

Examples > WiFi Library

WiFi Web Server

In this example, you will use your WiFi Shield and your Arduino to create a simple Web server. Using the WiFi library, your device will be able to answer a HTTP request with your WiFi shield. After opening a browser and navigating to your WiFi shield's IP address, your Arduino will respond with just enough HTML for a browser to display the input values from all six analog pins.

This example is written for a network using WPA encryption. For WEP or WPA, change the `Wifi.begin()` call accordingly.

Hardware Required

- Arduino WiFi Shield
- Shield-compatible Arduino board
- (optional) Six analog sensors attached to Analog in Pins 0-5

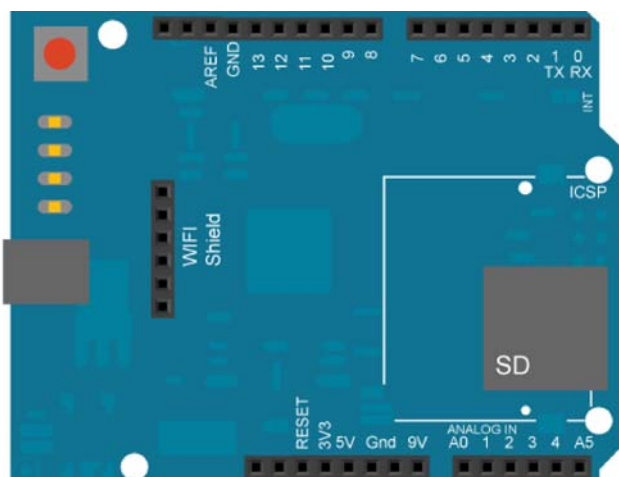
Circuit

The WiFi shield uses pins 10, 11, 12, and 13 for the SPI connection to the HDG104 module. Digital pin 4 is used to control the slave select pin on the SD card.

You should have access to a 802.11b/g wireless network that connects to the internet for this example. You will need to change the network settings in the sketch to correspond to your particular networks SSID.

For networks using WPA/WPA2 Personal encryption, you need the SSID and password. The shield will not connect to networks using WPA2 Enterprise encryption.

WEP network passwords are hexadecimal strings known as keys. A WEP network can have 4 different keys; each key is assigned a "Key Index" value. For WEP encrypted networks, you need the SSID, the key, and key number.



(http://www.arduino.cc/en/uploads/Tutorial/WiFiShield_bb.png)

In the above image, the Arduino would be stacked below the WiFi shield.

Warning

This example doesn't require an SD card. If an SD card is inserted but not used, it is possible for the sketch to hang, because pin 4 is used as SS (active low) of the SD and when not used it is configured as INPUT by default. Two possible solutions:

- remove the SD card;
- add these lines of code in the setup()

```
pinMode(4, OUTPUT);  
digitalWrite(4, HIGH);
```

[Get Code] ([//www.arduino.cc/en/Tutorial/WiFiWebServer?action=sourceblock&num=2](http://www.arduino.cc/en/Tutorial/WiFiWebServer?action=sourceblock&num=2))

Code:

```
/*  
  WiFi Web Server  
  
  A simple web server that shows the value of the analog input pins.  
  using a WiFi shield.  
  
  This example is written for a network using WPA encryption. For  
  WEP or WPA, change the Wifi.begin() call accordingly.  
  
  Circuit:  
  * WiFi shield attached  
  * Analog inputs attached to pins A0 through A5 (optional)  
  
  created 13 July 2010  
  by dlf (Metodo2 srl)  
  modified 31 May 2012  
  by Tom Igoe  
  
  */  
  
#include <SPI.h>  
#include <WiFi.h>  
  
char ssid[] = "yourNetwork";    // your network SSID (name)  
char pass[] = "secretPassword"; // your network password  
int keyIndex = 0;                // your network key Index number (needed only for WEP)  
  
int status = WL_IDLE_STATUS;  
  
WiFiServer server(80);  
  
void setup() {  
  //Initialize serial and wait for port to open:  
  Serial.begin(9600);  
  while (!Serial) {  
    ; // wait for serial port to connect. Needed for Leonardo only  
  }  
  
  // check for the presence of the shield:  
  if (WiFi.status() == WL_NO_SHIELD) {  
    Serial.println("WiFi shield not present");  
    // don't continue:  
    while(true);  
  }  
  
  // attempt to connect to Wifi network:  
  while ( status != WL_CONNECTED) {  
    Serial.print("Attempting to connect to SSID: ");  
    Serial.println(ssid);  
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:  
    status = WiFi.begin(ssid, pass);  
  
    // wait 10 seconds for connection:  
    delay(10000);  
  }  
}
```

```

server.begin();
// you're connected now, so print out the status:
printWifiStatus();
}

void loop() {
    // listen for incoming clients
    WiFiClient client = server.available();
    if (client) {
        Serial.println("new client");
        // an http request ends with a blank line
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                Serial.write(c);
                // if you've gotten to the end of the line (received a newline
                // character) and the line is blank, the http request has ended,
                // so you can send a reply
                if (c == '\n' && currentLineIsBlank) {
                    // send a standard http response header
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-Type: text/html");
                    client.println("Connection: close"); // the connection will be closed after completion of the
response
                    client.println("Refresh: 5"); // refresh the page automatically every 5 sec
                    client.println();
                    client.println("<!DOCTYPE HTML>");
                    client.println("<html>");
                    // output the value of each analog input pin
                    for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
                        int sensorReading = analogRead(analogChannel);
                        client.print("analog input ");
                        client.print(analogChannel);
                        client.print(" is ");
                        client.print(sensorReading);
                        client.println("<br />");
                    }
                    client.println("</html>");
                    break;
                }
            }
            if (c == '\n') {
                // you're starting a new line
                currentLineIsBlank = true;
            }
            else if (c != '\r') {
                // you've gotten a character on the current line
                currentLineIsBlank = false;
            }
        }
        // give the web browser time to receive the data
        delay(1);

        // close the connection:
        client.stop();
        Serial.println("client disconnected");
    }
}

void printWifiStatus() {
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // print your WiFi shield's IP address:
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // print the received signal strength:
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
}

```

```
Serial.println(" dBm");  
}
```

[Get Code] (<http://www.arduino.cc/en/Tutorial/WiFiWebServer?action=sourceblock&num=1>)

See Also:

- [ConnectNoEncryption](http://www.arduino.cc/en/Tutorial/ConnectNoEncryption) (<http://www.arduino.cc/en/Tutorial/ConnectNoEncryption>) : Demonstrates how to connect to an open network
- [ConnectWithWEP](http://www.arduino.cc/en/Tutorial/ConnectWithWEP) (<http://www.arduino.cc/en/Tutorial/ConnectWithWEP>) : Demonstrates how to connect to a network that is encrypted with WEP
- [ConnectWithWPA](http://www.arduino.cc/en/Tutorial/ConnectWithWPA) (<http://www.arduino.cc/en/Tutorial/ConnectWithWPA>) : Demonstrates how to connect to a network that is encrypted with WPA2 Personal
- [ScanNetworks](http://www.arduino.cc/en/Tutorial/ScanNetworks) (<http://www.arduino.cc/en/Tutorial/ScanNetworks>) : Displays all WiFi networks in range
- [WiFiChatServer](http://www.arduino.cc/en/Tutorial/WiFiChatServer) (<http://www.arduino.cc/en/Tutorial/WiFiChatServer>) : Set up a simple chat server
- [WiFiWebClient](http://www.arduino.cc/en/Tutorial/WiFiWebClient) (<http://www.arduino.cc/en/Tutorial/WiFiWebClient>) : Connect to a remote webserver
- [WiFiWebClientRepeating](http://www.arduino.cc/en/Tutorial/WiFiWebClientRepeating) (<http://www.arduino.cc/en/Tutorial/WiFiWebClientRepeating>): Repeatedly make HTTP calls to a server
- [WiFiWebServer](http://www.arduino.cc/en/Tutorial/WiFiWebServer) (<http://www.arduino.cc/en/Tutorial/WiFiWebServer>) : Serve a webpage from the WiFi shield

Share



NEWSLETTER

Enter your email to sign up



©2017 Arduino [Copyright Notice](http://www.arduino.cc/en/Main/CopyrightNotice) (<http://www.arduino.cc/en/Main/CopyrightNotice>) [Contact us](http://www.arduino.cc/en/Main/ContactUs) (<http://www.arduino.cc/en/Main/ContactUs>)
[About us](http://www.arduino.cc/en/Main/AboutUs) (<http://www.arduino.cc/en/Main/AboutUs>) [Careers](http://www.arduino.cc/Careers) (<http://www.arduino.cc/Careers>)

<https://twitter.com/arduino>

<https://www.facebook.com/official.arduino>

<https://plus.google.com/+Arduino>

https://www.flickr.com/photos/arduino_cc

<https://youtube.com/arduinoteam>