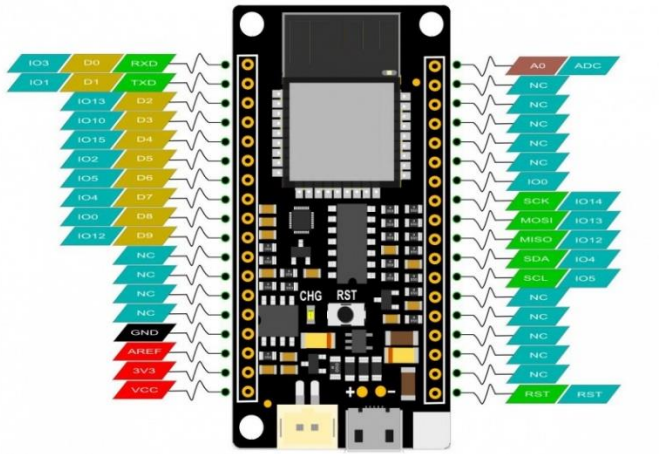
Workshop 9: "ESP8266 data logging"

1 Description

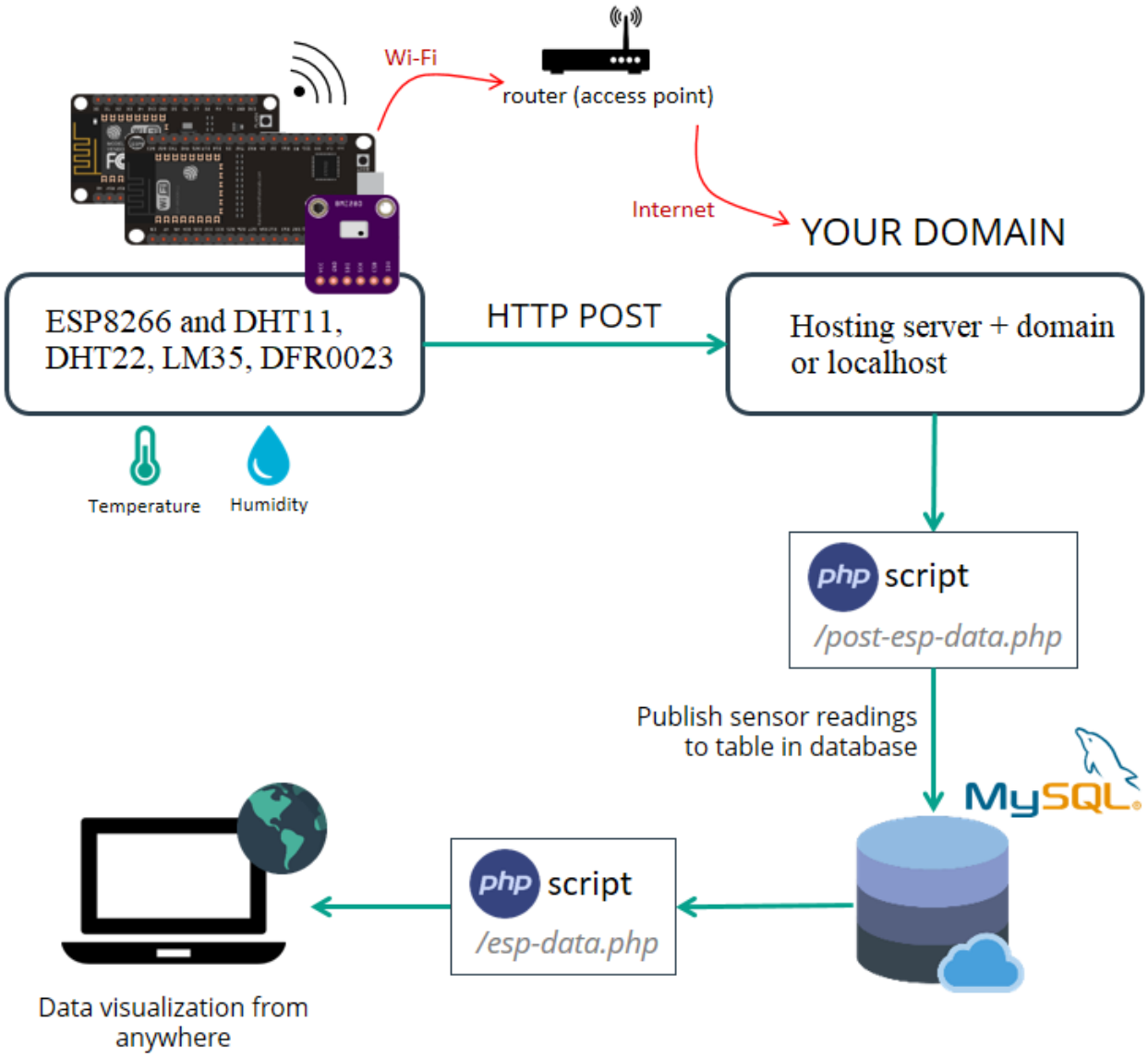


DFRobot FireBeetle is a series of low-power-consumption development hardware designed for Internet of Things (IoT). Firebeetle ESP8266 is a development board integrated with IoT WiFi, TCP/IP, 32-bit MCU, 10-bit ADC and multiple interfaces such as HSPI, UART, PWM, I2C and I2S. In DTIM10, the full power consumption to maintain WiFi connection reached to 1.2mW. Equipped with 16MB outer SPI flash memory, ESP8266 is available for programs and firmware storage. Compatible with Arduino programming enables Firebeetle ESP8266 to lower the barrier of programming. Operator can implement Arduino programming codes directly onto ESP8266 to reduce the difficulty of operating and increase the stability of board. Current is measured contactless via inductive current sensors. In the standard version currents up to 100 A can be measured. A simple changeover of the jumper enables the connection of any cable re-converter with a secondary current of 1 A. This allows currents of up to 1000 A and more to be detected.

2 Equipment

- ESP8266
- LM35 sensor of temperature
- Laptop
- Wires
- Support desk
- Breadboard

3 Technical info



Search on internet the datasheets of the several sensors

Write here the links of datasheet and info

3.1 Features

Application examples

- Monitor via network and Internet
- Monitoring of Temperature
- Integration into smart solutions

Software

- Reading temperature and other data
- Minutes store the values in the database
- Library Support
- Web interface for display and configuration

4 Shematic and other possibility of solutions

Find and draw other solutions (Fritzing or tinkercad)

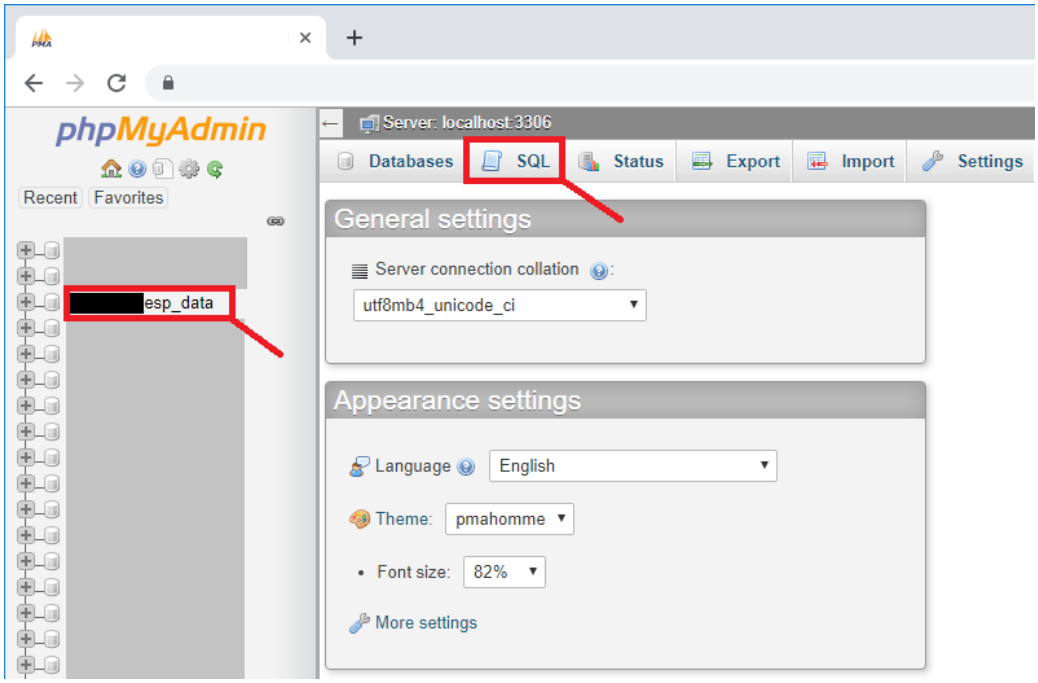
5 Programming

First create database in phpMyAdmin or you can use another tool.

Database – thermometer

Table – temps

Columns – id (int), temp(int), day(datetime)



Then write and edit php script in below (Name - Show):

```
<html>
<head>
<meta http-equiv="refresh" content="5">
</head>
<body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "thermometer";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM temps ORDER by id DESC LIMIT 33";
$result = $conn->query($sql);
```

```
if ($result->num_rows > 0) {
   // output data of each row
       echo   "<table   border='1'><th>ID</th><th>TEMP(deg   C)</th><th>DATE
TIME</th>";
   while($row = $result->fetch_assoc()) {
               echo "<tr>";
               echo "<td>".$row['id']."</td>";
               echo "<td>".$row['temp']."</td>";
               echo "<td>".$row['day']."</td>";
               echo "</tr>";
   }
       echo "</table>";
} else {
   echo "0 results";
}
$conn->close();
?>
</body>
</html>
```

And second script (Name – ADD)

```
<?php
$temp = $_GET['temp'];

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "thermometer";

// Create connection
$conn = new mysqli($servername, $username,$password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$val = $_GET['temp'];
$sql = "INSERT INTO temps(temp) VALUES ($val);";

if ($conn->query($sql) === TRUE) {
   echo "Temperature Saved Successfully!";
} else {
   echo "Error:" . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

Now you can write and edit Arduino code:

```
//include libraries
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>

//Access point credentials
const char* ssid = "insert your wifi name here";
const char* pwd = "insert your password here";

int sensor;
float temperature, refvoltage, temp;

int ledPin = 13; //connected to pin 7 in nodemcu 1.0 esp-12e module
WiFiServer server(80);  // open port 80 for server connection

void setup()
{
 Serial.begin(115200); //initialise the serial communication
 delay(10);

 //defining the pins, i/p and o/p
 pinMode(ledPin, OUTPUT);
 pinMode(A0, INPUT);
 digitalWrite(ledPin, LOW);

  refvoltage = 2.25;   //reference voltage for temperature calculation

 //connecting to wifi
 Serial.println();
 Serial.println();
 Serial.print("Connecting to wifi ");
 Serial.println(ssid);

 WiFi.begin(ssid, pwd);

 //attempt to connect to wifi
 while (WiFi.status() != WL_CONNECTED)
 {
  delay(500);
  Serial.print("."); //progress with dots until the wifi is connected
 }
  Serial.println("");

  //while connected print this
  Serial.println("Wifi connected");

  //starting the server
  server.begin();
  Serial.println("Server started.");
```

```
    //get the ip address and print it
    Serial.print("This is your ip address: ");
    Serial.print("http://");
    Serial.print(WiFi.localIP());
    Serial.println("/");
}

void loop()
{
     WiFiClient client = server.available();
    if (!client)
    {
      return;
    }

    //wait for the data to be sent from client
    Serial.println("New client connection");
    while(!client.available())
    {
      delay(1);
    }

    //Read the first line of the request
    String request = client.readStringUntil('\r');
    Serial.println(request);
    client.flush();

   int value = LOW; // set the thermometer off initially
   sensor = analogRead(A0);
   temperature = (refvoltage * sensor * 100) / 1023; //converting degree to celcius

    //check the index of the browser and act accordingly
    if (request.indexOf("/Thermo=ON") != -1)
    {
        digitalWrite(ledPin, HIGH);
        value = HIGH;
        sensor = analogRead(A0);
        temperature = (refvoltage * sensor * 100) / 1023; //converting degree to celcius

        //save the data to mysql, access the php file to write
        HTTPClient http;
        String                                    url                                    =
"http://192.168.1.5/iotproject/thermotry/add.php?temp="+String(temp);
        Serial.println(url);
        http.begin(url);

        //GET method
```

```
      int httpCode = http.GET();
      if(httpCode > 0)
      {
        Serial.printf("[HTTP] GET...code: %d\n", httpCode);
        if(httpCode == HTTP_CODE_OK)
        {
           String payload = http.getString();
           Serial.println(payload);
        }
      }
      else
      {
         Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
      }
        http.end();
        delay(3000);
  }

   if (request.indexOf("/Thermo=OFF") != -1)
   {
    digitalWrite(ledPin, LOW);
    value = LOW;
   }
    temp = temperature;

    //Return the response
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");

    if (request.indexOf("/AlwaysOn") != -1)
    {
     digitalWrite(ledPin, HIGH);
     value = HIGH;
     sensor = analogRead(A0);
     temperature = (refvoltage * sensor * 100) / 1023; //converting degree to celcius
     //temp = temperature;
     client.println("Refresh: 5");
    }

    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("<body style=background-color:skyblue> </body>");
    client.println("<style> h1 {text-align: center}</style>");
    client.println("<style> h3 {text-align: center}</style>");
    client.println("<style> b {text-align: center}</style>");
    client.println("<style> p {text-align: center}</style>");
```

```
    client.println("<head><style>div.relative{position:relative; left:200px;
height:20px; width:350px; border:2px solid #73AD21;}</style></head>");
    client.println();

    client.println("<div class=relative>  <b> UNIVERSITY OF TURKU  </b> </div>");
    client.println("<div class=relative> Ashish Shrestha  </div>");
    client.println("<div class = relative>  Rituraj Nepal </div>");

    //client.println("<br>");
    client.println("<br><br><h1>IOT Final Project:
THERMOMETER</h1><br><br>");
    //client.println("<h3>Probable LM35 analog reading: <h3>");
    //client.print(sensor); //analog reading of the sensor
    //client.print("<br>");

    client.print("<h3>Thermometer's current status: </h3>");
    if(request.indexOf("/AlwaysOn") != -1)
    {
      client.print("<h3>Always On<h3>");
      client.print("(Temperature updated every 5 seconds.)");
      client.println("<h3>Current Temperature: <h3>");
      client.print(temp);
      client.print("<b> deg C.</b>");

      //save the data to mysql, with http client
    HTTPClient http;
    String url =
"http://192.168.1.5/iotproject/thermotry/add.php?temp="+String(temp);    //access
the php file to write data
    Serial.println(url);
    http.begin(url);

    //using GET method to write to sql
    int httpCode = http.GET();
    if(httpCode > 0)
    {
       Serial.printf("[HTTP] GET...code: %d\n", httpCode);
        if(httpCode == HTTP_CODE_OK)
        {
          String payload = http.getString();
          Serial.println(payload);
        }
    }
      else
      {
        Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
      }
```

```
        http.end();
      //delay(3000);
   }

    else if(value == HIGH)
    {
        client.print("<h3>On Once<h3>");
        //client.println("<br>");
        client.println("<h3>Current Temperature: <h3>");
        client.print(temp);
        client.print("<b> deg C.</b>");
        //delay(100);
     }
     else if(value == LOW)
     {
        client.print("<h3>Off<h3>");
       // client.println("<br>");
        client.println("<h3>Last measured temperature: <h3>"); //print the last value
of temp
        client.print(temp);
        client.print("<b> deg C.</b>");
     }

     client.println("<br><br>");
     //create buttons
     client.println("<a       href=\"/Thermo=ON\"\"><button>Turn       On       Once
</button></a>");
     client.println("<a href=\"/AlwaysOn\"\"><button>Keep It On </button></a>");
     client.println("<a            href=\"/Thermo=OFF\"\"><button>Turn            Off
</button></a><br />");

     client.println("<h3>Check the Temperature records by following the below link:
<h3>");
     client.println("<p>http://localhost/iotproject/Thermometer/show.php</p>");

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

     client.println("</html>");

     client.println();
     delay(1);
     Serial.println("Client disconnected!");
     Serial.println("");
```

```
}
```

6 Links

-
https://www.youtube.com/watch?v=OZpvE4U2pPU&ab_channel=EmbedotronicsTechnologies

- https://github.com/Embedotronics/How-to-send-DHT11-data-to-MySQL-server-using-NodeMCU

- https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/

- https://www.instructables.com/How-to-Connect-NodeMCU-ESP8266-to-MySQL-Database-1/

https://www.youtube.com/watch?v=my0cEr5QGAs&ab_channel=1111rik

https://github.com/theagleye/Wifi-temperature-sensor