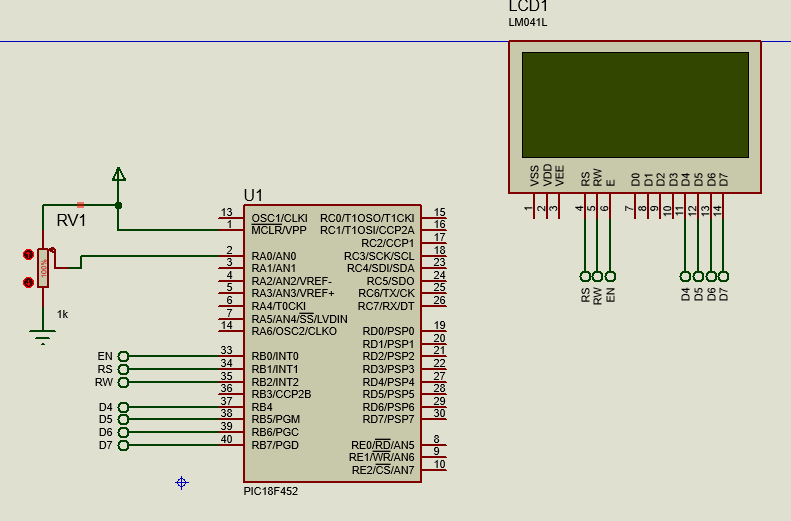
|  |  |  |
| --- | --- | --- |
|  | Dirección General de Educación Tecnológica IndustrialCentro de Bachillerato Tecnológico industrial y de servicios No. 168 **“Francisco I. Madero”**  **Carrera: Mecatrónica** |  |

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

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

Título: ADC Convertidor de Analógico a Digital

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



1. Código en CCS SIN LIBRERÍA

//----------- MAIN LIBRARY ----------

#include <18F452.h>

//-------- ADC CONFIGURATION --------

#device adc=10

//------- FUSES CONFIGURATION -------

#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,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

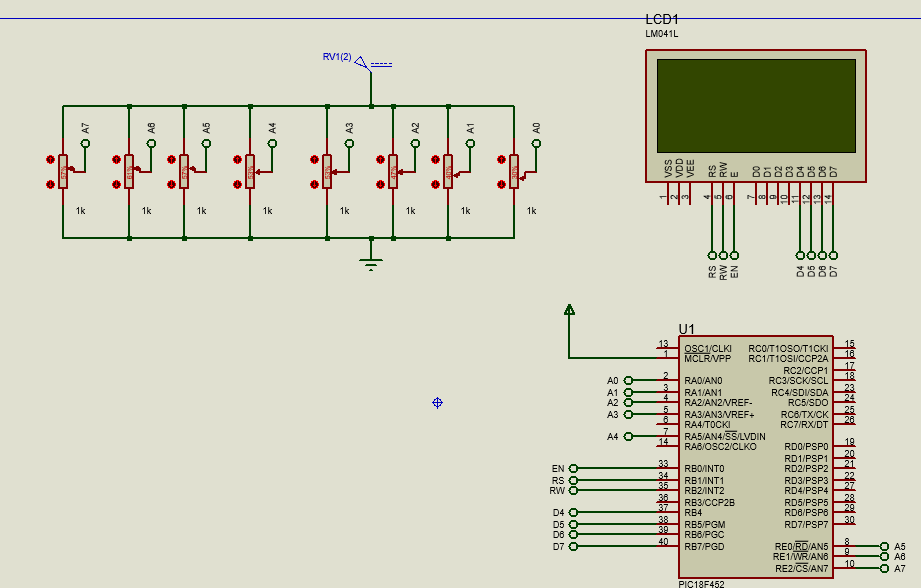
printf(lcd\_putc,"Volt= %01.2fV",var\_analog);

delay\_ms(100);

}//end while

}//end main

1. Explicación del código
2. Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
3. 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.
4. Se declara como entero la variable del adc (Var\_adc) y como flotante el voltaje o variable analógica (var\_analog)-
5. 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
6. 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).
7. 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
8. Diagrama Electronico Todos los canales Valor ADC/Valor Voltaje



1. Código en CCS Valor ADC

//----------- MAIN LIBRARY ----------

#include <18F452.h>

//-------- ADC CONFIGURATION --------

#device adc=10

//------- FUSES CONFIGURATION -------

#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,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

}//end main

1. Explicación del codigo Valor ADC
2. Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
3. 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.
4. 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.
5. 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.
6. 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
7. 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…).
8. Codigo en CCS Valor Voltaje

//----------- MAIN LIBRARY ----------

#include <18F452.h>

//-------- ADC CONFIGURATION --------

#device adc=10

//------- FUSES CONFIGURATION -------

#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,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){

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

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]);

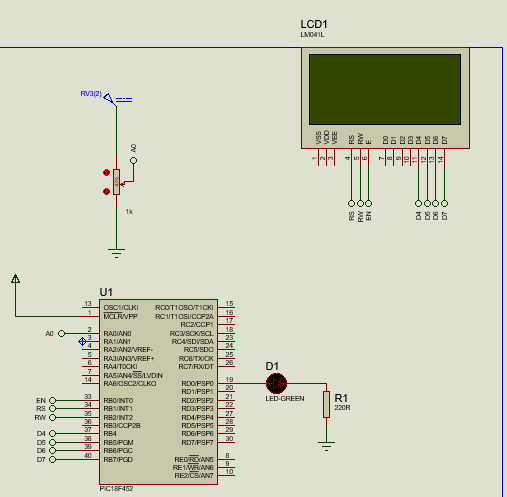
lcd\_gotoxy(26,2);

printf(lcd\_putc,"8=%01.2fV",array2[7]);

}//end while

}//end main

1. Explicación del código Valor Voltaje
2. Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
3. 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.
4. 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.
5. 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.
6. 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
7. 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…).
8. Diagrama electrónico sensor LED



1. Código en CCS Sensor LED

//----------- MAIN LIBRARY ----------

#include <18F452.h>

//-------- ADC CONFIGURATION --------

#device adc=10

//------- FUSES CONFIGURATION -------

#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,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--

void main(){

//Set Outputs

Tris\_D=0x00;

Port\_D=0x00;

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 Reyes");

//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);

printf(lcd\_putc,"2.0V>%01.2fV",var\_analog);

if(var\_analog>2.0){

Port\_D=0xFF;

lcd\_gotoxy(17,2);

printf(lcd\_putc,"LED ON ");

}

else{

Port\_D=0x00;

lcd\_gotoxy(17,2);

printf(lcd\_putc,"LED OFF");

}

}//end while

}//end main

1. Explicación del código Sensor LED
2. Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
3. 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.
4. 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.
5. Se imprime un mensaje que contiene “Orlando” en la primera fila y “Contreras Reyes” en la 2da fila, después entra el bucle infinito
6. 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
7. Si es mayor a 2.0 Encenderá un led e imprimirá LED ON. Y si no lo apagará e imprimirá LED OFF