

Arduino: Comandare da remoto l'accensione/spegnimento di due led

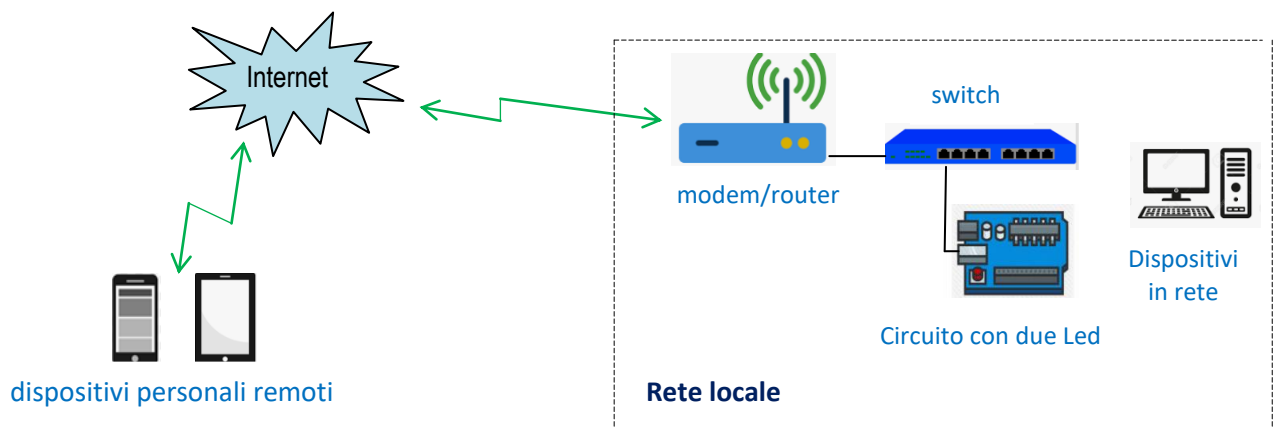
Il progetto illustrato nel precedente articolo permetteva di leggere, via Internet, i valori remoti di temperatura e luminosità misurati da una scheda Arduino.

In questo nuovo progetto vediamo come comandare Arduino da un dispositivo remoto attraverso un'applicazione che permette di accendere o spegnere due led.

Per le competenze di Sistemi e Reti necessarie per capirne il funzionamento e per le abilità richieste per configurare alcuni dispositivi, il progetto è particolarmente adatto agli studenti del quarto/quinto anno delle specializzazioni Informatica e Telecomunicazioni degli Istituti Tecnici Tecnologici.

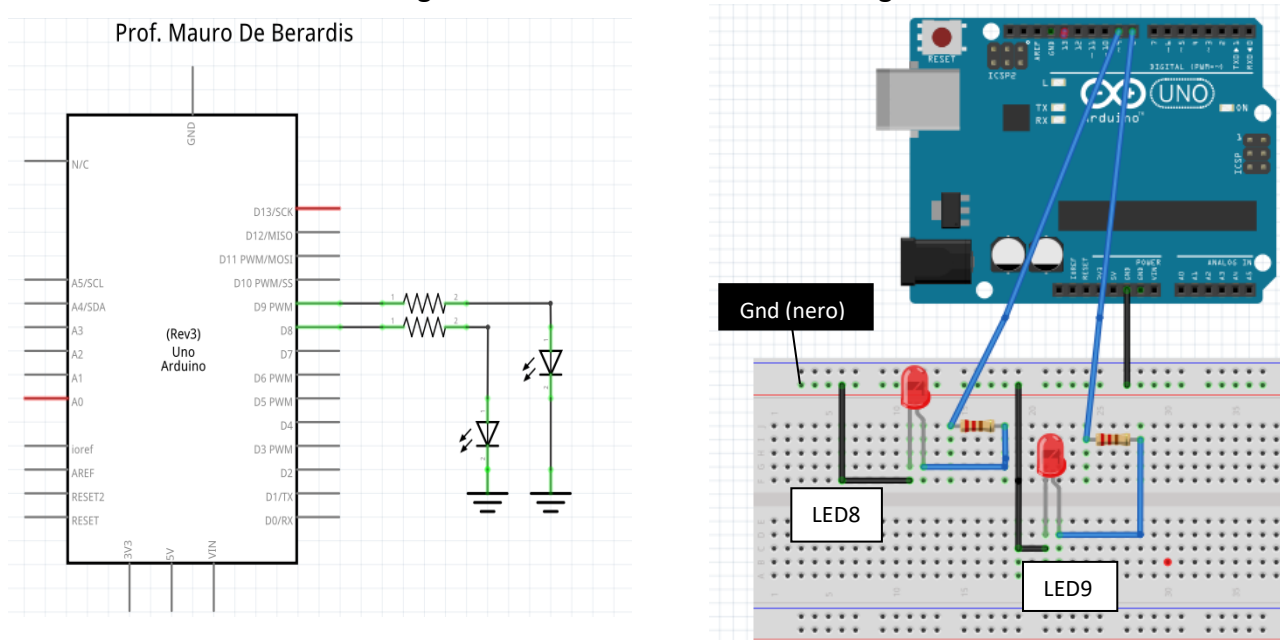
L'applicazione scelta è semplice e facilmente riproducibile e fa capire come sia possibile realizzare con Arduino controlli remoti molto utili e diffusi (ad esempio l'accensione/spegnimento di un motore o di una caldaia, di un impianto di irrigazione o di un cancello automatico, delle luci di un giardino o di una telecamera di sorveglianza).

Lo schema generale dell'applicazione è il seguente:



Arduino, connesso ad un router di una rete locale, e quindi a Internet, tramite una Ethernet Shield e la libreria "Ethernet Library", esegue i comandi ricevuti da un dispositivo remoto per accendere/spegnere due led che sono connessi sui pin digitali D8 e D9.

Lo schema elettrico e i collegamenti sulla breadboard sono i seguenti:

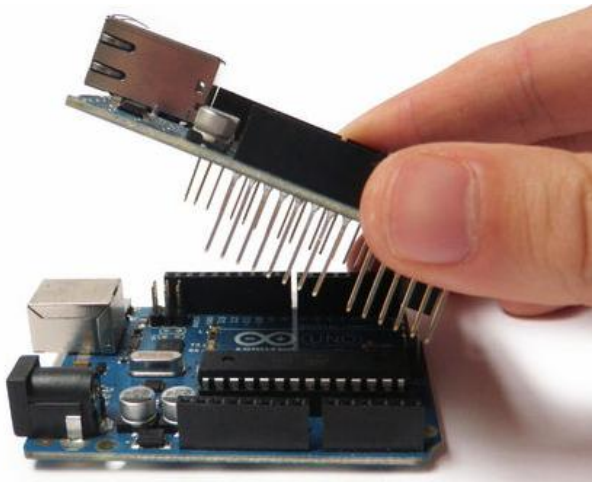


Componenti necessari

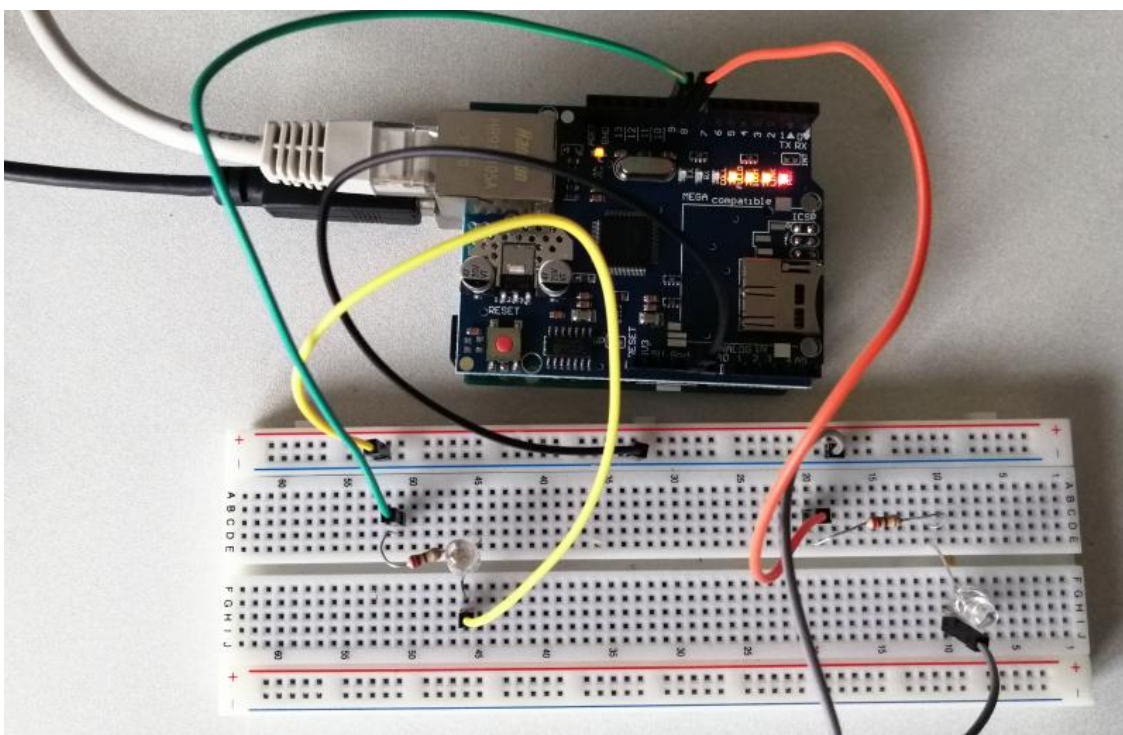
- Arduino collegato ad un PC Windows X
- Ethernet Shield con "Ethernet Library"
- Breadboard con cavetteria
- 2 Led
- 2 Resistenze di limitazione della corrente dei led da 220Ohm

Circuito di prova

Dopo aver inserito Ethernet Shield in Arduino



realizziamo, utilizzando i piedini D8, D9 e Gnd di Arduino "replicati" su Ethernet Shield, i collegamenti previsti per realizzare il circuito. Colleghiamo quindi Arduino al PC (cavo USB nero) ed Ethernet Shield allo switch di rete (cavo di rete bianco)



L'Ethernet Shield, tramite il cavo di rete, connette Arduino alla rete locale, e quindi al router della nostra abitazione, consentendo ad un qualsiasi dispositivo (pc, tablet o smartphone) connesso ad Internet di modificare lo stato Acceso/Spento dei due led connessi su D8 e D9. Infatti, come vedremo tra poco, lo sketch utilizzato implementa in Arduino un web server con cui è possibile scambiare informazioni via protocollo http sulla porta 80.

Un po' di teoria

Prima di andare avanti è opportuno ricordare i principi di funzionamento sui quali si basa la suite di protocolli TCP/IP (Transport Control Protocol/Internet Protocol) che è alla base del funzionamento della stragrande maggioranza delle reti pubbliche e private. In un network basato sul TCP/IP, ciascun sistema è identificato in modo univoco da un indirizzo IP, costituito da quattro byte del tipo, in decimale, 192.168.1.100 e da un numero di porta di comunicazione. Tramite IP e numero di porta comunica con altri sistemi scambiando messaggi sotto forma di pacchetti. L'indirizzo IP, analogamente ad un numero di telefono, garantisce la possibilità di instaurare la comunicazione mentre la porta non è altro che un numero che serve a differenziare il servizio di rete, cioè l'applicazione usata per la comunicazione stessa (ad esempio, il servizio http, che ci permette di navigare sul web, ha tipicamente un numero di porta uguale ad 80, il servizio ftp, che permette di scaricare i files, ha un numero di porta uguale a 21 ecc.).

Gli indirizzi IP possono essere privati oppure pubblici.

Gli IP privati sono utilizzabili soltanto all'interno di reti locali e sono replicabili quante volte si vuole su altre reti locali. In genere tali indirizzi fanno parte delle seguenti serie di indirizzi riservati e che non possono essere instradati su Internet:

- 10.0.0.0 ÷ 10.255.255.255
- 172.16.0.0 ÷ 172.31.255.255
- 192.168.0.0 ÷ 192.168.255.255

Gli indirizzi pubblici sono invece non replicabili (non ne possono esistere due uguali) e consentono di accedere ad Internet.

Ogni dispositivo di una rete locale è contraddistinto da un indirizzo IP privato (ad esempio 192.168.1.10) ed è collegato, via cavo o via wifi, ad un router con IP appartenente alla stessa rete locale (generalmente l'IP di default del router è 192.168.1.1).

Per connettersi ad Internet il router ha bisogno di un indirizzo pubblico, indirizzo che viene fornito dal Provider dei servizi che fornisce la connettività Internet. Tale indirizzo può essere determinato in vari modi, il più semplice è quello di collegarci ad uno dei tanti siti, ad esempio www.mioip.it, che forniscono il numero Ip pubblico a cui il router della rete locale da cui si accede risulta connesso.

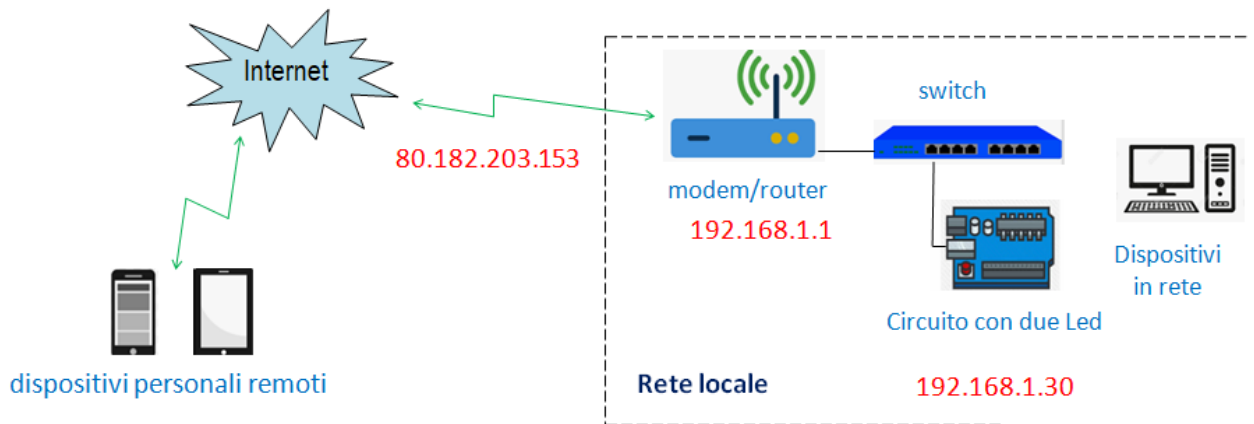
Accedendo ora, mioip.it mi dice:

Il tuo indirizzo IP è:

80.182.203.153

Torniamo sul progetto

Per quanto detto, la configurazione degli IP pubblici e privati di cui tener conto, è la seguente:



Il modem/router (Ip privato 192.168.1.1) si collega ad Internet con l'IP pubblico 80.182.203.153 e, grazie agli indirizzi IP privati della rete locale, instrada i vari pacchetti di dati verso Arduino e gli altri dispositivi in rete (PC desktop, notebook, stampanti di rete, ecc.)

Il circuito con due led realizzato con Arduino ed Ethernet Shield si collega alla rete locale, e quindi al router, con l'IP privato 192.168.1.30.

Pertanto, per accendere/spegnere da remoto i due led, dobbiamo prima raggiungere l'indirizzo pubblico del router e poi essere instradati verso l'indirizzo privato di Arduino.

Si presentano due problemi:

a) Dobbiamo conoscere l'indirizzo IP pubblico del router

Purtroppo tale indirizzo non sempre è fisso; anzi, per le utenze domestiche generalmente è dinamico e può variare con una certa frequenza. Il sito mioip.it oggi ci dice che l'IP pubblico della nostra rete è 80.182.203.153, domani tale indirizzo potrebbe essere un altro.

La soluzione migliore è ovviamente quella di attivare un IP fisso sul nostro router, ma spesso questa strada è costosa e impraticabile.

In alternativa possiamo utilizzare il servizio di DNS dinamico che diversi siti forniscono gratuitamente. Io utilizzo il servizio gratuito di **www.noip.com** che mi ha fornito un alias del tipo **xxxx.ddns.net**: a tale alias viene associato l'IP pubblico del router. L'aggiornamento dinamico di tale indirizzo viene effettuato da un app fornita da **www.noip.com**, presente sul mio server di rete e che si avvia come servizio Windows. Pertanto, se sul mio cellulare nella barra degli indirizzi del browser, digito **xxxx.ddns.net**, vengo reindirizzato verso l'IP pubblico che in quel momento risulta attivo nel mio router.

b) Dobbiamo configurare il router in modo che, una volta raggiunto da remoto attraverso l'IP pubblico, stabilisca una comunicazione con la Ethernet Shield attraverso la porta 80.

Per fare questo occorre impostare un "port forwarding", ovvero una nuova regola di routing per far sì che ogni volta che da remoto giunge una richiesta http sulla porta 80, questa venga reindirizzata all'IP privato (nel nostro caso 192.168.1.30) a cui è associato il misuratore di temperatura e luminosità. Come fare?

Facendo riferimento al modem/router della mia rete (le modalità operative variano da modello a modello ma generalmente le opzioni che ho utilizzato sono presenti in tutti i modelli commerciali):

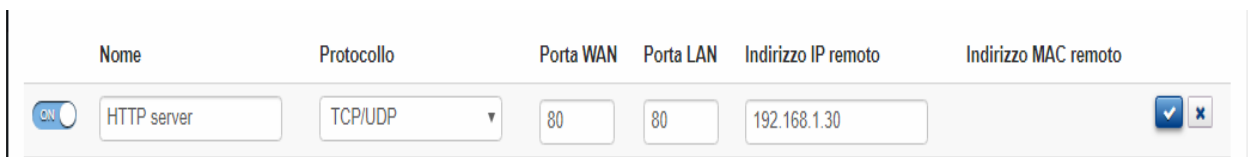
- nella sezione "Rete locale" aggiungo un IP statico associato al MAC address che utilizzerò nello sketch di Arduino. Ethernet Shield è una scheda basata sul chip intelligente W5100 che non ha un MAC Address preassegnato e può essere configurato con un indirizzo di 6 cifre esadecimali qualsiasi, purchè diverso dai MAC Address di altri dispositivi della LAN. Ad esempio scegliamo l'indirizzo 80:80:80:80:80:80.



Hostname	MAC address	IP
Arduino	80:80:80:80:80:80	192.168.1.30

Aggiungere un indirizzo IP statico

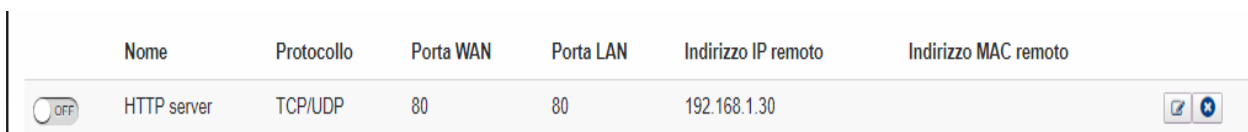
- nella sezione "servizi Wan", imposto la seguente regola di routing:



Nome	Protocollo	Porta WAN	Porta LAN	Indirizzo IP remoto	Indirizzo MAC remoto
HTTP server	TCP/UDP	80	80	192.168.1.30	

che instrada le richieste HTTP verso l'IP privato associato alla scheda Arduino.

Corre l'obbligo a questo punto precisare che se la rete non è ben protetta sia internamente che esternamente, il port forwarding espone a problemi di sicurezza informatica. Tralasciando altre considerazioni e tenendo conto che il progetto è puramente didattico, consiglio di disabilitare la porta 80, aperta sull'IP 192.168.1.30, una volta testato il progetto.



Nome	Protocollo	Porta WAN	Porta LAN	Indirizzo IP remoto	Indirizzo MAC remoto
HTTP server	TCP/UDP	80	80	192.168.1.30	

In alternativa, se si vuole testare il progetto in rete locale, evitando di intervenire sul modem/router, è sufficiente digitare sulla barra degli indirizzi del browser l'IP assegnato ad Arduino sullo sketch (nel nostro caso 192.168.1.30)

Sketch di Arduino

```
// Comandare da remoto l'accensione/spegnimento di due led
// Mauro De Berardis
//
// SPI.h, Ethernet.h, Sting.h sono tre librerie necessarie per realizzare il progetto
// Con la libreria Ethernet.h è possibile far diventare Arduino un mini server web
// Con la libreria String.h gestiamo le stringhe

#include <SPI.h>
#include <Ethernet.h>
#include <String.h> //gestione stringhe
/*
 * Ethernet Shield è una scheda basata sul chip intelligente W5100 che non ha un MAC Address
 * preassegnato e può essere configurato con un indirizzo di 6 cifre esadecimali qualsiasi
 * purchè diverso dai MAC Address di altri dispositivi della LAN.
 * Scegliamo l'indirizzo 80:80:80:80:80:80 utilizzato per configurare un IP statico sul router.
 */
byte macaddress[] = {0x80, 0x80, 0x80, 0x80, 0x80, 0x80 };
/*
 * 192.168.1.30 è l'IP privato della scheda Arduino
 */
byte ipprivato[]={192,168,1,30};
/*
 * 192.168.1.1 è l'IP privato del router(il gateway che permette ai dispositivi della rete
 ] di accedere ad Internet)
 */
byte gateway[]={192,168,1,1};
byte subnetmask[] = { 255, 255, 255, 0 }; //subnet mask
/*
 * La porta di default del protocollo HTTP è la porta 80
 */
String StringaLetta;
boolean LED8_Acceso = false;
boolean LED9_Acceso = false;
// Viene inizializzata la libreria Ethernet e il Server Web viene eseguito sulla porta 80
// http standard
EthernetServer Server(80);

void setup() {
  //inizialmente I due led sono spenti;
  pinMode(8,OUTPUT);
  digitalWrite(8,LOW);
  pinMode(9,OUTPUT);
  digitalWrite(9,LOW);
  Ethernet.begin(macaddress, ipprivato,gateway,subnetmask);
  Server.begin();
}

void loop() {
  // in ascolto per l'arrivo di nuovi client
  EthernetClient client = Server.available();
  if (client)
  {
    StringaLetta = "";
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        StringaLetta +=c;
        // Se viene completato l'invio della richiesta HTTP, allora il server invia la risposta
        if (c == '\n')
        {
          // Viene fatta una risposta HTTP, in pratica viene creata una pagina WEB in HTML
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
        }
      }
    }
  }
}
```

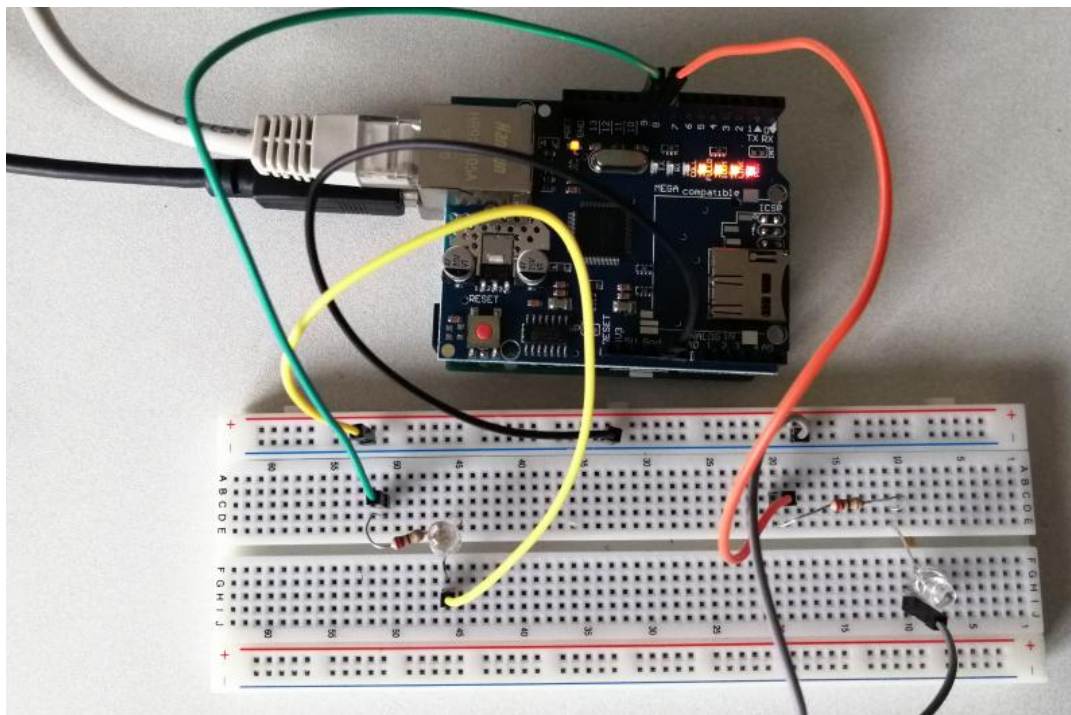
```

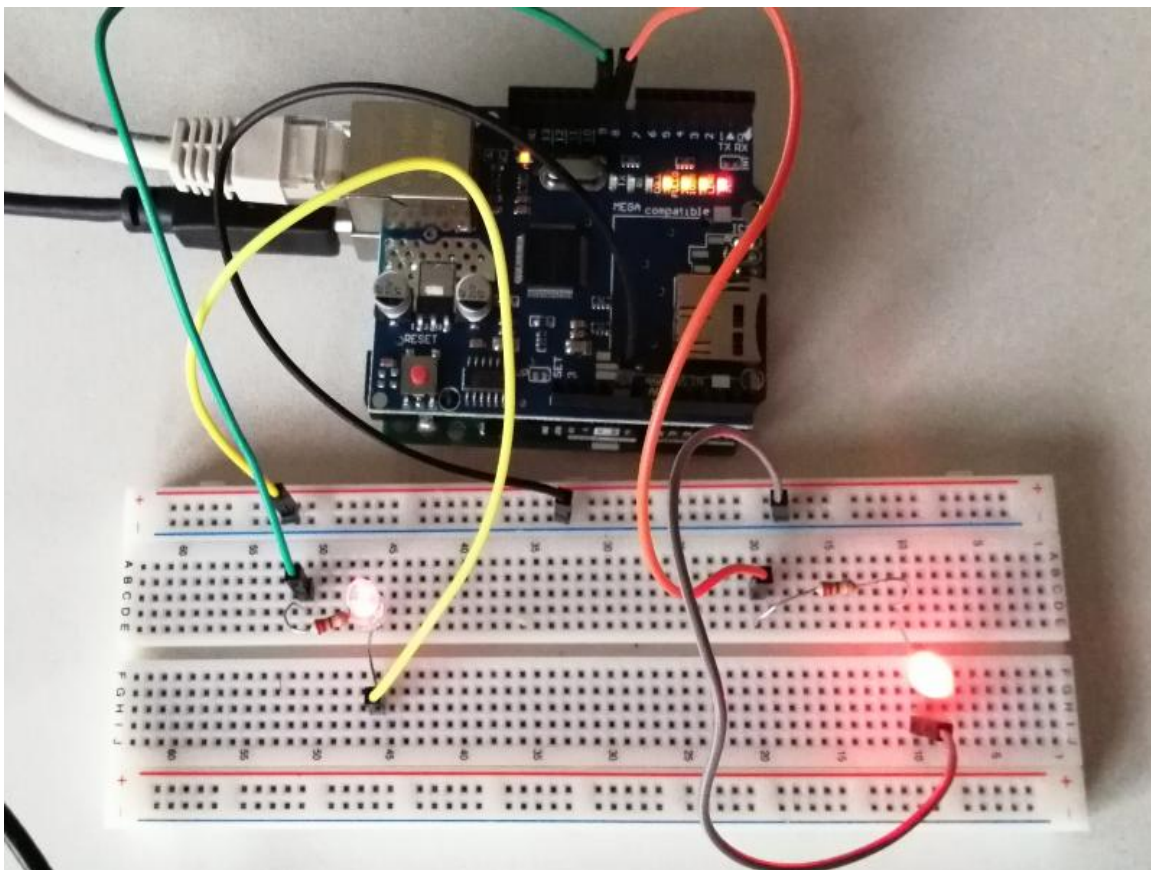
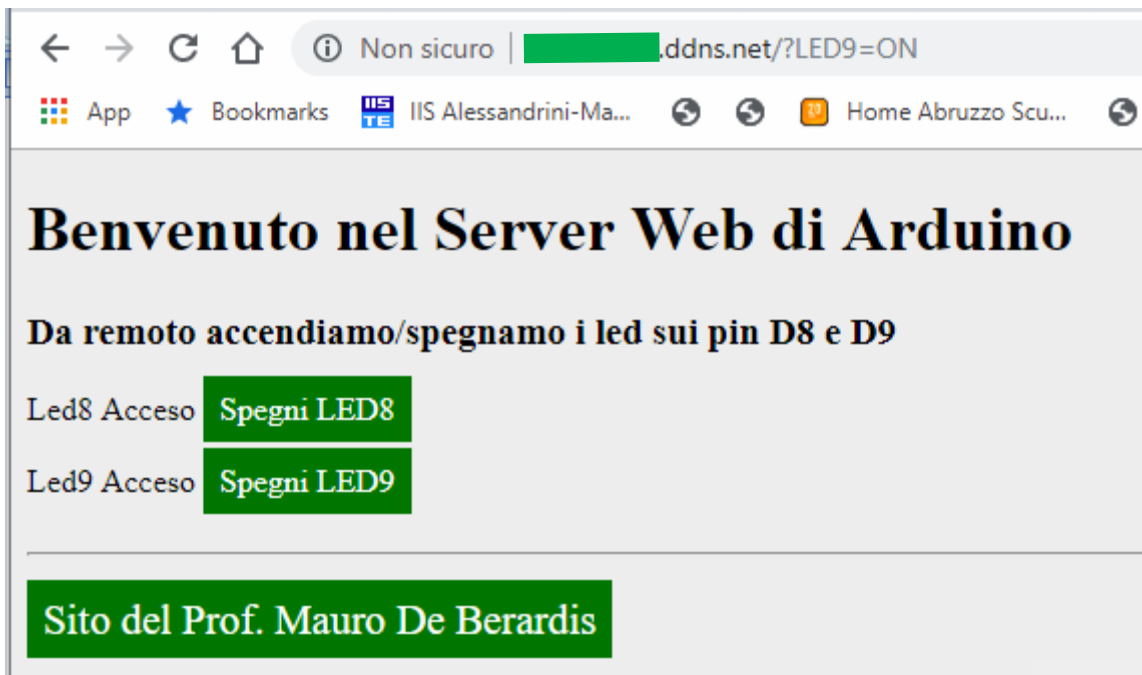
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<TITLE>Arduino Accendere due led da remoto </TITLE>");
client.println("<meta charset=UTF-8>");
client.println("<link rel='stylesheet' type='text/css' href='http://www.maurodeberardis
.it/CodiciPHP/mycssarduino.css' />");
// la riga precedente che ho usato per migliorare la pagina web, si può togliere
client.println("</head>");
client.println("<body>");
client.println("<h1> Benvenuto nel Server Web di Arduino </h1>");
client.println("<h3> Da remoto accendiamo/spegniamo i led sui pin D8 e D9 </h3>");

//controllo il contenuto di StringaLetta, ovvero quel che è scritto nell'url
if(StringaLetta.indexOf("LED8=ON")>0)
{
    digitalWrite(8,HIGH);
    LED8_Acceso=true;
}
if(StringaLetta.indexOf("LED8=OFF")>0)
{
    digitalWrite(8,LOW);
    LED8_Acceso=false;
}
if(StringaLetta.indexOf("LED9=ON")>0)
{
    digitalWrite(9,HIGH);
    LED9_Acceso=true;
}
if(StringaLetta.indexOf("LED9=OFF")>0)
{
    digitalWrite(9,LOW);
    LED9_Acceso=false;
}
if(LED8_Acceso) client.println("<a href='/?LED8=OFF'>Spegni LED8</a><br/><br/>");
if(LED8_Acceso==false) client.println("<a href='/?LED8=ON'>Accendi LED8</a><br/><br/>");
if(LED9_Acceso) client.println("<a href='/?LED9=OFF'>Spegni LED9</a><br/><br/>");
if(LED9_Acceso==false) client.println("<a href='/?LED9=ON'>Accendi LED9</a><br/><br/>");
client.println("<hr />");
client.println("<p><a target='_blank' href='http://www.maurodeberardis.it'>Sito del Prof.
Mauro De Berardis</a></p>");
client.println("</body>");
client.println("</html>");
delay(5);
StringaLetta="";
// Viene chiusa la connessione con il client
client.stop();
}
}
}
}
}

```

Test del progetto





In rete locale...

← → ↻ 🏠 ⓘ Non sicuro | 192.168.1.30/?LED9=OFF

📱 App ★ Bookmarks 📄 IIS Alessandrini-Ma... 🔄 🔄 📅 Home Abruzzo Scu...

Benvenuto nel Server Web di Arduino

Da remoto accendiamo/spegnamo i led sui pin D8 e D9

Led8 Acceso **Spegni LED8**

Led9 Spento **Accendi LED9**

Sito del Prof. Mauro De Berardis

