**Home**  |  **Beginners**  |  **Projects**  |  **Tutorials**  |  **Articles**  |  **Reviews**  |

**Software** |

Search...

STARTING ELECTRONICS

Electronics for Beginners and Beyond

Blog    YouTube    Donate

# Basic Arduino Web Server

Created on: 21 January 2013

**Part 2 of the Arduino Ethernet Shield Web Server Tutorial**

A very basic web server that serves up a single web page using the Arduino Ethernet shield. An SD card is not used in this example as the web page forms part of the Arduino sketch.

## Serving Up a Web Page from the Arduino

The following Arduino sketch will enable the Arduino with Ethernet shield to serve up a single web page that can be viewed in a web browser.

```
/*------------------------------------------------
  Program:      eth_websrv_page


  Description:  Arduino web server that serves up a
                page. Does not use the SD card.


  Hardware:     Arduino Uno and official Arduino Eth
                shield. Should work with other Ardui
```

**Donate to Starting Electronics**

Donate

VISA

**Arduino Ethernet Shield Tutorial**

**Tutorials**

**Arduino**

**Part 1: Ethernet Shield Tutorial Introduction and Hardware**

**Part 2: Basic Arduino Web Server**

**Part 3: HTML Web Page Structure**

**Important Note!**

If an uninitialized SD card is left in the SD card socket of the shield, it can cause problems with code in the sketch that is accessing the Ethernet chip. This may cause symptoms such as the sketch running once or twice, then hanging up.

This is because both the Ethernet chip and the SD card are accessed by the Arduino using the same SPI bus.

If the SD card is not being used with an Ethernet application, either remove it from the socket or add the following code to disable the SD card:

```
void setup()
{
    // disable the SD card by switching pin 4 high
    // not using the SD card in this program, but i:
    // it may cause a problem with accessing the Et
    pinMode(4, OUTPUT);
    digitalWrite(4, HIGH);

    // other initialization code goes here...
}
```

⟨      ▭      ⟩

## Using the Sketch

Copy the above sketch and paste it into the Arduino IDE. Change the MAC address in the sketch to match the numbers on the sticker on the bottom of your Ethernet shield. Change the IP address in the sketch to match the IP address range of your network.

Your hardware must be set up as described in part 1 of this tutorial.

Load the sketch to the Arduino and then open a web browser on a computer that is connected to the same network as the Arduino.

Surf to the Arduino by typing the IP address of the Arduino into the URL field of the browser, e.g. 10.0.0.20 in the above sketch.

The browser should display a web page as shown below.



**Web Page Served by Arduino Web Server**

**Problem Solving**
**Resetting**

If you were not able to connect to the Arduino, try resetting it by pressing the reset button on the Ethernet shield and then surf to the web server again.

**IP Address and Address Range**
Make sure that you have set the correct Arduino IP address for the address range of your network. The first three numbers of the IP address must match your network. The last number must be unique – i.e. it must be the only device on the network with that number.

**Gateway and Subnet Mask**
Try specifying the network gateway and subnet mask in the sketch if there are still network connection problems. You will need to change the addresses in the code below to match your network.

Add the gateway and subnet under the MAC address in the sketch:

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// the router's gateway address:
byte gateway[] = { 10, 0, 0, 1 };
// the subnet:
byte subnet[] = { 255, 255, 0, 0 };
```
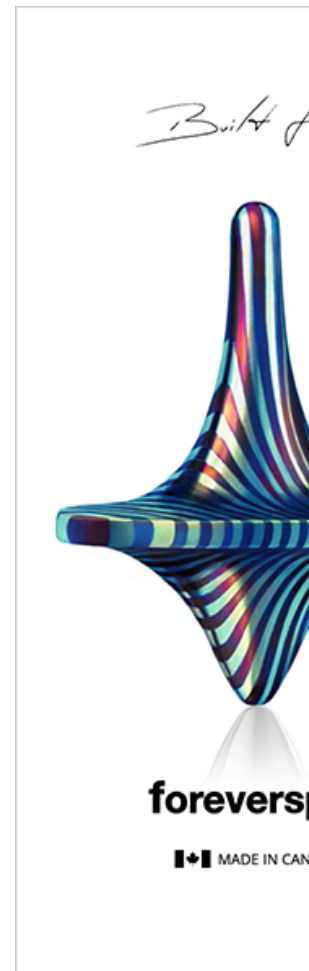
And then initialize the Ethernet device with these settings in the **setup()** part of the sketch:

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

**Ethernet Cable**
When connecting to the network through an Ethernet router/hub/switch, an Ethernet cable that is wired one-to-one must be used to connect the Arduino. Do not use a crossover cable.

## Basic Web Server Explained

Read the comments in the above sketch to see what specific lines of code do. This explanation shows what request the server must respond to and what data it must send back.

### Client Request

When you surf to the IP address of the Arduino server, the web browser (client) will send a request, such as the one shown below, to the server.

```
GET / HTTP/1.1\r\n
Host: 10.0.0.20\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:1
Accept: text/html,application/xhtml+xml,application/xm
Accept-Language: en-ZA,en-GB;q=0.8,en-US;q=0.5,en;q=0.
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
\r\n
```

The information in the request will differ, depending on the browser and operating system that the request is sent from.

The **\r\n** characters that you see at the end of every line of text in the request are non-visible characters (non-printable characters). **\r** is the carriage return character and **\n** is the linefeed character (or newline character).

The last line of the request is simply **\r\n** without and preceding text. This is the blank line that the Arduino sketch checks for before sending a response to the client web browser.

In other words, the sketch reads every character from the above request and knows when the end of the request has been reached because it finds the blank line.

### Server Response

After receiving the request for a web page from the client, the server first sends a standard HTTP response and then the web page itself.

The response sent from the Arduino is as follows:

```
HTTP/1.1 200 OK\r\n
Content-Type: text/html\r\n
Connection: close\r\n
\r\n
```

Again the non-visible characters **\r\n** are shown in the above response. The **println()** function in the the sketch automatically adds the **\r\n** characters to the end of each line. The empty **println()** function at the end of the HTTP response simply sends the **\r\n** with no text in front of it.

The above request and response are part of HTTP (Hypertext Transfer Protocol).

## Web Page

After the server has sent the HTTP response, it sends the actual web page which is then displayed in the browser.

The web page consists of text with HTML tags. You do not see the tags in the browser as these tags are interpreted by the browser.

To see the actual HTML source code, in the browser right-click on the page from the Arduino server and then click **View Page Source**.

The actual HTML markup tags are shown below.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Arduino Web Page</title>
5  </head>
6  <body>
7  <h1>Hello from Arduino!</h1>
8  <p>A web page from the Arduino server</p>
9  </body>
10 </html>
11
```

**Web Page HTML Code (Markup)**

HTML and other web page code is explained in the next part of this tutorial.

← Go back to Part 1                    Go to Part 3 →

## Comments                                                Login

There are no comments posted yet. Be the first one!

## Post a new comment

Enter text right here!

Comment as a Guest, or login:

| Name | Email | Website (optional) |
|---|---|---|
| | | |
| Displayed next to your comments. | Not displayed publicly. | If you have a website, link to it here. |

Subscribe to  None ▾                    **Submit Comment**

Arduino  |  Pinout  |  About  |  Contact  |  Donate  |  Privacy Policy  |
Amazon Adverts  |

© 2012 – 2021, Starting Electronics