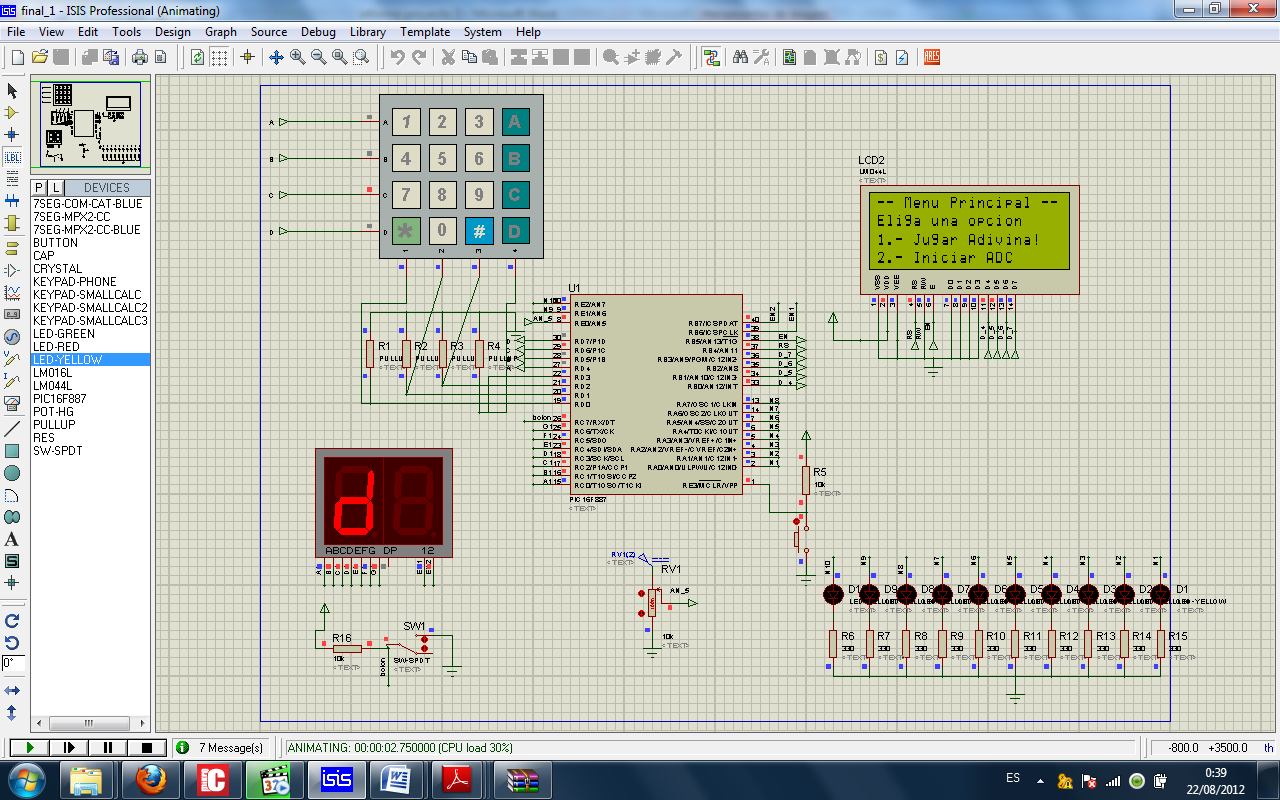
|  |
| --- |
| **Proyecto 3 de Laboratorio de Microcontroladores:** |
|  |
| Alumno: David Vaca B.  Paralelo: 9 |
|  |
|  |
|  |
|  |
| **23/08/2012** |





**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FIEC**

**Laboratorio de Microcontroladores**

**Proyecto #1**

CONTADOR DE CAJAS Y UNIDADES

**Alumno**:

David Vaca B.

**Paralelo #** **9**

**Grupo: # 3**

**Fecha de presentación:**

11 Julio 2012

**2012 – 1° TÉRMINO**

**OBJETIVO:**

* Diseñar un programa basado en lenguaje C, que interactúe con el usuario, mediante el uso de pantallas LCD y teclado 4x4.

**1.- Enunciado del proyecto**

Para este proyecto, se utilizara el PIC 16F887 para, mediante una pantalla LCD de 20x4, mostrar un menú, donde se pueda elegir entre el uso del ADC para lectura de números en decimal y binario, o iniciar una juego “adivinador de números”. En la aplicación del ADC, se mostrará en pantalla el valor en binario y en decimal, leído por medio de un potenciómetro, mientras que para el juego, el sistema proveerá de 10 puntos, y generará un número aleatorio entre 1 a 9, tal que el usuario tiene que adivinar, mediante ingreso por teclado 4x4, si acaso se llegará a adivinar, se le darán 5 ptos al usuario, mientras si se equivoca, resta uno, cuando el usuario se quede sin puntos, se mostrará un mensaje de fin del juego, para esto se recurrirá a funciones implementadas en MikroC Pro.

**2.- Diagrama de Bloques**



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

***Programa Principal:***



***Subrutinas***





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

1. En este programa utilizaremos el oscilador interno de 4MHz.
2. Se debe configurar el pin AN5 como entrada analógica, mientras el resto digital, RE0,PORTD,RC7 como entradas, RE1, RE2, RC6-RC0,PORTA, PORTB como salidas.
3. Se imprime el menú en pantalla, y queda en un lazo tal que la única forma de salir, es cuando se presione un tecla, para enviar este valor a una función donde la convierte, en un número del 1 al 10, si el valor es 1 se ejecuta el lazo de control para iniciar el juego, si es 2 se inicia el lazo para la ejecución del ADC, caso contrario se la detecta como tecla invalida, así mismo, mediante multiplexación se muestra las iniciales de nombre y apellido en un display de 7 segmentos.
4. Si el valor de key es 1, entonces se carga la variable puntos con 10, y se empieza a generar números aleatorios, mientras no se presione una tecla, para que luego de esto, se compara, el valor ingresado con el número aleatorio generado.
5. Si ambos números son iguales entonces se le da al usuario 5 ptos, y se muestra el mensaje “UD. ADIVINO” y muestra el puntaje global.
6. Si los números no fueron iguales, entonces se le resta un punto al usuario, y se muestra en pantalla “UD. NO ADIVINO”, y luego de un retardo de tiempo para ambos casos, se vuelve a generar otro aleatorio, salvo que el valor de puntos sea menor igual a 0, en tal caso entra a otro lazo de control donde indica que no hay puntos, y se regresa al menú principal.
7. Si acaso el valor de la tecla presionada es 2, se inicia el ADC mostrándose en pantalla LCD el valor en binario y decimal de los detectado, más el valor binario en LEDs indicadores.

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

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; PROYECTO DE LAB. DE MICROCONTROLADORES

; FIEC-ESPOL

; ADC Y ADIVINADOR DE NUMEROS

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; NOMBRE: FINAL1.C

; FECHA: 11/jul/2010

; VERSION: 1.00

; PROGRAMADOR: David Vaca B.

; PARALELO: 9

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

// Keypad module connections

char keypadPort at PORTD;

// LCD module connections

sbit LCD\_RS at RB4\_bit;

sbit LCD\_EN at RB5\_bit;

sbit LCD\_D4 at RB0\_bit;

sbit LCD\_D5 at RB1\_bit;

sbit LCD\_D6 at RB2\_bit;

sbit LCD\_D7 at RB3\_bit;

sbit LCD\_RS\_Direction at TRISB4\_bit;

sbit LCD\_EN\_Direction at TRISB5\_bit;

sbit LCD\_D4\_Direction at TRISB0\_bit;

sbit LCD\_D5\_Direction at TRISB1\_bit;

sbit LCD\_D6\_Direction at TRISB2\_bit;

sbit LCD\_D7\_Direction at TRISB3\_bit;

unsigned int temp\_res;

int key,cnt=0,puntos=10,pot,temp,i;

char txt[7];

void interrupt() {

//TMR0 = 227; // Timer TMR0 is returned its initial value

//INTCON = 0x20; // Bit T0IE is set, bit T0IF is cleared

TMR0 = 233; // Timer TMR0 is returned its initial value

INTCON = 0x20;

asm{BSF STATUS,7}

PORTB.F6=~PORTB.F6;

PORTB.F7=~PORTB.F7;

if(PORTB.F6==1 && PORTB.F7==0)

{

PORTC.F0=0;

PORTC.F1=1;

PORTC.F2=1;

PORTC.F3=1;

PORTC.F4=1;

PORTC.F5=1;

PORTC.F6=0;

}

else

{

PORTC.F0=0;

PORTC.F1=1;

PORTC.F2=1;

PORTC.F3=1;

PORTC.F4=1;

PORTC.F5=0;

PORTC.F6=1;

}

}

char teclado(char key){

switch(key){

case 1: return 1; break; // 1 // Uncomment this block for keypad4x4

case 2: return 2; break; // 2

case 3: return 3; break; // 3

case 4: return 10; break; // A

case 5: return 4; break; // 4

case 6: return 5; break; // 5

case 7: return 6; break; // 6

case 8: return 10; break; // B

case 9: return 7; break; // 7

case 10: return 8; break; // 8

case 11: return 9; break; // 9

case 12: return 10; break; // C

case 13: return 10; break; // \*

case 14: return 10; break; // 0

case 15: return 10; break; // # = 6; break; // 6

case 16: return 10; break; // D

}

}

void main() {

ANSEL=0x20; // Configure AN5 pin as analog

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

C1ON\_bit=0; // Disable comparators

C2ON\_bit=0;

Lcd\_Init();

asm{BCF STATUS,7}

ADC\_Init();

Keypad\_Init();

TRISB=0; // PORTD is output

/\*asm{

bsf PORTB,7

bcf PORTB,6

} \*/

//TRISD=0XFF;

TRISE=0b1001;

TRISA=0;

TRISC=0x80;

PORTA=0;

PORTC.F0=0;

PORTC.F1=1;

PORTC.F2=1;

PORTC.F3=1;

PORTC.F4=1;

PORTC.F5=0;

PORTC.F6=1;

PORTB.F6=0;

PORTB.F7=1 ;

Lcd\_Cmd(\_LCD\_CLEAR); // Clear display

Lcd\_Cmd(\_LCD\_CURSOR\_OFF); // Cursor off

temp\_res=0;

do{

OPTION\_REG = 0x84; // Prescaler is assigned to timer TMR0

TMR0 = 200; // Timer T0 counts from to 255

INTCON = 0xA0; // Enable interrupt TMR0

PORTB.F6=0;

PORTB.F7=1 ;

Lcd\_Cmd(\_LCD\_CLEAR);

//delay\_ms(5);

asm{BcF STATUS,7}

Lcd\_Out(1, 1, "-- Menu Principal --");

asm{BcF STATUS,7}

Lcd\_Out(2, 1, "Eliga una opcion");

Lcd\_Out(3, 1, "1.- Jugar Adivina! ");

asm{BcF STATUS,7}

Lcd\_Out(4, 1, "2.- Iniciar ADC"); asm{BcF STATUS,7}

do{

key=keypad\_key\_click();

}while(!key);

Lcd\_Cmd(\_LCD\_CLEAR);

INTCON=0;

PORTB.F6=1;

PORTB.F7=1 ;

key=teclado(key);

PORTC=0x00;

if(key==1)

{

//key=0;

cnt=1;

puntos=10;

do{

Lcd\_Cmd(\_LCD\_CLEAR);

asm{BCF STATUS,7}

Lcd\_Out(2,1,"Adivina el Numero!"); asm{BCF STATUS,7}

Lcd\_Out(3,1,"ingresa tu eleccion"); asm{BSF STATUS,7}

do{

key=Keypad\_Key\_Click(); // Store key code in key variable

if(cnt==9)

{

cnt=1;

}

else

cnt++;

//WordToStr(cnt,txt);

//Lcd\_Out(4,8,txt);

} while (!key);

key=teclado(key);

if(key==1)//AQUI VENDRA cnt Y NO 1

{

Lcd\_Cmd(\_LCD\_CLEAR);

puntos=puntos+5;

intToStr(puntos,txt);

asm{BcF STATUS,7}

Lcd\_Out(3,1,"Total Puntos: "); asm{BsF STATUS,7}

lcd\_out\_cp(txt); asm{BcF STATUS,7}

lcd\_out(1,1,"Ud. acertoo!");asm{BSF STATUS,7}

delay\_ms(1500);

}

else

{

Lcd\_Cmd(\_LCD\_CLEAR);

puntos--;

wordToStr(puntos,txt);

asm{BsF STATUS,7}

Lcd\_Out(3,1,"Total Puntos: ");

asm{BsF STATUS,7}

lcd\_out\_cp(txt); asm{BsF STATUS,7}

lcd\_out(1,1,"Ud. NO acertoo!");

delay\_ms(1500);

if(puntos<=0)

{

asm{BsF STATUS,7}

Lcd\_Cmd(\_LCD\_CLEAR);

asm{BsF STATUS,7}

Lcd\_Out(3,1,"El juego termino"); asm{BsF STATUS,7}

Lcd\_Out(2,1,"Puntos Agotados");

delay\_ms(2500);

lcd\_Cmd(\_LCD\_CLEAR);

key=0;

}

}

}while(puntos>0);

}

else

{

if(key==2)

{

PORTC=0x00;

Lcd\_Cmd(\_LCD\_CLEAR);

//while(portc.f7==1)

//while(portc.f7==0);

do {

asm{BsF STATUS,7}

lcd\_cmd(\_LCD\_CLEAR); asm{BSF STATUS,7}

Lcd\_Out(2, 1,"Numero Binario ADC"); asm{BsF STATUS,7}

Lcd\_Out(3, 1,"Rep. Decimal:");

//while(portc.f7==1)

//{}

//while(portc.f7==0)

//{}

//delay\_ms(200);

temp\_res = ADC\_Read(5); // Get 10-bit results of AD conversion

PORTA = temp\_res; // Send lower 8 bits to PORTA

if( (temp\_res>=256 && temp\_res<=511)|| (temp\_res>=768 && temp\_res<=1023))

PORTE.F1 = 1; // Send 2 most significant bits to RC1, RC0

else

PORTE.F1 = 0;

if( (temp\_res>=512 && temp\_res<=1023))

PORTE.F2 = 1; // Send 2 most significant bits to RC1, RC0

else

PORTE.F2 = 0;

WordToStr(temp\_res,txt);

Lcd\_Out\_cp(txt);

LCD\_CMD(\_LCD\_RETURN\_HOME);

temp=temp\_res;

for(i=0;i<11;i++)

{

pot=(temp%2);

IntTostr(pot,txt); asm{BsF STATUS,7}

Lcd\_Out(4,(14-i),txt); asm{BsF STATUS,7}

temp=temp/2;

}

} while(portc.f7==1);

while(portc.f7==0){}

temp\_res=0;

PORTA=0;

Lcd\_Cmd(\_LCD\_CLEAR);

}

else

{

LCD\_CMD(\_LCD\_CLEAR);

delay\_ms(5);

asm{BCF STATUS,7}

Lcd\_Out(2,1,"ingreso invalido"); asm{BSF STATUS,7}

delay\_ms(1000);

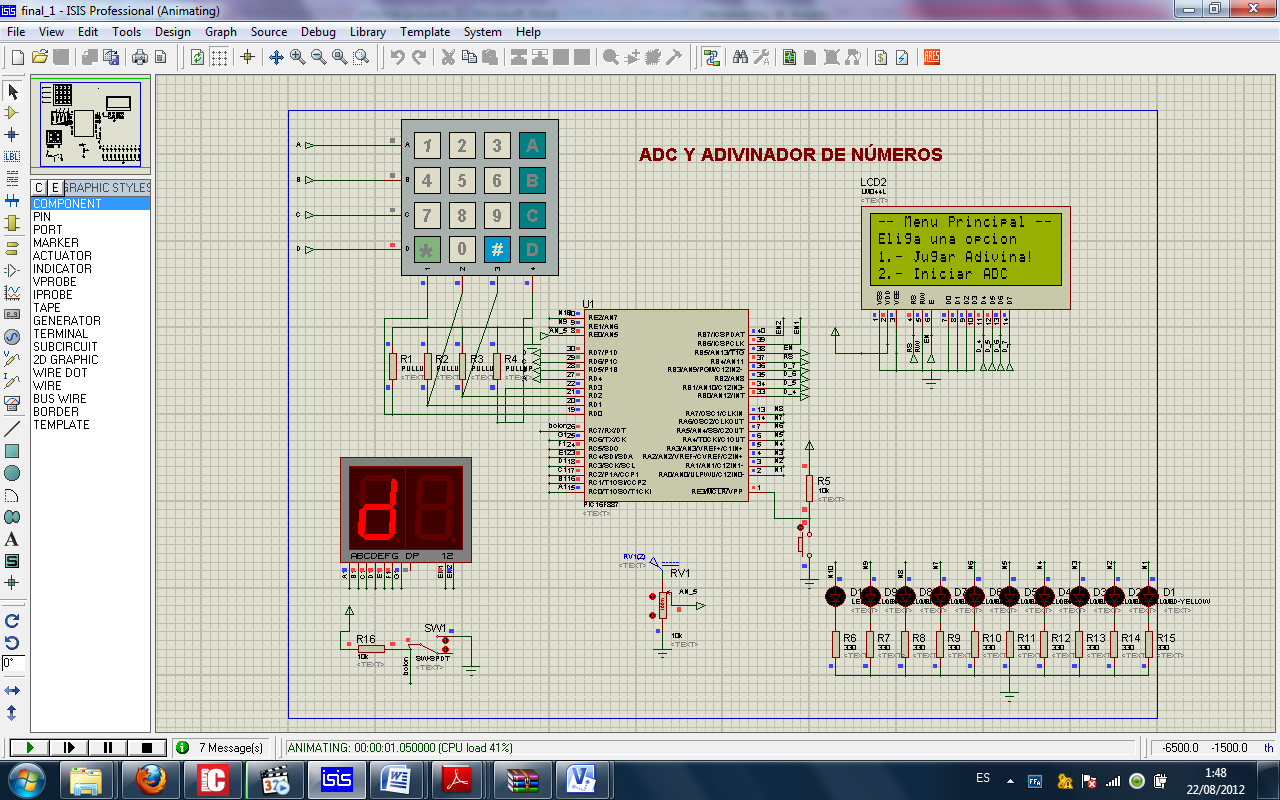
}

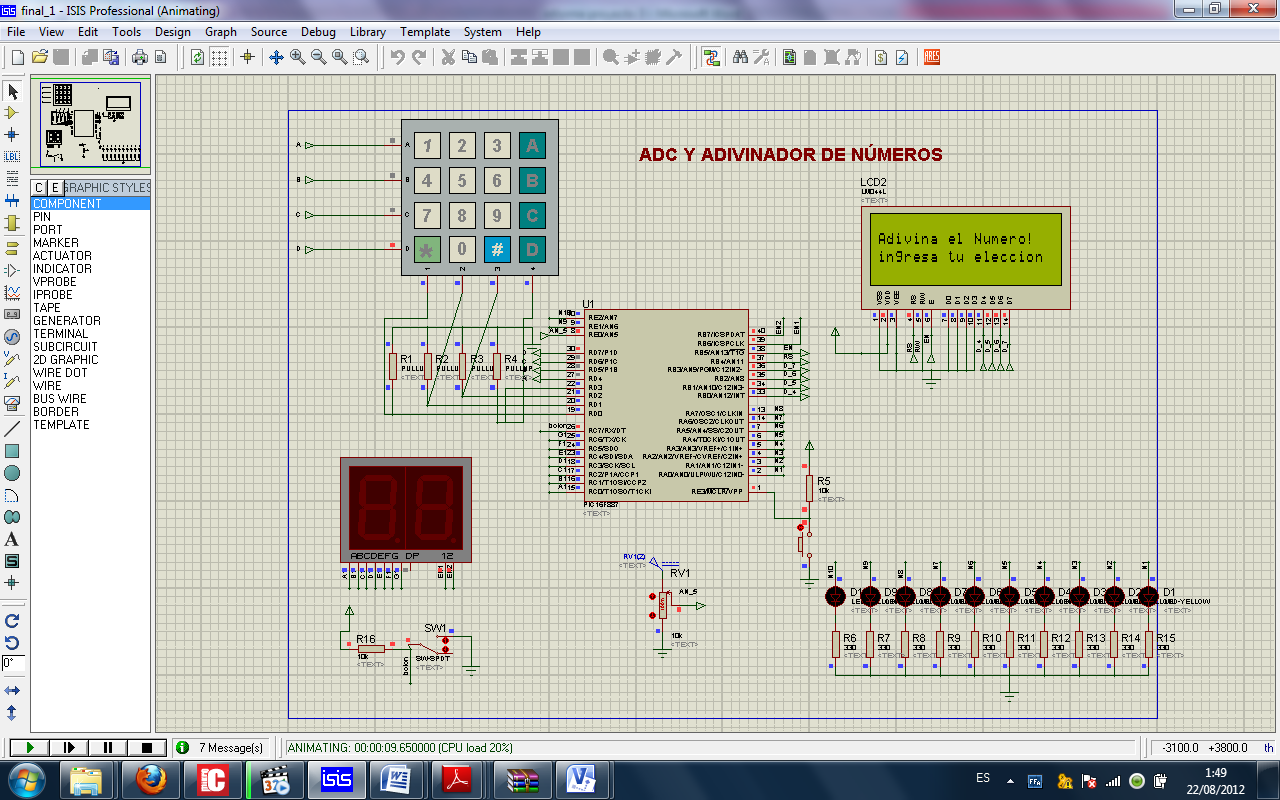
}

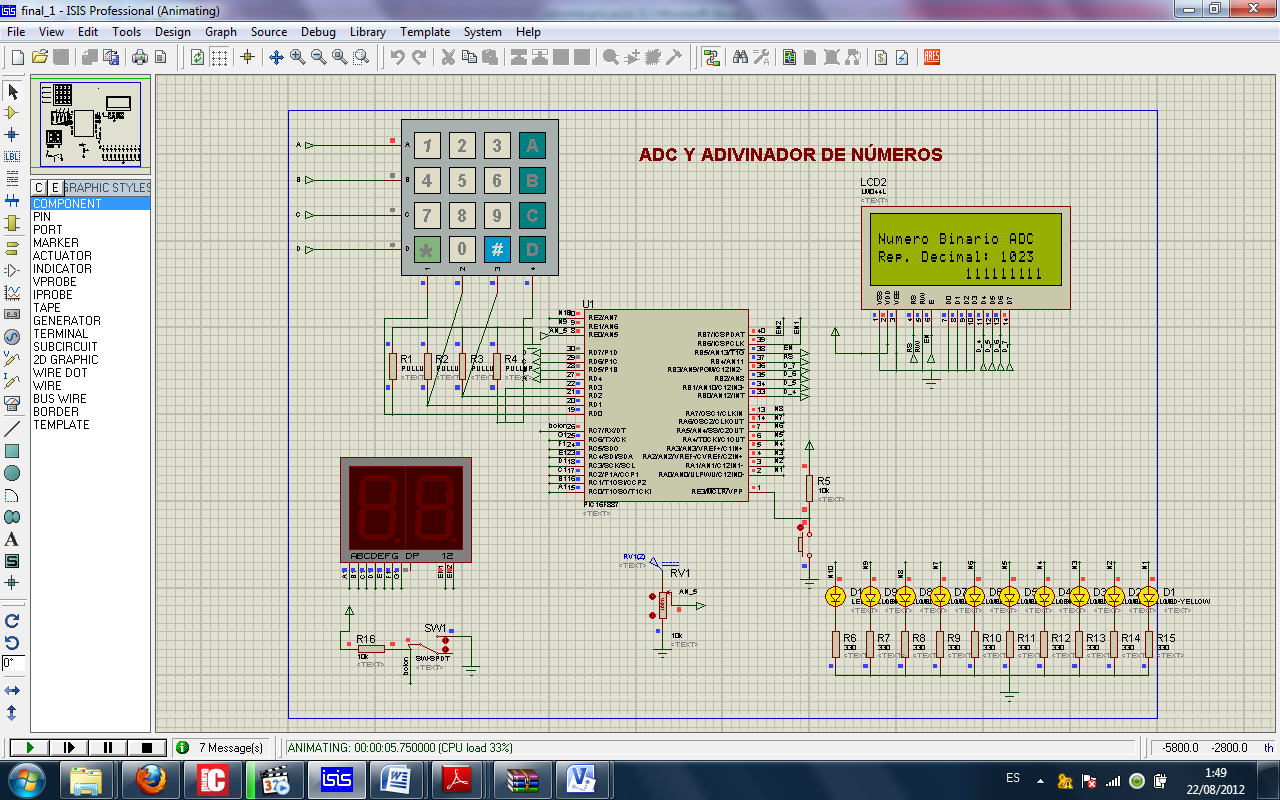
}while(1);

}

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

****

****

****

**7.- Conclusiones**

* Se observó que fue posible mostrar en una pantalla LCD mensajes y valores de variables que internamente usa el PIC para realizar operaciones, los cuales permiten un interfaz amigable y elegante con el usuario, el cual puede hacer uso de menús, así mismo con la interacción con el teclado, el cual sirvió como fuente de información para comparación y validación de funciones internas del microcontrolador
* Fue posible hacer uso correcto, e implementación en este programa de funciones en Mikro C, siendo estas ya creadas para uso y gestión de teclado y pantalla LCD, así mismo de la obtención de datos numéricos por medio de un potenciómetro, es decir datos analógicos para convertirlos en valores digitales, los cuales si pueden ser interpretados por el microcontrolador, y con las cuales se puede realizar operaciones internas.
* Se pudo comprobar, la mayor facilidad para implementar programas, mediante lenguaje se, que incorpora las estructuras de control ya conocidas, junto con funciones implementadas, como ya se menciono antes, las cuales permiten hacer uso de las capacidades del PIC, fácilmente, sin trabajar conociendo directamente la arquitectura interna del chip, como era lo requerido para poder manejar el lenguaje de ensamblador

.

* Cabe hacer notar que, a pesar que se está trabajando en lenguaje C, se puede escribir en lenguaje ensamblador cuando uno lo desee, mediante una instrucción implementada en el programa de mikroC, la cual encierra entre corchetes las líneas de código que deseemos escribir en este otro lenguaje, permitiendo en ocasiones ahorrar recursos del PIC, como los son memorias o registros.

**8.- Recomendaciones**

* Se debe tomar en cuenta que para poder hacer el correcto uso de TMR0, no hay que olvidar encerar la bandera, puesto que a pesar que se está trabajando en otro lenguaje de programación, las características del PIC, no hay cambiado.
* Para imprimir, valores de variables internas del PIC en una pantalla LCD, es necesario realizar la conversión respectiva del dato en un arreglo de caracteres, el cual si es entendido por esta pantalla, lo que permite imprimir sobre ella, lo que deseamos.
* Se debe recordar que, a pesar que se esta trabajando con un lenguaje que permite muchísimas aplicaciones, no se debe perder de vista, las capacidades limitantes del PIC como lo son las memorias RAM y ROM, ya que sobrepasarnos de su capacidad, sería in impedimento para poder cargar el programa sobre ellas.