leeassam Updated Online Weather Station Code

1 contributor

131 lines (103 sloc) | 3.5 KB

```
/*
  Arduino Bootcamp

  - Online Weather Station - Using the Arduino Wifi shield to retrieve weather from Weather Unc

  This sketch connects to the Weather Underground API to make a request
  using an Arduino Wifi shield. It shows the raw response that is returned

  created 10/30/2016
  modified 10/30/2016
  by: Lee Assam
*/


#include <SPI.h>
#include <WiFi.h>

char ssid[] = "Your network name";      //  your network SSID (name)
char pass[] = "Your network password";   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)

int status = WL_IDLE_STATUS;
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
//IPAddress server(74,125,232,128);  // numeric IP for Google (no DNS)
char server[] = "api.wunderground.com";

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
WiFiClient client;
```

```
32
33  //API information
34  String apiKey = "Your api key";
35  //US
36  String state = "IL";
37  String city = "Bloomington";
38  String weatherLocation = String(state + "/" + city);
39
40  //International
41  //String country = "Australia";
42  //String city = "Sydney";
43  //String weatherLocation = String(country +  "/" + city);
44
45  void setup() {
46    //Initialize serial and wait for port to open:
47    Serial.begin(9600);
48    while (!Serial) {
49      ; // wait for serial port to connect. Needed for native USB port only
50    }
51
52    // check for the presence of the shield:
53    if (WiFi.status() == WL_NO_SHIELD) {
54      Serial.println("WiFi shield not present");
55      // don't continue:
56      while (true);
57    }
58
59    String fv = WiFi.firmwareVersion();
60    if (fv != "1.1.0") {
61      Serial.println("Please upgrade the firmware");
62    }
63
64    // attempt to connect to Wifi network:
65    while (status != WL_CONNECTED) {
66      Serial.print("Attempting to connect to SSID: ");
67      Serial.println(ssid);
68      // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
69      status = WiFi.begin(ssid, pass);
70
71      // wait 10 seconds for connection:
72      delay(10000);
73    }
74    Serial.println("Connected to wifi");
75    printWifiStatus();
76
77    Serial.println("\nStarting connection to server...");
78    // if you get a connection, report back via serial:
79    if (client.connect(server, 80)) {
80      Serial.println("connected to server");
81      // Make a HTTP request:
82      String urlRequest = String("GET /api/" + apiKey + "/conditions/q/" + weatherLocation + ".j
83      client.println(urlRequest);
```

```
    client.println("Host: api.wunderground.com");
    client.println("User-Agent: ArduinoWiFi/1.1");
    client.println("Connection: close");
    client.println();
  }
}

void loop() {
  // if there are incoming bytes available
  // from the server, read them and print them:
  while (client.available()) {
    char c = client.read();
    Serial.write(c);
  }

  // if the server's disconnected, stop the client:
  if (!client.connected()) {
    Serial.println();
    Serial.println("disconnecting from server.");
    client.stop();

    // do nothing forevermore:
    while (true);
  }
}


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");
}
```