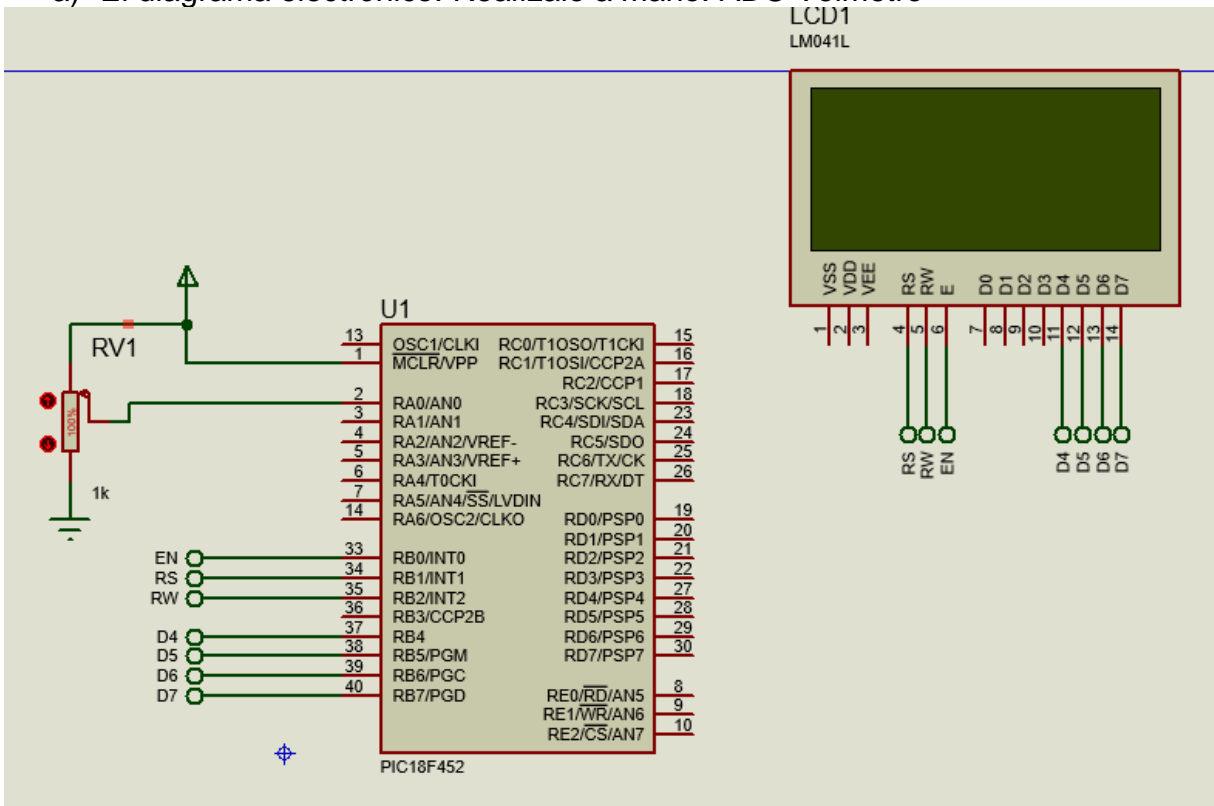


Reporte de la Actividad 23. ADC Convertidor de Analógico a Digital

NOMBRE: Orlando Contreras Reyes	NL: 6
--	--------------

Título: **ADC Convertidor de Analógico a Digital**

a) El diagrama electrónico. Realízalo a mano. ADC Volmetro



b) Código en CCS SIN LIBRERÍA

```
//----- MAIN LIBRARY -----
#include <18F452.h>
//----- ADC CONFIGURATION -----
#device adc=10
//----- FUSES CONFIGURATION -----
#fuses
NOWDT,HS,PUT,NOPROTECT,NOBR
OWNOUT,NOLVP,NOCPD
#use delay(clock=4MHz)
//----- EXT LIBRARIES -----
#include <lcd420.c>
//----- SET OUTPUTS -----
//--Ports-
//--Var--
int16 var_adc;
float var_analog;
```

```
//--Inicio--
void main(){
    //Set Outputs
    setup_adc_ports(AN0);

    setup_adc(ADC_CLOCK_INTERNAL);
    Set the speed of clock
    lcd_init();//initialize the lcd
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Orlando");
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Contreras");

    //Infinite Loop
    while(true){
```

```

        set_adc_channel(0);      //Enable
Channel 0
        delay_us(20);

        var_adc=read_adc();//save    the
value of adc on "var_adc"
        var_analog=5.0*var_adc/1024.0;

        lcd_gotoxy(17,1);//row 1

printf(lcd_putc,"ADC=%4ld",var_adc);

        lcd_gotoxy(17,2);//row 2
        printf(lcd_putc,"Volt=
%01.2fV",var_analog);
        delay_ms(100);

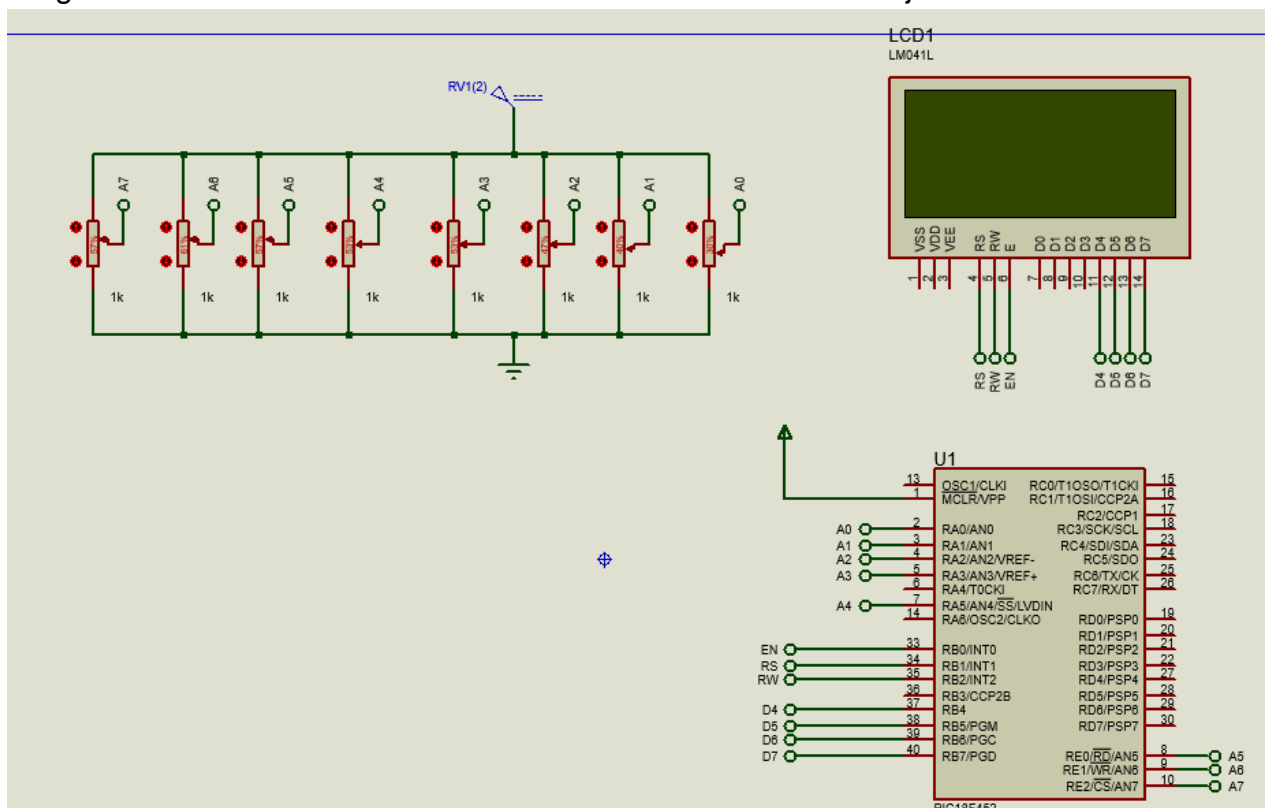
    }//end while

}//end main

```

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 -----
#include <18F452.h>
//----- ADC CONFIGURATION -----
#device adc=10
//----- FUSES CONFIGURATION -----
#fuses
NOWDT,HS,PUT,NOPROTECT,NOBR
OWNOUT,NOLVP,NOCPD
#use delay(clock=4MHz)
//----- EXT LIBRARIES -----
#include <lcd420.c>
//----- SET OUTPUTS -----
//--Ports-
//--Var--
int16 array[7],adc;
//--Inicio--

void main(){
    //Set Outputs

    setup_adc_ports(ALL_ANALOG);//Set
    the ports

    setup_adc(ADC_CLOCK_INTERNAL);//
    Set the speed of clock
    lcd_init();//initialize the lcd

    //Infinite Loop
    while(true){

        for(adc=0;adc<8;adc++){

                set_adc_channel(adc);
                delay_us(20);
                array[adc]=read_adc();
            }
            lcd_gotoxy(1,1);

            printf(lcd_putc,"A1=%4ld",array[0]);
            lcd_gotoxy(9,1);

            printf(lcd_putc,"A2=%4ld",array[1]);
            lcd_gotoxy(1,2);

            printf(lcd_putc,"A3=%4ld",array[2]);
            lcd_gotoxy(9,2);

            printf(lcd_putc,"A4=%4ld",array[3]);
            lcd_gotoxy(17,1);

            printf(lcd_putc,"A5=%4ld",array[4]);
            lcd_gotoxy(25,1);

            printf(lcd_putc,"A6=%4ld",array[5]);
            lcd_gotoxy(17,2);

            printf(lcd_putc,"A7=%4ld",array[6]);
            lcd_gotoxy(25,2);

            printf(lcd_putc,"A8=%4ld",array[7]);
            //end while

        }
    }
}
```

f) Explicación del código 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) Código en CCS Valor Voltaje

```
//----- MAIN LIBRARY -----
#include <18F452.h>
//----- ADC CONFIGURATION -----
#device adc=10
//----- FUSES CONFIGURATION -----
#fuses
NOWDT,HS,PUT,NOPROTECT,NOBR
OWNOUT,NOLVP,NOCPD
#use delay(clock=4MHz)
//----- EXT LIBRARIES -----
#include <lcd420.c>

//----- SET OUTPUTS -----
//--Ports-
//--Var--
int16 array[7];
float array2[7];
int16 analog,adc;
//--Inicio--

void main(){
    //Set Outputs

    setup_adc_ports(ALL_ANALOG);//Set
    the ports

    setup_adc(ADC_CLOCK_INTERNAL);//
    Set the speed of clock
    lcd_init();//initialize the lcd

    //Infinite Loop
    while(true){

        set_adc_channel(adc);
        delay_us(20);
        array[adc]=read_adc();

        array2[analog]=5.0*array[adc]/1024.0;
    }

    lcd_gotoxy(1,1);

    printf(lcd_putc,"1=%01.2fV",array2[0]);
    lcd_gotoxy(10,1);

    printf(lcd_putc,"2=%01.2fV",array2[1]);
    lcd_gotoxy(1,2);

    printf(lcd_putc,"3=%01.2fV",array2[2]);
    lcd_gotoxy(10,2);

    printf(lcd_putc,"4=%01.2fV",array2[3]);
    lcd_gotoxy(17,1);

    printf(lcd_putc,"5=%01.2fV",array2[4]);
    lcd_gotoxy(26,1);

    printf(lcd_putc,"6=%01.2fV",array2[5]);
    lcd_gotoxy(17,2);

    printf(lcd_putc,"7=%01.2fV",array2[6]);
    lcd_gotoxy(26,2);

    printf(lcd_putc,"8=%01.2fV",array2[7]);
    }//end while

}

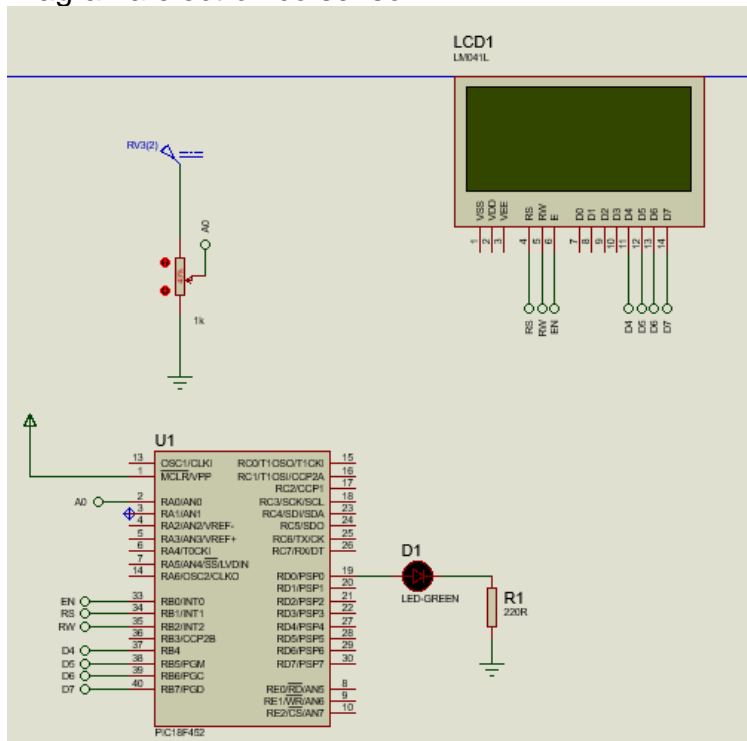
for(adc=0,analog=0;adc<7;adc++,analo
g++){
```

h) Explicación del código Valor Voltaje

- 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...).

i) Diagrama electrónico sensor LED



j) Código en CCS Sensor LED

```
//----- MAIN LIBRARY -----
#include <18F452.h>
//----- ADC CONFIGURATION -----
#device adc=10
//----- FUSES CONFIGURATION -----
#fuses
NOWDT,HS,PUT,NOPROTECT,NOBR
OWNOUT,NOLVP,NOCPD
#use delay(clock=4MHz)
//----- EXT LIBRARIES -----
```

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

