

```

// Controllo remoto 05/05/2012
// poichè sono ancora ai primi passi ho usato
// a larghe mani idee e listati di altri. Ringrazio tutti loro.
// Librerie necessarie per il progetto
#include <Wire.h>
#include <SHT1x.h>
#include <WProgram.h>
#include <SPI.h>
#include <Ethernet.h>
#include "Ultrasonic.h"
//creo un array di byte per specificare il mac address
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
//creo un array di byte per specificare l'indirizzo ip
byte ip[] = { 192, 168, 0, 77 }; //modificare questo valore in base alla propria rete
char Data_RX;
String msg = "";
int cmd = 0;
//creo un oggetto server che rimane in ascolto sulla porta specificata
Server ArduinoServer(80);
int inPin = 2; // pulsante connesso al digital pin 2
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
// variabili di data/ora
int hour;
int minute;
int second;
int month;
int day_of_week;
int day;
int year;
// definizione per SHT11 - temperatura interna
#define dataPin 6
#define clockPin 7
SHT1x sht1x(dataPin, clockPin);
//pin dedicati al sensore ultrasuoni
Ultrasonic ultrasonic(8,9);
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(inPin, INPUT);
pinMode(alarmPin, OUTPUT);

```

```

}
void loop()
{
//IMPORTANTE pulisco la variabile msg e val3
msg = "";
val3 = 0;
Client client = ArduinoServer.available();
//controllo se client è true
if (client != false)
{
//controllo continuamente che il client sia connesso
while (client.connected())
{
//Controllo se ci sono byte disponibili per la lettura
if (client.available())
{
//leggo i byte disponibili provenienti dal client
Data_RX = client.read();
//ricostruisco la stringa ricevuta concatenando i singoli byte
msg += Data_RX;
//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>");
/*
//questo codice html permette di eseguire il refresh automatico della pagina html
//funziona su chiamata diretta ma vorrei usarlo per refresh elenco dati su mio sito
client.print("<META HTTP-EQUIV='refresh' CONTENT='5; />");
client.print("</head>");
*/
//Piloto l'uscita e invio lo stato al browser
cmd = msg.indexOf("ON");
if (cmd > 0)
{
val3 = 1;
digitalWrite(3, HIGH);
client.print("</head>");
/*
//questo codice html permette di eseguire il refresh automatico della pagina html
//funziona su chiamata diretta ma vorrei usarlo per refresh elenco dati su mio sito

```

```

client.print("<META HTTP-EQUIV='refresh' CONTENT='0; />");
client.print("</head>");
*/
}
cmd = msg.indexOf("OFF");
if (cmd > 0)
{
val3= 1;
digitalWrite(3, LOW);
client.print("</head>");
/*
//questo codice html permette di eseguire il refresh automatico della pagina html
//funziona su chiamata diretta ma vorrei usarlo per refresh elenco dati su mio sito
client.print("<META HTTP-EQUIV='refresh' CONTENT='0; />");
client.print("</head>");
*/
}
cmd = msg.indexOf("BLK");
if (cmd > 0)
{
val3 = 1;
digitalWrite(3, HIGH);
client.print("</head>");
/*
//questo codice html permette di eseguire il refresh automatico della pagina html
//funziona su chiamata diretta ma vorrei usarlo per refresh elenco dati su mio sito
client.print("<META HTTP-EQUIV='refresh' CONTENT='5; />");
client.print("</head>");
*/
delay(6000); // ritardo n-sec per eventuale avvio utilizzi esterni
digitalWrite(3, LOW);
client.print("</head>");
/*
//questo codice html permette di eseguire il refresh automatico della pagina html
//funziona su chiamata diretta ma vorrei usarlo per refresh elenco dati su mio sito
client.print("<META HTTP-EQUIV='refresh' CONTENT='5; />");
client.print("</head>");
*/
}
cmd = msg.indexOf("DATI");
if (val3=1)
{
cmd=1;
}
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("Test real time on line -- Ciao, Arrigo ");
client.println("<br />");
client.println("<br />");
// 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(300);
// trasmette il valore di ciascun ingresso analogico
for (int analogChannel = 0; analogChannel < 4; analogChannel++) {
client.print("analog input ");

```

```

client.print(analogChannel);
client.print(" is ");
client.print(analogRead(analogChannel));
client.println("<br />");
client.println("<br />");
}
val1 = digitalRead(3); // read the input pin
client.print(" Relay_1 ");
if (val1 > 0){client.print(" ON ");}
if (val1 < 1){client.print(" OFF ");}
client.println("<br />");
delay(100);
// controlla lo stato del sensore ultrasuoni
delay(200);
val2=ultrasonic.Ranging(CM);
delay(100);
if (val2<70){
digitalWrite(alarmPin, HIGH); //eccita il relè
delay(100);
client.println("<br />");
client.print(" Area libera cm ");
client.print(val2);
delay(100);
client.print(" -->>> ALLARME "); // trasmette l'allarme
client.println("<br />");
delay(2000);
}
else
{
digitalWrite(alarmPin, LOW); // relè diseccitato
client.println("<br />");
client.print(" cm ");
client.print(val2);
client.print(" -->> TUTTO OK "); // trasmette OK
client.println("<br />");
}
//legge Temperatura e Umidità del sensore SHT11
float temp_c;
int humidity;
temp_c = sht1x.readTemperatureC();
humidity = sht1x.readHumidity();
delay(300);
client.println("<br />");
client.print(" Temperatura interna ");
client.print((temp_c-2),1);
client.print(" gradiC");
client.println("<br />");
client.println("<br />");
client.print(" Umidita' interna ");
client.print(humidity); //Umidità
client.print("% ");

```

```
client.println("<br />");
client.println("<br />");
}
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;
}
}
}
//chiudo la connessione
client.stop();
}
}
```