

```

/*
NodeMCU ESP8266 GPIO Funzione
D0  GPIO16    LED_BUILTIN or BUILTIN_LED
D1  GPIO5     I2C-SCL or CLK
D2  GPIO4     I2C-SDA
D3  GPIO0     SPI-RES or RST
D4  GPIO2     SPI-DC
D5  GPIO14    SPI-SCL or CLK
D6  GPIO12    SPI-MISO
D7  GPIO13    SPI-SDA or MOSI
D8  GPIO15    SPI-CS or SS
D9  GPIO3     Serial RX0
D10 GPIO1     Serial TX0
D11 GPIO16    Relai
D12 GPIO10    Sensore AM2302
*/
#include "SPI.h"
#include <SDFS.h>
#include <SdFat.h>
#include "SD.h"
#include <TimeLib.h>
#include <DS1307RTC.h>
#include "DHT.h"
#include <Wire.h>
#include <OneWire.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
#include <AM2320.h>
#include <Adafruit_INA219.h>
//*****//
const char* ssid = "mia_rete";
const char* password = "mia_password";
const char* ID = "SERRA_01";
const int chipSelect = D8; // used for ESP8266
ESP8266WebServer server(8081);
int ora, minuto, secondi, giorno, mese, anno;
DHT dht1;
AM2320 th;
Adafruit_INA219 ina219_A(0x40);
Adafruit_INA219 ina219_B(0x44);
int lumen; int hy; int hy2;
float celsius, fahrenheit;
unsigned long previousMillis = 0;

```

```

const long interval = 300000;
const int ventola = 16;
//*****
void setup(void) {
  delay(1000);
  pinMode(ventola, OUTPUT);
  Serial.begin(9600);
  IPAddress ip(192, 168, 1, 135);
  IPAddress gat(192, 168, 1, 1);
  IPAddress sub(255, 255, 255, 0);
  WiFi.config(ip, gat, sub);
  Serial.println();
  Wire.begin();
  Serial.println(" GESTIONE DATI SERRA - TRENTO - V3.9 ");
  Serial.println();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Serial.print(" Apparato :\t\t"); Serial.println(ID);
  Serial.print(" Connesso a\t\t");
  Serial.println(ssid);
  Serial.print(" Indirizzo IP :\t");
  Serial.println(WiFi.localIP());
  if (MDNS.begin("esp8266")) {
    Serial.println();
    Serial.println(" Inizializzato risponditore MDNS ");
  }
  server.on("/", handleRoot);
  server.on("/inline", []() {
    server.send(200, "text/plain", "Questo funziona bene ");
  });
  server.onNotFound(handleNotFound);
  server.begin();
  delay(500);
  Serial.println(" inizializzato server HTTP ");
  //*****
  delay(1000);
  if (!SD.begin(chipSelect)) {
    Serial.println(" >>> Inizializzazione SD fallita!");
    while (1);
  }
  else {
    Serial.println(" SD Inizializzata !");
  }
}

```

```

dht1.setup(10); // data pin
if (! ina219_A.begin()) {
  Serial.println(" >>> Scheada misure BATTERIA non trovata !!");
  delay(500);
}
if (! ina219_B.begin()) {
  Serial.println(" >>> Scheada misure PANNELLO non trovata !!");
  delay(500);
}
uint32_t currentFrequency;
ina219_A.begin();
ina219_B.begin();
}
//*****
void loop(void)
{
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    scriviSD();
    delay(500);
    Serial.println(" >>>Ho scritto su SD ");
    Serial.println();
  }
  Serial.println();
  Serial.print (" GESTIONE DATI SERRA - TRENTO - V3.9 ");
  Serial.println();
  delay(500);
  tmElements_t tm;
  if (RTC.read(tm)) {
    Serial.println();
    Serial.print(" Data =\t\t");
    giorno = (tm.Day);
    Serial.print(giorno);
    Serial.write('/');
    mese = (tm.Month);
    Serial.print(mese);
    Serial.write('/');
    anno = ((tm.Year) - 30);
    Serial.println(tm.YearToCalendar(tm.Year));
    Serial.print(" Ora =\t\t");
    ora = (tm.Hour);
    Serial.print(ora);
    Serial.write(':');
    minuto = (tm.Minute);

```

```

Serial.print(minuto);
Serial.write(':');
secondi = (tm.Second);
Serial.print(secondi);
Serial.println();
Serial.println();
} else {
if (RTC.chipPresent()) {
  Serial.println(" >>> Scheda DS1307 ferma. Usare esempio SetTime");
  Serial.println();
} else {
  Serial.println(" >>> DS1307 errore lettura! Controlla il circuito.");
  Serial.println();
}
delay(200);
}
switch (th.Read()) {
case 2:
  Serial.println(" CRC errato");
  Serial.println();
  break;
case 1:
  Serial.println(" >>> SENSORE ESTERNO OFFLINE !!! <<<");
  Serial.println();
  break;
case 0:
  Serial.print(" Temperatura esterna:\t");
  int ttht = th.t;
  Serial.print(ttht);
  Serial.println("*C");
  Serial.print(" Umidita' esterna\t");
  hy2 = (th.h);
  hy2 = hy2 + 3;
  Serial.print(hy2);
  Serial.println("%");
  Serial.println();
  break;
}
leggicorrentebatteria();
delay(500);
server.handleClient();
}
//*****
void handleRoot() {
  // Sensore DHT2003

```

```

dht1.setup(10); // data pin
delay(50);
float temperature = dht1.getTemperature();
delay(50);
float humidity = dht1.getHumidity();
delay(50);
int xx = digitalRead(ventola);
//Sensore luminosita'
int lumen = analogRead(A0);
lumen = lumen / 2;
if (lumen > 100)(lumen = 100);
//Dati schede misure
float A_busvoltage = (ina219_A.getBusVoltage_V());
if (A_busvoltage < 2)(A_busvoltage = 0);
int A_current_mA = ina219_A.getCurrent_mA();
int B_busvoltage = (ina219_B.getBusVoltage_V() * 10);
int B_current_mA = ina219_B.getCurrent_mA();
// Costruzione Messaggio da spedire su interrogazione
int cc = humidity + 10;
int tt = (temperature - 1);
int tt2 = th.t;
String message = "\n\n";
message += "\tGESTIONE DATI SERRA - v3.9";
message += "\n\n";
message += "\tData :t";
if (giorno < 10) message += "0";
message += (giorno);
message += "/";
if (mese < 10) message += "0";
message += (mese);
message += "/";
message += (anno);
message += "\n";
message += "\tOra :t";
if (ora < 10) message += "0";
message += (ora);
message += ":";
if (minuto < 10) message += "0";
message += (minuto);
message += ":";
if (secondi < 10) message += "0";
message += (secondi);
message += "\n\n";
message += "\tTemperatura esterna:\t";
message += (tt2);

```

```

message += " ^C";
message += "\n";
message += "\tUmidita' esterna\t";
message += (hy2);
message += "%\n";
switch (th.Read()) {
    case 1:
        message += ("\t>>> SENSORE OFFLINE !!! <<<");
        break;
}
message += "\n\n";
message += "\tTemperatura serra\t";
message += (tt);
message += " ^C ";
message += "\n";
message += "\tUmidita' serra\t\t";
message += (cc);
message += "%\n\n";
if (!xx)( message += "\t>>> Ventole in funzione <<<\n\n");
message += "\tLuminosita'\t\t";
message += (lumen);
message += "%\n\n";
message += "\tTensione batteria \t";
message += (A_busvoltage);
message += " V\n";
message += "\tCorrente carico \t";
message += (A_current_mA);
message += " mA\n\n";
if ((A_busvoltage) < 11.8)message += "\t>>> BATTERIA BASSA !!!! BATTERIA BASSA <<<\n\n";
message += "\tTensione Pannello \t";
message += (B_busvoltage);
message += " V\n";
message += "\tCorrente Pannello \t";
message += (B_current_mA);
message += " mA\n\n";
message += " -----";
for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
}
server.send(200, "text/plain", message);
}
//*****
void handleNotFound() {
    String message = "File Not Found\n\n";

```

```

message += "URI: ";
message += server.uri();
message += "\nMethod: ";
message += (server.method() == HTTP_GET) ? "GET" : "POST";
message += "\nArguments: ";
message += server.args();
message += "\n";
for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
}
server.send(404, "text/plain", message);
}
//*****
void leggicorrentebatteria() {
    float temperature = dht1.getTemperature();
    int t1 = temperature - 2;
    int vt;
    Serial.print("  Temperatura serra\t");
    Serial.print(t1); Serial.println(" ^C");
    delay(500);
    float humidity = dht1.getHumidity();
    int hy1 = humidity + 10;
    Serial.print("  Umidita' serra\t");
    Serial.print(hy1); Serial.println(" %");
    Serial.println();
    if (temperature > 31) {
        digitalWrite(ventola, LOW);
        delay(1000);
    };
    if (temperature < 26) {
        digitalWrite(ventola, HIGH);
        delay(1000);
    };
    delay(500);
    vt = digitalRead(ventola);
    if (vt)( Serial.println(" >>> VENTOLE SPENTE"));
    if (!vt)( Serial.println(" >>> VENTOLE ACCESE"));
    Serial.println();
    int lumen = analogRead(A0); //Sensore luminosita'
    lumen = lumen / 2;
    if (lumen > 100)(lumen = 100);
    Serial.print("  Luminosita'\t\t");
    Serial.print(lumen); Serial.println(" %"); Serial.println(" ");
    delay(500);
    //*****

```

```
float A_shuntvoltage = 0;
float A_busvoltage = 0;
float A_current_mA = 0;
float A_loadvoltage = 0;
float A_power_mW = 0;
```

```
float B_shuntvoltage = 0;
float B_busvoltage = 0;
float B_current_mA = 0;
float B_loadvoltage = 0;
float B_power_mW = 0;
```

```
A_shuntvoltage = ina219_A.getShuntVoltage_mV();
A_busvoltage = ina219_A.getBusVoltage_V();
if (A_busvoltage < 2)(A_busvoltage = 0);
A_current_mA = ina219_A.getCurrent_mA();
if (A_current_mA < 2)(A_current_mA = 0);
A_power_mW = ina219_A.getPower_mW();
A_loadvoltage = A_busvoltage + (A_shuntvoltage / 1000);
```

```
B_shuntvoltage = ina219_B.getShuntVoltage_mV();
B_busvoltage = ina219_B.getBusVoltage_V();
B_current_mA = ina219_B.getCurrent_mA();
B_power_mW = ina219_B.getPower_mW();
B_loadvoltage = B_busvoltage + (B_shuntvoltage / 1000);
```

```
Serial.println(" BATTERIA");
Serial.print(" Tensione Batteria:\t");
Serial.print(A_busvoltage);
Serial.println(" V");
Serial.print(" Corrente carico:\t");
Serial.print(A_current_mA);
Serial.println(" mA");
if ((A_busvoltage) < 11.8)( Serial.println(" >>> BATTERIA BASSA !!!!
BATTERIA BASSA <<<"));
Serial.println(" ");
Serial.println(" PANNELLO SOLARE");
Serial.print(" Tensione Pannello:\t");
Serial.print(B_busvoltage);
Serial.println(" V");
Serial.print(" Corrente erogata:\t");
Serial.print(B_current_mA);
Serial.println(" mA");
Serial.println(" ");
Serial.println();
```



```

Serial.println(" ");
delay(5000);
}
//*****
void scriviSD() {
File dataFile = SD.open("datalog.txt", FILE_WRITE);
if (dataFile) {
// Sensore DHT2003
dht1.setup(10); // data pin
delay(50);
float temperature = dht1.getTemperature();
delay(50);
float humidity = dht1.getHumidity();
delay(50);
//Sensore luminosita'
int lumen = analogRead(A0);
lumen = lumen / 2;
if (lumen > 100)(lumen = 100);
if (lumen < 8)(lumen = 0);
int A_busvoltage = (ina219_A.getBusVoltage_V() * 10);
if (A_busvoltage < 20)(A_busvoltage = 0);
int A_current_mA = ina219_A.getCurrent_mA();
int B_busvoltage = (ina219_B.getBusVoltage_V() * 10);
int B_current_mA = ina219_B.getCurrent_mA();
int cc = humidity + 10;
int tt = (temperature - 1);
int tt2 = th.t;
int vnt = digitalRead (ventola);
// Costruzione Messaggio da REGISTRARE
String dataString = "";
dataString += (giorno);
dataString += ",";
dataString += (mese);
dataString += ",";
dataString += (anno);
dataString += ",";
dataString += (ora);
dataString += ",";
dataString += (minuto);
dataString += ",";
dataString += (tt2);
dataString += ",";
dataString += (hy2);
dataString += ",";
dataString += (vnt);
}
}

```

```
dataString += ",";
dataString += (tt);
dataString += ",";
dataString += (cc);
dataString += ",";
dataString += (lumen);
dataString += ",";
dataString += (A_busvoltage);
dataString += ",";
dataString += (A_current_mA);
dataString += ",";
dataString += (B_busvoltage);
dataString += ",";
dataString += (B_current_mA);
dataFile.println(dataString);
dataFile.close();
Serial.print("  Dati registrati ");
Serial.println(dataString);
Serial.println("
*****")
;
}
else {
  Serial.println("error opening datalog.txt");
}
delay(1000);
}
//*****
```