PART 1 - Send Arduino data to the Web (PHP/ MySQL/ D3.js)

by **apais** on June 30, 2014

Table of Contents

Intro: PART 1 - Send Arduino data to the Web (PHP/ MySQL/ D3.js)

The objective of this project was to use and Arduino to read a sensor and send the values to the internet, to be stored in a Web Server and displayed.

It consists in an Arduino Uno with an Ethernet Shield and a DHT 11 temperature / moisture sensor, acting as a Web Client. It sends POST requests with the readings to a web server running a custom Database and PHP application.

The PHP app stores the values when new POST requests are received and also serves the pages that display the information. In Part 2, i will explain the use of D3.js to dynamically show the data stored in the Database.

The Arduino it's configured to use a Dynamic IP Address, in order to solve any conflicting IP issues, and also to work easily with most home networks/routers.

This project is divided in 2 main parts:

PART 1

```
- Arduino Web client Application: reads the sensor values and sends them to the webserver.

- PHP/MySQL Application: handles the POST requests that are sent to the server and serves the pages to clients who connect
```

PART 2

```
- Data Visualization: The PHP application will use the Javascript Framework D3.js to display the values stored in the DB with graphics. It will allow
```

REQUIREMENTS

HARDWARE

1. Arduino Uno
2. Ethernet Shield (eBay clone)
3. DHT 11 sensor
4. breadboard
5. 10k Ohm resistor
6. USB cable
7. Ethernet cable
8. wires
9. piece of acrylic
10. PCB spacers

Software

- You need access to a web server (can be from a free hosting company) with capability to run PHP applications and also to create databases. (possibly cPanel with phpMyAdmin)

RESOURCES

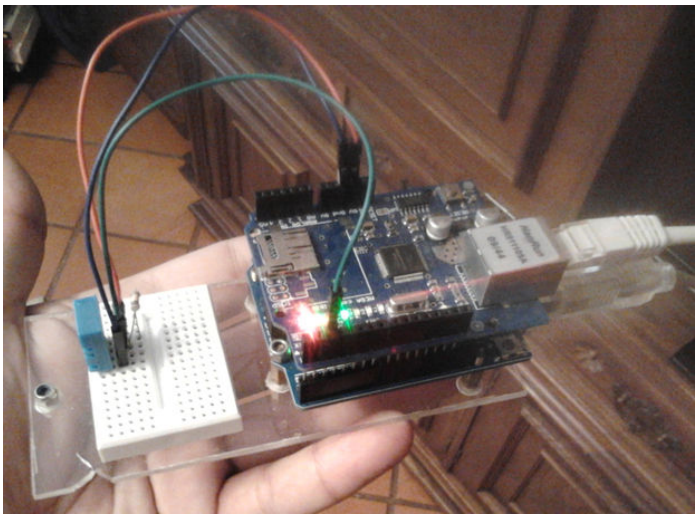
Request Maker: This online tool is very useful to test the PHP application. You can simulate the POST requests that will be made by the Arduino and check if everything is working well.

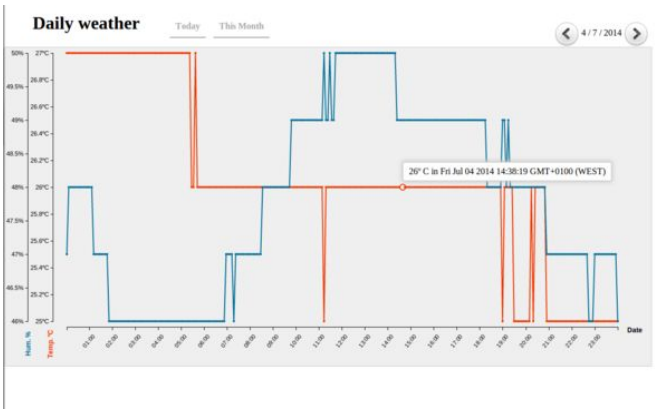
DHT11 sensor library from Adafruit

Arduino Web Client

Arduino IDE

Arduino Ethernet

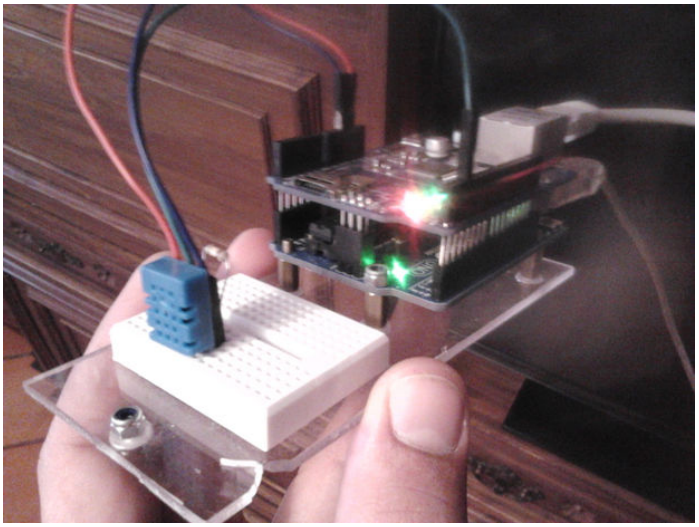


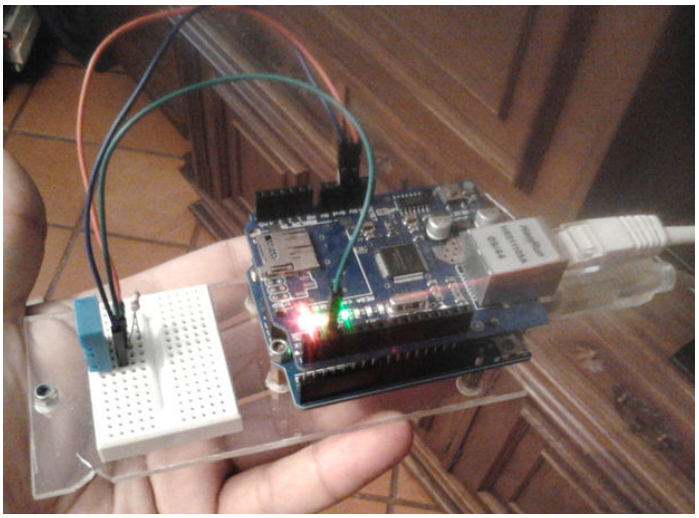


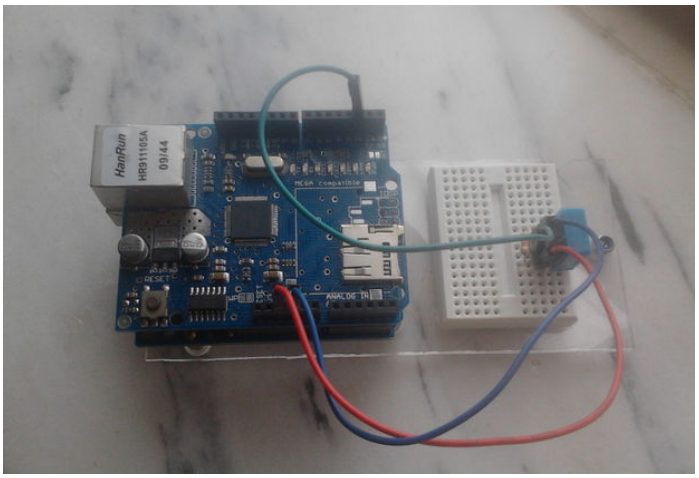
Step 1: Arduino Web client + DHT11 sensor

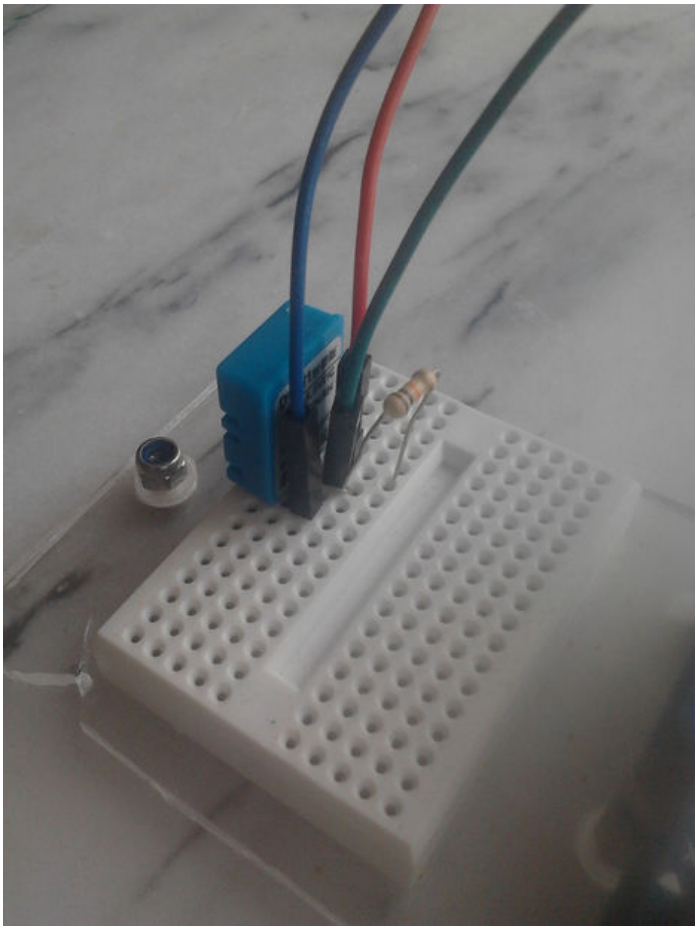
The code is very simple, all the important section are commented. If you have any doubt feel free to ask.

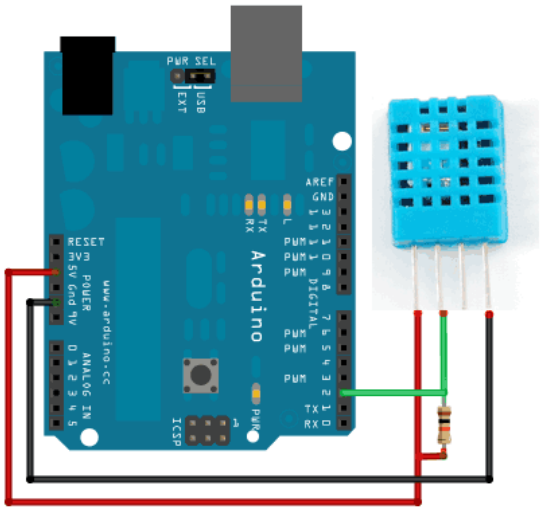
```
#include <DHT.h>
#include <Ethernet.h>
#include <SPI.h>

byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x01 }; // RESERVED MAC ADDRESS
EthernetClient client;

#define DHTPIN 2 // SENSOR PIN
#define DHTTYPE DHT11 // SENSOR TYPE - THE ADAFRUIT LIBRARY OFFERS SUPPORT FOR MORE MODELS
DHT dht(DHTPIN, DHTTYPE);

long previousMillis = 0;
unsigned long currentMillis = 0;
long interval = 250000; // READING INTERVAL

int t = 0;  // TEMPERATURE VAR
int h = 0;  // HUMIDITY VAR
String data;

void setup() {
    Serial.begin(115200);

    if (Ethernet.begin(mac) == 0) {
        Serial.println("Failed to configure Ethernet using DHCP");
    }

    dht.begin();
    delay(10000); // GIVE THE SENSOR SOME TIME TO START

    h = (int) dht.readHumidity();
    t = (int) dht.readTemperature();

    data = "";
}

void loop(){

    currentMillis = millis();
    if(currentMillis - previousMillis > interval) { // READ ONLY ONCE PER INTERVAL
        previousMillis = currentMillis;
        h = (int) dht.readHumidity();
        t = (int) dht.readTemperature();
    }

    data = "temp1=" + t + "&hum1=" + h;

    if (client.connect("www.*****.**********.com",80)) { // REPLACE WITH YOUR SERVER ADDRESS
        client.println("POST /add.php HTTP/1.1");
        client.println("Host: *****.**********.com"); // SERVER ADDRESS HERE TOO
        client.println("Content-Type: application/x-www-form-urlencoded");
        client.print("Content-Length: ");
        client.println(data.length());
        client.println();
        client.print(data);
    }

    if (client.connected()) {
        client.stop();  // DISCONNECT FROM THE SERVER
    }

    delay(300000); // WAIT FIVE MINUTES BEFORE SENDING AGAIN
}
```











File Downloads

Adafruit DHT Arduino Library.zip (2 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Adafruit DHT Arduino Library.zip']

client.ino (1 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'client.ino']

Step 2: PHP / MySQL Application

In this second part i will explain briefly the PHP application and the database. The database is used obviously to store the sensor readings, so that they can be accessed later. It's a very simple DB, with just one table with 3 columns. It stores the time stamp and the corresponding temperature and humidity values.

```
CREATE TABLE tempLog (
 timeStamp TIMESTAMP NOT NULL PRIMARY KEY,
 temperature int(11) NOT NULL,
 humidity int(11) NOT NULL,
);
```

The PHP application consists of 3 files:

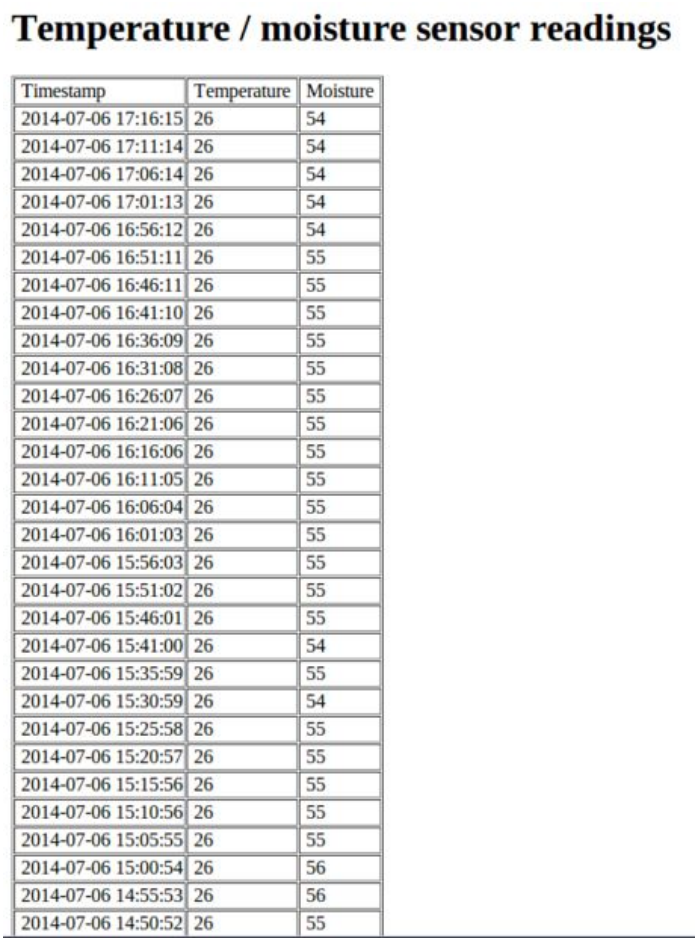
- **connect.php**: this file is loaded every time we need to access to the database. It's loaded in the beginning of the almost each file. It contains a function that returns a new connection to be used by the PHP to execute query's to the DB. You need to store the **DB configs (hostname, database, user, password)** in this file.

- **add.php**: when the Arduino sends POST requests to the server, is to to this page. The PHP receives the values sent in the request and executes an insertion query with those values.

Sometimes you need to change the permissions of this file (should be 644), because it might be protected to allow only executions from the localhost.

- **index.php**: this is the website landing page. It displays the values that are stored in the database. Right now, it will display all the values in a single HTML table, just to show that works.

So, this concludes the first part of this Instructable. Feel free to ask questions about anything related, i'm glad to help.



Temperature / moisture sensor readings

Timestamp	Temperature	Moisture
2014-07-06 17:16:15	26	54
2014-07-06 17:11:14	26	54
2014-07-06 17:06:14	26	54
2014-07-06 17:01:13	26	54
2014-07-06 16:56:12	26	54
2014-07-06 16:51:11	26	55
2014-07-06 16:46:11	26	55
2014-07-06 16:41:10	26	55
2014-07-06 16:36:09	26	55
2014-07-06 16:31:08	26	55
2014-07-06 16:26:07	26	55
2014-07-06 16:21:06	26	55
2014-07-06 16:16:06	26	55
2014-07-06 16:11:05	26	55
2014-07-06 16:06:04	26	55
2014-07-06 16:01:03	26	55
2014-07-06 15:56:03	26	55
2014-07-06 15:51:02	26	55
2014-07-06 15:46:01	26	55
2014-07-06 15:41:00	26	54
2014-07-06 15:35:59	26	55
2014-07-06 15:30:59	26	54
2014-07-06 15:25:58	26	55
2014-07-06 15:20:57	26	55
2014-07-06 15:15:56	26	55
2014-07-06 15:10:56	26	55
2014-07-06 15:05:55	26	55
2014-07-06 15:00:54	26	56
2014-07-06 14:55:53	26	56
2014-07-06 14:50:52	26	55

File Downloads

index.php (810 bytes)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'index.php']

connect.php (329 bytes)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'connect.php']

add.php (310 bytes)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'add.php']

Related Instructables

Interface Arduino to MySQL using Python by mangopeach

Arduino, Temp, Humidity, WiFi, MySQL and Highcharts by timscott22

Save data of temperature and humidity on MySQL with Arduino Uno and Wifly by camilo.n1012

Control Access of Arduino YÚN with MySQL, PHP5 and Python by camilo.n1012

myHome - home automation with Arduino and XBee by jweymarn

RFIDuino by what5150

Comments