



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

Grado en Ingeniería Electrónica Industrial y Automática

# DESARROLLO DE UNA UNIDAD DE ADQUISICIÓN Y MONITORIZACIÓN DE DATOS PARA UN VEHÍCULO



**Volumen III**

**Anexos**

**Autor:** Oriol Bel Carbó  
**Director:** Manuel Manzanares Brotons  
**Convocatòria:** Mayo 2018



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

```
#include "18f4550.h"           // Pic a utilizar
#device adc=10                // Usa resolución de 10 bits
#use delay(clock=4000000)     // Cristal a utilizar
#fuses xt,nowdt,noput,nobrownout,nolvp,noprotect
#include "lcd.c"               // Libreria para utilizar LCD por el puerto D
#use standard_io (D)
#use standard_io (B)
#use standard_io (C)
#use I2C(master, I2C1, FAST = 400000)
#use rs232(uart1,baud=9600,xmit=PIN_c6,rcv=PIN_c7,stream=HC,parity=N, bits=8)
```

```
int Medir_barrido(){
```

```
float valor,temp;
```

```
float valor1,t0,temp1;
```

```
int value;
```

```
    setup_adc_ports (AN0);           //tots el sporta analogicsvalor positiu
```

```
    set_adc_channel (0);           // choses channel A0
```

```
    delay_us(20);
```

```
    valor=read_adc ();           // conversio adc
```

```
    temp=valor*(0.488);
```

```
    //! setup_adc (adc_off);           // Apaga ADC
```

```
    //! setup_adc_ports (AN1);           //tots el sporta analogics valor negatiu
```

```
    set_adc_channel (1);          // choses channel A0
    delay_us(20);
    valor1=read_adc ();          // conversio adc
    temp1=valor1*(0.488);
    t0=temp-temp1;

    value=t0;

    return value;

}

int Luz_barrido(){

float valor,llum;
//char Bigdata;
int Bigdata;

setup_adc (adc_clock_internal);
delay_ms (500);
//!setup_adc_ports ();          //tots el sporta analogics
    set_adc_channel (2);          // choses channel A0
    delay_us(20);
    valor=read_adc ();          // conversio adc
    llum=valor;

    if(llum>75){                  //10 bits de adc valor de 1024
        //output_high(PIN_C0);
        Bigdata=1;}
    else{
```

```
//output_low(PIN_C0);
Bigdata=0;}

return bigdata;

}

int Temps_barrido(){

int second, minute, hour, date, month, year, Bigdata;

char time[] = "TIME: : : ";
char calendar[] = "DATE: / /20 ";

i2c_start();           // Start I2C protocol
i2c_write(0xD0);       // DS1307 address
i2c_write(0);          // Send register address (time seconds register)
i2c_start();           // Restart I2C
i2c_write(0xD1);       // Initialize data read
second = i2c_read(1);   // Read seconds from register 0
minute = i2c_read(1);  // Read minutes from register 1
hour = i2c_read(1);    // Read hours from register 2
i2c_read(1);           // Read day from register 3 (not used)
date = i2c_read(1);    // Read date from register 4
month = i2c_read(1);   // Read month from register 5
year = i2c_read(0);    // Read year from register 6
i2c_stop();            // Stop I2C protocol
//DS1307_display();    // Diaplay time & calendar
delay_ms(50);          // Wait 50ms

second = (second >> 4) * 10 + (second & 0x0F);
```

```
minute = (minute >> 4) * 10 + (minute & 0x0F);
hour = (hour >> 4) * 10 + (hour & 0x0F);
date = (date >> 4) * 10 + (date & 0x0F);
month = (month >> 4) * 10 + (month & 0x0F);
year = (year >> 4) * 10 + (year & 0x0F);
// End conversion
time[12] = second % 10 + 48;
time[11] = second / 10 + 48;
time[9] = minute % 10 + 48;
time[8] = minute / 10 + 48;
time[6] = hour % 10 + 48;
time[5] = hour / 10 + 48;
calendar[14] = year % 10 + 48;
calendar[13] = year / 10 + 48;
calendar[9] = month % 10 + 48;
calendar[8] = month / 10 + 48;
calendar[6] = date % 10 + 48;
calendar[5] = date / 10 + 48;

Bigdata=second;
//!// bigdata2=minute;
// printf(lcd_putc,"%i",bigdata);
//! delay_ms(2000);
//! lcd_putc(" temps segons");
//! delay_ms(2000);

return second;
}

int Bateria_barrido(){
```

```
float bat,valor,value;

int BB,Bigdata;

setup_adc (adc_clock_internal);
delay_ms (100);

set_adc_channel (3);          // choses channel A0
delay_us(20);
valor=read_adc ();           // conversio adc
// value=valor*0,357;

if (valor < 658)//658
    bat=0;
if (valor > 1020)//1020
    bat=100;
if ((valor > 658)&&(valor < 1020))//658
    bat=(valor-658)/3,66;//658

BB=bat;
Bigdata=BB;
return bigdata;

}

//-----0-dades de temps-----

int distancia_barrido (){

int8 cont;

int valor;
```

```
SET_TRIS_C(0x7F);

while (input(pin_C1)==1);
    cont++;
while (input(pin_C1)==0);
valor=cont;
return valor;

}

//-----SAVE DATA BARRIDO-----

void save(select,Bigdata){

int s,h,ss,sec,u;
int data,byta,bytb;
long max;
int16 vel;

u++;

switch (select){

case 1:
lcd_putc("\f");
    lcd_gotoxy(5,1); // posa cursor aquesta posicio
printf(lcd_putc,"Temperatura");
```

```
break;
```

```
case 2:
```

```
//! if (bigdata==1)
```

```
//! data="ON";
```

```
//! else
```

```
//! data="OFF";
```

```
    lcd_putc("\f");
```

```
lcd_gotoxy(5,1); // posa cursor aquesta posicio
```

```
printf(lcd_putc,"LLUM");
```

```
break;
```

```
case 3:
```

```
lcd_putc("\f");
```

```
    lcd_gotoxy(5,1); // posa cursor aquesta posicio
```

```
printf(lcd_putc,"SEGONS");
```

```
if(h==0)
```

```
s=bigdata;
```

```
h++;
```

```
if(h==10) //380 metres porque entra cada a =38
```

```
    //es coregeixen els seg si hi han canvi de minut
```

```
    ss=bigdata;
```

```
break;
```



case 4:

```
lcd_putc("\f");
```

```
lcd_gotoxy(5,1); // posa cursor aquesta posicio
```

```
printf(lcd_putc,"M/S");
```

```
if(h==10){
```

```
    sec=ss-s;
```

```
    vel=380/sec;
```

```
if(s>ss){ //es coregeixen els seg si hi han canvi de minut
```

```
    sec=(60-s)+ss;
```

```
    vel=380/sec;
```

```
    bigdata=vel;
```

```
    h=0;
```

```
}
```

```
bigdata=vel;
```

```
}
```

```
break;
```

```
//!
```

case 5:

```
lcd_putc("\f");
```

```
lcd_gotoxy(5,1); // posa cursor aquesta posicio
```

```
printf(lcd_putc,"BATERIA");
```

```
break;
```

```
}

//!max=(byte0)*(byte1+1);
//!max=(max/32000)*100;
//!lcd_gotoxy(13,1);
//!printf(lcd_putc,"%Lu",max);

if (bytb==0xFF){
byta=byta+1;
bytb=0x00;
}

bytb++;

//!if ((byta==0xFF)&&(bytb==0x7F)){
//!printf(lcd_putc,"memory full");
//!delay_ms(1000);
//!byta=0x00;
//!bytb=0x00;
//!}
lcd_putc("\f");
lcd_gotoxy(13,2);
printf(lcd_putc,"%i",bytb);
lcd_gotoxy(13,1);
printf(lcd_putc,"%i",u);

I2C_Start();
I2C_write(0xA2);
I2C_write(byta);
```

```
I2C_write(bytb);
I2C_write(bigdata);
I2C_Stop();

delay_ms(200);

I2C_Start();
I2C_write(0xA2);
I2C_write(byta); //direcci me significativa
I2C_write(bytb); //direccio menys significativa

I2C_Start();
I2C_write(0xA3); //utim bit es 1 per llegir
data=I2C_read();
I2C_Stop();

//lcd_putc("\f");
lcd_gotoxy(5,2);
printf(lcd_putc,"%i",data);
lcd_gotoxy(1,2);
printf(lcd_putc,"%i",bigdata);
delay_ms(200);

}

//-----SAVE DATA-----

void save(t0,valor,valor1){

float data;
```

```
if (valor1>valor){
    lcd_gotoxy(9,1); // posa cursor aquesta posicio
    printf(lcd_putc,"%c",'-'); // °C
}
else{
    lcd_gotoxy(9,1); // posa cursor aquesta posicio
    printf(lcd_putc,"%c",'+'); // °C
}

I2C_Start();
I2C_write(0xA2);
I2C_write(0x00);
I2C_write(0x01);
I2C_write(t0);
I2C_Stop();

delay_ms(300);

I2C_Start();
I2C_write(0xA2);
I2C_write(0x00);
I2C_write(0x01);
I2C_write(t0);
I2C_Start();
I2C_write(0xA3);
data=I2C_read();
I2C_Stop();

lcd_gotoxy(1,1);
lcd_putc("SData \n");
```

```
lcd_gotoxy(10,1);
printf(lcd_putc,"%f",data);

lcd_gotoxy(14,1);
lcd_putc("oC");
delay_ms(300);
}

//-----1-Medir temperatura-----

void medir (select){

float valor,temp;
float valor1,t0,temp1;
int8 a,Bigdata;

switch (select){

case 0:
lcd_init();           // Inicia LCD
lcd_putc(" Termometro\n");      // eSCRIU
delay_ms(500);
lcd_putc("\f");           // Clean
lcd_putc("Temperatura\n");      // Escriu
lcd_putc("actual");         // Escriu
delay_ms(500);
lcd_gotoxy(14,2);         // Posa cursor segona linea segment 14
lcd_putc("oC");

output_low(PIN_B2); //mANTE EL PIN 0 A 1 PERQUE QUAN POLSI EL 6 SURTI DEL LOOP
```

```
setup_adc (adc_clock_internal);

delay_ms (300);

a=0;

while (input(PIN_B7)==1){           //quan es polsa el pulsador que tanca el pin 6 surt del lop per
tonrar depres al main menu

    setup_adc_ports (AN0);         //tots el sporta analogicsvalor positiu
    set_adc_channel (0);           // choses channel A0
    delay_us(20);
    valor=read_adc ();             // conversio adc
    temp=valor*(0.488);

    //! setup_adc (adc_off);       // Apaga ADC

    //! setup_adc_ports (AN1);     //tots el sporta analogics valor negatiu
    set_adc_channel (1);           // choses channel A0
    delay_us(20);
    valor1=read_adc ();            // conversio adc
    temp1=valor1*(0.488);

    if (valor1>valor){
    t0=temp1;
    lcd_gotoxy(7,2); // posa cursor aquesta posicio
    printf(lcd_putc,"%c",'-'); // °C
    }
    else{
```

```
lcd_gotoxy(7,2); // posa cursor aquesta posicio
printf(lcd_putc,"%c",'+'); // °C
t0=temp-temp1;
}

// el "0.48875" sale de dividir 5/1023 y el resultado de eso dividirlo por 10--->El 10 son els mV/°C,
cada 10mV canvia un grau la temp.

// el 5 sale de los 5 voltios aplicado a el voltage de referencia, los 1023 salen
// de los 10 bit de resolucio del conversor analogico digital que seleccionamos.

//!

lcd_gotoxy(8,2); // POsa cursor aquesta posicio
lcd_putc(" "); // cleans this screen section
lcd_gotoxy(8,2); // posa cursor aquesta posicio
printf(lcd_putc,"%02.1f",t0); // °C
delay_ms (100);

if(input(pin_B6)==0)
    a++;

switch (a){

case 0:

a=0;
break;

case 1:
```

```
save(t0,valor,valor1);
break;
case 2:

lcd_putc("\f");          // Clean
lcd_putc("Stop saving\n");      // Escriu
delay_ms(1000);
    a=0;
break;
}

}

//! reset_cpu();
lcd_init();          // -----
lcd_putc("\f");
lcd_init();
lcd_putc(" Main Menu\n");      //Prepara el lcd per tonrar al menu principal
delay_ms(1000);
lcd_putc("\f");
lcd_putc(" choose sensor\n");
delay_ms(1000);
lcd_putc("\f");
lcd_putc(" 1-T°C 2-L 3-T\n");      //-----
lcd_gotoxy(1, 2);
lcd_putc(" 4-D 5-S 6-B");
//break;
```



```
break;

//-----
//   if (select==1){
    case 1:
        setup_adc_ports (AN0);      //tots el sporta analogicsvalor positiu
        set_adc_channel (0);        // choses channel A0
        delay_us(20);
        valor=read_adc ();          // conversio adc
        temp=valor*(0.488);
        //! setup_adc (adc_off);      // Apaga ADC

        //! setup_adc_ports (AN1);    //tots el sporta analogics valor negatiu
        set_adc_channel (1);        // choses channel A0
        delay_us(20);
        valor1=read_adc ();         // conversio adc
        temp1=valor1*(0.488);
        t0=temp-temp1;

        bigdata=t0;
        save(select,bigdata);

    }
//-----
}

//-----2-LUZ-----
void luz (select){
```

```
float valor,llum;
//char Bigdata;
int Bigdata;

switch (select){

    case 0:

        setup_adc (adc_clock_internal);
        delay_ms (500);
        //!setup_adc_ports ();      //tots el sporta analogics
        set_adc_channel (2);      // choses channel A0
        delay_us(20);
        valor=read_adc ();      // conversio adc
        llum=valor;

        if(llum>75)      //10 bits de adc valor de 1024
            output_high(PIN_C0);
        else
            output_low(PIN_C0);

        lcd_init();      // -----
        lcd_putc("\f");
        lcd_init();
        lcd_putc(" Main Menu\n");      //Prepara el lcd per tonrar al menu principal
        delay_ms(500);
        lcd_putc("\f");
        lcd_putc(" choose sensor\n");
```

```
    delay_ms(500);
    lcd_putc("\f");
    lcd_putc(" 1-T°C 2-L 3-T\n");          //-----
    lcd_gotoxy(1, 2);
    lcd_putc(" 4-D 5-S 6-B");

break;
    case 2:
        setup_adc (adc_clock_internal);
    delay_ms (500);
    //!setup_adc_ports ();          //tots el sporta analogics
    set_adc_channel (2);          // choses channel A0
    delay_us(20);
    valor=read_adc ();          // conversio adc
    llum=valor;

    if(llum<947){          //10 bits de adc valor de 1024
        output_high(PIN_C0);
        Bigdata=1;}
    else{
        output_low(PIN_C0);
        Bigdata=0;}
        save(select,Bigdata);
break;
    }
}

//-----3-TEMPS-----

void DS1307_display(){
```

```
int8 edit(parameter, x, y){
int8 i;
while(input(PIN_B5)==0);          // Wait until button RB2 released
while(input(PIN_B7)==1){
while(input(PIN_B6)==0){        // If button RB3 is pressed
parameter++;
if(i == 0 && parameter > 23)     // If hours > 23 ==> hours = 0
parameter = 0;
if(i == 1 && parameter > 59)     // If minutes > 59 ==> minutes = 0
parameter = 0;
if(i == 2 && parameter > 31)     // If date > 31 ==> date = 1
parameter = 1;
if(i == 3 && parameter > 12)     // If month > 12 ==> month = 1
parameter = 1;
if(i == 4 && parameter > 99)     // If year > 99 ==> year = 0
parameter = 0;

lcd_gotoxy(x, y);
printf(lcd_putc,"%02u", parameter); // Display parameter
delay_ms(70);                    // Wait 200ms
}

lcd_gotoxy(x, y);
lcd_putc(" ");
delay_ms(200);
lcd_gotoxy(x, y);                // Print two spaces
printf(lcd_putc,"%02u", parameter); // Print parameter
delay_ms(200);
if(input(PIN_B5)==0){           // If button RB2 is pressed
```

```
    i++;                // Increment 'i' for the next parameter
    return parameter;   // Return parameter value and exit
}
}
}
```

```
void Temps (select){
```

```
int i, second, minute, hour, date, month, year, Bigdata;
```

```
    char time[] = "TIME: : : ";
```

```
char calendar[] = "DATE: / /20 ";
```

```
lcd_init();           // -----
```

```
    lcd_putc("\f");
```

```
output_low(PIN_B2);
```

```
switch (select){
```

```
    case 0:            //NO BARRIDO
```

```
while(input(PIN_B7)==1){
```

```
    if(input(PIN_B5)==0){           // If RB2 button is pressed
```

```
        i = 0;
```

```
        hour = edit(hour, 6, 1);
```

```
        minute = edit(minute, 9, 1);
```

```
        date = edit(date, 6, 2);
```

```
        month = edit(month, 9, 2);
```

```
        year = edit(year, 14, 2);
```

```
        // Convert decimal to BCD
```

```
        minute = ((minute / 10) << 4) + (minute % 10);
```

```
hour = ((hour / 10) << 4) + (hour % 10);
date = ((date / 10) << 4) + (date % 10);
month = ((month / 10) << 4) + (month % 10);
year = ((year / 10) << 4) + (year % 10);
// End conversion
// Write data to DS3207 RTC
i2c_start();           // Start I2C protocol
i2c_write(0xD0);       // DS1307 address
i2c_write(0);          // Send register address (time seconds register)
i2c_write(0);          // Reset seconds and start oscillator
i2c_write(minute);     // Write minutes value to DS
i2c_write(hour);      // Write hours value to DS
i2c_write(1);         // Write day value (not used)
i2c_write(date);      // Write date value to DS
i2c_write(month);     // Write month value to DS
i2c_write(year);      // Write year value to DS
delay_ms(200);        // Wait 200ms
}
i2c_start();           // Start I2C protocol
i2c_write(0xD0);       // DS1307 address
i2c_write(0);          // Send register address (time seconds register)
i2c_start();           // Restart I2C
i2c_write(0xD1);       // Initialize data read
second = i2c_read(1);  // Read seconds from register 0
minute = i2c_read(1);  // Read minutes from register 1
hour = i2c_read(1);    // Read hours from register 2
i2c_read(1);           // Read day from register 3 (not used)
date = i2c_read(1);    // Read date from register 4
month = i2c_read(1);   // Read month from register 5
```

```
year = i2c_read(0);           // Read year from register 6
i2c_stop();                   // Stop I2C protocol
//DS1307_display();          // Display time & calendar
delay_ms(50);                 // Wait 50ms

second = (second >> 4) * 10 + (second & 0x0F);
minute = (minute >> 4) * 10 + (minute & 0x0F);
hour = (hour >> 4) * 10 + (hour & 0x0F);
date = (date >> 4) * 10 + (date & 0x0F);
month = (month >> 4) * 10 + (month & 0x0F);
year = (year >> 4) * 10 + (year & 0x0F);
// End conversion
time[12] = second % 10 + 48;
time[11] = second / 10 + 48;
time[9] = minute % 10 + 48;
time[8] = minute / 10 + 48;
time[6] = hour % 10 + 48;
time[5] = hour / 10 + 48;
calendar[14] = year % 10 + 48;
calendar[13] = year / 10 + 48;
calendar[9] = month % 10 + 48;
calendar[8] = month / 10 + 48;
calendar[6] = date % 10 + 48;
calendar[5] = date / 10 + 48;

//! lcd_init();               // -----
//! lcd_putc("\f");
lcd_gotoxy(1, 1);             // Go to column 1 row 1
```

```

printf(lcd_putc, time);           // Display time
lcd_gotoxy(1, 2);                 // Go to column 1 row 2
printf(lcd_putc,calendar);       // Display calendar
}

lcd_init();                       // -----
lcd_putc("\f");
lcd_init();
lcd_putc(" Main Menu\n");        //Prepara el lcd per tonrar al menu principal
delay_ms(1000);
lcd_putc("\f");
lcd_putc(" choose sensor\n");    //-----
break;

case 3:                            //BARRIDO

i2c_start();                      // Start I2C protocol
i2c_write(0xD0);                  // DS1307 address
i2c_write(0);                     // Send register address (time seconds register)
i2c_start();                      // Restart I2C
i2c_write(0xD1);                  // Initialize data read
second = i2c_read(1);              // Read seconds from register 0
minute = i2c_read(1);             // Read minutes from register 1
hour = i2c_read(1);               // Read hours from register 2
i2c_read(1);                      // Read day from register 3 (not used)
date = i2c_read(1);               // Read date from register 4
month = i2c_read(1);              // Read month from register 5
year = i2c_read(0);               // Read year from register 6
i2c_stop();                       // Stop I2C protocol

```



```
//DS1307_display();           // Diaplay time & calendar
delay_ms(50);                 // Wait 50ms

second = (second >> 4) * 10 + (second & 0x0F);
minute = (minute >> 4) * 10 + (minute & 0x0F);
hour = (hour >> 4) * 10 + (hour & 0x0F);
date = (date >> 4) * 10 + (date & 0x0F);
month = (month >> 4) * 10 + (month & 0x0F);
year = (year >> 4) * 10 + (year & 0x0F);
// End conversion

time[12] = second % 10 + 48;
time[11] = second / 10 + 48;
time[9] = minute % 10 + 48;
time[8] = minute / 10 + 48;
time[6] = hour % 10 + 48;
time[5] = hour / 10 + 48;
calendar[14] = year % 10 + 48;
calendar[13] = year / 10 + 48;
calendar[9] = month % 10 + 48;
calendar[8] = month / 10 + 48;
calendar[6] = date % 10 + 48;
calendar[5] = date / 10 + 48;

Bigdata=second;

//! printf(lcd_putc,"%i",bigdata);
//! delay_ms(2000);
//! lcd_putc(" temps segons");
```

```
//! delay_ms(2000);

// save(select,Bigdata);
lcd_gotoxy(11,2);
printf(lcd_putc,"%i",second);
delay_ms(5000);

break;
}
}

//-----DISTANCIA A PARTIR DE LA QUE MESSURA EL BARRIDO-----
-----

void distancia (){
    int8 a;
    SET_TRIS_C(0x7F);

    while (a<38) {          //cada 38 fa el barrido de data
        while (input(pin_C1)==1);
        a++;
        while (input(pin_C1)==0);

    }
}

//-----4-Distancia-----

void velocidad (select){
```

```
//int tx,a,valor,s,m,h;
//float L=0.5,distancia,velocitat;
int second,minute,hour, date, month,year,ss,mm,hh,s,m,h,Bigdata;
    char time[] = "TIME: : : ";
    char calendar[] = "DATE: / /20 ";
int16 a;
float L=1,distancia,temps_0,velocitat;

    SET_TRIS_C(0x7F);
    //!setup_timer_0(RTCC_INTERNAL|RTCC_DIV_2);
    //enable_interrupts(INT_TIMER0);
    //enable_interrupts(GLOBAL);
    output_low(PIN_B2);
    lcd_putc("\f");

a=0;

switch (select){

    case 0:

while (input(PIN_B7)==1)
{

i2c_start();                // Start I2C protocol
    i2c_write(0xD0);        // DS1307 address
    i2c_write(0);          // Send register address (time seconds register)
    i2c_start();          // Restart I2C
    i2c_write(0xD1);      // Initialize data read
```

```
second = i2c_read(1);           // Read seconds from register 0
minute = i2c_read(1);          // Read minutes from register 1
hour = i2c_read(1);            // Read hours from register 2
i2c_read(1);                   // Read day from register 3 (not used)
date = i2c_read(1);            // Read date from register 4
month = i2c_read(1);           // Read month from register 5
year = i2c_read(0);            // Read year from register 6
i2c_stop();                     // Stop I2C protocol
//DS1307_display();            // Display time & calendar
delay_ms(50);                  // Wait 50ms

second = (second >> 4) * 10 + (second & 0x0F);
minute = (minute >> 4) * 10 + (minute & 0x0F);
hour = (hour >> 4) * 10 + (hour & 0x0F);
date = (date >> 4) * 10 + (date & 0x0F);
month = (month >> 4) * 10 + (month & 0x0F);
year = (year >> 4) * 10 + (year & 0x0F);
//! // End conversion

time[12] = second % 10 + 48;
time[11] = second / 10 + 48;
time[9] = minute % 10 + 48;
time[8] = minute / 10 + 48;
time[6] = hour % 10 + 48;
time[5] = hour / 10 + 48;
calendar[14] = year % 10 + 48;
calendar[13] = year / 10 + 48;
calendar[9] = month % 10 + 48;
calendar[8] = month / 10 + 48;
calendar[6] = date % 10 + 48;
```

```
calendar[5] = date / 10 + 48;
```

```
// MESURA DE VELOCITAT
```

```
while (input(pin_C1)==1);
```

```
a++;
```

```
switch (a){ // a incrementa una unitat cada cop que pasa liman i el primer cop que passa registre e tempd dinici
```

```
case 0:
```

```
    a++;
```

```
    break;
```

```
case 1:
```

```
    m=minute;
```

```
    s=second;
```

```
    h=hour;
```

```
    break;
```

```
}
```

```
if (a==1000){ //es cobreix el kilimetre, es resta el temps actual amb el del registre dinici
```

```
    mm=minute-m;
```

```
    hh=hour-h;
```

```
    ss=second-s;
```

```
if((hh>0)&&(m>minute)){ //es coregeixen els minuts si hi han canvi dhora
    hh=hh-1;
    mm=(60-m)+minute;
}
if((mm>0)&&(s>second)){ //es coregeixen els seg si hi han canvi de minut
    mm=mm-1;
    ss=(60-s)+second;
}

temps_0=(hh*3600)+(mm*60)+ss;
velocitat=1000/temps_0;
a=0;

}
```

```
while (input(pin_C1)==0);
```

```
//! if(second-s==7){
//! distancia=a*L;
//! velocitat=(distancia/10)*3,6;
//! a=0;
//! tx=0;
//! }
```

```
lcd_gotoxy(1, 1); // Go to column 1 row 1
printf(lcd_putc, time); // Display time
lcd_gotoxy(1,2);
printf(lcd_putc,"%f",velocitat);
```

```
lcd_gotoxy(5,2);
lcd_putc(" km/h");
lcd_gotoxy(12,2);
printf(lcd_putc,"%Lu",a);

}

lcd_init();           // -----
lcd_putc("\f");
lcd_init();
lcd_putc(" Main Menu\n");           //Prepara el lcd per tonrar al menu principal
delay_ms(1000);
lcd_putc("\f");
lcd_putc(" choose sensor\n");
delay_ms(1000);
lcd_putc("\f");
lcd_putc(" 1-T°C 2-L 3-T\n");           //-----
lcd_gotoxy(1, 2);
lcd_putc(" 4-D 5-S 6-B");
break;

case 4:               // BARRIDO-----

    i2c_start();           // Start I2C protocol
    i2c_write(0xD0);           // DS1307 address
    i2c_write(0);           // Send register address (time seconds register)
```

```
i2c_start();           // Restart I2C
i2c_write(0xD1);       // Initialize data read
second = i2c_read(1);  // Read seconds from register 0
minute = i2c_read(1); // Read minutes from register 1
hour = i2c_read(1);    // Read hours from register 2
i2c_read(1);          // Read day from register 3 (not used)
date = i2c_read(1);   // Read date from register 4
month = i2c_read(1);  // Read month from register 5
year = i2c_read(0);   // Read year from register 6
i2c_stop();           // Stop I2C protocol
//DS1307_display();   // Display time & calendar
delay_ms(50);         // Wait 50ms

second = (second >> 4) * 10 + (second & 0x0F);
minute = (minute >> 4) * 10 + (minute & 0x0F);
hour = (hour >> 4) * 10 + (hour & 0x0F);
date = (date >> 4) * 10 + (date & 0x0F);
month = (month >> 4) * 10 + (month & 0x0F);
year = (year >> 4) * 10 + (year & 0x0F);
//! // End conversion
time[12] = second % 10 + 48;
time[11] = second / 10 + 48;
time[9] = minute % 10 + 48;
time[8] = minute / 10 + 48;
time[6] = hour % 10 + 48;
time[5] = hour / 10 + 48;
calendar[14] = year % 10 + 48;
calendar[13] = year / 10 + 48;
calendar[9] = month % 10 + 48;
```



```
calendar[8] = month / 10 + 48;  
calendar[6] = date % 10 + 48;  
calendar[5] = date / 10 + 48;
```

```
// MESURA DE VELOCITAT
```

```
while (input(pin_C1)==1);
```

```
a++;
```

```
switch (a){ // a incrementa una unitat cada cop que passa liman i el primer cop que passa  
registre e tempd dinici
```

```
case 0:
```

```
    a++;
```

```
    break;
```

```
case 1:
```

```
    m=minute;
```

```
    s=second;
```

```
    h=hour;
```

```
    break;
```

```
}
```

```
if (a==1000){ //es cobreix el kilimetre, es resta el temps actual amb el del registre dinici
```

```
    mm=minute-m;
```

```
    hh=hour-h;
```

```
ss=second-s;

if((hh>0)&&(m>minute)){ //es coregeixen els minuts si hi han canvi dhora
    hh=hh-1;
    mm=(60-m)+minute;
}
if((mm>0)&&(s>second)){ //es coregeixen els seg si hi han canvi de minut
    mm=mm-1;
    ss=(60-s)+second;
}

temps_0=(hh*3600)+(mm*60)+ss;
velocitat=1000/temps_0;
a=0;

}

while (input(pin_C1)==0);

Bigdata=velocitat;
save(select,Bigdata);
break;
}
}

void metros (selct) {

long cont;
int valor;
```

```
SET_TRIS_C(0x7F);
output_low(PIN_B2);
valor=0;
cont=0;

while (input(PIN_B7)==1){

while (input(pin_C1)==1);
    cont++;

    while (input(pin_C1)==0);

if (cont==1000){
    cont=0;
    valor=valor+1;
}

    lcd_putc("\f");
lcd_gotoxy(7,2); // posa cursor aquesta posicio
printf(lcd_putc,"%lu",cont);
lcd_gotoxy(10,2); // posa cursor aquesta posicio
lcd_putc(" metros");
    lcd_gotoxy(7,1); // posa cursor aquesta posicio
printf(lcd_putc,"%i",valor);
lcd_gotoxy(10,1); // posa cursor aquesta posicio
lcd_putc(" Km");
delay_ms (50);

}
```

```
    lcd_putc("\f");
    lcd_putc(" 1-T°C 2-L 3-T\n");          //-----
    lcd_gotoxy(1, 2);
    lcd_putc(" 4-D 5-S 6-B");

}
//-----5-Bateria-----

void Bateria(select){

//int16 valor;
float bat,valor,value;
int BB,Bigdata;
char H,a;
char BBB[] = " ";
setup_adc (adc_clock_internal);
delay_ms (100);

output_low(PIN_B2);

switch (select){

    case 0:

        lcd_putc("\f");
        lcd_putc("Bateria");

        while (input(PIN_B7)==1){
```

```
//!setup_adc_ports ();      //tots el sporta analogics
// setup_adc_ports (3);
set_adc_channel (3);      // choses channel A0
delay_us(20);
valor=read_adc ();      // conversio adc
// value=valor*0,357;

if (valor < 5)//658
    bat=0;
if (valor > 950)//1020
    bat=100;
if ((valor > 0)&&(valor < 1023))//658
    bat=(valor)/10,23;//658

if (valor < 200){
    lcd_putc("\f");
    lcd_gotoxy(1,1);
    lcd_putc(" LOW BATTERY");
    lcd_gotoxy(8,2);
    lcd_putc(" %");
    lcd_gotoxy(3,2);
    printf(lcd_putc,"%u",BB);}
else{
    lcd_putc("\f");
    lcd_gotoxy(1,1);
    lcd_putc("BATERIA");
    lcd_gotoxy(8,2);
    lcd_putc(" %");
    lcd_gotoxy(3,2);
```

```
printf(lcd_putc,"%u",BB);
}
BB=bat;

lcd_gotoxy(8,2);
lcd_putc(" %");
lcd_gotoxy(3,2);
printf(lcd_putc,"%u",BB);

//*****Visualitzacio en el
movi|*****

//output_low(PIN_c6);
//output_low(PIN_c7);

//!while (TRUE){
//if(kbhit()){
//!putc(5);
//!}

//enable_interrupts(global);
//enable_interrupts(INT_RDA);
//!}

//printf("AT+BAUD4");
//printf("AT+PN");
//a=getc();
//USART_Init(9600);

//TRIS_C6=0;
//TRISC7=1;
```

```
//TXSTA=0x20;
//RCSTA=0x90;

H=bat;
//while(TXIF==0);

BB = ((BB / 10) << 4) + (BB % 10);
time[12] = second % 10 + 48;
time[11] = second / 10 + 48;
//printf("%s",H);
//printf("hola mundo\n\r");

printf("%u",BB);
delay_ms(1000);
//printf(" \n\r");
//delay_ms(1000);
//printf(bat);

//putc(bat);

}
break;

case 5:
    set_adc_channel (3);          // choses channel A0
    delay_us(20);
```

```
valor=read_adc ();           // conversio adc
// value=valor*0,357;

if (valor < 658)//658
    bat=0;
if (valor > 1020)//1020
    bat=100;
if ((valor > 658)&&(valor < 1020))//658
    bat=(valor-658)/3,66;//658

BB=bat;
Bigdata=BB;
save(select,Bigdata);
break;
}
}
//-----6-BARRIDO-----
```

```
void Barrido(select){

int data,read,byta,bytb,h,s,ss,sec,vel;
output_low(PIN_B2);

h=0;
byta=0x00;
bytb=0x01;

while (input(PIN_B7)==1){
```



```
select++;

if (bytb==0xFF){
byta=byta+1;
bytb=0x00;
}
bytb++;

switch (select){

case 1:
    distancia();

break;

case 2://temperatura

    data=medir_barrido();
    lcd_putc("\f");
    lcd_gotoxy(5,1); // posa cursor aquesta posicio
    printf(lcd_putc,"Temperatura");

break;

case 3://llum

    data=Luz_barrido();
    lcd_putc("\f");
```

```
    lcd_gotoxy(5,1); // posa cursor aquesta posicio
printf(lcd_putc,"Luz");

break;

case 4://temps

    data=Temps_barrido();
    // temps(3);
    if(h==0)
s=data;

h++;

if(h==10) //380 metres perque entra cada a =38
//es coregeixen els seg si hi han canvi de minut
    ss=data;

    lcd_putc("\f");
    lcd_gotoxy(5,1); // posa cursor aquesta posicio
printf(lcd_putc,"Temps");
    lcd_gotoxy(13,2);
printf(lcd_putc,"%i",h);

break;

case 5://velocitat

data=distancia_barrido());
```

```
lcd_putc("\f");
    lcd_gotoxy(5,1); // posa cursor aquesta posicio
    printf(lcd_putc,"Distancia");
    lcd_gotoxy(13,2);
    printf(lcd_putc,"%i",h);

    break;

case 6://bateria

    data=Bateria_barrido();
    lcd_putc("\f");
    lcd_gotoxy(5,1); // posa cursor aquesta posicio
    printf(lcd_putc,"Bateria");
    select=0;
    break;
}

I2C_Start();
I2C_write(0xA2);
I2C_write(byta);
I2C_write(bytb);
I2C_write(data);
I2C_Stop();

delay_ms(200);

I2C_Start();
```

```
I2C_write(0xA2);
I2C_write(byta); //direcció me significativa
I2C_write(bytb); //direcció menys significativa
//I2C_write(data);
I2C_Start();
I2C_write(0xA3); //últim bit es 1 per llegir
read=I2C_read();
I2C_Stop();

//lcd_putc("\f");
lcd_gotoxy(5,2);
printf(lcd_putc,"%i",bytb);
lcd_gotoxy(10,2);
printf(lcd_putc,"%i",read);
delay_ms(100);

}
}
//-----OPTIONS-----
void options(code){

int select;
select=0;

switch (code){

case 1:
```

```
    medir(select);  
    code=0;  
  
break;  
  
case 2:  
  
    lcd_init();           // Inicia LCD  
    lcd_putc(" LUZ\n");  
    luz(select);  
    delay_ms(1000);  
  
    code=0;  
break;  
  
case 3:  
  
    Temps(select);  
    code=0;  
  
break;  
  
case 4:  
  
    metros(select);  
    code=0;  
break;  
  
case 5:
```

```
Bateria(select);  
code=0;  
  
break;  
  
case 6:  
  
Barrido(select);  
code=0;  
  
break;  
  
case 7:  
  
  lcd_init();           // Inicia LCD  
  lcd_putc("\f");      // Clean  
  lcd_putc(" BACK BOTTON\n");  
  delay_ms(500);  
  lcd_init();          // -----  
  lcd_putc("\f");  
  lcd_init();  
  lcd_putc(" Main Menu\n");      //Prepara el lcd per tonrar al menu principal  
  delay_ms(500);  
  lcd_putc("\f");  
  lcd_putc(" choose sensor\n");  
  delay_ms(500);  
  lcd_putc("\f");  
  lcd_putc(" 1-T°C 2-L 3-T\n");    //-----
```

```
lcd_gotoxy(1, 2);
lcd_putc(" 4-D 5-S 6-B");
    code=0;

break;

}
}

void main (void)
{
    int8 code;

    lcd_init();
    lcd_putc("\f");
    lcd_init();           // Inicia LCD
    lcd_putc(" Main Menu\n");
    delay_ms(1000);
    lcd_putc("\f");

    port_B_PULLUPS(TRUE);
    SET_TRIS_B(0x07);    //1F 000 trs primeres ports de sortida, 07 0000 0111
    SET_TRIS_C(0xFF);   // tots son 1 configurats com entrada

    output_high(PIN_B2);
    output_high(PIN_B3);
    output_high(PIN_B4);
```

```
output_high(PIN_B5);
output_high(PIN_B6);
output_high(PIN_B7);
output_low(PIN_C5);

lcd_init();
lcd_putc(" choose sensor\n");
delay_ms(1000);
lcd_putc("\f");
lcd_putc(" 1-T°C 2-L 3-T\n");          //-----
lcd_gotoxy(1, 2);
lcd_putc(" 4-D 5-S 6-B");

code=0;

while(TRUE){

code=0;
output_low(PIN_B2);

if(input(PIN_B5)==0)
    code=1;
if(input(PIN_B6)==0)
    code=4;
if(input(PIN_B7)==0)
    code=7;
```



```
if(input(PIN_C5)==1)
  code=10;
delay_ms(50);
output_high( PIN_B2);
output_low( PIN_B3);
```

```
if(input(PIN_B5)==0)
  code=2;
if(input(PIN_B6)==0)
  code=5;
if(input(PIN_B7)==0)
  code=8;
if(input(PIN_C5)==1)
  code=11;
```

```
delay_ms(50);
output_high( PIN_B3);
output_low( PIN_B4);
```

```
if(input(PIN_B5)==0)
  code=3;
if(input(PIN_B6)==0)
  code=6;
if(input(PIN_B7)==0)
  code=9;
if(input(PIN_C5)==1)
  code=12;
```

```
delay_ms(50);  
output_high( PIN_B4);  
  
options(code);  
  
}  
}
```