

Progetto: Rilevatore di temperatura, umidità e intensità luminosa tramite scheda Arduino, e gestione dati tramite database SQL.

1. Premessa:

Si intende realizzare un sistema che rilevi la temperatura, l'umidità e l'intensità luminosa all'interno di un determinato ambiente, e che memorizzi tali dati rilevati in un database SQL, in modo che tali dati siano accessibili all'utente tramite una pagina PHP. Potenzialmente questo sistema permette all'utente di rilevare in remoto tali valori, e visionarli anche se l'utente non si trova nello stesso luogo in cui è posizionato il sistema. In via dimostrativa, tale progetto sarà realizzato in locale (tramite web server e database server, che saranno gestiti direttamente all'interno del pc che si utilizzerà per dimostrare il sistema), ma sia chiaro che tale sistema si può estendere in dei server in remoto.

2. Componenti utilizzati per la realizzazione del progetto:

- Arduino Uno
- Arduino Ethernet Shield
- DHT 11
- Resistenza 10k Ω
- Fotoresistenza
- Cavo Ethernet incrociato
- Display LCD (opzionale)

3. Descrizione:

Per realizzare tale progetto, si utilizza la scheda programmabile Arduino Uno. Tale scheda permette di rilevare i vari dati quali temperatura, umidità e intensità luminosa dai vari sensori. Questo è possibile grazie agli input digitali ed analogici che mette a disposizione la scheda Arduino. Il rilevamento dell'intensità luminosa è possibile grazie ad un sensore che in base all'intensità luminosa di un determinato ambiente, varia la resistenza (fotoresistenza). Quindi per ottenere dei valori si crea un partitore di tensione, utilizzando una resistenza equivalente alla fotoresistenza (resistenza da 10k Ω), e collegando in serie il sensore, con la resistenza. Ora si misura la tensione ai capi del sensore. Date alcune difficoltà a trasformare la tensione rilevata in LUX (unità di misura dell'intensità luminosa), si è deciso di mostrare all'utente direttamente la tensione ai capi del sensore, che dall'Arduino è codificata con 10 bit. Quindi verrà mostrato un valore compreso da 0 a 1023 (2^{10}). Per quanto riguarda il rilevamento della temperatura e dell'umidità, si è scelto di utilizzare il sensore digitale DHT11. Dopo aver provato a rilevare la temperatura tramite un termistore utilizzando un partitore di tensione, e un sensore di temperatura analogico, si è appurato che i risultati ottenuti non erano accurati. Quindi si è scelto, sia per accuratezza che per semplicità

d'uso di utilizzare il sensore digitale DHT11 che, per appunto, permette oltre alla misurazione della temperatura, la misurazione dell'umidità. Per quanto riguarda l'inserimento dei dati all'interno di un database, si utilizza la scheda di rete Ethernet che permette l'interfacciamento con la scheda Arduino. Si è detto precedentemente che i server, in via dimostrativa, saranno gestiti in locale. Quindi la scheda Ethernet dell'Arduino verrà collegata al PC tramite un cavo Ethernet incrociato, dato che i due dispositivi fanno parte dello stesso livello ISO/OSI. Ora, per l'invio dei dati al database, si utilizza il seguente metodo. Si crea una pagina PHP di input che permetta la ricezione di parametri GET, e li inserisca all'interno del database. Il programma caricato sulla scheda si occuperà di generare una stringa che servirà a passare i parametri misurati dalla scheda tramite, appunto, il metodo GET, ogni tot di secondi (nel caso del nostro esempio si effettuerà la scrittura dei dati sul database ogni 10 secondi). Inoltre, come esempio di utilizzo dei dati, si crea una pagina PHP che faccia vedere all'utente gli ultimi 10 valori registrati dalla scheda Arduino, all'interno di una tabella. Tali valori saranno: data e ora della misurazione (data e ora verranno prese dal web server), temperatura, umidità e intensità luminosa.

Si allega lo schema del progetto, e il codice caricato sulla scheda Arduino.

OPZIONALE: Al sistema si potrebbe aggiungere un Display LCD da implementare alla scheda Arduino, in modo che l'utente, oltre a visionare i dati rilevati tramite la pagina PHP, possa visionarli tramite un Display LCD

4. Codice Arduino:

```
#include <Ethernet.h>
```

```
#include <dht11.h>
```

```
#define SEC 10
```

```
float milli = 1000;
```

```
float temp = 0;
```

```
int analogPin0 = A0;
```

```
int digitalPin0 = 9;
```

```
float temp_celsius;
```

```
float umidita;
```

```
float lum = 0;
```

```
int c=0;
```

```
String strURL = "";
```

```
dht11 DHT;
```

```
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x3F, 0x5C};
```

```
//indirizzo server web (locale)
```

```
IPAddress server(192, 168, 1, 8);
```

```
//indirizzo ip dell'Arduino
```

```
IPAddress ip(192, 168, 1, 200);
```

```
EthernetClient client;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  pinMode(analogPin0, INPUT);
```

```
  pinMode(digitalPin0, INPUT);
```

```
  Ethernet.begin(mac, ip);
```

```
}
```

```
void loop() {
```

```
  for(c=0; c<20; c++){
```

```
    lum += analogRead(analogPin0);
```

```
  }
```

```
  lum = lum / 20;
```

```
  int check = DHT.read(digitalPin0);
```

```
  temp_celsius = DHT.temperature;
```

```
umidita = DHT.humidity;

if(client.connect(server, 80))
{
  strURL = "GET /arduino/input.php?temp=";
  strURL += (float)temp_celsius;
  strURL += "&lum=";
  strURL += (float)lum;
  strURL += "&umidita=";
  strURL += (float)umidita;
  strURL += " HTTP/1.1";

  //invio la richiesta al server
  Serial.println(strURL);
  client.println(strURL);
  client.println("Host: localhost");
  client.println("Connection: close");
  client.println();
  //chiudo la connessione
  client.stop();
}else
{
  Serial.println("Errore di connessione");
}

if(client.connected()){
  client.stop();
}

milli = SEC * 1000;
```

