**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

**Nombre:**

Gerardo Chávez Andrade

**Profesor:**

Ing. Valdivieso Armendariz Carlos Enrique

**Paralelo: 8**

**Grupo # 2**

**Fecha de Presentación:**

25 de Enero del 2012

**2011 – 2012**

**1.- Enunciado del proyecto**

Este proyecto será en lenguaje C y deberá ejecutar dos ejercicios:

1. Ejercicio propuesto por el profesor 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.
2. Ejercicio de juego propuesto por el estudiante haciendo uso de los recursos de la tarjeta prototipo que consta de cuatro botoneras, 8 LEDs y dos displays de 7 segmentos.

Funcionamiento:

* Al encenderse el circuito deberán aparecer sus iniciales en los display de 7 segmentos.
* Al presionar un botón deberá iniciar el ejercicio 1
* Al presionar otro botón deberá iniciar el ejercicio 2

**2.- Diagrama de Bloque**

Ejercicio #1



Ejercicio #2



**3.- Diagrama de Flujo funcional del Programa principal y de las subrutinas**

Presentación Inicial



Ejercicio #1



Ejercicio #2



**4. Descripción del algoritmo o estrategia utilizada.**

Declaración

* Declaramos las variables que serán usadas
* Definimos las entradas como digitales y la entrada analógica y deshabilitamos los comparadores
* Activamos la entrada analógica que será utilizada para la conversión analógica - digital
* Se define el puerto A como entrada, el B y C como salidas

Presentación Inicial

* Al iniciar el programa se presentara en 2 display de 7 segmentos las iniciales G y C correspondientes a Gerardo Chávez.
* Preguntara constantemente por la botonera RA4 o RA3 los cuales indicaran si se ingresa al Ejercicio 1 o al Ejercicio 2.
* Si no se presiona ninguna botonera se mantendrá en la presentación inicial

Ejercicio #1

* Usamos la función ADC\_Read para adquirir la conversión analógica.
* Realizamos la conversión para que el máximo en los led sea 255 y en los display sea de 99.
* Preguntamos constantemente por el estado de RA4 ya que si es presionado y soltado durante el lazo infinito este saldría del ejercicio 1 y volverá al estado inicial del programa

Ejercicio #2

* Usamos la función ADC\_Read para adquirir la conversión analógica.
* Nos valemos del valor de conversión del potenciómetro para fijar la dificultad del juego implementado en el ejercicio 2.
* Se genera un número aleatorio el cual tendrá un código grafico que deberá coincidir con el código ingresado en las botoneras.
* Se presenta en los display y en los led el código grafico y el puntaje actual en los led.
* Se pide el código correspondiente al código grafico presentado y si el código ingresado es correcto ser incrementa la puntuación y se salta a la siguiente generación aleatoria de código grafico.
* En caso de que se ingrese un código incorrecto seguirá esperando durante un tiempo configurado según la dificultad incrementada por el potenciómetro en la entrada analógica.
* Cuando se agote el tiempo configurado se salta a la siguiente generación de código pero en este caso se decremento la puntuación.
* Cuando se llegue a los 16 o cero puntos se sale del Ejercicio 2 presentando previamente un parpadeo dependiendo de la puntuación.

**5.- Listado del programa fuente en lenguaje ensamblador con comentarios en las líneas de código que considere fundamentales**

int bin\_a\_7 (int valor);

void print\_display (int d2,int d1);

void print\_led (int led);

void parpadeo\_all (int d2, int d1, int valor);

void codigo (int array[],int num);

short game\_pad (void);

void print\_score(int score);

int control\_score(int score,int act);

void juego\_terminado(int estado);

unsigned int lectura=0,bin\_255=0,bin\_99=0;

const int parpadeo=500;

void main()

{

unsigned int uni=0,dec=0,temp1=0,temp2=0,temp4=0,i=0;

unsigned int x=0,y=0,z=0, cont=0,temp3=0;

unsigned short pad=00000000;

unsigned int n\_gen=0, nivel=0;

unsigned int score=3, salto=0,bandera=0,play=1;

unsigned int juego=0;//sera 1 para ejercicio 1 y 2 para ejercicio 2

ANSEL = 0x01; // Configure AN0 pin as analog

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

C1ON\_bit = 0; // Disable comparators

C2ON\_bit = 0;

TRISA = 0xFF; // PORTA is input

TRISB = 0;

TRISC = 0; // PORTC is output

TRISD = 0; // PORTD is output

PORTB=0; // inicializo los puertos en cero

PORTC=0;

PORTD=0;

while(1)//lazo infinito para repetir programacion general desde el inicio

{

uni=0; dec=0; temp1=0; temp2=0; temp4=0; i=0;

x=0; y=0; z=0; cont=0 ;temp3=0;

pad=00000000;

n\_gen=0; nivel=0;

score=12; salto=0; bandera=0; play=1;

while(temp1==0) //Mostrando iniciales de nombre

{

print\_display(125,57);

if (RA4\_bit==1)

{

delay\_ms(1);

if (RA4\_bit==0)

{

temp1=1;

delay\_ms(1);

juego=1;//entrar a ejercicio 1

}

}

else

{ delay\_ms(1);

if (RA3\_bit==1)

{

delay\_ms(1);

if (RA3\_bit==0)

{

temp1=1;

delay\_ms(1);

juego=2;//entrar a ejercicio 2

}

}

}

}

if(juego==1)//si se presiona RA4 entrar en Ejercico 1

{

while (temp2==0)

{

lectura = ADC\_Read(0); // Conversion analogica Digital (Ejercicio #1)

bin\_255 = lectura\*0.25;

bin\_99 = bin\_255\*0.389;

nivel = bin\_255+1;

uni = bin\_99%10;

dec = bin\_99/10;

print\_display(bin\_a\_7(dec),bin\_a\_7(uni));

print\_led(bin\_255);

if (RA4\_bit==1)

{

delay\_ms(1);

while(temp2==0)

{

if (RA4\_bit==0)//saldra del ejercico 1 si se presiona nuevamente

{

delay\_ms(1);

temp2=1;

delay\_ms(1);

}

}

}

else temp2=0;

}

juego=0;//enceramos nuevamente la variable

}

if(juego==2)

{

for(i=0;i<3;i++) //parpadeo de transicion entre Ejercicio #1 y #2

parpadeo\_all (64, 64, 0xff);

while (temp3==0)//BUCLE PARA CONTROLAR EL JUEGO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

lectura = ADC\_Read(0); // Conversion analogica Digital (Ejercicio #1)

bin\_255 = lectura\*0.25;

bin\_99 = bin\_255\*0.389;

nivel = bin\_255+1;

n\_gen=(rand()%10)+1; //Numero aleatorio

cont=((257-nivel)\*2)+200;

bandera=0;

switch (n\_gen)

{

case 1:

{

x=54;y=0;z=3;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 2:

{

x=48;y=48;z=5;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 3:

{

x=48;y=6;z=9;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 4:

{

x=6;y=48;z=6;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 5:

{

x=6;y=6;z=10;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 6:

{

x=0;y=54;z=12;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 7:

{

x=48;y=0;z=1;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 8:

{

x=6;y=0;z=2;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 9:

{

x=0;y=48;z=4;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

case 10:

{

x=0;y=6;z=8;

do

{

print\_display (x,y);//presento grafico de juego

print\_score(score);//presento puntuacion tipo barra

pad=game\_pad (); //adquiero datos del teclado

if((pad==z)&&(bandera==0))//pregunta si coincide grafico y codigo ingresado

{

score++;//aumentar puntuacion

bandera=1;

break;//si coincide salir y aumentar puntuacion

}

cont--;

if(cont==0)//si se acaba el tiempo salir y disminuir puntuacion

{score--;break;}//score=control\_score(score,0);

//parpadeo\_all (73, 73, 0xff);//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}while(1); //////////////////////////////////////////////////////////////////////////////////////////////////////

break;

}

default: break;

}

if(score<=0)//si el puntaje llega a cero entonces juego terminado

{print\_led(0);temp3=1;juego\_terminado(0);}

else

if(score>=16)

{print\_led(0);temp3=1;juego\_terminado(1);}

}

}

}

}

int bin\_a\_7 (int valor)

{

switch (valor)

{

case 0: return 63;

case 1: return 6;

case 2: return 91;

case 3: return 79;

case 4: return 102;

case 5: return 109;

case 6: return 125;

case 7: return 7;

case 8: return 127;

case 9: return 103;

default: break;

}

}

void print\_display (int d2,int d1)

{

PORTC=d1;

PORTD=d2;

}

void print\_led (int led)

{

PORTB=led;

}

void parpadeo\_all (int d2, int d1, int valor)

{

print\_display (d2,d1);

print\_led (valor);

delay\_ms(parpadeo);

print\_display (0,0);

print\_led (0);

delay\_ms(parpadeo);

}

short game\_pad (void)

{

unsigned short temp=000000;

temp.F3=RA4\_bit;

delay\_ms(1);

temp.F2=RA3\_bit;

delay\_ms(1);

temp.F1=RA2\_bit;

delay\_ms(1);

temp.F0=RA1\_bit;

delay\_ms(1);

return temp;

}

void print\_score(int score)

{

if(score==0)

print\_led(0);

if((score>0)&&(score<=2))

print\_led(128);

if((score>2)&&(score<=4))

print\_led(192);

if((score>4)&&(score<=6))

print\_led(224);

if((score>6)&&(score<=8))

print\_led(240);

if((score>8)&&(score<=10))

print\_led(248);

if((score>10)&&(score<=12))

print\_led(252);

if((score>12)&&(score<=14))

print\_led(254);

if(score==15)

print\_led(255);

}

int control\_score(int score,int act) //con 1 aumentar score y con 0 disminuir

{

delay\_ms(1);

if(score==0)

{

if(act==1)

return score++;

else

{

if(act==0)

return score;

}

}

if(score==15)

{

if(act==1)

return score;

else

{

if(act==0)return score--;

}

}

if(act==1)

return score++;

else

return score--;

}

void juego\_terminado(int estado)//estado en 1 entonces termine y gane sino termine y perdi

{

int g=0,i=0;

if (estado==1)//si se gano el juego

{

for(g=0;g<3;g++)

{

print\_display(63,63);

delay\_ms(350);

print\_display(64,64);

delay\_ms(350);

}

}

if (estado==0)//si se gano el juego

{

for(g=0;g<3;g++)

{

print\_display(84,84);

delay\_ms(350);

print\_display(8,8);

delay\_ms(350);

}

}

for(i=0;i<3;i++)

{

print\_display(125,63);

delay\_ms(500);

print\_display(0,0);

delay\_ms(500);

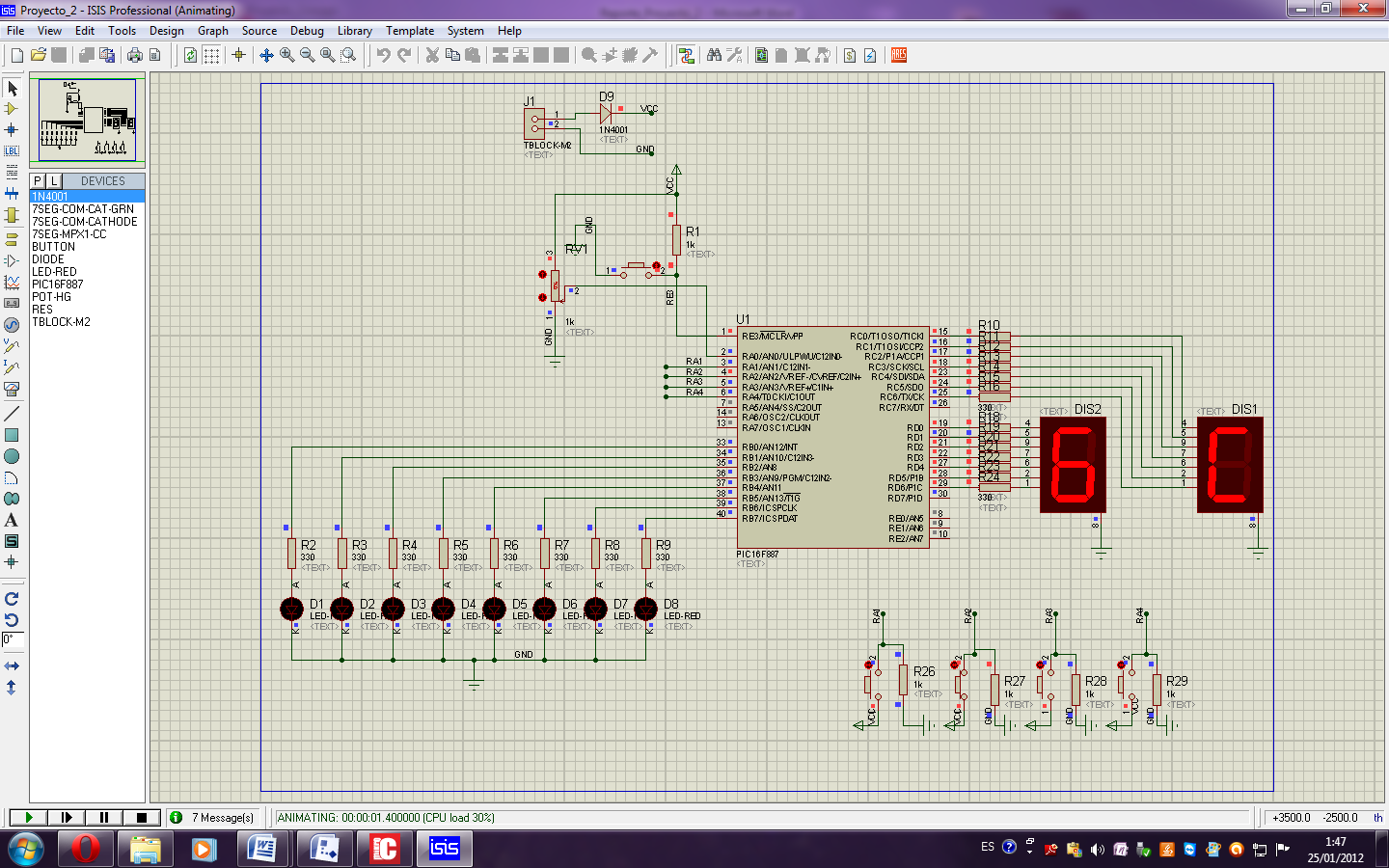
}

return;

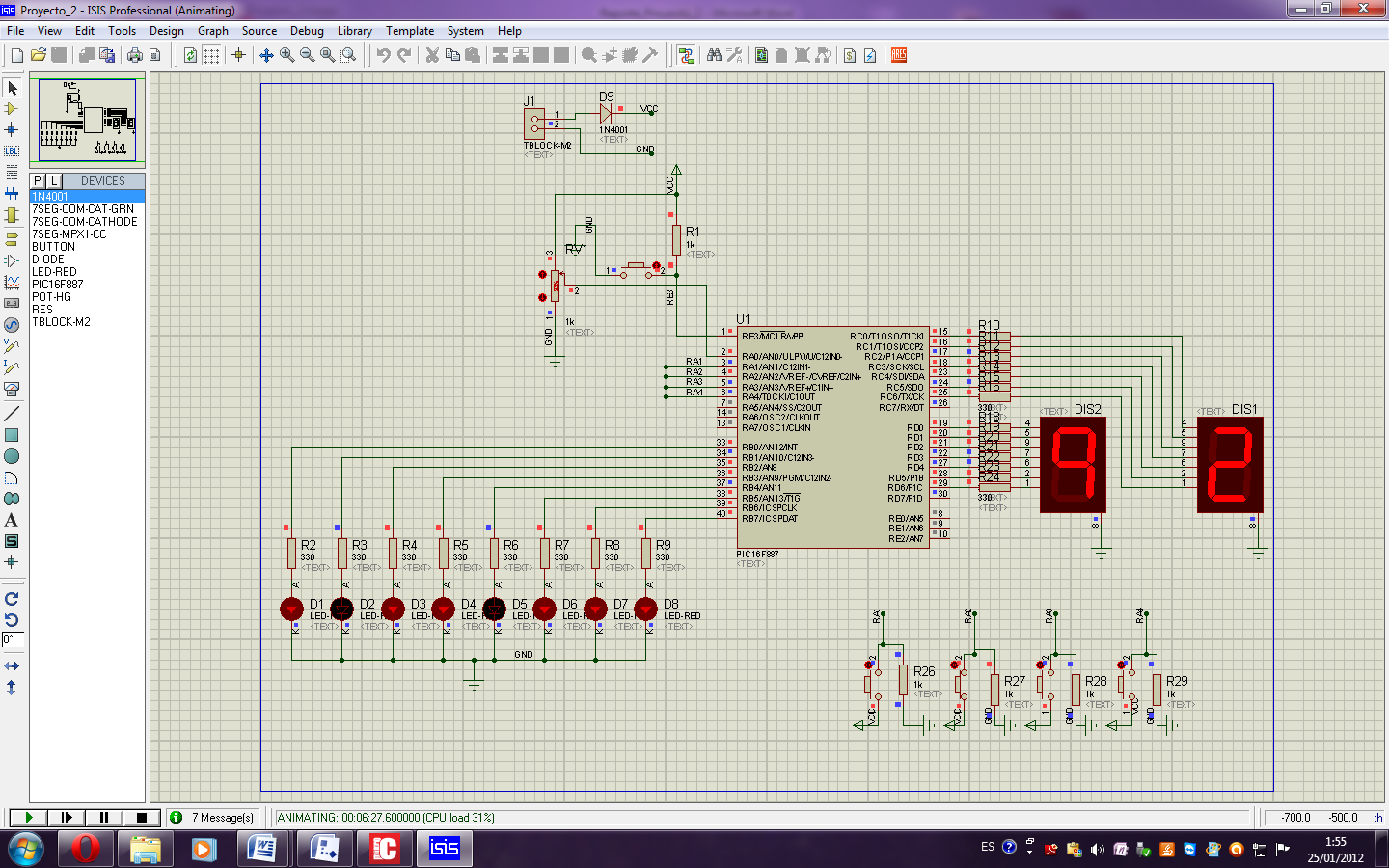
}

**6.- Copia impresa del circuito armado en PROTEUS para la simulación en el momento de su ejecución**

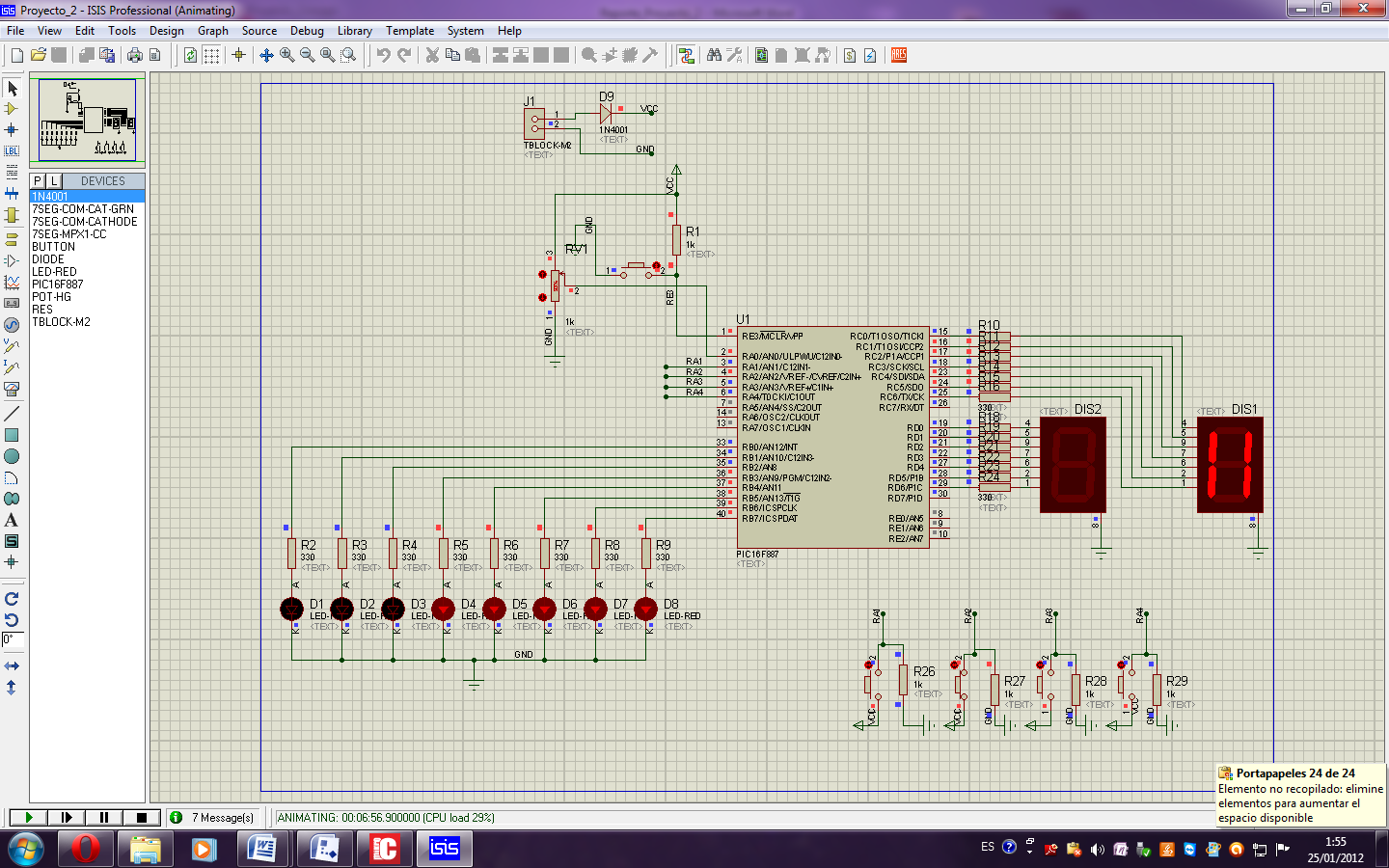
Presentación Inicial

****

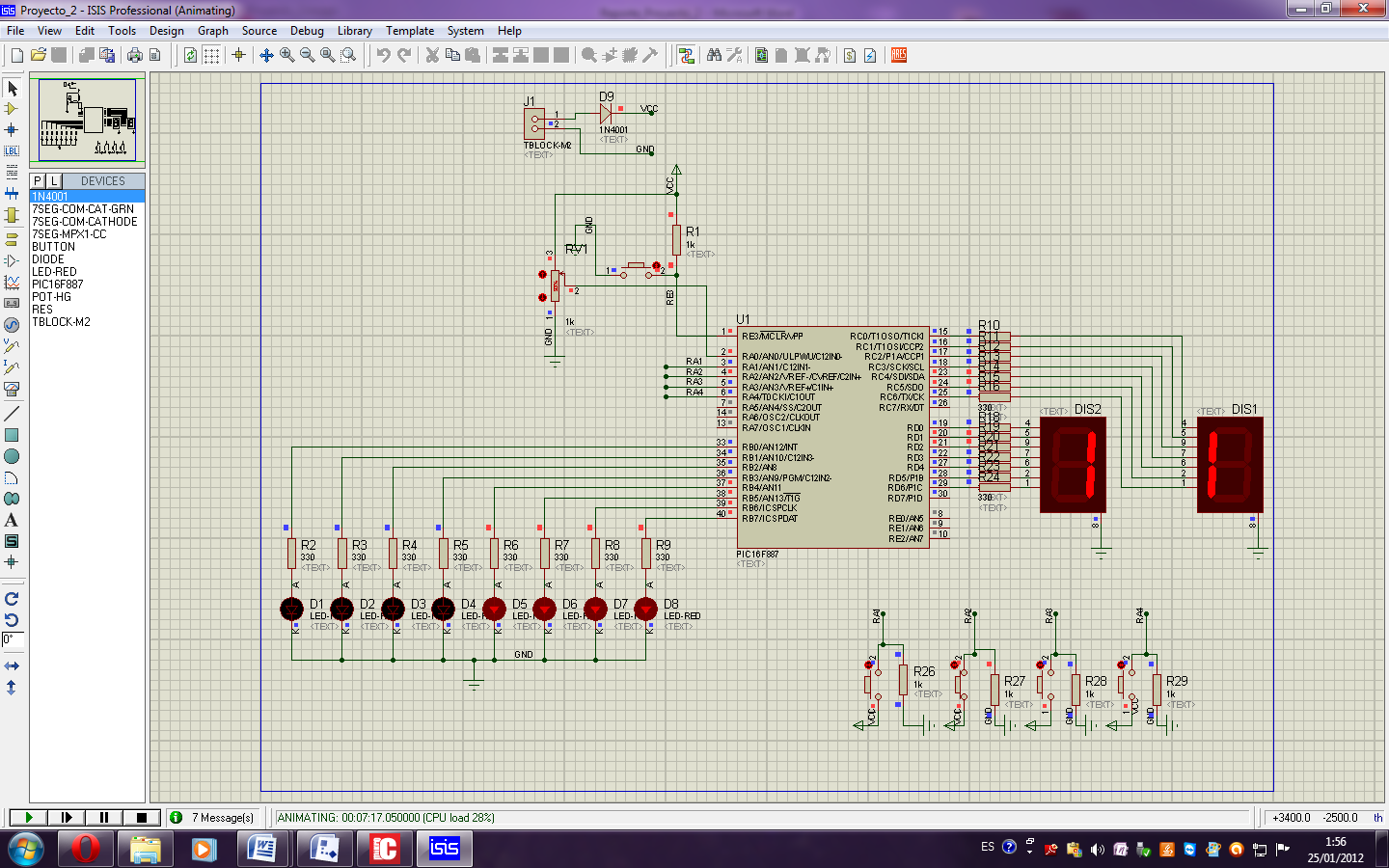
Ejercicio #1 (Conversión Analógica - Digital)

****

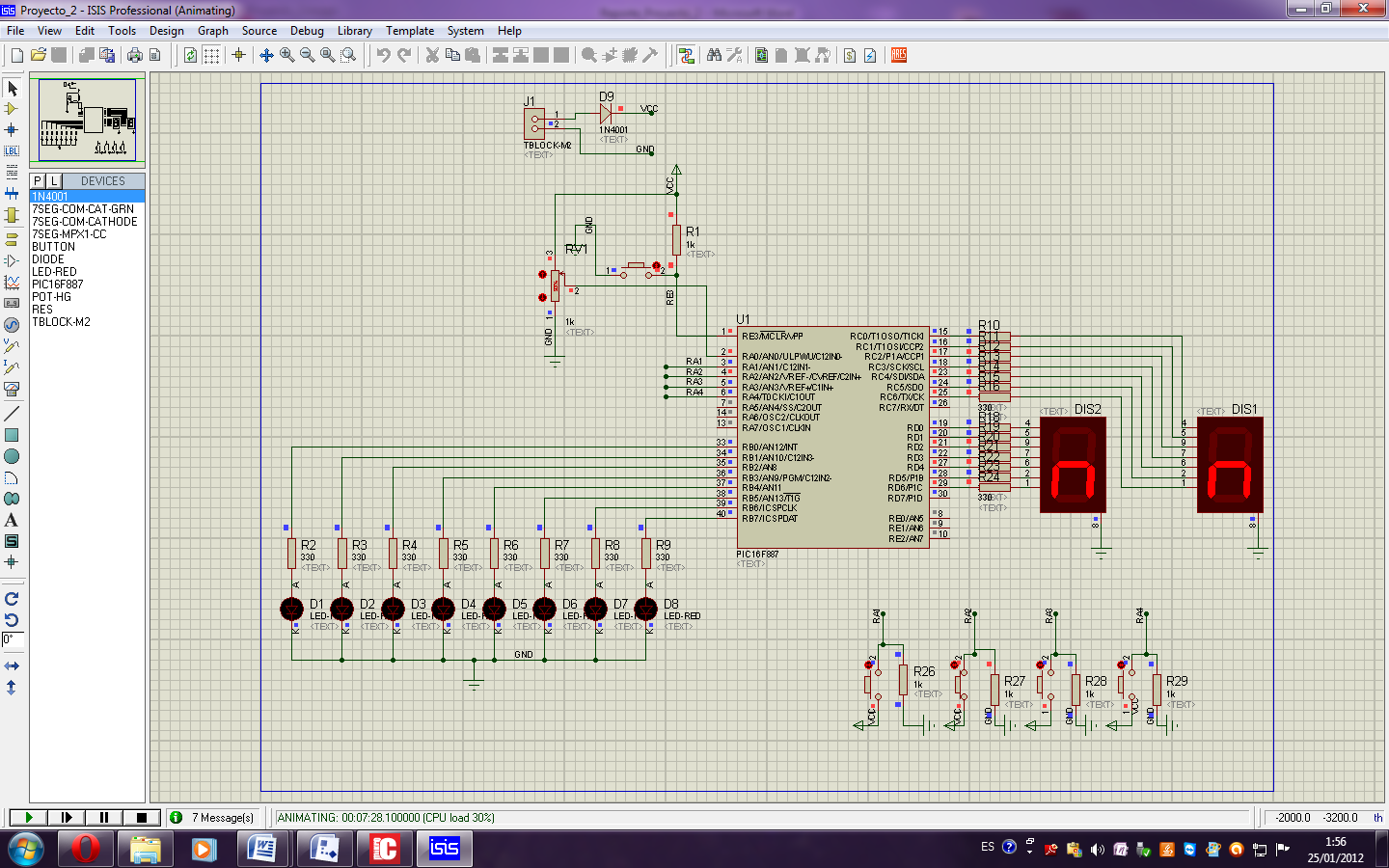
Ejercicio #2 (Juego)

****

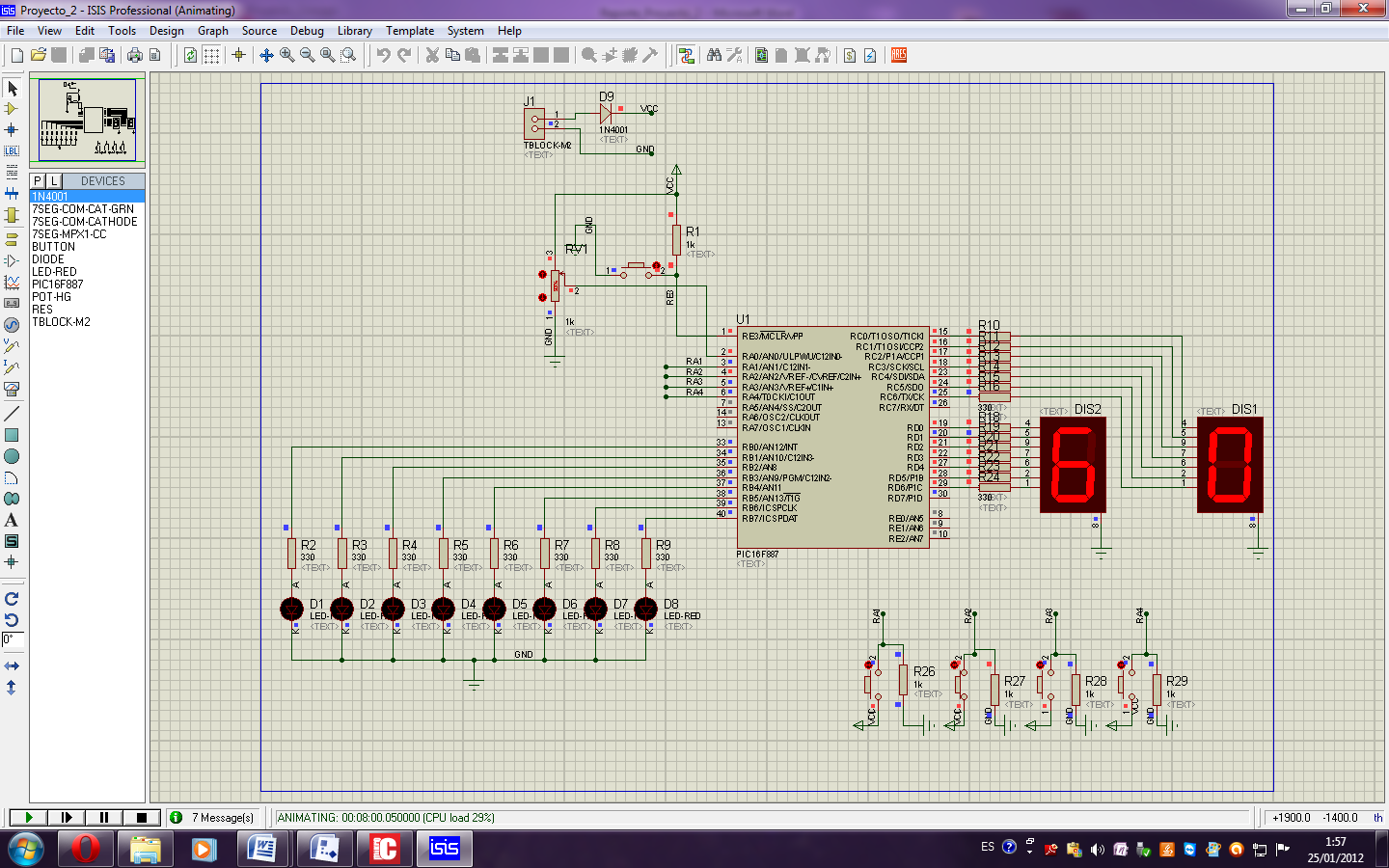
Decremento de puntuación

****

Parpadeo de juego perdido

****

Iniciales Game Over de juego terminado

****

**Conclusiones**

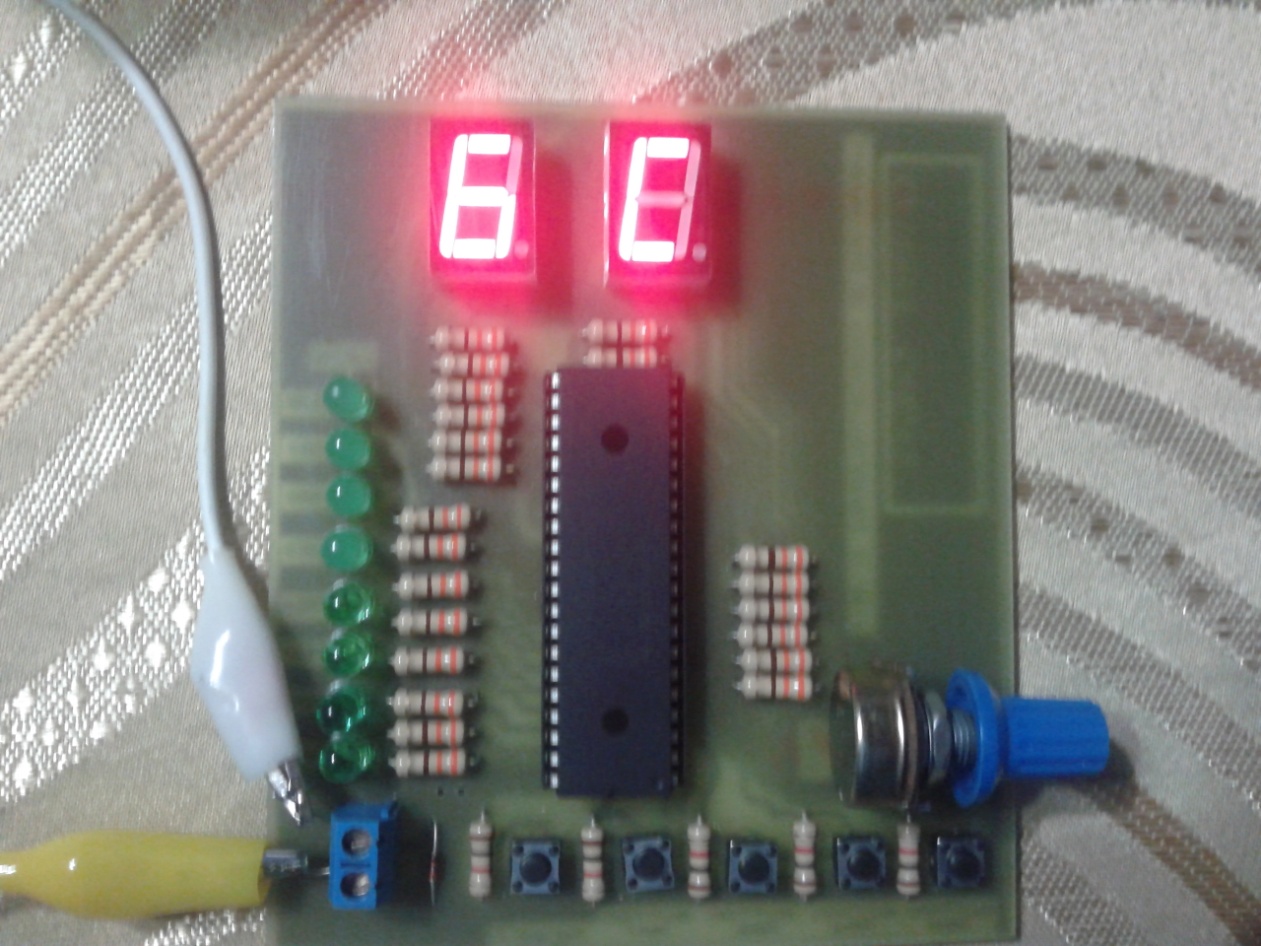
* Se pudo ver que en comparación con el lenguaje de programación de bajo nivel llamado Assembler, la codificación necesaria en el lenguaje de programación para micro controladores basado en C, es mucho más simple otorgándole al usuario una herramienta que realmente simplificara los procesos necesarios para implementar una conversión analógica digital.
* Gracias a las facilidades de programación y de abstracción proporcionadas por el lenguaje C de programación, el implementar aplicaciones comúnmente usadas o incluso de gran utilidad como lo es el convertir un fenómeno físico en una señal digital o realizar comunicaciones seriales full duplex resulta relativamente sencillo.
* Si en algún momento se necesita optimizar recursos y maximizar el rendimiento de un micro controlador lo cual resulta imperativo al momento de implementar sistemas a gran escala o que requieran controlar un gran número de señales, se necesita usar técnicas de multiplicación particularmente para los display lo cual incrementara en gran manera las opciones para controlar un mayor número de señales ya que quedaran mas pines que pueden ser orientados a adquisición o presentación de datos.

**Recomendaciones**

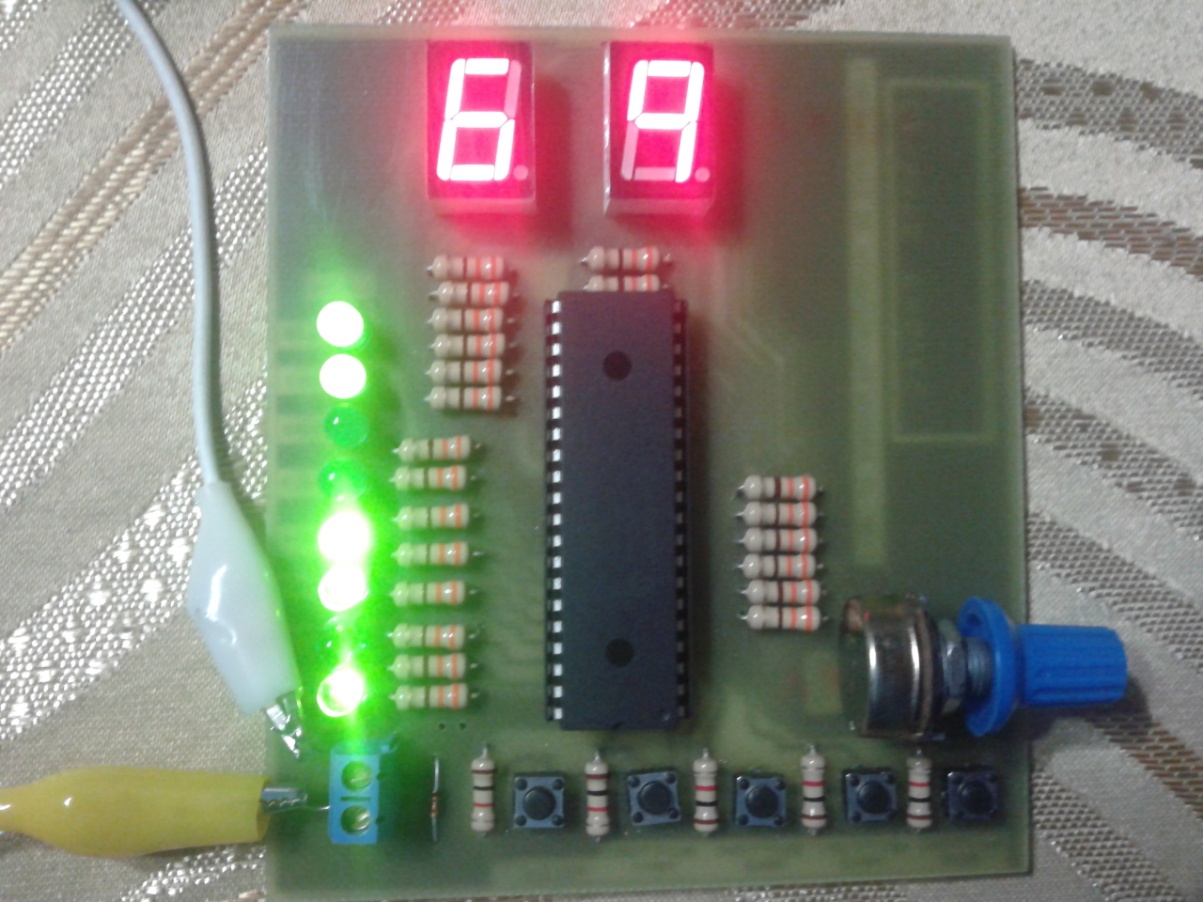
* Es recomendable siempre realizar una copia de respaldo del último programa que se ejecutó, antes de modificar el programa, ya que al momento de realizar modificaciones de podría alterar de manera indebida las líneas de comando arruinando por completo el programa creado y esto nos retrasaría mucho tiempo al momento de laborar el informe.
* Se recomienda que al momento de implementar un convertidor analógico – digital nos aseguremos de cuál es la resolución deseada en la conversión digital resultante ya que de lo contrario podríamos estar desperdiciando recursos e incluso las conversiones no estarían dentro de la escala deseada.
* Es recomendable revisar a detalle cada una de las instrucciones ya que el programa solo detecta errores de sintaxis y no lógicos, así que al menos en cuanto a la sintaxis deberíamos los típicos errores que se presentan, como podría ser el llamado de una subrutina inadecuadamente.

**ANEXOS**

Mensaje Inicial



Presentando Conversión Analógica – Digital



Presentando Juego

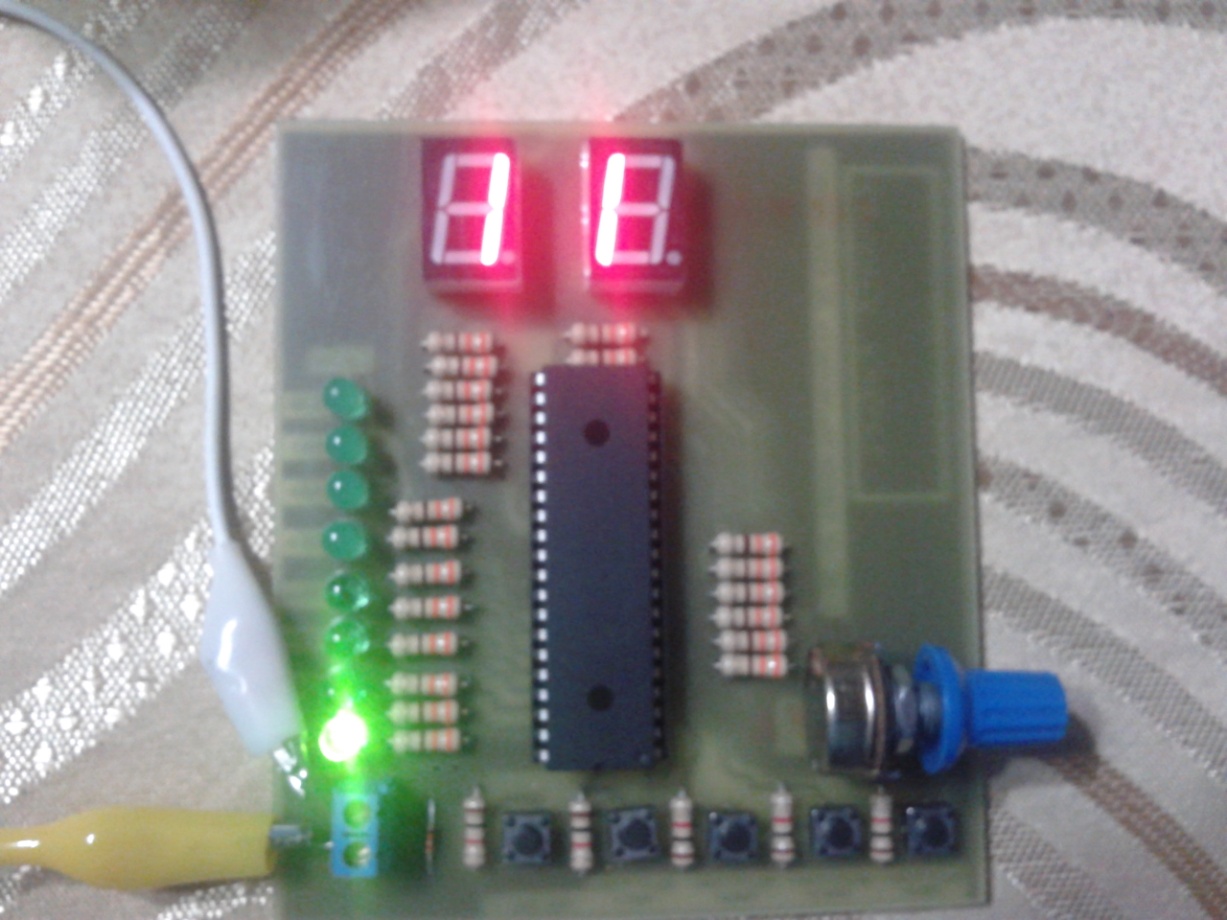


Foto del Autor

