# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

# **FACULTAD DE INGENIERIA EN**

# **EN ELECTRICIDAD Y COMPUTACIÓN**



**PROYECTO SEGUNDO PARCIAL**

Lectura de valor analógico de un potenciómetro y Un juego de números pares e impares con elementos de la tarjeta

**ALUMNA:**

Maria Auxiliadora Coronel Lainez

**Paralelo:**

**12**

**Fecha de presentación:**

25/01/2012

**1.- ENUNCIADO DEL PROYECTO**

**EJERCICIO 1**

**DESCRIPCION:**

Lectura de los valores analógicos de un potenciómetro colocado en la bornera de entrada analógica, presentando las variaciones de valores que están en binario, en 8 LEDs y en dos displays de 7 segmentos que son en dos dígitos decimales disponibles. Todo esto está programado en C.

**2.- dIAGRAMA DE BLOQUES**

Lectura de valor en binario en Leds del PORTB

Lectura de valor decimal en Displays de 7 segmentos

PIC16F887

BOTONERAS

POTENCIOMETRO

**3.- descripcion del algoritmo**

a. Para darle inicio debemos presionar el botón **RA0.** El cual genera un pulso que activara la entrada analógica en este caso el potenciómetro.

b. Pasa por el proceso de conversión de analógico a digital

c. En el PORTB se encuentran los leds y estos leerán el valor de la ADC en binario

d. En el PORTC Y PORTD se encuentran los displays de siete segmentos que mostraran el valor de la conversión ADC pero esta vez en decimal.

**4.- Diagrama de Flujo**

INICIO

PARAMETROS

PUERTOS ANALOGICOS Y DIGITALES

SELECCIÓN DE PUERTOS

LECTURA DEL MODULO ADC

ENVIAR AL PUERTO B,C y D

**1.- ENUNCIADO DEL PROYECTO**

**EJERCICIO 2**

**DESCRIPCION:**

Programa en C, que consiste en el cambio de secuencias de los leds dependiendo si el número mostrado en los displays es par o impar, el cambio de los displays de siete segmentos son controladas por las botoneras ubicadas en el PORTA

**2.- dIAGRAMA DE BLOQUES**

LEDS

Displays de 7 segmentos

PIC16F887

BOTONERAS

**3.- descripcion del algoritmo**

1. Primero se inicializan las variables que se utilizaran en el programa.
2. Seteamos los puertos del **PIC16F887** como entradas o salidas digitales
3. Al presionar el botón RA4 estamos dando inicio a nuestro programa, a su vez este botón hace de Reset.
4. Una vez iniciado el programa comenzamos a configurar los displays con las botoneras **RA2** y **RA3** que corresponden a las decenas y a las unidades de un número decimal.
5. Una vez con un valor mostrado en los displays al presionar el botón en **RA1** mostrara dos secuencias diferentes en los leds que dependerá del número mostrado en los displays si este es par o impar.

**3.-DIAGRAMA DEL ALGORITMO**

INICIO

Inicialización de variables

Seteo de Puertos

Presiona el botón RA4

NO

SI

Configuración de los displays de 7 segmentos en RA2 y RA3

Presiona el botón RA1

NO

SI

Es par

SI

Presenta una secuencia 2 en los leds

NO

Presenta una secuencia 1 en los leds

**4.-Diagramas de Flujo**

INICIO

PORTB=0; PORTC=0x39; PORTD=0x77; PORTA=255; cont=0; conta=0.

RA4

NO

SI

PORTC=0X3F; PORTD=0X3F; PORTB=0X00; CONT=0; CONTA=0

NO

RA1

SI

NO

NO

RA3

RA2

SI

PORTD=mask(cont)

Cont=0

Cont=10

SI

NO

Conta=10

NO

Cont=cont+1

numero=(cont\*10)+conta

SI

Conta=conta+1

PORTC=mask(conta)

Conta=0

SI

SI

NO

(numero%2)==0)&&(numero!=0)

PORTB=0XC0;

PORTB=0X30;

PORTB=0X03;

PORTB=0X0C

PORTB=0X81;

PORTB=0X42;

PORTB=0X24;

PORTB=0X18

**5.- LISTADO DELPROGRAMA FUENTE**

/\* ======= PROYECTO DE MICROCONTROLADORES ========

SEGUNDO PARCIAL 2011-II TERMINO

Autor: Maria Auxiliadora Coronel Lainez

Descripcion del proyecto: Al inicializar el programa mostrara las iniciales de mi nombre AC (Auxiliadora Coronel)

PARTE I

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.

PARTE II

Descripcion del proyecto: Dependiendo del valor mostrado en los displays (PORTC Y PORTD)

Que son controladas por las botoneras RA2 y RA3

si el numero es par o impar este mostrara una secuencia

en los leds ubicados en el PORTB \*/

long int temp\_res, temporal, num,uni\_num, dec\_num;

short salir=1;

unsigned int temp\_res;

unsigned short cont,conta,numero,conta2; //declaracion de Variables Globales

char i;

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; //Retorna numeros decimales en los

case 5 : return 0x6D; //displays de siete segmentos

case 6 : return 0x7D;

case 7 : return 0x07;

case 8 : return 0x7F;

case 9 : return 0x6F;

}

}

void presentacion(unsigned short numero)

{

numero=(cont\*10)+conta;

if(((numero%2)==0)&&(numero!=0)) //si es un numero PAR en los displays

{

delay\_ms(150);

PORTB=0X81;

delay\_ms(150); // Muestra la secuencia uno

PORTB=0X42;

delay\_ms(150);

PORTB=0X24;

delay\_ms(150);

PORTB=0X18;

delay\_ms(150);

PORTB=0X24;

delay\_ms(150);

PORTB=0X42;

delay\_ms(150);

PORTB=0X81;

delay\_ms(150);

}

else if(((numero%2)==1)&&(numero!=0)){ //Si es un numero IMPAR MOSTRADO EN

delay\_ms(150); //LOS DISPLAYS

PORTB=0XC0;

delay\_ms(150);

PORTB=0X30; // Muestra la secuencia 2

delay\_ms(150);

PORTB=0X0C;

delay\_ms(150);

PORTB=0X03;

delay\_ms(150);

PORTB=0X0C;

delay\_ms(150);

PORTB=0X30;

delay\_ms(150);

PORTB=0XC0;

delay\_ms(150);

}

else{ //Si el numero es igual a cero

Delay\_ms(150);

PORTB=0XFF;

}

}

void potenciometro()

{

while(salir==0){

temp\_res = ADC\_Read(0); //se pregunta por la señal analogica

temporal = (temp\_res)\*97;

num = (temporal)/1000;

PORTB = num; //se muestra en binario a traves de PORTB

dec\_num = num/(10);

uni\_num = num-((dec\_num)\*10);

PORTC = mask(uni\_num); //Se muestra las unidades en PORTC

PORTD = mask(dec\_num); //Se muestra las decenas en PORTD

delay\_ms(50); // Espera 50 mseg

if(RA1\_bit==1){

salir=1;

}

}

}

void main()

{

ANSEL=0x01; // entrada analogica para potenciometro RA0

ANSELH=0; //entradas o salidas digitales

PORTD=0;

TRISD=0; //PORTD, PORTC Y PORTB COMO salidas digitales

PORTB=0;

TRISB=0; //configurando como salidas digitales

PORTC=0;

TRISC=0;

PORTA=255; //PORTA COMO ENTRADA DIGITAL

TRISA=255;

PORTE=255;

TRISE=255;

OPTION\_REG = 0x80; // Set timer TMR0

TMR0 = 0;

PORTD=0X77; //inicilaes de mi nombre A (AUXILIADORA)

PORTC=0X39; //iniciales de mi apellido C (CORONEL)

Delay\_MS(200);

cont=0;

conta=0;

conta2=0;

while(1)

{

if(RA1\_bit==1){ //Si el primer boton es presionado

conta2++; // presenta el resultado en los leds

if(conta2==1){ //Contador que sigue la secuencia de las unidades

PORTC=0X3F; //y decenas

PORTD=0X3F;

// PORTB=0X00;

}

Delay\_ms(200);

presentacion(numero);

}

if (RA2\_bit==1 &&(conta2>=1))

{

Delay\_ms(200);

if(cont==10)

cont=0; //Boton 2 controla las decenas

PORTD=mask(cont);

cont++;

}

if (RA3\_bit==1 &&(conta2>=1)) //Boton 3 controla las unidades

{

Delay\_ms(200);

if(conta==10)

conta=0;

PORTC=mask(conta);

conta++;

}

if(RA4\_bit=1) //4to boton RESETEA y empieza a funcionar

{ //el potenciometro

Delay\_ms(200);

PORTC=0x3F;

PORTD=0X3F;

cont=0;

conta=0;

salir=0;

potenciometro();

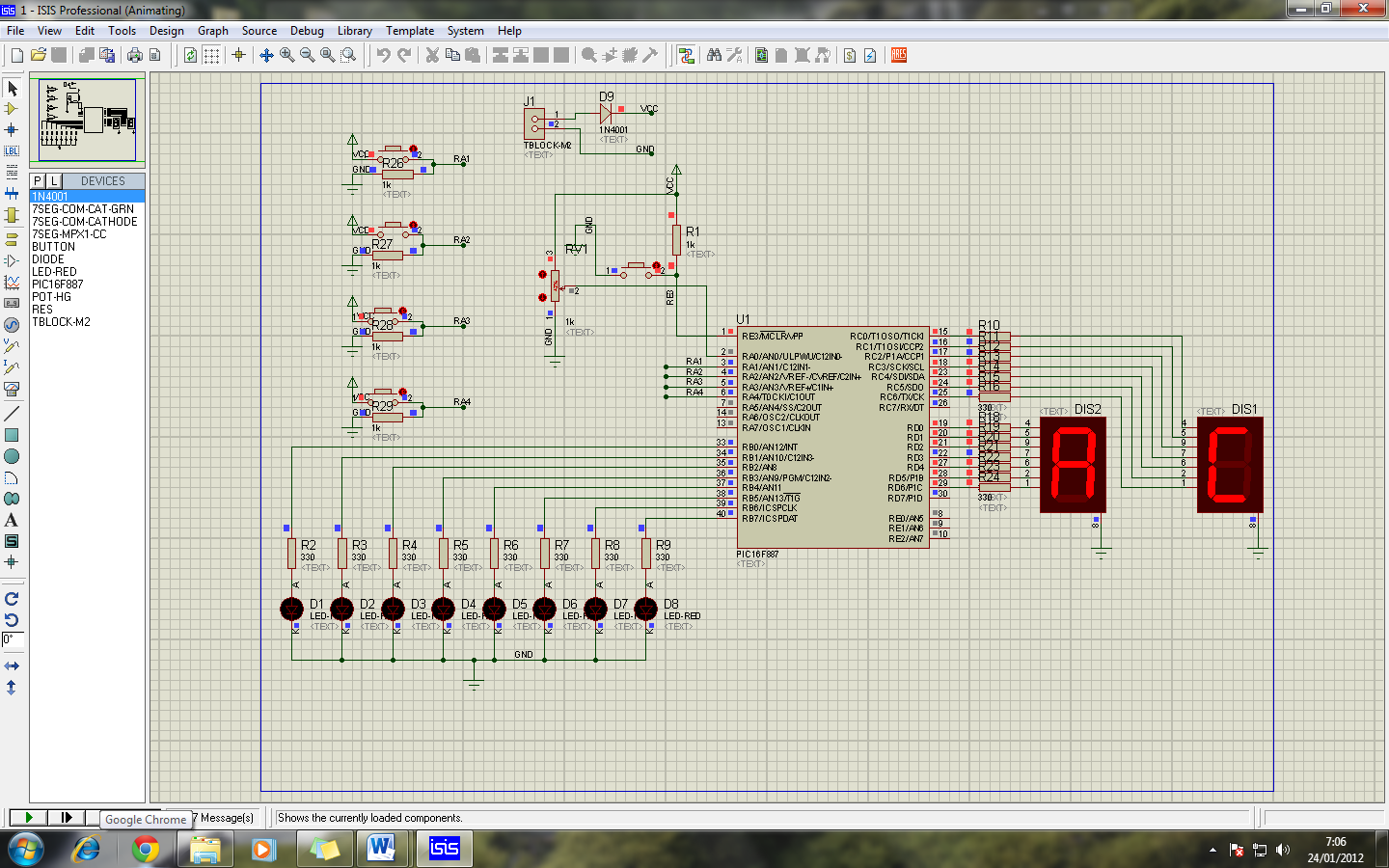
}

}

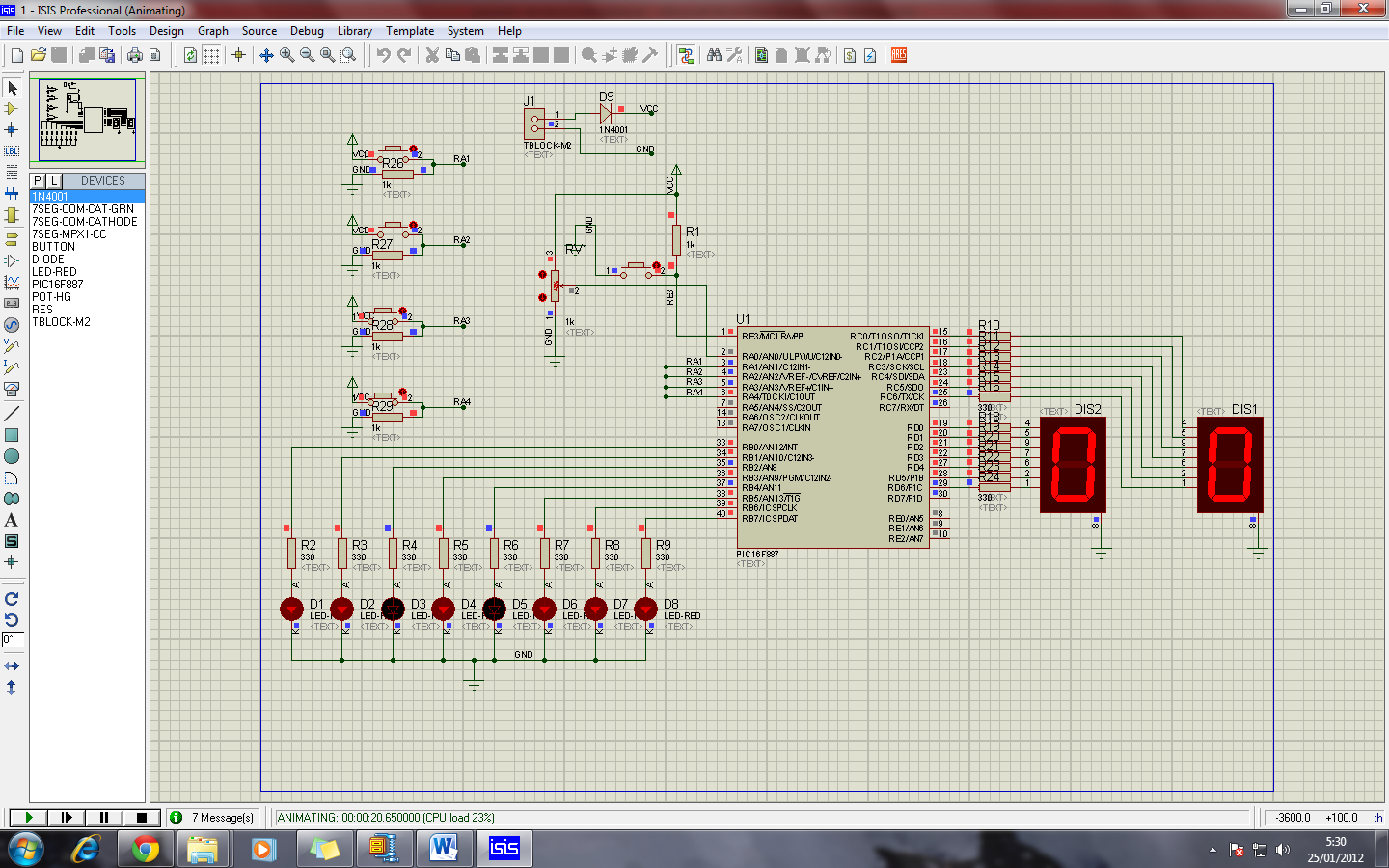
}

**6.- SIMULACIONES**

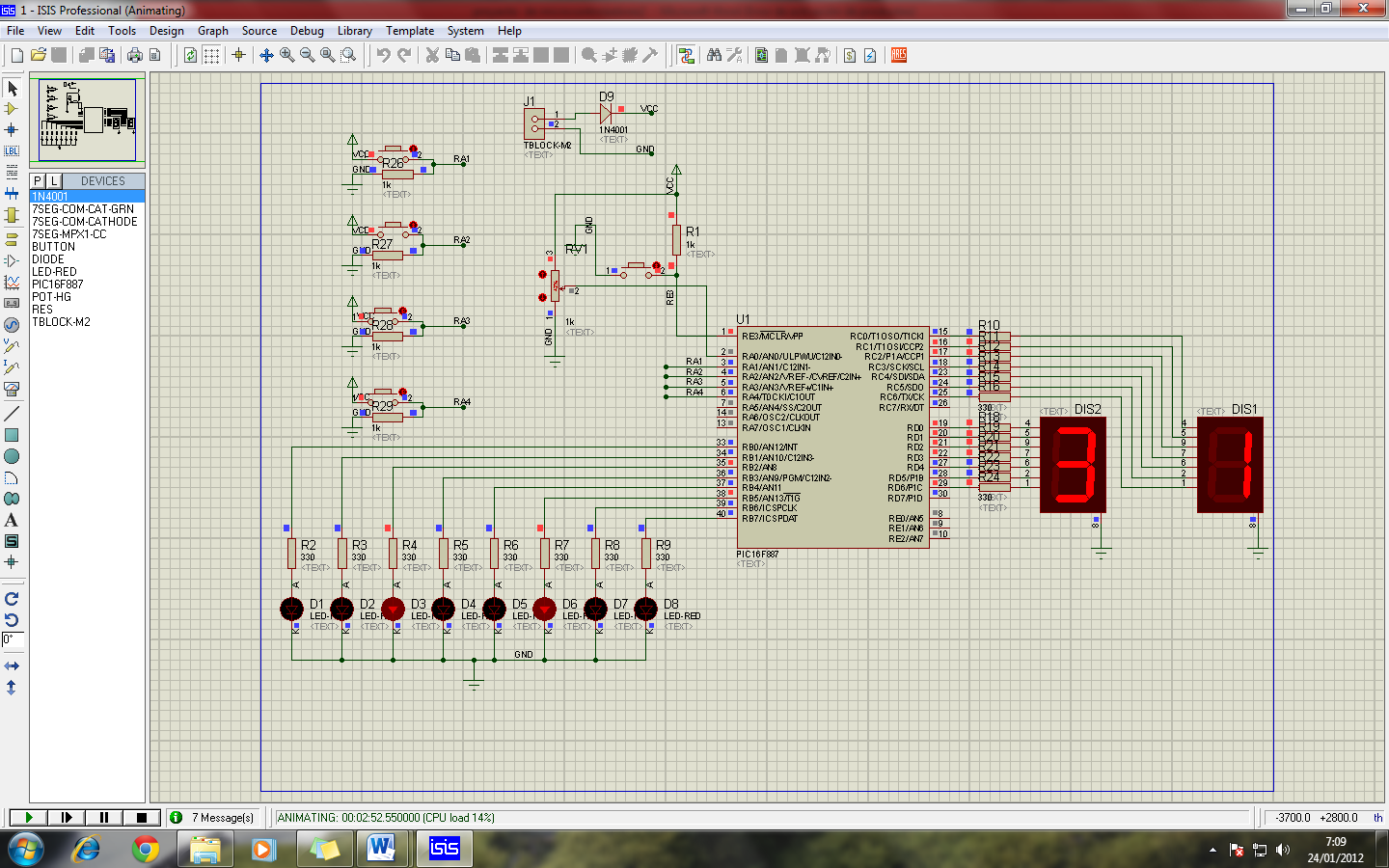
**Mostrar las iniciales del nombre**



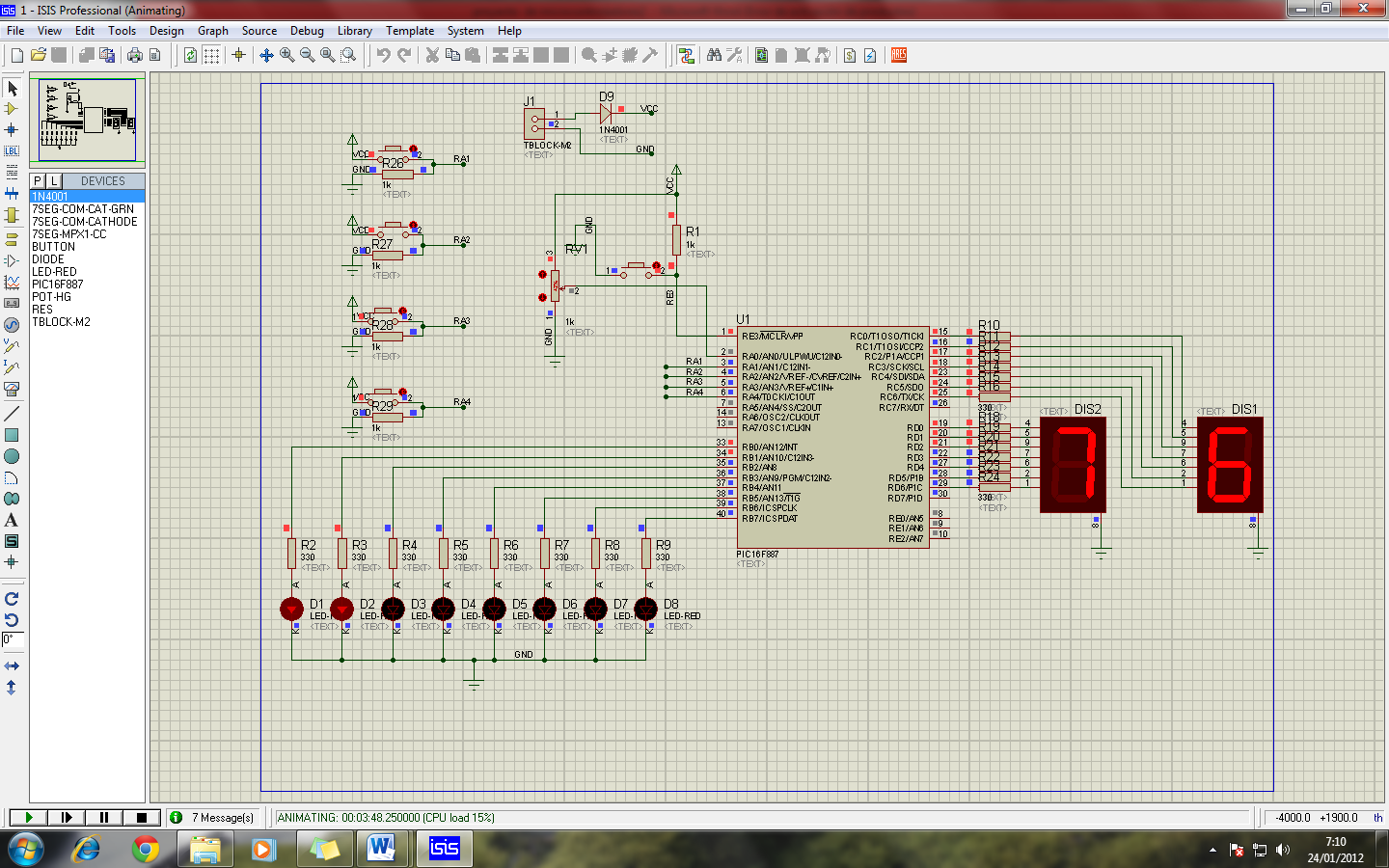
**PARTE I**



**PARTE II**



1. Muestra secuencia de números impar



1. Muestra secuencia de números par

**7.- CONCLUSIONES**

* Nos hemos familiarizado con el entorno de programación en lenguaje C para la programación del microcontrolador, este lenguaje es más amigable que el lenguaje anterior, además lo importante y esencial es dominar un lenguaje para la programación de PIC, pero no es más óptimo y eficiente que el lenguaje anterior.
* Podemos programar cualquier microcontrolador de diversas maneras y crear e implementar juegos con los diferentes dispositivos electrónicos.
* MikroC dispone de una gama de librerías que podemos utilizar sin problema en donde solo necesitamos enviar el parámetro ya que MikroC nos facilita implementar o programar esta función.
* Una señal analógica puede llegar al micro y ser transformada a señal digital con los módulos ADC y así poder gobernar un dispositivo que recepte y muestre la señal ya convertida.
* Puedo concluir diciendo que los dispositivos tienen un tiempo de retardo por lo tanto el retardo que nosotros colocamos en delay en realidad no está en tiempo real.

**8.- RECOMENDACIONES**

1. Siempre revisar el circuito y todas las conexiones que tengamos para que después se produzca la simulación de manera correcta muchas veces, las simulaciones producen errores es por mala conexión de circuito.
2. Compilar el programa cada vez que se lo vaya a simular en el PROTEUS ya que si no se lo hace, el programa en .C no se actualiza por lo tanto puede que no estemos compilando la simulación vigente.
3. Antes de utilizar una función, informarnos bien para que sirven ya sea revisando nuestros apuntes o podemos también verlo en la librería dentro del programa MikroC, para poderla llamar y usarla con éxito.
4. Se recomienda que antes de empezar a programar tengamos una idea global de lo que vamos a realizar, caso contrario perderíamos mucho tiempo en programar un diseño.

**9. CIRCUITO IMPRESO FUNCIONANDO**

