ESCUELA DE
CIENCIAS MATEMÁTICAS Y
TECNOLOGÍA INFORMÁTICA

Wireless Network

LAB9B: ESP8266 WiFi Module based Projects

Objective

1. Describe the basic principles of radio communications.
2. Understand the usage of WiFi serial communication module to set the control parameters.

Discussion of fundamentals

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC (Negative Temperature Coefficient → the resistance decreases as the temperature increases) temperature measurement component, and connects to a highperformance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

The pins in the DHT11 are:

- GND: connection to ground
- DATA: data transmission
- VCC: Power

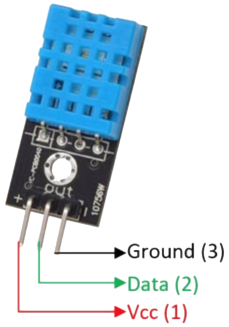


Figure 1: ESP8266 WiFi Module

Exercise 5. Temperature and Humidity - DHT Library.

This exercise demonstrates a way to use as an output of the GPIO2 pin to an LED that will flash every second.

Building Circuit

Before making the connection make sure to unplug the power source from Arduino UNO. Make following circuit. Note that the GIOI0 is connected to ground so that the firmware recording is possible. Once the firmware is loaded, we can disconnect it so that it works correctly.

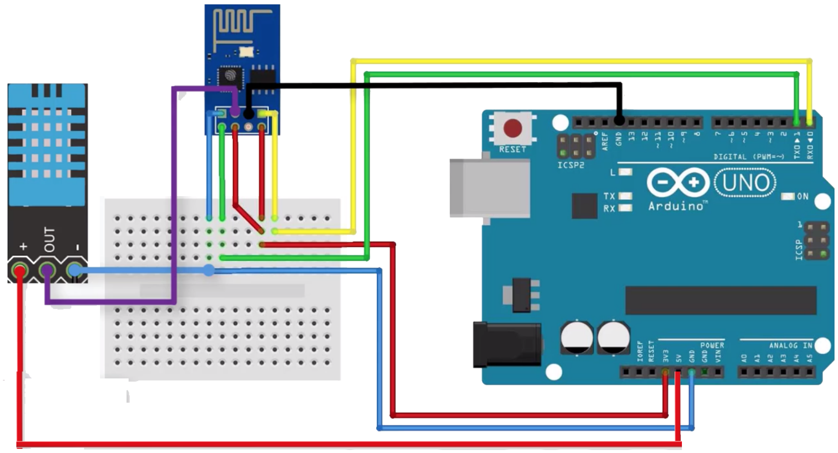


Figure 2: Assembly Scheme 1

Programming

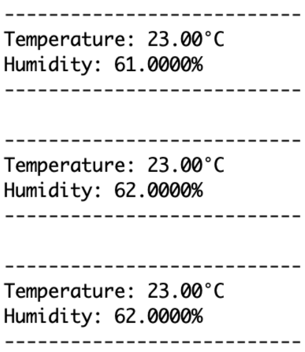
Open up the Arduino IDE and write the following code into a new sketch:

Code

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

/*** Sensor model ***/
#define DHTTYPE DHT11 //'DHT21, DHT22'

/*** Pin GPIO2***/
#define DHTPIN 2 // GPIO2

DHT dht(DHTPIN, DHTTYPE, 27); // 'DHT11 works fine for ESP8266 threshold => MHZ CPU'

/*** Variables for Humidity and Temperature ****/
float temperature; // 'double'
float humidity;

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

}

void loop(){
   temperature = dht.readTemperature();
   humidity = dht.readHumidity();
   Serial.println("----------------------------");
   Serial.print("Temperature: ");
   Serial.print(temperature);
   Serial.println(" C ");
   Serial.print("Humidity: ");
   Serial.print(humidity, 4);
   Serial.println("%");
   Serial.println("----------------------------");
   Serial.println();

   delay(3000);
} // EOF loop()
```

Output Result

Once you have successfully set up the circuit connections and your arduino code has been uploaded, you can connect a power source to your arduino board. Remember to check the port, click `Tools -> Port` and select the COM port which your WiFi module is connected. See the result on Serial Monitor



```
---------------------------
Temperature: 23.00°C
Humidity: 61.0000%
---------------------------

---------------------------
Temperature: 23.00°C
Humidity: 62.0000%
---------------------------

---------------------------
Temperature: 23.00°C
Humidity: 62.0000%
---------------------------
```

Figure 3: Serial Monitor

Exercise 6. Read Temperature, Humidity and Display at the Web

In this exercise, we will use XAMPP as a suite of Web development tools, created by Apache Friends, makes it easy to run PHP (Personal Home Pages) scripts locally on your computer. For this, you must:

1. Install, start and test XAMPP
2. Create a folder `/esp8266/` in `/htdocs`
3. Create index.php and copy in the folder: `/htdocs/esp8266/`

Programming the index.php file

Open up an PhP editor and write the following code into a new file:

Code

```
<?php
    //'Create a text  file  to save the data sent by ESP8266'
    if (! file_exists ("miTemp&Hum.txt")){
        // 'If the file does not exist , we create it'
         file_put_contents ("miTemp&Hum.txt", "0.0\r\n0.0");
    }

    // 'If we receive Data with the GET Method, we process it'
    if ( isset ($_GET['Temp']) && isset($_GET['Hum'])){
        $var3 = $_GET['Temp'];
        $var4 = $_GET['Hum'];
        $fileContent  = $var3 . "\r\n" . $var4;
         $fileSave  =  file_put_contents ("miTemp&Hum.txt", $fileContent);
    }

    // 'We read the data in the  file  to save them in variables'
     $fileStr  =  file_get_contents ("miTemp&Hum.txt");
    $pos1 = strpos($fileStr ,  "\r\n");
    $var1 = substr($fileStr,  0,  $pos1);
    $var2 = substr($fileStr,  $pos1 + 1);
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="refresh" content="15">
    <title>Exercise6</title>

</head>
    <style>
        h1 {
            color: antiquewhite;
            background-color: dodgerblue;
            text-align: center;
        }
    </style>
<body>
    <header>
        <h1>Welcome to UEYT</h1>
    </header>

    <section>
        <h3>Temperature: <?php echo $var1; ?></h3>
        <h3>Humedad: <?php echo $var2; ?></h3>
    </section>
</body>
</html>
```

Programming ESP8266

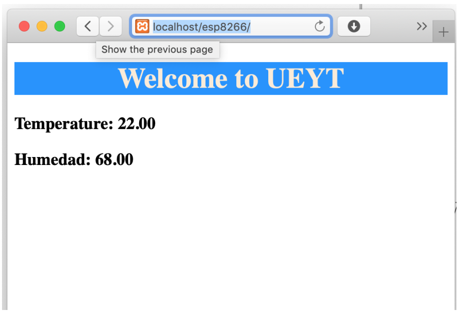
Open up the Arduino IDE and write the following code into a new sketch:

Code

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

//'Parameters WIFI'
const char* ssid = "<your network>";
const char* password = "<your password>";

const char* host = "192.168.0.102"; //'localhost -IP from your PC'


/*** 'Sensor model' ***/
#define DHTTYPE DHT11 //DHT21, DHT22

/*** Pin GPIO2***/
#define DHTPIN 2 // GPIO2

DHT dht(DHTPIN, DHTTYPE, 27);

/*** 'Variables for Humidity and Temperature' ****/
float temperature;
float humidity;

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

  dht.begin();

  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" connected");
}

void loop()
{
  WiFiClient client;

  Serial.printf("\n[Connecting to %s ... ", host);
  if (client.connect(host, 80))
  {
    Serial.println("connected]");

    temperature = dht.readTemperature();
    humidity = dht.readHumidity();

    Serial.println("[Sending a request]");
    client.print(String("GET /esp8266/?Temp=") + temperature + "&Hum=" + humidity + "
          HTTP/1.1\r\n" +
                "Host: " + host + "\r\n" +
                "Connection: close\r\n" +
                "\r\n"
               );

    Serial.println("[Response:]");
    while (client.connected())
    {
      if (client.available())
      {
        String line = client.readStringUntil('\n');
        Serial.println(line);
      }
    }
    client.stop();
    Serial.println("\n[Disconnected]");
  }
  else
  {
    Serial.println("connection failed!]");
    client.stop();
  }
  delay(5000);
}
```

Output Result

Once you have successfully set up the circuit connections and your arduino code has been uploaded, you can connect a power source to your arduino board. Remember to check the port, click `Tools -> Port` and select the COM port which your WiFi module is connected. See the result on the bowser online `http://localhost/esp8266/`



Welcome to UEYT

Temperature: 22.00

Humedad: 68.00

Figure 4: Web Browser `http://localhost/esp8266/`

Exercise 7. Humidity & Temperature Monitoring using DHT11 & 8266 on ThingSpeak

In the exercise you must use Thingspeak as cloud service provider and DHT11 to measure temperature and humidity. Thingspeak is the easiest way to get Arduino devices connected to ThingSpeak IoT services.

Exercise 8. Humidity & Temperature Monitoring using DHT11 & 8266 on Blynk app

In this exercise using an esp8266, to show the temperature and humidity DHT11 sensor on your Smartphone or tablet. The ESP8266 collects the temperature and humidity from DHT11 sensor and sends it to Blynk app every second.

Excersice 9. Hand-in (at the start of next lab)

1. Write a report considering the following format
   (a) Course Title, Lab no, Lab title, your name, and date.
   (b) Section on the lab experiment:
      - Insert all the generated graphics
      - Share your open source code via GitHub
   (c) The lab report is an important part of the laboratory. The report is individual. Write it carefully, be clear and well organized using $\LaTeX$.