**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y**

**COMPUTACIÓN**

**LABORATORIO DE MICROCONTROLADORES**

**PROYECTO # 2**

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

**Libera Al Gusanito**

**.**

**Nombre:**

Alfredo Sotomayor

**Profesor:**

Dr. Msc. Carlos valdiviezo

**Paralelo: 8**

**Grupo # 5**

**Fecha de Presentación:**

25 de Enero del 2012

**2012 – 2013**

**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 un *temporizador programable en segundos* 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 liberar a un gusanito atrapado. Presionando la BOTONERA1 se genera el numero aleatorio entre el 1 y el 3, luego se espera que el usuario ingrese el numero correspondiente mediante las BOTONERAS 2, 3 y 4 que equivalen al 1, 2 ,3 respectivamente para asi liberarlo. Se tienen 3 intentos, si se acierta dentro de este intervalo, el gusanito escapa por el camino de LEDs, caso contrario queda atrapado para siempre.

**Diagrama de Bloques:**



**Diagrama de Flujo del programa principal:**



**Descripción del Algoritmo**

Se escriben los parámetros de configuración y 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 numero 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 numero 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 la BOTONERA1 para generar el numero aleatorio entre 1 y 3. Generado el numero aleatorio quedamos en otro lazo infinito por la espera de algunas de las otras BOTONERAS, 2, 3, 4. Si se presiona la BOTONERA correcta, Se muestra un pequeño movimiento del gusanito buscando la salida en el DISPLAY y saliendo por el camino de LEDs en el PORTB y regresando al Main principal. Caso contrario se muestra el intento fallido del gusanito dando una vuelta en el DISPLAY y mostrando en numero de intentos que va en el segundo DISPLAY. Esto sucede 3 veces de no adivinar y finalmente queda encerrado el gusanito dando vueltas en el DISPLAY.

**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}; //Codigo Hexa para cada nuemro

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 const snk[7] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40};

int espejo(int temp1) //Funcion 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 analogica 0

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

if(aux<=99)

{

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

uni=aux%10; //modulo (residuo de la division) 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 unidaades en el PORTC.

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

}

}

void p2()

{

int alea=0, num=0, cont=0; //declaracion de variables

do{ //se enceran displays

PORTC=disp[0];

PORTD=disp[0];

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

while(1)

{

while(!PORTA.F1)

{

alea++;

if (alea==4)

alea=0;

}

if(alea==0)

alea=1;

PORTB=alea;

Delay\_ms(500);

while(1)

{

if(PORTA.F2) // si presiona la botonera PORTA.pin1

{num=1;

Delay\_ms(300);

break;

}

if(PORTA.F3)

{

num=2;

Delay\_ms(300);

break;

}

if(PORTA.F4)

{

num=3;

Delay\_ms(300);

break;

}

}

if(num==alea)

{

PORTD=snk[6];

Delay\_ms(300);

PORTD=snk[1];

Delay\_ms(300);

PORTD=snk[0];

Delay\_ms(300);

PORTD=snk[5];

Delay\_ms(300);

PORTB=1;

Delay\_ms(300);

PORTB=2;

Delay\_ms(300);

PORTB=4;

Delay\_ms(300);

PORTB=8;

Delay\_ms(300);

PORTB=16;

Delay\_ms(300);

PORTB=32;

Delay\_ms(300);

PORTB=64;

Delay\_ms(300);

PORTB=128;

Delay\_ms(300);

PORTB=0;

return;

}

PORTC=disp[1];

PORTD=snk[6];

Delay\_ms(500);

PORTD=snk[1];

Delay\_ms(500);

PORTD=snk[0];

Delay\_ms(500);

PORTD=snk[5];

Delay\_ms(500);

PORTD=snk[4];

Delay\_ms(500);

PORTD=snk[3];

Delay\_ms(500);

PORTD=snk[2];

Delay\_ms(500);

PORTD=snk[6];

while(1)

{

if(PORTA.F2) // si presiona la botonera PORTA.pin1

{num=1;

Delay\_ms(300);

break;

}

if(PORTA.F3)

{

num=2;

Delay\_ms(300);

break;

}

if(PORTA.F4)

{

num=3;

Delay\_ms(300);

break;

}

}

if(num==alea)

{

PORTD=snk[6];

Delay\_ms(300);

PORTD=snk[1];

Delay\_ms(300);

PORTD=snk[0];

Delay\_ms(300);

PORTD=snk[5];

Delay\_ms(300);

PORTB=1;

Delay\_ms(300);

PORTB=2;

Delay\_ms(300);

PORTB=4;

Delay\_ms(300);

PORTB=8;

Delay\_ms(300);

PORTB=16;

Delay\_ms(300);

PORTB=32;

Delay\_ms(300);

PORTB=64;

Delay\_ms(300);

PORTB=128;

Delay\_ms(300);

PORTB=0;

return;

}

PORTC=disp[2];

PORTD=snk[6];

Delay\_ms(500);

PORTD=snk[1];

Delay\_ms(500);

PORTD=snk[0];

Delay\_ms(500);

PORTD=snk[5];

Delay\_ms(500);

PORTD=snk[4];

Delay\_ms(500);

PORTD=snk[3];

Delay\_ms(500);

PORTD=snk[2];

Delay\_ms(500);

PORTD=snk[6];

while(1)

{

if(PORTA.F2) // si presiona la botonera PORTA.pin1

{num=1;

Delay\_ms(300);

break;

}

if(PORTA.F3)

{

num=2;

Delay\_ms(300);

break;

}

if(PORTA.F4)

{

num=3;

Delay\_ms(300);

break;

}

}

if(num==alea)

{

PORTD=snk[6];

Delay\_ms(300);

PORTD=snk[1];

Delay\_ms(300);

PORTD=snk[0];

Delay\_ms(300);

PORTD=snk[5];

Delay\_ms(300);

PORTB=1;

Delay\_ms(300);

PORTB=2;

Delay\_ms(300);

PORTB=4;

Delay\_ms(300);

PORTB=8;

Delay\_ms(300);

PORTB=16;

Delay\_ms(300);

PORTB=32;

Delay\_ms(300);

PORTB=64;

Delay\_ms(300);

PORTB=128;

Delay\_ms(300);

PORTB=0;

return;

}

while(1)

{

PORTC=disp[3];

PORTD=snk[6];

Delay\_ms(500);

PORTD=snk[1];

Delay\_ms(500);

PORTD=snk[0];

Delay\_ms(500);

PORTD=snk[5];

Delay\_ms(500);

PORTD=snk[4];

Delay\_ms(500);

PORTD=snk[3];

Delay\_ms(500);

PORTD=snk[2];

Delay\_ms(500);

}

}

}

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[19]; // iniciales

PORTD=id[0];

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

p1();

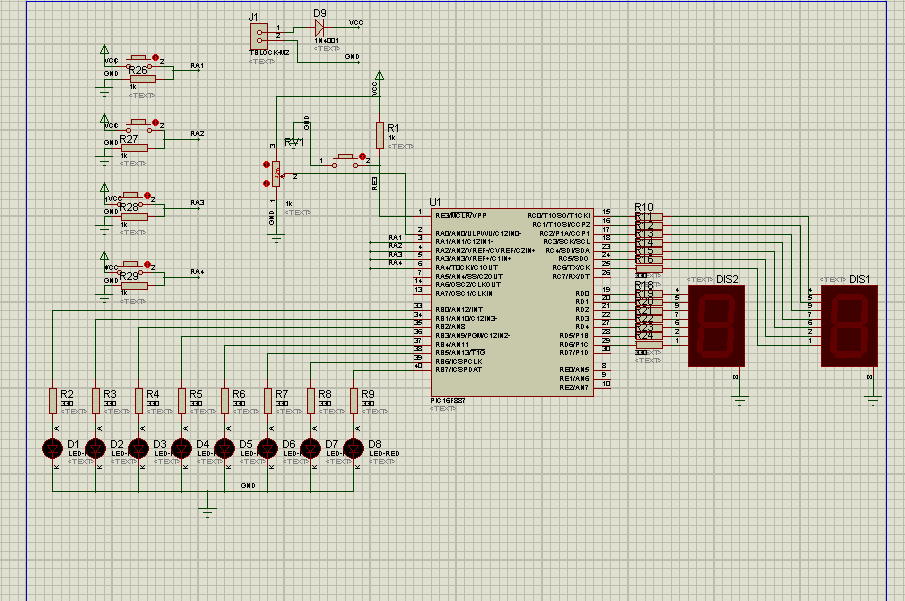
if(PORTA.F2=1) //prueba el bit 2 del PORTA

p2();

}

}

**CIRCUITO EN PROTEUS**



**Conclusiones:**

* En el proyecto 1 logre 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.
* Se escriben los parámetros de configuración y 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. El PROGRAMA1: Se configura el modulo ADC y se leen los Datos Analógicos desde el PIN0 del PORTA
* El PROGRAMA2 consiste en la liberar a un gusanito atrapado. Presionando la BOTONERA1 se genera el numero aleatorio entre el 1 y el 3, luego se espera que el usuario ingrese el numero correspondiente mediante las BOTONERAS 2, 3 y 4 que equivalen al 1, 2 ,3 respectivamente para asi liberarlo. Se tienen 3 intentos, si se acierta dentro de este intervalo, el gusanito escapa por el camino de LEDs, caso contrario queda atrapado para siempre

**Recomendaciones:**

* Al momento de realizar la placa del proyecto tener en cuenta la suelda que no este muy grande ya que esta puede hacer contacto con otro punto o nodo del circuito ya que esto provocaria que se realice un corto en la placa por lo cual no funcionaria.
* Al momento de colocar el pic hay que tener cuidado con la polariadad del mismo ya que podemos ponerlo del lado contrario y puede quemarse o simplemente dañar los pines que van colocados en la placa.
* Al momento de colocar los cables en las borneras tratar de que sean gruesos ya que si no logran hacer contacto con el cobre o simplemente quede flojo no dara el resultado esperado y esto puedo uno tomarlo como que el pic esta mal programado o algun elemento se encuentra dañado.