# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**FACULTAD DE INGENIERIA EN**

**EN ELECTRICIDAD Y COMPUTACIÓN**

**FIEC**



**PROYECTO DE LABORATORIO DE MICROCONTROLADORES**

* **CONVERSION ANALOGICA – DIGITAL**
* **MEMORIZAR NUMEROS**

**Abraham Zari**

**Paralelo # 5**

**Ing. Carlos Valdivieso**

**Lunes, 23 de Agosto 2012**

**CONVERTIDOR**

1. **Enunciado del Proyecto.**

Este proyecto tiene como tema central el uso del Convertidor Analogico-Digital disponible en el PIC y a su vez hacer uso de la pantalla LCD y led para visualizar la lectura

1. **Diagrama de Bloques**

**PIC16F887**

**Pantalla LCD**

**Bloque de Visualización de LEDs**

**Lectura del Potenciom.**

**Oscilador Interno**

1. **Diagrama de Flujo.**



**Algoritmo**

1. Empezamos seteando los parámetros de configuración, como des habilitación del perro guardián, protección del código, entre otras configuraciones.
2. Definimos variables.
3. Ajustamos configuraciones del PIC, selección de puertos.
4. Empieza programa principal.
5. Presentamos menú principal y liego esperamos que el usuario seleccione la tecla
6. Si selecciona tecla 1 realiza la conversión
7. Si selecciona tecla2 realiza el juego, el mismo que consiste memorizar unos números presentados por pantalla.

Para detener la muestra de dichos números se solicita al usuario que presione la tecla3.

1. Una vez creado oculto dicho numeros, solicita al usuario que ingrese los numeros va a participar, si corresponde al almacenado en memoria, el usuario GANA.
2. **Listado del Programa Fuente**

**Programa Principal**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; NOMBRE:

; FECHA: 23/08/2012

; VERSION: 1.00

; PROGRAMADOR:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; DESCRIPCION:

;

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Header\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// conexiones del modulo LCD

sbit LCD\_RS at RB4\_bit;

sbit LCD\_EN at RB5\_bit;

sbit LCD\_D4 at RB0\_bit;

sbit LCD\_D5 at RB1\_bit;

sbit LCD\_D6 at RB2\_bit;

sbit LCD\_D7 at RB3\_bit;

sbit LCD\_RS\_Direction at TRISB4\_bit;

sbit LCD\_EN\_Direction at TRISB5\_bit;

sbit LCD\_D4\_Direction at TRISB0\_bit;

sbit LCD\_D5\_Direction at TRISB1\_bit;

sbit LCD\_D6\_Direction at TRISB2\_bit;

sbit LCD\_D7\_Direction at TRISB3\_bit;

// Final de las conexiones del módulo LCD

char keypadPort at PORTC; //conexion del teclado

char tabla\_ascii[]={'0','1','2','3','4','5','6','7','8','9'};

unsigned char ch;

unsigned int temp\_res;

char \*text2,\*text,\*text3,\*text6,\*text9,\*text10,\*text7;

long tlong;

int kp=0,bandera=0,band=0,cont=0,bien=0,avance=0,i=0;

int facil[]={1,9,8,6,4,5};

int fac[7],numero=0;

#define display7seg PORTA

#define display\_on PORTE

void main() {

INTCON = 0; // Todas las interrupciones deshabilitadas

ANSEL = 0x00; // Todos los pines son digitales

TRISA = 0x00;

PORTA=0;

PORTE=0;

TRISE=0b00000001;

ANSELH = 0; // Rest of pins is configured as digital

TRISB = 0x3F; // Port B pins RB7 and RB6 are configured as outputs

TRISD = 0; // All port D pins are configured as outputs

PORTD=0;

ADCON1 = 0x80; // Voltaje de referencia para la conversión A/D es VCC

Lcd\_Init(); // Inicialización del visualizador LCD

Lcd\_Cmd(\_LCD\_CURSOR\_OFF); // Comando LCD (apagar el cursor)

Lcd\_Cmd(\_LCD\_CLEAR); // Comando LCD (borrar el LCD)

Delay\_ms(300);

text = "LECTURA:";

text2 = "TECLA 1:LEER POTENC.";

text3 = "TECLA 2:MEMORIA";

text6 = "MEMORIA:#3 [198645] ";

text9 = "GANASTE!!!";

text10 = "PERDISTE!!!";

text7 = "INGRESE:";

Lcd\_Out(3,2,text2);

Lcd\_Out(4,2,text3);

display7seg=0b01110111;

display\_on=2;

Delay\_ms(400);

display\_on=0;

display7seg=0b01101101;

display\_on=4;

Delay\_ms(400);

display\_on=0;

while(1) {

kp=Keypad\_key\_click();

if(kp==1){

Lcd\_Cmd(\_LCD\_CLEAR);

do{

bandera=1;

temp\_res = ADC\_Read(5)>>2; // Result of A/D conversion is copied to temp\_res

Lcd\_Out(2,1,text);

tlong = (long)temp\_res \* 5000; // Convertir el resultado en milivoltios

tlong = tlong / 255; // 0..1023 -> 0-5000mV

ch = tlong / 1000; // Extraer voltios (miles de milivoltios)

Lcd\_Chr(2,9,48+ch); // Escribir resultado en formato ASCII

Lcd\_Chr\_CP('.');

ch = (tlong / 100) % 10; // Extraer centenas de milivoltios

Lcd\_Chr\_CP(48+ch); // Escribir resultado en formato ASCII

ch = (tlong / 10) % 10; // Extraer decenas de milivoltios

Lcd\_Chr\_CP(48+ch); // Escribir resultado en formato ASCII

ch = tlong % 10; // Extraer unidades de milivoltios

Lcd\_Chr\_CP(48+ch); // Escribir resultado en formato ASCII

Lcd\_Chr\_CP('V');

Delay\_ms(1);

PORTD = temp\_res; // 8 LSBs are moved to port D

}while(bandera==1);

}

if(kp==2){

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(2,1,text6);

do{

band= Keypad\_key\_click();

}while(band!=3);

if(band==3){

bandera=0;

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(3,2,text7);

do{

do{

bandera= Keypad\_key\_click();

}while(bandera==0);

fac[avance]=bandera;

numero=tabla\_ascii[bandera];

Lcd\_Chr(3,12+avance,numero);

avance++;

bandera=0;

}while(avance!=6);

for(cont=0;cont<6;cont++){

if(facil[cont]==fac[cont]){

bien++;

}

}

if(bien==6){

delay\_ms(1000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(3,4,text9);

}

else{

delay\_ms(1000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(3,4,text10);

}

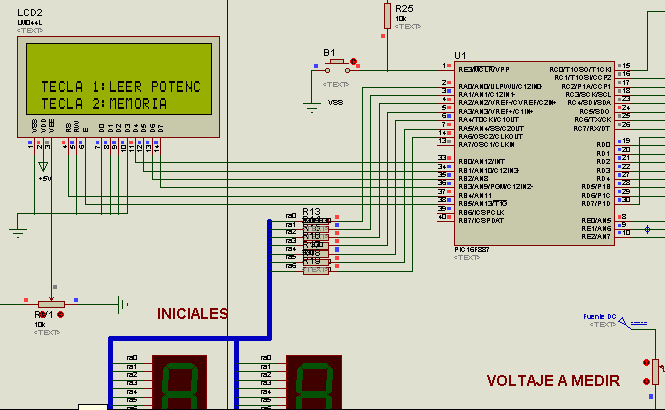
} // fin de (band==3)

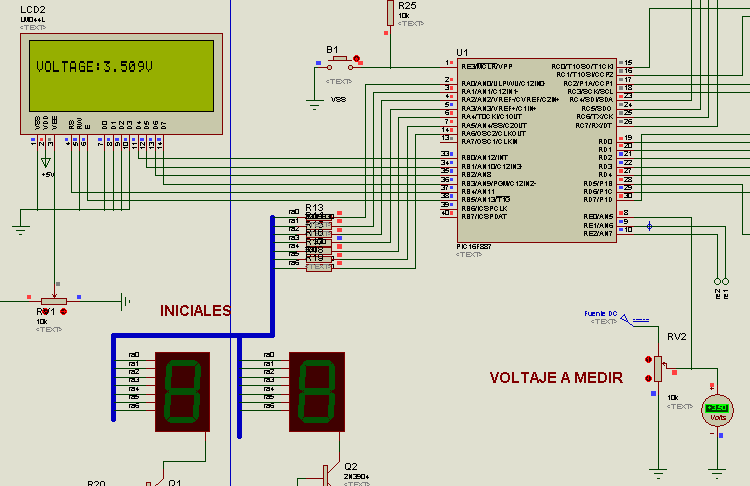
}

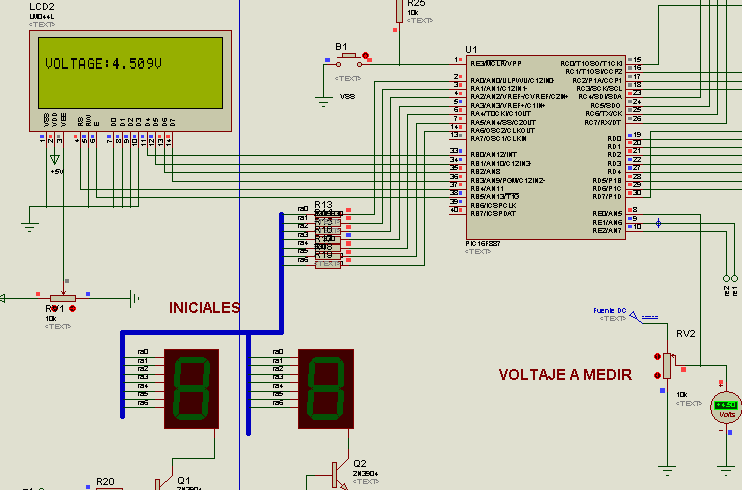
}

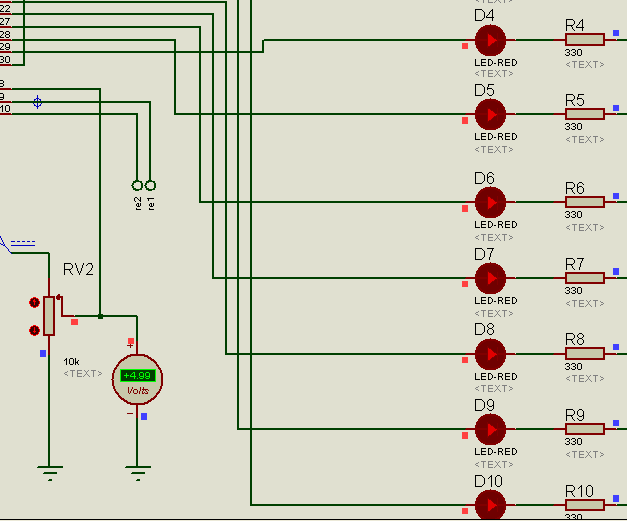
}

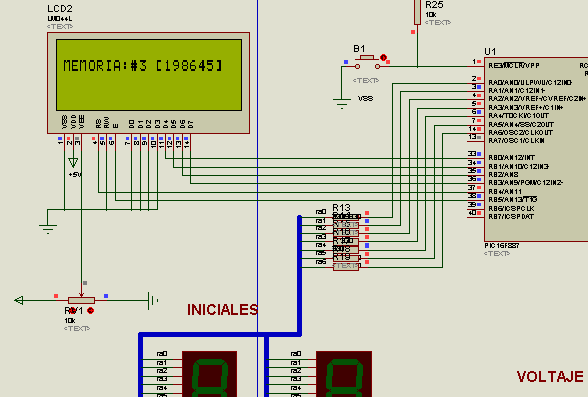
1. **Circuito Armado en Proteus**

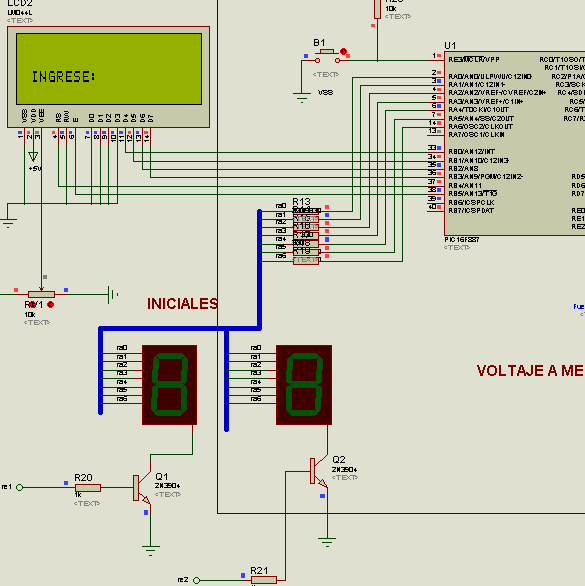
****

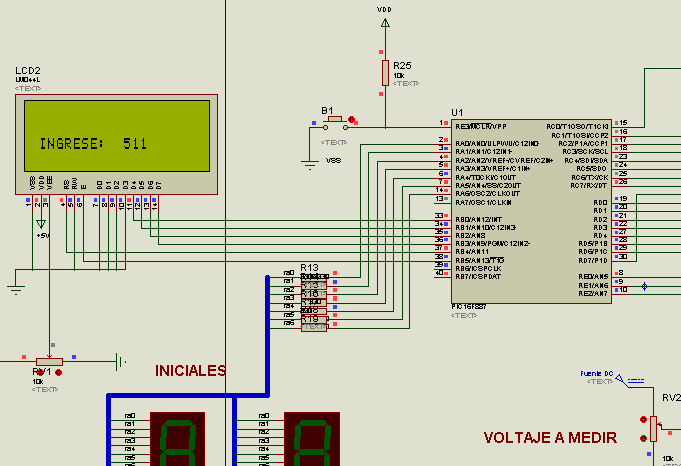
****

****

****

****

****

****

**Conclusiones**

* El uso adecuado de retardos nos permitirán implementar cualquier tipo de circuito, darle un toque elegante al programa, MikroC PRO dispone de una cierta cantidad de librerías que nos facilitan la programación del PIC, aquellas librerías son muy fáciles de utilizar.
* Una vez culminada el proyecto pudimos darnos cuenta que el PIC 16F887 es un micro controlador muy empleado puesto que tiene mayor cantidad de recursos que aquellos de menor gama, y con el conocimiento necesario de cómo funcionan sus registros podemos hacer cualquier tipo de programación que tengamos en mente.

**Recomendaciones.**

* Para poder realizar la práctica y sacarle el mayor provecho es de vital importancia que el estudiante tenga el conocimiento necesario del comportamiento de los lazos anidados, de esta manera no tendrá ningún inconveniente si es instructor le pide que realice cambios en el programa para aumentar el retardo de tiempo o hacer alguna variante en ello.
* De igual forma se pide al estudiante que tenga mucho cuidado al momento de programar el micro controlador (PIC), para ello debe recordar las opciones que debe habilitar o deshabilitar para la ejecución del programa en un momento dado con el propósito de que no se vayan a presentar problemas en el proyecto y podemos llegar a una feliz culminación.
* Se recomienda al estudiante llevar adecuadamente la materia, esto es, revisar el material proporcionado en el aula de clases y revisar cada uno de los programas realizar en el laboratorio para obtener conocimientos sólidos que nos permitirán desenvolvemos de mejor manera en el laboratorio y cuando abandonemos las aulas de clases en nuestra vida laboral.