instructables (/) let's make

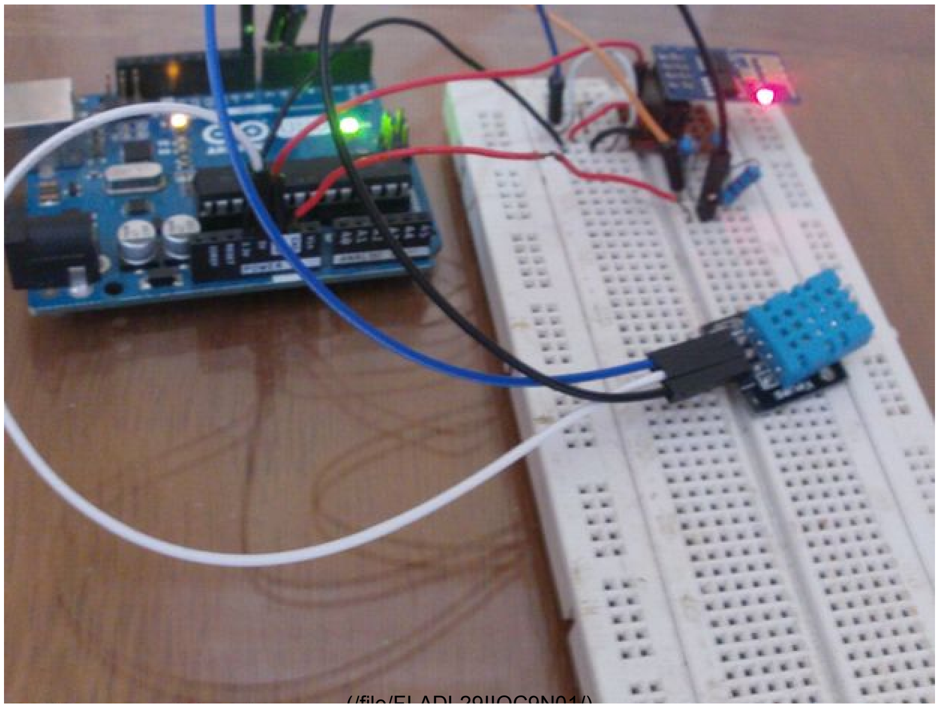
Explore (/tag/type-id/) advertisement Contests (/contests/) Login (/account/login/) | Sign Up (/account/gopro) Classes (/classes/)

Publish (/about/create.jsp)

(http://www.autodesk.com/)

Featured: 3D Printing Class (https://www.instructables.com/class/3D-Printing-Class/) Arduino (/tag/type-id/category-technology/channel-arduino/) AUTODESK Make anything.

Sewing (https://www.instructables.com/tag/type-id/category-craft/channel-sewing/) advertisement


(/file/FLADL29IIOC9N01/)

(/file/FXTUAH2ILIKL3OC/)

(/file/FPM258KILIKL3B2/)

advertisement

About This Instructable

166,989 views

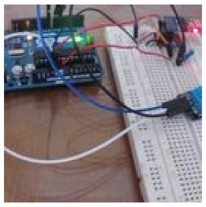
113 favorites

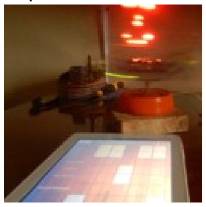
License: BY-NC-SA

Khalilm (/member/Khalilm/)
(https://plus.google.com/102644

(/member/Khalilm/) Follow 34

More by Khalilm:


(/id/Arduino-Esp8266-Post-Data-to-Website/)


(/id/Arduino-Simple-POV-Display/)

Related

Get Started With ESP8266 Using "AT Commands" Via Arduino (/id/Get-Started-With-ESP8266-Using-AT-

Make WiFi Jammer with ESP8266 - Hack any WiFi without password! (/id/DIY-Wifi-Jammer-With-ESP8266-

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.When ESP8266 hosts the application, and when it is the only application processor in the device, it is able to boot up directly from an external flash. It has integrated cache to improve the performance of the system in such applications, and to minimize the memory requirements.

For this project We are going to use The **Esp8266-01 .**

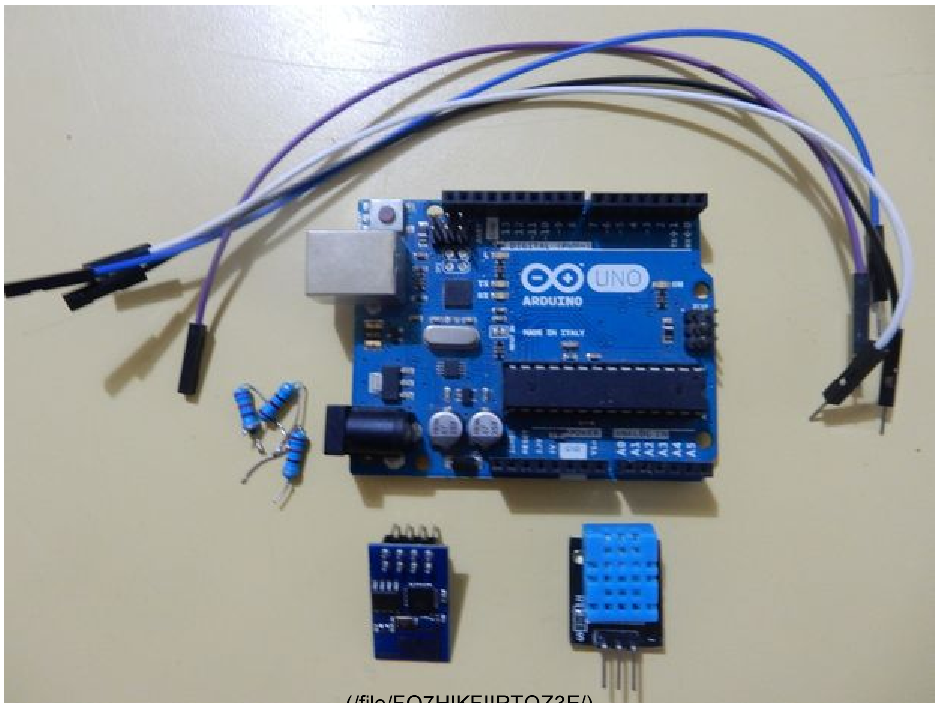
IoT ESP8266 Series: 1- Connect to WIFI Router

advertisement

ESP8266 ESP-12(Standalone)+ Blynk 101 (/id/ESP8266-ESP-12Standalone-Blynk-101/)

3D printed Rover will be IoT. (/id/3D-Printed-Rover-Will-Be-IoT/)
by 明和自動車

Step 1: What You Will Need ??



(/file/FQZHIKFIIPTQZ3E/)

To accomplish this simple yet amazing project you will need a few components that I personally got from **Gearbest.com** mainly thanks to their **High-Quality** products and also for their **Free-shipping WORLDWIDE**. I mean who wouldn't like that !! So the list of the components needed is down below:

- Esp8266 Module **3.32 $** : http://www.gearbest.com/transmitters-receivers-mod... (http://www.gearbest.com/transmitters-receivers-module/pp_218376.html) (Compare this to an Arduino Yun that costs around **75 $** or a WIFI Shield that costs around **85 $**)
- Arduino Uno (you can get the official one or other clones ,same for this project)

Official **8.00 $** : http://www.gearbest.com/development-boards/pp_4772... (http://www.gearbest.com/development-boards/pp_47723.html)

- DHT11 Temperature Humidity Sensor Module **2.68 $** :

http://www.gearbest.com/sensors/pp_238712.html (http://www.gearbest.com/sensors/pp_238712.html)

- 3X 220 Ohm resistors (Voltage divider, explaination in the circuit diagram below) from your local electronics-shop.
- A few Male to Female jumper Wires **1.88 $** :

http://www.gearbest.com/diy-parts-components/pp_23... (http://www.gearbest.com/diy-parts-components/pp_232918.html)

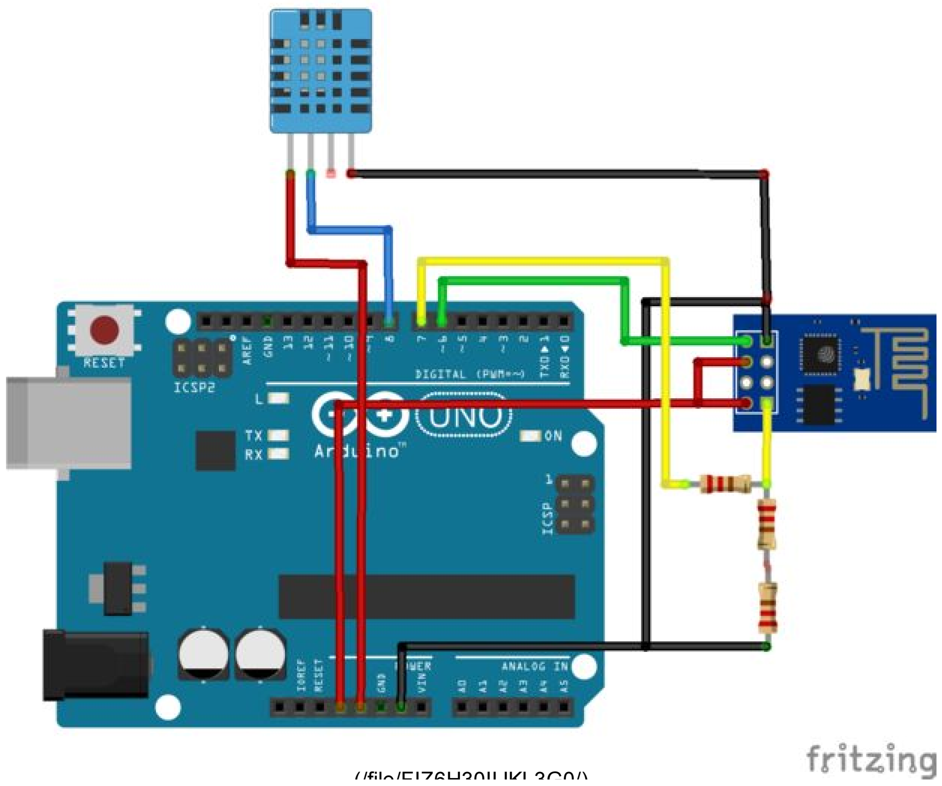
- A website : **0.00 $**

Finally this project won't cost you any higher than **20 $. Isn't That Amazing ?!!**

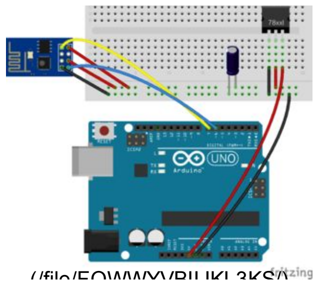
advertisement

advertisement

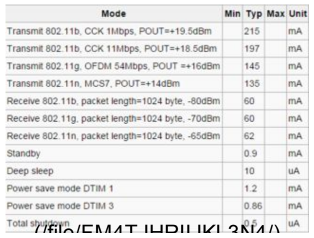
Step 2: Circuit Diagram :



(/file/FIZ6H30IUKL3G0/)

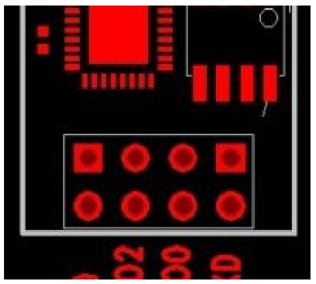


(/file/FOWWYVRIUKL3KS/)



Mode	Min	Typ	Max	Unit
Transmit 802.11b, CCK 1Mbps, POUT=+19.5dBm		215		mA
Transmit 802.11b, CCK 11Mbps, POUT=+18.5dBm		197		mA
Transmit 802.11g, OFDM 54Mbps, POUT =+16dBm		145		mA
Transmit 802.11n, MCS7, POUT=+14dBm		135		mA
Receive 802.11b, packet length=1024 byte, -80dBm		60		mA
Receive 802.11g, packet length=1024 byte, -70dBm		60		mA
Receive 802.11n, packet length=1024 byte, -65dBm		62		mA
Standby		0.9		mA
Deep sleep		10		uA
Power save mode DTIM 1		1.2		mA
Power save mode DTIM 3		0.86		mA
Total shutdown		0.5		uA

(/file/FM4TJHRIUKL3N4/)



This project is based on a very simple Circuit that anyone can do it .

Power Consumption:

First, you need to keep in mind that the Esp8266 runs on 3.3V , connecting it to 5V could work but it is not recommanded.According to its Datasheet,the amperage needed for the Esp8266 to work correctly varies according to its state (transmitting, receiving, deep sleep ,etc...), and reaches its highest at around **200 mA** while the Arduino's DC Current for 3.3V Pin **is Only 50mA,**it actually works but you won't get a perfect result.

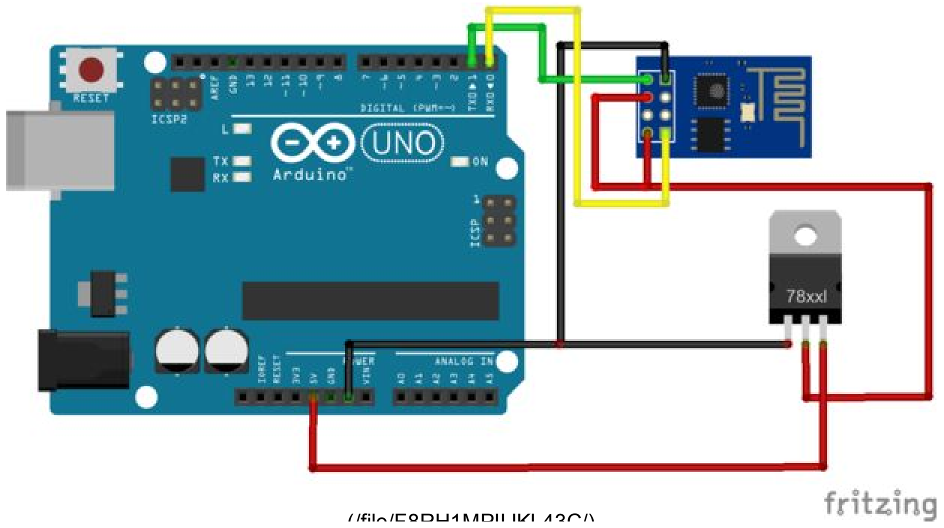
So, there are plenty of other ways to power the module (using diodes, using Zener diode, voltage regulator). I believe that the best solution is to make an independant 5 to 3.3V power regulator circuit, you will need the following components :

- LM1117 or LD1117 : A very common and cheap 3.3 V regulator that allows you to convert the 5V power used by the Arduino to the 3.3V needed by the Esp8266 module.
- A 10 µF capacitor.

Then, when it comes to the TX and RX connections, you have to avoid connecting the TX of the Arduino directly to the RX of the Esp module since that could damage both of them, so two easy solutions you can use to solve this issue. The first one is to use a simple voltage divider with resistors (as in the diagram above) to drop down the 5V voltage from the TX Pin of the Arduino to 3.3 V or you can use a level shifter with a diode, they both do the job correctly and it's up to you to choose either one for your project.

advertisement

Step 3: Testing the Module :



(/file/F8BH1MPIUKL43C/)

If this is the First time you use your Esp8266 module, you probably should test it first by sending AT Commands manually to it . You can do that with an FTDI USB to Serial Converter like this one http://www.gearbest.com/other-accessories/pp_22726... (http://www.gearbest.com/other-accessories/pp_227267.html) or you can simply use your arduino.

If you're going to use your arduino you have to arrange your connection as in the image above :

Pin 0 Arduino (Rx pin) ---->> Rx Pin Module

Pin 1 Arduino (Tx pin) ---->> Tx Pin Module

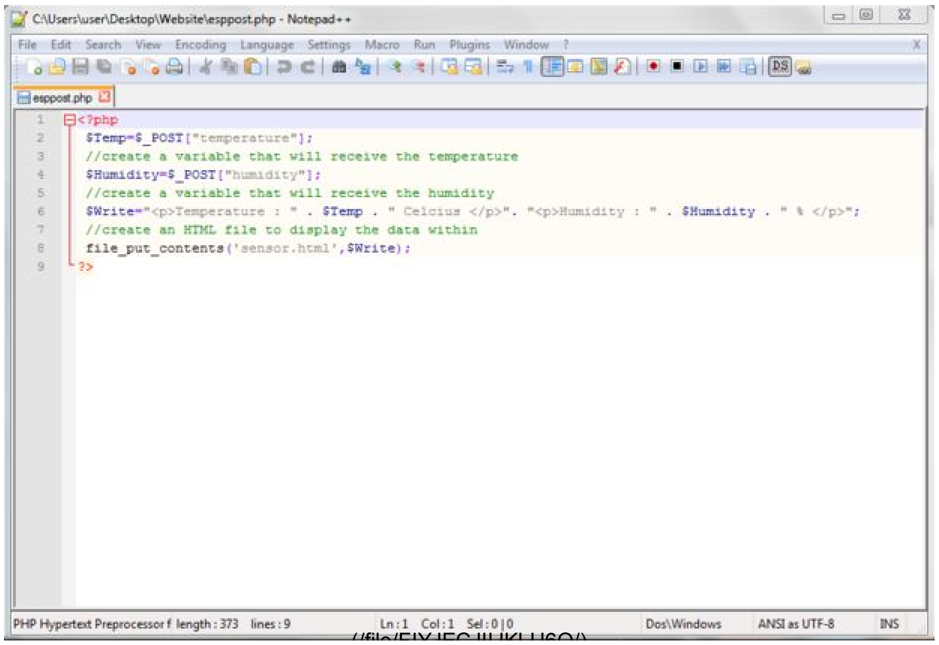
And for the power supply use the power regulator circuit as described in the previous step of this article.

Then, you will have to upload into the arduino an empty code means (void setup() {} void loop () {}) and open the Serial monitor window to type the commands just remember that newer versions of the esp module use **9600 baudrate** while older ones use **57600 or 115200 baud rates.** The Esp8266 responds to a few commands called AT Commands, this is a list of the most basic ones to help you get going and you can find more complex ones just google it.

http://www.pridopia.co.uk/pi-doc/ESP8266ATCommands... (http://www.pridopia.co.uk/pi-doc/ESP8266ATCommandsSet.pdf)

After checking out that your module is working correctly by responding positively to the commands sent you can finally move to the next step and start your HTTP post project.

Step 4: Create a Website First


C:\Users\user\Desktop\Website\esppost.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
esppost.php
PHP Hypertext Preprocessor f length : 373 lines : 9
Ln : 1 Col : 1 Sel : 0 | 0
Dos\Windows
ANSI as UTF-8
INS

```php
<?php
 $Temp=$_POST["temperature"];
 //create a variable that will receive the temperature
 $Humidity=$_POST["humidity"];
 //create a variable that will receive the humidity
 $Write="<p>Temperature : " . $Temp . " Celcius </p>". "<p>Humidity : " . $Humidity . " % </p>";
 //create an HTML file to display the data within
 file_put_contents('sensor.html',$Write);
?>
```

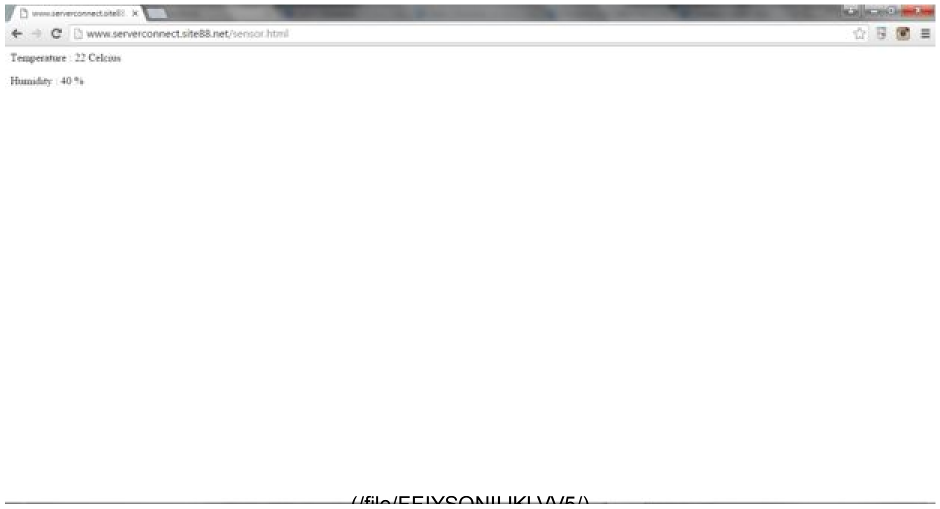
(/file/FIXJFC1IUKLU6O/)

advertisement

So first we need to create our own website and to do that there are plenty of free domain and hosting websites. I personally chose https://www.000webhost.com/ which sounds pretty much convenient for our simple task.

You can create a free website using their free subdomain service then just follow their instructions and you should receive an email that lists your FTP Username and FTP Password and your website will be fully functional within **24 hours**. After you will actually need a software that allows you to send the files (php, HTML , images ...) from your computer to the host servers and to do that I used **FileZilla** you can download it from here (http://www.01net.com/telecharger/windows/Internet/ftp/fiches/17966.html). Before sending any file we need to make a file, so in this example we need to make a php file that we will name httppost.php using **NotePad ++** ,which you can download from here (http://www.01net.com/telecharger/windows/Internet/editeur_de_site/fiches/29119.html), (check the image above for the code and a little explaination).After creating your php server file you are going to actually send that file to the servers that host your website and this is where FileZilla comes in,to accomplish that we first need to establish a connection between your computer and the host servers then you right-click on the file that you wish to send ,in this case esp8266.php, and then transfer.

Step 5: Sending the Data



www.serverconnect.site88.net/sensor.html

Temperature : 22 Celcius

Humidity : 40 %

(/file/FEIYSONIUKLVV5/)

advertisement

First you need to understand the basics of an HTTP.The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a web site may be the server. Example: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content. On our example the esp8266 is the client and the server that is hosting our website is the server.

So performing an HTTP Post request has to be under a certain form:

POST /esppost.php HTTP/1.0

Host: serverconnect.site88.net

Accept: */*

Content-Length: "name1=value1&name2=value2".Length

Content-Type: application/x-www-form-urlencoded

name1=value1&name2=value2

You can find more explaination and informations in the Arduino sketch below.After uploading the Sketch , wait for a few seconds then refresh you webpage.

#include "SoftwareSerial.h"
String ssid ="yourSSID";

String password="yourPassword";

SoftwareSerial esp(6, 7);// RX, TX

String data;

String server = "yourServer"; **// www.example.com** (http://www.example.com)

String uri = "yourURI";**// our example is /esppost.php**

int DHpin = 8;**//sensor pin**

byte dat [5];

String temp ,hum;

advertisement

```
void setup() {
pinMode (DHpin, OUTPUT);
esp.begin(9600);
Serial.begin(9600);
reset();
connectWifi();
}
//reset the esp8266 module
void reset() {
esp.println("AT+RST");
delay(1000);
if(esp.find("OK") ) Serial.println("Module Reset");
}
//connect to your wifi network
void connectWifi() {
String cmd = "AT+CWJAP=\"" +ssid+"\",\"" + password + "\"";
esp.println(cmd);
delay(4000);
if(esp.find("OK")) {
Serial.println("Connected!");
}
else {
connectWifi();
Serial.println("Cannot connect to wifi"); }
}
byte read_data () {
byte data;
for (int i = 0; i < 8; i ++) {
if (digitalRead (DHpin) == LOW) {
while (digitalRead (DHpin) == LOW); // wait for 50us
delayMicroseconds (30); // determine the duration of the high level to
determine the data is '0 'or '1'
if (digitalRead (DHpin) == HIGH)
data |= (1 << (7-i)); // high front and low in the post
while (digitalRead (DHpin) == HIGH);
// data '1 ', wait for the next one receiver
}
} return data; }
void start_test () {
digitalWrite (DHpin, LOW); // bus down, send start signal
```

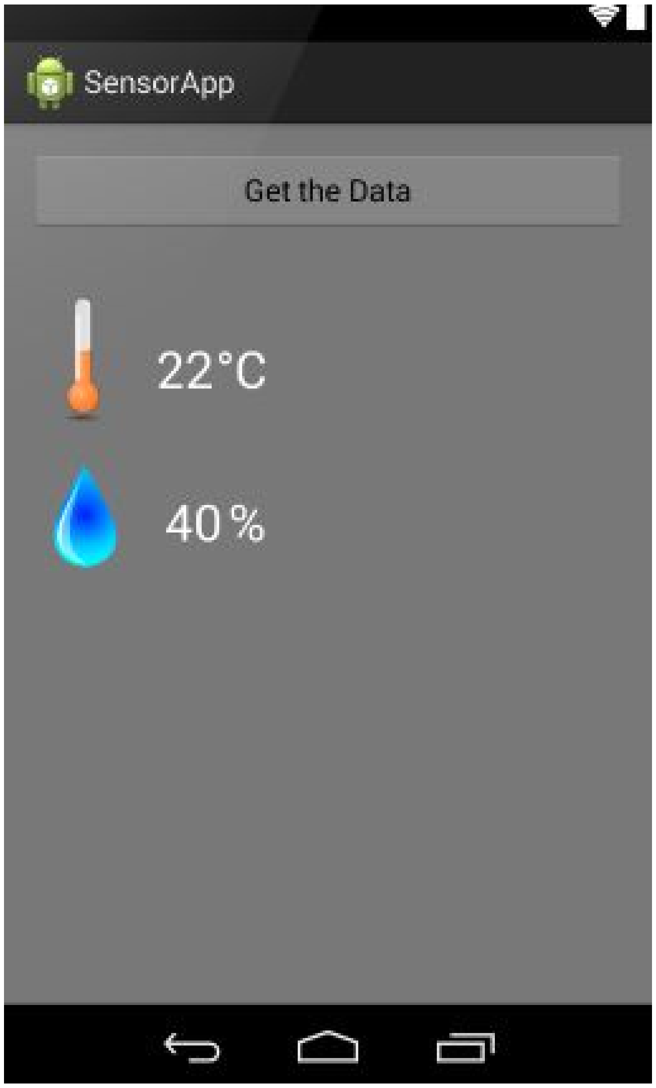
advertisement

```
delay (30); // delay greater than 18ms, so DHT11 start signal can be
detected
digitalWrite (DHpin, HIGH);
delayMicroseconds (40); // Wait for DHT11 response
pinMode (DHpin, INPUT);
while (digitalRead (DHpin) == HIGH);
delayMicroseconds (80);
// DHT11 response, pulled the bus 80us
if (digitalRead (DHpin) == LOW);
delayMicroseconds (80);
// DHT11 80us after the bus pulled to start sending data
for (int i = 0; i < 4; i ++)
// receive temperature and humidity data, the parity bit is not considered
dat[i] = read_data ();
pinMode (DHpin, OUTPUT);
digitalWrite (DHpin, HIGH);
// send data once after releasing the bus, wait for the host to open the next
Start signal
}
void loop () {
start_test ();
// convert the bit data to string form
hum = String(dat[0]);
temp= String(dat[2]);
data = "temperature=" + temp + "&humidity=" + hum;// data sent must be
under this form //name1=value1&name2=value2.
httppost();
delay(1000);
}
void httppost () {
esp.println("AT+CIPSTART=\"TCP\",\"" + server + "\",80");//start a TCP
connection.
if( esp.find("OK")) {
Serial.println("TCP connection ready");
} delay(1000);
String postRequest =
"POST " + uri + " HTTP/1.0\r\n" +
"Host: " + server + "\r\n" +
"Accept: *" + "/" + "*\r\n" +
"Content-Length: " + data.length() + "\r\n" +
"Content-Type: application/x-www-form-urlencoded\r\n" +
"\r\n" + data;

String sendCmd = "AT+CIPSEND=";//determine the number of caracters to be sent.

esp.print(sendCmd);

esp.println(postRequest.length() );

delay(500);

if(esp.find(">")) { Serial.println("Sending.."); esp.print(postRequest);

if( esp.find("SEND OK")) { Serial.println("Packet sent");

while (esp.available()) {

String tmpResp = esp.readString();

Serial.println(tmpResp);

}

// close the connection

esp.println("AT+CIPCLOSE");

}

}}
```

advertisement

If everything went well you should see a result similar to the image above if not unplug then replug your Arduino and wait for a few seconds.

Step 6: You Can Go Even Further



advertisement

This isn't the end of this project ,there are plenty of other stuff you can do like for example I made an **Android App** that will display the data instead of displaying it on a website which is a little bit boring. So to do so the app send an HTTP Get request to the website, retrieve its HTML code as string then get both the Temperature and Humidity.You can find more explaination about HTTP POST and GET from an Android App in this Instructables article (https://www.instructables.com/id/Android-HTTP-GETPOST-Request/) that I made a while ago .You can download the App's source code down below.

The ESP8266 is just a very awesome device !! . you can perform a lot of other stuff with it like (**Home automation system , Mesh network, IP cameras streaming**,etc...)

Thanks for reading my article, if you have any questions about anything just leave them in the comment section below or email me at : **Khalilmerchaoui@gmail.com :D :D :D**

SensorApp.rar Download (https://cdn.instructables.com/ORIG/FS5/08KP/IIJKLYKX/FS508KPIIJKLYKX.rar)
(https://cdn.instructables.com/ORIG/FS5/08KP/IIJKLYKX/FS508KPIIJKLYKX.rar)

advertisement