

```

/*
 * DATALOGGER UNIVERSAL
 *
 *
 *
 */

#include <Wire.h>
#include<Adafruit_ADS1015.h>
Adafruit_ADS1115 ads; /* Use this for the 16-bit version */
#include <RTCLib.h>
RTC_DS1307 RTC;
#include<ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <SPI.h>
#include <SD.h>
#include<LiquidCrystal_I2C.h>
#define I2C_ADDR    0x20 // <-Dirección del esclavo A1/2/3 a masa del LCD
#define BACKLIGHT_PIN    3
#define En_pin  2
#define Rw_pin  1
#define Rs_pin  0
#define D4_pin  4
#define D5_pin  5
#define D6_pin  6
#define D7_pin  7
LiquidCrystal_I2C  lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,
D7_pin);

/***** WiFi Access Point *****/
const char* WLAN_SSID = "JAZZTEL-QMGDLN";
const char* WLAN_PASS = "3QEJCR4S";////
//const char WLAN_SSID = "JAZZTEL-QMGDLN";
//const char WLAN_PASS = "3QEJCR4S";
const char* WLAN_SSID_2 = "lucad";
const char* WLAN_PASS_2 = "qwertyuiop";

/***** Adafruit.io Setup *****/
#define AIO_SERVER    "io.adafruit.com"
#define AIO_SERVERPORT  1883                // use 8883 for SSL
#define AIO_USERNAME    "lucascastorina"
#define AIO_KEY          "c066edc9cc8b4a7f92c7e4c361fd5c20"

/***** Global State SIN CAMBIOS *****/

```

```

WiFiClient client;// Create an ESP8266 WiFiClient class to connect to the MQTT
server.

// Store the MQTT server, username, and password in flash memory.
// This is required for using the Adafruit MQTT library.
const char MQTT_SERVER[] PROGMEM = AIO_SERVER;
const char MQTT_USERNAME[] PROGMEM = AIO_USERNAME;
const char MQTT_PASSWORD[] PROGMEM = AIO_KEY;

// Setup the MQTT client class by passing in the WiFi client and MQTT server and
login details.
Adafruit_MQTT_Client mqtt(&client, MQTT_SERVER, AIO_SERVERPORT, MQTT_USERNAME,
MQTT_PASSWORD);

/***** CREACION DE
Feeds*****/
/*****/

//////////////////////////////////// Setup feeds for
PUBLISHING////////////////////////////////////
//////////////////////////////////// Recordar que MQTT paths for AIO follow the
form:<username>/feeds/<feedname>

/*****/

////////////////////////////////////ESP-->INTERNET
(PUBLISHING)////////////////////////////////////
/*****PUBLICACION
ANALOGICA*****/

//FEEDs de las analogicas de 10bits (del MUX)
const char sensor0_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor0";
Adafruit_MQTT_Publish sensor0 = Adafruit_MQTT_Publish(&mqtt, sensor0_FEED);
const char sensor1_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor1";
Adafruit_MQTT_Publish sensor1 = Adafruit_MQTT_Publish(&mqtt, sensor1_FEED);
const char sensor2_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor2";
Adafruit_MQTT_Publish sensor2 = Adafruit_MQTT_Publish(&mqtt, sensor2_FEED);
const char sensor3_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor3";
Adafruit_MQTT_Publish sensor3 = Adafruit_MQTT_Publish(&mqtt, sensor3_FEED);
const char sensor4_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor4";
Adafruit_MQTT_Publish sensor4 = Adafruit_MQTT_Publish(&mqtt, sensor4_FEED);
const char sensor5_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor5";

```

```

Adafruit_MQTT_Publish sensor5 = Adafruit_MQTT_Publish(&mqtt, sensor5_FEED);
const char sensor6_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor6";
Adafruit_MQTT_Publish sensor6 = Adafruit_MQTT_Publish(&mqtt, sensor6_FEED);
const char sensor7_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor7";
Adafruit_MQTT_Publish sensor7 = Adafruit_MQTT_Publish(&mqtt, sensor7_FEED);

//FEEDs de las analogicas de 16bits (del ads)
const char sensor00_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor00";
Adafruit_MQTT_Publish sensor00 = Adafruit_MQTT_Publish(&mqtt, sensor00_FEED);
const char sensor11_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor11";
Adafruit_MQTT_Publish sensor11 = Adafruit_MQTT_Publish(&mqtt, sensor11_FEED);
const char sensor22_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor22";
Adafruit_MQTT_Publish sensor22 = Adafruit_MQTT_Publish(&mqtt, sensor22_FEED);
const char sensor33_FEED[] PROGMEM = AIO_USERNAME "/feeds/sensor33";
Adafruit_MQTT_Publish sensor33 = Adafruit_MQTT_Publish(&mqtt, sensor33_FEED);

/*****PUBLICACION DIGITAL *****/
/*****ESP-->INTERNET (PUBLISHING)*****/
const char PUSH_FEED[] PROGMEM = AIO_USERNAME "/feeds/push"; //OJO igual nombre
que en adafruit.iofeed
Adafruit_MQTT_Publish push1 = Adafruit_MQTT_Publish(&mqtt, PUSH_FEED); //push a
secas sera variable
//de entrada al micro cuando se apreta

/*****
*****/
/////////////////////////INTERNET-->ESP
(SUBCRIBING)/////////////////////////
/*****
*****/

/////////////////////////DIGITALES
(INTERNET-->ESP)/////////////////////////
const char ONOFF_FEED[] PROGMEM = AIO_USERNAME "/feeds/onoff";
Adafruit_MQTT_Subscribe onoffbutton = Adafruit_MQTT_Subscribe(&mqtt, ONOFF_FEED);
//onoffbutton se usa para activar desactivar GRABACION en SD
const char Dout2_FEED[] PROGMEM = AIO_USERNAME "/feeds/lamp";
Adafruit_MQTT_Subscribe Dout2 = Adafruit_MQTT_Subscribe(&mqtt, Dout2_FEED);

/////////////////////////ANALOGICAS(INTERNET-->ESP)/////////////////////////
/////////
// Setup a feed called 'analogical' for subscribing to changes.
const char analogical_FEED[] PROGMEM = AIO_USERNAME "/feeds/analogical";
Adafruit_MQTT_Subscribe analogical = Adafruit_MQTT_Subscribe(&mqtt,
analogical_FEED);

```

```

void MQTT_connect(); // Function to connect and reconnect as necessary to the
MQTT server.

/////////////////////////////////////////INICIO
VARIABLES/////////////////////////////////////////
volatile int m=1;//volatile para que se copie el flanco de interrupcion sin mirar
registros
volatile int mant=0;
int tiempo = 1000;
int p[8];//Array de diferencias
int sensorValueANT[8]={0,0,0,0,0,0,0,0};//Array de lecturas anteriores
int sensorValue[8]={0,0,0,0,0,0,0,0};//Array de lecturas
int largoparam=0;
//String parametros;
char parametros[80];
int s=0 ;//para maquina de estados, cambia de estado con interrupcion de pulsador
bool GrabaStatus=1;//Si GrabaStatus=1 me graba
File myFile;
byte MCPportIN= 0;
int hora;
int minuto;
int segundo=0;
int any;
int mes;
int dia;
bool GRABon=1;
int ADS_in0, ADS_in1, ADS_in2, ADS_in3;
const uint8_t MCP1address = 0x21; //segun se configura A0 A1 A2
const uint8_t MCP2address = 0x22;
boolgp0,gp1,gp2,gp3,gp4,gp5,gp6,gp7;
int DIF16b[4];
int lectural6B[4];
int lectural6B_ANT[4];
int leerMUX[8];
int Thereshold_1 =50;// Variacion del filtro de 10bits en mV
int Thereshold_2=70; //valor del filtro de 16bits para marcar la diferencia en mV
//cada vez que varia mas que el Thereshold se envia la lectura
byte portMCP;
long debouncing_time = 200;long debouncing_time2 = 200; //Debouncing Time in
Milliseconds, tiempo ANTIRREBOTE
volatile unsigned long last_micros;volatile unsigned long last_micros2;
char c;
//*****variables de webserver:
unsigned long previousMillis = 0; // will store last time LED was updated
const long interval = 100; // interval at which to blink (milliseconds)
WiFiServer server(80);
int j;

```

```
int i;

////////////////////////////////////// VOID SETUP
//////////////////////////////////////
//////////////////////////////////////
//////////////////////////////////////
//////////////////////////////////////
/*****
*****/

void setup() {

static const uint8_t D1    = 5;//Asignacion de pines de ESP8266 DEV BOARD
static const uint8_t D2    = 4;
static const uint8_t D3    = 0;//PUSH BUTTON INTERRUPT
static const uint8_t D4    = 2;//PUSH BUTTON INTERRUPT
static const uint8_t D5    = 14;
static const uint8_t D6    = 12;
static const uint8_t D7    = 13;
static const uint8_t D8    = 15;//chip select del SPI


Serial.begin(115200);
lcd.begin (16,2); // <--- pantalla LCD 16x2
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE); // Enciende el backlight
lcd.setBacklight(HIGH);
RTC.begin(); // Inicia la comunicación con el RTC
Wire.begin(D2,D1); // Inicia el puerto I2C, SDA scl
    delay(100);

    //////////////////////////////////////////SETUP DEL ADC de 16bits
    //////////////////////////////////////////
    //
    //                                     ADS1115
    //                                     -----
ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V  1 bit =      0.1875mV
(default)
//con gain=2/3 se pierden bit desde 26683 al 32767 pero se cubren los 0-5 volts
//ads.setGain(GAIN_ONE);          // 1x gain   +/- 4.096V  1 bit =           0.
125mV
//ads.setGain(GAIN_TWO);          // 2x gain   +/- 2.048V  1 bit =           0.0625mV
//ads.setGain(GAIN_FOUR);         // 4x gain   +/- 1.024V  1 bit =           0.03125mV
//ads.setGain(GAIN_EIGHT);        // 8x gain   +/- 0.512V  1 bit =           0.015625mV
//ads.setGain(GAIN_SIXTEEN);      // 16x gain  +/- 0.256V  1 bit =           0.0078125mV
ads.begin();

/*****
*****/

/*****
```

```

***** /
/*****
***** /
/*****
***** /
/*****
***** /
//////////LECTURA SD PARAMETROS de para.
txt//////////

LecturaparametrosSD();/*Se guarda en parametros[X] y su largo es largoparam*/
configuraMCP1(MCP1address);//CONFIGURACION DEL MCP y ademas se mira el
interruptor switch que se guarda en gp6/
configuraMCP2(MCP2address);//CONFIGURACION DEL MCP/

//////////Asignacion de entradas
INTERRUPCIONES//////////
pinMode(D4, INPUT_PULLUP); // PUSH button de interrupcion
attachInterrupt(digitalPinToInterrupt(D4),debounceInterrupt,RISING);//subrutina
interrupcion iniciada por push D4
pinMode(D3, INPUT_PULLUP); // PUSH button de interrupcion
attachInterrupt(digitalPinToInterrupt(D3),debounceInterrupt2,RISING);

/*****
***** /
/*****WEBSERVERON
SETUP*****/
if (gp6==1){
  WiFi.begin(WLAN_SSID_2, WLAN_PASS_2);
  lcd.clear();
  lcd.setCursor(0,0); lcd.print(" MODO WEBSERVER");
  lcd.setCursor(0,1); lcd.print("  ACTIVADO");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi conectado");
  server.begin();
  Serial.print("La URL de conexion es: ");
  Serial.print("http://"); Serial.print(WiFi.localIP()); Serial.println("/");

```

```

lcd.clear();
lcd.setCursor(0,0); lcd.print("URL de conexion: ");
lcd.setCursor(0,1);lcd.print(WiFi.localIP());
delay(2000);

/*****
*****/

/*****BROOKER SERVER ON
SETUP*****/
}else{
  WiFi.begin(WLAN_SSID, WLAN_PASS);
  Serial.println(); Serial.print("Connecting to ");Serial.println(WLAN_SSID);
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("  MODO BROOKER");
  lcd.setCursor(0,1); lcd.print("    ACTIVADO");
  delay(2000);
  lcd.clear();lcd.setCursor (0,0);  lcd.print("CONECTANDO....");
  lcd.setCursor (0,1);  lcd.print(WLAN_SSID);
  delay(1000);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected"); Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  lcd.clear();
  lcd.setCursor (0,0);  lcd.print("    RED WIFI ");
  lcd.setCursor (0,1);  lcd.print("    CONECTADO");
  delay(1500);
  lcd.clear();
  lcd.setCursor (0,0);  lcd.print("IP address:");
  lcd.setCursor (0,1);  lcd.print(WiFi.localIP());
  delay(1500);

  /***** Setup MQTT subscription for feeds.
  *****/

  ////////////////////////////////////AGREGAR LAS SUSCRIPCIONES
  CREADAS/////////////////////////////////

  mqtt.subscribe(&onoffbutton);
  mqtt.subscribe(&analogical);
  mqtt.subscribe(&Dout2);

  IMPRIMElectura_RTC();

```

```

delay(1000);

} //fin IF del setup de webserver o brooker

portMCP=0B10000000; //Se activa la salida led de grabacion en setup porque
inicialmente-->bool GRABon=1; GrabaStatus=1
escribirMCP(portMCP);
lcd.clear();
lcd.setCursor(0,0); lcd.print("Grabacion de ");
lcd.setCursor(0,1); lcd.print("Datos ACTIVADA");
delay(1500);

} //ACABA VOIP SETUP

/*//////////////////////////////////// VOIP
LOOP////////////////////////////////////
////////////////////////////////////
////////////////////////////////////*/
/*****
*****/

void loop() {

if (gp6==1){////WEBSERVER ON en
LOOP*****

WiFiClient client = server.available();// Check if a client has connected
  if (!client) {
    return;
  }
  Serial.println("new client");// Espera que se envíe algun dato
  lcd.clear();
  lcd.setCursor (0,0); lcd.print("Refrescando");
  lcd.setCursor (0,1); lcd.print(" DATOS.....");

  while (!client.available()) {
    delay(1);
  }
  Serial.println("Client disconnected");
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
  }
  lecturaMULTIPLEXADA();
}

String webpage = "<head><meta http-equiv=\"\"refresh\"\" content=\"\"3\"\"></head>";
webpage += "<h2>Analog_IN 0 (10 bits) : </h1><h2>";

```

```

sensorValue[0]=leerMUX[0];
webpage += String(sensorValue[0]);
webpage += "</h1>";
client.println(webpage);

String webpage1 = "<head><meta http-equiv=\"\"refresh\"\" content=\"\"3\"\"></head>";
webpage1 += "<h2>Analog_IN 1 (10 bits) :</h1><h2>";
sensorValue[1]=leerMUX[1];
webpage1 += String(sensorValue[1]);
webpage1 += "</h1>";
client.println(webpage1);

String webpage2 = "<head><meta http-equiv=\"\"refresh\"\" content=\"\"3\"\"></head>";
webpage2 += "<h2>Analog_IN 2 (10 bits) :</h1><h2>";
sensorValue[2]=leerMUX[2];
webpage2 += String(sensorValue[2]);
webpage2 += "</h1>";
client.println(webpage2);

lectura_ADS();
sensorValue[3]=lectural6B[0];

String webpage3 = "<head><meta http-equiv=\"\"refresh\"\" content=\"\"3\"\"></head>";
webpage3 += "<h2>Analog_IN 8 (16 bits) :</h1><h2>";
webpage3 += String(sensorValue[3]);
webpage3 += "</h1>";
client.println(webpage3);

String webpage4 = "<head><meta http-equiv=\"\"refresh\"\" content=\"\"3\"\"></head>";
webpage4 += "<h2>Analog_IN 9 (16 bits) :</h1><h2>";
sensorValue[4]=lectural6B[1];
webpage4 += String(sensorValue[4]);
webpage4 += "</h1>";
client.println(webpage4);

String webpage5 = "<head><meta http-equiv=\"\"refresh\"\" content=\"\"3\"\"></head>";
webpage5 += "<h2>Analog_IN 10 (16 bits) :</h1><h2>";
sensorValue[5]=lectural6B[2];
webpage5 += String(sensorValue[5]);
webpage5 += "</h1>";
client.println(webpage5);

```

```

//*****
// Read the first line of the request
//HTTP/1.1 200 OK Content-Type: text/html

```

```

String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();
// Match the request
int value = LOW;

if (request.indexOf("/LED=ON") != -1) {
    bitWrite (portMCP,7,1);    //Se escribe el bit 7 del mcp con un
1
    escribirMCP(portMCP);
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("Grabacion de ");
    lcd.setCursor(0,1); lcd.print("Datos ACTIVADA");
    value = HIGH;
}

if (request.indexOf("/LED=OFF") != -1) {
    //Escribir gp7=0 del puerto del MCPout
    GrabaStatus=0;
    bitWrite (portMCP,7,0);
    escribirMCP(portMCP);
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("Grabacion de ");
    lcd.setCursor(0,1); lcd.print("Datos DESACTIVADA");
    value = LOW;
}

// Return the response
// client.println("HTTP/1.1 200 OK");
// client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.print("GRABACION DE DATOS: ");
    if(value == HIGH) {
        client.print("On");
    } else {
        client.print("Off");
    }
client.println("<br><br>");
client.println("<a href=\" /LED=ON\"><button>Turn On </button></a>");
client.println("<a href=\" /LED=OFF\"><button>Turn off </button></a><br />");
client.println("</html>");
delay(1);
Serial.println("Client disonnected");
Serial.println("");
/*****

```

```

*****
*****//*****
*****/
/*****
*****/
/*****BROOKER SERVER ON
LOOP*****/
}else{
lectura_RTC();//Llamada a funcion de lectura RTC
MCPportIN = leerMCPport(MCPladdress);//MCPport es el estado del puerto del MCP,de
izquierda a derecha-->7..0
entrarRTCPulsador(MCPladdress);

MQTT_connect();//Llamada funcion de reconexion al servidor

int j=lectura_ADS(); //guarda en lectural6B[X] los cuatro valores leidos y
devuelve el indice que ha cambiado en j (99=SIN CAMBIOS)
switch (j){
    case 0:
        if (! sensor00.publish(lectural6B[j])) { //Se publica el valor y
devuelve true/false
            Serial.println(F("Failed"));
            lcd.clear(); lcd.setCursor (0,0);lcd.print("ERROR AL ENVIAR");
            lcd.setCursor (0,1);lcd.print("DATOS AL BROOKER");
        } else {
            Serial.print("Sending ANALOG_");Serial.print(j+8);Serial.
print(":");Serial.println(lectural6B[j]);
            lcd.clear();
            lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
            lcd.setCursor(0,1); lcd.print("AN_"); lcd.print(j+8);lcd.
print(":");lcd.print(lectural6B[j]);lcd.print("mV");
            if (GrabaStatus==1){ //grabara datos mientras no haya
pulsado LED D3=ON
                grabacion (lectural6B[j],(j+8)); //FUNCION GRABAR DATOS
            }
        }
        break;

    case 1:
        if (! sensor11.publish(lectural6B[j])) { //Se publica el valor y devuelve
true/false
            Serial.println(F("Failed"));
            lcd.clear(); lcd.setCursor (0,0);lcd.print("ERROR AL ENVIAR");
            lcd.setCursor (0,1);lcd.print("DATOS AL BROOKER");

```

```

        } else {
            Serial.print("Sending ANALOG_");Serial.print(j+8);Serial.
print(":");Serial.println(lectural6B[j]);
            lcd.clear();
            lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
            lcd.setCursor(0,1); lcd.print("AN_"); lcd.print(j+8);lcd.
print(":");lcd.print(lectural6B[j]);lcd.print("mV");
            if (GrabaStatus==1){ //grabara datos mientras no haya
pulsado LED D3=ON
                grabacion (lectural6B[j],(j+8)); //FUNCION GRABAR DATOS
                }
        }

break;
case 2:
    if (! sensor22.publish(lectural6B[j])) {//Se publica el valor y devuelve
true/false
        Serial.println(F("Failed"));
        lcd.clear(); lcd.setCursor (0,0);lcd.print("ERROR AL ENVIAR");
lcd.setCursor (0,1);lcd.print("DATOS AL BROOKER");
    } else {
        Serial.print("Sending ANALOG_");Serial.print(j+8);Serial.
print(":");Serial.println(lectural6B[j]);
        lcd.clear();
        lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
        lcd.setCursor(0,1); lcd.print("AN_"); lcd.print(j+8);lcd.
print(":");lcd.print(lectural6B[j]);lcd.print("mV");
        if (GrabaStatus==1){ //grabara datos mientras no haya
pulsado LED D3=ON
            grabacion (lectural6B[j],(j+8)); //FUNCION GRABAR DATOS
            }
    }

break;
// case 3:
// if (! sensor33.publish(lectural6B[j])) {//Se publica el valor y devuelve
true/false
//
// Serial.println(F("Failed"));
// lcd.clear(); lcd.setCursor (0,0);lcd.print("ERROR AL ENVIAR");
lcd.setCursor (0,1);lcd.print("DATOS AL BROOKER");
//
// } else {
//
// Serial.print("Sending ANALOG 33IN_");Serial.print(j+8);
Serial.print(":");Serial.println(lectural6B[j]);
//
// lcd.clear();
// lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
// lcd.setCursor(0,1); lcd.print("AN_"); lcd.print(j+8);lcd.
print(":");lcd.print(lectural6B[j]);lcd.print("mV");
//
// if (GrabaStatus==1){ //grabara datos mientras no haya

```

```

pulsado LED D3=ON
//                                grabacion (lectura16B[j],(j+8)); //FUNCION GRABAR
DATOS
//                                }
//                                }

    }//fin switch case

Adafruit_MQTT_Subscribe *subscription;//LECTURA Y ENVIO DE
TIEMPOS////////////////////////////////////

while ((subscription = mqtt.readSubscription(tiempo))){//variable tiempo
iniciada a 1 seg
////////////////////////////////////SUSCRIPCIONDIGITALES (INTERNET-->ESP)
////////////////////////////////////

    if (subscription == &onoffbutton){//Se LEE variable del broker y se
desac/activa la GRABACION
        Serial.print(F("GRABACION: "));
        Serial.println((char *)onoffbutton.lastread);
        char *value = (char *)onoffbutton.lastread;
        int current = atoi(value);
        if (current == 1 ){
            //Escribir gp7=1 del puerto del MCPout para indicar LED que se graba
o no

            GrabaStatus=1;
            bitWrite (portMCP,7,1);    //Se escribe el bit 7 del mcp con un
1

            escribirMCP(portMCP);
            lcd.clear();
            lcd.setCursor(0,0); lcd.print("Grabacion de ");
            lcd.setCursor(0,1); lcd.print("Datos ACTIVADA");
        }else{
            //Escribir gp7=0 del puerto del MCPout
            GrabaStatus=0;
            bitWrite (portMCP,7,0);
            escribirMCP(portMCP);
            lcd.clear();
            lcd.setCursor(0,0); lcd.print("Grabacion de ");
            lcd.setCursor(0,1); lcd.print("Datos DESACTIVADA");
        }
    }

    if (subscription == &Dout2){//Se LEE variable del broker y se desac/activa la

```

GRABACION

```
Serial.print(F("GRABACION: "));
Serial.println((char *)Dout2.lastread);
char *value = (char *)Dout2.lastread;
int current = atoi(value);
if (current == 1 ){
    bitWrite (portMCP,6,1); //Se escribe el bit 6 del mcp con un 1
    Serial.println("Se escribio 1 en portMCP");
    escribirMCP(portMCP);
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("Salida digital");
    lcd.setCursor(0,1); lcd.print("Dout_1 ACTIVADA");
} else{
    bitWrite (portMCP,6,0); //Se escribe el bit 6 del mcp con un
0
    escribirMCP(portMCP);
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("Salida digital");
    lcd.setCursor(0,1); lcd.print("Dout_1 DESACTIVADA");
}
}
```

////////////////////SUSCRIPCION ANALOGICA

(INTERNET-->ESP)////////////////////

```
if (subscription == &analogical) { //
    Serial.print(F("analogical: "));
    Serial.println((char *)analogical.lastread);
    char *valueT = (char *)analogical.lastread;
    tiempo = atoi(valueT);
    Serial.println(tiempo);
    lcd.clear(); lcd.setCursor (0,0); lcd.print("Tiempo de ciclo :"); lcd.setCursor
(0,1); lcd.print("deenvio:"); lcd.print(tiempo); lcd.print("mS");
}
```

} //FIN WHILE de readSubscription

////////////////////LECTURA A0

MULTIPLEXADA////////////////////

lecturaMULTIPLEXADA(); //Guarda las lecturas en leerMUX{}

```

////////////////////////////////////ESP8266-->INTERNET////////////////////////////////////
////////////////////////////////////

/*****
*****/

////////////////////////////////////ANALOGICA(ESP8266-->INTERNET)
////////////////////////////////////
////BUCLE ENVIO DE DATOS 10b y grabacion de analogicas
for(int i=0; i <= 3; i++){
    sensorValue[i]=leerMUX[i];
    p[i]=sensorValueANT[i]-sensorValue[i];
    p[i]=abs(p[i]);
    Serial.print("DIFERENCIA");Serial.print(i);Serial.print(":");Serial.
println(p[i]);
    if (p[i]>Thereshold_1){ //Thereshold_1=30 es Valor que mientras la diferencia
de lectura actual y anterior no sea mas grande,
        //no envia datos, osea ENVIARA si la diferencia es MAYOR a algo para
filtrar error de lectura
        switch (i){
            case 0:
                if (! sensor0.publish(sensorValue[0])) { //Se publica el valor y
devuelve true/false
                    Serial.println(F("Failed"));
                    lcd.clear(); lcd.setCursor (0,0);lcd.print("ERROR AL ENVIAR");
lcd.setCursor (0,1);lcd.print("DATOS A BROOKER");
                } else {
                    Serial.print(F("\nSending ANALOG IN "));Serial.print(i);Serial.
print(":");Serial.println(sensorValue[i]);
                    lcd.clear();
                    lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
                    lcd.setCursor(0,1); lcd.print("AN "); lcd.print(i);lcd.
print(":");lcd.print(sensorValue[i]);lcd.print("mV");
                    if (GrabaStatus==1){ //grabara datos mientras no haya pulsado
LED D3=ON
                        grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS
                    }
                    sensorValueANT[i]=sensorValue[i];
                    goto sigue;    //Para que salga del bicle for directo
                }

            case 1:
                if (! sensor1.publish(sensorValue[1])) { //Se publica el valor y
devuelve true/false
                    Serial.println(F("Failed"));
                } else {
                    lcd.clear();

```

```

        lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
        lcd.setCursor(0,1); lcd.print("AN"); lcd.print(i);lcd.
print(":");lcd.print(sensorValue[i]);lcd.print("mV");
        Serial.print(F("\nSending ANALOG IN ")); Serial.print(i);
Serial.print(":");Serial.println(sensorValue[i]);
        if (GrabaStatus==1){ //grabara datos mientras no haya pulsado
LED D3=ON

                grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS
                }
        sensorValueANT[i]=sensorValue[i];
        goto sigue;
    }

    case 2:
        if (! sensor2.publish(sensorValue[2])) {//Se publica el valor y devuelve
true/false

                Serial.println(F("Failed"));
                } else {

                lcd.clear();
                lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
                lcd.setCursor(0,1); lcd.print("AN"); lcd.print(i);lcd.
print(":");lcd.print(sensorValue[i]);lcd.print("mV");
                Serial.print(F("\nSending ANALOG IN "));Serial.print(i);
Serial.print(":");Serial.println(sensorValue[i]);
                if (GrabaStatus==1){ //grabara datos mientras no haya
pulsado LED D3=ON

                        grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS
                        }
                sensorValueANT[i]=sensorValue[i];
                goto sigue;
        }

//    case 3:
//        if (! sensor3.publish(sensorValue[3])) {//Se publica el valor y devuelve
true/false
//
//                Serial.println(F("Failed"));
//                } else {
//
//                lcd.clear();
//                lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
//                lcd.setCursor(0,1); lcd.print("AN"); lcd.print(i);lcd.
print(":");lcd.print(sensorValue[i]);lcd.print("mV");
//                Serial.print(F("\nSending ANALOG IN ")); Serial.print(i);
Serial.print(":");Serial.println(sensorValue[i]);
//
//

```

```

//
//
//          if (GrabaStatus==1){ //grabara datos mientras no haya
//
//          pulsado LED D3=ON
//
//          grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS
//
//          }
//
//          sensorValueANT[i]=sensorValue[i];
//
//          goto sigue;
//
//      }
//
//
//
//
//      case 4:
//
//      if (! sensor4.publish(sensorValue[4])) { //Se publica el valor y
//
//      devuelve true/false
//
//          Serial.println(F("Failed"));
//
//          } else {
//
//
//
//          lcd.clear();
//
//          lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
//
//          lcd.setCursor(0,1); lcd.print("AN"); lcd.print(i);lcd.
//
//          print(":");lcd.print(sensorValue[i]);lcd.print("mV");
//
//          Serial.print(F("\nSending ANALOG IN "));
//
//          Serial.print(i);
//
//          Serial.print(":");
//
//          Serial.println(sensorValue[i]);
//
//          if (GrabaStatus==1){ //grabara datos mientras no haya
//
//          pulsado LED D3=ON
//
//          grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS
//
//          }
//
//          sensorValueANT[i]=sensorValue[i];
//
//          goto sigue;
//
//      }
//
//
//      case 5:
//
//      if (! sensor5.publish(sensorValue[5])) { //Se publica el valor y devuelve
//
//      true/false
//
//          Serial.println(F("Failed"));
//
//          } else {
//
//
//
//          lcd.clear();
//
//          lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
//
//          lcd.setCursor(0,1); lcd.print("AN"); lcd.print(i);lcd.
//
//          print(":");lcd.print(sensorValue[i]);lcd.print("mV");
//
//          Serial.print(F("\nSending ANALOG IN "));
//
//          Serial.print(i);
//
//          Serial.print(":");
//
//          Serial.println(sensorValue[i]);
//
//          if (GrabaStatus==1){ //grabara datos mientras no haya
//
//          pulsado LED D3=ON
//
//          grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS

```

```
//
//                                     }
//
//         sensorValueANT[i]=sensorValue[i];
//
//         goto sigue;
//
//     }
//
//
//
//
//     case 6:
//         if (! sensor6.publish(sensorValue[6])) { //Se publica el valor y devuelve
true/false
//
//             Serial.println(F("Failed"));
//
//             } else {
//
//
//
//                 lcd.clear();
//                 lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
//                 lcd.setCursor(0,1); lcd.print("AN"); lcd.print(i);lcd.
print(":");lcd.print(sensorValue[i]);lcd.print("mV");
//                 Serial.print(F("\nSending ANALOG IN "));
//                 Serial.print(i);
//                 Serial.print(":");
//                 Serial.println(sensorValue[i]);
//                 if (GrabaStatus==1){ //grabara datos mientras no haya
pulsado LED D3=ON
//                 grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS
//                 }
//                 sensorValueANT[i]=sensorValue[i];
//                 goto sigue;
//
//             }
//
//
//
//
//     case 7:
//         if (! sensor7.publish(sensorValue[7])) { //Se publica el valor y devuelve
true/false
//
//             Serial.println(F("Failed"));
//
//             } else {
//
//
//
//                 lcd.clear();
//                 lcd.setCursor(0,0); lcd.print("Enviando ANALOG");
//                 lcd.setCursor(0,1); lcd.print("AN"); lcd.print(i);lcd.
print(":");lcd.print(sensorValue[i]);lcd.print("mV");
//                 Serial.print(F("\nSending ANALOG IN "));
//                 Serial.print(i);
//                 Serial.print(":");
//                 Serial.println(sensorValue[i]);
//                 if (GrabaStatus==1){ //grabara datos mientras no haya
pulsado LED D3=ON
//                 grabacion (sensorValue[i],i); //FUNCION GRABAR DATOS
//                 }
//                 sensorValueANT[i]=sensorValue[i];
//                 goto sigue;
```

```

//      }

    }//FIN CASE

} //FIN PRIMER IF

} //FIN BUCLE FOR ENVIO DATOS y grabacion de analogicas
sigue:

//////////////////////////////////////DIGITAL
(ESP8266-->INTERNET)////////////////////////////////////
if (m!=mant){ //if para activar indicador en web cuando se pulsa en D3
  /*Interrupcion de RESERVA en pin D3*/
  if (! push1.publish(m)) { //Se publica el valor y devuelve true/false
    Serial.println(F("Failed to send push1"));
  } else {
    Serial.println(F("envio correcto de push1 yeaa!"));
  }
if (m==0){
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("  Dout_2");
  lcd.setCursor(0,1); lcd.print(" DESACTIVADA");
  delay(1000);
}
if (m==1){
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("  Dout_2");
  lcd.setCursor(0,1); lcd.print(" ACTIVADA");
  delay(1000);
}
}
mant=m;
}

} //FIN IF DEL switch brooker/webserver

Serial.println();
for (int i=0; i <= 3; i++){
  Serial.print("An10B_");
  Serial.print(i);
  Serial.print(" :");
  Serial.println(leerMUX[i]);
}
} //FIN VOID LOOP GRAL

```

```

/*////////////////////////////////////////////////////////////////////////////////////////////////////////////////*/
//////////////////////////////////////
/*****FUNCIONES*****/
*****
//////////////////////////////////////
//////////////////////////////////////
//////////////////////////////////////
//////////////////////////////////////*/

/*////////////////////////////////////////funcion de configuracion del MCP como
ENTRADA////////////////////////////////**/
void configuraMCP1 (const uint8_t MCP1address){
Wire.beginTransmission(MCP1address); //begins talking to the slave device
Wire.write(0x00); //selects the IODIRA register
Wire.write(0B11111111); //1=inputs
Wire.endTransmission(); //stops talking to device

Wire.beginTransmission(MCP1address); //begins talking to the slave device
Wire.write(0x01); //Registro IPOL inversion de entrada
Wire.write(0B00000000); //Si tengo 0=activo al quitar GND si tengo 1=activo si
colocho GND
Wire.endTransmission(); //stops talking to device

Wire.beginTransmission(MCP1address); //begins talking to the slave device
Wire.write(0x06); //Registro GPPU activa Rs pullup entrada
Wire.write(0B11111111); //Todas entradas con Pullup
Wire.endTransmission(); //stops talking to device

//Se mira el SWITCH del para decidir si es webserver o brooker, se tiene que
hacer dos veces porque la primera no la coge
Wire.beginTransmission(MCP1address); //starts talking to slave device
Wire.write(0x09); //selects the GPIO pins
Wire.requestFrom(MCP1address,1);//Se lee un BYTE del MCP1
MCPportIN = Wire.read(); //MCPport me guarda el puerto del MCP
Wire.endTransmission();
gp6=bitRead(MCPportIN,6);

Wire.beginTransmission(MCP1address); //starts talking to slave device
Wire.write(0x09); //selects the GPIO pins
Wire.requestFrom(MCP1address,1);//Se lee un BYTE del MCP1
MCPportIN = Wire.read(); //MCPport me guarda el puerto del MCP
Wire.endTransmission();
Serial.println("TODO IN EN SETUP_2: ");Serial.println(MCPportIN,BIN);
gp6=bitRead(MCPportIN,6);

} //fin funcion configuraMCP1

```

```

/*////////////////////////////////////////funcion de
configuracion del MCP como SALIDA////////////////////////////////*/
void configuraMCP2 (const uint8_t MCP2address){
Wire.beginTransmission(MCP2address); //begins talking to the slave device
Wire.write(0x00); //selects the IODIRA register
Wire.write(0B00000000); //0=outputs
Wire.endTransmission(); //stops talking to device
}

////////////////////////////////////////Funcion de escritura
de puerto MCP
void escribirMCP (byte port){
Wire.beginTransmission(MCP2address); //starts talking to slave device
Wire.write(0x09); //selects the GPIO pins
Wire.write(port); // turns on pins 0 and 1 of GPIOA
Wire.endTransmission();
}

/*////////////////////////////////////////funcion de lectura
de entradas analogicas de mayor precision////////////////////////////////*/
int lectura_ADS(){
    int jj;
    for (int jj=0; jj <= 3; jj++){
        lectural6B[jj] = ads.readADC_SingleEnded(jj);
        lectural6B[jj]= (float)lectural6B[jj]/5.3306;//ESCALADO 5volts=50000 esta sin
la COMA
        DIF16b[jj]=lectural6B[jj]-lectural6B_ANT[jj];
        DIF16b[jj]=abs(DIF16b[jj]);
    }//fin for

for (int u=0; u <= 3; u++){
    if (DIF16b[u]>Thereshold_2){ //valor para marcar la diferencia, son
milivolts directos, si es mayor la dif se enviará
        lectural6B_ANT[u]=lectural6B[u];
        Serial.print("LA u dentro de la funcion ");      Serial.println(u);
        return u;
    }
}

return 99; //No hay cambios devuelve 99

} //fin funcion lectura ADS

```

```

/*////////////////////////////////////funcion de
lectura de entradas analogicas de 101 bit MULTIPLEXADAS////////////////////////////////*/
void lecturaMULTIPLEXADA() {
Serial.end();////PARA PODER USAR SALIDAS TX RX se deshabilita el puerto serie
delay(10);
static const uint8_t RX    = 3;    //MUX_B
static const uint8_t TX    = 1;    //MUX_A
static const uint8_t D0    = 16;   //MUX--C
pinMode(TX, OUTPUT); pinMode(D0, OUTPUT); pinMode(RX, OUTPUT);
for (int abc=0; abc <= 7; abc++){
    int r=10;
    if (abc ==0){
        digitalWrite(D0, 0);
        digitalWrite(RX, 0);
        digitalWrite(TX, 0);
        delay(r);
        leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
    }

    if (abc==1){
        digitalWrite(D0, 0);
        digitalWrite(RX, 0);
        digitalWrite(TX, 1);
        delay(r);
        leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
    }

    if (abc==2){
        digitalWrite(D0, 0);
        digitalWrite(RX, 1);
        digitalWrite(TX, 0);
        delay(r);
        leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
    }

    if (abc==3){
        digitalWrite(D0, 0);
        digitalWrite(RX, 1);
        digitalWrite(TX, 1);
        delay(r);
        leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
    }
}

```

```

if (abc==4) {
    digitalWrite(D0, 1);
    digitalWrite(RX, 0);
    digitalWrite(TX, 0);
    delay(r);
    leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
}

if (abc==5) {
    digitalWrite(D0, 1);
    digitalWrite(RX, 0);
    digitalWrite(TX, 1);
    delay(r);
    leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
}

if (abc==6) {
    digitalWrite(D0, 1);
    digitalWrite(RX, 1);
    digitalWrite(TX, 0);
    delay(r);
    leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
}

if (abc==7) {
    digitalWrite(D0, 1);
    digitalWrite(RX, 1);
    digitalWrite(TX, 1);
    delay(r);
    leerMUX[abc] = analogRead(A0)/0.322;//Lectura de entrada analogica
}
} //FIN FOR int abc=0; abc <= 7; abc++
Serial.begin(115200); //PARA PODER USAR SALIDAS TX RX
delay(200); //PROBAR TIEMPOS DE LECTURA diferentes
}

/*////////////////////////////////////funcion de
lectura de parametros guardados en la SD////////////////////////////////*/
void LecturaparametrosSD () {
    if (!SD.begin(D8)) { //Comienzo con chip select
        Serial.println( ); Serial.println("ERROR en inicio de SD ");
    }
}

```

```

    }else{
        Serial.println("Inicio de SD OK.");
        Serial.println("Lectura de parametros de inicio (para.txt)");
    }
myFile = SD.open("para.txt");
    if (myFile) {
        while (myFile.available()) {
            //      parametros+=myFile.read();
            parametros[largoparam]=myFile.read();
            largoparam=largoparam+1;
        }
        myFile.close();
        Serial.print("Parametros Inicio:");
        Serial.println(parametros);
        Serial.print("Control parametros:");
        Serial.println(largoparam);
    } else {
        // if the file didn't open, print an error:
        Serial.println("error opening para.txt");
    }
}

/*//////////////////////////////////////Segun
se pulsa se entran los parametros al RTC////////////////////////////////*/
void entrarRTCPulsador(const uint8_t MCP1address) {

    if (s==1){
        Serial.println("PULSE S2 P/INCREMENTAR HORA");
        delay(1500);

        while (bitRead (MCPportIN,7)==0 and s==1 ){    //and hora<26al poner a gnd el
gpio7 va incrementando
            Serial.print("DEJE DE PULSAR PARA GUARDAR HORA= ");
            MCPportIN = leerMCPport(MCP1address);
            if (bitRead (MCPportIN,7)==1 ){
                lcd.clear(); lcd.setCursor (0,0);lcd.print("Hora: ");lcd.print(hora,
DEC);lcd.print(" guardada");
                delay(1500);
                goto fuera0;//Si no esta el goto suma un al salir del while
            }
            hora=hora+1;
            if (hora<=24){

```

```

        lcd.clear(); lcd.setCursor (0,0);lcd.print("Deje pulsador p/");
        lcd.setCursor (0,1);lcd.print("guardar hs:");lcd.print(" ");lcd.print(hora,
DEC);

        Serial.println(hora,DEC);
        delay(700);//Tiempo en el que se va incrementando
    }else{
        hora=0;
    }
}

fuera0:
lcd.clear(); lcd.setCursor (0,0); lcd.print("PULSE S1 PARA");          lcd.setCursor
(0,1);lcd.print("SALIR de Hs");
delay(1600);

lcd.clear(); lcd.setCursor (0,0);lcd.print("PULSE S2 PARA ");          lcd.setCursor
(0,1);lcd.print("INCREMENTAR HORA");
delay(1600);

RTC.adjust(DateTime(any, mes, dia, hora, minuto, 11));
    Serial.print("hora introducida: ");
    Serial.println(hora,DEC);
    IMPRIMElectura_RTC();

}

if (s==2){
    Serial.println("PULSE PARA INCREMENTAR MINUTOS");
    while (bitRead (MCPportIN,7)==0 and s==2 ){
        Serial.print("DEJE DE PULSAR PARA GUARDAR MINUTOS= ");
        MCPportIN = leerMCPport(MCP1address);
        if (bitRead (MCPportIN,7)==1 ){
            lcd.clear(); lcd.setCursor (0,0);lcd.print("Minuto: ");lcd.print(minuto,
DEC);lcd.print(" guardado");
            delay(1500);
            goto fueral;//Si no esta el goto suma un al salir del while
        }
        minuto=minuto+1;
        if (minuto<61){
            lcd.clear(); lcd.setCursor (0,0);lcd.print("Suelte pulsador p/");
            lcd.setCursor (0,1);lcd.print("guardar min:");lcd.print(" ");lcd.
print(minuto,DEC);
            Serial.println(minuto,DEC);
            delay(450);
        }else {
            minuto=-1;
        }
    }
    fueral:
        lcd.clear(); lcd.setCursor (0,0); lcd.print(" PULSE S1 PARA");          lcd.
setCursor (0,1);lcd.print("SALIR de MIN");

```

```

delay(1600);
lcd.clear(); lcd.setCursor (0,0);lcd.print(" PULSE S2 PARA ");          lcd.
setCursor (0,1);lcd.print("INCREMENTAR MIN");
delay(1600);
    RTC.adjust(DateTime(any, mes, dia, hora, minuto, 0));
    Serial.print("MINUTOS introducidos: ");    Serial.println(minuto,DEC);
    IMPRIMElectura_RTC();
    //delay(2000);
}

if (s==3){
Serial.println("PULSE PARA INCREMENTAR DIA");
    while (bitRead (MCPportIN,7)==0 and s==3 ){
        Serial.print("DEJE DE PULSAR PARA GUARDAR DIA= ");
        MCPportIN = leerMCPport(MCPladdress);
        if (bitRead (MCPportIN,7)==1 ){
            lcd.clear(); lcd.setCursor (0,0);lcd.print("Dia: ");lcd.print(dia,DEC);lcd.
print(" guardado");
                delay(1500);
                goto fuera3;//Si no esta el goto suma un al salir del while
            }
            dia=dia+1;
            if (dia<32){
                lcd.clear(); lcd.setCursor (0,0);lcd.print("Suelte pulsador p/");
                lcd.setCursor (0,1);lcd.print("guardar dia:");lcd.print(" ");lcd.print(dia,
DEC);
                Serial.println(dia,DEC);
                delay(400);
            } else {
                dia=0;
            }
        }
        fuera3:
        lcd.clear(); lcd.setCursor (0,0); lcd.print("PULSE S1 PARA");          lcd.
setCursor (0,1);lcd.print("SALIR de DIA");
        delay(1600);
        lcd.clear(); lcd.setCursor (0,0);lcd.print("PULSE S2 PARA ");          lcd.setCursor
(0,1);lcd.print("INCREMENTAR DIA");
        delay(1600);
        RTC.adjust(DateTime(any, mes, dia, hora, minuto, 0));
        Serial.print("dia introducido: ");
        Serial.println(dia,DEC);
        IMPRIMElectura_RTC();
        //delay(2000);
    }

    if (s==4){
Serial.println("PULSE PARA INCREMENTAR MES");
        while (bitRead (MCPportIN,7)==0 and s==4 ){

```

```

        Serial.print("DEJE DE PULSAR PARA GUARDAR MES= ");
MCPportIN = leerMCPport(MCPladdress);
    if (bitRead (MCPportIN,7)==1 ){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Mes: ");lcd.print(mes,DEC);lcd.
print(" guardado");
        delay(1500);
        goto fuera4;//Si no esta el goto suma un al salir del while
    }
    mes=mes+1;
    if (mes<13){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Suelte pulsador p/");
        lcd.setCursor (0,1);lcd.print("guardar mes:");lcd.print(" ");lcd.print(mes,
DEC);
        Serial.println(mes,DEC);
        delay(400);
    }else{
        mes=0;
    }
}

fuera4:
    lcd.clear(); lcd.setCursor (0,0); lcd.print(" PULSE S1 PARA");          lcd.
setCursor (0,1);lcd.print("SALIR de MES");
delay(1600);
lcd.clear(); lcd.setCursor (0,0);lcd.print(" PULSE S2 PARA ");          lcd.
setCursor (0,1);lcd.print("INCREMENTAR MES");
delay(1600);
RTC.adjust(DateTime(any, mes, dia, hora, minuto, 11));
Serial.print("mes introducido: ");
Serial.println(mes,DEC);
    IMPRIMElectura_RTC();
    //delay(2000);
}

if (s==5){
Serial.println("PULSE PARA INCREMENTAR ANY");
    while (bitRead (MCPportIN,7)==0 and s==5 ){
        Serial.print("DEJE DE PULSAR PARA GUARDAR ANY= ");
MCPportIN = leerMCPport(MCPladdress);
    if (bitRead (MCPportIN,7)==1 ){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("ANO guardado: ");
        lcd.setCursor (0,1); lcd.print(any,DEC);
            delay(1500);
            goto fuera5;//Si no esta el goto suma un al salir del while
    }
    any=any+1;
    lcd.clear(); lcd.setCursor (0,0);lcd.print("Suelte pulsador p/");
        lcd.setCursor (0,1);lcd.print("guardar ANO:");lcd.print(any,DEC);
Serial.println(any,DEC);
    delay(400);
}

```

```

    if (any>2050) {
        any=2000;
    }
}

fuera5:
    lcd.clear(); lcd.setCursor (0,0); lcd.print(" PULSE S1 PARA");          lcd.
setCursor (0,1);lcd.print("SALIR de ANO");
delay(1600);//*****
lcd.clear(); lcd.setCursor (0,0);lcd.print(" PULSE S2 PARA ");          lcd.
setCursor (0,1);lcd.print("INCREMENTAR ANO");
delay(1600);//*****
    RTC.adjust(DateTime(any, mes, dia, hora, minuto, 11));
    Serial.print("any introducido: ");
    Serial.println(any,DEC);
    IMPRIMElectura_RTC();
    // delay(2000);
}

if (s==6){
    lcd.clear(); lcd.setCursor (0,0);lcd.print("Mantenga S2 para ");    lcd.
setCursor (0,1);lcd.print("NO GRABAR DATOS");
    //delay(1500);
    Serial.println("PULSE PARA DESACTIVAR GRABACION");
    while (bitRead (MCPportIN,7)==0 and s==6 ){
        MCPportIN = leerMCPport(MCP1address);
        if (bitRead (MCPportIN,7)==1 ){//si alguien ha dejado de pulsar me enseña
resultado
            if (GRABon==1){
                lcd.clear();
                lcd.setCursor(0,0); lcd.print("GRABACION DATOS");
                lcd.setCursor(0,1); lcd.print("ACTIVADA");
                Serial.println("GRABACION ACTIVADA");
                GrabaStatus=1;
                portMCP=0B10000000;
                escribirMCP(portMCP);
                delay(1000);
            }else{
                lcd.clear();
                lcd.setCursor(0,0); lcd.print("GRABACION DATOS");
                lcd.setCursor(0,1); lcd.print("DESACTIVA");
                Serial.println("GRABACION DESACTIVADA");
                GrabaStatus=0;
                portMCP=0B00000000;
                escribirMCP(portMCP);
                delay(1000);
            }
            goto fuera6;//Si no esta el goto cambia al salir del while
        }
        GRABon=!GRABon;

```

```

    lcd.clear();
    lcd.setCursor(0,0); lcd.print("GRABACION DATOS");
    if (GRABon==1){
        lcd.setCursor(0,1); lcd.print("ACTIVADA");
    }else{
        lcd.setCursor(0,1); lcd.print("DESACTIVADA");
    }
fuera6:
    delay(1000);

    }//fin while (MCPportIN,7)==0 and s==6

} //fin if (s==6)

} //FIN FUNCION entrarRTCPulsador

/*//////////////////////////////////////FUNCION
LECTURA PUERTO MCP por I2C////////////////////////////////////*/

byte leerMCPport (const uint8_t MCP1address){
    //static const uint8_t MCP1address = 0x21; //la direccion se
compone=0/1/0/0/A2/A1/A0
Wire.beginTransmission(MCP1address); //starts talking to slave device
Wire.write(0x09); //selects the GPIO pins
Wire.requestFrom(MCP1address,1); //Se lee un BYTE del MCP1
byte MCPportx = Wire.read(); //MCPport me guarda el puerto del MCP
Wire.endTransmission(); //ends communication with the device
return MCPportx;
}

```

```
/*//////////////////////////////////////FUNCIONES
deANTIRREBOTE////////////////////////////////*/
```

```
void debounceInterrupt() {
    if((long)(micros() - last_micros) >= debouncing_time * 1000) {
        interrup1();
        last_micros = micros();
    }
}
```

```
void debounceInterrupt2() {
    if((long)(micros() - last_micros2) >= debouncing_time2 * 1000) {
        interrup2();
        last_micros2 = micros();
    }
}
```

```
/*//////////////////////////////////////Funcion Rutina de
interrupcionpulsador1////////////////////////////////*/
```

```
void interrup1(){
    s=s+1;

    if (s== 1){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Pulse para ");   lcd.setCursor (0,
1);lcd.print("incrementar HORA");
    }
    if (s== 2){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Pulse para ");   lcd.setCursor (0,
1);lcd.print("incrementar MIN");
    }
    if (s== 3){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Pulse para ");   lcd.setCursor (0,
1);lcd.print("incrementar DIA");
    }
}
```

```

    }
    if (s== 4){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Pulse para ");   lcd.setCursor (0,
1);lcd.print("incrementar MES");
    }
    if (s== 5){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Pulse para ");   lcd.setCursor (0,
1);lcd.print("incrementar ANO");
    }
    if (s== 6){
        lcd.clear(); lcd.setCursor (0,0);lcd.print("Mantenga S2 para ");   lcd.
setCursor (0,1);lcd.print("NO GRABAR DATOS");

    }

```

```

Serial.print("s interrupt = "); Serial.println(s);

```

```

if (s>=7){//0-6 ESTADOS maximo
    s=0;
    lcd.clear(); lcd.setCursor (0,0);lcd.print(" MODO NORMAL DE");   lcd.setCursor
(0,1);lcd.print(" FUNCIONAMIENTO");
}

```

```

} //fin interrup

```

```

/*//////////////////////////////////////////Funcion Rutina de
interrupcionpulsador2////////////////////////////////////////*/

```

```

void interrup2() {
    m=!m;
    Serial.print("m interrupt = ");Serial.println(m);
}

```

```

} //fin interrup

```

```

/*////////////////////////////////////////// Funcion de
reconexionaMQTTserver////////////////////////////////////////*/

```

```

void MQTT_connect() {
    int8_t ret;

```

```

    // Stop if already connected.
    if (mqtt.connected()) {
        return;
    }

```

```

Serial.print("Connecting to MQTT... ");

```

```

uint8_t retries = 3;
while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000); // wait 5 seconds
    retries--;
    if (retries == 0) {
        // basically die and wait for WDT to reset me
        while (1);
    }
}
Serial.println("MQTT Connected!");
}

```

```

/*//////////////////////////////////////Funcion
lecturaRTC////////////////////////////////////*/

```

```

void lectura_RTC()
{
    DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC
    Serial.print(now.year(), DEC); // Año
    any=now.year();
    //Serial.print(any);////para aislar la variable
    Serial.print(' ');
    Serial.print('/');
    Serial.print(' ');
    Serial.print(now.month(), DEC); // Mes
    mes=now.month();
    Serial.print(' ');
    Serial.print('/');
    Serial.print(' ');
    Serial.print(now.day(), DEC); // Dia
    dia=now.day();
    Serial.print(' ');
    Serial.print(' ');
    Serial.print(' ');
    Serial.print(now.hour(), DEC); // Horas
    hora=now.hour();
    Serial.print(':');
    Serial.print(now.minute(), DEC); // Minutos
    minuto=now.minute();
    Serial.print(':');
    Serial.print(now.second(), DEC); // Segundos
    segundo=now.second();
}

```

```
/*//////////////////////////////////////Funcion
Impresion de Hora y fecha actual////////////////////////////////*/
```

```
void IMPRIMElectura_RTC()
{
    DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC
    lcd.clear();
    lcd.setCursor(0,0);lcd.print("Hora Fecha: ");lcd.print(now.year(), DEC);
    lcd.setCursor(0,1);lcd.print(now.month(), DEC);lcd.print('/');lcd.print(now.
day(), DEC);lcd.print(' ');lcd.print(' ');
    lcd.print(now.hour(), DEC); lcd.print(':');
    if (now.minute()<=9){
        lcd.print('0');
    }

    lcd.print(now.minute(), DEC);lcd.print(':');lcd.print(now.second(), DEC);
    lcd.print(' ');
}
```

```
/*//////////////////////////////////////ESCRIBIENDO DATOS en
SD////////////////////////////////*/
```

```
void grabacion (int DATO, int i)//DATO es el valor del sensor
{
    // File myFile;
    String nomarx ; //variable para darle nombre al archivo que se guarda en la
sd
    nomarx +=dia;
    //nomarx += " ";
    nomarx +=mes;//año/mes/dia
    //nomarx += " ";
    nomarx +=any;//se concatenan variables para crear nombre de archivo
    nomarx+=" .txt";
    myFile = SD.open(nomarx, FILE_WRITE);
    if (myFile) { //si se abre el archivo me escribe los datos
        myFile.print(hora);
        myFile.print(":");
        myFile.print(minuto);
        myFile.print(":");
        if(segundo<10){
            myFile.print("0");
        }
        myFile.print(segundo);
        myFile.print("\t");
    }
}
```

```
myFile.print(" Sens ");
myFile.print(i);
myFile.print(" ");
myFile.print(":");
myFile.print("\t");
myFile.println(DATO);
myFile.close();
Serial.println("...CAMBIOS GRABADOS OK");
    } else {
        // if the file didn't open, print an error:
        Serial.println("ERROR de GRABACION");
        Serial.println(nomarx);
    }
}
```