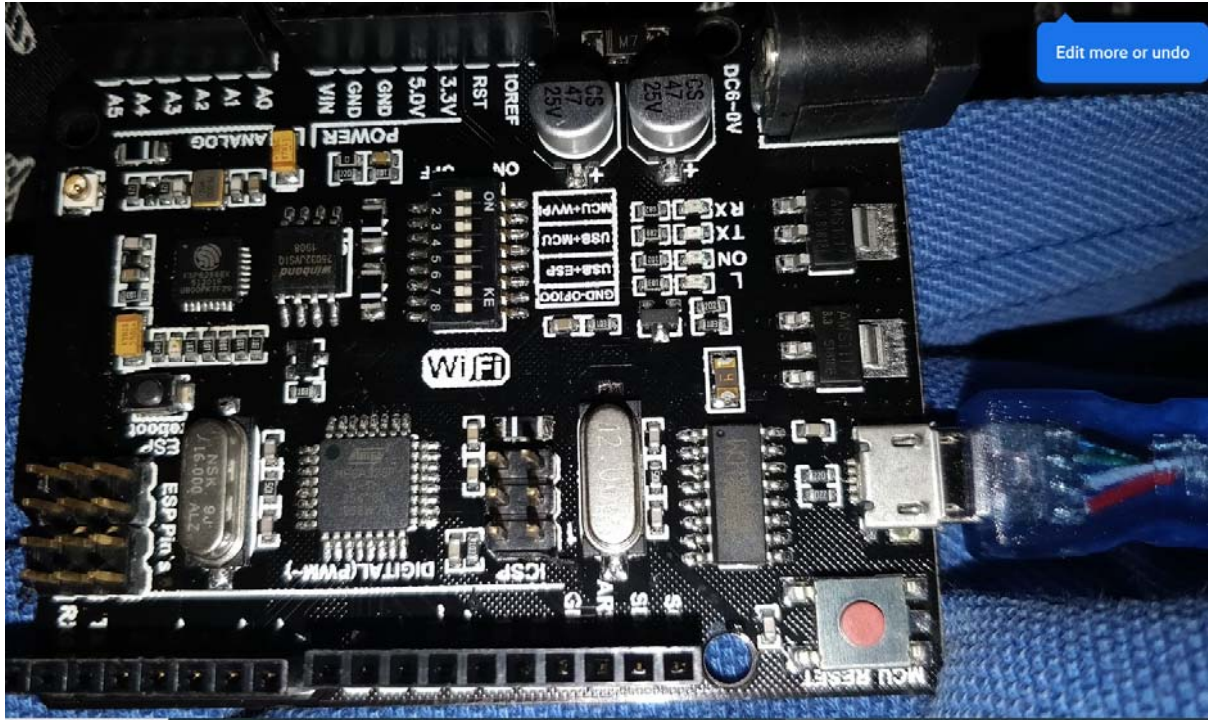
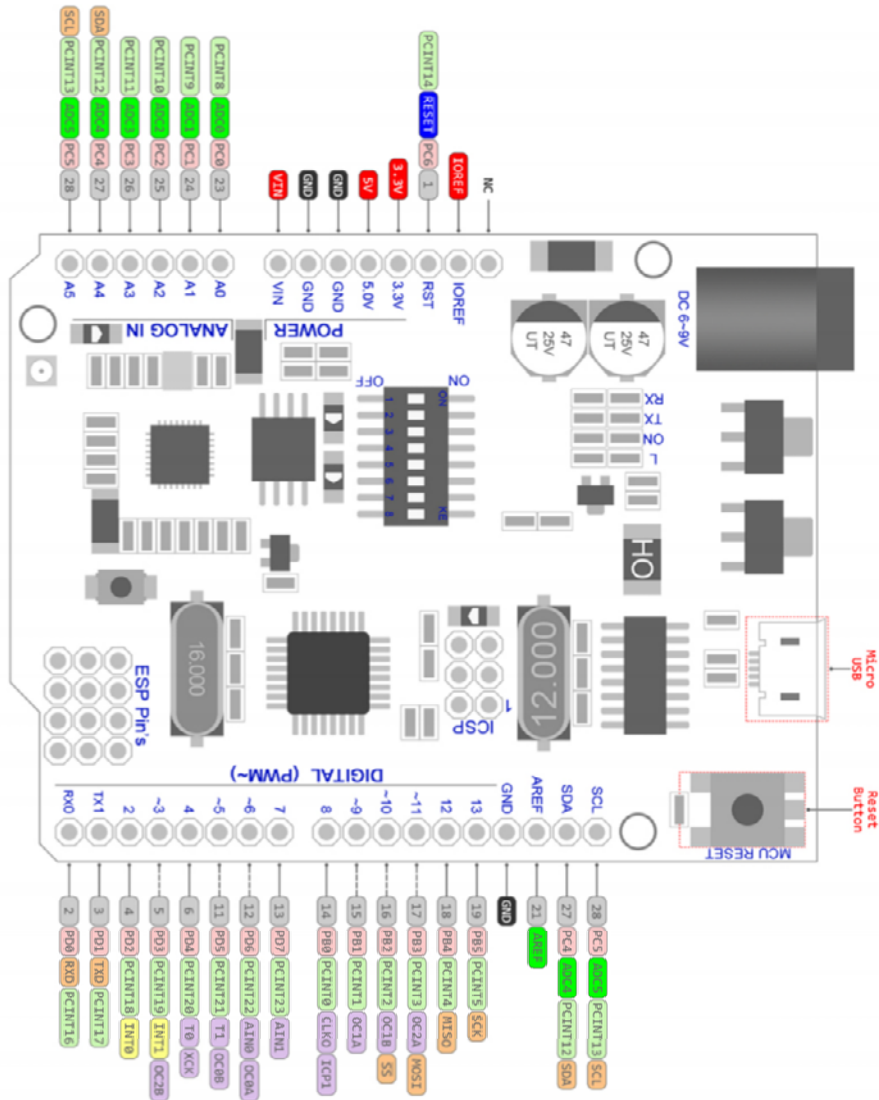


Programming Guide



Switch status and mode selection:

[illegible]



Introduction

I started programming an Arduino a couple of months ago and found it frustrating trying to piece it all together to get a working solution. I've put this together to hopefully speed up the learnings for others and would like to thank everyone who contributes to the various forums.

The Project

Using a UNO+WiFi R3 read the Temperature & Humidity from a DHT22 sensor and write to a SQL Database via wireless then graph the results.

My home network has a Buffalo Linkstation LS-QVL183 NAS which can be configured to run a SQL server.

In my example:

1. ATmega328 reads from the DHT22 sensor and sends values to the ESP8266 (Wifi);
2. The ESP8266 gets a time stamp the Posts the data to the SQL Database

Programming the ESP8266 Module Wifi

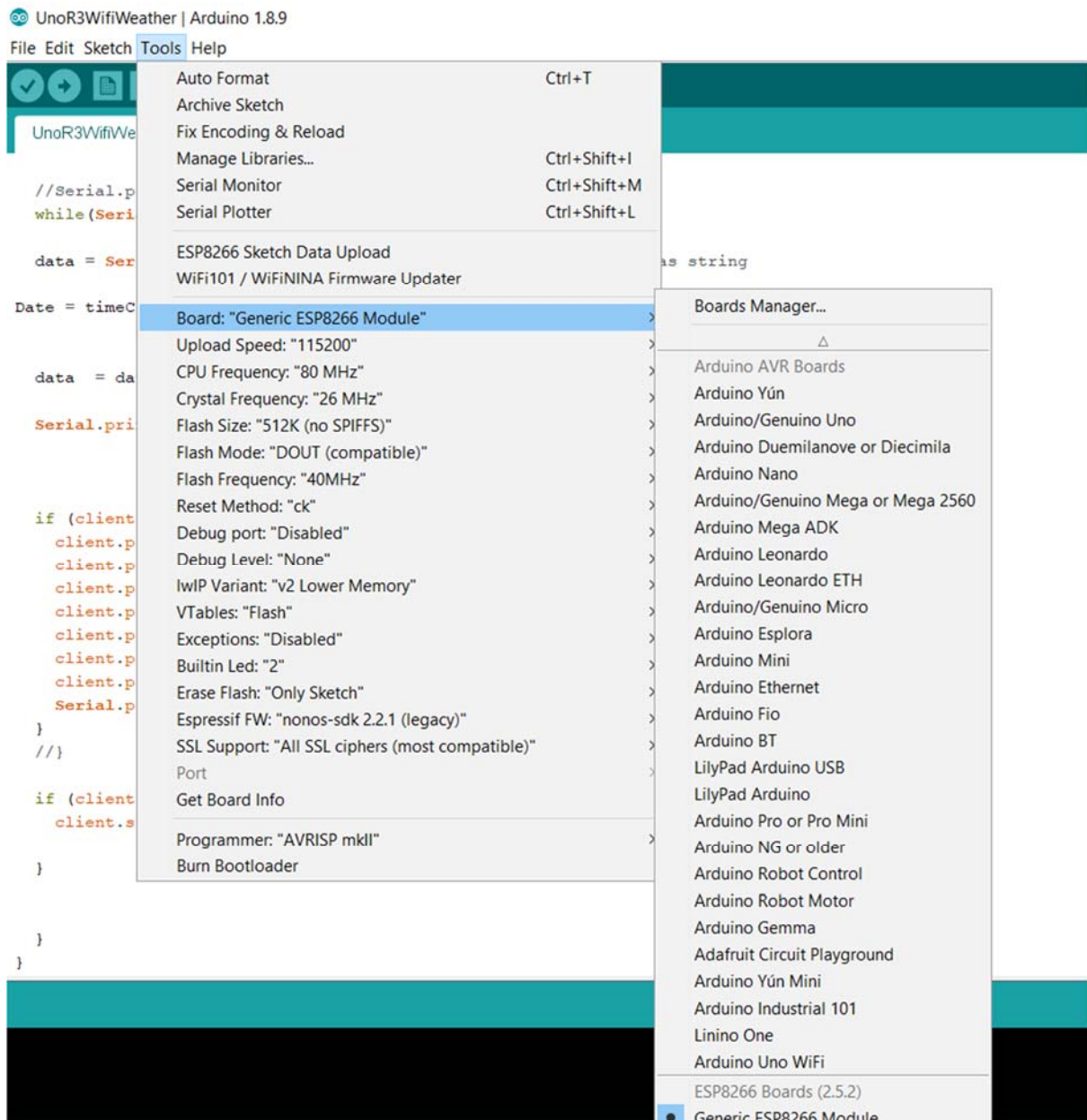
On the UNO R3 set the dip switches 5,6,7 to ON (others OFF) to Upload (I always unplug from USB and/or power before changing dip switches)

Connect the antenna

Select Generic ESP8266 Module; then

Sketch – Upload (Rather than ESP8266 Sketch data upload shown above board selection in screen shot below);

You don't need to do anything with the ESP Reboot button



UnoR3WifiWeather.ino

```
#include <NTPClient.h>
```

```
// https://github.com/arduino-libraries/NTPClient/issues/36#issuecomment-334130633
```

```
// I used the above to get the correct date & time stamp format
```

```
#include <ESP8266WiFi.h>
```

```
#include <WiFiUdp.h>
```

```
#include <time.h>
```

```
WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP, "pool.ntp.org", 28800, 60000);
```

```
// You can specify the time server pool and the offset, (in seconds)
```

```
// additionally you can specify the update interval (in milliseconds).
```

```
// NTPClient timeClient(ntpUDP, "pool.ntp.org", 3600, 28800);
```

```
char ssid[] = "YourWirelessNetwork";           // Network Name
```

```
char pass[] = "WirelessPassword";              // Network Password
```

```
byte mac[6];
```

```
String Date;
```

```
WiFiClient client;
```

```
IPAddress server_addr(192, 168, 1, 5);          // MySQL server IP
```

```
char user[] = "admin";                          // MySQL user
```

```
char password[] = "password";                   // MySQL password
```

```
void setup() {
```

```
    Serial.begin(74880); //Card Default Baud Rate
```

```
    Serial.println("Initialising connection");
```

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

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

```
    Serial.print("Connecting to ");
```

```
    Serial.println(ssid);
```

```
    WiFi.begin(ssid, pass);
```

```
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}
```

```
Serial.println("");  
Serial.println("WiFi Connected");
```

```
WiFi.macAddress(mac);  
Serial.print("MAC: ");  
Serial.print(mac[5],HEX);  
Serial.print(":");  
Serial.print(mac[4],HEX);  
Serial.print(":");  
Serial.print(mac[3],HEX);  
Serial.print(":");  
Serial.print(mac[2],HEX);  
Serial.print(":");  
Serial.print(mac[1],HEX);  
Serial.print(":");  
Serial.println(mac[0],HEX);  
Serial.println("");  
Serial.print("Assigned IP: ");  
Serial.print(WiFi.localIP());  
Serial.println("");  
timeClient.begin();  
}
```

```

void loop(){
  yield();
  timeClient.update();

  int pos = 0;
  String data = "";

  //Serial.println(timeClient.getFullFormattedTime()); //used for debug
  while(Serial.available()) {
    data = Serial.readStringUntil('\n');// read the incoming data as string from the
    ATmega328
    Date = timeClient.getFullFormattedTime(); // Not in standard package
    data = data + "&Date=" + Date;
    Serial.println(data);

    if (client.connect("192.168.1.5",81)) { // REPLACE WITH YOUR SERVER
    ADDRESS
      client.println("POST /addWeather.php HTTP/1.1"); // Explained later
      client.println("Host: 192.168.1.5"); // 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);
      Serial.println(data); //for testing using monitor
    }
    //}

    if (client.connected()) {

```

```
client.stop(); // DISCONNECT FROM THE SERVER

}

}

}
```

Test ESP8266 WIFI Module

On the UNO R3 set the dip switches 5,6 to ON (others OFF) to Run the sketch

Connect to PC serial port

Tools – Serial Monitor & set the baud rate to 74880 and push the reset button on the card. The reason for this, is is the default baud rate for the card – if you set a different baud rate you get random characters until the board changes the baud rate to match the Serial monitor.

If you uncomment the line

```
//Serial.println(timeClient.getFullFormattedTime()); //used for debug
```

After the reset, you should see the IP address assigned to the UNO R3

and the Time Stamp (If uncommented)

If you don't get an IP address and the Antenna is connected the you will need to Flash the ESP8266

Download the ESP8266 Download Tool & bin files in the AT section

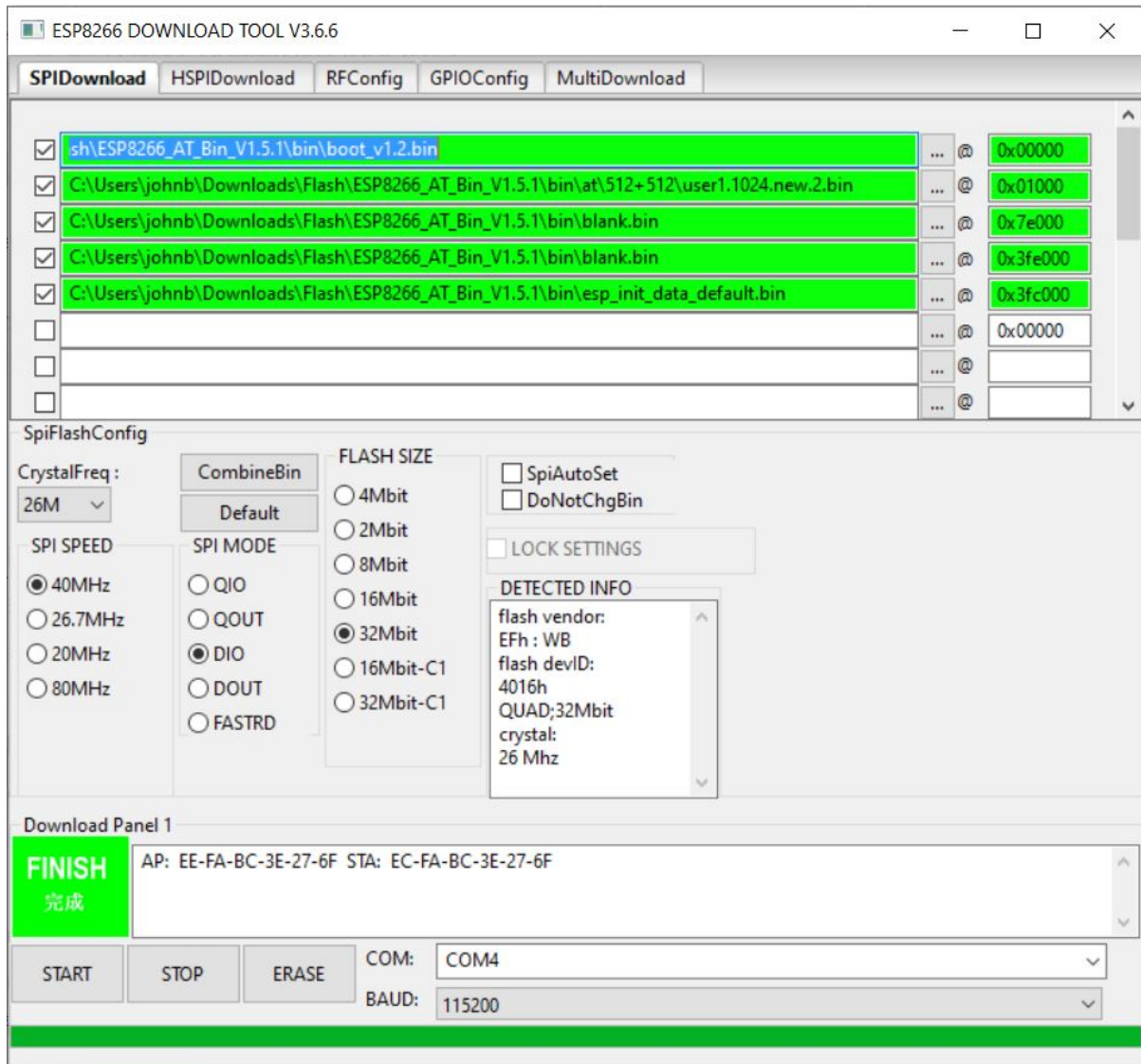
<https://www.espressif.com/en/products/hardware/esp8266ex/resources>

On the UNO R3 set the dip switches 5,6,7 to ON (others OFF)

Setup as per the image below Click start & wait for the finish

Upload **UnoR3WifiWeather.ino** as described above & retry.

There are more recent versions then V1.5.1 but in my case I got it working and decided not to fiddle.



Programming the ATmega328P

On the UNO R3 set the dip switches 3,4 to ON (others OFF), select the board type to “Arduino Wifi UNO” and upload the Sketch below

TempHumidity.ino

/ DHT Temperature & Humidity Sensor

// Unified Sensor Library Example

// Written by Tony DiCola for Adafruit Industries

// Released under an MIT license.

```
// REQUIRES the following Arduino libraries:  
  
// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library  
// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit\_Sensor
```

```
#include <Adafruit_Sensor.h>  
  
#include <DHT.h>  
  
#include <DHT_U.h>
```

```
String Data;  
  
float Temp;  
  
float Humidity;  
  
#define DHTPIN 7    // Digital pin connected to the DHT sensor  
  
// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --  
// Pin 15 can work but DHT must be disconnected during program upload.
```

```
// Uncomment the type of sensor in use:  
  
// #define DHTTYPE    DHT11    // DHT 11  
  
#define DHTTYPE    DHT22    // DHT 22 (AM2302)  
  
// #define DHTTYPE    DHT21    // DHT 21 (AM2301)  
  
  
// See guide for details on sensor wiring and usage:  
//  https://learn.adafruit.com/dht/overview
```

```
DHT_Unified dht(DHTPIN, DHTTYPE);
```

```
uint32_t delayMS;
```

```
void setup() {
```

```

Serial.begin(74880);

// Initialize device.

dht.begin();

Serial.println(F("DHTxx Unified Sensor Example"));

// Print temperature sensor details.

sensor_t sensor;

dht.temperature().getSensor(&sensor);

Serial.println(F("-----"));

Serial.println(F("Temperature Sensor"));

Serial.print (F("Sensor Type: ")); Serial.println(sensor.name);

Serial.print (F("Driver Ver: ")); Serial.println(sensor.version);

Serial.print (F("Unique ID: ")); Serial.println(sensor.sensor_id);

Serial.print (F("Max Value: ")); Serial.print(sensor.max_value);
Serial.println(F("°C"));

Serial.print (F("Min Value: ")); Serial.print(sensor.min_value);
Serial.println(F("°C"));

Serial.print (F("Resolution: ")); Serial.print(sensor.resolution);
Serial.println(F("°C"));

Serial.println(F("-----"));

// Print humidity sensor details.

dht.humidity().getSensor(&sensor);

Serial.println(F("Humidity Sensor"));

Serial.print (F("Sensor Type: ")); Serial.println(sensor.name);

Serial.print (F("Driver Ver: ")); Serial.println(sensor.version);

Serial.print (F("Unique ID: ")); Serial.println(sensor.sensor_id);

Serial.print (F("Max Value: ")); Serial.print(sensor.max_value);
Serial.println(F("%"));

```

```

    Serial.print (F("Min Value:  ")); Serial.print(sensor.min_value);
    Serial.println(F("%"));

    Serial.print (F("Resolution:  ")); Serial.print(sensor.resolution);
    Serial.println(F("%"));

    Serial.println(F("-----"));

    // Set delay between sensor readings based on sensor details.

    delayMS = sensor.min_delay / 1000;
}

```

```

void loop() {

```

```

    // Delay between measurements.

    delay(delayMS);

    // Get temperature event and print its value.

    sensors_event_t event;

    dht.temperature().getEvent(&event);

    if (isnan(event.temperature)) {

        Serial.println(F("Error reading temperature!"));

    }

    else {

        //Serial.print(F("Temperature: "));

        //Serial.print(event.temperature);

        //Serial.println(F("°C"));

        Temp = event.temperature;

    }

    // Get humidity event and print its value.

    dht.humidity().getEvent(&event);

    if (isnan(event.relative_humidity)) {

        Serial.println(F("Error reading humidity!"));

    }
}

```

```
else {
```

```
    //Serial.print(F("Humidity: "));
```

```
    //Serial.print(event.relative_humidity);
```

```
    //Serial.println(F("%"));
```

```
    Humidity = event.relative_humidity;
```

```
}
```

```
Data = FltToStr(Temp,"Temp")+
```

```
    FltToStrL(Humidity,"Humidity");
```

```
Serial.println(Data);
```

```
delay(600000); //Comment out for Testing
```

```
}
```

```
String FltToStr( float instr, String var)
```

```
{
```

```
return( var+"="+ String(instr,2)+"&");
```

```
}
```

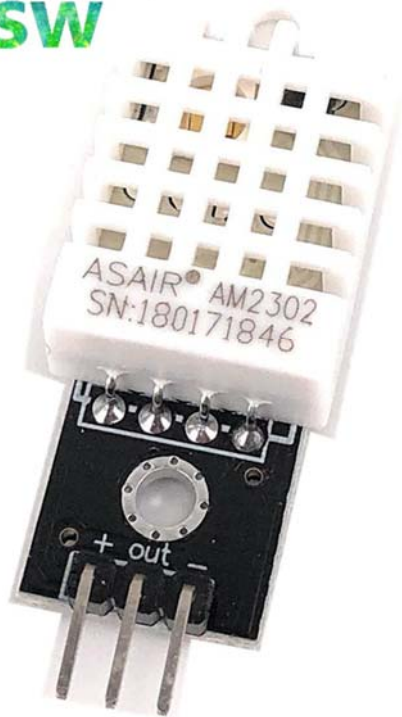
```
String FltToStrL( float instr, String var)
```

```
{
```

```
return( var+"="+ String(instr,2));
```

```
}
```

cuiisw



Connect the sensor to 3.3V, GND and digital input 7

Testing

Uncomment the Highlighted Code

Connect to PC serial port

Tools – Serial Monitor & set the baud rate to 74880 and push the reset button on the card.

No dip switches need to be changed

The serial monitor should display the Temperature & Humidity

Running

Reinstate the Comments

On the UNO R3 set the dip switches 1,2 to ON (others OFF)

In this mode

`Serial.println(Data);` in TempHumidity.ino sends the "Data" to the WIFI Module

`data = Serial.readStringUntil('\n');` in UnoR3WifiWeather.ino reads the Data sent in the line above;

The below lines then POST the data to the SQL Server

```
client.println("POST /addWeather.php HTTP/1.1");
```

```
client.println("Host: 192.168.1.5");
```

SQL Server

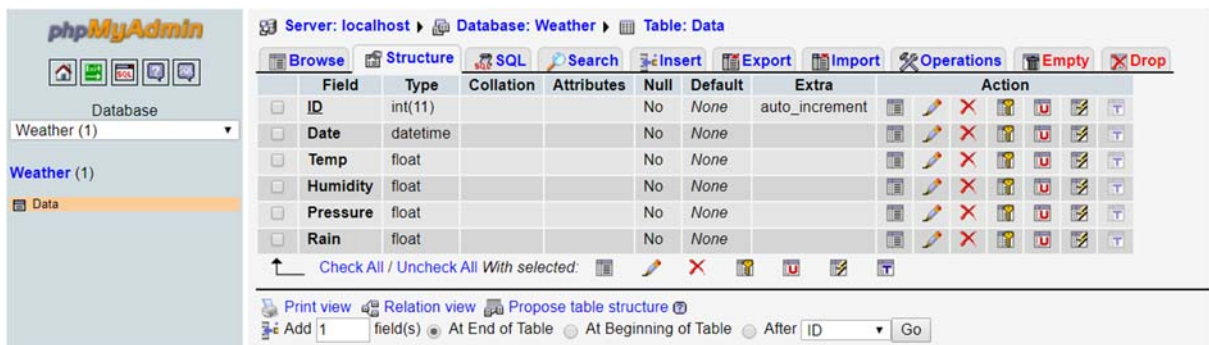
Start Chrome & start the SQL Admin & Login

<http://192.168.1.5:81/phpMyAdmin/index.php>



The image shows the phpMyAdmin welcome page. At the top is the phpMyAdmin logo, which features a stylized sailboat. Below the logo, the text "Welcome to phpMyAdmin" is displayed. There is a language selection dropdown menu set to "English". Below that is a login section with fields for "Username:" (containing "admin") and "Password:", and a "Go" button.

Create Database



The image shows the phpMyAdmin interface with the SQL tab selected. The left sidebar shows the database structure: "Database" (Weather (1)), "Weather (1)", and "Data". The main area displays the table structure for "Table: Data". The table has the following fields:

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> ID	int(11)			No	None	auto_increment	
<input type="checkbox"/> Date	datetime			No	None		
<input type="checkbox"/> Temp	float			No	None		
<input type="checkbox"/> Humidity	float			No	None		
<input type="checkbox"/> Pressure	float			No	None		
<input type="checkbox"/> Rain	float			No	None		

Below the table, there are buttons for "Check All / Uncheck All", "With selected:", and "Add 1 field(s)". There are also buttons for "Print view", "Relation view", and "Propose table structure". At the bottom, there are radio buttons for "At End of Table", "At Beginning of Table", and "After ID", and a "Go" button.

In the SQL Directory [\\ls-qvl183\Httpx\htdocs](http://192.168.1.5:81/phpMyAdmin/index.php)

Create files below

connectWeather.php

```
<?php

function Connection(){
    $server="localhost";
    $user="admin";
    $pass="password";
    $db="Weather";

    $connection = mysql_connect($server, $user, $pass);

    if (!$connection) {
        die('MySQL ERROR: '. mysql_error());
    }

    mysql_select_db($db) or die( 'MySQL ERROR: '. mysql_error() );

    return $connection;
}

?>
```

addWeather.php

```
<?php

include("connectWeather.php");

$link=Connection();

$Temp=$_POST["Temp"];
$Humidity=$_POST["Humidity"];
$Pressure=$_POST["Pressure"];
```

```
$Rain=$_POST["Rain"];
```

```
$Date=$_POST["Date"];
```

```
$query = "INSERT INTO `Data` (`Temp`,`Humidity`,`Pressure`,`Rain`,`Date`)
```

```
VALUES ('".$Temp."','".$Humidity."','".$Pressure."','".$Rain."','".$Date."')";
```

```
mysql_query($query,$link);
```

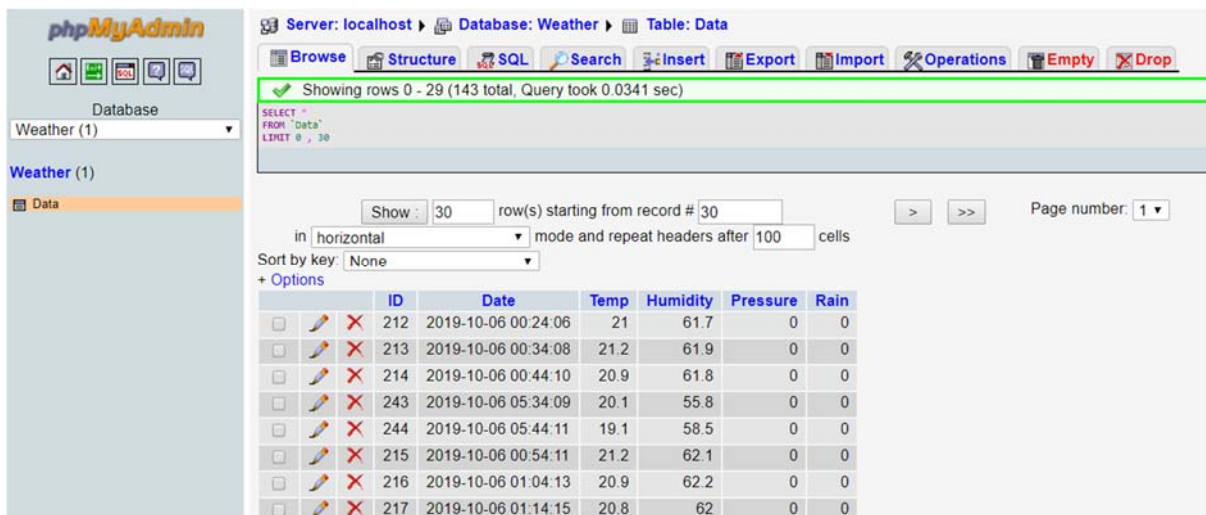
```
mysql_close($link);
```

```
header("Location: index.php");
```

```
?>
```

When you reboot the UNO R3 it should write to the database – some zero values are written into the database while booting; this could be prevented with a delay;

When up & running the database should be populated every 10 minutes



Server: localhost Database: Weather Table: Data

Showing rows 0 - 29 (143 total, Query took 0.0341 sec)

SELECT * FROM `Data` LIMIT 0, 30

Show 30 row(s) starting from record # 30 in horizontal mode and repeat headers after 100 cells

Sort by key: None

	ID	Date	Temp	Humidity	Pressure	Rain
<input type="checkbox"/>	212	2019-10-06 00:24:06	21	61.7	0	0
<input type="checkbox"/>	213	2019-10-06 00:34:08	21.2	61.9	0	0
<input type="checkbox"/>	214	2019-10-06 00:44:10	20.9	61.8	0	0
<input type="checkbox"/>	243	2019-10-06 05:34:09	20.1	55.8	0	0
<input type="checkbox"/>	244	2019-10-06 05:44:11	19.1	58.5	0	0
<input type="checkbox"/>	215	2019-10-06 00:54:11	21.2	62.1	0	0
<input type="checkbox"/>	216	2019-10-06 01:04:13	20.9	62.2	0	0
<input type="checkbox"/>	217	2019-10-06 01:14:15	20.8	62	0	0

Displaying the Data

In the SQL Directory <\\\\ls-qv1183\\Httpx\\htdocs>

Create files below

GraphHumidityTemp.php

```
<html>
```

```
<body>
```

```
<form action="HumidityTemp.php">
```

```
  Date:
```

```
  <input type="date" name="sday", value='<?php echo date('Y-m-d');?>'>
```

```
  <input type="submit">
```

```
</body>
```

```
</html>
```

HumidityTemp.php

```
<?PHP
```

```
define('DB_SERVER', 'localhost');
```

```
define('DB_USER', 'admin');
```

```
define('DB_PASS', 'password');
```

```
$db_handle = mysql_connect(DB_SERVER, DB_USER, DB_PASS);
```

```
$database = "Weather";
```

```
$db_found = mysql_select_db($database);
```

```
$datetime = date_create()->format('Y-m-d');
```

```
$datetime = $_REQUEST['sday'];
```

```
$edate=strtotime($_POST['edate']);
```

```
$edate=date("Y-m-d",$edate);
```

```
echo "<h2>" . $datetime . "</h2>";
```

```
if ($db_found) {
```

```
$SQL = "SELECT `Data`.*
```

```
FROM `Data`
```

```
WHERE ( `Data`.`Date` >= '$datetime' and `Data`.`Date` < date_add('$datetime',INTERVAL 1 DAY) )
```

```
ORDER BY `ID`";
```

```
$result = mysql_query($SQL);
```

```
while ( $db_field = mysql_fetch_assoc($result) ) {
```

```
else {
```

```
print "Database NOT Found ";
```

```
//fetch data
```

```
$result = mysql_query($SQL);
```

```

while ($row = mysql_fetch_array($result)) {

    $entry .= "[".$row{'Date'}."", ".$row{'Humidity'}."", ".$row{'Temp'}.""], ";

}

mysql_close($db_handle);

?>

<html>
<head>

<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">

    // Load Charts and the corechart package.
    google.charts.load('current', {'packages':['corechart']});

    // Draw the pie chart for Sarah's pizza when Charts is loaded.
    google.charts.setOnLoadCallback(drawChart1);

    // Draw the pie chart for the Anthony's pizza when Charts is loaded.
    //google.charts.setOnLoadCallback(drawChart2);

    function drawChart1() {
        var data = google.visualization.arrayToDataTable([
            ['Date', 'Humidity', 'Temp'],
            //['Date', 'V_A', 'V_B', 'V_C'],

            <?php echo $entry ?>
        ]);

        var options = {
            zoomEnabled: true,

```

```

        title: 'Humidity/Temp',
        curveType: 'function',
        legend: { position: 'bottom' },
        'width':1400,
        'height':600,
        vAxis: {minValue: 0},
        explorer: {
            actions: ['dragToZoom', 'rightClickToReset'],
            axis: 'horizontal',
            keepInBounds: false,
            maxZoomIn: 4.0},

    };

    var chart = new google.visualization.LineChart(document.getElementById('Chart1_div'));
    chart.draw(data, options);
}

</script>
</head>
<body>
    <!--Table and divs that hold the pie charts-->
    <table class="columns">
        <tr>
            <td><div id="Chart1_div" style="border: 1px solid #ccc"></div></td>

        </tr>
    </table>
</body>

```

</html>

Start Chrome

<http://192.168.1.5:81/GraphHumidityTemp.php>

should show the below where the date can be selected

Date:

October 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

After submit the data should be displayed

2019-10-06

