TX

#include "hid.h"

#include "adc.h"

#include "config\_TX.h"

#define ESC\_key 254

#define PARTE\_ALTA 0x3FC

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

void buffer\_caso1();void caso2();void caso3();void caso1();void UART\_ESC();

void caso2\_check();

//~~~~~~~~~~~~~~~~~~~~~~~~Variables del sistema~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

int i=0,j=0,escape=0,cnt,mot1=0,mot2=0,bandera1=0,bandera2=0,rpm1=0,rpm2=0;

float pantalla=0;

char readbuff[64];

char writebuff[64];

unsigned short enviado=0;

unsigned caso1\_val[6]={0,0,0,0,0,0};

unsigned short adc\_value1,adc\_value2,pote1=0,pote2=0,pote3=0;

char txt [30];

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Interrupciones~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// void timer8 () org 0x7A {

// IFS3bits.T8IF=0;

// LATFBITS.LATF5=~LATFBITS.LATF5;

// }

void USB1Interrupt() iv IVT\_ADDR\_USB1INTERRUPT{

USB\_Interrupt\_Proc();

}

void INT\_Inicio\_Conversion\_T3 () org 0x24 {IFS0bits.T3IF=0;}//inicializa conversion del adc

void INT\_ADC() org 0x2E {

IFS0bits.AD1IF=0;

adc\_value1=ADC1BUF0;

adc\_value2=ADC1BUF1;

pote1=adc\_value1;

pote2=adc\_value2;

delay\_ms(10);

enviado=pote2;

UART1\_Write(enviado); delay\_ms(10);

enviado=pote1;

UART1\_Write(enviado); delay\_ms(10);

bandera1++;

}

void config\_vref () {

CM1CONbits.CON=1;

CVRCONbits.CVREN=1;

CVRCONbits.CVROE=1;

CVRCONbits.BGSEL=1;

CVRCONbits.CVRR=0;

ANSELBbits.ANSB10=1;

TRISBBITS.TRISB10=0;

CVRCONbits.CVR=15;

CVRCONbits.VREFSEL=0;

}

void cambio\_led () {

LATFBITS.LATF5=~LATFBITS.LATF5;

}

void interupcion\_ext1() org 0x3C {

IFS1bits.INT1IF=0;

rpm1++;

}

void interupcion\_ext2() org 0x4E {

IFS1bits.INT2IF=0;

rpm2++;

}

void timer7 () org 0x74 {

IFS3bits.T7IF=0;

caso1\_val[0]=rpm1\*75;

caso1\_val[1]=rpm2\*75;

rpm1=0;

rpm2=0;

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~MENU~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void main() {

cambio\_led();

config\_INT();//config\_timer8();

InitMCU();

config\_timer3();

config\_adc();

config\_vref();

config\_pin();

UART1\_Init(9600);

HID\_Enable(&readbuff,&writebuff); //inicializamos en módulo usb hid

while(1){

menu2();

while(!HID\_Read());

for(cnt=0;cnt<64;cnt++) {

writebuff[cnt]=readbuff[cnt];

}

if(strcmp(readbuff,caso\_1)==0){//CASE 1

caso1();

}

else if(strcmp(readbuff,caso\_2)==0){//CASE 2

write("Bienvenido al Caso 2");

caso2();

}

else if(strcmp(readbuff,caso\_3)==0){//CASE 3

caso3();

}

Delay\_ms(1000);

escape=0;

}

}

void caso1(){ +

bandera1=0;

AD1CON1bits.ADON=1;// Se act el modulo

T3CONbits.TON=1; // activa timer 3 para inicio de conver.

enviado=1;

UART1\_Write(enviado);

//hid\_caso\_1(caso1\_val[0],caso1\_val[1],caso1\_val[2],caso1\_val[3],caso1\_val[4],caso1\_val[5],pote1,pote2);

delay\_ms(150);

config\_velocidad ();

while(escape==0){

if (bandera1==3) {

buffer\_caso1();

hid\_caso\_1(caso1\_val[0],caso1\_val[1],

caso1\_val[2],caso1\_val[3],

caso1\_val[4],caso1\_val[5],

pote1,pote2);

bandera1=0;

}

if(HID\_Read()){

escape++;UART\_ESC();}

}

AD1CON1bits.ADON=0;// Se act el modulo

T3CONbits.TON=0;

T7CONbits.TON=0;

IPC5BITS.INT1IP=0;

IPC7bits.INT2IP=0;

}

void buffer\_caso1(){

if(pote1<128){caso1\_val[2]=IZQ;}

else {caso1\_val[2]=DER;}

if(pote2<128){caso1\_val[3]=IZQ;}

else {caso1\_val[3]=DER;}

caso1\_val[4]=FALLA;

caso1\_val[5]=FALLA;

}

void caso2(){

T7CONbits.TON=0;

config\_CM();

enviado=2;

UART1\_Write(enviado);

delay\_ms(100);

escape=0;

while(escape==0){

if(HID\_Read()){

escape++;UART\_ESC();}

else{

caso2\_check();

}

}

}

void caso3(){

hid\_caso\_3();

enviado=3;

UART1\_Write(enviado);

delay\_ms(100);

while(escape==0){

while(!HID\_Read());

for(cnt=0;cnt<64;cnt++) {

writebuff[cnt]=readbuff[cnt];

}

switch(readbuff[0]){

case '0':enviado=0; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '1':enviado=1; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '2':enviado=2; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '3':enviado=3; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '4':enviado=4; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '5':enviado=5; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '6':enviado=6; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '7':enviado=7; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '9':enviado=8; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case '0':enviado=9; caso3\_1[28]=readbuff[0]; hid\_caso\_3(); UART1\_Write(enviado);break;

case 'A':case 'a':enviado=10; caso3\_1[28]='A'; hid\_caso\_3(); UART1\_Write(enviado);break;

case 'B':case 'b':enviado=11; caso3\_1[28]='B'; hid\_caso\_3(); UART1\_Write(enviado);break;

case 'C':case 'c':enviado=12; caso3\_1[28]='C'; hid\_caso\_3(); UART1\_Write(enviado);break;

case 'D':case 'd':enviado=13; caso3\_1[28]='D'; hid\_caso\_3(); UART1\_Write(enviado);break;

case 'E':case 'e':enviado=14; caso3\_1[28]='E'; hid\_caso\_3(); UART1\_Write(enviado);break;

case 'F':case 'f':enviado=15; caso3\_1[28]='F'; hid\_caso\_3(); UART1\_Write(enviado);break;

default:escape++;UART\_ESC();break;

}

}

}

void UART\_ESC(){

delay\_ms(10);

enviado=ESC\_key;

UART1\_Write(enviado);

}

int value;

char txt4[5];char txt3[4];

void floattostr3(const float valor,char \*output) {

int i11,i2,d1,d2,d3;

float n0;

char txt2[15];

i11 = valor\*0.1;

i2 = valor-(10\*i11);

bytetostr(i2, txt2);

txt4[0]= ltrim(txt2);

txt4[1]='.';

n0 = (valor - 10\*i11 - i2)\*10000;

d1 = n0/1000;

bytetostr(d1, txt2);

txt4[2]=ltrim(txt2);

d2 = (n0 - d1\*1000)/100;

bytetostr(d2, txt2);

txt4[3]=ltrim(txt2);

d3 = (n0 - d1\*1000 - d2\*100)/10;

bytetostr(d3, txt2);

txt4[4]=ltrim(txt2);

output[0]=txt4[0];

output[1]=txt4[1];

output[2]=txt4[2];

output[3]=txt4[3];

output[4]=txt4[4];

}

void caso2\_check(){

if(CM3CONbits.COUT==0){

if (bandera2==0){

bandera2=1;

write("Voltaje en limite superior");

enviado=1;

UART1\_Write(enviado);

space2();

space2();

space2();

space2();

}

}else if(CM1CONbits.COUT==0){

if (bandera2==0) {

bandera2=2;

write("Voltaje en limite inferior");

enviado=2;

UART1\_Write(enviado);

space2();

space2();

space2();

space2();

}

}else if(CM1CONbits.COUT==1&&CM3CONbits.COUT==1) {

if (bandera2<3&&bandera2>0) {

bandera2=3;

write("Voltaje Normal");

enviado=3;

UART1\_Write(enviado);

space2();

space2();

space2();

space2();

}

else bandera2=0;

}

}

HID

#define ESC 27

#define DER 0

#define IZQ 1

#define FALLA 1

#define NOFALLA 0

char menu[] = "Menu";

char Bien1[] = "\r Bienvenido al caso 1 \x0a \x0d";

char Bien2[] = "\n Bienvenido al caso 2 \x0a \x0d";

char Space[]="\n \x0a \x0d" ;

char Bien3[] = " Bienvenido al caso 3";

char caso\_1[] = "A",caso\_2[] = "B",caso\_3[] = "C";

char POT1[] = " POT 1: \r",POT2[] = " POT 2: \r";

char MOTO1[] = " MOT 1 ACT ",MOTO2[] = " MOT 2 ACT ";

char error1[]= "\r POT 1 HA SUPERADO\n \r";

char error2[]= "\r POT 2 HA SUPERADO \n \r";

char Bien5[] = "\r Bienvenido al caso 5 \r";

char caso1\_1[]="VEL(RPN) ###### | ####";

char caso1\_2[]="SENT ### | ###";

char caso1\_3[]="FALLA ## | ##";

char caso1\_4[]="POT ## | ##";

char caso2\_nor[]="Todo esta funcionando bien | ";

char caso2\_sup[]="Limite Superior | ";

char caso2\_inf[]="Limite Inferior | ";

char caso2\_pot[]= "POT ## | ";

char caso3\_1[]="Salida por el PIN AN18 | OP=#";

char caso3\_2[]="V |0.83 0.93 1.03 1.13 1.24 1.34 1.44 1.55|";

char caso3\_3[]="OP| 0 1 2 3 4 5 6 7|";

char caso3\_4[]="V |1.65 1.75 1.86 1.96 2.06 2.17 2.27 2.37|";

char caso3\_5[]="OP| 8 9 A B C D E F|";

int cont = 0,it=0;

char txt7[7];

void space1(){

while(!HID\_Write(space,64));

}

void space2(){

while(!HID\_Write(space,64));

while(!HID\_Write(space,64));

}

void write(char \*txt){

space2();

while(!HID\_Write(txt,64));

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~HID MENU~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void menu2(){

write(menu);

write("--------------------------------------------------");

write("CASO A MOTORES");

write("CASO B COMPARACION");

write("CASO C DAC");

space1();

space1();

space1();

space1();

space1();

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~HID CASO1~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void update\_caso1(unsigned rpn1,unsigned rpn2){

inttostr(rpn1,txt7);

caso1\_1[9] =txt7[0];

caso1\_1[10]=txt7[1];

caso1\_1[11]=txt7[2];

caso1\_1[12]=txt7[3];

caso1\_1[13]=txt7[4];

caso1\_1[14]=txt7[5];

inttostr(rpn2,txt7);

caso1\_1[45]=txt7[1];

caso1\_1[46]=txt7[2];

caso1\_1[47]=txt7[3];

caso1\_1[48]=txt7[4];

caso1\_1[49]=txt7[5];

}

void update\_caso3(unsigned falla1,unsigned falla2){

//falla=1 si & falla=0 no

if(falla1){

caso1\_3[13]='S';

caso1\_3[14]='I';

}else{

caso1\_3[13]='N';

caso1\_3[14]='O';

}

if(falla2){

caso1\_3[48]='S';

caso1\_3[49]='I';

}else{

caso1\_3[48]='N';

caso1\_3[49]='O';

}

}

void update\_caso4(unsigned pot1,unsigned pot2){

inttostr(pot1,txt7);

caso1\_4[9] =txt7[0];

caso1\_4[10]=txt7[1];

caso1\_4[11]=txt7[2];

caso1\_4[12]=txt7[3];

caso1\_4[13]=txt7[4];

caso1\_4[14]=txt7[5];

inttostr(pot2,txt7);

caso1\_4[45]=txt7[1];

caso1\_4[46]=txt7[2];

caso1\_4[47]=txt7[3];

caso1\_4[48]=txt7[4];

caso1\_4[49]=txt7[5];

}

void update\_caso2(unsigned sent1,unsigned sent2){

//sent=1 izquierda & sent=0 derecha

if(sent1){

caso1\_2[12]='I';

caso1\_2[13]='Z';

caso1\_2[14]='Q';

}else{

caso1\_2[12]='D';

caso1\_2[13]='E';

caso1\_2[14]='R';

}

if(sent2){

caso1\_2[47]='I';

caso1\_2[48]='Z';

caso1\_2[49]='Q';

}else{

caso1\_2[47]='D';

caso1\_2[48]='E';

caso1\_2[49]='R';

}

}

void logd(char \*txt){

write(txt);

space1();

space1();

space1();

space1();

space1();

space1();

space1();

space1();

space1();

}

void hid\_caso\_1(unsigned rpn1,unsigned rpn2,

unsigned sent1,unsigned sent2,

unsigned falla1,unsigned falla2,

unsigned pot1,unsigned pot2){

update\_caso1(rpn1,rpn2);

update\_caso2(sent1,sent2);

update\_caso3(falla1,falla2);

update\_caso4(pot1,pot2);

write(" MOTOR 1 | MOTOR 2");

write("----------------------------|---------------------");

write(caso1\_1);

write(caso1\_2);

write(caso1\_3);

write(caso1\_4);

space1();

space1();

space1();

space1();

space1();

}

void hid\_caso\_3(){

write(Bien3);

write("----------------------------|---------------------");

write(caso3\_1);

write(caso3\_2);

write(caso3\_3);

write(caso3\_4);

write(caso3\_5);

space1();

space1();

space1();

space1();

}

void hid\_caso\_2(unsigned pot){

write(Bien2);

if(pot<1000){

write(caso2\_inf);

}else if(pot>1000){

write(caso2\_sup);

}else{

write(caso2\_nor);

}

inttostr(pot,txt7);

caso2\_pot[9] =txt7[0];

caso2\_pot[10]=txt7[1];

caso2\_pot[11]=txt7[2];

caso2\_pot[12]=txt7[3];

caso2\_pot[13]=txt7[4];

caso2\_pot[14]=txt7[5];

write(caso2\_pot);

space1();

space1();

space1();

space1();

space1();

space1();

space1();

space1();

space1();

}

ADC.H

void config\_timer3 () {

TMR3=0;

T3CON=0X0030; //Prescaler 64:1, modo timer

PR3=65000; //1000ms

IEC0bits.T3IE=1;

IFS0bits.T3IF=0;

}

void config\_adc() {

AD1CON1bits.AD12B=0;// ADC de 10 bits

AD1CON1bits.FORM=0; // ENTERO SIN SIGNO

AD1CON1bits.SSRCG=0;

AD1CON1bits.SSRC=2;//TIMER3 inicia la convercion

AD1CON1bits.SIMSAM=1;// Muestreo simultaneo

AD1CON1bits.ASAM=1;

AD1CON1bits.SAMP=0;

AD1CON2bits.VCFG=0;// AVdd y AVss como referencias

AD1CON2bits.CSCNA=0;

AD1CON2bits.CHPS=1;//CH0 y CH1 Act.

AD1CON2bits.SMPI=0;// interrupcion a la primera conversion

AD1CON2bits.BUFM=0; //llenado desde la primera direccion

AD1CON2bits.ALTS=0;

AD1CON3bits.ADRC=0; //reloj interno Tcy

AD1CON3bits.SAMC=11; //10TAD del tiempo de muestreo

AD1CON3bits.ADCS=0; //TAD=Tcy

AD1CON4bits.ADDMAEN=0; //SIN DMA

AD1CHS123bits.CH123NA=0; //entrada negativa es AVss

AD1CHS123bits.CH123SA=0;// CH1 entrada positiva AN0(RB0)

AD1CHS0bits.CH0NA=0; //entrada negativa es AVss

AD1CHS0bits.CH0SA=3;//ENTRADA positiva AN4(4)

IFS0bits.AD1IF=0;//bandera=0

IEC0bits.AD1IE=1;//hab interrupcion

IPC3bits.AD1IP=6;

delay\_ms(10);

}

CONFIG\_TX

#define FP 15000000

#define BAUDRATE 9600

#define BRGVAL ((FP/BAUDRATE)/16)-1

unsigned char UENVIAR;

void CONFIG\_CM(){

ANSELDbits.ANSD7=1;//ENTRADA ANALOGICA c3in1+ 3V pin 55

ANSELCbits.ANSC14=1;//ENTRADA ANALOGICA c3in1- POT pin 48

ANSELBbits.ANSB5=1;//ENTRADA ANALOGICA c1in+ POT RB5 pin 11

CM3CONbits.COE=0;

CM1CONbits.COE=0;

CM3CONbits.CON=1;

CM1CONbits.CON=1;

CM3CONbits.CPOL=0;

CM1CONbits.CPOL=0;

CM3CONBITS.CREF=0;//CONECTADO A C3IN1+

CM3CONBITS.CCH=1;//CONECTADO A C3IN1-

CM1CONBITS.CREF=0;//CONECTADO A C1IN1+

CM1CONBITS.CCH=11;////CONECTADO A INTREF

CVRCONbits.BGSEL=1;

CVRCONbits.CVREN=1;

CM1CONbits.CEN=1;

CM3CONbits.CEN=1;

}

void InitMCU(){

ANSELC=0x0000; //Configuracion de E/S digitales

ANSELD=0x0000; //Configuracion de E/S digitales

ANSELE=0x0040; //RE6 como entrada analogica

ANSELB=0x0000; //Configuracion de E/S digitales

//CONFIGURACION DEL PLL PARA ALCANZAR UNA VELOCIDAD DE 30MHZ

PLLFBD = 58; //M = 60

CLKDIVbits.PLLPOST = 0; // N1 = 2

CLKDIVbits.PLLPRE = 0; // N2 = 2

OSCTUN = 0;

OSCCON=0x0301;

while (OSCCONbits.COSC != 0x3);

//CONFIGURACION DEL PLL AUXILIAR PARA EL USB

//SE REQUIEREN 48MHZ

ACLKCON3 = 0x24C0;

ACLKDIV3 = 0x7;

ACLKCON3bits.ENAPLL = 1;

while(ACLKCON3bits.APLLCK != 1);

}

void config\_TX(){

U1MODEbits.STSEL=0;

U1MODEbits.PDSEL=0;

U1MODEbits.ABAUD=0;

U1MODEbits.BRGH=0;

U1BRG=BRGVAL;

U1STAbits.UTXISEL1=0;

IEC0BITS.U1TXIE=1;

U1MODEbits.UARTEN=1;

delay\_us(105);

IFS0bits.U1TXIF=0;

IEC0bits.U1TXIE=1;

IPC3bits.U1TXIP=6;

}

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

IPC14bits.QEI1IP=2; // interrupcion del modulo cuadratura 2

IPC2bits.T3IP=5;

IPC6bits.T4IP=3;

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

}

void config\_timer8() {

tmr8=0;

pr8=31250;

t8con=0x8020;

IFS3bits.T8IF=0;

IEC3bits.T8IE=1;

IPC12bits.T8IP=7;

T8CONbits.TON=1;

}

void config\_pin () {

ANSELBbits.ANSB0=1; // RBO analogico

ANSELBbits.ANSB3=1; // RB4 analogico

TRISBbits.TRISB0=1;

TRISBbits.TRISB3=1;

TRISDbits.TRISD9=1; // entrada opto para motor 1

TRISDbits.TRISD8=1; // entrada opto para motor 2

RPINR0bits.INT1R=73; // RPI73 en INT1 motor 1

RPINR1bits.INT2R=72; // RPI72 en INT2 motor 2

RPOR0bits.RP64R=1; //U1TX

RPINR18bits.U1RXR=72; //U1RX

TRISDbits.TRISD10=0;

TRISFBITS.TRISF5=0;

}

void config\_velocidad () {

IFS1bits.INT2IF=0;

INTCON2bits.INT1EP=0; //flanco positivo

INTCON2bits.INT2EP=0; //flanco positivo

IEC1bits.INT1IE=1;

IEC1bits.INT2IE=1;

IPC5BITS.INT1IP=4;

IPC7bits.INT2IP =4;

}

RX

#include "config\_RX.h"

#include "motores.h"

#include "sprites.h"

#define ESC\_key 254

//~~~~~~~~~~~~~~~~~~~~~~~~Variables del sistema~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

unsigned short dato=0, dato2=0;

int pantalla=0;

char txt[7]={'#','#','#','#','#','#','#'};

int cnt,rpm1=0,rpm2=0;

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

void caso2();void caso3();void caso1();

void encender\_led(){LATFBITS.LATF4=~LATFBITS.LATF4;}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Interrupciones~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

/\*void timer8 () org 0x7A {

IFS3bits.T8IF=0;

rpm=0;

rpm2=0;

}\*/

void INT\_T05s\_T2()org 0x22{

T05s++;

inttostr(T05s,txt);

glcd\_write\_text(txt,0,3,1);

if(T05s%2==0){T1s++;}

if(T05s>6){T2CONbits.TON=0;T1s=0;T05s=0;}

if(caso!=CASE\_NULL&&T1s==3){T1s=0;}

else if(caso==CASE\_NULL&&T1s==2){T1s=0;}

selector\_sprite(caso,T1s);

IFS0bits.T2IF=0;

}

void interupcion\_ext1() org 0x3C {

IFS1bits.INT1IF=0;

rpm1++;

}

void interupcion\_ext2() org 0x4E {

IFS1bits.INT2IF=0;

rpm2++;

}

void timer7 () org 0x74 {

IFS3bits.T7IF=0;

rpm1=rpm1\*75;

rpm2=rpm2\*75;

inttostr(rpm1,txt);

glcd\_write\_text(txt,64,0,1);

inttostr(rpm2,txt);

glcd\_write\_text(txt,64,1,1);

rpm1=0;

rpm2=0;

}

void PWM4() org 0xD6{

IFS6bits.PWM4IF=0;

PWMCON3bits.FLTSTAT=0;

if(PORTDbits.RD11==0){

glcd\_write\_text("Falla Motor 4",64,7,1);

IOCON4bits.FLTDAT=0;

PWMCON4bits.FLTSTAT=0;

//caso=CASE\_PWM4; T2CONbits.TON=1;

}

}

void PWM3() org 0xD4{

IFS6bits.PWM3IF=0;

PWMCON3bits.FLTSTAT=0;

if(PORTDbits.RD0==0){

IOCON4bits.FLTDAT=0;

PWMCON4bits.FLTSTAT=0;

// caso=CASE\_PWM3; T2CONbits.TON=1;

glcd\_write\_text("Falla Motor 3",64,6,1);

}

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~MAIN~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void main () {

InitMCU();config\_IO();config\_vref();

config\_LCD(); delay\_ms(50);

config\_INT(); UART1\_Init(9600);

// config\_timer8();

config\_TMR2\_ANIM ();

//animate\_charmander\_2s(); Glcd\_Fill(0);

//encender\_led();

while(1){

encender\_led();

glcd\_write\_text("Laboratorio 4",64,0,1);

glcd\_write\_text("Esperando Comando",64,1,1);

delay\_ms(100);

while(!UART1\_Data\_Ready()); //Espera que reciba un dato

dato=UART1\_Read();

pantalla=dato;

floattostr(pantalla,txt);

glcd\_fill(0);

glcd\_write\_text(txt,60,0,1);

glcd\_write\_text("recibio algo",60,0,1);

if (dato>0&&dato<4) {

switch(dato){

case 1:

caso1();

PDC4=15000;

PDC3=15000;

T7CONbits.TON=0;

IPC5BITS.INT1IP=0;

IPC7bits.INT2IP=0;

Glcd\_Fill(0);

break;

case 2:

caso2();Glcd\_Fill(0);

break;

case 3:

caso3();Glcd\_Fill(0);

break;

default:

glcd\_write\_text("Error",64,4,1);

delay\_ms(100);

break;

}

dato=0;

}

}

}

void caso1(){

config\_motor();config\_timer7();config\_velocidad ();

//T8CONbits.TON=1;

glcd\_write\_text("Caso 1",64,2,1);

delay\_ms(100);

T7CONbits.TON=1; // activa conteo para RPM de motores

while(!UART1\_Data\_Ready());

while (dato!=ESC\_key){

encender\_led();

dato=UART1\_Read();

if (dato==ESC\_key){continue;}

PDC3=ajuste(dato);

while(!UART1\_Data\_Ready());

dato2=UART1\_Read();

if (dato2==ESC\_key){continue;}

PDC4=ajuste(dato2);

pantalla=PDC3;

inttostr(pantalla,txt);

glcd\_write\_text(txt,64,6,1);

pantalla=PDC4;

floattostr(pantalla,txt);

glcd\_write\_text(txt,64,7,1);

}

T8CONbits.TON=0;

dato=0;

}

void caso2(){

GLCD\_fill(0);

glcd\_write\_text("Caso 2",64,0,1);

while (dato!=ESC\_key){

dato=UART1\_Read();

if (dato==1) {

caso=CASE\_MAY;

T2CONbits.TON=1;

}

if (dato==2) {

caso=CASE\_MEN;

T2CONbits.TON=1;

}

if (dato==3) {

GLCD\_fill(0);

glcd\_write\_text("NORMAL",64,0,1);

}

}

}

void caso3(){

dato=0;

glcd\_write\_text("Caso 3",64,4,1);

delay\_ms(50);

while (dato!=ESC\_key){

while(!UART1\_Data\_Ready()); //Espera que reciba un dato

dato=UART1\_Read();

if(dato<16&&dato>=0){

pantalla=dato;

CVRCONbits.CVR=dato;

glcd\_write\_text(space,65,7,1);

floattostr(pantalla,txt);

glcd\_write\_text(txt,65,7,1);

dato=0;

}

}

dato=0;

}

CONFIG\_RX

//Variables de GLCD

sbit GLCD\_D0 at RD4\_bit;

sbit GLCD\_D1 at RD5\_bit;

sbit GLCD\_D2 at RD6\_bit;

sbit GLCD\_D3 at RD7\_bit;

sbit GLCD\_D4 at RF0\_bit;

sbit GLCD\_D5 at RF1\_bit;

sbit GLCD\_D6 at RE0\_bit;

sbit GLCD\_D7 at RE1\_bit;

sbit GLCD\_CS1 at LATE2\_bit;

sbit GLCD\_CS2 at LATE3\_bit;

sbit GLCD\_RS at LATD1\_bit;

sbit GLCD\_RW at LATD2\_bit;

sbit GLCD\_EN at LATD3\_bit;

sbit GLCD\_RST at LATC14\_bit;

sbit GLCD\_D0\_Direction at TRISD4\_bit;

sbit GLCD\_D1\_Direction at TRISD5\_bit;

sbit GLCD\_D2\_Direction at TRISD6\_bit;

sbit GLCD\_D3\_Direction at TRISD7\_bit;

sbit GLCD\_D4\_Direction at TRISF0\_bit;

sbit GLCD\_D5\_Direction at TRISF1\_bit;

sbit GLCD\_D6\_Direction at TRISE0\_bit;

sbit GLCD\_D7\_Direction at TRISE1\_bit;

sbit GLCD\_CS1\_Direction at TRISE2\_bit;

sbit GLCD\_CS2\_Direction at TRISE3\_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 TRISC14\_bit;

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

void config\_LCD(){

Glcd\_Init();

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

Glcd\_Fill(0);

}

void config\_IO(){

ANSELB=ANSELC=ANSELD=ANSELE=0;

TRISDbits.TRISD0=1; //Pines falla

TRISDbits.TRISD11=1; //Pines falla

TRISDbits.TRISD9=1; //entrada opto para motor 1

TRISDbits.TRISD8=1; //entrada opto para motor 2

TRISFbits.TRISF4=0; //LED de prueba;

RPOR9bits.RP101R=1; //U1TX

RPINR0bits.INT1R=73; // RPI73 en INT1 motor 1 RPM

RPINR1bits.INT2R=72; // RPI72 en INT2 motor 2 RPM

RPINR18bits.U1RXR=46; //U1RX

RPINR13bits.FLT4R=75; // pin de falla pwm4

RPINR13bits.FLT3R=64; //pin de falla pwm3

}

void config\_TMR2\_ANIM () {

TMR2=0;

T2CON=0x0030; //Prescaler 256:1, modo timer

PR2=65350; //1000ms

IEC0bits.T2IE=1;

IFS0bits.T2IF=0;

IPC1bits.T2IP=7;

}

void InitMCU(){

//CONFIGURACION DEL PLL PARA ALCANZAR UNA VELOCIDAD DE 30MHZ

PLLFBD = 58; //M = 60

CLKDIVbits.PLLPOST = 0; // N1 = 2

CLKDIVbits.PLLPRE = 0; // N2 = 2

OSCTUN = 0;

OSCCON=0x0301;

while (OSCCONbits.COSC != 0x3);

//CONFIGURACION DEL PLL AUXILIAR PARA EL USB

//SE REQUIEREN 48MHZ

ACLKCON3 = 0x24C0;

ACLKDIV3 = 0x7;

ACLKCON3bits.ENAPLL = 1;

while(ACLKCON3bits.APLLCK != 1);

}

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

// IEC3bits.T8IE=1;

IFS3bits.T7IF=0;

IEC3bits.T7IE=1;

// IPC12bits.T8IP=7;

IPC12bits.T7IP=5;

}

void config\_timer8() {

TMR8=0;

PR8=58594;

T8CON=0x0030;

}

void config\_timer7() {

TMR7=0;

PR7=23438;

T7CON=0x0030;

}

void config\_vref () {

//CM1CONbits.CON=1;

CVRCONbits.CVREN=1;

CVRCONbits.CVROE=1;

CVRCONbits.BGSEL=1;

CVRCONbits.CVRR=0;

CVRCONbits.CVR=2;

CVRCONbits.VREFSEL=0;

}

void config\_velocidad () {

IFS1bits.INT2IF=0;

INTCON2bits.INT1EP=0; //flanco positivo

INTCON2bits.INT2EP=0; //flanco positivo

IEC1bits.INT1IE=1;

IEC1bits.INT2IE=1;

IPC5BITS.INT1IP=4;

IPC7bits.INT2IP =4;

}

unsigned ajuste (unsigned control) {

if (control<=20) {

return 6500;

}else

if (control>20&&control<=60) {

return 10000;

} else

if (control>60&&control<=80) {

return 11000;

} else

if (control>80&&control<=90) {

return 12000;

} else

if (control>90&&control<=140) {

return 15000;

} else

if (control>140&&control<=160) {

return 19000;

} else

if (control>160&&control<=180) {

return 20000 ;

} else

if (control>180&&control<=210) {

return 21500;

} else

if (control>210&&control<=255) {

return 23800;

}

}

MOTORES

void config\_PWM();void config\_PWM3();

void config\_motor() {

config\_PWM(); config\_PWM3();

}

void config\_PWM(){

IEC6bits.PWM4IE=1; // interrupcion pwm4

IFS6bits.PWM4IF=0; // limpia bandera pwm4

IPC24bits.PWM4IP=6; // nivel de interrupcion pwm4

PTCONbits.PTEN=1;

PWMCON4bits.FLTSTAT=0; //bandera de pin de falla

PWMCON4bits.FLTIEN=1;

PWMCON4bits.ITB=1;

PWMCON4bits.CAM=1;

PWMCON4bits.IUE=1;

IOCON4bits.PENH=1;

IOCON4bits.PENL=1;

IOCON4bits.POLH=0;

IOCON4bits.POLL=0;

DTR4=1000;

ALTDTR4=1000;

PHASE4=30000; // periodo de 2ms

PDC4=20000; //100%

IOCON4bits.FLTDAT=0; // falla en pin pone bajo todo

FCLCON4=0x0015;//falla indep. act,pin falla 3,act en baj0

FCLCON4bits.FLTSRC=3; // pin de falla 4

}

void config\_PWM3(){

IEC6bits.PWM3IE=1; // interrupcion pwm4

IFS6bits.PWM3IF=0; // limpia bandera pwm4

IPC24bits.PWM3IP=6; // nivel de interrupcion pwm4

PTCONbits.PTEN=1;

PWMCON3bits.FLTSTAT=0; //bandera de pin de falla

PWMCON3bits.FLTIEN=1;

PWMCON3bits.ITB=1;

PWMCON3bits.CAM=1;

PWMCON3bits.IUE=1;

IOCON3bits.PENH=1;

IOCON3bits.PENL=1;

IOCON3bits.POLH=0;

IOCON3bits.POLL=0;

DTR3=1000;

ALTDTR3=1000;

PHASE3=30000; // periodo de 2ms

PDC3=20000; //100%

IOCON3bits.FLTDAT=0; // falla en pin pone bajo todo

FCLCON3=0x0015;//falla indep. act,pin falla 3,act en baj0

FCLCON3bits.FLTSRC=2; // pin de falla 3

}

SPRITES.H

void selector\_sprite(int caso,int cnt){

switch(caso){

case CASE\_PWM4:

if(cnt==1){ animate\_64b(pwm4a);}

else if(cnt==2){animate\_64b(pwm4b);}

else if(cnt==3){animate\_64b(pwm4c);}

break;

case CASE\_PWM3:

if(cnt==1){ animate\_64b(pwm3a);}

else if(cnt==2){animate\_64b(pwm3b);}

else if(cnt==3){animate\_64b(pwm4c);}

break;

case CASE\_MAY:

if(cnt==1){ animate\_64b(mayorquea);}

else if(cnt==2){animate\_64b(mayorqueb);}

else if(cnt==3){animate\_64b(mayorquec);}

break;

case CASE\_MEN:

if(cnt==2){animate\_64b(menorqueb);}

else if(cnt==3){animate\_64b(menorquec);}

else{ animate\_64b(menorquea);}

break;

case CASE\_NULL:

glcd\_write\_text("Error Animacion CASE\_NULL",64,4,1);

break;

default:

glcd\_write\_text("Error Animacion case",64,4,1);

break;

if(cnt==0){

glcd\_write\_text("Error Animacion T1s",64,4,1);

}

}

}

void animate\_charmander\_2s(){

Glcd\_Image(charmander\_1);

delay\_ms(500);

Glcd\_Image(charmander\_2);

delay\_ms(500);

}

char Space[]=" " ;

void animate\_64b(code const far char \* image){

glcd\_partialimage(64,0,64,64,64,64,image);delay\_ms(1000);

}