# big8051_01**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERIA EN**

**EN ELECTRICIDAD Y COMPUTACIÓN**

**FIEC**

**PROYECTO#2 DE LABORATORIO DE MICROCONTROLADORES**

**JUEGO DE OPERACIONES MATEMATICAS Y lectura de valores analógicos en re0**

**PROFESOR:**

**DR. CARLOS VALDIVIESO**

**REALIZADO por:**

***Boris puero***

**Paralelo:** *8*

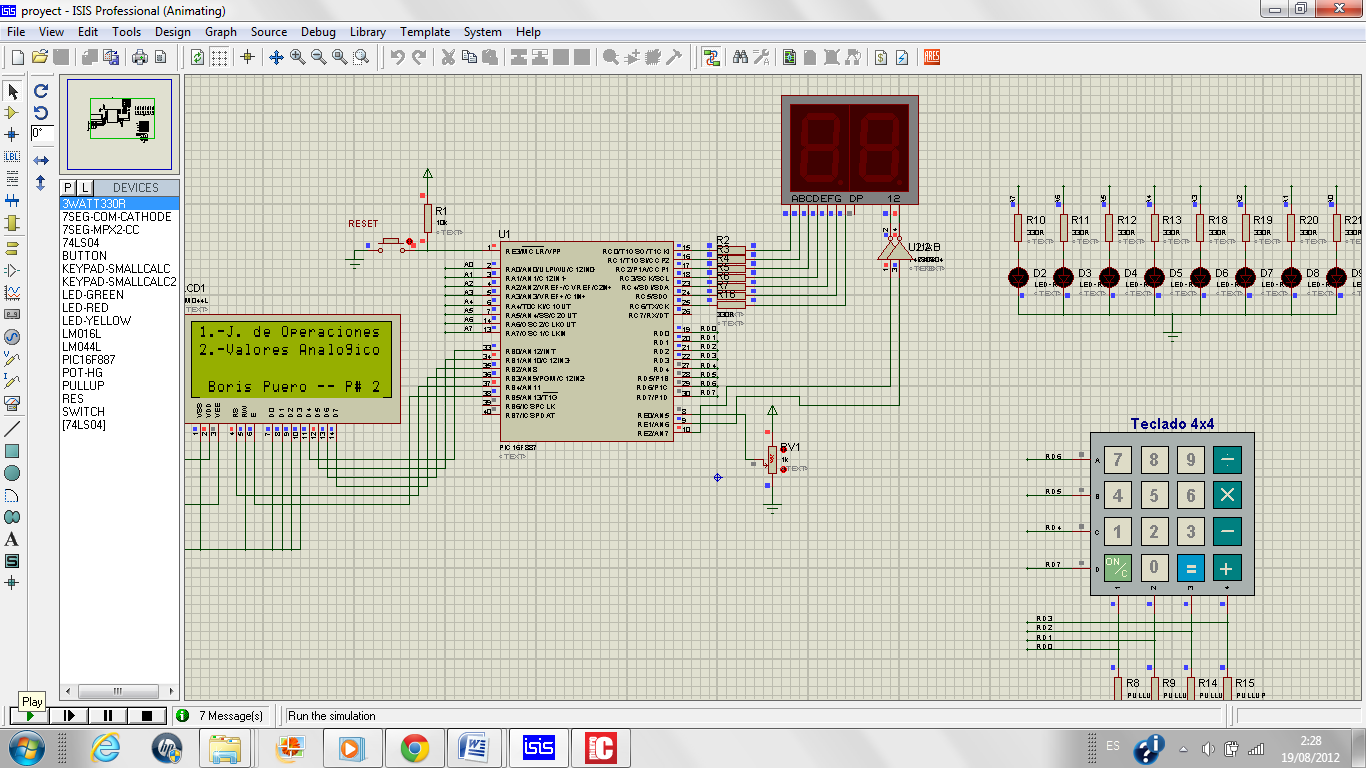
**Grupo:** *1*

***Agosto 20, 2012***

**1.- Especificaciones Técnicas del Proyecto**

**Juego de Operaciones Matemáticas**

El proyecto consta de un menú en donde se puede elegir la acción a realizar sea esta realizar el juego de operaciones matemáticas o realizar la lectura de valores analógicos en RE0.

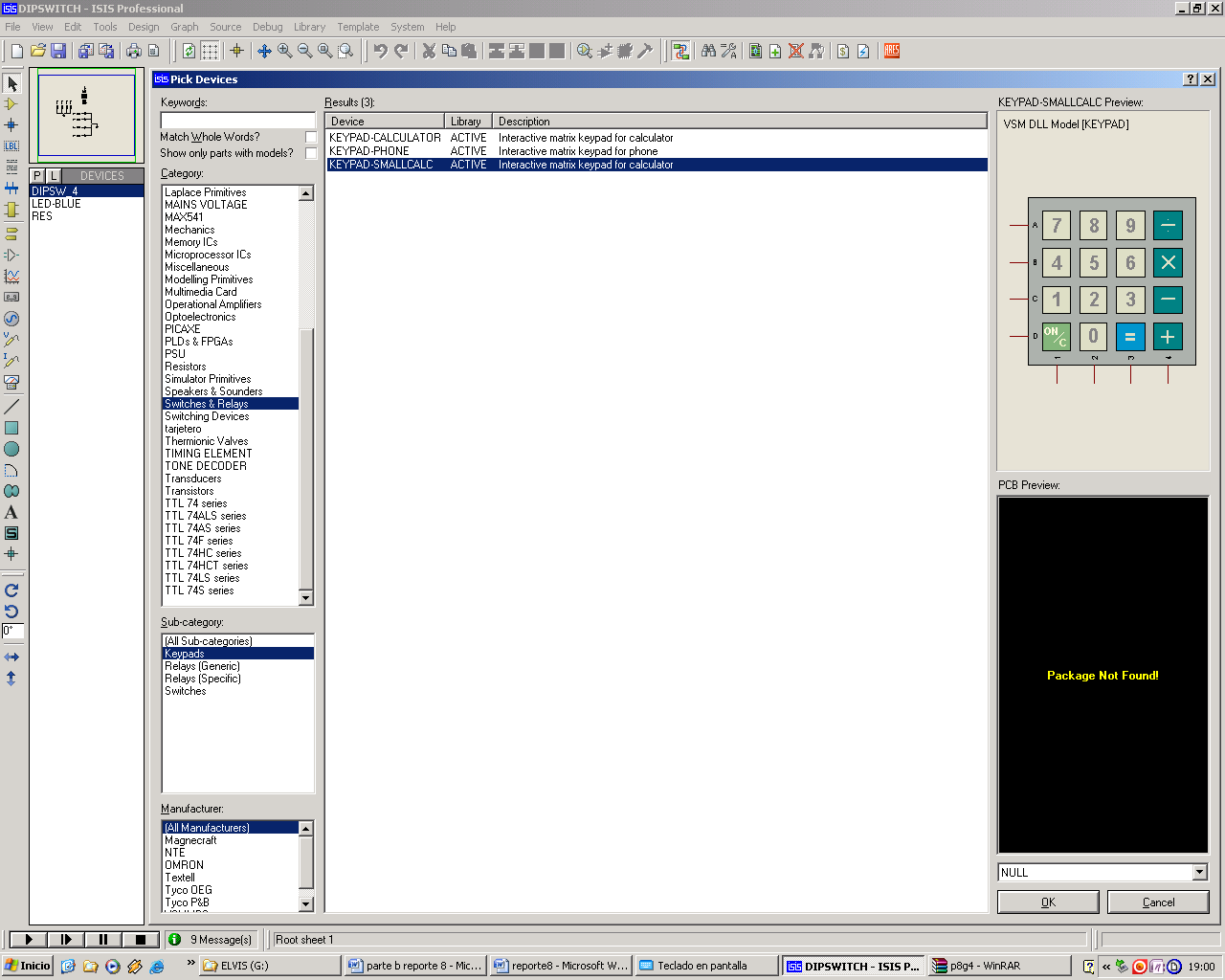


Si presionamos la tecla 1 iremos al juego de operaciones matemáticas en la que el usuario puede elegir entre suma, resta y multiplicación pero antes de esto se debe presionar el enable del teclado que será la tecla ON/C que al ser presionada genera un numero aleatorio entre 0 y 20 el mismo que se guardará en la variable cnt, luego seleccionaremos el tipo de operación que se va a jugar (ya sea suma , resta o multiplicación) y se genera otro numero q se guarda en la variable cnt\_2 y realiza la operación respectiva sea suma, resta o multiplicación y lo que debe hacer el usuario es por medio del teclado presionar la respuesta respectiva de la operación realizada y por cada acierto este va contándolos y mostrándolos en los leds del PORTA, luego de cada operación acertada se genera otra operación hasta que el usuario falle; una vez que ha fallado muestra los mensajes **¡Perdiste! y Numero Incorrecto** y luego de un retardo de 2 segundos vuelve al menú inicial donde se puede volver a realizar el juego o puedo ir a la simulación de la lectura de valores analógicos en la patita RE0l.

Si presionamos la tecla 2 iremos a la lectura de los valores analógicos en la patita RE0, aquí haremos la representación binaria y decimal de un potenciómetro colocado en la patita RE0 en una LCD, para esto se hizo uso de la instrucción ANSEL=0x01 que me configura la patita RE0 como entrada analógica y además seteamos el ADCON1=0x80 que es el registro de control A/D y esto hace que me lea el resultado justificado a la derecha, en cuanto a la función realizada en software, esta se encuentra en un lazo infinito mientras no se presione una tecla del teclado 4x4, en este lazo se está sensando la patita RE0 y nos está mostrando su valor binario y decimal en la LCD, esto se hace gracias a un algoritmo creado para que solo nos bote el valor binario y en cuanto al decimal pues el ADC lo reconoce como un decimal de 1023 ya que son 10 bits que usa el ADC; así mismo si se presiona la tecla (=) se sale del lazo regresando al estado inicial donde se preguntara por una acción a realizar sea el juego de operaciones matemáticas o volver a realizar la lectura de RE0.

Nota: El programa siempre está mostrando en dos display’s de 7 segmentos multiplexados las iniciales de mi nombre por medio de una interrupción por TIMER0, esto se lo ha realizado debido a los requerimientos del proyecto.

**2.- Diagrama de Bloques**

****







**PIC16F887**



**3.- Diagrama de flujo del programa principal**



**4.- Algoritmos a usar para el proyecto**

* Keypad module connections
* LCD module connections
* Inicio **declaración de variables**
* Prototipo de funciones
* Seteo de Puertos
* Inicializamos el Keypad, Lcd y el UART.
* Presentación del menú.
* Estado donde se presiona ‘1’ para jugar o ‘2’ para leer valor en RE0.
* Habilito y selecciono el tipo de Operación en caso de haber presionado ‘1’.
* Procedemos a jugar y visualizar resultados.
* Retorno a estado inicial.
* En caso de haber presionado ‘2’.
* Estado donde estamos sensando el valor en la patita RE0.
* Si presiono ’=’ regreso a estado inicial.

**5.- Programa Fuente en Lenguaje C (todo)**

**/\***

**\* Nombre del Proyecto:**

**Proyecto\_2.c**

**\* Nombre del Autor:**

**Boris Puero**

**\* Description:**

**El ejercicio consta de 2 partes, donde la primera es un Juego de operaciones matemáticas y en la segunda parte consiste en la lectura de valores analógicos de un potenciómetro en el pin RE0**

**\* Test configuration:**

**MCU: PIC16F887**

**Oscillator: HS, 08.0000 MHz**

**SW: mikroC PRO for PIC**

**\*/// Keypad module connections**

**char keypadPort at PORTD;**

**#include "mask.c"**

**// End Keypad module connections**

**// 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;**

**// End LCD module connections**

**//variables**

**unsigned char ch;**

**char \*text,\*text1,\*text2,\*text3;**

**char txt3[14] = "";**

**char txt4[14] = "";**

**char txt1[]="1.-J. de Operaciones";**

**char txt2[]="2.-Valores Analogico" ;**

**char nombre[] = " BORIS PUERO ";**

**char txt5[] = "BORIS PUERO ";**

**long arr[10], temp1[9];**

**long adc,tlong,maximo,uni,dec;**

**int i=0,cnt=0,cnt\_2=0,ans=0,wins=0,band=0;**

**int j,name2, name1,dig\_no;**

**unsigned short kp;**

**//Funciones**

**void juego();**

**void valor();**

**void salto();**

**void interrupt() ;**

**void poner(char \*p);**

**int tecla(int a);**

**int answer(int res);**

**void salto(){**

**Lcd\_Cmd(\_LCD\_CLEAR); // Comando LCD (borrar el LCD)**

**kp=0;**

**asm{**

**GOTO 0X00**

**}**

**}**

**void main() {**

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

**TMR0 = 0;**

**INTCON = 0xA0; // Disable interrupt PEIE,INTE,RBIE,T0IE**

**PORTC = 0x00;**

**TRISC = 0x00; // PORTC como salida**

**PORTE=0;**

**ADCON1 = 0x82; // Voltaje de referencia para la conversión A/D es VCC**

**TRISE = 0x01; // RE0 is input, [RE1-RE7] are output**

**Delay\_ms(20);**

**//Lcd\_Cmd(\_LCD\_CLEAR); // Comando LCD (borrar el LCD)**

**band=0;**

**ANSEL = 0x01; // Pin RE0 configura como entrada analógica**

**ANSELH = 0; // Los demás pines configuran como digitales**

**PORTA = 0x00;**

**TRISA = 0x00; // PORTA como salida**

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

**C2ON\_bit = 0;**

**cnt = 0; // Resetear contador**

**Lcd\_Init(); // Inicializa Lcd**

**Lcd\_Cmd(\_LCD\_CLEAR); // Borrar display**

**Lcd\_Cmd(\_LCD\_CURSOR\_OFF); // Cursor apagado**

**Delay\_ms(1);**

**Lcd\_Out(1,1,txt1); // Escribe texto en la primera línea**

**Lcd\_Out(2,1,txt2); // Escribe texto en la primera línea**

**Lcd\_Out(4,2,"Boris Puero -- P# 8");// Escribe texto en la primera línea**

**Delay\_ms(2000);**

**Lcd\_Cmd(\_LCD\_CLEAR); // Borrar display**

**Lcd\_Out(1,4,"Boris Puero"); // Escribe texto en la primera línea**

**Delay\_ms(100);**

**name1 = mask(14); // Prepare mask for**

**name2 = mask(1); // Prepare mask for**

**do{**

**for(i=0; i<10; i++) { // Mover el texto a la derecha**

**Lcd\_Cmd(\_LCD\_SHIFT\_RIGHT);**

**Delay\_ms(100);**

**kp = Keypad\_Key\_Click(); // Almacena el código de la tecla en kp**

**if(kp==1 || kp==2)**

**i=10;**

**}**

**if(kp!=1 && kp!=2){**

**for(i=0; i<10; i++) { // Mover el texto a la izquierda**

**Lcd\_Cmd(\_LCD\_SHIFT\_LEFT);**

**Delay\_ms(100);**

**kp = Keypad\_Key\_Click(); // Almacena el código de la tecla en kp**

**if(kp==1 || kp==2)**

**i=10;**

**}**

**}**

**}while (kp!=1 && kp!=2);**

**DELAY\_MS(10);**

**Lcd\_Cmd(\_LCD\_CLEAR); // Borrar display**

**do {**

**if(kp==1){**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**juego();}**

**else{**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**valor();}**

**} while (1);**

**}**

**void juego(){**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**Lcd\_Out(1,3,"ON/C.-Habilitar");**

**Lcd\_Out(2,3,"Elegir\_Operacion");**

**Delay\_ms(1000);**

**do{**

**kp = Keypad\_Key\_Click(); // Almacena el código de la tecla en kp**

**cnt\_2++;**

**if(cnt\_2>=20)**

**cnt\_2=1;**

**}while (kp!=13);**

**do{**

**kp = Keypad\_Key\_Click(); // Almacena el código de la tecla en kp**

**cnt++;**

**if(cnt>=20)**

**cnt=1;**

**}while (kp!=4 && kp!=8 && kp!=16);**

**do{**

**band=1;**

**delay\_ms(50);**

**Lcd\_Cmd(\_LCD\_CLEAR); // Borrar display**

**delay\_ms(50);**

**WordToStr(cnt,txt1); // Transform counter value to string**

**WordToStr(cnt\_2,txt2); // Transform counter value to string**

**Lcd\_Out(1,1,txt1);**

**Lcd\_Out(1,8,txt2);**

**switch (kp) {**

**case 4:**

**Lcd\_Out(1,6," -");**

**i= cnt-cnt\_2;**

**i=abs(i);**

**break; // 2**

**case 8:**

**Lcd\_Out(1,6," x");**

**i= cnt\*cnt\_2;**

**break; // 3**

**case 16:**

**Lcd\_Out(1,6," +");**

**i= cnt+cnt\_2;**

**break; // 3**

**}**

**ans = answer(i);**

**if(ans==i)**

**porta = ++wins;**

**else**

**wins=0;**

**}while(wins!=0);**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**delay\_ms(500);**

**Lcd\_Out(1,2,"Perdiste");**

**Lcd\_Out(2,2,"Numero Incorrecto");**

**delay\_ms(2000);**

**salto();**

**}**

**void valor(){**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**text2="Bin ADC:";**

**text3="Dec ADC:";**

**band=1 ;**

**Lcd\_Cmd(\_LCD\_CLEAR); // Comando LCD (borrar el LCD)**

**delay\_ms(1000);**

**do{**

**kp = Keypad\_Key\_Click();**

**adc= ADC\_Read(5); // Conversión A/D. Pin RE0 es una entrada.**

**Lcd\_Out(3,1,text2); // Escribir el resultado en la tercera línea**

**Lcd\_Out(4,1,text3); // Escribir el resultado en la cuarta línea**

**delay\_ms(100);**

**//MOSTRAR EL VALOR DECIMAL DEL ADC**

**WordToStr(adc,txt1); // Transform counter value to string**

**Lcd\_Out(4,13, txt1); // Display counter value on Lcd**

**//MOSTRAR EL VALOR ANALOGICO EN BINARIO**

**for(j=0;j<10;j++){**

**temp1[j]=(int)adc%2;**

**Lcd\_Chr(3,20-j,48+temp1[j]);**

**adc=adc/2;**

**Delay\_ms(1);**

**}**

**}while(kp!=15);**

**salto();**

**}**

**// Lcd\_Cmd(\_LCD\_CLEAR);**

**void interrupt(){**

**if (dig\_no==0) {**

**PORTE = 0; // Turn off both displays**

**PORTC = name1; // Set mask for displaying ones on PORTD**

**PORTE = 2; // Turn on display for ones (LSD)**

**dig\_no = 1;**

**} else {**

**PORTE = 0; // Turn off both displays**

**PORTC = name2; // Set mask for displaying tens on PORTD**

**PORTE = 4; // Turn on display for tens (MSD)**

**dig\_no = 0;**

**}**

**TMR0 = 0; // Reset counter TMRO**

**INTCON = 0x20; // Bit T0IF=0, T0IE=1**

**}**

**int tecla(int a){**

**int x=0;**

**switch (a) {**

**case 1: x = 1; break; // 7 // Uncomment this block for keypad4x4**

**case 2: x = 2; break; // 8**

**case 3: x = 3; break; // 9**

**case 5: x = 4; break; // 4**

**case 6: x = 5; break; // 5**

**case 7: x = 6; break; // 6**

**case 9: x = 7; break; // 1**

**case 10: x = 8; break; // 2**

**case 11: x = 9; break; // 3**

**case 14: x = 0; break; // 0**

**default: return -1;**

**}**

**return x;**

**}**

**int answer(int res){**

**int kpk=0,num1=0;**

**do{**

**kpk = Keypad\_Key\_Click(); // Almacena el código de la tecla en kp**

**cnt\_2++;**

**if(cnt\_2>=20) cnt\_2=1;**

**delay\_ms(10);**

**}while (!kpk);**

**num1=tecla(kpk);**

**cnt=cnt\_2;**

**if(kp==4) cnt=cnt+13;**

**if(res>9){**

**do{**

**kpk = Keypad\_Key\_Click();**

**cnt\_2++;**

**if(cnt\_2>=20) cnt\_2=1;**

**delay\_ms(10);**

**}while (!kpk);**

**num1= num1\*10 + tecla(kpk);**

**}**

**if(res>99){**

**do**

**kpk = Keypad\_Key\_Click();**

**while (!kpk);**

**num1= num1\*10 + tecla(kpk);**

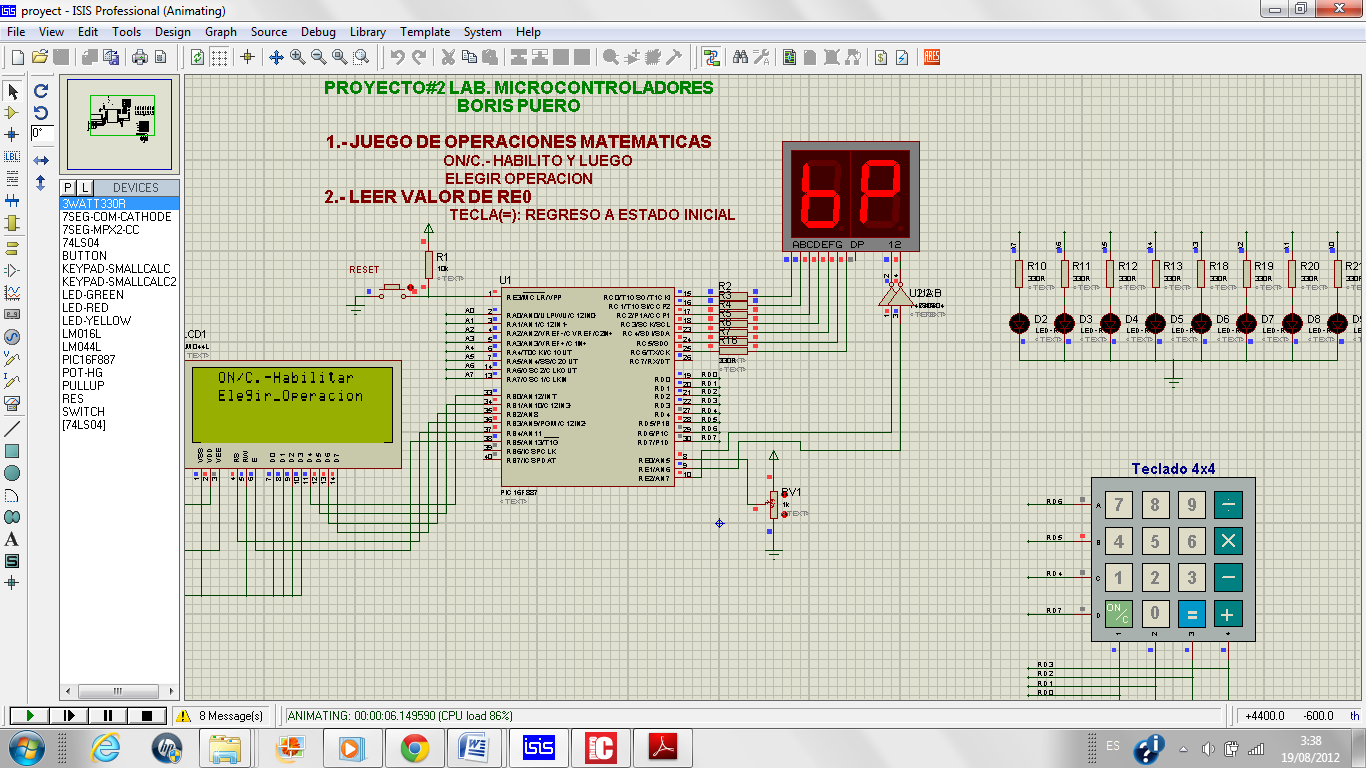
**}**

**return num1;**

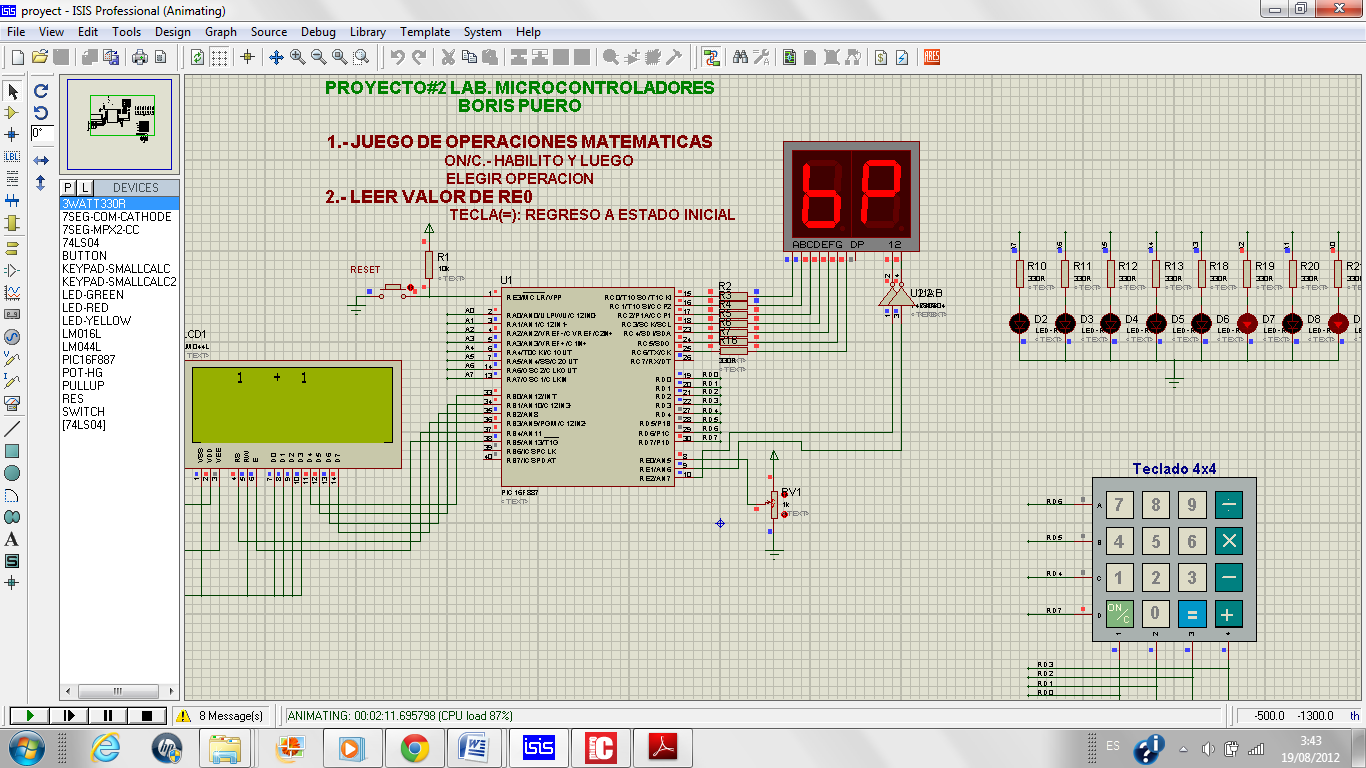
**}**

**6.- Simulación en PROTEUS**

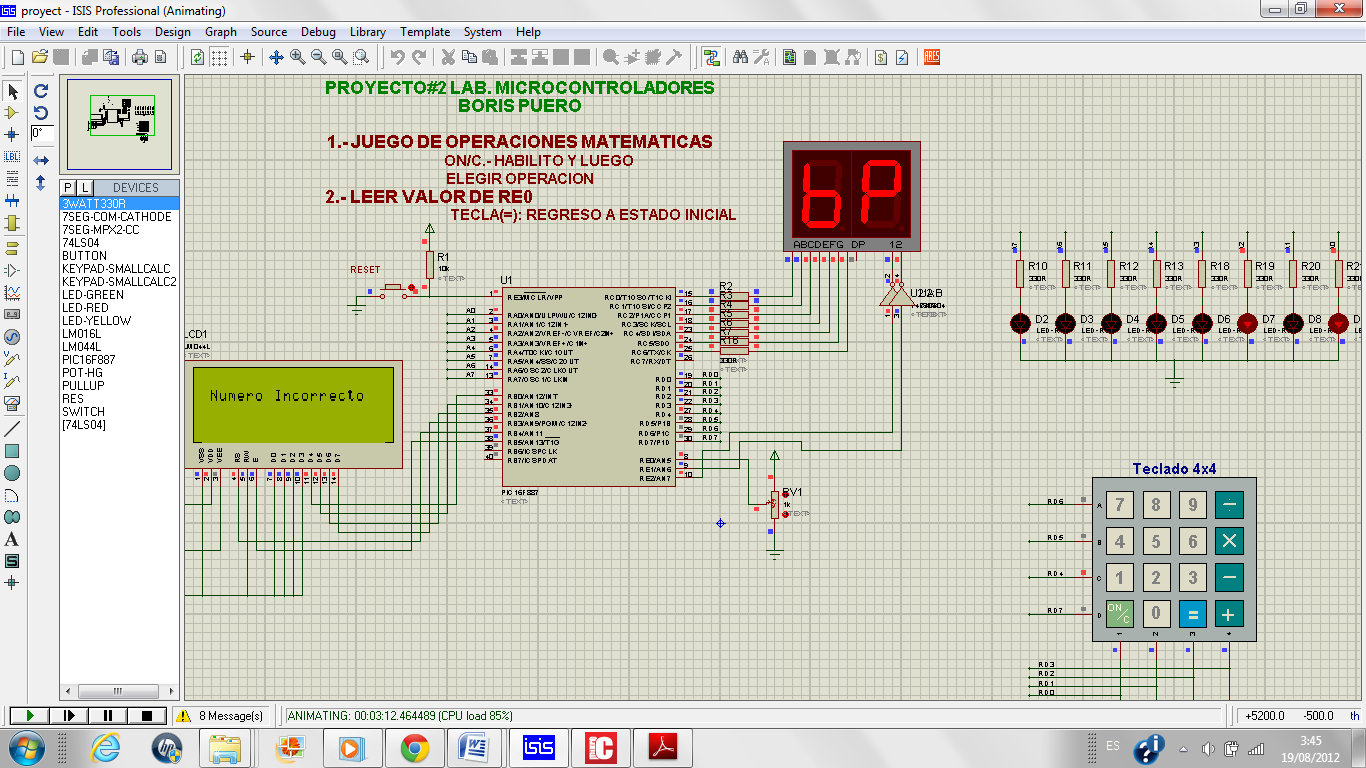
**Juego de Operaciones Matemáticas**

****

Aquí ya se ha ingresado al juego de operaciones matemáticas donde me dice que habilite y elija la operación.

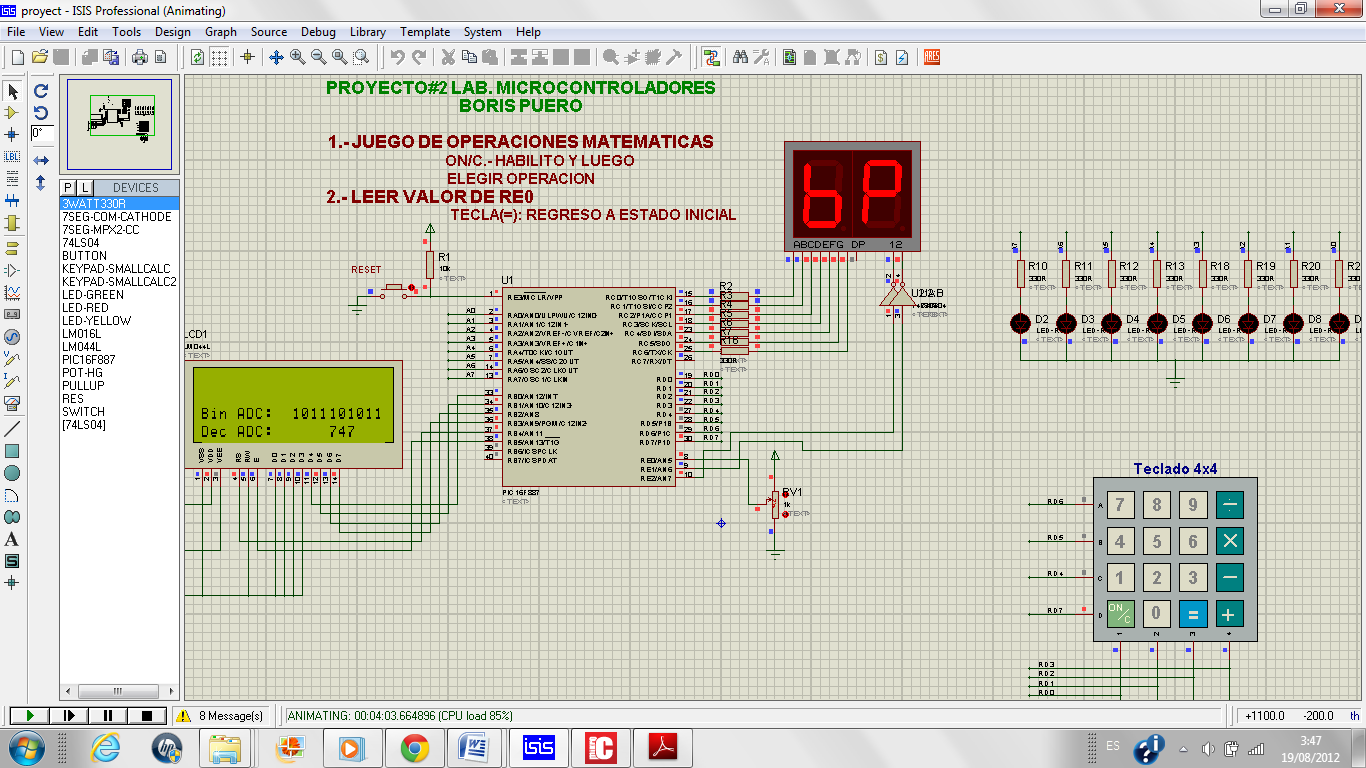


Ahí estamos realizando un juego con la suma como podemos observar en esta imagen aquí ya llevo 5 sumas acertadas como muestra su valor binario en los leds.



Instancia donde puse un número incorrecto y perdí el juego quedándome con 5 aciertos.

**Simulación de Lectura de RE0**



Aquí observamos como en la LCD se muestra el valor binario y decimal que se esta sensando en la patita RE0 donde hay un potenciómetro colocado en este caso el decimal es 747 y el binario es 1011101011.

**8.- Conclusiones:**

* La simulación de situaciones reales tales como un juego de operaciones matemáticas o la lectura de un valor analógico representando su valor binario nos permite comprender con mayor claridad la elaboración de circuitos controladores, más aun con lenguaje C, teniendo presente que la forma en cómo fue programada, es mucho mas minimizada que con Assembler pero a la vez mas óptima y esto nos da la pauta para el diseño de similares artefactos que existen a nuestro alrededor.
* El haber realizado este proyecto fue de gran utilidad para enriquecer nuestros conocimientos ya que no solamente se trabajo con algo que comúnmente se aplica a la realidad como lo son las LCD y los teclados matriciales, sino que también se hizo el uso de importantes librerías que nos ofrece el software con funciones para facilitarnos y realizar de manera rápida y entendible nuestro trabajo.
* Al haber realizado este proyecto podemos comprobar como una simple idea acerca de un juego puede desarrollarse mediante la programación de un PIC con lenguaje C y la implementación de unos cuantos leds, teclado y Lcd; es decir los PIC son herramientas tan poderosas que según la utilidad que le queramos dar se la puede manipular para funcionamiento de cualquier proyecto que nuestra imaginación pueda crear.

**9.- Recomendaciones:**

* Se recomienda realizar un análisis exhaustivo del circuito antes de implementarlo, con la finalidad de tener una idea clara del proceso a seguir y de su funcionamiento, considerando y siguiendo paso a paso el diagrama de flujo del programa principal, con sus respectivas subrutinas y así no habrán errores a futuro que nos perjudiquen.
* Es recomendable el uso de todas las funciones de mikroC pro for PIC’s para encontrar errores en nuestro programa si es el caso de que los tuviéramos, entre estas funciones tenemos la herramienta de ejecución paso a paso, que es una muy buena alternativa para una adecuada programación y también podemos usar los breakpoints que son muy útiles.
* Revisar que el mikroC pro tenga activado su archivo de licencia ya que el demo que se instala normalmente no puede compilar un programa de salida (fichero.hex) de más de 2Kb de programa de palabras, y es por este motivo fundamental que nos vimos obligados a poner limitaciones en nuestro proyecto ya que en algunas ocasiones no compilaba por aquel motivo.