UNIVERSIDAD NACIONAL EXPERIMENTAL POLITECNICA

“ANTONIO JOSE DE SUCRE”

VICE-RECTORADO DE PUERTO ORDAZ

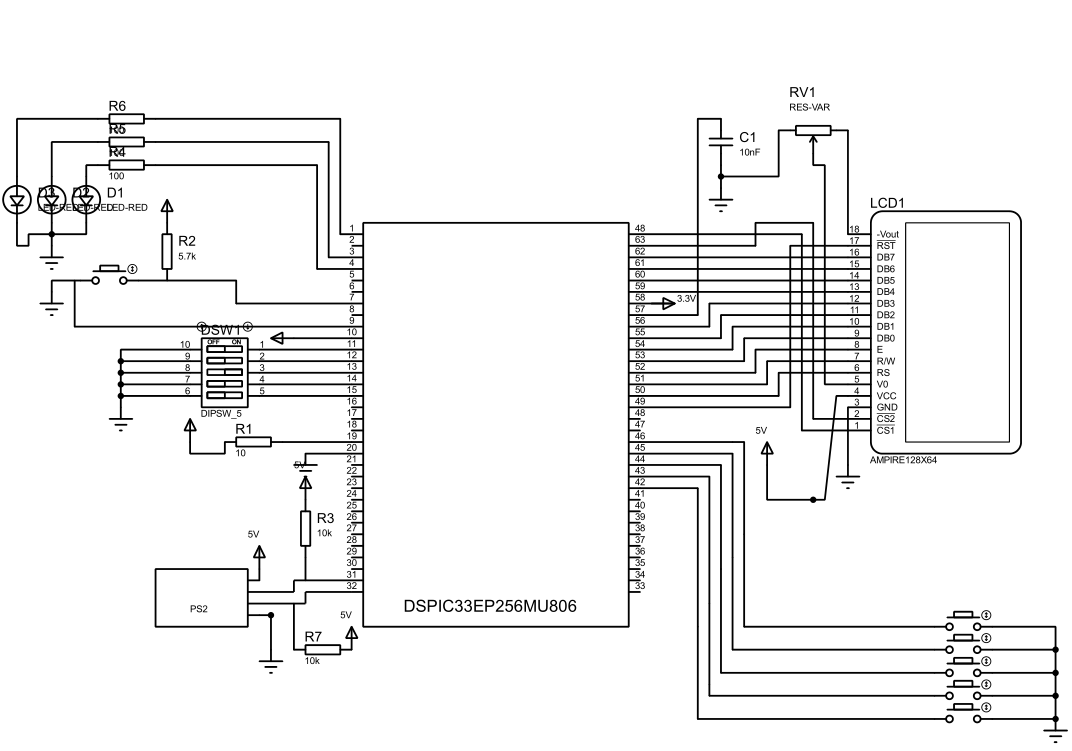
DEPARTAMENTO DE ELECTRONICA

**MICROPROCESADORES II**

Laboratorio 1

|  |  |  |
| --- | --- | --- |
| Profesor: |  | Bachilleres: |
| Rafael Surga Luvo |  | Brito Eduardo C.I:24560857 |
|  |  | Duran Alejandro C.I: 25395510 |
|  |  | Zorrilla Paola C.I:24848197 |

Ciudad Guayana. Marzo, 2018

Esquemático

**CODIGO**

**LAB1.C**

#include "barras.h"

#include "sprites.h"

//~Constantes del dsPIC

//Variables de PS2

sbit PS2\_Data at RF4\_bit;

sbit PS2\_Clock at RF5\_bit;

sbit PS2\_Data\_Direction at TRISF4\_bit;

sbit PS2\_Clock\_Direction at TRISF5\_bit;

//Variables de GLCD

sbit GLCD\_D7 at RE1\_bit;

sbit GLCD\_D6 at RE0\_bit;

sbit GLCD\_D5 at RF1\_bit;

sbit GLCD\_D4 at RF0\_bit;

sbit GLCD\_D3 at RD7\_bit;

sbit GLCD\_D2 at RD6\_bit;

sbit GLCD\_D1 at RD5\_bit;

sbit GLCD\_D0 at RD4\_bit;

sbit GLCD\_D7\_Direction at TRISE1\_bit;

sbit GLCD\_D6\_Direction at TRISE0\_bit;

sbit GLCD\_D5\_Direction at TRISF1\_bit;

sbit GLCD\_D4\_Direction at TRISF0\_bit;

sbit GLCD\_D3\_Direction at TRISD7\_bit;

sbit GLCD\_D2\_Direction at TRISD6\_bit;

sbit GLCD\_D1\_Direction at TRISD5\_bit;

sbit GLCD\_D0\_Direction at TRISD4\_bit;

sbit GLCD\_CS1 at LATE3\_bit;

sbit GLCD\_CS2 at LATE2\_bit;

sbit GLCD\_RS at LATD1\_bit;

sbit GLCD\_RW at LATD2\_bit;

sbit GLCD\_EN at LATD3\_bit;

sbit GLCD\_RST at LATE4\_bit;

sbit GLCD\_CS1\_Direction at TRISE3\_bit;

sbit GLCD\_CS2\_Direction at TRISE2\_bit;

sbit GLCD\_RS\_Direction at TRISD1\_bit;

sbit GLCD\_RW\_Direction at TRISD2\_bit;

sbit GLCD\_EN\_Direction at TRISD3\_bit;

sbit GLCD\_RST\_Direction at TRISE4\_bit;

//Constantes del sistema

//Constantes de las ubicaciones de los pines

const int BTN\_PPS\_LOCATIONS[5]={-1,37,36,35,34};//5 posiciones para las 4 Ubicaciones de los botones de Interrupciion del ejercicio 1

// el 0 nunca se usa porque INT0 esta fijo en el pin 46 RP64. RB2-5

const int DIP\_PPS\_LOCATIONS[5]={32,33,38,39,40};//5 posiciones para 5 Dipswitches RPI32,33,38,39,40 Y RB0,1,6,7,8

const int KEYBOARD\_PPS\_LOCATIONS[2]={100,101};//RF4,RF5

//const int LED\_PPS\_LOCATIONS[3]={85,87,118};//3 posiciones para 3 LEDS de salida RE5,RE7,RG6

//Variables de trabajo

int counters[5]={0,0,0,0,0};

int valores\_cn[4]={0,0,0,0};

unsigned short keydata = 0, special = 0, down = 0,casoQactivo=0;

//~~~~~~~~~~~~~~~~Declaraciones de Funciones~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void InterrAdapter(int Ix);

void bar\_drawer(unsigned short x\_left,unsigned short contador\_barra);

int DrawableBars(int real\_number\_of\_bars);

//Interrupciones

void INT0() org 0x14{

InterrAdapter(0);

delay\_ms(600);

IFS0bits.INT0IF=0;

}

void INT1() org 0x3C{

InterrAdapter(1);

delay\_ms(600);

IFS1bits.INT1IF=0;

}

void INT2() org 0x4E{

InterrAdapter(2);

delay\_ms(600);

IFS1bits.INT2IF=0;

}

void INT3() org 0x7E{

InterrAdapter(3);

delay\_ms(600);

IFS3bits.INT3IF=0;

}

void INT4() org 0x80{

InterrAdapter(4);

delay\_ms(600);

IFS3bits.INT4IF=0;

}

void INT\_CN() org 0x3A{

IFS1bits.CNIF = 0;

if(PORTBbits.RB5 ==1){

valores\_cn[0]=1;

valores\_cn[1]=0;

valores\_cn[2]=0;

valores\_cn[3]=0;

}else if(PORTBbits.RB4==1){

valores\_cn[0]=0;

valores\_cn[1]=1;

valores\_cn[2]=0;

valores\_cn[3]=0;

}else if(PORTBbits.RB3==1){

valores\_cn[0]=0;

valores\_cn[1]=0;

valores\_cn[2]=1;

valores\_cn[3]=0;

}else if(PORTBbits.RB2==1){

valores\_cn[0]=0;

valores\_cn[1]=0;

valores\_cn[2]=0;

valores\_cn[3]=1;

}

if(PORTBbits.RB1==1){

RCONbits.SWDTEN=1;

Glcd\_Write\_TEXT("INT WDT",0,0,1);

}

delay\_ms(1000);

Glcd\_Fill(0);

}

void InterrAdapter(int Ix){

if(casoQactivo==1){

counters[Ix]++;

barra(Ix,counters[Ix]);

}

}

//~~~~~~~~~~~~~~~~~~~~~~Configuraciones Iniciales~~~~~~~~~~~~~~~~~~~~~~~~~~

void config\_IO(){

ANSELB=0;

ANSELC=0;

ANSELD=0;

ANSELE=0;

ANSELG=0; //ANA//logiCO SON B Y F

TRISB=0xffff;

TRISDbits.TRISD0=1;

TRISDbits.TRISD11=1;

TRISDbits.TRISD10=1;

TRISDbits.TRISD9=1;

TRISE=0;

TRISG=0;

//Entrada Botones y Dipswitches

//Las entradas del teclado y las salidas de la pantalla son manejadas por las librerias;

}

void config\_LCD(){

Glcd\_Init();

Glcd\_Set\_Font(font5x7 , 5, 7, 32);

Glcd\_Fill(0);

}

void config\_CN(){//CNENx, CNPUx,CNIEx

CNENBbits.CNIEB5=1;CNENBbits.CNIEB4=1; CNENBbits.CNIEB3=1;

CNENBbits.CNIEB2=1;CNENBbits.CNIEB1=1;

CNPUBbits.CNPUB5=1;//CNPU RB1-rb5, rpi33-37

CNPUBbits.CNPUB4=1;CNPUBbits.CNPUB3=1;

CNPUBbits.CNPUB2=1;CNPUBbits.CNPUB1=1;

IPC4bits.CNIP=7;//Prioridad

IFS1bits.CNIF=0; IEC1bits.CNIE=1;//Resset Interrupcion

}

void config\_INT(){

SRbits.IPL =0;// iNTERRUPCION DE CPU ES DE NIVEL 0

INTCON1bits.NSTDIS =0;// INTERRUPCION ANIDADAS ACTIVADAS

INTCON2bits.GIE=1; //interrupciones habilitadas

CORCONbits.IPL3 = 0; // El nivel del cpu es de nivel 0, las interrupciones por perifericos habilitadas

//------------------------- habilitacion de interrupcion

IEC0bits.INT0IE=1; //HABILITA INTERRUPCION POR INT0

IEC1bits.INT1IE=1; //HABILITA

IEC1bits.INT2IE=1; IEC3bits.INT3IE=1; IEC3bits.INT4IE=1;

//prioridad int

// Banderas de Interrupcion post Reset (Limpieza)

IFS0bits.INT0IF=0; //interrupcion de INT0 ACTIVADA

IFS1bits.INT2IF =0; // interrupcion de INT1 ACTIVADA

IFS1bits.INT2IF =0 ; // INTERRIPCION DE INT2 ACTVADA

IFS3bits.INT3IF =0; //INTERRUPCION DE INT3 ACTIVADA

IFS3bits.INT4IF =0; // INTERRUPCION DE INT4 ACTIVADA

//Prioridad de Interrupciones

counters[0]=0; counters[1]=0; counters[2]=0;

counters[3]=0; counters[4]=0;

//--------------------interrupcion flanco positivo

INTCON2bits.INT0EP=0; INTCON2bits.INT1EP=0;

INTCON2bits.INT2EP=0; INTCON2bits.INT3EP=0;

INTCON2bits.INT4EP =0;

//PPS para los pines de Interrupcion

RPINR0bits.INT1R=75; RPINR1bits.INT2R=74;

RPINR1bits.INT3R=73; RPINR2bits.INT4R=72;

}

void disable\_INT(){

IPC0bits.INT0IP= 0; IPC5BITS.INT1IP=0;

IPC7bits.INT2IP =0; IPC13bits.INT3IP=0;

IPC13bits.INT4IP=0;

counters[0]=0; counters[1]=0;

counters[2]=0; counters[3]=0;

counters[4]=0;

casoQactivo=0;

}

//~~Casos~~~~~~~~~~~~~~~~~~~~~~~~

void casoQ(){

IPC0bits.INT0IP= 6; IPC5BITS.INT1IP=5;

IPC7bits.INT2IP =4; IPC13bits.INT3IP=3;

IPC13bits.INT4IP=2; Glcd\_Fill(0);

casoQactivo=1;

do{

down=0;special=0;keydata=0;

Ps2\_Key\_Read(&keydata, &special, &down);

}while(keydata!=34 && !down & !special);

disable\_INT();

Glcd\_Fill(0);

disable\_INT();

}

void casoW(){

valores\_cn[0]=0; valores\_cn[1]=0;

valores\_cn[2]=0; valores\_cn[3]=0;

Glcd\_Fill(0) ;

Glcd\_Write\_TEXT("Caso 2",0,0,1);

delay\_ms(1000);

do{

Glcd\_Fill(0);

down=0;special=0;keydata=0;

if(valores\_cn[0]==1){

animate\_charmander\_5s();

}

else if(valores\_cn[1]==1){

animate\_kirby\_5s(0,0);

}

else if(valores\_cn[2]==1){

animate\_shell\_5s(0,0);

}

else if(valores\_cn[3]==1){

animate\_bullet\_5s()

}

Ps2\_Key\_Read(&keydata, &special, &down);

}while(keydata!=34 && !down & !special);

}

void casoE(){

Glcd\_Fill(0);

Glcd\_Write\_TEXT("Caso E",60,0,1);

do{

down=0;

special=0;

if (RCONbits.WDTO==1){

Glcd\_Write\_TEXT("WDT",0,1,1);

RCONbits.WDTO=0;

PORTEbits.RE5=1;

animate\_dog\_20s();

}else if(RCONbits.EXTR==1){

Glcd\_Write\_TEXT("MCLR",0,2,1);

RCONbits.EXTR=0;

RCONbits.POR=0;

PORTEbits.RE7=1;

animate\_blooper\_20s();

continue;

}else if (RCONbits.POR==1){

Glcd\_Write\_TEXT("POR",0,3,1);

RCONbits.POR=0;

PORTGbits.RG6=1;

animate\_charmander\_20s();

break;

}

Ps2\_Key\_Read(&keydata, &special, &down);

}while(keydata!=34&&!down&&!special);

LATEbits.LATE5=0;

LATEbits.LATE7=0;

LATGbits.LATG6=0;

// PORTGbits.RG6=0;

// PORTEbits.RE7=0;

// PORTEbits.RE5=0;

}

void main(){

config\_IO(); config\_LCD();

config\_INT(); config\_CN();

animate\_charmander\_5s();

PS2\_Config(); Glcd\_Fill(0);

while(1){

Glcd\_Write\_TEXT("Laboratorio 1",31,0,1);

Glcd\_Write\_TEXT("'Q' para Caso Q",0,1,1);

Glcd\_Write\_TEXT("'W' para Caso W",0,2,1);

Glcd\_Write\_TEXT("'E' para Caso E",0,3,1);

Glcd\_Write\_TEXT("'R' para WDT ",0,4,1);

if(Ps2\_Key\_Read(&keydata, &special, &down)){

if(down &&!special){

switch(keydata){

case 'q':

case 'Q':

Glcd\_Write\_TEXT("Caso Q",60,0,1);

casoQ();

break;

case 'w':

case 'W':

Glcd\_Write\_TEXT("Caso W",60,0,1);

casoW();

Glcd\_Fill(0);

break;

case 'e':

case 'E':

Glcd\_Fill(0);

Glcd\_Write\_TEXT("Ultimo Reset",60,0,1);

casoE();

Glcd\_Fill(0);

break;

default:

Glcd\_Fill(0);

Glcd\_Write\_TEXT("Erroneo ",60,0,1);

delay\_ms(2000);

break;

}

}

}

}

}

**BARRAS.H**

const int BAR\_HEIGHT=2;

const int BAR\_WIDTH=20;

const int WHITE=0;

const int BLACK=1;

const int MAX\_BAR\_HEIGHT=BAR\_HEIGHT\*25;

unsigned short BARS\_H\_LOCATIONS[5]={4,29,54,79,104};

char txt[5];

unsigned short x\_left,x\_right,y\_top,y\_bottom,count,contador\_reset,Ix;

int quotient;

void bar\_drawer(unsigned short x\_left,unsigned short contador\_barra);

int DrawableBars(int real\_number\_of\_bars);

void box(int x\_left,int count);

void barra(int INTx,int contador\_barra) {

// void Glcd\_Box(unsigned short x\_upper\_left, unsigned short y\_upper\_left, unsigned short x\_bottom\_right, unsigned short y\_bottom\_right, unsigned short color);

box(BARS\_H\_LOCATIONS[INTx],contador\_barra);

ShortToStr(contador\_barra,txt);

// txt=Ltrim(txt);

Glcd\_Write\_Text(txt,BARS\_H\_LOCATIONS[INTx],0,1);

}

void box(int x\_left,int count){

y\_top=64-DrawableBars(count)\*2;

y\_bottom=y\_top+2;

x\_right=x\_left+20;

contador\_reset=DrawableBars(count);

if(count>1){

if (contador\_reset==1){

Glcd\_Box(x\_left,7,x\_right,64,0);

// Glcd\_Box(x\_left,64,x\_right,62, 1);

} else {

Glcd\_Box(x\_left,y\_top,x\_right,y\_bottom, 1);}

}else {

Glcd\_Box(x\_left,y\_top,x\_right,y\_bottom, 1);

}

}

int DrawableBars(int real\_number\_of\_bars){

//Funcion para devolver el numero necesario de barras a formar si el contador>25 dentro de un rango de [0-25].

//Ejemplo 80-->5

if(real\_number\_of\_bars>25){

quotient=real\_number\_of\_bars/25;

quotient=floor(quotient);

return real\_number\_of\_bars-25\*quotient;

}else{

return real\_number\_of\_bars;

}

}

**SPRITES.H**

void animate\_charmander\_5s(){

Glcd\_Image(charmander\_1);

delay\_ms(2500);

Glcd\_Image(charmander\_2);

delay\_ms(2500);

}

void animate\_charmander\_20s(){

animate\_charmander\_5s();

animate\_charmander\_5s();

animate\_charmander\_5s();

animate\_charmander\_5s();

}

void animate\_bullet\_5s(){

Glcd\_PartialImage(100,23,18,18,18,18,bullet); delay\_ms(830);Glcd\_Fill(0);

Glcd\_PartialImage(80,23,18,18,18,18,bullet); delay\_ms(830);Glcd\_Fill(0);

Glcd\_PartialImage(60,23,18,18,18,18,bullet); delay\_ms(830);Glcd\_Fill(0);

Glcd\_PartialImage(40,23,18,18,18,18,bullet); delay\_ms(830);Glcd\_Fill(0);

Glcd\_PartialImage(20,23,18,18,18,18,bullet); delay\_ms(830);Glcd\_Fill(0);

Glcd\_PartialImage(0,23,18,18,18,18,bullet); delay\_ms(830);Glcd\_Fill(0);

}

void animate\_shell(unsigned int x\_left, unsigned int y\_top,code const far char \* image){

Glcd\_PartialImage(x\_left,y\_top,16,16,16,16,image);

}

void animate\_shell\_5s(unsigned int x\_left, unsigned int y\_top){

Glcd\_Fill(0);

animate\_shell(x\_left,y\_top,shell\_1);

delay\_ms(1667);

animate\_shell(x\_left,y\_top,shell\_2);

delay\_ms(1667);

animate\_shell(x\_left,y\_top,shell\_1);

delay\_ms(1667);

}

void animate\_dog(unsigned int x\_left, unsigned int y\_top,code const far char \* image){

Glcd\_PartialImage(x\_left,y\_top,31,40,31,40,image);

}

void animate\_dog\_5s(unsigned int x\_left, unsigned int y\_top){

Glcd\_Fill(0);

Glcd\_Write\_TEXT("WDT",60,0,1);

// animate\_shell(x\_left,y\_top,shell\_1);

animate\_dog(x\_left,y\_top,dog\_1);

delay\_ms(1667);

animate\_dog(x\_left,y\_top,dog\_2);

delay\_ms(1667);

animate\_dog(x\_left,y\_top,dog\_3);

delay\_ms(1667);

}

void animate\_dog\_20s(){

animate\_dog\_5s(48,12);animate\_dog\_5s(48,12);animate\_dog\_5s(48,12);animate\_dog\_5s(48,12);

}

void animate\_kirby(unsigned int x\_left, unsigned int y\_top,code const far char \* image){

Glcd\_PartialImage(x\_left,y\_top,21,19,21,19,image);

}

void animate\_kirby\_5s(int x\_left,int y\_top){

Glcd\_Fill(0);

animate\_kirby(x\_left,y\_top,kirby\_1);delay\_ms(833);

animate\_kirby(x\_left,y\_top,kirby\_2) ;delay\_ms(833);

animate\_kirby(x\_left,y\_top,kirby\_3) ;delay\_ms(833);

animate\_kirby(x\_left,y\_top,kirby\_4) ;delay\_ms(833);

animate\_kirby(x\_left,y\_top,kirby\_5) ;delay\_ms(833);

animate\_kirby(x\_left,y\_top,kirby\_6);delay\_ms(833);

}

void animate\_blooper(unsigned int x\_left, unsigned int y\_top,code const far char \* blooper){

Glcd\_PartialImage(x\_left,y\_top,35,34,35,34,blooper);

}

void animate\_blooper\_5s(int x\_left,int y\_top){

Glcd\_Fill(1);

Glcd\_Write\_TEXT("MCLR",60,0,0);

animate\_blooper(x\_left,y\_top,blooper\_1);

delay\_ms(1000);

animate\_blooper(x\_left,y\_top,blooper\_2);

delay\_ms(1000);

animate\_blooper(x\_left,y\_top,blooper\_1);

delay\_ms(1000);

animate\_blooper(x\_left,y\_top,blooper\_2);

delay\_ms(1000);

}

void animate\_blooper\_20s(){

animate\_blooper\_5s(46,15);

animate\_blooper\_5s(46,15);

animate\_blooper\_5s(46,15);

animate\_blooper\_5s(46,15);

}