# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

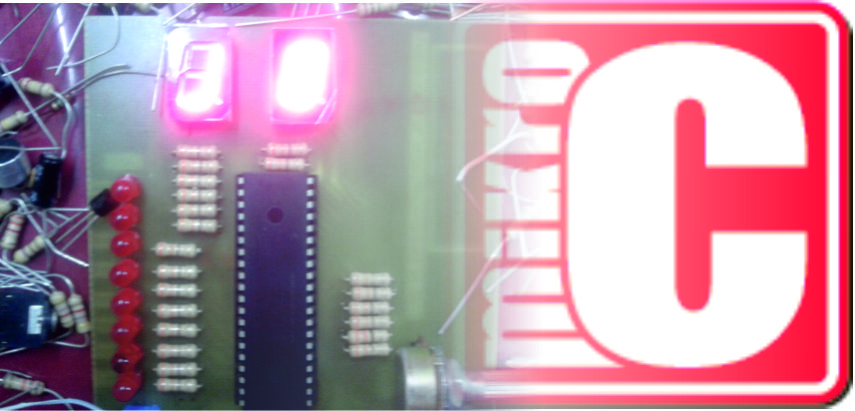




fiec

**FACULTAD DE INGENIERIA EN**

**EN ELECTRICIDAD Y COMPUTACIÓN**

****

**LABORATORIO DE MICROCONTROLADORES**

**Proyecto 2do Parcial**

**Lectura de valor Analógico de un potenciómetro y juego de adivinanza de números**

**José Olmedo Bedón Sánchez**

**Paralelo: 8**

**Profesor: Ing. Carlos Valdivieso**

**Fecha de entrega: Miércoles 25/01/2012**

1. **ENUNCIADO DE PROYECTO**

En este proyecto se procedió a incorporar, 2 proyectos por separado, los cuales se podrán seleccionar mediante los 2 primero botones, para los cuales con la botonera 1, se accederá al proyecto de “LECTURA DE VALOR ANALÓGICO DE UN POTENCIOMETRO”, y con la botonera 2 se accederá al proyecto de “JUEGO DE ADIVINANZA DE NUMEROS”; en un comienzo se presentará las iniciales del autor.

Para el primer proyecto, consiste en mediante una entrada analógica que va conectada a un potenciómetro, se lea estos valores, para realizar su respectiva conversión ADC, que posee el PIC, luego se procederá a presentar en un display de 7 segmentos cátodo común, donde el máximo valor de lectura será 99 y el mínimo 00; también se debe mostrar en los leds, de forma binaria su respectivo valor, tomando en cuenta q su máximo valor es de 8 bits es decir hasta 255 y el mínimo valor es 0.

Para el segundo proyecto, consiste en un principio elegir el nivel de dificultad del juego que contiene 6 niveles, del 0 al 5, donde dependerá el nivel del transcurso entre la presentación del próximo número y el número de oportunidades para el jugador, el juego irá de largo hasta cuando se hallan agotado todas sus oportunidades, el jugador tendrá un tiempo determinado para adivinar el número luego de esto, se mostrará el display una figura de ÉXITO o FALLO, si es FALLO mostrará el número generado, todos los leds se encenderán y procederá con el siguiente número, y quitando oportunidades, en caso de ÉXITO se darán puntos y los leds parpadearán, para la selección del número esto se realizará con las 4 botoneras disponibles, esto nos deja que el máximo número generado será 15 y el mínimo 0.

Para todos los casos anteriores, durante cualquier instante de cualquier proyecto, con el botón reset, regresará el proyecto a su estado inicial, donde se presenta las iniciales.

1. **DIAGRAMA DE BLOQUES**

Botón

Reset

Botón 1

Proyecto 1

Start (juego Bit 3)

Botón 2

Proyecto 2

(juego Bit 2)

Botón 3

(juego Bit 1)

Botón 4

(juego Bit 0)

**PIC 16F887**

Potenciómetro

LEDS (8bits)

Display 7 segmentos

decena

Display 7 segmentos

unidad

1. **DIAGRAMA DE FLUJO FUNCIONAL DEL PROGRAMA PRINCIPAL**

**MAIN**

Configuraciones entradas analógicas

Deshabilitar comparadores

Configurar Puertos I/O

Entradas: PORT A y PORT E

Salidas: PORTB, PORTC y PORTD

RA1==1

RA2==1

Delay 30ms

Delay 30ms

1

11

11

01

01

11

Declarar variables globales

**FUNCIONES - PROCEDIMIENTOS**

unsigned short mask(unsigned short num);

Num==0

Num==1

Num==2

Num==3

Num==4

Num==5

Num==6

Num==7

Num==8

Num==9

11

11

11

11

11

11

11

11

11

11

01

01

01

01

01

01

01

01

01

void iniciales();

Setear J en PORTD

PORTD=0b00001110

Setear B en PORTC

PORTD=0b01111111

!(RA1\_bit || RA2\_bit)

1

0

void winDisplay();

void loseDisplay();

PORTD=0b00001001

PORTC=0b00111001

PORTD=0b00001001

PORTC=0b00001111

void secuenciaWin();

i=0

i<10

PORTB=0b10101010;

Delay\_ms(50);

PORTB=0b01010101;

Delay\_ms(50);

PORTB=0b00000000;

0

1

void secuenciaLose();

i=0

PORTB=0b11111111;

Delay\_ms(500);

PORTB=0b00000000;

Delay\_ms(50);

void configGame();

oportunidades=3;

ptsWin=2\*(nivel+1);

ptsLose=nivel+1;

void velocidad();

Nivel==0

Nivel==1

Nivel==2

Nivel==3

Nivel==4

Nivel==5

Delay\_ms(6000)

Delay\_ms(5000)

Delay\_ms(4000)

Delay\_ms(3000)

Delay\_ms(2000)

Delay\_ms(1000)

void proyecto2();

Adquirir valor analógico a digital

Temp\_res=ADC\_Read(0)

Transformar adquisición 10bits a 6posiblidades (niveles juego)

resDisplay1=((double)temp\_res)/1024;

resDisplay2=(short)(resDisplay1\*6);

Guardar nivel elegido

RA1==0

Randear número

Entradas 4bits == numero aleatorio

puntaje=puntaje+ptsWin;

WinDisplay();

Delay\_ms(500);

secuenciaWin();

1

0

1

0

Puntaje!=0

puntaje=puntaje-ptsLose;

Oportunidades--;

Delay\_ms(500)

Delay\_ms(500)

PORTC=0;

PORTD=0;

Oportunidades!=0

1

0

1

0

void proyecto1();

PORTD=0

PORTC=0

Adquirir valor analógico a digital

Temp\_res=ADC\_Read(0)

Convertir valor para leds

resBin1=((double)temp\_res)/1024;

resBin2=(short)(resBin1\*256);

PORTB=resBin2;

Convertir valor para Display 99

resBin1=((double)temp\_res)/1024;

resBin2=(short)(resBin1\*100);

Delay\_ms(5);

1

void presentarDisplay(int a);

Obtener unidad

digit = a % 10u;

Mostrar en Display unidades

PORTC=digit1;

Obtener decena

digit = (char)(a / 10u) % 10u;

Mostrar en Display decenas

PORTD=digit10;

1. **DESCRIPCION DEL ALGORITMO**

General

1. Configurar entradas Analógicas
2. Configurar entradas y salidas en los puertos
3. Realizar el estado Inicial
4. Mostrar las iniciales J y B sobre los display, mientras no se presiones las botoneras para los dos proyectos
5. Realizar el proyecto 1 en caso de presionar el botón 1
6. Realizar el proyecto 2 en caso de presionar el botón 2
7. Repetir todo de manera indefinida (lazo infinito)

Proyecto 1

1. Setear los display en 0, esto es poner en 0 PORTD y PORTC.
2. Adquirir el valor en digital de la adquisición analógica en AN0
3. Convertir valor de 10bits en valor de 8bits para mostrarlos sobre PORTB
4. Convertir valor de 10bits en valor máximo de 99 para mostrarlos sobre los displays, haciendo la respectiva conversión para unidades y decenas, en código de 7 segmentos.
5. Retardo de 5ms
6. Repetir todo de manera indefinida (lazo infinito).

Proyecto 2

1. Pedir al usuario el nivel de juego,
2. Adquirir el valor en digital de la adquisición analógica en AN0.
3. Convertir valor de 10bits en valor máximo de 5 para definir el nivel, y guardar el nivel en una variable global
4. Presentar el nivel seleccionado en los display, haciendo la conversión respectiva
5. Realiza desde el paso 1 mientras no se presione la botonera 1 de arranque.
6. Configurar el juego dependiendo del nivel seleccionado, ver recuadro de valores para cada nivel
7. Seleccionar velocidad depende del nivel elegido
8. Generar un número aleatorio que será el número a adivinar.
9. Preguntar si las entradas de las cuatro botoneras, son iguales bit a bit al número generado aleatoriamente
10. Realizar sesión WIN en caso del paso anterior sean iguales.
11. Realizas sesión LOSE en caso de paso anterior sean distintos.
12. Dar puntos, mostrar WinDisplay, mostrar efectos en leds, en caso de sesión WIN
13. Si tiene puntos, quitar los puntos, restar oportunidades, mostrar LoseDisplay, mostrar efecto en leds, en caso de sesión LOSE
14. Terminar el juego en caso de acabarse todas las oportunidades
15. Mostrar puntaje final sobre los display
16. Regresar a los pasos generales.

*Cuadro de Configuraciones del juego genérico*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nivel** | **Oportunidades** | **Retardo (ms)** | **Pts Ganados** | **Pts Perdidos** |
| 0 | 3 | 6000 | 2 | 1 |
| 1 | 3 | 5000 | 4 | 2 |
| 2 | 3 | 4000 | 6 | 3 |
| 3 | 3 | 3000 | 8 | 4 |
| 4 | 3 | 2000 | 10 | 5 |
| 5 | 3 | 1000 | 12 | 6 |

**CODIGO FUENTE DEL PROGRAMA**

/\*

\* Nombre del Proyecto:

Proyecto2.c

\* Nombre del Autor: José Olmedo Bedón Sánchez

(c) Mikroelektronika, 20011.

\* Description:

Realiza dos proyectos para los dos primero botones, el primero consiste en lectura del valor de un potenciómetro y el segundo, de un juego de adivinanza, que dependerá del nivel de dificultad elegido en un principio

\* Test configuration:

MCU: PIC16F887

Oscillator: HS, 08.0000 MHz

SW: mikroC PRO for PIC

\*/

unsigned int temp\_res;

unsigned double resBin1,resDisplay1;

unsigned int digit\_no, digit10, digit1, digit,retardo;

unsigned short resBin2,resDisplay2;

unsigned short mask(unsigned short num);

unsigned char nivel=0,nRand,ptsWin,ptsLose,oportunidades,puntaje=0;

void iniciales();

void proyecto1();

void proyecto2();

void presentarDisplay(int a);

void configGame();

void winDisplay();

void loseDisplay();

void secuenciaWin();

void secuenciaLose();

void velocidad();

void main() {

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

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

TRISA = 0xFF; // PORTA is input

C1ON\_bit = 0; // Disable comparators C2ON\_bit = 0;

C2ON\_bit = 0;

TRISB =0; // PORTB is output

TRISC=0; // PORTC is output

TRISD=0; // PORTD is output

TRISE=0xFF; // PORTE is input

PORTB=0; // Turn off all display segments

PORTC=0; // Turn off all display segments

PORTD=0; // Turn off all display segments

do{

iniciales();

if(RA1\_bit){

Delay\_ms(30);

proyecto1();

}

else if(RA2\_bit){

Delay\_ms(30);

proyecto2();

}

}while(1);

}

unsigned short mask(unsigned short num) {

switch (num) {

case 0 : return 0x3F;

case 1 : return 0x06;

case 2 : return 0x5B;

case 3 : return 0x4F;

case 4 : return 0x66;

case 5 : return 0x6D;

case 6 : return 0x7D;

case 7 : return 0x07;

case 8 : return 0x7F;

case 9 : return 0x6F;

}

}

void iniciales(){

do{

PORTD=0b00001110; //letra J

PORTC=0b01111111; //letra B

}while(!(RA1\_bit || RA2\_bit));

}

void proyecto1(){

PORTD=0;

PORTC=0;

do{

temp\_res = ADC\_Read(0);

resBin1=((double)temp\_res)/1024;

resBin2=(short)(resBin1\*256);

PORTB=resBin2;

resDisplay1=((double)temp\_res)/1024;

resDisplay2=(short)(resDisplay1\*100);

presentarDisplay(resDisplay2);

Delay\_ms(5);

}while(1);

}

void presentarDisplay(int a){

digit = a % 10u;

digit1 = mask(digit); // Prepare mask for displaying ones

PORTC=digit1;

digit = (char)(a / 10u) % 10u;

digit10 = mask(digit); // Prepare mask for displaying tens

PORTD=digit10;

}

void proyecto2(){

//seleccion de velocidad

unsigned char a;

do{

temp\_res = ADC\_Read(0);

resDisplay1=((double)temp\_res)/1024;

resDisplay2=(short)(resDisplay1\*6);

nivel=resDisplay2;

presentarDisplay(resDisplay2);

}while(!RA1\_bit);

configGame();

do{

velocidad();

nRand=rand()%16;

if(nRand.B3==RA1\_bit && nRand.B2==RA2\_bit && nRand.B1==RA3\_bit && nRand.B0==RA4\_bit){

puntaje=puntaje+ptsWin;

winDisplay();

Delay\_ms(500);

secuenciaWin();

}else{

if(puntaje!=0)

puntaje=puntaje-ptsLose;

oportunidades--;

loseDisplay();

Delay\_ms(500);

presentarDisplay(nRand);

Delay\_ms(500);

secuenciaLose();

}

PORTC=0;

PORTD=0;

}while(oportunidades!=0);

//guardarPuntaje(puntaje);

do{

presentarDisplay(puntaje);

Delay\_ms(2000);

}while(1);

}

void configGame(){

oportunidades=3;

ptsWin=2\*(nivel+1);

ptsLose=nivel+1;

}

void winDisplay(){

PORTD=0b00001001;

PORTC=0b00001111;

}

void loseDisplay(){

PORTD=0b00001001;

PORTC=0b00111001;

}

void secuenciaWin(){

unsigned char i;

for(i=0;i<10;i++){

PORTB=0b10101010;

Delay\_ms(50);

PORTB=0b01010101;

Delay\_ms(50);

}

PORTB=0b0000000;

}

void secuenciaLose(){

PORTB=0b11111111;

Delay\_ms(500);

PORTB=0b00000000;

Delay\_ms(50);

}

void velocidad(){

switch(nivel){

case 0:

Delay\_ms(6000);

break;

case 1:

Delay\_ms(5000);

break;

case 2:

Delay\_ms(4000);

break;

case 3:

Delay\_ms(3000);

break;

case 4:

Delay\_ms(2000);

break;

case 5:

Delay\_ms(1000);

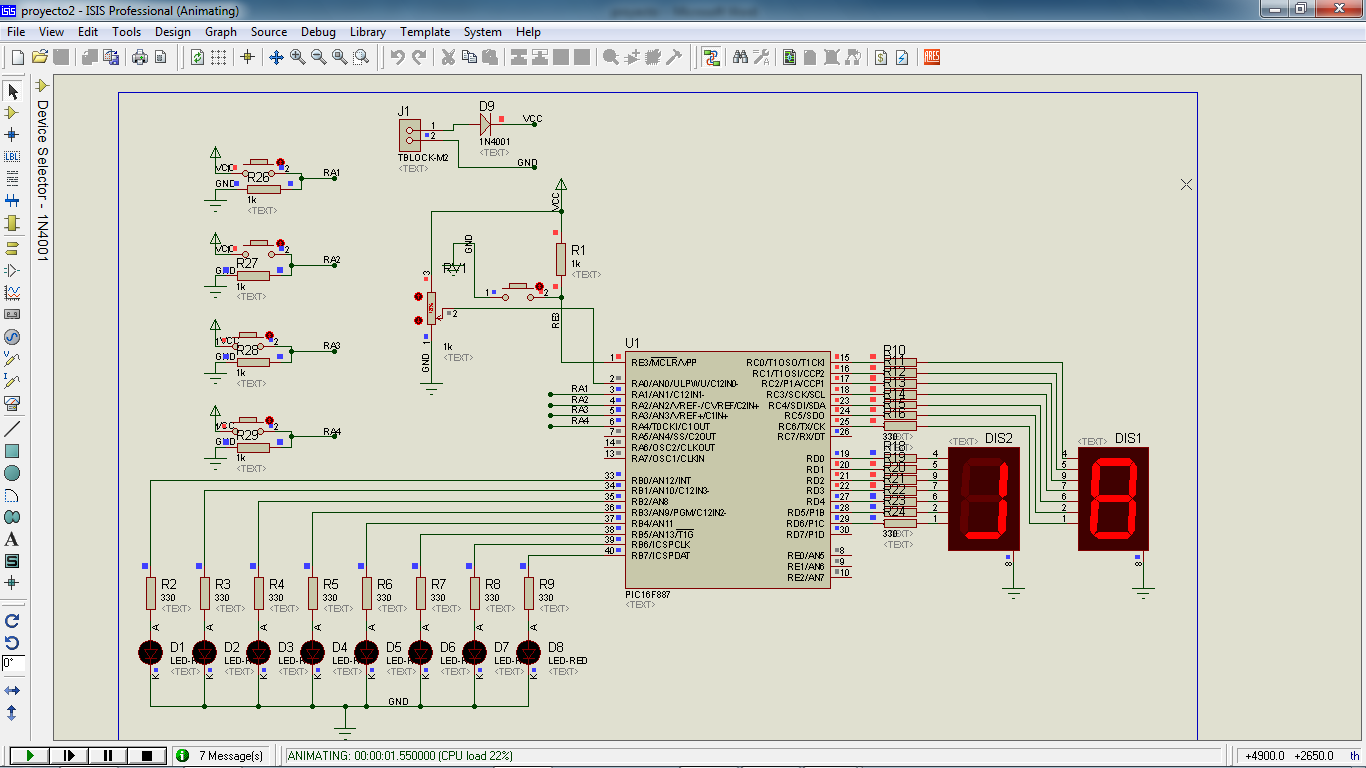
break;

}

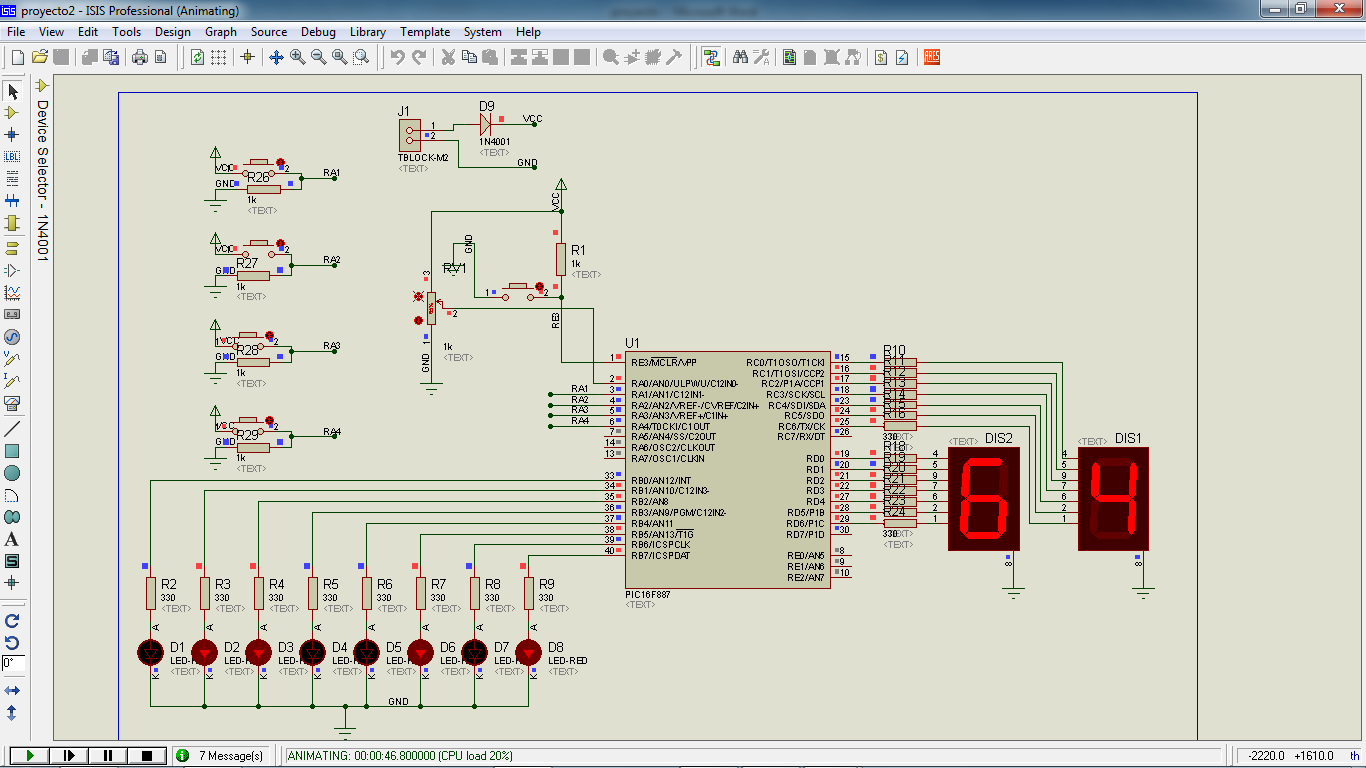
}

1. **CIRCUITO IMPRESO**

**Estado Inicial**

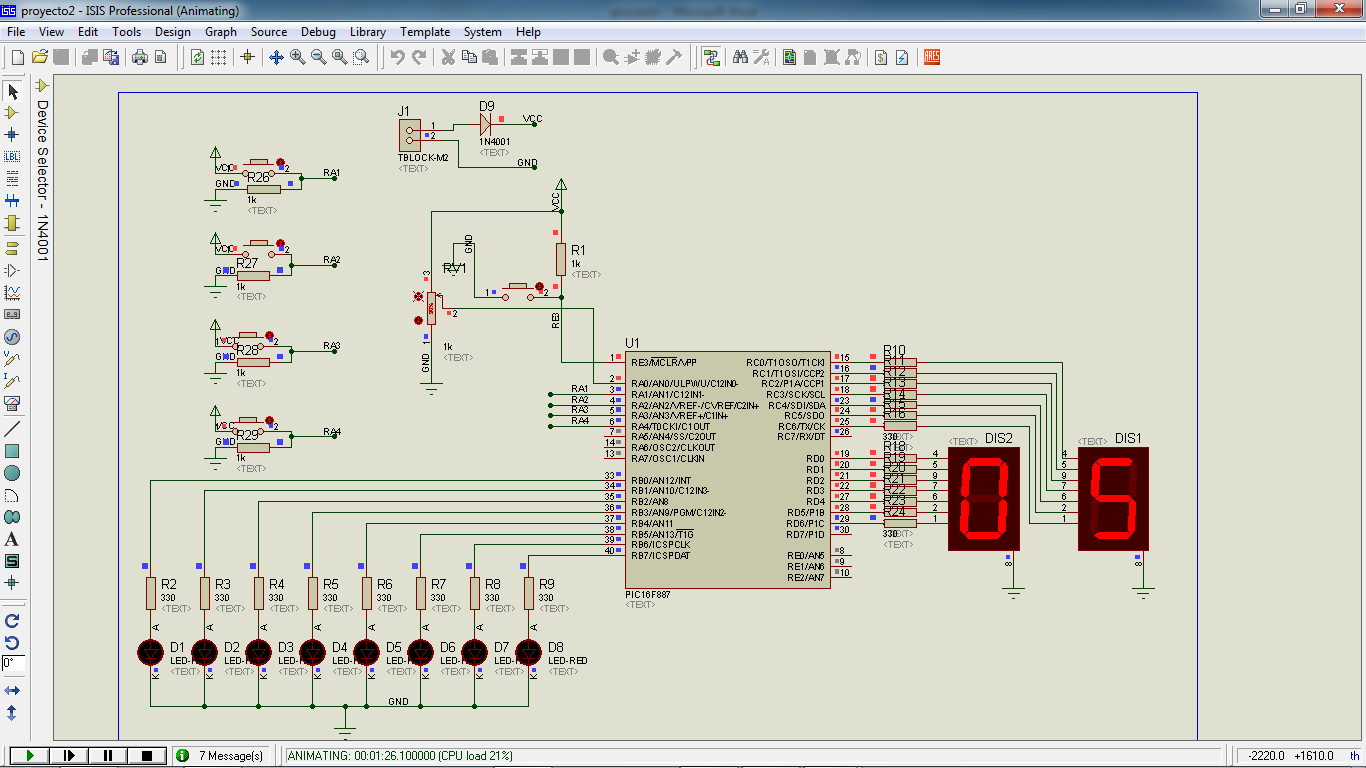


**Proyecto1**

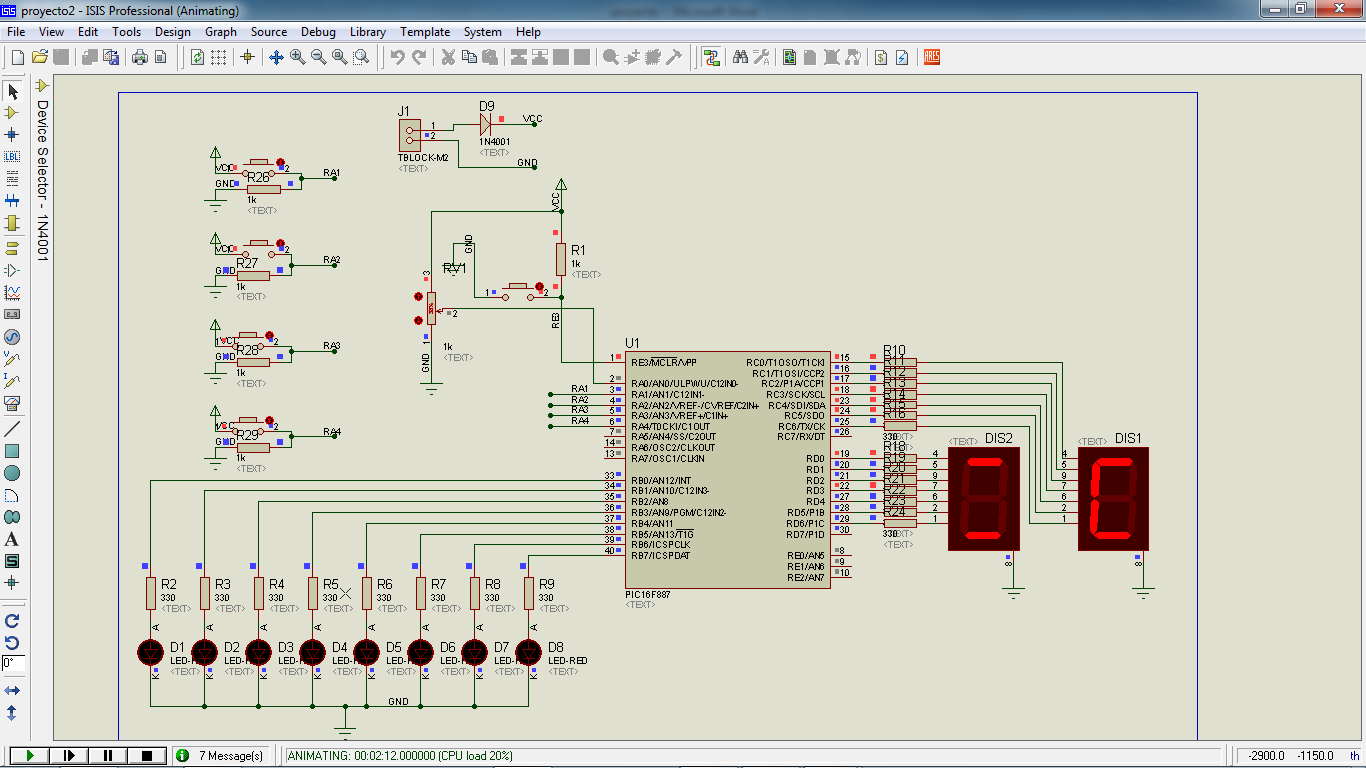


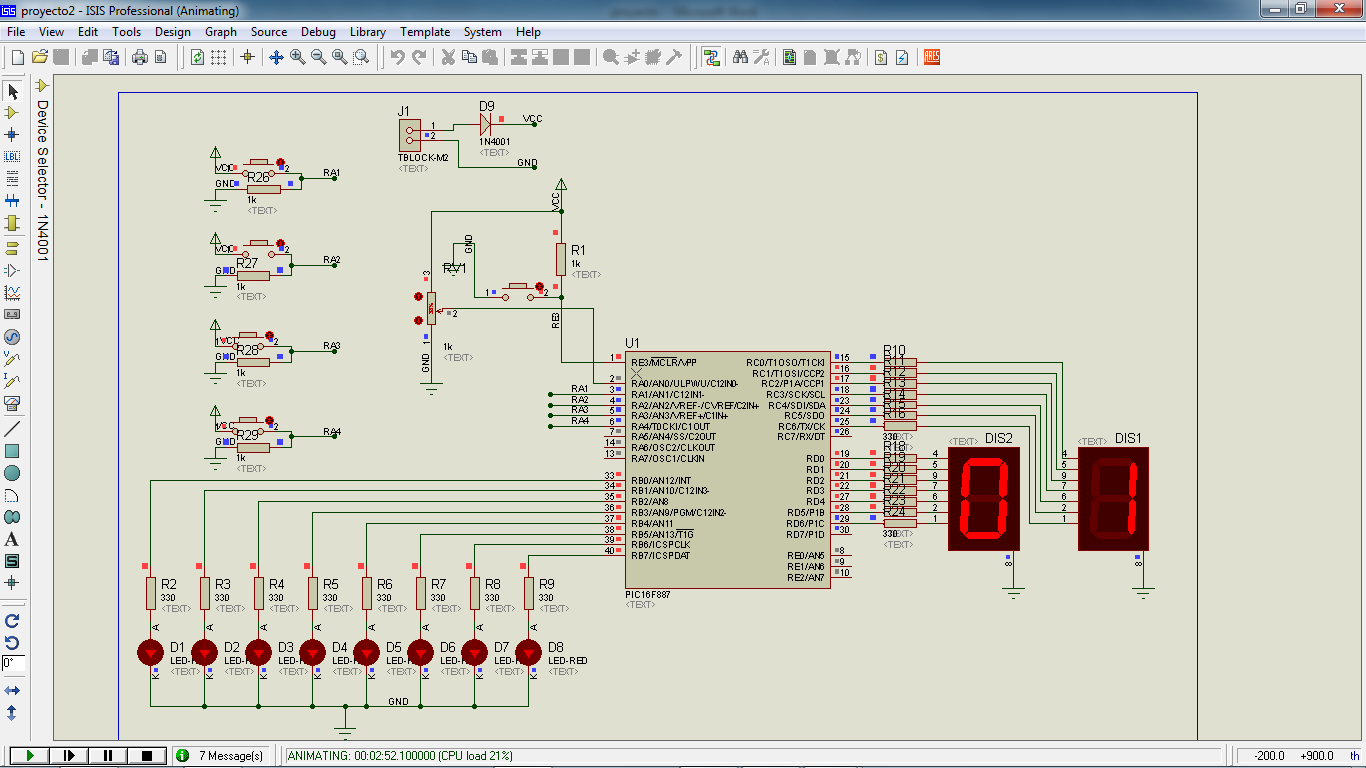
**Proyecto 2**

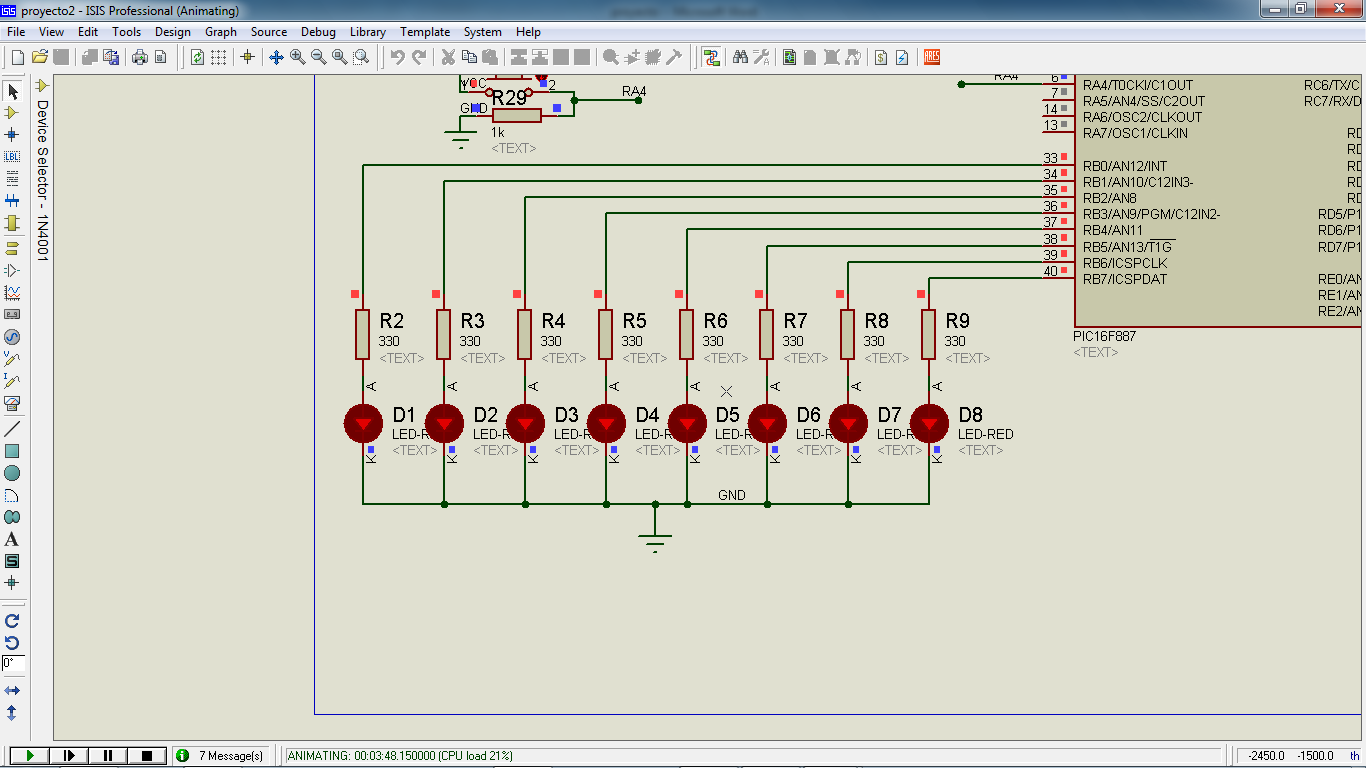
Selección de nivel

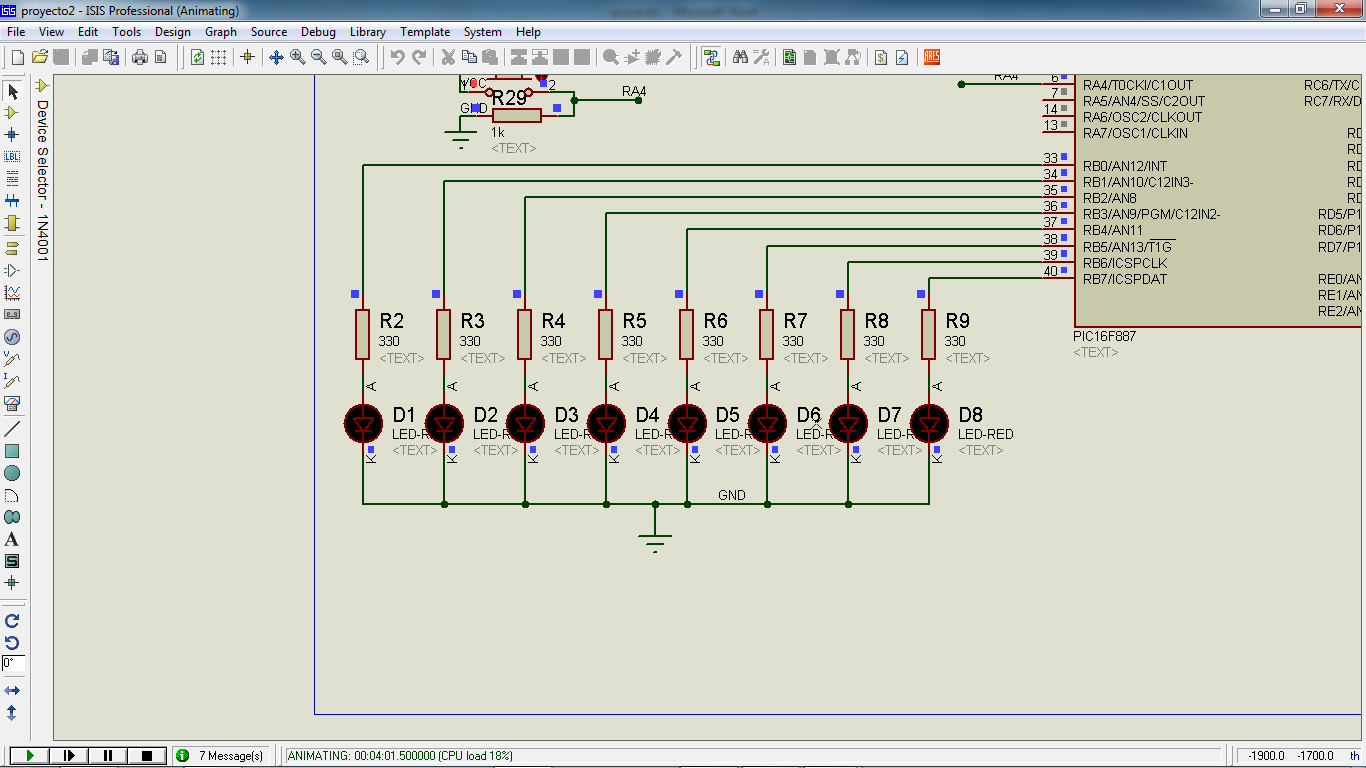


Número errado

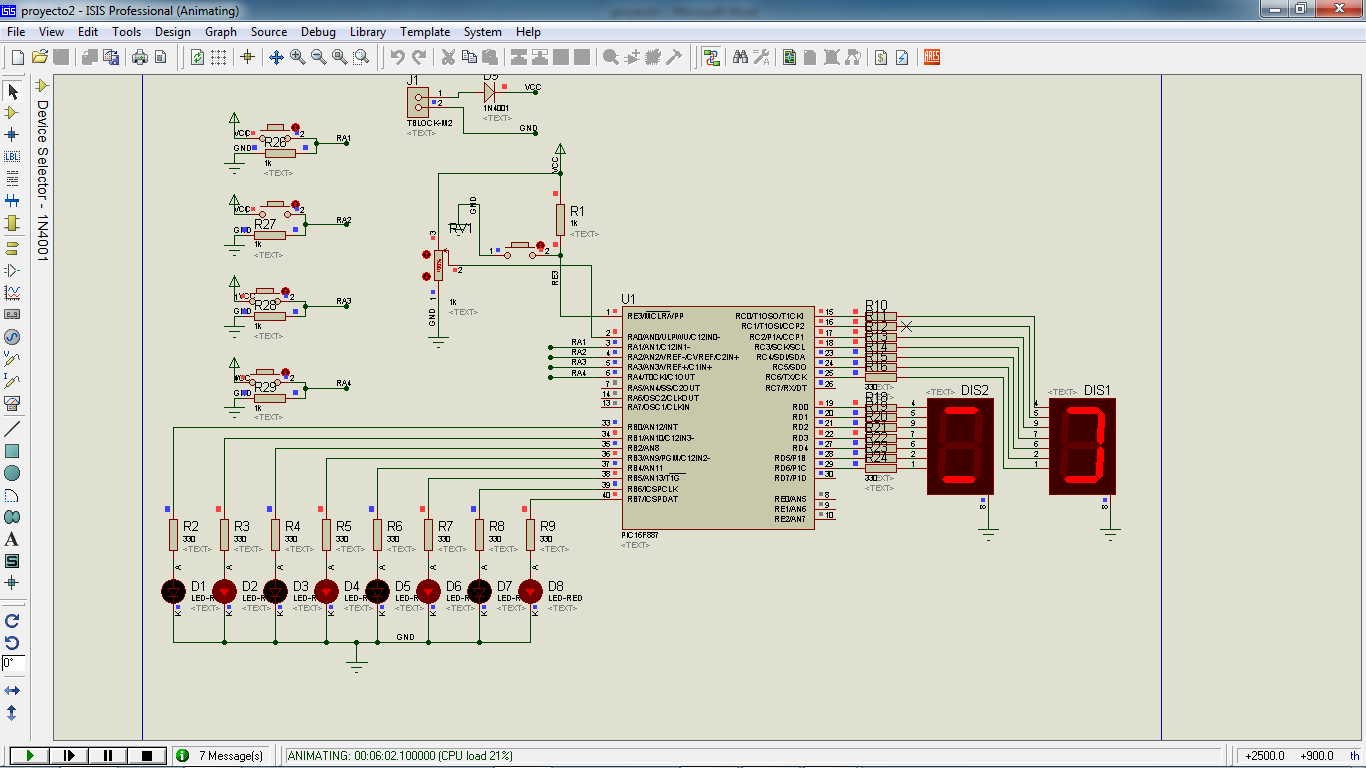


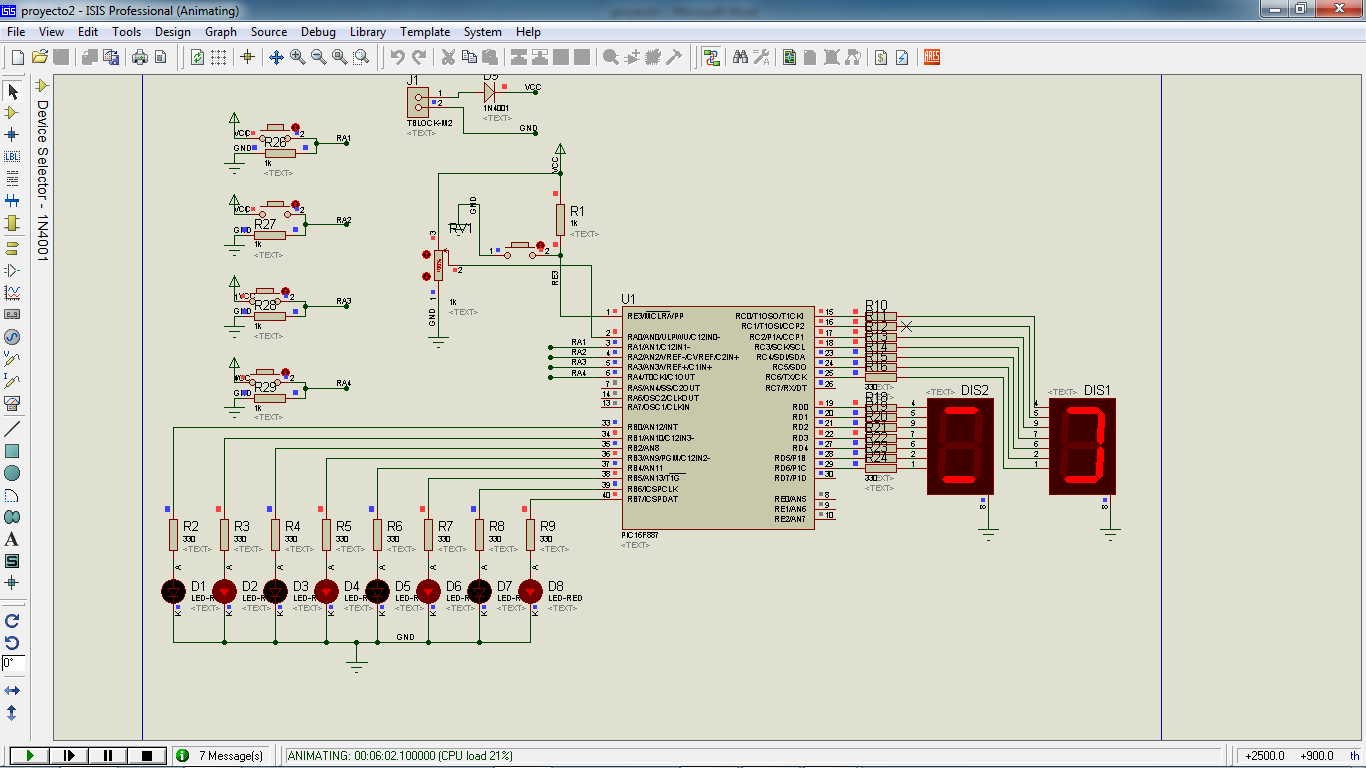


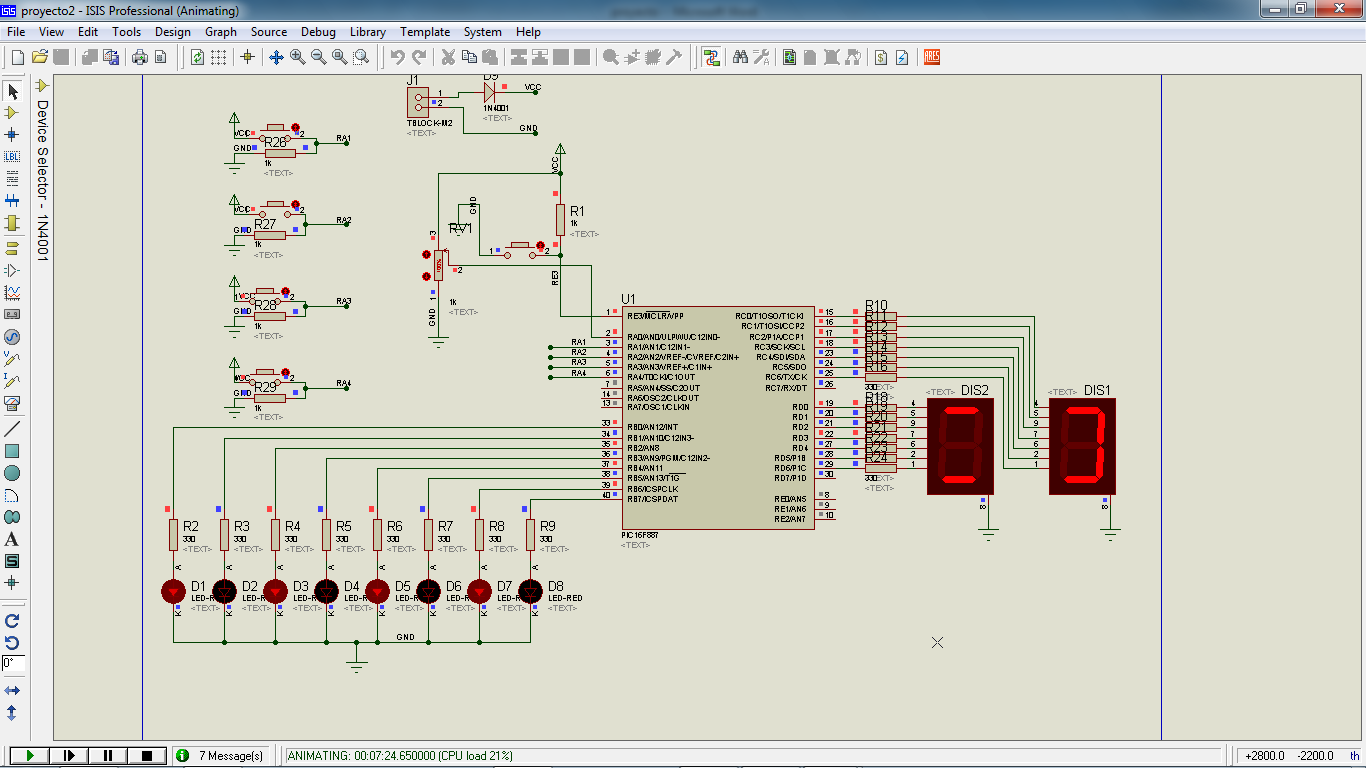




Número acertado







1. **CONCLUSIONES**
2. En este proyecto, se tuvo la necesidad, de realizar un minucioso estudio, para poder modularizar correctamente el programa, esto es creando las funciones o procedimientos necesarios, y correctos, para reutilizarlos, y evitar escribir código que pudiera verse repetitivo a lo largo del programa, un caso particular es el procedimiento de presentación sobre el display, durante todo el programa se vio la necesidad de usar varias veces esta función para su respectivo funcionamiento, es aquí donde colocar este código en un procedimiento era la mejor solución.
3. Pudimos constatar las diferentes herramientas y métodos para resolver problemas, que hemos ido adquiriendo durante el paso del curso de laboratorio de Microcontroladores, como es el caso, del uso de convertido Analógico Digital que posee el PIC, y que se hizo uso, tanto en el primer proyecto para la lectura del potenciómetro, como para el segundo proyecto, que mediante el potenciómetro regulabas el nivel de dificultad del juego.
4. En cuanto al ensamblaje del proyecto, fue de gran ayuda para poder familiarizarnos con los componentes físicos que van conectados al PIC y de los cuales podemos usar información, como son de los pulsadores, darnos valores sobre nuestras entradas, o elementos de salida como son los leds, las cuales nos permitieron confirmar de manera visual el funcionamiento del proyecto, así también como la entrada analógica que era el potenciómetro, y también el uso de los display de 7 segmentos, que nos permitió visualizar en decimal, y evitar estar leyendo en código binario en los leds.
5. Proteus resultó ser un gran soporte para el desarrollo del proyecto, debido que nos evitamos, gastos por tratar de probar nuestros proyectos en funcionamientos, también evitar perdida de tiempo, debido a que estar modificando a cada rato nuestro programa, nos lleva a estar grabando sobre el PIC, varias veces, y resulta una pérdida de tiempo al momento de ver el funcionamiento del mismo, es por eso que esta herramienta es de gran ayuda para la simulación y es de gran soporte para poder confirmar posteriormente el funcionamiento correcto del proyecto
6. **RECOMENDACIONES**
7. Durante el proceso de soldar, sobre la placa PCB, se procura tomar ciertas reglas al momento de soldar, para que no haya problemas en un futuro, estas reglas podrían ser como uso, de un cautín en excelente estado, que no tenga las puntas oxidadas, o estén con desperdicios de soldadas pasadas, hacer uso de la pasta para soldar, antes de insertar los elementos proceder a limar los pines o las patitas, porque hay momentos en las cuales, están sucias, o vienen de fabrica recubierta de un esmalte de protección.
8. Al momento de probar la placa, y comprobar que este correctamente soldada, hacer prueba de continuidad, dando referencia tierra, e ir tocando con la punta del multímetro sobre cada pin que haya en la placa, también probar continuidad sobre carriles que estén juntos, pueda ser que este alguno haciendo contacto, y en la primera prueba de continuidad a tierra, no lo hubiésemos notado.
9. Tener a la mano la hoja con los bancos y registros necesarios para poder desarrollar nuestro proyecto, así como el contenido bit a bit de cada registro, esto nos facilita estar buscando a cada rato, ciertos registros que estemos buscando
10. Verificar polaridades de elementos q los requieran, como es el caso de los leds y el sentido del PIC, la cual se puede ver en el diagrama proporcionado para el proyecto, al cual nos muestra en que sentido debemos colocarlos, la mal conexión de los mismos nos lleva a la posibilidad de quemar algún elemento del circuito, mas aún el PIC, donde su señalización nos damos cuenta por una pestaña que tiene en un extremos del PIC, donde empieza la marca, empieza el pin 1

**CIRCUITO IMPRESO**

