

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica
UNITAT DE MONITORITZACIÓ DEL SISTEMA
D'ALIMENTACIÓ FOTOVOLTAIC D'UN HABITATGE
UNIFAMILIAR AÏLLAT DE LA XARXA



Annexes

Autor: Pol Castañé Guzman
Director: Manuel Manzanares Brotons
Convocatòria: Gener 2017

Índex annexes

1. PROGRAMA PIC 18F4550	3
2. PROGRAMA PIC 16F883 (1)	45
3. PROGRAMA PIC 16F883 (2)	47
4. PROGRAMA PIC 16F883 (3)	49
5. PROGRAMA PIC 16F883 (4)	51



1. Programa PIC 18F4550

```
#include <18F4550.h>

#device ADC=8

#use delay(clock=48000000)

#use RS232(BAUD=9600,BITS=8,PARITY=N,XMIT=PIN_C6,RCV=PIN_C7)

#use I2C(MASTER, SDA=PIN_B0 , SCL=PIN_B1)

#fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,PLL5,CPUDIV1

#define SLAVE1_WRT_ADDR 0x12

#define SLAVE1_READ_ADDR 0x13

#define SLAVE2_READ_ADDR 0xA0

#define SLAVE3_READ_ADDR 0xA1

#define SLAVE4_READ_ADDR 0xA2

#define LCD_DB4 PIN_D7 //pin 14 LCD

#define LCD_DB5 PIN_D6 //pin 13 LCD

#define LCD_DB6 PIN_D5 //pin 12 LCD

#define LCD_DB7 PIN_C2 //pin 11 LCD

#define LCD_E PIN_C1 //pin 06 LCD

//#define LCD_RW PIN_B0 //pin 14 LCD (a tierra desde LCD)

#define LCD_RS PIN_C0 //pin 04 LCD

#define UP PIN_D0

#define DOWN PIN_D1

#define IN PIN_D2
```



```
#define BACK PIN_D3

#define HOME PIN_D4

#define PROBA PIN_D5

#define LED_I PIN_A4

#define LED_E PIN_E0

//Fixe

//!#include "C:\Users\Pol\Dropbox\TFG\PROGRAMAS\18F4550\18F4550 24-10-16\flex_lcd420.c"

//!#include "C:\Users\Pol\Dropbox\TFG\PROGRAMAS\18F4550\18F4550 24-10-16\DS1307.c"

//!#include "C:\Users\Pol\Dropbox\TFG\PROGRAMAS\18F4550\18F4550 24-10-16\241025.c"

//Portatil

#include "C:\Users\polca\Dropbox\TFG\PROGRAMAS\18F4550\18F4550 24-10-16\flex_lcd420.c"

#include "C:\Users\polca\Dropbox\TFG\PROGRAMAS\18F4550\18F4550 24-10-16\DS1307.c"

#include "C:\Users\polca\Dropbox\TFG\PROGRAMAS\18F4550\18F4550 24-10-16\241025.c"

     //// write_ext_eeprom(a, d); Write the byte d to the address a    ////  
     //// d = read_ext_eeprom(a); Read the byte d from the address a    ////  
  
signed int8 MENU_1=0;  
  
signed int8 MENU_2=0;  
  
signed int8 MENU_3=0;  
  
signed int8 MENU_4=0;  
  
unsigned int1 ENTER_1=0;  
  
unsigned int1 ENTER_2=0;  
  
unsigned int1 ENTER_3=0;
```

```
unsigned int1 ENTER_4=0;  
  
signed int8 CNT_IN=0;  
  
//!unsigned int8 TA=25;  
  
unsigned int32 i=0;  
  
unsigned int32 j=0;  
  
unsigned int32 adr=0;  
  
unsigned int32 DR=1;  
  
unsigned int32 aux=0;  
  
unsigned int32 aux2=0;  
  
unsigned int8 temp;  
  
unsigned int8 V1;  
  
unsigned int8 I1;  
  
unsigned int8 lum;  
  
unsigned int8 PSW=10;  
  
unsigned int8 V2;  
  
unsigned int8 I2;  
  
unsigned int8 BW=140;  
  
unsigned int8 temp_p;  
  
unsigned int8 C_VOL;  
  
unsigned int8 cnt_led=0;  
  
unsigned int1 ini=0;
```

```
int sec=55,min=59,hora=1,dia=13,mes=5,anio=16,dow=3; //introduir dia de la setmana

//!int sec,min,hora,dia,mes,anio,dow;

char fecha[4]; //numero de lletres del dia de la setmana que es veu en el display

unsigned int1 M0=0;

unsigned int1 M1=0;

unsigned int1 M2=0;

unsigned int1 M3=0;

unsigned int1 M4=0;

unsigned int1 M5=0;

void main(){

lcd_init();

init_ext_eeprom();

ds1307_set_date_time(dia,mes,anio,dow,hora,min,sec);

enable_interrupts(global);

enable_interrupts(int_rda);

setup_adc_ports(NO_ANALOGS|VSS_VDD);

setup_adc(ADC_CLOCK_DIV_16);

output_low(LED_I);

//! output_low(P_V1);

//! output_low(P_I1);

//! output_low(P_V2);

//! output_low(P_I2);
```

```
while(true){  
  
    cnt_led++; // unsigned int16: 0 to 65535  
  
    ds1307_get_date(dia,mes,anio,dow);    /// se obtiene la fecha  
  
    ds1307_get_time(hora,min,sec);        /// se obtiene la hora  
  
    if (cnt_led==250) {  
  
        output_toggle(LED_I);  
  
        cnt_led=0;  
  
    }  
  
    if (ini==0) { // CONFIG INICIAL  
  
        //EXECUTAR 1 SOL COP  
  
        ini=1;  
  
    }  
  
    if (1==1) { //BOTONS  
  
        if (input(UP)==0) {  
  
            if ((ENTER_1==0)&&(ENTER_2==0)&&(ENTER_3==0)&&(ENTER_4==0)) MENU_1=MENU_1+1;  
  
            if ((ENTER_1==1)&&(ENTER_2==0)&&(ENTER_3==0)&&(ENTER_4==0)) MENU_2=MENU_2+1;  
  
            if ((ENTER_1==1)&&(ENTER_2==1)&&(ENTER_3==0)&&(ENTER_4==0)) MENU_3=MENU_3+1;  
  
            if ((ENTER_1==1)&&(ENTER_2==1)&&(ENTER_3==1)&&(ENTER_4==0)) MENU_4=MENU_4+1;  
  
            //if ((ENTER_1==1)&&(ENTER_1==1)&&(ENTER_1==1)&&(ENTER_1==1))  
            MENU_5=MENU_5+1;  
  
            while (input(UP)==0) {};  
  
        }  
    }  
}
```

```
if (input(DOWN)==0) {

    if ((ENTER_1==0)&&(ENTER_2==0)&&(ENTER_3==0)&&(ENTER_4==0)) MENU_1=MENU_1-1;

    if ((ENTER_1==1)&&(ENTER_2==0)&&(ENTER_3==0)&&(ENTER_4==0)) MENU_2=MENU_2-1;

    if ((ENTER_1==1)&&(ENTER_2==1)&&(ENTER_3==0)&&(ENTER_4==0)) MENU_3=MENU_3-1;

    if ((ENTER_1==1)&&(ENTER_2==1)&&(ENTER_3==1)&&(ENTER_4==0)) MENU_4=MENU_4-1;

    while (input(DOWN)==0) {};

}

if (input(IN)==0) {

    if (CNT_IN==0) ENTER_1=1;

    if (CNT_IN==1) ENTER_2=1;

    if (CNT_IN==2) ENTER_3=1;

    if (CNT_IN==3) ENTER_4=1;

    CNT_IN=CNT_IN+1;

    if (CNT_IN==5) CNT_IN=0;

    while (input(IN)==0) {};

}

if (input(BACK)==0) {

    CNT_IN=CNT_IN-1;

    if (CNT_IN==0) ENTER_1=0;

    if (CNT_IN==1) ENTER_2=0;

    if (CNT_IN==2) ENTER_3=0;

    if (CNT_IN==3) ENTER_4=0;
```

```
if (CNT_IN==5) CNT_IN=0;

lcd_putc("\f"); //limpia pantalla

while (input(BACK)==0) {};

}

if (input(HOME)==0) {

MENU_1=0;

MENU_2=0;

MENU_3=0;

MENU_4=0;

ENTER_1=0;

ENTER_2=0;

ENTER_3=0;

ENTER_4=0;

CNT_IN=0;

lcd_putc("\f"); //limpia pantalla

while (input(HOME)==0) {};

}

if (1==1) { //LECTURA VARIABLES

set_adc_channel(9);

temp=read_adc();

//temp=temp/3; //temp=((temp/6)*100); //(((temp/6)*100)/204) //2.45
```

```
set_adc_channel(11);

lum=read_adc();

set_adc_channel(4);

temp_p=read_adc();

set_adc_channel(8);

C_VOL=read_adc();

//!    set_adc_channel(1);

//!    I1=read_adc();

//!

//!    set_adc_channel(2);

//!    V2=read_adc();

//!

//!    set_adc_channel(3);

//!    I2=read_adc();

//!

////////////////////

i2c_start();

i2c_write(SLAVE1_READ_ADDR);

V1 = i2c_read(0);

//!    i2c_stop();

//!    delay_ms(300);

////////////////////
```

```
//!  
  
    i2c_start();      // Restart  
  
    i2c_write(SLAVE2_READ_ADDR);  // to change data direction  
  
    l1=i2c_read(0); // Now read from slave  
  
    i2c_stop();  
  
//!    delay_ms(300);  
  
    ///////////////////////  
  
//!    i2c_start();      // Restart  
  
//!    i2c_write(SLAVE3_READ_ADDR);  // to change data direction  
  
//!    V2=i2c_read(0); // Now read from slave  
  
    i2c_stop();  
  
//!    delay_ms(300);  
  
//!  
  
//!    ///////////////////////  
  
//!  
  
//!    i2c_start();      // Restart  
  
//!    i2c_write(SLAVE4_READ_ADDR);  // to change data direction  
  
//!    l2=i2c_read(0); // Now read from slave  
  
    i2c_stop();  
  
//!
```



```
}

if (1==1) { //MONITORITZACIÓ

    //write_ext_eeprom(a, d); Write the byte d to the address a

    //d = read_ext_eeprom(a); Read the byte d from the address a

    switch (min) {

        case 0:

            if (M0==0){

                write_ext_eeprom(i, dia); //Write the byte d to the address a

                write_ext_eeprom(i+1, mes); //Write the byte d to the address a

                write_ext_eeprom(i+2, anio); //Write the byte d to the address a

                write_ext_eeprom(i+3, hora); //Write the byte d to the address a

                write_ext_eeprom(i+4, min); //Write the byte d to the address a

                write_ext_eeprom(i+5, temp); //Write the byte d to the address a

                write_ext_eeprom(i+6, temp_p); //Write the byte d to the address a

                write_ext_eeprom(i+7, lum); //Write the byte d to the address a

                write_ext_eeprom(i+8, V1); //Write the byte d to the address a

                write_ext_eeprom(i+9, I1); //Write the byte d to the address a

                write_ext_eeprom(i+10, V2); //Write the byte d to the address a

                write_ext_eeprom(i+11, I2); //Write the byte d to the address a

                write_ext_eeprom(i+12, C_VOL); //Write the byte d to the address a

            M0=1;

            M5=0;
```

```
i=i+13;  
  
aux++;  
  
}  
  
break;  
  
case 10:  
  
if (M1==0){  
  
    write_ext_eeprom(i, dia); //Write the byte d to the address a  
  
    write_ext_eeprom(i+1, mes); //Write the byte d to the address a  
  
    write_ext_eeprom(i+2, anyo); //Write the byte d to the address a  
  
    write_ext_eeprom(i+3, hora); //Write the byte d to the address a  
  
    write_ext_eeprom(i+4, min); //Write the byte d to the address a  
  
    write_ext_eeprom(i+5, temp); //Write the byte d to the address a  
  
    write_ext_eeprom(i+6, temp_p); //Write the byte d to the address a  
  
    write_ext_eeprom(i+7, lum); //Write the byte d to the address a  
  
    write_ext_eeprom(i+8, V1); //Write the byte d to the address a  
  
    write_ext_eeprom(i+9, I1); //Write the byte d to the address a  
  
    write_ext_eeprom(i+10, V2); //Write the byte d to the address a  
  
    write_ext_eeprom(i+11, I2); //Write the byte d to the address a  
  
    write_ext_eeprom(i+12, C_VOL); //Write the byte d to the address a  
  
    M1=1;  
  
    M0=0;  
  
    i=i+13;
```



```
aux++;

}

break;

case 20:

if (M2==0){

    write_ext_eeprom(i, dia); //Write the byte d to the address a

    write_ext_eeprom(i+1, mes); //Write the byte d to the address a

    write_ext_eeprom(i+2, anio); //Write the byte d to the address a

    write_ext_eeprom(i+3, hora); //Write the byte d to the address a

    write_ext_eeprom(i+4, min); //Write the byte d to the address a

    write_ext_eeprom(i+5, temp); //Write the byte d to the address a

    write_ext_eeprom(i+6, temp_p); //Write the byte d to the address a

    write_ext_eeprom(i+7, lum); //Write the byte d to the address a

    write_ext_eeprom(i+8, V1); //Write the byte d to the address a

    write_ext_eeprom(i+9, l1); //Write the byte d to the address a

    write_ext_eeprom(i+10, V2); //Write the byte d to the address a

    write_ext_eeprom(i+11, l2); //Write the byte d to the address a

    write_ext_eeprom(i+12, C_VOL); //Write the byte d to the address a

M2=1;

M1=0;

i=i+13;

aux++;
```

```
}

break;

case 30:

if (M3==0){

    write_ext_eeprom(i, dia); //Write the byte d to the address a

    write_ext_eeprom(i+1, mes); //Write the byte d to the address a

    write_ext_eeprom(i+2, anyo); //Write the byte d to the address a

    write_ext_eeprom(i+3, hora); //Write the byte d to the address a

    write_ext_eeprom(i+4, min); //Write the byte d to the address a

    write_ext_eeprom(i+5, temp); //Write the byte d to the address a

    write_ext_eeprom(i+6, temp_p); //Write the byte d to the address a

    write_ext_eeprom(i+7, lum); //Write the byte d to the address a

    write_ext_eeprom(i+8, V1); //Write the byte d to the address a

    write_ext_eeprom(i+9, I1); //Write the byte d to the address a

    write_ext_eeprom(i+10, V2); //Write the byte d to the address a

    write_ext_eeprom(i+11, I2); //Write the byte d to the address a

    write_ext_eeprom(i+12, C_VOL); //Write the byte d to the address a

    M3=1;

    M0=0;

    i=i+13;

    aux++;

}
```



```
break;

case 40:

if (M4==0){

    write_ext_eeprom(i, dia); //Write the byte d to the address a

    write_ext_eeprom(i+1, mes); //Write the byte d to the address a

    write_ext_eeprom(i+2, anio); //Write the byte d to the address a

    write_ext_eeprom(i+3, hora); //Write the byte d to the address a

    write_ext_eeprom(i+4, min); //Write the byte d to the address a

    write_ext_eeprom(i+5, temp); //Write the byte d to the address a

    write_ext_eeprom(i+6, temp_p); //Write the byte d to the address a

    write_ext_eeprom(i+7, lum); //Write the byte d to the address a

    write_ext_eeprom(i+8, V1); //Write the byte d to the address a

    write_ext_eeprom(i+9, I1); //Write the byte d to the address a

    write_ext_eeprom(i+10, V2); //Write the byte d to the address a

    write_ext_eeprom(i+11, I2); //Write the byte d to the address a

    write_ext_eeprom(i+12, C_VOL); //Write the byte d to the address a

    M4=1;

    M0=0;

    i=i+13;

    aux++;

}

break;
```

case 50:

```
if (M5==0){

    write_ext_eeprom(i, dia); //Write the byte d to the address a

    write_ext_eeprom(i+1, mes); //Write the byte d to the address a

    write_ext_eeprom(i+2, anyo); //Write the byte d to the address a

    write_ext_eeprom(i+3, hora); //Write the byte d to the address a

    write_ext_eeprom(i+4, min); //Write the byte d to the address a

    write_ext_eeprom(i+5, temp); //Write the byte d to the address a

    write_ext_eeprom(i+6, temp_p); //Write the byte d to the address a

    write_ext_eeprom(i+7, lum); //Write the byte d to the address a

    write_ext_eeprom(i+8, V1); //Write the byte d to the address a

    write_ext_eeprom(i+9, I1); //Write the byte d to the address a

    write_ext_eeprom(i+10, V2); //Write the byte d to the address a

    write_ext_eeprom(i+11, I2); //Write the byte d to the address a

    write_ext_eeprom(i+12, C_VOL); //Write the byte d to the address a

    M5=1;

    M4=0;

    i=i+13;

    aux++;

}

break;

}
```



```
}

if (1==1) { //MENU

if (CNT_IN<0) {

CNT_IN=0;

}

switch (MENU_1) {

case -1:

MENU_1=7;

break;

case 0: // HORA

if (ENTER_1==1) {

ENTER_1=0;

CNT_IN=0;

}

ds1307_get_date(dia,mes,anio,dow);    /// se obtiene la fecha

ds1307_get_time(hora,min,sec);        /// se obtiene la hora

ds1307_get_day_of_week(fecha);

lcd_gotoxy(1,1);

printf(lcd_putc, " %02u/%02u/20%02u",dia,mes,anio);

lcd_gotoxy(1,2);

printf(lcd_putc, " %02u:%02u:%02u ",hora, min, sec);

break;

}
```

case 1: // TEMPERATURAS

```
if (ENTER_3==1) {  
    ENTER_3=0;  
  
    CNT_IN=CNT_IN-1;  
  
}  
  
if (ENTER_1==0) {  
  
    lcd_gotoxy(1,1);  
  
    lcd_putc("Visualizar    ");  
  
    lcd_gotoxy(1,2);  
  
    lcd_putc("Temp. y Lum.    ");  
  
}  
  
if ((ENTER_1==1)&&(ENTER_2==0)) {  
  
    lcd_gotoxy(1,1);  
  
    lcd_putc("Temp. y Lum.    ");  
  
    switch (MENU_2) {  
  
        case -1:  
  
            MENU_2=2;  
  
            break;  
  
        case 0:  
  
            lcd_gotoxy(1,2);  
  
            lcd_putc("T. Ambiental    ");  
    }  
}
```



```
break;

case 1:

lcd_gotoxy(1,2);

lcd_putc("T. Placas S.    ");

break;

case 2:

lcd_gotoxy(1,2);

lcd_putc("Luminosidad    ");

break;

case 3:

MENU_2=0;

break;

}

if ((ENTER_2==1)&&(MENU_2==0)) {

lcd_gotoxy(1,1);

lcd_putc("T. Ambiental    ");

lcd_gotoxy(1,2);

printf(lcd_putc,"T [°C]= %u ",temp);

}

if ((ENTER_2==1)&&(MENU_2==1)){
```

```
lcd_gotoxy(1,1);

lcd_putc("T. Placas    ");

lcd_gotoxy(1,2);

printf(lcd_putc,"T [ºC]= %u ",temp_p);

}

if ((ENTER_2==1)&&(MENU_2==2)){

lcd_gotoxy(1,1);

lcd_putc("Luminosidad    ");

lcd_gotoxy(1,2);

printf(lcd_putc,"Lum= %u ",lum);

}

break; // TEMPERATURAS

case 2: // PLACAS SOLARES

if (ENTER_3==1) {

ENTER_3=0;

CNT_IN=CNT_IN-1;

}

if (ENTER_1==0) {

lcd_gotoxy(1,1);

lcd_putc("Visualizar    ");

lcd_gotoxy(1,2);

lcd_putc("Placas Solares  ");

}
```

```
}

if ((ENTER_1==1)&&(ENTER_2==0)) {

lcd_gotoxy(1,1);

lcd_putc("Placas Solares  ");

switch (MENU_2) {

case -1:

    MENU_2=2;

break;

case 0:

    lcd_gotoxy(1,2);

    lcd_putc("Voltaje      ");

break;

case 1:

    lcd_gotoxy(1,2);

    lcd_putc("Intensidad      ");

break;

case 2:

    lcd_gotoxy(1,2);

    lcd_putc("Potencia      ");

break;

case 3:

    MENU_2=0;
```

```
break;  
}  
}  
  
if ((ENTER_2==1)&&(MENU_2==0)){  
  
lcd_gotoxy(1,1);  
  
lcd_putc("P.S. Voltaje ");  
  
lcd_gotoxy(1,2);  
  
printf(lcd_putc,"V [V]= %u ",V1);  
  
}  
  
if ((ENTER_2==1)&&(MENU_2==1)){  
  
lcd_gotoxy(1,1);  
  
lcd_putc("P.S. Intensidad ");  
  
lcd_gotoxy(1,2);  
  
printf(lcd_putc,"I [A]= %u ",I1);  
  
}  
  
if ((ENTER_2==1)&&(MENU_2==2)){  
  
lcd_gotoxy(1,1);  
  
lcd_putc("P.S. Potencia ");  
  
lcd_gotoxy(1,2);  
  
printf(lcd_putc,"P [W]= %u ",PSW);  
  
}  
  
break; // PLACAS SOLARES
```



case 3: // BATERIAS

```
if (ENTER_3==1) {  
    ENTER_3=0;  
    CNT_IN=CNT_IN-1;  
}  
  
if (ENTER_1==0) {  
    lcd_gotoxy(1,1);  
    lcd_putc("Visualizar      ");  
    lcd_gotoxy(1,2);  
    lcd_putc("Baterias      ");  
}  
  
if ((ENTER_1==1)&&(ENTER_2==0)) {  
    lcd_gotoxy(1,1);  
    lcd_putc("Baterias      ");  
    switch (MENU_2) {  
        case -1:  
            MENU_2=2;  
            break;  
        case 0:  
            lcd_gotoxy(1,2);  
            lcd_putc("Voltaje      ");  
            break;  
    }  
}
```

case 1:

```
lcd_gotoxy(1,2);  
lcd_putc("Intensidad    ");  
break;
```

case 2:

```
lcd_gotoxy(1,2);  
lcd_putc("Potencia    ");  
break;
```

case 3:

```
MENU_2=0;
```

```
break;
```

```
}
```

```
}
```

```
if ((ENTER_2==1)&&(MENU_2==0)){
```

```
lcd_gotoxy(1,1);  
lcd_putc("B. Voltaje    ");  
lcd_gotoxy(1,2);  
printf(lcd_putc,"V [V]= %u ",V2);
```

```
}
```

```
if ((ENTER_2==1)&&(MENU_2==1)){
```

```
lcd_gotoxy(1,1);  
lcd_putc("B. Intensidad    ");
```



```
lcd_gotoxy(1,2);

printf(lcd_putc,"I [A]= %u ",I2);

}

if ((ENTER_2==1)&&(MENU_2==2)){

lcd_gotoxy(1,1);

lcd_putc("B. Potencia    ");

lcd_gotoxy(1,2);

printf(lcd_putc,"P [W]= %u ",BW);

}

break; // BATERIAS

case 4: // CONTADOR VOLUMETRICO

if (ENTER_3==1) {

ENTER_3=0;

CNT_IN=CNT_IN-1;

}

if (ENTER_1==0) {

lcd_gotoxy(1,1);

lcd_putc("Visualizar    ");

lcd_gotoxy(1,2);

lcd_putc("Contador Vol.  ");

}

if ((ENTER_1==1)&&(ENTER_2==0)) {
```

```
lcd_gotoxy(1,1);

lcd_putc("Contador Vol. ");

switch (MENU_2) {

    case -1:

        MENU_2=0;

        break;

    case 0:

        lcd_gotoxy(1,2);

        lcd_putc("Volumen      ");

        break;

    case 1:

        MENU_2=0;

        break;

    }

}

if ((ENTER_2==1)&&(MENU_2==0)) {

    lcd_gotoxy(1,1);

    lcd_putc("Volumen      ");

    lcd_gotoxy(1,2);

    printf(lcd_putc,"V [porcj.]=%u ",C_VOL);

}

break; // CONTADOR VOLUMETRICO
```



```
case 5: // RS232
```

```
if (ENTER_3==1) {  
  
    ENTER_3=0;  
  
    CNT_IN=CNT_IN-1;  
  
}  
  
if (ENTER_1==0) {  
  
    lcd_gotoxy(1,1);  
  
    lcd_putc("Comunicacion    ");  
  
    lcd_gotoxy(1,2);  
  
    lcd_putc("RS232      ");  
  
}  
  
if ((ENTER_1==1)&&(ENTER_2==0)) {  
  
    lcd_gotoxy(1,1);  
  
    lcd_putc("RS232      ");  
  
    switch (MENU_2) {  
  
        case -1:  
  
            MENU_2=1;  
  
            break;  
  
        case 0:  
  
            lcd_gotoxy(1,2);  
  
            lcd_putc("Conexion PC    ");  
  
            break;  
    }  
}
```

case 1:

```
lcd_gotoxy(1,2);  
lcd_putc("Reset EEPROM  ");  
break;
```

case 2:

```
MENU_2=0;
```

```
break;
```

```
}
```

```
}
```

```
if ((ENTER_2==1)&&(MENU_2==0)){
```

```
    if (aux2<aux) {
```

```
        aux2++;
```

```
        printf("##IS01## ");
```

```
        j = read_ext_eeprom(adr);
```

```
        adr++;
```

```
        printf("%Lu",j);
```

```
        printf("-");
```

```
        j = read_ext_eeprom(adr);
```

```
        adr++;
```

```
        printf("%Lu",j);
```

```
        printf("-");
```



```
j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(" ");

j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(".");

j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(", DATA RECORD %Lu",DR);

DR++;

printf(": ");

j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(" ");

j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(" ");
```

```
j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(",");

j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(",");
```

```
j = read_ext_eeprom(adr);

adr++;

printf("%Lu",j);

printf(",");

printf("\r\n");

}

if (aux2==aux) {

lcd_gotoxy(1,3);

lcd_putc("Guardado ");

delay_ms(1000);

ENTER_2=0;

CNT_IN=CNT_IN-1;

lcd_gotoxy(1,3);

lcd_putc("      ");

aux2=0;

adr=0;

DR=1;

}

}

if ((ENTER_2==1)&&(MENU_2==1)){

i=0;

DR=1;
```

```
adr=0;  
  
aux=0;  
  
aux2=0;  
  
lcd_gotoxy(1,3);  
  
lcd_putc("Guardado ");  
  
delay_ms(1000);  
  
ENTER_2=0;  
  
CNT_IN=CNT_IN-1;  
  
lcd_gotoxy(1,3);  
  
lcd_putc("      ");  
  
}  
  
break; // RS232  
  
case 6: // CONFIGURACIÓN FECHA Y HORA  
  
if (ENTER_1==0) {  
  
lcd_gotoxy(1,1);  
  
lcd_putc("Configuración   ");  
  
lcd_gotoxy(1,2);  
  
lcd_putc("Fecha y Hora   ");  
  
}  
  
if ((ENTER_1==1)&&(ENTER_2==0)) {  
  
lcd_gotoxy(1,1);  
  
lcd_putc("C. Fecha y Hora   ");
```



```
switch (MENU_2) {  
    case -1:  
        MENU_2=4;  
        break;  
  
    case 0:  
        lcd_gotoxy(1,2);  
        lcd_putc("Editar Dia      ");  
        MENU_3=dia;  
        break;  
  
    case 1:  
        lcd_gotoxy(1,2);  
        lcd_putc("Editar Mes      ");  
        MENU_3=mes;  
        break;  
  
    case 2:  
        lcd_gotoxy(1,2);  
        lcd_putc("Editar Año      ");  
        MENU_3=anio;  
        break;  
  
    case 3:  
        lcd_gotoxy(1,2);  
        lcd_putc("Editar Hora      ");
```

```
MENU_3=hora;  
  
break;  
  
case 4:  
  
lcd_gotoxy(1,2);  
  
lcd_putc("Editar Minutos ");  
  
MENU_3=min;  
  
break;  
  
case 5:  
  
MENU_2=0;  
  
break;  
  
}  
  
}  
  
if ((ENTER_2==1)&&(MENU_2==0)){  
  
if (MENU_3>31) {  
  
MENU_3=1;  
  
}  
  
if (MENU_3<1) {  
  
MENU_3=31;  
  
}  
  
if (MENU_3<10) {  
  
lcd_gotoxy(1,2);  
  
printf(lcd_putc,"Fecha= 0%u ",MENU_3);
```

```

}

if (MENU_3>9) {

    lcd_gotoxy(1,2);

    printf(lcd_putc,"Fecha= %u ",MENU_3);

}

if (mes<10) {

    lcd_gotoxy(11,2);

    printf(lcd_putc,"/ 0%u ",mes);

}

if (mes>9) {

    lcd_gotoxy(11,2);

    printf(lcd_putc,"/ %u ",mes);

}

lcd_gotoxy(1,1);

lcd_putc("C. Fecha y Hora ");

lcd_gotoxy(1,3);

lcd_putc(" --");

if (ENTER_3==1){

    dia=MENU_3;

    ds1307_set_date_time(dia,mes,anio,dow,hora,min,sec);

    ds1307_get_date(dia,mes,anio,dow);    /// se obtiene la fecha
}

```

```
lcd_gotoxy(1,3);

lcd_putc("Guardado ");

delay_ms(1000);

ENTER_2=0;

ENTER_3=0;

CNT_IN=CNT_IN-2;

MENU_3=0;

lcd_gotoxy(1,3);

lcd_putc("      ");

}

}

if ((ENTER_2==1)&&(MENU_2==1)) {

if (MENU_3>12) {

MENU_3=1;

}

if (MENU_3<1) {

MENU_3=12;

}

if (dia<10) {

lcd_gotoxy(1,2);

printf(lcd_putc,"Fecha= 0%u ",dia);

}

}
```



```
if (dia>9) {

lcd_gotoxy(1,2);

printf(lcd_putc,"Fecha= %u ",dia);

}

if (MENU_3<10) {

lcd_gotoxy(11,2);

printf(lcd_putc,"/ 0%u ",MENU_3);

}

if (MENU_3>9) {

lcd_gotoxy(11,2);

printf(lcd_putc,"/ %u ",MENU_3);

}

lcd_gotoxy(1,1);

lcd_putc("C. Fecha y Hora ");

lcd_gotoxy(1,3);

lcd_putc("      --");

if (ENTER_3==1){

mes=MENU_3;

ds1307_set_date_time(dia,mes,anio,dow,hora,min,sec);

ds1307_get_date(dia,mes,anio,dow);    /// se obtiene la fecha

lcd_gotoxy(1,3);
```

```
lcd_putc("Guardado ");

delay_ms(1000);

ENTER_2=0;

ENTER_3=0;

CNT_IN=CNT_IN-2;

MENU_3=0;

lcd_gotoxy(1,3);

lcd_putc("      ");

}

}

if ((ENTER_2==1)&&(MENU_2==2)) {

if (MENU_3>99) {

MENU_3=16;

}

if (MENU_3<16) {

MENU_3=99;

}

lcd_gotoxy(1,1);

lcd_putc("C. Fecha y Hora ");

lcd_gotoxy(1,2);

printf(lcd_putc,"Año= 20%u ",MENU_3);
```



```
if (ENTER_3==1){

    anio=MENU_3;

    ds1307_set_date_time(dia,mes,anio,dow,hora,min,sec);

    ds1307_get_date(dia,mes,anio,dow);    /// se obtiene la fecha

    lcd_gotoxy(1,3);

    lcd_putc("Guardado ");

    delay_ms(1000);

    ENTER_2=0;

    ENTER_3=0;

    CNT_IN=CNT_IN-2;

    MENU_3=0;

    lcd_gotoxy(1,3);

    lcd_putc("      ");

}

}

if ((ENTER_2==1)&&(MENU_2==3)){

    if (MENU_3==255) {

        MENU_3=23;

    }

    if (MENU_3>23) {

        MENU_3=0;

    }

}
```

```
if (MENU_3<10) {  
  
    lcd_gotoxy(1,2);  
  
    printf(lcd_putc,"Hora= 0%u ",MENU_3);  
  
}  
  
if (MENU_3>9) {  
  
    lcd_gotoxy(1,2);  
  
    printf(lcd_putc,"Hora= %u ",MENU_3);  
  
}  
  
if (min<10) {  
  
    lcd_gotoxy(10,2);  
  
    printf(lcd_putc,: 0%u ",min);  
  
}  
  
if (min>9) {  
  
    lcd_gotoxy(10,2);  
  
    printf(lcd_putc,: %u ",min);  
  
}  
  
lcd_gotoxy(1,1);  
  
lcd_putc("C. Fecha y Hora ");  
  
lcd_gotoxy(1,3);  
  
lcd_putc(" --");
```



```
if (ENTER_3==1){

    hora=MENU_3;

    ds1307_set_date_time(dia,mes,anio,dow,hora,min,sec);

    ds1307_get_time(hora,min,sec);

    lcd_gotoxy(1,3);

    lcd_putc("Guardado ");

    delay_ms(1000);

    ENTER_2=0;

    ENTER_3=0;

    CNT_IN=CNT_IN-2;

    MENU_3=0;

    lcd_gotoxy(1,3);

    lcd_putc("      ");

}

}

if ((ENTER_2==1)&&(MENU_2==4)){

    if (MENU_3==255) {

        MENU_3=59;

    }

    if (MENU_3>59) {

        MENU_3=0;

    }

}
```

```
if (hora<10) {  
  
    lcd_gotoxy(1,2);  
  
    printf(lcd_putc,"Hora= 0%u ",hora);  
  
}  
  
if (hora>9) {  
  
    lcd_gotoxy(1,2);  
  
    printf(lcd_putc,"hora= %u ",hora);  
  
}  
  
if (MENU_3<10) {  
  
    lcd_gotoxy(10,2);  
  
    printf(lcd_putc,: 0%u ",MENU_3);  
  
}  
  
if (MENU_3>9) {  
  
    lcd_gotoxy(10,2);  
  
    printf(lcd_putc,: %u ",MENU_3);  
  
}  
  
lcd_gotoxy(1,1);  
  
lcd_putc("C. Fecha y Hora ");  
  
lcd_gotoxy(1,3);  
  
lcd_putc("    --");  
  
if (ENTER_3==1){
```



```

min=MENU_3;

ds1307_set_date_time(dia,mes,anio,dow,hora,min,sec);

ds1307_get_time(hora,min,sec);    /// se obtiene la fecha

lcd_gotoxy(1,3);

lcd_putc("Guardado ");

delay_ms(1000);

ENTER_2=0;

ENTER_3=0;

CNT_IN=CNT_IN-2;

MENU_3=0;

lcd_gotoxy(1,3);

lcd_putc("      ");

}

}

break;// CONFIGURACIÓN FECHA Y HORA

case 7:

MENU_1=0;

break;

}

}

}

}

}

```

2. Programa PIC 16F883 (1)

```
#include <16F883.h>

#device adc=8

#fuses XT, NOWDT, NOLVP, NOPROTECT

#use delay(crystal=20MHz)

#use i2c(SLAVE, SDA=PIN_C4, SCL=PIN_C3, address=0x13)

#use RS232(BAUD=9600,BITS=8,PARITY=N,XMIT=PIN_C6,RCV=PIN_C7)

int8 x;

#INT_SSP

void ssp_interrupt()

{

    int8 incoming, state;

    state = i2c_isr_state();

    if(state < 0x80) // Master is sending data

    {

        incoming = i2c_read();

    }

    if(state >= 0x80) // Master is requesting data from slave

    {

        i2c_write(x);

    }

}
```



```
//=====

void main () {

setup_adc_ports(NO_ANALOGS|VSS_VDD);

setup_adc(ADC_CLOCK_DIV_2);

set_adc_channel(0);

enable_interrupts(INT_SSP);

enable_interrupts(GLOBAL);

while(1)

{

x=read_adc();

delay_ms(500);

}

}
```

3. Programa PIC 16F883 (2)

```
#include <16F883.h>

#device adc=8

#fuses XT, NOWDT, NOLVP, NOPROTECT

#use delay(crystal=20MHz)

#use i2c(SLAVE, SDA=PIN_C4, SCL=PIN_C3, address=0xA0)

#use RS232(BAUD=9600,BITS=8,PARITY=N,XMIT=PIN_C6,RCV=PIN_C7)

int8 x;

#INT_SSP

void ssp_interrupt()

{

    int8 incoming, state;

    state = i2c_isr_state();

    if(state < 0x80) // Master is sending data

    {

        incoming = i2c_read();

    }

    if(state >= 0x80) // Master is requesting data from slave

    {

        i2c_write(x);

    }

}
```



```
//=====
void main () {
    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_CLOCK_DIV_2);
    set_adc_channel(0);
    enable_interrupts(INT_SSP);
    enable_interrupts(GLOBAL);
    while(1)
    {
        x=read_adc();
        delay_ms(500);
    }
}
```

4. Programa PIC 16F883 (3)

```
#include <16F883.h>

#device adc=10

#fuses XT, NOWDT, NOLVP, NOPROTECT

#use delay(crystal=20MHz)

#use i2c(SLAVE, SDA=PIN_C4, SCL=PIN_C3, address=0xA1)

#use RS232(BAUD=9600,BITS=8,PARITY=N,XMIT=PIN_C6,RCV=PIN_C7)

int8 x;

#INT_SSP

void ssp_interrupt()

{

    int8 incoming, state;

    state = i2c_isr_state();

    if(state < 0x80) // Master is sending data

    {

        incoming = i2c_read();

    }

    if(state >= 0x80) // Master is requesting data from slave

    {

        i2c_write(x);

    }

}
```



```
//=====

void main () {

setup_adc_ports(NO_ANALOGS|VSS_VDD);

setup_adc(ADC_CLOCK_DIV_2);

set_adc_channel(0);

enable_interrupts(INT_SSP);

enable_interrupts(GLOBAL);

while(1)

{

x=read_adc();

delay_ms(500);

}

}
```

5. Programa PIC 16F883 (4)

```
#include <16F883.h>

#device adc=8

#fuses XT, NOWDT, NOLVP, NOPROTECT

#use delay(crystal=20MHz)

#use i2c(SLAVE, SDA=PIN_C4, SCL=PIN_C3, address=0xA2)

#use RS232(BAUD=9600,BITS=8,PARITY=N,XMIT=PIN_C6,RCV=PIN_C7)

int8 x;

#INT_SSP

void ssp_interrupt()

{

    int8 incoming, state;

    state = i2c_isr_state();

    if(state < 0x80) // Master is sending data

    {

        incoming = i2c_read();

    }

    if(state >= 0x80) // Master is requesting data from slave

    {

        i2c_write(x);

    }

}
```



```
//=====

void main () {

setup_adc_ports(NO_ANALOGS|VSS_VDD);

setup_adc(ADC_CLOCK_DIV_2);

set_adc_channel(0);

enable_interrupts(INT_SSP);

enable_interrupts(GLOBAL);

while(1)

{

x=read_adc();

delay_ms(500);

}

}
```