**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FIEC**

**Laboratorio de Micro controladores**

**PROYECTO SEGUNDO PARCIAL**

**LECTOR DE ADC Y JUEGO DE ADIVINAR PALABRAS**

**Alumno**

**Freddy Santiago Hidalgo Castro**

**Paralelo: #**

**9**

**Fecha de presentación:**

**22/08/12**

**2012 – 1° TÉRMINO**

**1.- Enunciado del proyecto**

El proyecto cuenta con un teclado 4x4; una pantalla LCD un potenciómetro y un display de 7 segmentos multiplexado; el proyecto consiste en dos partes la primera es la lectura del valor del potenciómetro a través de la conversión analógica a digital mientras que la segunda parte es un juego el cual consiste en adivinar la palabra correcta tres veces. Para seleccionar entre estas dos se hace uso del teclado.

En este proyecto se hace uso del display de 7 segmentos para visualizar en todo momento las iniciales del nombre del estudiante; mientras que la LCD se la usa como interfaz gráfica para interactuar con el PIC y poder seleccionar así las diferentes opciones dentro de los menús. Para la selección y de dichas opciones apreciables se usa un teclado con el cual escogemos o variamos entre las diferentes opciones dadas por el PIC.

**2.- Diagrama de Bloques**

****

**3.- Diagrama de Flujo funcional del Programa principal**

**Diagrama de Flujo funcional del Programa principal**



***Subrutina POT***



***Subrutina Code\_txt***



***Subrutina Conv***



***Subrutina Interrupción***



***Subrutina JUEGO*:**



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

1. Se realiza la iniciación del programa dando la configuración al PIC; en dicha configuración incluimos un reloj de 8MHz; el apagado del perro guardián.
2. Configura el teclado en el puertoD y configuración de la pantalla para funcionar con el puertoB.
3. Creación de la variable dig\_no y de los punteros a cadenas de caracteres constantes txt1,txt2,txt3,txt4,txt5,txt6,txt7,txt8,txt9.
4. Se genera la función para extraer los datos apuntados por los punteros anteriormente mencionados, de manera que se la usará cuando se imprima dichas cadenas en pantalla.
5. Se crea la función convertir, que permite convertir un valor numérico en su valor binario dentro de una cadena de caracteres, dicho valor se usará mas adelante para visualizar en pantalla.
6. Se crea la función POT la cual es accedida desde el programa principal; dentro de esta función se hace la declaración de subvariables y mientras no se presione la tecla 16 se ejecuta la lectura del valor ADC a través de AN5 que es el pin 3 del puertoE , dicha lectura se muestra en pantalla y se hace llamado a la función convertir y se muestra su valor en binario en pantalla; en caso de presionar una tecla diferente de 16 se mostrará en pantalla un mensaje indicando que para salir se debe presionar dicha tecla.
7. Se hace la creación de la función JUEGO la cual es llamada a través del menú principal, en esta función se declara una cadena de carácter palabra, un carácter y variables tipo short; primero se imprime en pantalla un submenú para escoger entre “Comenzar juego”, o “Reglas”, y se va a mostrar esto hasta que se selección una de las dos opciones o se presione la tecla 16.
8. Una vez presionada una de las tecla pregunta por la variable tecla3, si tecla3 es 1 se asigna un valor random y en base a ese valor se substrae una cadena del arreglo txt8; se mira el tamaño de dicho arreglo y se imprime el primer y último carácter de dicho arreglo; mientras no se presione la tecla 16, muestra dichas letras y se puede comenzar a insertar los caracteres, en caso de presionar 2 se avanza las letras hacia arriba y si se presionan hacia abajo se regresa la letra a su estado anterior y en caso de presionar la tecla 6 se avanza y en caso de que el carácter no sea igual al del arreglo substraído permanece la pantalla en dicha posición esperando que se ingrese correctamente y se avance, en caso de ingresar tres veces mal algún carácter de esta palabra automáticamente pierde y se regresa al submenú JUEGO
9. En caso de acertar la palabra se mostrará otra palabra y se vuelve a repetir el proceso de acertar palabra; hasta que se logre acertar 3 palabras correctas en cuyo caso se gana el juego y se regresa al menú principal de JUEGO.
10. En caso de haber escogido dentro del menú JUEGO la opción 2; en pantalla se mostraran las cadenas de caracteres extraídos del arreglo txt4,txt5,txt6 y txt7; de manera que se muestra en pantalla las instrucciones del juego; y esperas a que se presione la tecla 6 en caso de presionar muestra el siguiente juego de cadenas y dentro de esta si se presiona la tecla 4 regresa a mostrar la anterior y si se presiona 6 se muestra la siguiente y así sucesivamente hasta que se haya mostrado todas las instrucciones en cuyo caso sale a mostrar el menú JUEGO(); en caso de haber presionado la tecla 16 mientras se enseñaban las reglas, automáticamente se salía y mostraba el menú JUEGO().
11. Se declara la función de interrupción que es la encargada de mostrar en el display de 7 segmentos multiplexados el valor de las iniciales “J” y “C” alternando rápidamente entre ellos, se accede a esta función cuando se ha hecho sobrecarga del TIMER1.
12. Se declara las instrucción del programa principal; primero declarando los puertos como digitales excepto la pata AN5; deshabilitando los comparadores y declarando el puerto A,B, C y D como salidas y el puerto E como entrada analógicas.
13. Se inicializa la pantalla LCD y se inicializa el teclado y se desactiva el cursor, luego pasa a mostrar en pantalla un mensaje de bienvenida durante 0,5 segundos para luego pasar a mostrar un menú para selección entre potenciómetro y juego; pasa a preguntar por la tecla presionada hasta que se seleccione la tecla 1 o la tecla 2; en caso de seleccionar la tecla 1 se procede a acceder a la función POT y en caso de haber presionado la tecla 2 se hace acceso a la función JUEGO(); todo esto lo realiza eternamente.

**5.- Listado del programa fuente en lenguaje ensamblador**

**Proyecto**

/\*

 \* Nombre del Proyecto:

 P9b\_adc.c

 \* Nombre del Autor:

Freddu hidalgo castro

\* Description:

 (Explicación del ejercicio)

 \* Test configuration:

 MCU: PIC16F887

 Oscillator: HS, 08.0000 MHz

 SW: mikroC PRO for PIC

 \* NOTES:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 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;

 // End LCD module connections

unsigned short dig\_no=0;

const char \*txt1[] = {"BIENVENIDO","SR USUARIO","MENU PRINCIPAL","1)POTENCIOMETRO ","2)JUEGO LIBRE"};

const char \*txt2[] = {"el ADC es: ","Valor leido por","binario: ","decimal: ","voltage: "};

const char \*txt3[] = {"Nota: Para salir ","presione la tecla D"};

const char \*txt4[] = {"1)Comenzar juego","2)Reglas","ATENCION","Juego consiste en"};

const char \*txt5[] = {"adivinar palabras","ingresando "};

const char \*txt6[] = {"los caracteres","correspondientes","a dicha palabra,","si falla 3 letras"};

const char \*txt7[] = {"de dicha palabra","pierde el juego","si acierta en 3","palabras gana"};

const char \*txt8[] = {"AMARILLO","VIOLETA","ASESINATO","GRIS","AZUL","ROJO","SIMPLE","SACO","PONER",

 "PALMA","PIE","DIENTE","NIEVE","CALOR","FRIO","TIEMPO","RISA","LLANTO","VISTA"};

const char \*txt9[] = {" ","mala","USTED HA","PERDIDO EL JUEGO","GANADO EL JUEGO","PALABRA: "};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

char\* codetxt\_to\_ramtxt(const char\* ctxt)

 {

 static char txt[20];

 char i;

 for(i =0; txt[i] = ctxt[i]; i++);

 return txt;

 }

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 void conv(unsigned int val,unsigned char txt15[11] ){

 unsigned short i;

 unsigned bit tmp\_bit;

 unsigned char txt\_tmp[4]; // ByteToStr function require 4 byte array

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

 tmp\_bit = val; // tmp\_bit takes last bit of var\_num

 ByteToStr(tmp\_bit, txt\_tmp); // converting bit to string

 txt15[10 - 1 - i] = txt\_tmp[2]; // placing char to the appropriate place in string

 val >>= 1; // shifting var\_num bits one place to the right

 }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void POT() {

 unsigned short tecla2=0;

 int temp\_res;

 unsigned char txt[4];

 unsigned char txt15[11];

 Lcd\_cmd(\_lcd\_clear);

 delay\_ms(40);

 Irp\_bit=0;

 while(tecla2!=16){

 Lcd\_out(1,1,codetxt\_to\_ramtxt(txt2[1]));

 Lcd\_out(2,1,codetxt\_to\_ramtxt(txt2[0]));

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

 delay\_ms(10);

 wordtostr(temp\_res,txt);

 Lcd\_out(3,12,txt);

 Lcd\_out(3,1,codetxt\_to\_ramtxt(txt2[3]));

 Irp\_bit=1;

 conv(temp\_res,txt15);

 delay\_ms(10);

 Lcd\_out(4,1,codetxt\_to\_ramtxt(txt2[2]));

 Lcd\_out(4,10,txt15);

 IRP\_bit = 0;

 tecla2=keypad\_key\_click();

 if(tecla2!=0 && tecla2!=16)

 {

 Lcd\_cmd(\_Lcd\_clear);

 delay\_ms(40);

 Lcd\_out(2,1,codetxt\_to\_ramtxt(txt3[0]));

 Lcd\_out(3,1,codetxt\_to\_ramtxt(txt3[1]));

 delay\_ms(1000);

 Lcd\_cmd(\_lcd\_clear);

 }

 }

 Lcd\_cmd(\_lcd\_clear);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void JUEGO(){

 unsigned short tecla3=0;

 unsigned short cadena=0,perder=0,ganar=0,tamano,j=0,sal=0;

 char palabra[20];

 char i='A';

 Lcd\_cmd(\_lcd\_clear);

 do{

 Lcd\_out(2,1,codetxt\_to\_ramtxt(txt4[0]));

 Lcd\_out(3,1,codetxt\_to\_ramtxt(txt4[1]));

 while(tecla3!=1 && tecla3!=2 && tecla3!=16)

 {

 tecla3=keypad\_key\_click();

 delay\_ms(100);

 }

 if(tecla3==1)

 {

 do{

 lcd\_cmd(\_lcd\_clear);

 cadena=rand()%19;

 tamano=strlen(codetxt\_to\_ramtxt(txt8[cadena]));

 strcpy(palabra,codetxt\_to\_ramtxt(txt8[cadena]));

 lcd\_chr(2,1,palabra[0]);

 lcd\_chr(2,tamano,palabra[tamano-1]);

 while(tecla3 !=16)

 {

 lcd\_cmd(\_LCD\_UNDERLINE\_ON);

 lcd\_chr(2,j+2,i);

 lcd\_cmd(\_lcd\_move\_cursor\_left);

 do

 {

 tecla3=keypad\_key\_click();

 delay\_ms(100);

 }while(tecla3!=2 && tecla3!=16 && tecla3!=7 && tecla3!=10);

 if (tecla3 == 16)

 sal++;

 if (tecla3 == 2)

 { i++;

 lcd\_chr(2,j+2,i);

 }

 if (tecla3 == 10)

 { i--;

 lcd\_chr(2,j+2,i);

 }

 if (tecla3 == 7)

 {

 lcd\_cmd(\_lcd\_cursor\_off);

 if(memcmp(i, palabra[1+j],3 )!=0)

 {

 lcd\_out(4,2,codetxt\_to\_ramtxt(txt9[1]));

 perder++;

 j--;

 delay\_ms(500);

 }

 j++;

 i='A';

 lcd\_out(4,2,codetxt\_to\_ramtxt(txt9[0]));

 }

 lcd\_cmd(\_lcd\_cursor\_off);

 if (perder == 3)

 {

 lcd\_cmd(\_lcd\_clear);

 tecla3=16;

 lcd\_out(2,1,codetxt\_to\_ramtxt(txt9[2]));

 lcd\_out(3,1,codetxt\_to\_ramtxt(txt9[3]));

 lcd\_out(1,1,codetxt\_to\_ramtxt(txt9[5]));

 lcd\_out\_cp(palabra);

 delay\_ms(1000);

 }

 if((j+3) > tamano)

 {

 ganar++;

 tecla3=16;

 j=0;

 perder=0;

 }

 }

 tecla3=0;

 if (sal == 1)

 {

 tecla3=16;

 sal=0;

 j=0;

 }

 if(ganar == 3)

 {

 lcd\_cmd(\_lcd\_clear);

 tecla3=16;

 lcd\_out(2,1,codetxt\_to\_ramtxt(txt9[2]));

 lcd\_out(3,1,codetxt\_to\_ramtxt(txt9[4]));

 ganar=0;

 delay\_ms(1000);

 j=0;

 }

 if (perder == 3)

 {

 tecla3=16;

 perder = 0;

 j=0;

 }

 }while(tecla3!=16);

 tecla3=0;

 }

 if(tecla3==2)

 {

 Primero:

 Lcd\_cmd(\_lcd\_clear);

 Lcd\_out(1,1,codetxt\_to\_ramtxt(txt4[2]));

 Lcd\_out(2,1,codetxt\_to\_ramtxt(txt4[3]));

 Lcd\_out(3,1,codetxt\_to\_ramtxt(txt5[0]));

 Lcd\_out(4,1,codetxt\_to\_ramtxt(txt5[1]));

 do

 {

 tecla3=keypad\_key\_click();

 delay\_ms(100);

 }while(tecla3!=7 && tecla3!=16);

 if(tecla3 == 7)

 {

 Segundo:

 Lcd\_cmd(\_lcd\_clear);

 Lcd\_out(1,1,codetxt\_to\_ramtxt(txt6[0]));

 Lcd\_out(2,1,codetxt\_to\_ramtxt(txt6[1]));

 Lcd\_out(3,1,codetxt\_to\_ramtxt(txt6[2]));

 Lcd\_out(4,1,codetxt\_to\_ramtxt(txt6[3]));

 do

 {

 tecla3=keypad\_key\_click();

 delay\_ms(100);

 } while(tecla3!=7 && tecla3!=16 && tecla3!=5);

 if (tecla3 == 5)

 goto Primero;

 if (tecla3 == 7)

 {

 Lcd\_cmd(\_lcd\_clear);

 Lcd\_out(1,1,codetxt\_to\_ramtxt(txt7[0]));

 Lcd\_out(2,1,codetxt\_to\_ramtxt(txt7[1]));

 Lcd\_out(3,1,codetxt\_to\_ramtxt(txt7[2]));

 Lcd\_out(4,1,codetxt\_to\_ramtxt(txt7[3]));

 do

 {

 tecla3=keypad\_key\_click();

 delay\_ms(100);

 } while(tecla3!=7 && tecla3!=16 && tecla3!=5);

 }

 if (tecla3 == 5)

 goto Segundo;

 }

 tecla3=0;

 }

 Lcd\_cmd(\_lcd\_clear);

 }while(tecla3!=16);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void interrupt(){

 if (dig\_no==0) {

 PORTA = 0; // Turn off both displays

 PORTC = 0b0001110; // Set mask for displaying ones on PORTD

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

 dig\_no = 1;

 }

 else {

 PORTA=0;

 PORTC = 0b0001001; // Set mask for displaying tens on PORTD

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

 dig\_no = 0;

 }

 TMR0 = 0; // Reset counter TMRO

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

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

 unsigned short tecla=0;

 ANSEL = 0x01; // Configure AN5 pin as analog

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

 C1ON\_bit = 0; // Disable comparators

 C2ON\_bit = 0;

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

 TMR0 = 0;

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

 TRISA = 0x00; // PORTA is output

 TRISC = 0x00; // PORTC is output

 TRISD = 0x00; // PORTD is output

 TRISB = 0x00; // PortB is output

 PORTE = 0;

 TRISE = 0x01;

 PORTA = 0x00;

 PORTC = 0;

 LCD\_init();

 Keypad\_init();

 Lcd\_cmd(\_Lcd\_clear);

 Lcd\_cmd(\_lcd\_cursor\_off);

 do {

 Lcd\_out(2,6,codetxt\_to\_ramtxt(txt1[0]));

 Lcd\_out(3,6,codetxt\_to\_ramtxt(txt1[1]));

 delay\_ms(500);

 Lcd\_cmd(\_lcd\_clear);

 Lcd\_out(1,4,codetxt\_to\_ramtxt(txt1[2]));

 Lcd\_out(3,2,codetxt\_to\_ramtxt(txt1[3]));

 Lcd\_out(4,2,codetxt\_to\_ramtxt(txt1[4]));

 while(tecla!=1 && tecla!=2)

 {

 tecla=keypad\_key\_click();

 delay\_ms(100);

 }

 if (tecla==1)

 {

 POT();

 delay\_ms(80);

 tecla=0;

 }

 if (tecla==2)

 {

 JUEGO();

 delay\_ms(80);

 tecla=0;

 }

 }while (1); // Endless loop

}

**6.- Circuito armado en PROTEUS**



**Menu principal:presiona 1 en el teclado ejercicio del potenciometro, presiona 2 juego de adivinanza d palabras.**

****

**Al variar el potenciometro tenemos el valor leído por el adc en decimal y binario**

****

**Al precionar cualquier tecla q no sea las indicadas sale el mensaje siguiente si desea salir.**

****

****

**Usted tiene 3 intentos de fallar en adivanar la letra que falta.**

****

****

**7.- Conclusiones**

* Al realizar este proyecto se ha podido incursionar en el uso de varias librerías mejorando así la práctica y el uso del programa MikroCpro; hemos aprendido el mejor manera de estas librerías junto con las variadas funciones indicando así que para el uso de librerías que comunican con el medio externo siempre se las debe de inicializar como es el caso de la LCD, el teclado, el UART y el I2C.

* Se ha mejorado el uso de las interrupciones por desborde del Timer1 y de la multiplexación para poder mostrar a través del display las iniciales del nombre; también se puede generar interrupciones por botoneras para ejecutar comando o subfunciones debido a dicha interrupción, recordando siempre que se usan 8 pines para dar el valor al display y otros 2 para hacer la multiplexacion; se observan dos datos debido a la rapidez con la que se ejerce el cambio entre displays.
* Se trato de mejorar y evitar el uso de mucha memoria RAM, implementando la memoria ROM, usando punteros a cadenas de caracteres que contenían la mayoría de las palabras; siendo así que cuando se la llama se debe usar una función que llama de ROM a RAM para poder hacer uso de la cadena apuntada por dicho puntero; dicho uso puede ser LCD comparación de caracteres o descripciones de dicha cadena.

**8.- Recomendaciones**

* Evitar excederse de un 50% de uso de la memoria RAM, puesto que el compilador comienza a presentar problemas para acceder a las variables cuando se excede de dicho valor.
* Recordar bien como se configuro la pantalla y el teclado dentro del programa para que de igual manera se haga su conexión en PROTEUS y en caso de usar otros circuitos recordar a que pin hacen referencia dentro del PIC.
* No olvidarse de apagar el perro guardián, configurar con que oscilador se va a trabajar y desactivar mensajes por error, de manera que se facilite la implementación del programa y evitemos reset por desborde del perro.
* Hacer uso de las conversiones de tipos de valores y cadenas cuando sea necesario y del orden en que van las funciones a llamarse.