

# Control an LED over the Internet using the Arduino Ethernet Shield

16 Apr. 2013

How to control an LED over the Internet using an Arduino and an ethernet shield

SOURCE: <http://www.instructables.com/id/Control-an-LED-over-the-internet-using-the-Arduino/>

Original CODE: <http://pastebin.com/CZ6J62qJ>

PARTS :

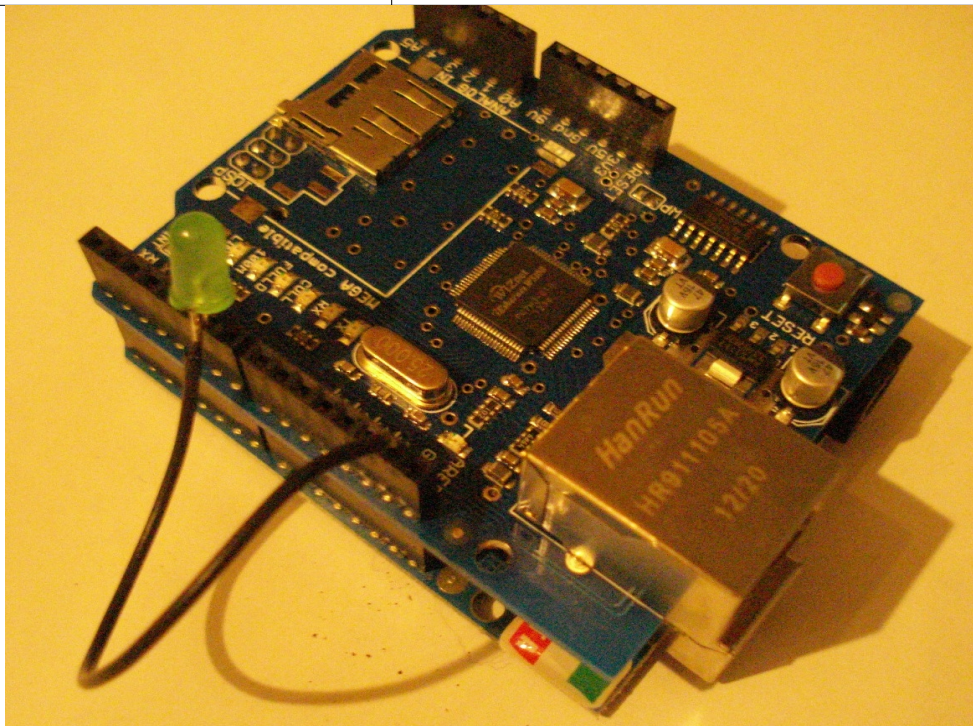
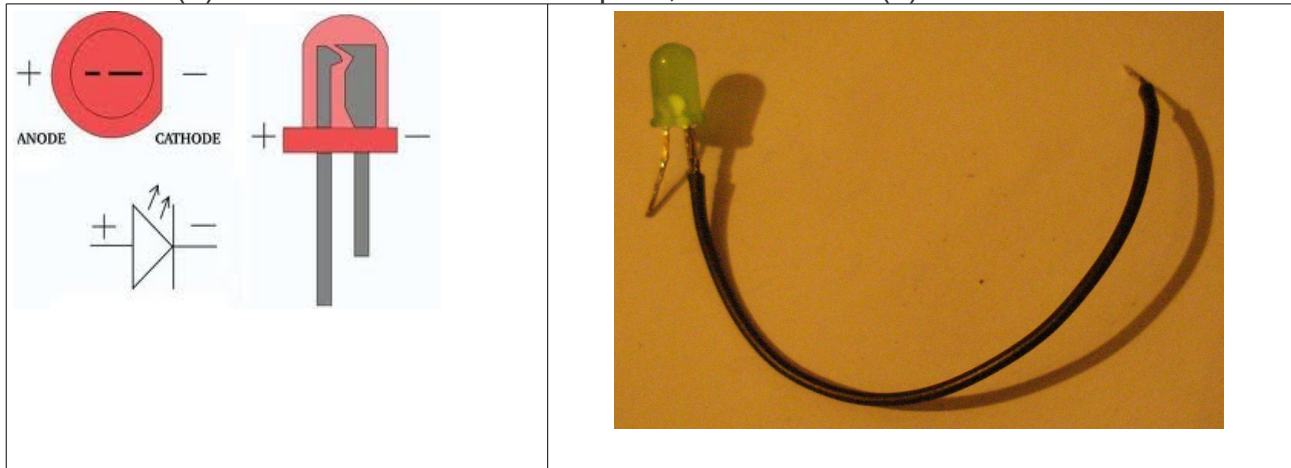
1x Arduino (I am using the Duemilanove)

1x Arduino Ethernet shield (with a W5100 chip)

1x normal LED

The circuit:

LED anode (+) to Arduino Ethernet shield pin 6; LED cathode (-) to GND



## THE CODE (MODIFICATIONS by AA, 16 APRIL 2013 )

```
// FILENAME = LED2.ino
// SOURCE= http://www.instructables.com/id/Control-an-LED-over-the-internet-using-the-Arduino/
// Control an LED over the internet using the Arduino Ethernet Shield and Transistor
// http://www.instructables.com/id/Control-an-LED-over-the-internet-using-the-Arduino/
// code SOURCE= http://pastebin.com/CZ6J62qJ

#include <SPI.h>
#include <Ethernet.h>

#include <Servo.h>
Servo myservo; // create servo object to control a servo

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //physical mac address
byte ip[] = { 192, 168, 1, 177 }; // fixed IP addr in LAN
byte gateway[] = { 192, 168, 1, 1 }; // internet access via router
byte subnet[] = { 255, 255, 255, 0 }; //subnet mask
EthernetServer server(80); //server port

String readString;
//////////
void setup(){
  pinMode(6, OUTPUT); //pin selected to control LED
  //start Ethernet
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin();
  //the pin for the servo co
  //enable serial data print
  Serial.begin(9600);
  Serial.println("server LED test 1.0 ---- FILENAME = LED2.ino "); // so I can keep track
}

void loop(){
  // Create a client connection
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();

        //read char by char HTTP request
        if (readString.length() < 100) {

          //store characters to string
          readString += c;
          //Serial.print(c);
        }

        //if HTTP request has ended
        if (c == '\n') {

          //////////
          Serial.println(readString); //print to serial monitor for debugging
          client.println("HTTP/1.1 200 OK"); //send new page
          client.println("Content-Type: text/html");
          client.println();
        }
      }
    }
  }
}
```

```

client.println("<HTML>");
client.println("<HEAD>");
client.println("<TITLE>Home Automation</TITLE>");
client.println("</HEAD>");
client.println("<BODY bgcolor='aqua'>");
client.println("<H1>Home Automation</H1>");
client.println("<hr />");
client.println("<br />");

client.println("<a href='\"/?lighton\"'>Turn On Light</a>");
client.println("<a href='\"/?lightoff\"'>Turn Off Light</a><br />");

client.println("</BODY>");
client.println("</HTML>");

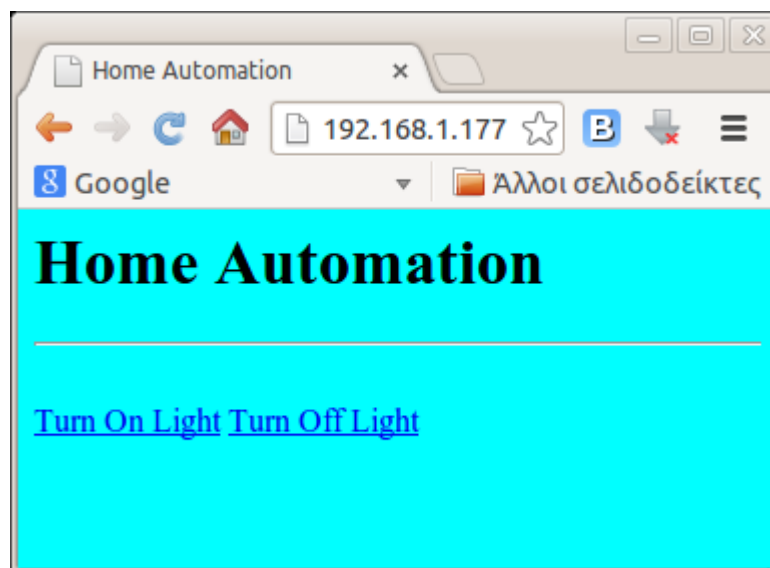
delay(1);
//stopping client
client.stop();

////////// control arduino pin
if(readString.indexOf("?lighton") >0)//checks for on
{
    digitalWrite(6, HIGH); // set pin 6 high
    Serial.println("LED On");
}
else{
if(readString.indexOf("?lightoff") >0)//checks for off
{
    digitalWrite(6, LOW); // set pin 6 low
    Serial.println("LED Off");
}
}
readString=""; //clearing string for next read
}
}
}
}
}
}
}

```

The screenshot shows the Arduino IDE interface. The top menu bar includes 'Αρχείο', 'Επεξεργασία', 'Σχέδιο', 'Εργαλεία', and 'Βοήθεια'. The toolbar contains icons for file operations and execution. The main text area shows the code from the previous block, with the filename 'sketch\_apr16a.s' at the top. The status bar at the bottom indicates 'Φόρτωση επιτυχής.' (Loading successful). On the right, the serial monitor is open, showing the output 'server LED test 1.0 ---- FILENAME = LED2.ino' and a status bar with 'Αυτόματο scroll' (Auto scroll) and 'Επιστροφή δρομέα' (Return cursor).

Next, go to the IP Address: 192.168.1.177 and turn on and off the LED!  
Open <http://192.168.1.177/> in browser:



SERIAL MONITOR SHOWS:

```
server LED test 1.0 ---- FILENAME = LED2.ino
GET / HTTP/1.1

GET /favicon.ico HTTP/1.1

GET /?lighton HTTP/1.1

LED On
GET /favicon.ico HTTP/1.1

GET /?lightoff HTTP/1.1

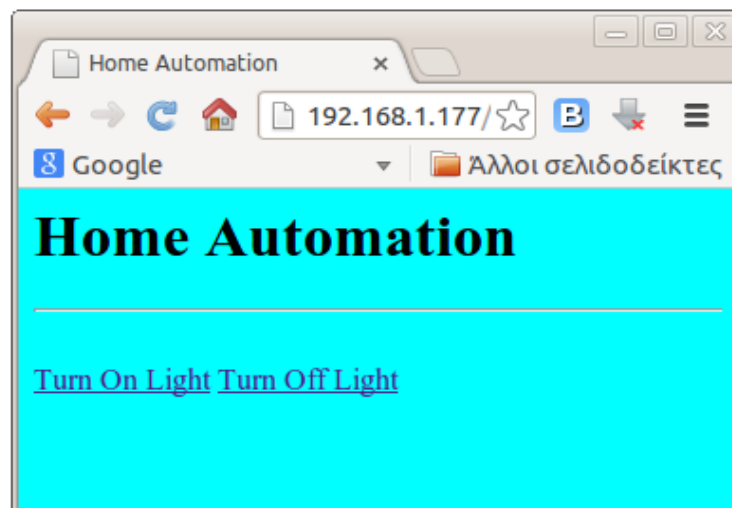
LED Off
GET /favicon.ico HTTP/1.1

GET /?lighton HTTP/1.1

LED On
GET /favicon.ico HTTP/1.1

GET /?lightoff HTTP/1.1

LED Off
GET /favicon.ico HTTP/1.1
```



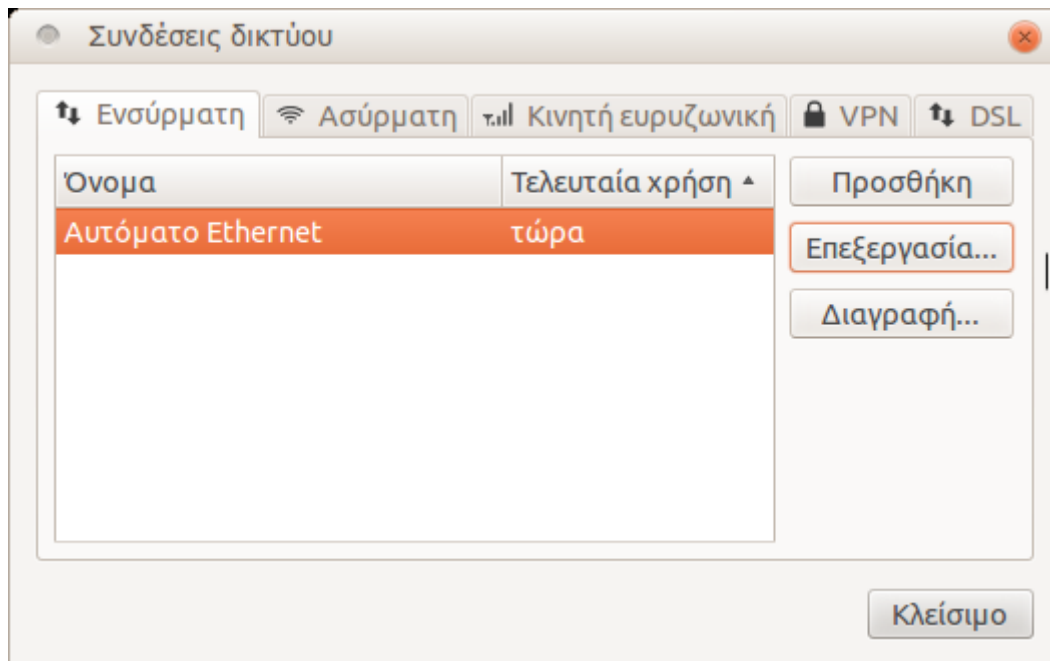
## DEMO W/O ROUTER

1/ USE CROSS-OVER ETHERNET CABLE TO CONNECT PC – ARDUINO ETHERNET SHIELD

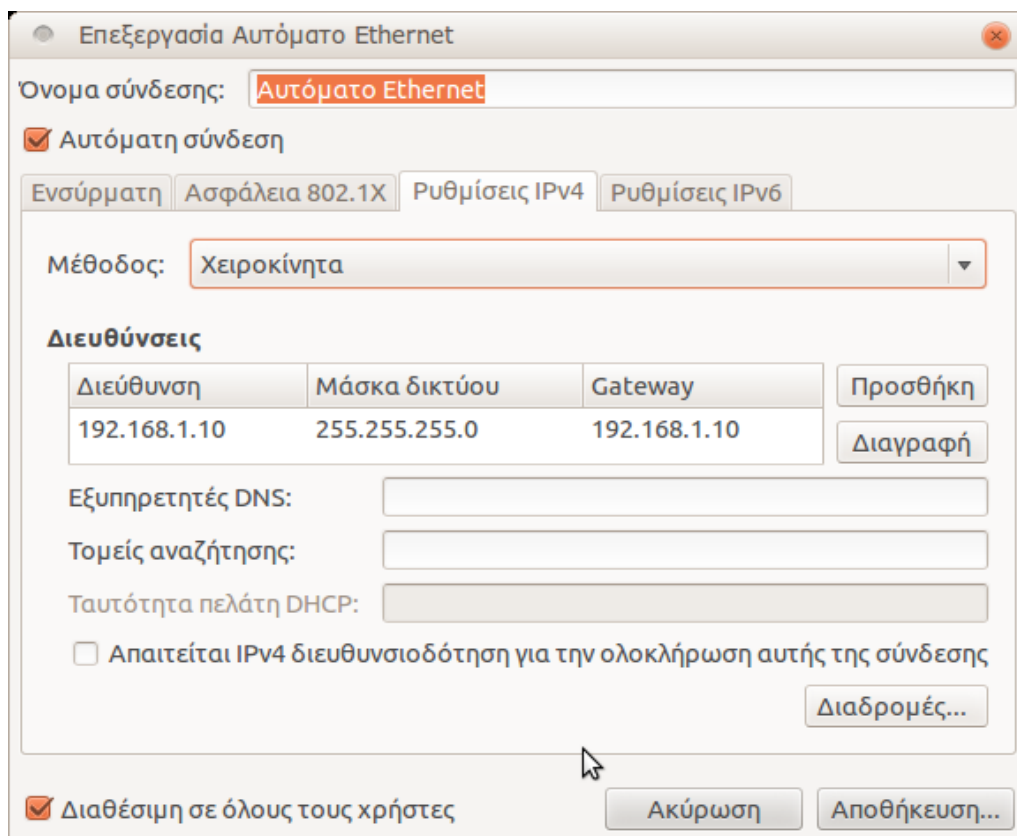
2/ SET MANUALLY PC IP ADDR

In Linux, edit connections:

Select wired connection & click on EDIT [Επεξεργασία] :



3/ Manually insert an IP address of the same subnet 192.168.1/24 (ie, 192.168.1.xxx), such as 192.168.1.10; insert subnet mask 255.255.255.0.



4/ Then check your IP addr with ifconfig.

5/ Then ping 192.168.1.177 [ARDUINO ETHERNET SHIELD]:



```
antony@DELL-N5110: ~  
Αρχείο Επεξεργασία Προβολή Αναζήτηση Τερματικό Βοήθεια  
antony@DELL-N5110:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 18:03:73:6d:78:7c  
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0  
          inet6 addr: fe80::1a03:73ff:fe6d:787c/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:54620 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:49509 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:49082886 (49.0 MB)  TX bytes:14248954 (14.2 MB)  
          Interrupt:52 Base address:0xc000  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:3060 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:3060 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:417965 (417.9 KB)  TX bytes:417965 (417.9 KB)  
  
antony@DELL-N5110:~$ ping -c 4 192.168.1.177  
PING 192.168.1.177 (192.168.1.177) 56(84) bytes of data.  
64 bytes from 192.168.1.177: icmp_req=1 ttl=128 time=0.088 ms  
64 bytes from 192.168.1.177: icmp_req=2 ttl=128 time=0.103 ms  
64 bytes from 192.168.1.177: icmp_req=3 ttl=128 time=0.091 ms  
64 bytes from 192.168.1.177: icmp_req=4 ttl=128 time=0.095 ms  
  
--- 192.168.1.177 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 2997ms  
rtt min/avg/max/mdev = 0.088/0.094/0.103/0.008 ms  
antony@DELL-N5110:~$
```

6/ Open <http://192.168.1.177/> in browser and switch LED on/off.



Downloaded from <http://t-h.wikispaces.com/arduino>

