Arduino Web Server Switch Status Using AJAX Manually

Created on: 4 February 2013

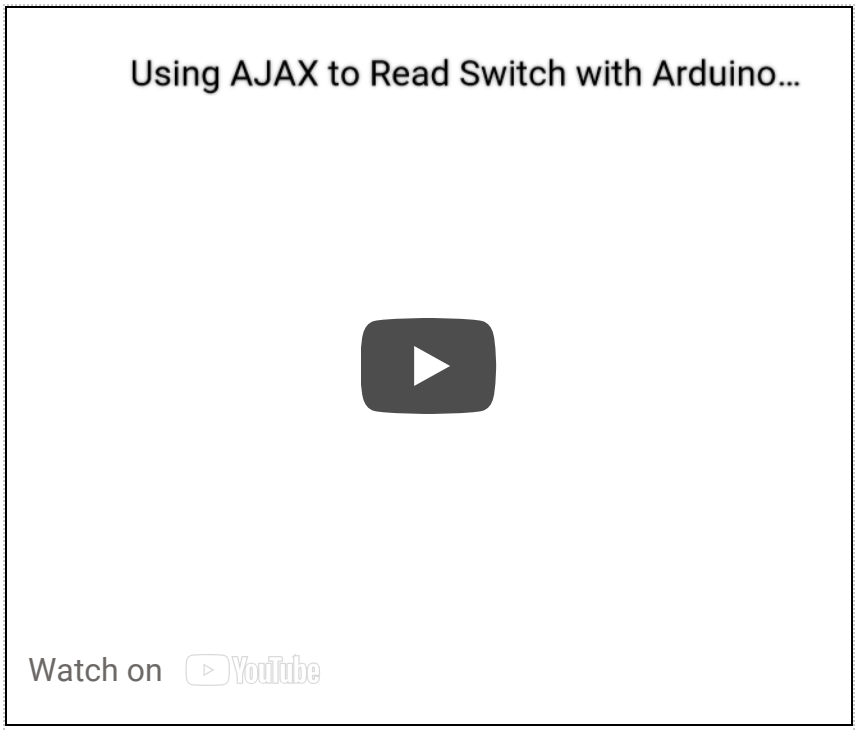
Part 7 of the Arduino Ethernet Shield Web Server Tutorial

The state of a switch connected to the Arduino / Ethernet shield is shown on a web page that is hosted by the Arduino. AJAX is used to fetch the state of the switch when a button on the web page is clicked.

The reason for using a button on the web page to refresh the state of the

switch is to keep the code simple for those who are new to AJAX. The next part of this series will automate the reading of the switch using AJAX for a more practical application.

This video shows the Arduino web server displaying the switch status using AJAX.



Can't see the video? View on YouTube →

What is AJAX?

AJAX stands for Asynchronous JavaScript and XML.

AJAX is basically the use of a set of JavaScript functions for getting information from the web server (our Arduino). This means that data on a web page can be updated without fetching the whole page each time.

Using AJAX will be an improvement on the previous part of this tutorial as HTML refresh code that makes the page flicker each time it is reloaded is no longer needed. Only the information that has changed (the state of the switch) will be updated on the page eliminating the flicker.

What is JavaScript?

JavaScript is a client side scripting language. This means that it is code that will run on the web browser.

JavaScript is included in the HTML page. When you surf to the HTML web page hosted by the Arduino, the page and the JavaScript is loaded to your browser. Your browser then runs the JavaScript code (provided that you have not disabled JavaScript in your browser).

Web Server Hardware

The switch is interfaced to the Arduino / Ethernet shield as done in the circuit diagram from this article: Project 4: Switch a LED on when Switch is Closed (Button) except that **the switch is connected to pin 3 and not pin 2 of the Arduino** (the article actually uses the circuit diagram from one of the Arduino examples on the Arduino website).

Arduino AJAX Sketch

The sketch for this part of the tutorial is shown below. Copy it and paste it into your Arduino IDE and then load it to the Arduino.

```
/*-----------------------------------------------
  Program:      eth_websrv_AJAX_switch

  Description:  Arduino web server shows the state o
                on a web page using AJAX. The state
                switch must be read by clicking a bu
                the web page - for demonstrating AJA
                Does not use the SD card.

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

Summary and Conclusion

HTML and JavaScript

The above sketch will send the following HTML and JavaScript to the web browser.

```
<!DOCTYPE html>
<html>
<head>
    <title>Arduino Web Page</title>
    <script>
        function GetSwitchState()
        {
            nocache = "&nocache=" + Math.random() * 1000000;
            var request = new XMLHttpRequest();
            request.onreadystatechange = function()
            {
                if (this.readyState == 4) {
                    if (this.status == 200) {
                        if (this.responseText != null) {
                            document.getElementById("switch_txt").innerHTML = this.responseText;
                        }
                    }
                }
            }
            request.open("GET", "ajax_switch" + nocache, true);
            request.send(null);
        }
    </script>
</head>
<body>
    <h1>Arduino AJAX Switch Status</h1>
    <p id="switch_txt">Switch state: Not requested...</p>
    <button type="button" onclick="GetSwitchState()">Get Switch State</button>
</body>
</html>
```

HTML and JavaScript Hosted by the Arduino

Page Structure

In the <head> part of the HTML code, a JavaScript function can be found between the opening and closing <script> tags.

Whenever the button on the web page is clicked, the **GetSwitchState()** JavaScript function is called.

JavaScript Function

When the web page button is clicked and the **GetSwitchState()** function is called, it sends a HTTP GET

request to the Arduino that contains the text "ajax_switch". This request looks as follows:

```
GET /ajax_switch&nocache=29860.903564600583 HTTP/1.1
Host: 10.0.0.20
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv
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 the Arduino receives this request (containing the ajax_switch text), it responds with a standard HTTP response header followed by text that contains the state of the switch.

In the Arduino code, the function **GetSwitchState()** will read the switch state on the Arduino pin and send the text **Switch state: ON** or **Switch state: OFF**.

When the JavaScript in the browser receives this response, it runs the code in the unnamed function **request.onreadystatechange = function()**. This function runs every time that the Arduino sends a response to the browser. It replaces the **Switch state: x** text on the web page (or the default text **Switch state: Not requested...**) with the new text received from the Arduino.

This JavaScript request from the browser and response from the Arduino is AJAX in action.

Random Number

The JavaScript function is based on the code from the book Learning PHP, MySQL, and JavaScript: A Step-By-Step Guide to Creating Dynamic Websites (Animal Guide), page 385 (2009 edition) – this book has been superseded by Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites (paperback) or

Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites (Kindle).

In the book, the author explains that when using a GET request, the web browser may cache the GET request. This means that the first request will work fine, but subsequent requests will fail as the first response to the request is cached and will be displayed by the browser every time.

The introduction of a random number in the request fixes the caching problem. Out of five different browsers (Firefox, IE, Safari, Chromium, Chrome) it was found that only Internet Explorer cached the request, so the random number has been added to fix the problem in IE.

AJAX Summarized

The AJAX operation performed in this example can be summarized as follows:

1. AJAX Request from Browser

When the button on the web page is clicked, the JavaScript function **GetSwitchState()** is run. This function does the following:

1. Generates a random number to send with the GET request: **nocache = "&nocache=" + Math.random() * 1000000;**
2. Creates a **XMLHttpRequest()** object called **request**: **var request = new XMLHttpRequest();**
3. Assigns a function to handle the response from the web server: **request.onreadystatechange = function()** (and following code between braces { }).
4. Sets up a HTTP GET request to send to the web server: **request.open("GET", "ajax_switch" + nocache, true);**
5. Sends the HTTP request: **request.send(null);**

2. Response from Arduino Web Server

When the Arduino web server receives the HTTP GET request, it sends back a standard HTTP response followed by text that represents the state of the switch. The state of

the switch and the text sent is obtained from the Arduino's own **GetSwitchState()** function.

3. Browser JavaScript Handles Response

The HTTP response from the Arduino web server is handled by the JavaScript code. The JavaScript event handler function runs when the response from the Arduino is received (the event handler function is the unnamed function assigned to **request.onreadystatechange**).

If the received response is OK and not empty, then this line of JavaScript is run:

```
document.getElementById("switch_txt").innerHTML = this
```

This JavaScript finds the paragraph in the HTML that is marked with the ID **switch_txt** and replaces the current text with the text received from the Arduino. The HTML for this paragraph looks as follows:

```
<p id="switch_txt">Switch state: Not requested...</p>
```

This example has illustrated the use of AJAX used to update a single paragraph of text in the browser. The next part of this tutorial will automate the AJAX request so that a button does not have to be clicked to initiate the request.

← Go back to Part 6 Go to Part 8 →