# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN**

**EN ELECTRICIDAD Y COMPUTACIÓN**

**FIEC**

**LABORATORIO DE MICROCONTROLADORES**



**PROYECTO # 2**

***Lectura de valor analógico de un potenciómetro y Tema Libre:***

***Mini Calculadora***

**NOMBRE**

***CARLOS CELY LEON.***

**PARALELO**

***8***

**GRUPO**

***5***

**FECHA DE ENTREGA**

***24/01/12***

**2011 – 2° TÉRMINO**

**Especificaciones técnicas del proyecto:**

El siguiente proyecto es desarrollado en lenguaje C y ejecuta dos programas:

* Programa #1 que consiste en la *lectura de los valores analógicos* de un potenciómetro colocado en la bornera de entrada analógica, presentando las variaciones de valores (en binario) en 8 LEDs y en dos DISPLAYs de 7 segmentos (en dos dígitos decimales) disponibles.
* Programa #2 que consiste en una *mini calculadora* haciendo uso de los recursos de la tarjeta prototipo que consta de cuatro botoneras, 8 LEDs y dos DISPLAYs de 7 segmentos.

**Descripción del Proyecto:**

El PROGRAMA1 consiste en hacer uso del módulo ADC que posee el microcontrolador. Lee el valor analógico del potenciómetro conectado en el PORTA y muestra su equivalente digital DECIMAL: unidades y decenas en los dos DISPLAYs conectados en los puertos C y D respectivamente; y su valor BINARIO en los 8 LEDs conectados al PORTB.

El PROGRAMA2 consiste en la implementación de una mini calculadora haciendo uso de 4 botoneras y mostrando el valor en los dos DISPLAYs de 7 segmentos. Si se presiona la BOTONERA1 incrementa el numero 1 con que se va a trabajar. Si se presiona la BOTONERA2 se incrementa el número 2 con que se va a trabajar. Ya seleccionados los números se presiona la BOTONERA3 para salir del ajuste y empezar con los cálculos. Los números son mostrados en los DISPLAYs. Ahora estaremos esperando la operación a realizar. Con la BOTONERA1 se mostrara la multiplicación, con la BOTONERA2 la suma y con la BOTONERA 3 la resta, con la BOTONERA4 regresamos al ajuste de los números.

**Diagrama de Bloques:**



**Diagrama de Flujo del programa principal:**



**Descripción del Algoritmo:**

Se escriben los parámetros de configuración e inicializan las variables y los puertos a ser utilizados. Se muestran en los puertos C y D las iniciales del estudiante. Quedamos encerrador en un lazo infinito que pregunta por la BOTONERA1 o la BOTONERA2.

Si se presiona la BOTONERA1, se ejecuta el PROGRAMA1: Se configura el modulo ADC y se leen los Datos Analógicos desde el PIN0 del PORTA. Una vez obtenido el valor digital se lo divide para 10.23 para obtener un rango de 0 a 99. Seguido separamos el número en unidades y decenas y Finalmente mostramos el valor binario en el PORTB y el valor Digital Decimal en los PUERTOS C y D. Usando un arreglo de códigos de 7 segmentos para poder mostrar el número correctamente.

Si se presiona la BOTONERA2, se ejecuta el PROGRAMA2: Se inicializan las variables y puertos a usar; y quedamos dentro de un lazo infinito también, en que se espera por la presión de alguna de las 4 botoneras a usar. Si presiona la BOTONERA1 se incrementa el valor del numero1 de uno en 1 y se muestra su valor en los DISPLAYs.

Si se presiona la BOTONERA2 se incrementa el valor del número 2 de uno en 1 y se muestra su valor en los DISPLAYS. Si se presiona la BOTONERA3 salimos de modo de ajuste para entrar al modo de cálculo. En este modo si se presiona la BOTONERA1 se muestra la multiplicación del numero1 y el numero2 en los DISPLAYs.

Si se presiona la BOTONERA2 se muestra la suma de los números en los DISPLAYs.

Si se presiona la BOTONERA3 se muestra la División de los números en los DISPLAYs en los PORTC y PORTD. Finalmente si se presiona la BOTONERA4 salimos del modo de cálculo para regresar al modo de ajuste en el que nuevamente se espera el ajuste de los números para realizar los cálculos y así infinitamente.

**Programa Fuente:**

int dec,uni,decimal,dato,aux;

int const disp[10] = {0x3F, 0x06, 0x5B, 0x4F, //Se declaran como constantes

 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F}; //Código Hexa para cada número

int const id[27] = {0X77,0X7C,0X39,0X5E,0x79,0x71,0x3d,0x76,0x06,0x1e,0x76,0x38,0x4F,0x54,0x55,0x5c,0x73,0x67,0x50,0x6d,0x70,0x1c,0x3e,0x4f,0x76,0x6e,0x5b};

 // A , B , C , D , E , F , G , H , I , J , K , L , M , N , Ñ , O , P , Q , R , S , T , U , V , W , X , Y , Z

 // 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 ,12 ,13 ,14 ,15 ,16 ,17 ,18 ,19 ,20 ,21 ,22 ,23 ,24 ,25 ,26

int espejo(int temp1) //Función Para invertir Bits de una variable

{ int temp2=0;

if ((temp1&0x01)==0x01)

temp2=temp2+128;

if ((temp1&0x02)==0x02)

temp2=temp2+64;

if ((temp1&0x04)==0x04)

temp2=temp2+32;

if ((temp1&0x08)==0x08)

temp2=temp2+16;

if ((temp1&0x10)==0x10)

temp2=temp2+8;

if ((temp1&0x20)==0x20)

temp2=temp2+4;

if ((temp1&0x40)==0x40)

temp2=temp2+2;

if ((temp1&0x80)==0x80)

temp2=temp2+1;

return temp2;

}

void p1()

{

ADC\_Init(); //inicializo el ADC

while(1)

{

dato=ADC\_Read(0); //entrada analógica 0

aux=(dato/10.23); //división perfecta para marcar rango

if(aux<=99)

{

dec=aux/10; //división para extraer decimal del dato anterior

uni=aux%10; //modulo (residuo de la división) para extraer unidades

}

if(aux==100)

{

dec=0;

uni=0;

}

PORTB=espejo(aux); // muestro el valor binario en el PORTB

PORTC=disp[uni]; // muestro valor unidades en el PORTC.

PORTD=disp[dec]; // muestro valor decenas en el PORTD

}

}

void p2()

{

int contu=0, contd=0, contt=0, temp1=0, temp2=0; //declaracion de variables

do{ //se enceran displays

PORTC=disp[0];

PORTD=disp[0];

}while(PORTA.F2); // se valida que suelte la resistencia

Delay\_ms(300);

while(1)

{

PORTC=disp[0];

PORTD=disp[0];

while(!PORTA.F3)

{

 if(PORTA.F1)

 {

 if(contu>9)

 contu=0;

 contu++;

 PORTC=disp[contu];

 Delay\_ms(300);

 }

 if(PORTA.F2)

 {

 if(contd>9)

 contd=0;

 contd++;

 PORTD=disp[contd];

 Delay\_ms(300);

 }

 temp1=contu;

 temp2=contd;

 }

Delay\_ms(500);

while(!PORTA.F4)

{

if(PORTA.F1)

 {

 contt=temp1\*temp2;

 contd=contt/10;

 contu=contt%10;

 PORTC=disp[contu];

 PORTD=disp[contd];

 Delay\_ms(1000);

 }

 if(PORTA.F2)

 {

 contt=temp1+temp2;

 contd=contt/10;

 contu=contt%10;

 PORTC=disp[contu];

 PORTD=disp[contd];

 Delay\_ms(1000);

 }

 if(PORTA.F3)

 {

 if(temp1>temp2)

 contt=temp1-temp2;

 if(temp2>temp1)

 contt=temp2-temp1;

 if(temp1==temp2)

 contt=0;

 contd=contt/10;

 contu=contt%10;

 PORTC=disp[contu];

 PORTD=disp[contd];

 Delay\_ms(1000);

 }

}

}

}

void main()

{

ANSEL=0x01;

ANSELH = 0; // Configure other AN pins as digital I/O

C1ON\_bit = 0; // Disable comparators

C2ON\_bit = 0;

TRISA = 0xFF; // PORTA como entrada

TRISC = 0; // PORTC como salida

TRISB = 0; // PORTB como salida

TRISD = 0; // PORTD como salida

PORTB=0; // inicializado en 0 el PORTB

while(1)

{

PORTC=id[2]; // iniciales

PORTD=id[2]; // iniciales

if(RA1\_bit=1) //prueba el bit 1 del PORTA

p1();

if(PORTA.F2=1) //prueba el bit 2 del PORTA

p2();

}

}

**Circuito armado en Proteus:**

****

Aplastando la botonera dos para configurar el primer digito.



Aplastando la botonera uno para configurar el segundo digito



Aplastando la botonera tres para correr el programa y luego presionando la botonera uno realiza la operación de multiplicacion.



Presionando la botonera dos realiza la operación de suma



Presionando la botonera 3 realiza la operación de resta



Presionando la botonera cuatro realiza un reset.



**Conclusiones:**

* Como conclusión se puede acotar que los módulos que tiene el PIC 16F887 como son el USART y I2C que sirven como comunicación entre el computador y el PIC o entre varios dispositivos cercanos que estén conectados al PIC ya sea como transmisor de datos como receptor de datos, así mismo la comunicación USART nos permite el enlace entre la PC con el PIC o entre PIC, esta comunicación es asincrónica sin depender de una señal de reloj, usando como sincronizador bit que determina cuando comienza a leer y cuando termina ya sea la transmisión o recepción de los datos.
* Los displays LCD son dispositivos muy útiles en situaciones en las que se desean presentar menús o en las cuales se requiere de un nivel elevado de interacción con el usuario. Así mismo se puede decir que son una gran alternativa en sistemas de mediciones múltiples y simultáneas donde es necesario visualizar los cálculos efectuados en donde el uso de displays de 7 segmentos fuera ineficiente.
* Los displays LCD poseen un controlador el mismo que les permite comprender los comandos provenientes del microcontrolador. Entre los pines importantes de este dispositivo se encuentran los siguientes: una entrada de datos de 8 bits(8 pines), un bit selector de modo: lectura o escritura(1 pin), un bit selector de dato o instrucción(1 pin), un bit habilitador(1 pin), +Vcc (1 pin) y Gnd (1 pin).

**Recomendaciones:**

* Se recomienda leer las funciones que involucran el display LCD de tal forma que se puedan aprovechar al máximo las prestaciones de este dispositivo, entrar en familia con el uso de las mismas, de tal manera, poder utilizar de mejor manera y de formas más eficiente los recursos que nos ofrece este software.
* Antes de realizar cualquier proyecto es necesario tener a la mano todos los materiales que vayamos a necesitar, si es posible tener un reemplazo de los mismos en caso de que alguno se nos dañe o nos falle y por supuesto verificar el buen estado de los mismos antes de conectar en el circuito y montar la placa, de esta manera evitamos tener malos resultados y aseguramos asi mismo el correcto funcionamiento de nuestro circuito.
* Como recomendación final, se puede decir que al terminar de soldar la placa de nuestro circuito, se debería medir con un multímetro la continuidad de la placa, y probar también si no existe algún corto circuito, ya que se puede dar el caso de que al soldar, se haya producido la unión de las pistas y esté haciendo corto circuito. Así mismo se recomienda también limpiar la placa con un cepillo y alcohol o diluyente, y de esta forma extraer algunos residuos que podrían causar conflictos en el correcto funcionamiento del circuito.