Descripción: fiec

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FIEC**

**Laboratorio de Microcontroladores**

**LECTURA DE VALOR ANALÓGICO DE UN POTENCIÓMETRO Y TEMA LIBRE:**

**LEDS PSICODELICOS**

**Alumna:**

**Blanche Jaqueline Verdezoto**

**Paralelo: # 9**

**Grupo: # 4**

**Fecha de presentación:**

**25/01/2012**

**2011 – 2° TÉRMINO**

**1.- Objetivo:**

El objetivo del siguiente proyecto es desarrollar en lenguaje C y ejecutar 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 #2que consiste en LEDs Psicodélicos haciendo uso de los recursos de la tarjeta prototipo que consta de cuatro botoneras, 8 LEDs y dos DISPLAYs de 7 segmentos.

**2.- Enunciado 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 LEDs Psicodélicos haciendo uso de 4 botoneras para la selección y total 4 tipos mostrándolos en los DISPLAYs de 7 segmentos. Si se presiona la BOTONERA1 se selecciona la velocidad mas baja de movimiento de los LEDs en los Displays. Si se presiona la Siguiente Botonera la velocidad aumenta y asi sucesivamente. Se dispone en de velocidades.

**3.- Diagrama de Bloques**



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



**5.- Descripción del algoritmo o estrategia utilizada.**

* 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 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 en el que esperamos se presione alguna BOTONERA. Si se presiona La BOTONERA1 se muestra patrón psicodélico en los DISPLAYs conectados a los puertos C y D y un patrón de LEDs en el PORTB indefinidamente hasta que se presione alguna otra BOTONERA. Si se presiona la BOTONERA2 se presenta el mismo el patrón pero a una velocidad más rápida. Esto ocurrirá del mismo modo para las BOTONERAS 3 y 4

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

Programa Fuente:

/\*

\* Nombre del Proyecto:

proy2.c

\* Nombre del Autor:

(c) Mikroelektronika, 2009.

\* Description:

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 LEDs Psicodélicos haciendo uso de los recursos de la tarjeta prototipo que consta de cuatro botoneras, 8 LEDs y dos DISPLAYs de 7 segmentos.

\* Test configuration:

MCU: PIC16F887

Oscillator: HS, 08.0000 MHz

SW: mikroC PRO for PIC

/\*Header\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

intdec,uni,decimal,dato,aux;

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

0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F}; //CodigoHexa para cada nuemro

intconst 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

intconst giro[6] = {1,2,4,8,16,32};

inti=0;

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()

{

//declaracion de variables

do{ //se enceran displays

PORTC=disp[0];

PORTD=disp[0];

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

while(1)

{if(PORTA.F1) // si presiona la botonera PORTA.pin1

{

Delay\_ms(300);

while(!(PORTA)){

PORTC=giro[0]; // iniciales

PORTD=giro[0];

Delay\_ms(500);

PORTB=129;

if(PORTA)

break;

PORTC=giro[1]; // iniciales

PORTD=giro[5];

Delay\_ms(500);

PORTB=66;

if(PORTA)

break;

PORTC=giro[2]; // iniciales

PORTD=giro[4];

Delay\_ms(500);

PORTB=36;

if(PORTA)

break;

PORTC=giro[3]; // iniciales

PORTD=giro[3];

Delay\_ms(500);

PORTB=24;

if(PORTA)

break;

PORTC=giro[4]; // iniciales

PORTD=giro[2];

Delay\_ms(500);

PORTB=36;

if(PORTA)

break;

PORTC=giro[5]; // iniciales

PORTD=giro[1];

Delay\_ms(500);

PORTB=66;

if(PORTA)

break;

}

}

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

{

Delay\_ms(300);

while(!(PORTA)){

PORTC=giro[0]; // iniciales

PORTD=giro[0];

Delay\_ms(350);

PORTB=129;

if(PORTA)

break;

PORTC=giro[1]; // iniciales

PORTD=giro[5];

Delay\_ms(350);

PORTB=66;

if(PORTA)

break;

PORTC=giro[2]; // iniciales

PORTD=giro[4];

Delay\_ms(350);

PORTB=36;

if(PORTA)

break;

PORTC=giro[3]; // iniciales

PORTD=giro[3];

Delay\_ms(350);

PORTB=24;

if(PORTA)

break;

PORTC=giro[4]; // iniciales

PORTD=giro[2];

Delay\_ms(350);

PORTB=36;

if(PORTA)

break;

PORTC=giro[5]; // iniciales

PORTD=giro[1];

Delay\_ms(350);

PORTB=66;

if(PORTA)

break;

}

}

if(PORTA.F3) // si presiona la botonera PORTA.pin1

{

Delay\_ms(300);

while(!(PORTA)){

PORTC=giro[0]; // iniciales

PORTD=giro[0];

Delay\_ms(200);

PORTB=129;

if(PORTA)

break;

PORTC=giro[1]; // iniciales

PORTD=giro[5];

Delay\_ms(200);

PORTB=66;

if(PORTA)

break;

PORTC=giro[2]; // iniciales

PORTD=giro[4];

Delay\_ms(200);

PORTB=36;

if(PORTA)

break;

PORTC=giro[3]; // iniciales

PORTD=giro[3];

Delay\_ms(200);

PORTB=24;

if(PORTA)

break;

PORTC=giro[4]; // iniciales

PORTD=giro[2];

Delay\_ms(200);

PORTB=36;

if(PORTA)

break;

PORTC=giro[5]; // iniciales

PORTD=giro[1];

Delay\_ms(200);

PORTB=66;

if(PORTA)

break;

}

}

if(PORTA.F4) // si presiona la botonera PORTA.pin1

{

Delay\_ms(300);

while(!(PORTA)){

PORTC=giro[0]; // iniciales

PORTD=giro[0];

Delay\_ms(50);

PORTB=129;

if(PORTA)

break;

PORTC=giro[1]; // iniciales

PORTD=giro[5];

Delay\_ms(50);

PORTB=66;

if(PORTA)

break;

PORTC=giro[2]; // iniciales

PORTD=giro[4];

Delay\_ms(50);

PORTB=36;

if(PORTA)

break;

PORTC=giro[3]; // iniciales

PORTD=giro[3];

Delay\_ms(50);

PORTB=24;

if(PORTA)

break;

PORTC=giro[4]; // iniciales

PORTD=giro[2];

Delay\_ms(50);

PORTB=36;

if(PORTA)

break;

PORTC=giro[5]; // iniciales

PORTD=giro[1];

Delay\_ms(50);

PORTB=66;

if(PORTA)

break;

}

}

}

}

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[22]; // iniciales

PORTD=id[9]; // iniciales

PORTB=0;

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

p1();

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

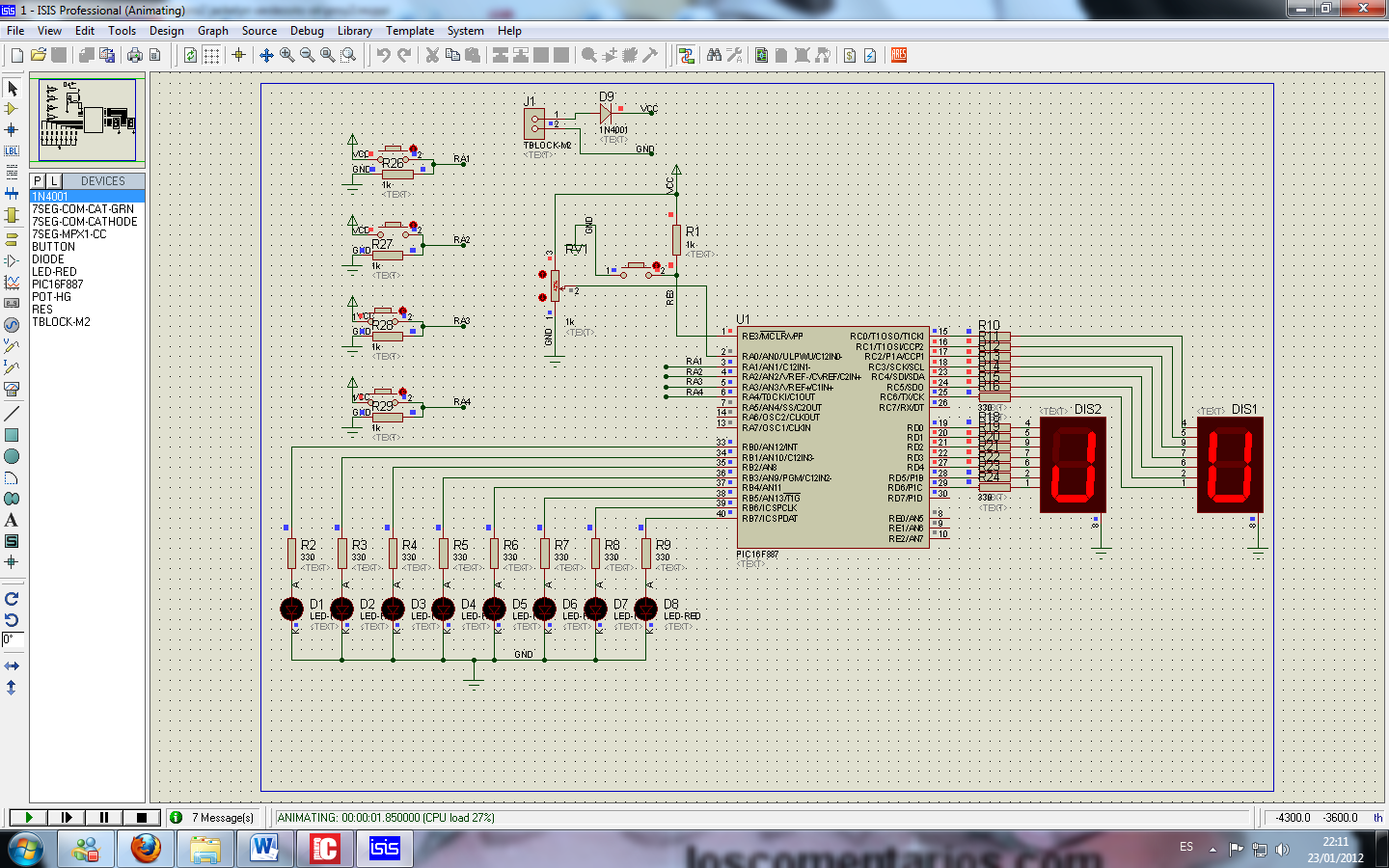
p2();

}

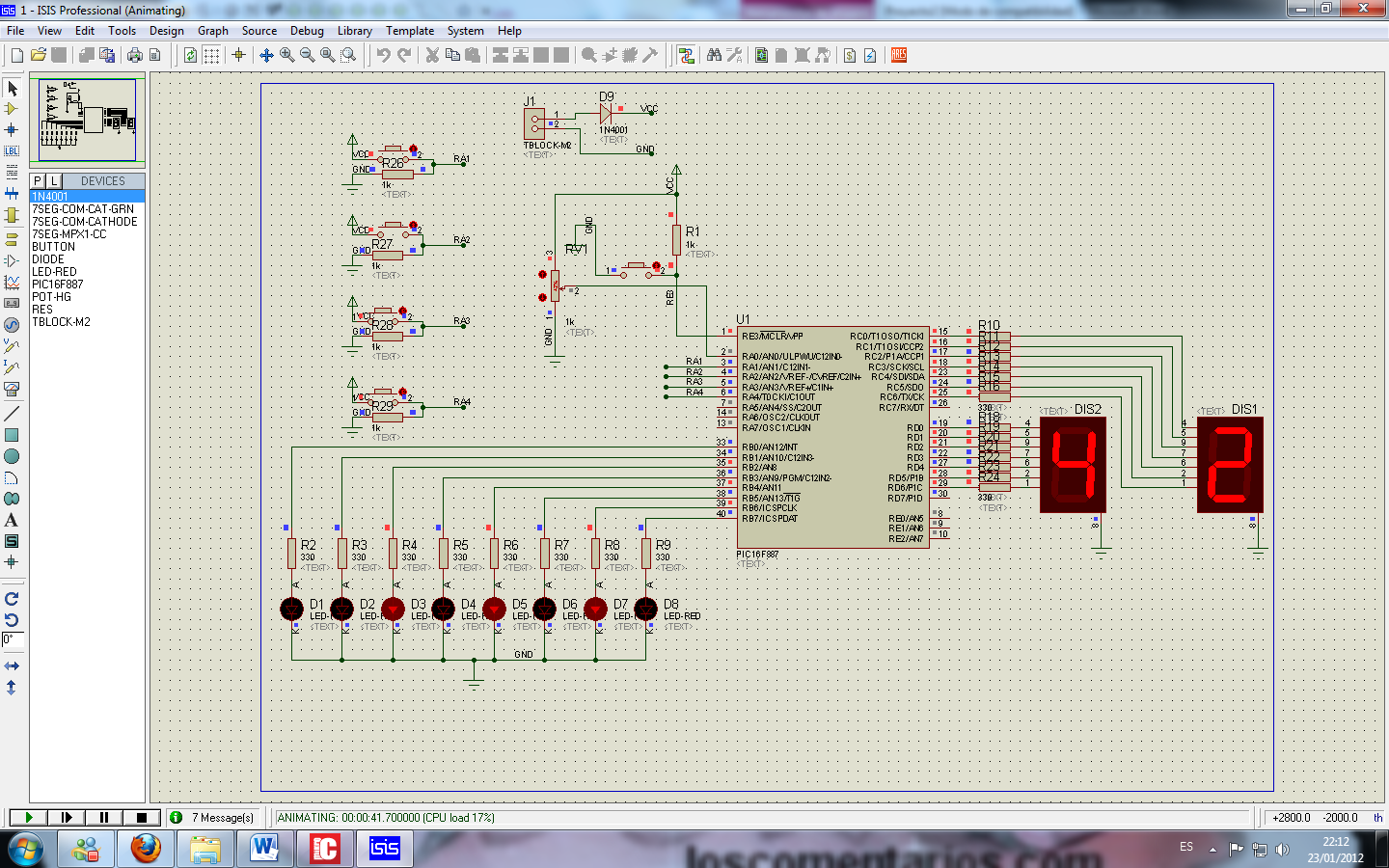
}

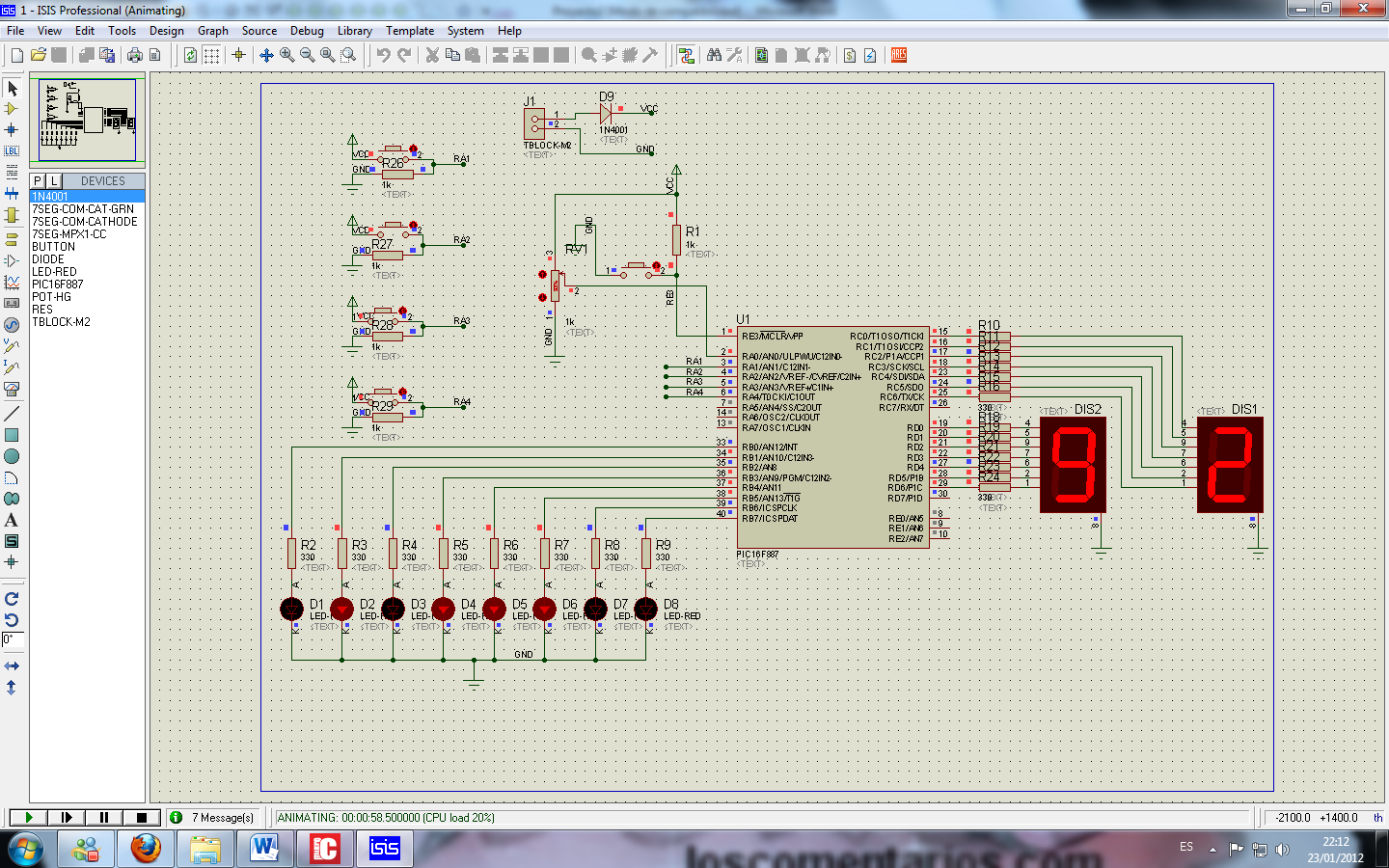
**7.- Copia impresa del circuito armado en PROTEUS para la simulación en el momento de su ejecución**

**Al encenderse el circuito aparece mis iniciales en los displays de 7 segmentos.**

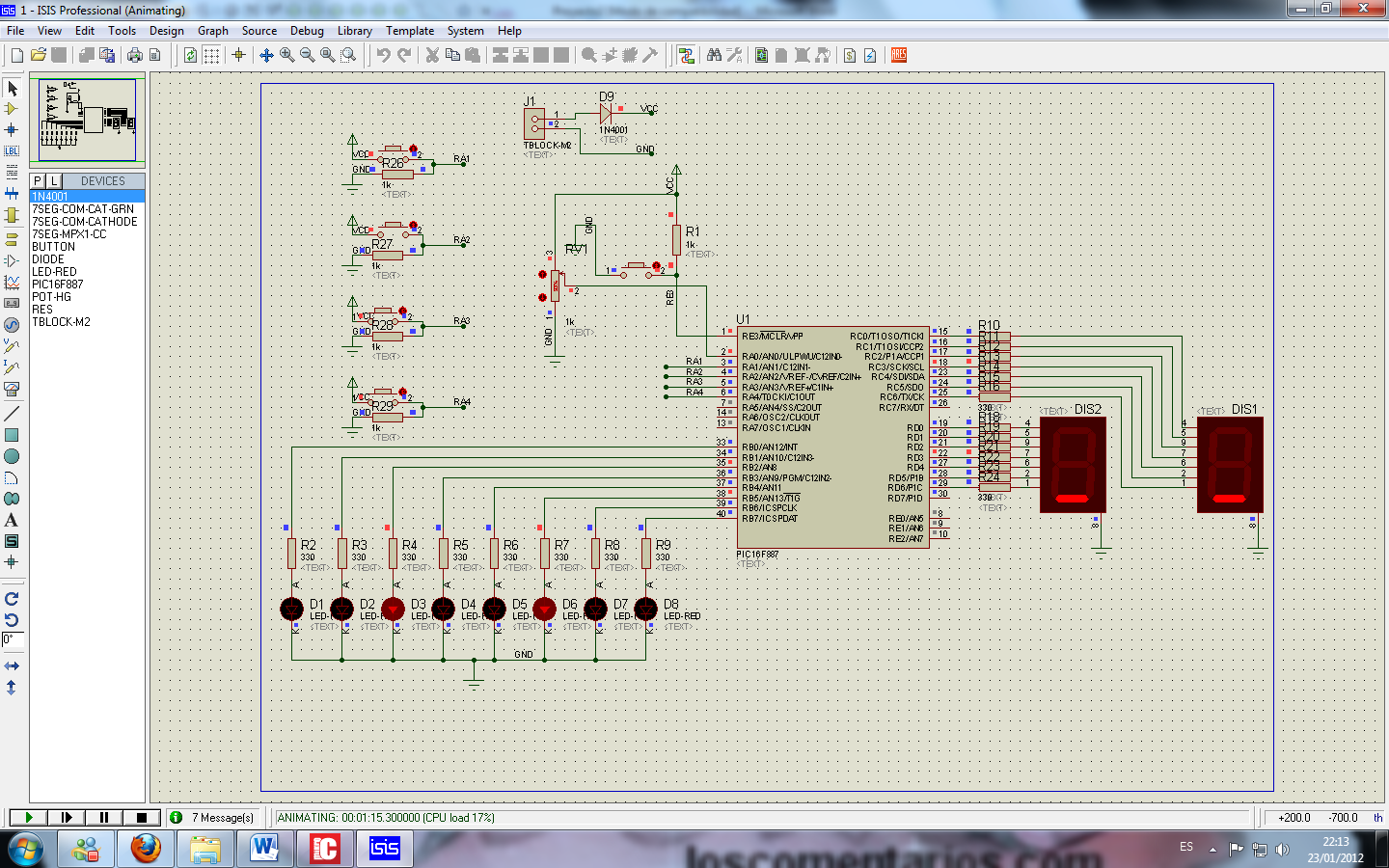


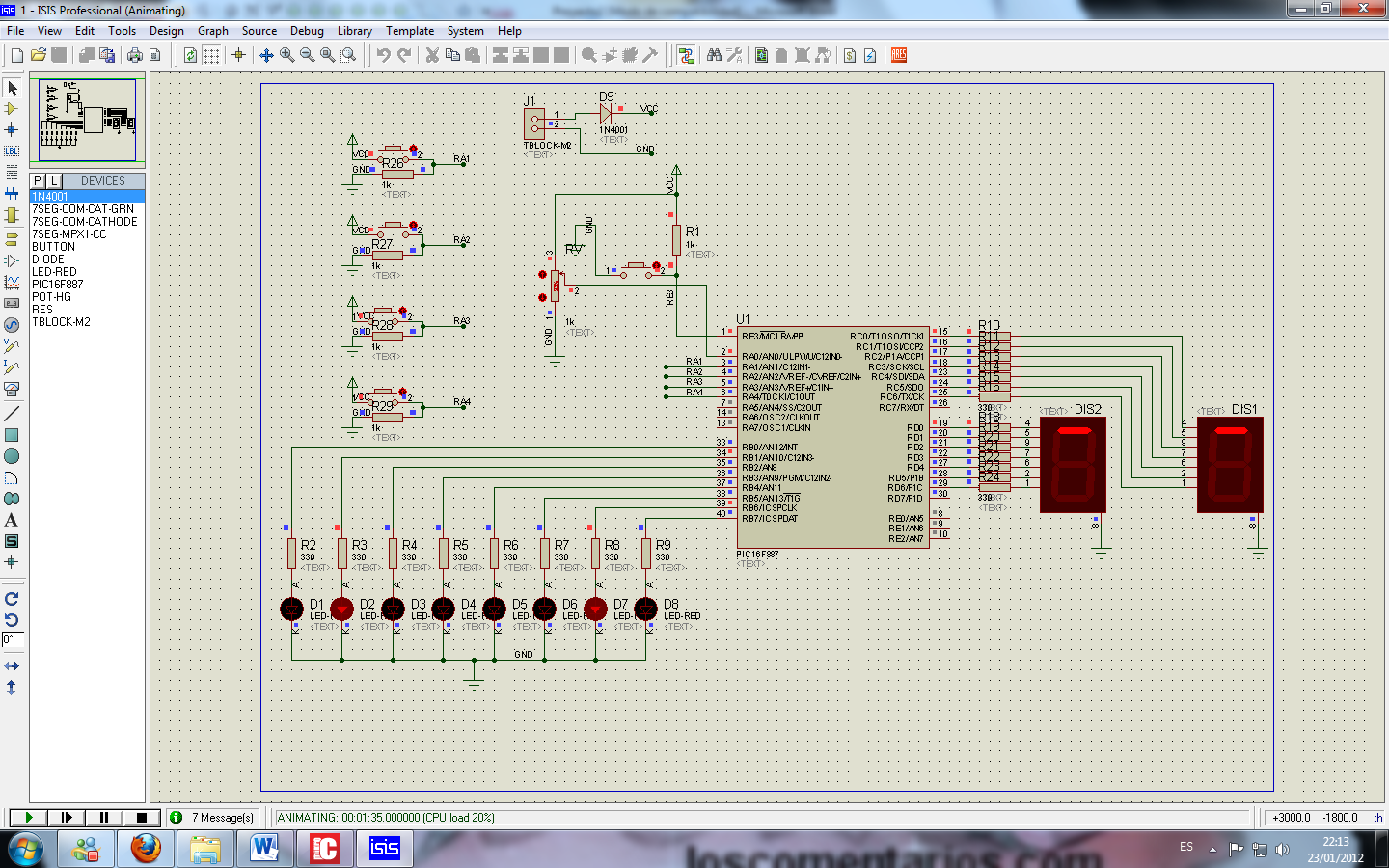
**Leyendo los valores analógicos de un potenciómetro colocado en la bornera de entrada analógica, presentando las variaciones de valores (en binario) en 8 LEDs y en dos DISPLAYs de 7 segmentos (en dos dígitos decimales) disponibles**

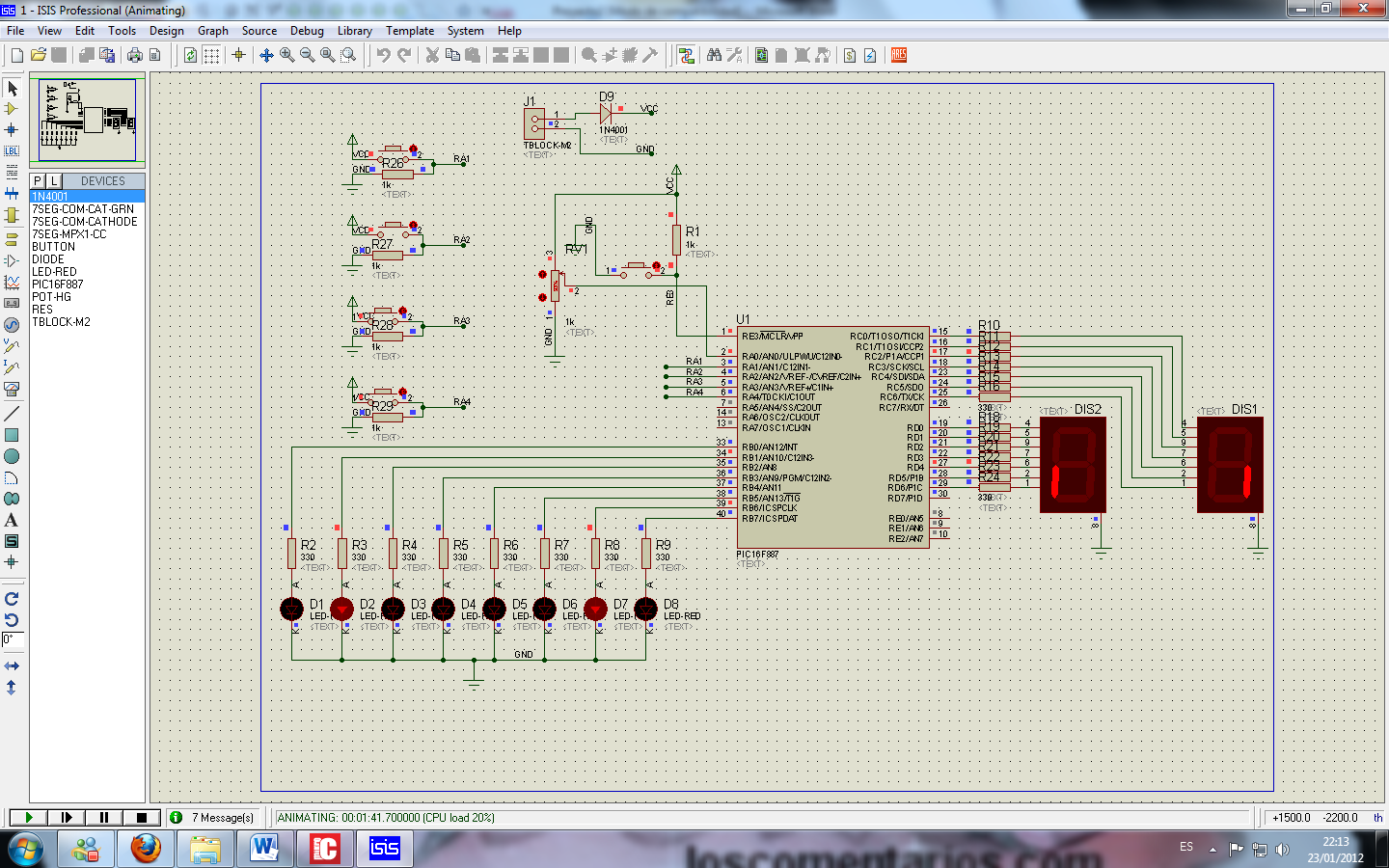




**LEDs Psicodélicos haciendo uso de los recursos de la tarjeta prototipo que consta de cuatro botoneras, 8 LEDs y dos DISPLAYs de 7 segmentos. Si se presiona la BOTONERA1 se selecciona la velocidad mas baja de movimiento de los LEDs en los Displays. Si se presiona la Siguiente Botonera la velocidad aumenta y así sucesivamente. Se dispone en total 4 tipos de velocidades**

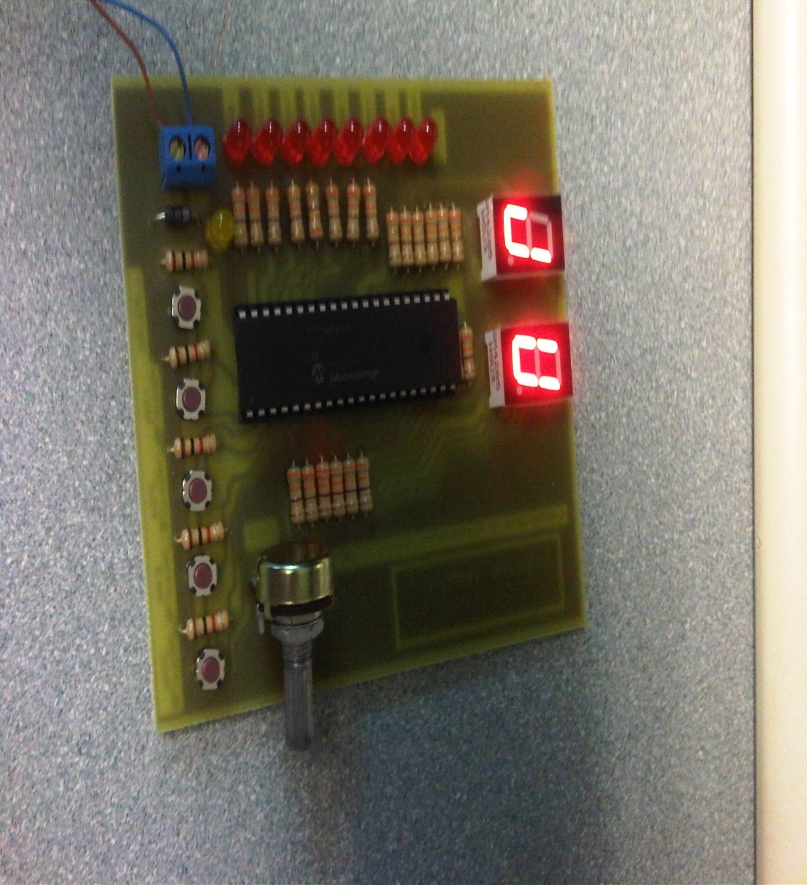


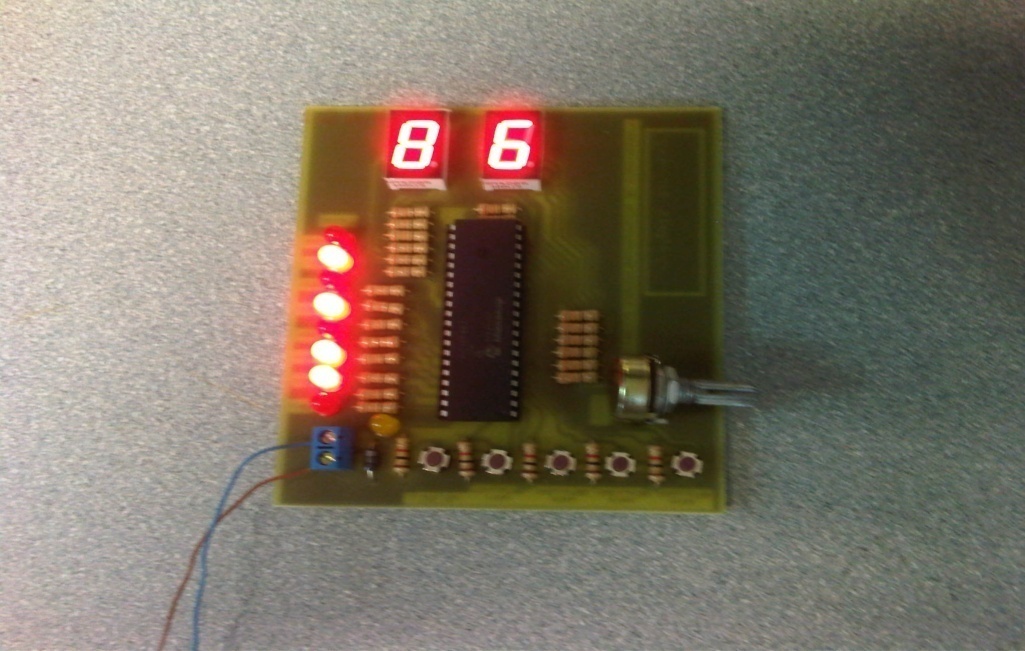


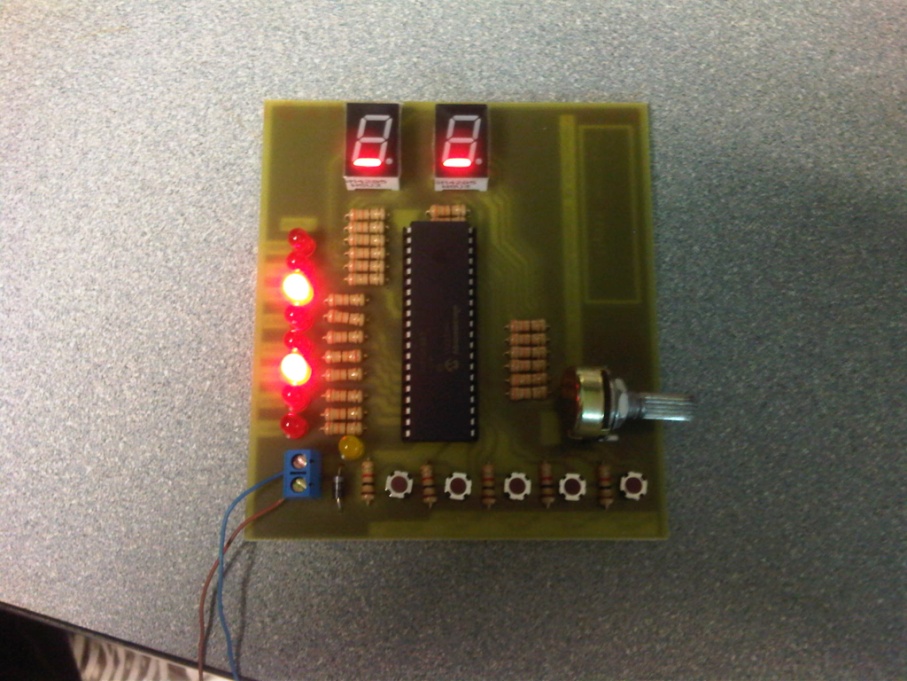


**8.- Circuito funcionando:**

* **Al encenderse el circuito aparece mis iniciales en los displays de 7 segmentos.**







**9.- Conclusiones**

* **En el primer proyecto de leer un un valor analógico del potenciómetro conectado en el PORTA PIN 0, para la programación en mickro C utilizamos el modulo ADC, para lo cual realizamos con la siguiente instrucción dato=ADC\_Read(0); la cual nos ayuda a obtener valores analógicos y lo convierte en digital, facilitándonos así para luego mostrar en 8 leds .**
* **Para el segundo proyecto la estrategia fue usar un patrón para los giros que serán mostrados en los displays y un patrón de leds para mostrar en el puerto ; también a medida que presionamos las diferentes botoneras incrementamos la velocidad de giro para lo cual utilizamos sus respectivos delays con menor retardo en caso d obtener un giro más rápido.**
* **Después de haber realizado estos dos proyectos el uso de la instrucción ADC\_Read nos ayudo para leer los datos analógicos del pot, también la ayuda que proporciona mickro C fue de gran apoyo para la programación; también para realizar la programación de c hicimos uso de los conocimientos adquiridos en clase.**

**10.- Recomendaciones**

* **Antes de soldar la placa de proyecto debemos observar las pistas del circuito probar continuidad, también cuando vayamos soldando cada elemento ir verificando si hay continuidad y una estrategia para que el punto de soldadura sea precisa debemos utilizar el estaño de 0.5mm, por ultimo al fin de soldado limpiar la placa con un limpiador electrónico de placa, para así obtener un mejor resultado.**
* **En el momento de grabar el pic16F887 tener mucho cuidado al colocar y sacar del grabador de Pics ya que sus pines pueden romperse y también recordar que hay que utilizar en configuración de bits el oscilador interno ( INTOSCIO oscillator:I/O function)y verificar que code protect este en off, CPD disabled.**
* **Para cualquier inquietud, conocimiento general y facilidad de programación, Mickro C nos facilita la herramienta de ayuda en HELP, también revisar los apuntes de clase y las anteriores prácticas realizadas en el laboratorio de microcontroladores son de muy buen apoyo para poder realizar un proyecto con éxito.**