Project – Ardunio Web Server

Team members –

Shrey Agarwal,
Himanshu Jaiswal,
Anurag Prabhakar,
Ankit Nagar
(Team- 'AURORA')

Mentor – Anurag Dwivedi

Main Aim –

To use an arduino platform as a webserver and access files stored in a SD card on a browser via Ethernet using arduino uno board and Ethernet shield.

How web servers work??

Whenever we type an url (http://www.xyz.com/abc) we get a web page, how??

Web Browser breaks the url into three parts:
1) the protocol (“http”) - predefined method of sending request to and recieving response from the server.
2)server name (“www.xyz.com”) - ip address of the server containing the required file.
3)file name(“abc”) - the file that we want(generally an html file).

The browser communicated with a name server to translate the server name "www.xyz.com" into an **IP Address**, which it uses to connect to the server machine. The browser then formed a connection to the server at that IP address on port 80 (by default).

Any server machine makes its services available to the Internet using numbered **ports**, one for each service that is available on the server.

IP address is a 32-bit number comprised of four octets separated by dots.Each octet range from 0-255,example(172.24.33.55).Every machine on the Internet has a unique IP address.

The browser communicates with a name server to translate the server name, “www.xyz.com”, into an **IP address**, which it uses to connect to that server machine. The browser then forms a connection to the Web server at that IP address on port 80. Following the HTTP protocol, the browser sends a GET request to the server, asking for the file ,“abc”.The server sends the HTML text for the Web page to the browser. The browser reads the HTML tags and formats the page onto your screen.

HTTP Message Format

The format of the request and response messages are similar. Both consists of:
1)an initial line,
2)zero or more header lines(Header lines provide information about the request or response, or about the object sent in the message body)
3)a blank line,
4)an optional message body.

Initial request line:

The initial line is different for the request than for the response. A request line has three parts, separated by spaces: a *method* name, the local path of the requested resource, and the version of HTTP being used. A typical request line is:

```
GET /path/to/file/index.html HTTP/1.0
```

Initial response line:

The initial response line, called the *status line*, also has three parts separated by spaces: the HTTP version, a *response status code* that gives the result of the request, and an English *reason phrase* describing the status code. Typical status lines are:

```
HTTP/1.0 200 OK
```

or

```
HTTP/1.0 404 Not Found
```

Sample HTTP Exchange

To retrieve the file at the URL

```
http://www.somehost.com/path/file.html
```

first open a socket to the host **www.somehost.com**, port 80 (use the default port of 80 because none is specified in the URL). Then, send something like the following through the socket:

```
GET /path/file.html HTTP/1.0
From: someuser@jmarshall.com
User-Agent: HTTPTool/1.0
[blank line here]
```

```
The server should respond with something like the following, sent back through the same socket:
```

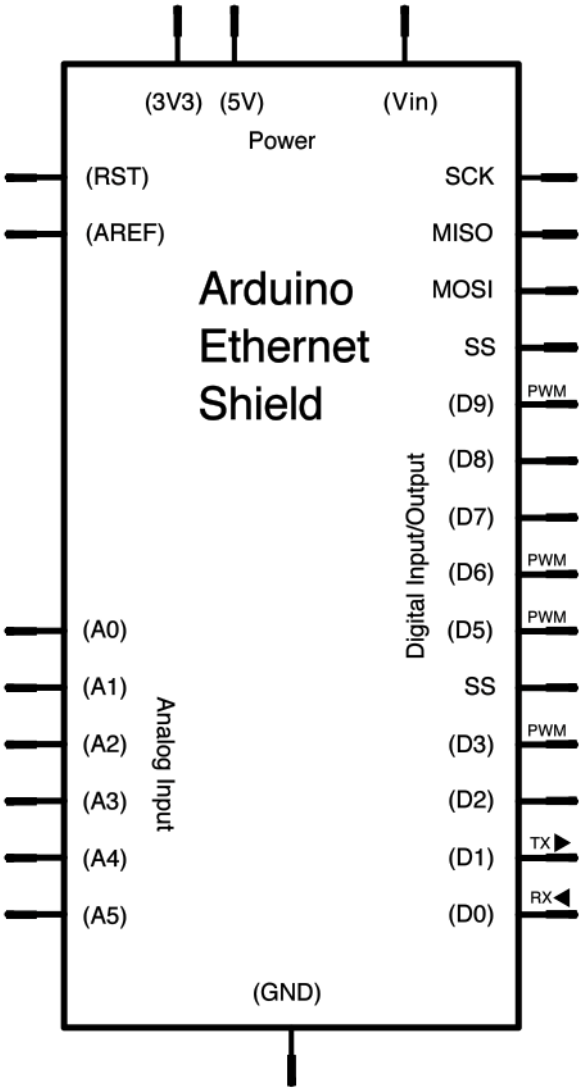
```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354

<html>
<body>
<h1>Happy New Millennium!</h1>
(more file contents)
  .
  .
  .
</body>
</html>
```

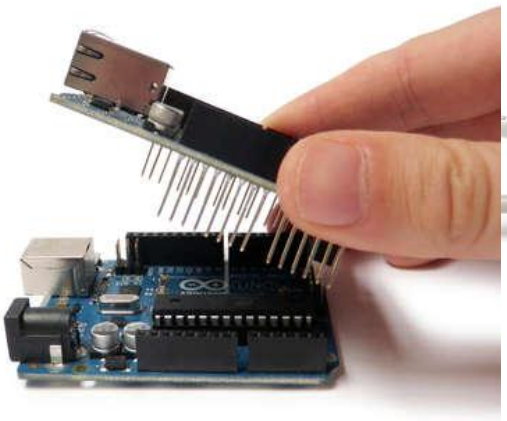
After sending the response, the server closes the socket.

Hardware configuration:

1)Arduino Ethernet Shield

2)Shield-compatible Arduino board


(3V3)
(5V)
(Vin)
Power
(RST)
(AREF)
Arduino
Ethernet
Shield
SCK
MISO
MOSI
SS
(D9)
(D8)
(D7)
(D6)
(D5)
SS
(D3)
(D2)
(D1)
(D0)
PWM
PWM
PWM
TX
RX
Digital Input/Output
(A0)
(A1)
(A2)
(A3)
(A4)
(A5)
Analog Input
(GND)



Circuit:

The Ethernet shield allows you to connect a WizNet Ethernet controller to the Arduino via the SPI bus. It uses pins 10, 11, 12, and 13 for the SPI connection to the WizNet. Ethernet shield also have an SD Card on board. Digital pin 4 is used to control the slave select pin on the SD card.

Finally we connected lcd and leds to the ardunio ethernet shield as the image below:

Software configuration:

CODING: We have done coding on ardunio environment,

we have used functions from SPI, ETHERNET, LIQUID CRYSTAL,SDFAT and SDFATUTIL libraries.

For more information on SPI,ETHERNET,LIQUID CRYSTAL library visit
http://arduino.cc/en/Tutorial/HomePage

For more information on SDFAT/SDFATUTIL library visit

http://code.google.com/p/sdfatlib/downloads/list

http://arconlab.com/lab/Arduino/Library/SD%20Reader%20-%20Fat32/html/_sd_fat_util_8h.html

http://arduino.cc/en/Reference/SD/

CENTRAL IDEA:

The main idea of the code is to interpret

and extract important information from http request and perform required operations using functions from different libraries.We have used firebug software to see what http request is recieved by server.First of all we are storing http request as a string(clientline) and then we are checking for substring “GET / “ (see the space after /)using strstr function,if this is so this means that there is no particular file name(according to http request format) so we list all the

files present on sd card using LIST function(created by using functions from SDFATUTIL library).

If this is not the case then we look for substring “GET /”(no space after /),this means that there is a file name after / and so we have to extract filename,for this we point the filename pointer(declared initially) after five characters of

clientline “filename = clientline + 5” and remove the last part of the request “(strstr(clientline, " HTTP"))[0] = 0”,so we get the file name and then we check the file format(text,html,jpg etc.) using strstr function and open it accordingly and if file is not found then we print error.

Finally there is an HTML file in our sd card which create a web page through which we can again send request to the server and by interpreting the request we can control ardunio.We are writing text on lcd and controlling leds through ardunio according to the request sent.For this also we used firebug to see what request is sent, where colour,status of led and text of lcd occurs.As before we extract important information from the request,we display text on lcd using LIQUID CRYSTAL library functions and control leds.

UTILITY

The major use of the system is for small scale web developers who want to launch their sites on a local area network and share it with a particular bunch of people. It is useful for spreading files like notes, assignments, ebooks, videos etc. on a network. Furthermore, it can be used to get different types of forms and other documents filled online(like google docs) and their information stored in a micro SD card.

A WORD OF THANKS

We would specially like to thank our team mentor Anurag Dwivedi, Other Club Coordinators Rudra Pratap Suman and Nikhil Gupta to contribute required help and guide given by them and making summers full of masti and learning experience.