Arduino Web Server LED Control

Created on: 29 January 2013

Part 5 of the Arduino Ethernet Shield Web Server Tutorial

In this part of the tutorial, the Arduino and Ethernet shield serves up a web page that allows an LED to be switched on and off. The LED is connected to one of the Arduino pins – this simple circuit can be built on a breadboard.

This video shows the LED being controlled from the web page:

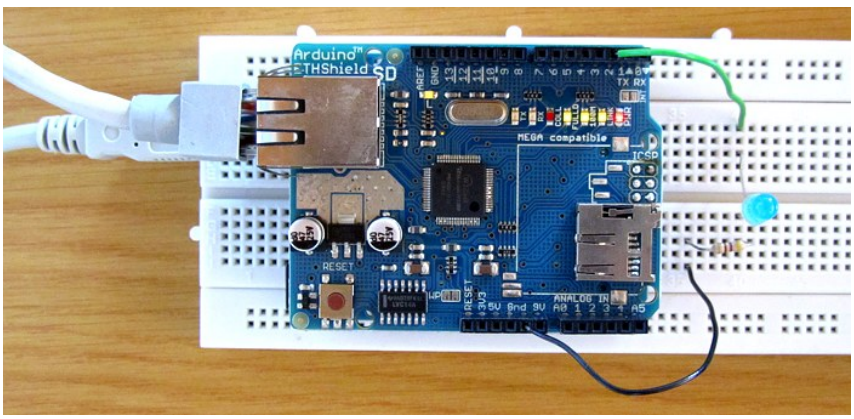
Can't see the video? View on YouTube →

Arduino Web Server LED Controller Hardware

The LED is interfaced to the Arduino as shown in the circuit diagram in the Starting with Arduino tutorial. It is simply an LED and series resistor connected between Arduino pin 2 and GND.

An SD card is not used in this web server.

The hardware is shown in the image below.



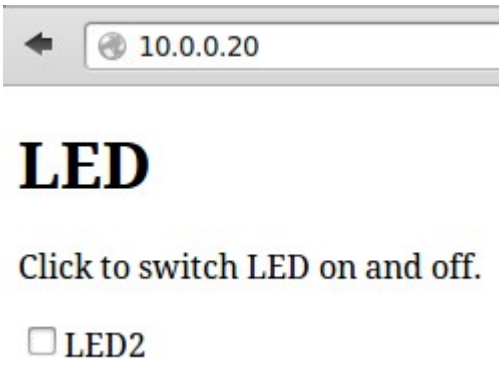
LED Web Server Hardware

How the LED is Controlled

Web Page and HTML

Web Page and HTML Code with Checkbox Unchecked

The Arduino web server serves up a page that allows the user to click a check box to switch the LED on and off. The web page is shown here:



LED Web Server Web Page - Checkbox Unchecked

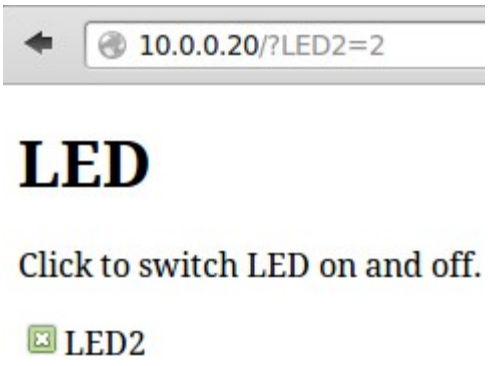
The HTML code that the Arduino web server sends to the web browser is shown below.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino LED Control</title>
    </head>
    <body>
        <h1>LED</h1>
        <p>Click to switch LED on and off.</p>
        <form method="get">
            <input type="checkbox" name="LED2" value="2" onclick="submit();">LED2
        </form>
    </body>
</html>
```

LED Web Server Web Page HTML Code - Checkbox Unchecked

Web Page and HTML Code with Checkbox Checked

After clicking the checkbox to switch the LED on, the web page and HTML code now look as follows:



LED Web Page with Checkbox Checked

Take note in the above image that the web browser added **/?LED2=2** to the end of the URL field after the checkbox was clicked.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino LED Control</title>
    </head>
    <body>
        <h1>LED</h1>
        <p>Click to switch LED on and off.</p>
        <form method="get">
            <input type="checkbox" name="LED2" value="2" onclick="submit();" checked>LED2
        </form>
    </body>
</html>
```

LED Web Page HTML Code with Checkbox Checked

In the above image, the Arduino changed the HTML page that it sent to the browser so that the checkbox will be shown with a check mark in it. The change to the code is highlighted in the image and it can be seen that **checked** was added.

New HTML Tags

Two new HTML tags are introduced in the above HTML code, namely **<form>** and **<input>**.

HTML <form> Tag

A form tag contains form controls, such as the checkbox used in this example. In this form, **method="get"** in the opening form tag will result in the form being submitted using an HTTP GET request. This also results in the **/?LED2=2** text being added in the URL field of the web browser.

HTML <input> Tag

A single control is added to the HTML form using the **<input>** tag. The input tag does not have a corresponding closing tag.

In this example, the input tag is used to create a checkbox. The following fields are included in the input tag:

- **type="checkbox"** – displays this input control as a checkbox
- **name="LED2"** – user defined name of the control
- **value="2"** – user defined value of the control
- **onclick="submit();"** – submit the form when the checkbox control is clicked
- **checked** – when present the checkbox is checked, otherwise it is blank

HTTP Request and Response

When the checkbox is clicked, it will generate an HTTP GET request that sends the name and value from the checkbox to the Arduino server.

The following is an example of an HTTP request sent from the Firefox browser to the Arduino server after clicking the checkbox:

```
GET /?LED2=2 HTTP/1.1
Host: 10.0.0.20
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
Referer: http://10.0.0.20/
Connection: keep-alive
```

When unchecking the checkbox, the following HTTP request is sent from the browser to the Arduino web server:

```
GET / HTTP/1.1
Host: 10.0.0.20
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
Referer: http://10.0.0.20/?LED2=2
Connection: keep-alive
```

The Arduino sketch in this example reads the HTTP request header and checks for the text **LED2=2** and if found, the Arduino will toggle the LED from off to on or on to off.

Both of the above requests contain the **LED2=2** text although in different places. When checking the box, the text is part of the GET request line. When unchecking the box, the text is part of the **Referer:** header.

With this background information, we can now see how the Arduino sketch works.

LED Web Server Sketch

The Arduino sketch for the LED web server is shown below.

```
/*-----------------------------------------------
  Program:      eth_websrv_LED

  Description:  Arduino web server that serves up a
                allowing the user to control an LED

  Hardware:     - Arduino Uno and official Arduino E
                  shield. Should work with other Arc
                  compatible Ethernet shields.
                - LED and resistor in series connect
                  Arduino pin 2 and GND

  Software:     Developed using Arduino 1.0.3 softwa
```

Modification to Sketch

This sketch is a modified version of the **eth_websrv_page** sketch from the basic Arduino web server.

The sketch creates the HTML page as usual, but calls the **ProcessCheckbox()** function to take care of the line that draws the checkbox.

The **ProcessCheckbox()** function checks to see if the HTTP request contains the text **LED2=2**. If the HTTP request does contain this text, then the LED will be toggled (switched from on to off or from off to on) and the web page is sent again with the checkbox control also toggled to reflect the state of the LED.

Improvements

The sketch has been kept simple for learning purposes, but some improvements can be made to this sketch to make it more reliable.

The sketch currently only checks for the presence of the text **LED2=2** in the HTTP request to see if the checkbox was clicked. It would be more reliable to check where the **LED2=2** text is in the HTTP message to determine whether the checkbox is being checked or unchecked. This would then make it impossible for the state of the LED and the state of the checkbox to become unsynchronized.

← Go back to Part 4 Go to Part 6 →

Comments (3)

Login

Sort by: **Date** Rating Last Activity

schel4ok · *20 weeks ago* 0

I can switch LED ON, but then it is not possible to switch it OFF. When I have input checked and click on it, then it becomes unchecked for a second and then page is refreshing and it is again becomes checked.

Reply

schel4ok · *20 weeks ago* 0

some more detail
When I load LED web-page for a first time after arduino restart I see checkbox unchecked and 2 requests in serial monitor
1 - GET / HTTP/1.1
2 - GET /favicon.ico HTTP/1.1
Referer: http://192.168.20.10/

When I click on it it becomes checked and LED is light just for a second and then immediately switch off itself, though in browser url I still see http://192.168.20.10/?LED2=2
In serial monitor I see 2 requests
1 - GET /?LED2=2 HTTP/1.1
Referer: http://192.168.20.10/
2 - GET /favicon.ico HTTP/1.1
Referer: http://192.168.20.10/?LED2=2

And on the page I see checkbox is checked.

Then I try to uncheck it and browser refresh the page
In browser url I see http://192.168.20.10/?
In serial monitor I see 2 requests
GET /? HTTP/1.1
Referer: http://192.168.20.10/?LED2=2
GET /favicon.ico HTTP/1.1
Referer: http://192.168.20.10/?

Then LED is always light. Every next click on checkbox doesn't change the state of LED. In browser url I always see http://192.168.20.10/?
Even if I delete ? or write something instead of it it doesnt change anything.

Reply

francis bacon · *14 weeks ago* 0

is there anyway to do this with the processcheckbox() function when reading from a SD card?

Reply

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