) En el bucle infinito introducimos un for que servirá para repetir la misma instrucción para todos los pines, la instrucción que repetiremos será configurar el canal en el que trabajaremos como la variable que incrementaremos después de cada vuelta (adc) para después guardarla en el array en la posición de la variable (adc). Esta operación se repetirá 7 veces
- 6) Cuando salga de la sentencia imprimirá en la LCD A1/A2/A3/A4/A5/A6/A7/A8 = y aquí pondremos el nombre del array en la posición correspondiente (para pin 0 posición 0 para pin 1 posición 1...).





```
#include <lcd420.c>
//------ SET OUTPUTS -----
//--Ports-
#byte Port_D = 0X0F83
#byte Tris_D = 0x0F95
//--Var--
int16 var_adc;
float var_analog;
//--Inicio--
```

```
void main(){
                                                           var_adc=read_adc();//save
                                                                                           the
     //Set Outputs
                                                     value of adc on "var adc"
     Tris D=0x00;
                                                           var analog=5.0*var adc/1024.0;
     Port D=0x00:
                                                           lcd_gotoxy(17,1);
     setup_adc_ports(AN0);//
                                                     printf(lcd putc,"2.0V>%01.2fV",var anal
setup_adc(ADC_CLOCK_INTERNAL);//
                                                     og);
Set the speed of clock
                                                           if(var analog>2.0){
     lcd_init();//initialize the lcd
                                                             Port_D=0xFF;
                                                            lcd_gotoxy(17,2);
                                                            printf(lcd_putc,"LED ON ");
     lcd gotoxy(1,1);
       printf(lcd_putc,"Orlando");
      lcd_gotoxy(1,2);
                                                           else{
                                                            Port_D=0x00;
       printf(lcd_putc,"Contreras
Reyes");
                                                            lcd_gotoxy(17,2);
     //Infinite Loop
                                                            printf(lcd putc,"LED OFF");
     while(true){
     set adc channel(0);
                                //Enable
                                                           }//end while
Channel 0
       delay_us(20);
                                                     }//end main
```

- k) Explicación del código Sensor LED
- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz, se declara la resolución del ADC que será de 10 bits, se incluye la librería LCD420.C porque se usará un LCD de 4x16.
- 3) Se usará el puerto D así que se declara el TRIS_D como PORT_D además de declarar como entero la variable del adc (var_adc) y como flotante el valor analógico (var_analog) además se configuran los pines analógicos a utilizar en este caso se utilizarán todos los pines, además de configurar la velocidad como la del reloj interno y se inicializa la lcd.
- 4) Se imprime un mensaje que contiene "Orlando" en la primera fila y "Contreras Reyes" en la 2da fila, después entra el bucle infinito
- 5) En el bucle infinito se setea el canal 0 del ADC y se lee el valor para almacenarlo en la variable adc (var_adc). Después esa variable es multiplicada por 5.0 y dividida por 1024 y guardada en euna variable flotante. Esa variable se imprimirá, después se evaluará si esa variable es mayo a 2.0
- 6) Si es mayor a 2.0 Encenderá un led e imprimirá LED ON. Y si no lo apagará e imprimirá LED OFF