

```

// Librerie necessarie per il progetto
#include <Wire.h>
#include <WProgram.h>
#include <SPI.h>
#include <dht11.h>
#include <Ethernet.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified(10085);
float p=0;
dht11 DHT11;
#define DHT11PINA 2
//creo un array di byte per specificare il mac address
byte mac[] = { 0xDE, 0xAE, 0xBE, 0xEF, 0xFE, 0xED };
//creo un array di byte per specificare l'indirizzo ip
//modificare questo valore in base alla propria rete
byte ip[] = { 10, 3, 3, 155 };
char Data_RX;
String msg = "";
int cmd = 0;
//creo un oggetto server che rimane in ascolto sulla porta specificata
Server ArduinoServer(8080);

    int val1 = 0;    // variabile per morizzare una misura
    int alarmPin = 5; // uscita allarme su digital pin 5
    int val2 = 0;    // variable memorizzazione allarme
    int val3 = 0 ;   // variabile per il controllo del refresh
    int lumen1 =0;
    int pir =0;

// variabili di data/ora
int hour;
int minute;
int second;
int month;
int day_of_week;
int day;
int year;

float temp_c;

void setup()
{
//inizializzo la I2C
Wire.begin();
delay(300);
//inizializza lo shield con il mac e l'ip
Ethernet.begin(mac, ip);
//inizializza l'oggetto server
ArduinoServer.begin();
//definisco il modo di pin3 e altri
pinMode(3,OUTPUT);

```

```

pinMode(alarmPin,OUTPUT);
if(!bmp.begin())
{
while(1);
}

}

void loop()
{
//IMPORTANTE pulisco la variabile msg e val3
msg = "";
val3 = 0;
Client client = ArduinoServer.available();
//controllo se client è true
if (client != false)
{
while (client.connected()) //controllo continuamente che il client sia connesso
{
if (client.available()) //Controllo se ci sono byte disponibili per la lettura
{
Data_RX = client.read(); //leggo i byte disponibili provenienti dal client
msg += Data_RX; //ricostruisco la stringa ricevuta concatenando i singoli byte
//Attendo che tutti i byte siano letti quando Data_RX contiene il carattere
//di nuova linea capisco che tutti i byte sono stati letti
if (Data_RX == '\n')
{
//cerco all'interno della stringa il parametro che mi interessa
delay(2000);
//Invio la risposta al client
//invio lo status code
client.println("HTTP/1.1 200 OK");
//imposto il data type
client.println("Content-Type: text/html");
client.println();
//invio codice html
client.print("<html><body>");
client.print("<head>");

cmd = msg.indexOf("DATI");
if (cmd > 0)
{
// Circuito data-ora I2C metto il registro indirizzi a 0.
Wire.beginTransmission(104); // trasmetto al device #104, il DS1307
Wire.send(0x00);
Wire.endTransmission(); // fine trasmissione
Wire.requestFrom(104, 7); // richiesta di 7 byte da slave DS1307

// determino l'appartenenza del byte
second = Wire.receive();
minute = Wire.receive();
hour = Wire.receive();
}
}
}
}
}

```

```

    day_of_week=Wire.receive();
    day = Wire.receive();
    month = Wire.receive();
    year = Wire.receive();
// Conversione di tutti i valori in decimale
    hour=hour/16 * 10 + hour % 16;
    minute=minute/16 * 10 + minute % 16;
    second=second/16 * 10 + second % 16;
    day=day/16 * 10 + day % 16;
    month=month/16 * 10 + month % 16;
    year=2000 + year/16 * 10 + year % 16;
    delay(500);
// client.print("Scheda Arduino2009 + scheda Ethernet ");
    client.println("<br />");
    client.print("Scheda Arduino Uno + scheda Ethernet + Sensori esterni ");
    // client.println("&nbsp");client.println("&nbsp"); client.println("&nbsp");
    client.println(" ---");
    client.println("&nbsp");client.println("&nbsp"); client.println("&nbsp");
// client.println("&nbsp");client.println("&nbsp");client.println("&nbsp");
// scrittura data
    if (day < 10) { client.print("0"); }
    client.print(day);
    delay(100);
    client.print("/");
    if (month < 10) { client.print("0"); }
    client.print(month);
    delay(100);
    client.print("/");
    client.print(year);
    delay(100);
    client.print(" --- ");
// scrittura ora
    if (hour < 10) { client.print("0"); }
    client.print(hour);
    delay(100);
    client.print(":");
    if (minute < 10) { client.print("0"); }
    client.print(minute);
    delay(100);
    client.print(":");
    if (second < 10) { client.print("0"); }
    client.print(second);
    delay(100);
    client.println("<br />");
    client.println("<br />");
    delay(100);

//legge la luminosità tramite Fotoresistenza
    client.print("Luminosita' ");
    lumen1=(analogRead(A0));
    lumen1=(lumen1-810) *100/210;

```

```

        client.print(lumen1);
        client.print(" % ");
        client.println("&nbsp");client.println("&nbsp");
        client.println("&nbsp");client.println("&nbsp");

//legge Temperatura e Umidità
int chk = DHT11.read(DHT11.PINA);
    client.print(" Temperatura ");
    client.print(((float)DHT11.temperature), 1);
    client.print("^C");
    client.println("&nbsp");client.println("&nbsp");
    client.println("&nbsp");client.println("&nbsp");
    delay(400);
    client.print(" Umidita' ");
    client.print((int)DHT11.humidity);
    client.print("% ");
    delay(300);
    client.println("&nbsp");client.println("&nbsp");
    client.println("&nbsp");client.println("&nbsp");

//Legge la Pressione
    sensors_event_t event;
    bmp.getEvent(&event);

//Visualizzo pressione in hPa
    if (event.pressure)
    {
        client.print("Pressione:  ");
        p=(event.pressure);
        p=p+58;
        client.print(p,0);
        client.println(" hPa");
    }
    }
    client.print("</body></html>");
//aspetto 4 ms affinché la risposta giunga al browser del client
    delay(4);
//esco dal ciclo while una volta completato l'invio della risposta
    break;
    }
    }
}

client.stop(); //chiudo la connessione
}
}

```