

CÓDIGO FUENTE MÓDULO MÁSTER

```
#include <18f2550.h> //Microcontrolador a utilizar
#fuses HS,NOWDT,NOPROTECT,NOLVP,NOMCLR,NOPBADEN,NODEBUG,USBDIV,VREGEN,PLL3,CPUDIV3
//Oscilador de 12MHz
#use delay(clock=4000000)
#include "usb_cdc.h" //Libreria para USB
#include <string.h>
#include <stdlib.h>
#include <STDLIBM.H>

#use rs232(baud=9600,xmit=pin_C6,rcv=pin_C7) //Configuración de RS232 para módulo
XBee

char datoR=' ';
int dato=0x00,cont=0,contUSB=0,i=0,j=0,msg=0,contDa=0,auxContDa=0,contrTCC=0;
int hhRut[50],mmRut[50];
char *hhRutc,*mmRutc;
char datoUSB;
char keypress[50]="";
int lenMSB=0x00,lenLSB=0x00,CRC=0x00;
char buffer[200],bufferUSB[200];
char
ruta,numControl,*dirMSB,*dirLSB,*segc,*minc,*horac,*diac,*mesc,*aaac,*ddRc,*mmRc,*aaR
c;
char *idRutaNc,*cantDatRutc;
int desMSB,desLSB, seg,min,hora,dia,mes,aaa,ddR,mmR,aaR,cantDatRut,sumaCRC=0;
int16 idRutaN;
char *ptr;

void EnviarXBee(int destinoMSB, int destinoLSB, int dat0, int dat1)
{
    lenLSB=0x08;
    CRC= 0xFF-((0x01+0x4D+destinoMSB+destinoLSB+0x01+dat0+dat1+'.')&0xFF);
    putc(0x7E); //Bit de inicio
    putc(lenMSB); //MSB Length
    putc(lenLSB); //LSB Length
    putc(0x01); //API identifier
    putc(0x4D); //Frame ID (M)
    putc(destinoMSB); //Destino MSB
    putc(destinoLSB); //Destino LSB
    putc(0x01); //options
    putc(dat0); //datos (R)
    putc(dat1); //datos(a)
    putc('.');
    putc(CRC); //Checksum
}

void EnviarDato2(int destinoMSB, int destinoLSB, int dat0, int dat1){
    //Pr(Pagar retraso),Bm(Borrar memoria) C# (num de Control)
    lenLSB=0x08;
    CRC= 0xFF-((0x01+0x4D + destinoMSB + destinoLSB + 0x01 + dat0 + dat1+'.')&0xFF);
    putc(0x7E); //Bit de inicio
    putc(lenMSB); //MSB Length
    putc(lenLSB); //LSB Length
    putc(0x01); //API identifier
    putc(0x4D); //Frame ID (M) 4D
    putc(destinoMSB); //Destino MSB
    putc(destinoLSB); //Destino LSB
    putc(0x01); //options
    putc(dat0); //datos (R) 8
    putc(dat1); //datos(a) 9
    putc('.');
}
```

```

    putc(CRC); //Checksum
}

void EnviarRuta(int destinoMSB,int destinoLSB,int dd,int mm,int aa,int16 idRuta,int
cantDat){ //Envia datos de Rn(Cargar Ruta n)
    sumaCRC=0x01+0x4D+destinoMSB+destinoLSB+0x01+'R'+ 't'+dd+mm+aa+idRuta+cantDat+'.';
    lenLSB=0x0D +(cantDat*2);
    //printf(usb_cdc_putc,"N%d %x",lenLSB,lenLSB);
    putc(0x7E); //Bit de inicio
    putc(lenMSB); //MSB Length
    putc(lenLSB); //LSB Length
    putc(0x01); //API identifier
    putc(0x4D); //Frame ID (M) 4D
    putc(destinoMSB); //Destino MSB
    putc(destinoLSB); //Destino LSB
    putc(0x01); //options
    putc('R'); //      8
    putc('t'); //      9
    putc(dd); //dia  10
    putc(mm); //mes  11
    putc(aa); //año  12
    putc(idRuta); //  13
    putc(cantDat); //  14
    for(i=0;i<cantDat;i++){
        putc(hhRut[i]);
        putc(mmRut[i]);
        sumaCRC=sumaCRC+hhRut[i]+mmRut[i];
    }
    putc('.');
    CRC= 0xFF-((sumaCRC)&0xFF);
    putc(CRC); //Checksum
}

```

```

void IgualarReloj(int destinoMSB,int destinoLSB,int ss,int mn,int hh,int dd,int
ms,int aa){
    lenLSB=0x0E;
    CRC= 0xFF-((0x01+0x4D+destinoMSB+destinoLSB+0x01+'I'+ 'r'+ss+mn+hh+dd+ms
+aa+'.')&0xFF);
    putc(0x7E); //Bit de inicio
    putc(lenMSB); //MSB Length
    putc(lenLSB); //LSB Length
    putc(0x01); //API identifier
    putc(0x4D); //Frame ID (M)
    putc(destinoMSB); //Destino MSB
    putc(destinoLSB); //Destino LSB
    putc(0x01); //options
    putc('I'); //I  8
    putc('r'); //r  9
    putc(ss); //ss  10
    putc(mn); //mn  11
    putc(hh); //hh  12
    putc(dd); //dd  13
    putc(ms); //ms  14
    putc(aa); //aa  15
    putc('.');
    putc(CRC); //Checksum
}

```

```

#int_RDA
void RDA_isr()
{
    if(kbhit())
    {
        dato=getc();
    }
}

```

```

    if (dato==0x7E)
        cont=0;
    buffer[cont]=dato;
    if (dato=='.') {
        if ( buffer[8]=='C') {
            printf(usb_cdc_putc, "PC#%c ",buffer[9]);
        }
        if ( buffer[8]=='N') {
            contDa=buffer[9];
            printf(usb_cdc_putc, "N%d", contDa);
            auxContDa=1;
        }
        if (buffer[8]=='T')
            printf(usb_cdc_putc, "T");
        if (buffer[8]=='P')
            printf(usb_cdc_putc, "P");
        if (buffer[8]=='B')
            printf(usb_cdc_putc, "B");
        if (buffer[8]=='L')
            printf(usb_cdc_putc, "L");

        if (buffer[8]=='D') {
printf(usb_cdc_putc, "U%02xR%02dC%02dF20%02x%02x%02xH%02x%02xD%c",buffer[5],buffer[10]
,buffer[11],buffer[12],buffer[13],buffer[14],buffer[15],buffer[16],buffer[17]);
            if (auxContDa==contDa)
                printf(usb_cdc_putc, "FIN");
            auxContDa++;
        }
    }
    cont++;
}
}
#int_rtcc
void rtcc_isr()
{
    contrTCC++;
    if (contrTCC==250)
    {
        contrTCC=0;
        cont++;
        EnviarXBee (0xFF,0xFF, 'C', '0');
    }
    set_rtcc(217);
}
void main()
{
    delay_ms(100);
    setup_timer_0 (RTCC_INTERNAL|RTCC_DIV_256|RTCC_8_BIT);
    set_timer0(217);
    enable_interrupts (GLOBAL);
    enable_interrupts (INT_RTCC);
    enable_interrupts (INT_RDA);
    usb_cdc_init();
    usb_init();
    cont=0;

    segc=malloc(2);
    minc=malloc(2);
    horac=malloc(2);
    diac=malloc(2);
    mesc=malloc(2);
    aaac=malloc(2);
    ddRc=malloc(2);
    mmRc=malloc(2);

```

```

aaRc=malloc(2);
dirMSB=malloc(2);
dirLSB=malloc(2);
cantDatRutc=malloc(2);
idRutaNc=malloc(2);
hhRutc=malloc(2);
mmRutc=malloc(2);

while (TRUE)
{
    usb_task();
    if (usb_enumerated()){
        if (usb_cdc_kbhit()){
            datoUSB=usb_cdc_getc();           //putc(datoUSB);
            if(datoUSB=='*'){
                contUSB=0;
                cantDatRut=0;
            }
            bufferUSB[contUSB]=datoUSB;

            if(datoUSB=='.'){
                sprintf(dirMSB,"%c%c",bufferUSB[3],bufferUSB[4]);
                sprintf(dirLSB,"%c%c",bufferUSB[5],bufferUSB[6]);
                desMSB=strtol(dirMSB,&ptr,16);
                desLSB=strtol(dirLSB,&ptr,16);
                if(bufferUSB[1]=='R'&& bufferUSB[2]=='t'){
                    sprintf(cantDatRutc,"%c%c",bufferUSB[7],bufferUSB[8]);
                    sprintf(aaRc,"%c%c",bufferUSB[9],bufferUSB[10]);
                    sprintf(mmRc,"%c%c",bufferUSB[11],bufferUSB[12]);
                    sprintf(ddRc,"%c%c",bufferUSB[13],bufferUSB[14]);
                    sprintf(idRutaNc,"%c%c",bufferUSB[15],bufferUSB[16]);
                    cantDatRut=strtol(cantDatRutc,&ptr,10);
                    cantDatRut=17+cantDatRut*4;
                    j=0;
                    for(i=17;i<cantDatRut;i++){
                        sprintf(hhRutc,"%c%c",bufferUSB[i],bufferUSB[i+1]);
                        sprintf(mmRutc,"%c%c",bufferUSB[i+2],bufferUSB[i+3]);
                        hhRut[j]=strtol(hhRutc,&ptr,10);
                        mmRut[j]=strtol(mmRutc,&ptr,10);
                        j++;
                        i=i+3;}
                    msg=3;
                }
                if(bufferUSB[1]=='I'&& bufferUSB[2]=='r'){
                    sprintf(segC,"%c%c",bufferUSB[7],bufferUSB[8]);
                    sprintf(minC,"%c%c",bufferUSB[9],bufferUSB[10]);
                    sprintf(horac,"%c%c",bufferUSB[11],bufferUSB[12]);
                    sprintf(diac,"%c%c",bufferUSB[13],bufferUSB[14]);
                    sprintf(mesc,"%c%c",bufferUSB[15],bufferUSB[16]);
                    sprintf(aaac,"%c%c",bufferUSB[19],bufferUSB[20]);
                    msg=5;}
                if(bufferUSB[1]=='P'&& bufferUSB[2]=='r'){
                    msg=1;}
                if(bufferUSB[1]=='B'&& bufferUSB[2]=='m'){
                    msg=2;}
                if(bufferUSB[1]=='D'&& bufferUSB[2]=='a'){
                    msg=6;}
                if(bufferUSB[1]=='C'){
                    numControl=bufferUSB[2];
                    msg=4;}
            }
            contUSB++;
        }
    }
}

```

```

if (msg==1) {
    EnviarDato2 (desMSB,desLSB, 'P', 'r');
    msg=0;}
if (msg==2) {
    EnviarDato2 (desMSB,desLSB, 'B', 'm');
    msg=0;}
if (msg==3) {
    ddR=strtol (ddRc, &ptr, 16);
    mmR=strtol (mmRc, &ptr, 16);
    aaR=strtol (aaRc, &ptr, 16);
    idRutaN=strtoul (idRutaNc, &ptr, 10);
    cantDatRut=strtol (cantDatRutc, &ptr, 10);
    EnviarRuta (desMSB,desLSB, ddR, mmR, aaR, idRutaN, cantDatRut);
    msg=0;}
if (msg==4) {
    EnviarDato2 (desMSB,desLSB, 'C', numControl);
    msg=0;}
if (msg==5) {
    seg=strtol (segc, &ptr, 16);
    min=strtol (minc, &ptr, 16);
    hora=strtol (horac, &ptr, 16);
    dia=strtol (diac, &ptr, 16);
    mes=strtol (mesc, &ptr, 16);
    aaa=strtol (aaac, &ptr, 16);
    IgualarReloj (desMSB,desLSB, seg, min, hora, dia, mes, aaa);
    msg=0;
}
if (msg==6) {
    EnviarDato2 (desMSB,desLSB, 'D', 'a');
    msg=0;}
}
}

```

CÓDIGO FUENTE MÓDULO NODO

```

#include <18f2550.h>
#fuses HS, NOWDT, NOPROTECT, NOLVP, NOMCLR, NOPBADEN, NODEBUG
#use delay (clock=12000000)
#include <string.h>
#include <stdlib.h>

#use rs232 (baud=9600, xmit=pin_C6, rcv=pin_C7) //Configuración de RS232 para módulo XBee
int lenLSB=0x00, lenMSB=0x00, CRC=0x00;
int contrRTCC, msg=0, dato=0, datoR=0x00, cont;
char buffer[20], rutaC, numControl, numControlAnt;

void EnviarXBee (int destinoMSB, int destinoLSB, int dat0, int dat1) {
    lenLSB=0x08;
    CRC= 0xFF- ((0x01+0x4d+destinoMSB+destinoLSB+0x04+dat0+dat1+'.') & 0xFF);
    putc (0x7E); //Bit de inicio
    putc (lenMSB); //MSB Length
    putc (lenLSB); //LSB Length
    putc (0x01); //API identifier
    putc (0x4d); //Frame ID (M)
    putc (destinoMSB); //Destino MSB
    putc (destinoLSB); //Destino LSB
    putc (0x04); //options
    putc (dat0); //datos (R)
    putc (dat1); //datos (a)
    putc ('.');
}

```

```

    putc(CRC); //Checksum
}

#int_rtcc
void rtcc_isr(){
    contrRTCC++;
    if(contrRTCC==50){
        contrRTCC=0;
        cont++;
        EnviarXBee(0xFF,0xFF,'P','n');// n simboliza al Número de Punto de Control
    }
    set_rtcc(139);
}

void main() {
    delay_ms(100);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256|RTCC_8_BIT);
    set_timer0(139);
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_RTCC);
    cont=0;
    while (TRUE){
    }
}

```

CÓDIGO FUENTE MÓDULO USUARIO

```

#include <18f4550.h> //Microcontrolador a utilizar
#fuses HS,NOWDT,NOPROTECT,NOLVP,NOMCLR,NOPBADEN,NODEBUG,USBDIV,VREGEN,PLL3,CPUDIV3
//Oscilador externo de 12MHz
#use delay(clock=4000000)
#include <HDM64GS12.c> //Libreria para GLCD
#include <graphics.c> //Libreria de gráficos para GLCD
#include <24128.c> //Libreria para EEPROM
#include <xbeeTx.c>
#include <operaciones.c>

#use rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7) //Configuración de RS232 para módulo
XBee

#use standard_io(D)
#use standard_io(A)
#use standard_io(C)
#use fast_io (b)

#use i2c(MASTER,SDA=PIN_b7, SCL=PIN_b6,FAST=400000) //Configuración I2C para RTC
DS1307 y EEPROM externa
#priority RTCC,RDA

//*****Declaración de variables*****
int seconds,minutes,hours,day,date,month,year,control;
signed minn;
int secondsV,minutesV,hoursV,ddR,mmR,aaR,cantR;
int hor,min,dia,mes,anio,hhRuta[30],mmRuta[30];
int minPC1,horPC1,minPC2,horPC2,minPC3,horPC3,minLlega,horLlega;
int lugar,descarga=0,horSalida,minSalida,contrRTCC=0,alertLlega=0;//,cont_ee=10;
int16 postTxt=0,ruta,datoR=0x00,msg=0,cont=0;//,postTxtManiana=100;
long int
pos_ee=0,cantDatosH=0,cantDatosL=0,pos=1,dd,mm,aa,i,j,idRuta,cantDat;//,idRutaHoy;
int minRetraso=0,minRetrasoTemp=0,cantDatos=0,alerta=0,cambio=0,regreso=0,retraso=0;
char buffer[100],numControl,numControlAnt,recorrido='N';
char txt[30]="",txtmsg[30]="",txtLugar[30]="";

#define verde pin_a0

```

```

#define rojo pin_a1
#define puls pin_a2

#define IPR1=0xf9f
//*****Registros EEPROM EXTERNA*****
//Posiciones de 0 a 20 reservadas para datos de vehículo y rutas
#define ee_numUnidad 0
#define ee_rutaHoy 1
#define ee_ddRutaHoy 2
#define ee_mmRutaHoy 3
#define ee_aaRutaHoy 4
#define ee_rutaManiana 5
#define ee_ddRutaManiana 6
#define ee_mmRutaManiana 7
#define ee_aaRutaManiana 8
#define ee_estadoUnidad 9
#define ee_minRetraso 10
#define ee_cantDatosL 11
#define ee_cantDatosH 12
#define ee_min 13
#define ee_hora 14
#define ee_ultimoLugar 15
#define ee_regreso 16
#define ee_cantHoy 17
#define ee_cantManiana 18
#define ee_descarga 19
#define ee_datos 1000

//*****Registros DS1307*****
#define ds_seconds 0x00
#define ds_minutes 0x01
#define ds_hours 0x02
#define ds_day 0x03
#define ds_date 0x04
#define ds_month 0x05
#define ds_year 0x06
#define ds_control 0x07

void Obtener_Hora(){
    i2c_start(); //Inicia I2C
    i2c_write(0xd0); //Direcciona DS1307 write 11010000
    i2c_write(ds_seconds); //Indica que ponga registro en ds_seconds
    i2c_stop();

    i2c_start();
    i2c_write(0xd1); //Lee DS1307 11010001
    seconds = i2c_read();
    minutes = i2c_read();
    hours = i2c_read();
    day = i2c_read();
    date = i2c_read();
    month = i2c_read();
    year = i2c_read(0); //no ack
    i2c_stop();
}

void CargarIdRuta(int16 idR){
    if(idR==0){
        sprintf(txt,"SNRUT");
    }
    else{
        postTxt=195+(idR*5);

        sprintf(txt,"%c%c%c%c%c",read_ext_eeprom(postTxt),read_ext_eeprom(postTxt+1),read_ext_eeprom(postTxt+2),read_ext_eeprom(postTxt+3),read_ext_eeprom(postTxt+4));
    }
}

```

```

    GLCD_RECT(95,15,125,23,1,1);
    GLCD_TEXT57(96,15,txt,1,0);
}

void Guardar_Tiempo(int lug,int mm,int hh, int ret){
    ruta=read_ext_eeprom(ee_rutaHoy);
    write_ext_eeprom(ee_ultimoLugar,lugar);
    cantDatosH=0;
    cantDatosL=0;
    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH); //1000
    write_ext_eeprom(pos_ee,lug);pos_ee++;//1001
    write_ext_eeprom(pos_ee,ruta);pos_ee++;//1002
    write_ext_eeprom(pos_ee,year);pos_ee++;//1003
    write_ext_eeprom(pos_ee,month);pos_ee++;//1004
    write_ext_eeprom(pos_ee,date);pos_ee++;//1005
    write_ext_eeprom(pos_ee,hh);pos_ee++;//1006
    write_ext_eeprom(pos_ee,mm);pos_ee++;//1007
    write_ext_eeprom(pos_ee,recorrido);pos_ee++;//1008
    write_ext_eeprom(pos_ee,minSalida);pos_ee++;//1009
    write_ext_eeprom(pos_ee,horSalida);pos_ee++;//1010
    write_ext_eeprom(pos_ee,ret);pos_ee++;//1011

    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,(pos_ee/256));
    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    retraso=0;
    glcd_circle(97,6,1,1,0);
}

void Descargar_Datos(){
    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    for(pos=1000;pos<pos_ee;pos++){
        cantDatos++;
        pos=pos+10;
    }
    EnviarCantDatosXBee(0x10,0x00,cantDatos);
    cantDatos=0;

    for(pos=1000;pos<pos_ee;pos++){
    EnviarDatosXBee(0x10,0x00,read_ext_eeprom(pos+1),read_ext_eeprom(pos),read_ext_eeprom
(pos+2),read_ext_eeprom(pos+3),read_ext_eeprom(pos+4),read_ext_eeprom(pos+5),read_ext
_eeprom(pos+6),read_ext_eeprom(pos+7),read_ext_eeprom(pos+8),read_ext_eeprom(pos+9),r
ead_ext_eeprom(pos+10));
        pos=pos+10;
    }
}

void Borrar_Datos(){
    for(pos=1000;pos<pos_ee;pos++)
        write_ext_eeprom(pos,0);
    pos_ee=1000;
    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,3);
    descarga=0;
    write_ext_eeprom(ee_descarga,descarga);
    glcd_circle(97,6,1,1,1);
}

```



```

void Despliega_Tiempo(int hh,int mm){
    glcd_rect(0,46,92,63,1,0);
    sprintf(txt,"%02d:%02d",hh,mm);
    GLCD_TEXT57(20,49,txt,2,1);
    write_ext_eeprom(ee_min,mm);
    write_ext_eeprom(ee_hora,hh);
    output_low(rojo);
    output_high(verde);
}

void Calcula_Tiempos(){
    if(regreso==0){
        minPC1=minSalida+5;
        minPC2=minSalida+22;
        horPC1=Verifica_hora(minPC1,horSalida);
        horPC2=Verifica_hora(minPC2,horSalida);
        minPC1=Verifica_min(minPC1);
        minPC2=Verifica_min(minPC2);
    }
    if(regreso==1){
        minPC1=minSalida+7;
        minPC2=minSalida+23;
        minPC3=minSalida+36;
        horPC1=Verifica_hora(minPC1,horSalida);
        horPC2=Verifica_hora(minPC2,horSalida);
        horPC3=Verifica_hora(minPC3,horSalida);
        minPC1=Verifica_min(minPC1);
        minPC2=Verifica_min(minPC2);
        minPC3=Verifica_min(minPC3);
    }
}

void AdelantaTiempo(){
    recorrido='A';
    minn=minSalida;
    minn=minn-4;
    if(minn<0){
        horSalida=horSalida-1;
        minn=minn+60;
    }
    minSalida=minn;
}

void Escribe_Lugar(int op){
    switch(op){
        case 0: sprintf(txtmsg,"Sale del Terminal");
                sprintf(txtLugar,"OTAVALO");
                break;
        case 1: sprintf(txtmsg,"Siguiete Registro");
                sprintf(txtLugar,"Agropecuario");
                break;
        case 2: sprintf(txtmsg,"Siguiete Registro");
                sprintf(txtLugar,"Atuntaqui");
                break;
        case 3: sprintf(txtmsg,"Ultimo Registro");
                sprintf(txtLugar,"Ibarra");
                break;
        case 4: sprintf(txtmsg,"Sale del Terminal");
                sprintf(txtLugar,"IBARRA");
                break;
        case 5: sprintf(txtmsg,"Siguiete Registro");
                sprintf(txtLugar,"La Florida");
                break;
        case 6: sprintf(txtmsg,"Siguiete Registro");
                sprintf(txtLugar,"Sto. Domingo");
    }
}

```

```

        break;
    case 7: sprintf(txtmsg,"Siguiente Registro:");
            sprintf(txtLugar,"Peguche");
            break;
    case 8: sprintf(txtmsg,"Ultimo Registro:");
            sprintf(txtLugar,"Otavalo");
            break;
    case 9: sprintf(txtmsg,"*-* CARGAR RUTA *-*");
            sprintf(txtLugar," ");
            break;
    default:break;
}
glcd_rect(0,29,118,38,1,0);
GLCD_TEXT57(1,29,txtmsg,1,1);
glcd_rect(0,39,92,63,1,0);
GLCD_TEXT57(2,39,txtLugar,1,1);
}

void Presenta_Alerta() {
    output_low(verde);
    write_ext_eeprom(ee_descarga,0);
    if(cambio==0) {
        output_high(rojo);
        sprintf(txt,"***ALERTA***");
        GLCD_TEXT57(7,52,txt,1,0);
        glcd_line(22,60,62,60,0);
        glcd_line(22,62,62,62,0);
        sprintf(txt,"%02d:%02d",read_ext_eeprom(ee_hora),read_ext_eeprom(ee_min));
        GLCD_TEXT57(20,49,txt,2,1);
        cambio=1;
    }
    else{
        output_low(rojo);
        sprintf(txt,"%02d:%02d",read_ext_eeprom(ee_hora),read_ext_eeprom(ee_min));
        GLCD_TEXT57(20,49,txt,2,0);
        glcd_line(22,60,62,60,1);
        glcd_line(22,62,62,62,1);
        sprintf(txt,"***ALERTA***");
        GLCD_TEXT57(7,52,txt,1,1);
        cambio=0;
    }
}

void Cargar_Ruta_glcd()
{
    int16 posMn;
    int cantR=0,cargar=0;
    cantR=read_ext_eeprom(ee_cantHoy);
    cantR = (cantR*2)+50;
    CargarIdRuta(read_ext_eeprom(ee_rutaHoy));
    if(read_ext_eeprom(ee_ddRutaHoy)==date && read_ext_eeprom(ee_mmRutaHoy)==month &&
read_ext_eeprom(ee_aaRutaHoy)==year) {
        cargar=1;
    }
    else{
        if(read_ext_eeprom(ee_ddRutaManiana)==date &&
read_ext_eeprom(ee_mmRutaManiana)==month && read_ext_eeprom(ee_aaRutaManiana)==year) {
            write_ext_eeprom(ee_rutaHoy,read_ext_eeprom(ee_RutaManiana));
            write_ext_eeprom(ee_ddRutaHoy,read_ext_eeprom(ee_ddRutaManiana));
            write_ext_eeprom(ee_mmRutaHoy,read_ext_eeprom(ee_mmRutaManiana));
            write_ext_eeprom(ee_aaRutaHoy,read_ext_eeprom(ee_aaRutaManiana));
            write_ext_eeprom(ee_cantHoy,read_ext_eeprom(ee_cantManiana));
            cantR=50 + 2*read_ext_eeprom(ee_cantManiana);
            posMn=100;
        }
    }
}

```

```

for(i=50;i<cantR;i++){
    write_ext_eeprom(i,read_ext_eeprom(posMn));posMn++;i++;
    write_ext_eeprom((posMn-1),0);
    write_ext_eeprom(i,read_ext_eeprom(posMn));posMn++;
    write_ext_eeprom((posMn-1),0);
}
write_ext_eeprom(ee_rutaManiana,0);
write_ext_eeprom(ee_ddRutaManiana,0);
write_ext_eeprom(ee_mmRutaManiana,0);
write_ext_eeprom(ee_aaRutaManiana,0);
write_ext_eeprom(ee_cantManiana,0);
GLCD_CIRCLE(122,32,3,1,0);
ruta=read_ext_eeprom(ee_rutaHoy);
cargar=1;
}else{
    horSalida=0;
    minSalida=0;
    CargarIdRuta(0);
    Escribe_Lugar(9);
}
}
if(cargar==1){
    if(((read_ext_eeprom(cantR-2)<(hours-HexToDec(hours))))||((read_ext_eeprom(cantR-2)==(hours-HexToDec(hours))))&&(read_ext_eeprom(cantR-1)<(minutes-HexToDec(minutes))))){
        horSalida=0;
        minSalida=0;
        CargarIdRuta(0);
        Escribe_Lugar(9);i=cantR;}
    for(i=50;i<cantR;i++){
        if(read_ext_eeprom(i)>(hours-HexToDec(hours))){
            horSalida=read_ext_eeprom(i);
            minSalida=read_ext_eeprom(i+1);
            if(i==50||i==54||i==58||i==62||i==66||i==70||i==74||i==78){
                write_ext_eeprom(ee_regreso,0);Escribe_Lugar(0);}
            if(i==52||i==56||i==60||i==64||i==68||i==72||i==76||i==80){
                write_ext_eeprom(ee_regreso,1);Escribe_Lugar(4);}
            i=cantR;
        }
        else{
            if(read_ext_eeprom(i)==(hours-HexToDec(hours)) &&
read_ext_eeprom(i+1)>=(minutes-HexToDec(minutes))){
                horSalida=read_ext_eeprom(i);
                minSalida=read_ext_eeprom(i+1);
                if(i==50||i==54||i==58||i==62||i==66||i==70||i==74||i==78){
                    write_ext_eeprom(ee_regreso,0);Escribe_Lugar(0);}
                if(i==52||i==56||i==60||i==64||i==68||i==72||i==76||i==80){
                    write_ext_eeprom(ee_regreso,1);Escribe_Lugar(4);}
                i=cantR;
            }
        }
        i++;
    }
    write_ext_eeprom(ee_min,minSalida);
    write_ext_eeprom(ee_hora,horSalida);
    sprintf(txt,"%02d:%02d",horSalida,minSalida);
    GLCD_RECT(20,49,82,63,1,0);
    GLCD_TEXT57(20,49,txt,2,1);
}
}

void Cargar_Ruta_ee(int16 rut,int cant,int dd,int mm,int aa){
    int16 posH=50,posM=100;
    int rutLow;

```

```

if(dd==date && mm==month && aa==year){
    rutLow=rut;
    write_ext_eeprom(ee_rutaHoy,rutLow);
    write_ext_eeprom(ee_ddRutaHoy,dd);
    write_ext_eeprom(ee_mmRutaHoy,mm);
    write_ext_eeprom(ee_aaRutaHoy,aa);
    write_ext_eeprom(ee_cantHoy,cant);
    ruta=read_ext_eeprom(ee_rutaHoy);

    for(i=0;i<cant;i++){
        write_ext_eeprom(posH,hhRuta[i]);posH++;
        write_ext_eeprom(posH,mmRuta[i]);posH++;
    }
    Calcula_Tiempos();
    EnviarOkRuta();
    Cargar_Ruta_glcd();
}
else{
    write_ext_eeprom(ee_rutaManiana,rut);
    write_ext_eeprom(ee_ddRutaManiana,dd);
    write_ext_eeprom(ee_mmRutaManiana,mm);
    write_ext_eeprom(ee_aaRutaManiana,aa);
    write_ext_eeprom(ee_cantManiana,cant);
    for(i=0;i<cant;i++){
        write_ext_eeprom(posM,hhRuta[i]);posM++;
        write_ext_eeprom(posM,mmRuta[i]);posM++;
    }
    EnviarOkRuta();
    GLCD_CIRCLE(122,32,3,1,1);
    glcd_circle(122,32,1,1,0);
}
}

void Calcula_Retraso(){
    if(retraso==1){
        minRetrasoTemp++;
        minRetraso=read_ext_eeprom(ee_minRetraso)+1;
        write_ext_eeprom(ee_minRetraso,minRetraso);
    }
}

void Cancela_Retraso(){
    write_ext_eeprom(ee_minRetraso,0);
}

void Presenta_Retraso(){
    GLCD_RECT(93,39,127,62,0,1);
    GLCD_RECT(95,41,125,60,1,1);
    sprintf(txt,"%02d",read_ext_eeprom(ee_minRetraso));
    GLCD_TEXT57(100,44,txt,2,0);
}

void Iguala_Reloj(int ss,int mm,int hh,int dia,int dd,int mes,int aa,int control){
    //Establece fecha y hora
    i2c_start(); //Inicia I2C
    i2c_write(0xd0); //Direcciona DS1307 write
    i2c_write(ds_seconds); //Indica que empiece en registro ds_seconds
    //Aquí establece fecha y hora respecto a la puesta arriba
    i2c_write(ss);
    i2c_write(mm);
    i2c_write(hh);
    i2c_write(dia); //1 es domingo
    i2c_write(dd);
    i2c_write(mes);
}

```

```

    i2c_write(aa);
    i2c_write(control);
    i2c_stop(); //Ya puso la hora y fecha
    //El RTC ya está marcando el tiempo y fecha
}
void Reserva_Memoria(int reg){
    if(reg==0){
        Guardar_Tiempo(1,0,0,0);
        Guardar_Tiempo(2,0,0,0);
    }
    if(reg==1){
        Guardar_Tiempo(5,0,0,0);
        Guardar_Tiempo(6,0,0,0);
        Guardar_Tiempo(7,0,0,0);
    }
}
void IniciaRecorrido(int regr){
    Reserva_Memoria(regr);
    output_low(rojo);
    output_high(verde);
}
void FinalRecorrido(){
    output_low(rojo);
    output_low(verde);
}
#int_rtcc
void rtcc_isr(){
    contrTCC++;
    if(contrTCC==10){//100ms
        contrTCC=0;
        Obtener_Hora();
        if(seconds!=secondsV){ //printf("\r\n%02x:%02x:%02x",hours,minutes,seconds);
            glcd_rect(68,12,89,26,1,0);
            sprintf(txt,"%02x", (int)seconds);
            glcd_text57(68,12,txt,2,1);
            secondsV=seconds;
            if(alerta==1)
                Presenta_Alerta();
        }
        if(minutes!=minutesV){
            glcd_rect(35,12,56,26,1,0);
            sprintf(txt,"%02x", (int)minutes);
            glcd_text57(35,12,txt,2,1);
            minutesV=minutes;
            if((read_ext_eeprom(ee_min))== (minutes-hexToDec(minutes))){
                if(read_ext_eeprom(ee_hora)== (hours-hexToDec(hours))){
                    alerta=1;
                }
            }
            else{
                alerta=0;
            }
            if(retraso==1){
                Calcula_Retraso();
                Presenta_Retraso();
            }
            if((read_ext_eeprom(ee_min)+1)== (minutes-hexToDec(minutes)) && (read_ext_eeprom(ee_min) !=minSalida) && (read_ext_eeprom(ee_min) !=minLlega)){
                if(read_ext_eeprom(ee_hora)== (hours-hexToDec(hours))){
                    retraso=1;
                    Calcula_Retraso();
                }
            }
        }
    }
}

```

```

        Presenta_Retraso();
    }
}
if((minSalida+1)==(minutes-hexToDec(minutes))){
    if(horSalida==(hours-hexToDec(hours))){
        regreso=read_ext_eeprom(ee_regreso);
        if(regreso==0)
            lugar=1;
        if(regreso==1)
            lugar=5;
        Escribe_Lugar(lugar);
        Calcula_Tiempos();
        Despliega_Tiempo(horPC1,minPC1);
        alerta=0;
        IniciaRecorrido(regreso);
        //recorrido='N';
    }
}
}

if(hours!=hoursV){
    glcd_rect(2,12,23,26,1,0);
    sprintf(txt,"%02x",hours);
    glcd_text57(2,12,txt,2,1);
    glcd_rect(2,2,92,9,1,0);
    sprintf(txt,"Fech:%02x-%02x-20%02x",date,month,year);
    GLCD_TEXT57(2,2,txt,1,1);
    hoursV=hours;
}
}
set_rtcc(217);
}

//*****Interrupción por recepción de dato RS232*****
#int_RDA
void RDA_isr(){
    if(kbhit()){
        datoR=getc();
        if(datoR==0x7E){
            cont=0;
        }
        buffer[cont]=datoR;

        if(datoR==' '){
            if(buffer[4]==0x10 && buffer[5]==0x00){
                if(buffer[8]=='C'){
                    numControl=buffer[9];
                    if(numControlAnt==numControl){
                        msg=0;break;}
                    else{
                        hor=hours;
                        min=minutes;
                        disable_interrupts(INT_RTCC);retraso=0;
                        msg=4;break;}
                }
            if(buffer[8]=='P' && buffer[9]=='r'){
                msg=1;break;}

            if(buffer[8]=='D' && buffer[9]=='a'){
                msg=6;break;}

            if(buffer[8]=='B' && buffer[9]=='m'){
                msg=2;break;}

```

```

        if(buffer[8]=='I' && buffer[9]=='r'){
            msg=5;
            seconds=(int)buffer[10];
            minutes=(int)buffer[11];
            hours=(int)buffer[12];
            date=(int)buffer[13];
            month=(int)buffer[14];
            year=(int)buffer[15];break;}

        if(buffer[8]=='R'&& buffer[9]=='t'){
            ddR=(int)buffer[10];
            mmR=(int)buffer[11];
            aaR=(int)buffer[12];
            idRuta=(int16)buffer[13];
            cantDat=(int)buffer[14];
            j=0;
            for(i=15;i<(cantDat*2+15);i++){
                hhRuta[j]=(int)buffer[i];
                mmRuta[j]=(int)buffer[i+1];
                j++;
                i++;}
            msg=3;
            break;}
    }
}
cont++;
}
}

void main(){
    delay_ms(100);

    IPR1=0b00000000;
    //*****Inicializa interrupciones externas*****
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_RDA);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256|RTCC_8_BIT);
    set_timer0(217);
    enable_interrupts(INT_RTCC);

    init_ext_eeprom();
    Obtener_Hora();
    secondsV=seconds;
    minutesV=minutes;
    hoursV=hours;

    //*****Inicializa GLCD*****

    glcd_init(on);
    GLCD_RECT(5,5,90,12,1,0);
    sprintf(txt,"Fech:%02x-%02x-20%02x",date,month,year);
    GLCD_TEXT57(2,2,txt,1,1);//37-2
    sprintf(txt,"%02x:%02x:%02x",hours,minutes,seconds);
    GLCD_TEXT57(2,12,txt,2,1);//20-2

    GLCD_RECT(93,2,127,25,0,1);//recuadro Unidad-Ruta
    GLCD_RECT(95,4,125,23,1,1);
    sprintf(txt,"U:%02d",read_ext_eeprom(ee_numUnidad));
    GLCD_TEXT57(100,6,txt,1,0);
    Cargar_Ruta_glcd();
    regreso=read_ext_eeprom(ee_regreso);
    Calcula_Tiempos();

    Presenta_Retraso();

```

```

cantDatosH=read_ext_eeprom(ee_cantDatosH);
cantDatosL=read_ext_eeprom(ee_cantDatosL);
pos_ee=cantDatosL+256*cantDatosH;
if(pos_ee==1000)
    glcd_circle(97,6,1,1,1);
else
    glcd_circle(97,6,1,1,0);
lugar=read_ext_eeprom(ee_ultimoLugar);
ruta=read_ext_eeprom(ee_rutaHoy);
dd=read_ext_eeprom(ee_ddRutaHoy);
mm=read_ext_eeprom(ee_mmRutaHoy);
aa=read_ext_eeprom(ee_aaRutaHoy);
cantR=read_ext_eeprom(ee_cantHoy);

    if(read_ext_eeprom(ee_rutaManiana)!=0){
        GLCD_CIRCLE(122,32,3,1,1);
        glcd_circle(122,32,1,1,0);
    }
}

do{
    if(input(puls)==0){
        AdelantaTiempo();
        Calcula_Tiempos();
        Despliega_Tiempo(horSalida,minSalida);
    }
    if(msg==1){
        Cancela_Retraso(); Presenta_Retraso();
        EnviarOkCancelaR();
        msg=0;}

    if(msg==2){
        Borrar_Datos(); write_ext_eeprom(ee_descarga,0);
        EnviarOkBorrarD();
        msg=0;}

    if(msg==3){
        Cargar_Ruta_ee(idRuta,cantDat,ddR,mmR,aaR);
        msg=0;}

    if(msg==4){
        numControlAnt=numControl;
        switch(numControl){
            case '1':if(lugar==1){
                cantDatosH=read_ext_eeprom(ee_cantDatosH);
                cantDatosL=read_ext_eeprom(ee_cantDatosL);
                pos_ee=cantDatosL+(256*cantDatosH);
                pos_ee=pos_ee-22;
                write_ext_eeprom(ee_cantDatosL,pos_ee);
                write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

Guardar_Tiempo(1,min,hor,minRetrasoTemp);minRetrasoTemp=0;
                cantDatosH=read_ext_eeprom(ee_cantDatosH);
                cantDatosL=read_ext_eeprom(ee_cantDatosL);
                pos_ee=cantDatosL+(256*cantDatosH);
                pos_ee=pos_ee+11;
                write_ext_eeprom(ee_cantDatosL,pos_ee);
                write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

                lugar=2;
                Escribe_Lugar(lugar);
                Despliega_Tiempo(horPC2,minPC2);
                alerta=0;
                retraso=0;

```



```

    } break;
case '2':if(lugar==2){
    retraso=0;
    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    pos_ee=pos_ee-11;
    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

```

```

Guardar_Tiempo(2,min,hor,minRetrasoTemp);minRetrasoTemp=0;

```

```

    lugar=3;
    FinalRecorrido();
    recorrido='N';
    alerta=0;
    Cargar_Ruta_glcd();
    Cargar_Ruta_glcd();
    retraso=0;

```

```

    } break;

```

```

case '5':if(lugar==5){
    retraso=0;
    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    pos_ee=pos_ee-33;
    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

```

```

Guardar_Tiempo(5,min,hor,minRetrasoTemp);minRetrasoTemp=0;

```

```

    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    pos_ee=pos_ee+22;
    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

```

```

    lugar=6;
    Escribe_Lugar(lugar);
    Despliega_Tiempo(horPC2,minPC2);//pc5
    retraso=0;
    alerta=0;

```

```

    } break;

```

```

case '6':if(lugar==6){
    retraso=0;
    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    pos_ee=pos_ee-22;
    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

```

```

Guardar_Tiempo(6,min,hor,minRetrasoTemp);minRetrasoTemp=0;

```

```

    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    pos_ee=pos_ee+11;
    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

```

```

    lugar=7;

```

```

        Escribe_Lugar(lugar);
        Despliega_Tiempo(horPC3,minPC3);//pc6
        alerta=0;
        retraso=0;
    } break;
case '7':if(lugar==7){
    retraso=0;
    cantDatosH=read_ext_eeprom(ee_cantDatosH);
    cantDatosL=read_ext_eeprom(ee_cantDatosL);
    pos_ee=cantDatosL+(256*cantDatosH);
    pos_ee=pos_ee-11;
    write_ext_eeprom(ee_cantDatosL,pos_ee);
    write_ext_eeprom(ee_cantDatosH,(pos_ee/256));

Guardar_Tiempo(7,min,hor,minRetrasoTemp);minRetrasoTemp=0;

        write_ext_eeprom(ee_descarga,1);
        lugar=8;
        FinalRecorrido();
        recorrido='N';
        alerta=0;
        Cargar_Ruta_glcd();
        Cargar_Ruta_glcd();
        Calcula_Tiempos();
        retraso=0;
    } break;
    default: break;
}
clear_interrupt(INT_RTCC);
enable_interrupts(INT_RTCC);
msg=0;
}
if(msg==5){
    Igualar_Reloj(seconds,minutes,hours,0x01,date,month,year,0x00);
    EnviarOkRTC();
    msg=0;}

if(msg==6){
    if(read_ext_eeprom(ee_descarga)==1)
        Descargar_Datos();
    msg=0;}
}while(true);
}

```