

Chapter 5: Interacting with Web Services

204335 : Microcontroller and IoT

Part: ESP8266

Suphakit Awiphan

Chiang Mai University

Chapter 5: Interacting with Web Services

204335 Microcontroller and IoT

Chapter 5 – Interacting with Web Services

Parts You'll Need for This Chapter

- ESP8266 board
- USB cable
- DHT11
- Photoresistor
- 1 k Ω resistor
- Breadboard
- Jumper wires

Outline

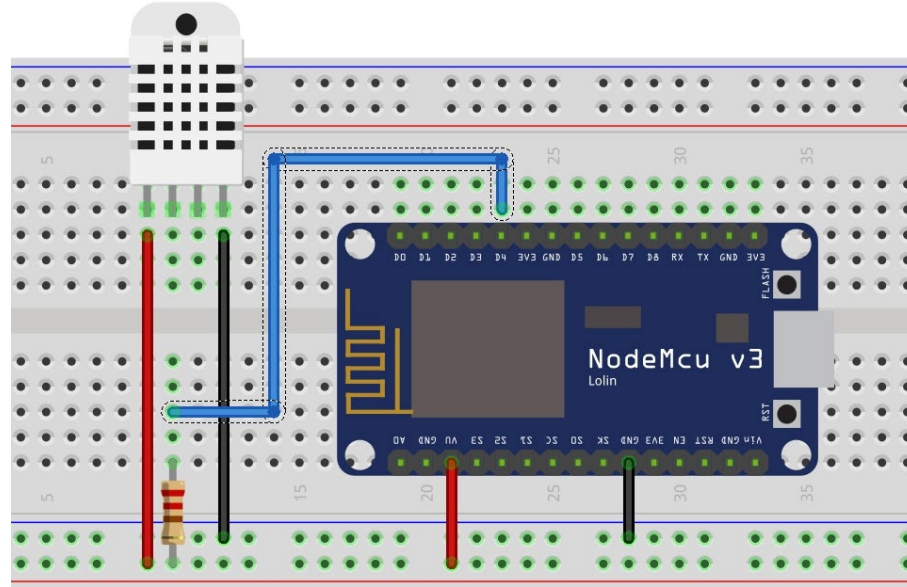
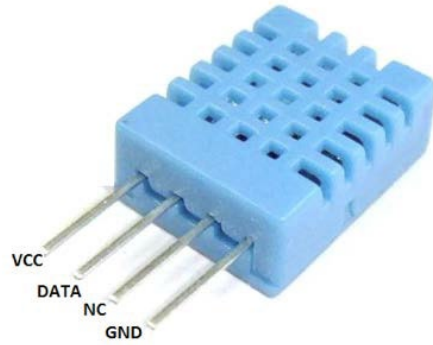
- **Automation with IFTTT**
- **Sending push notifications**
- **Sending e-mail notifications**
- **Sending text message notifications**

5.1 Automation with IFTTT

IFTTT

- If This Then That (IFTTT) is an online service that allows users to create simple conditional statements called applets
- The applets are triggered depending on changes made on or to other web services
- For instance, you can set an IFTTT applet that sends you an e-mail if a user includes a certain hashtag in their tweets

Getting Ready



IFTTT

- 1) Click on the drop-down menu on the top-right corner of the page and select New Applet. Click on this on the page that appears:

if  this then that

Select the Maker channel that has this icon



IFTTT

- 2) Click on the Connect button on the page that appears. This leads you to a page that prompts you to create a trigger that fires every time Maker service gets a web request.
- 3) Click on Receive a web request and specify the event name which will be sending a web request to Maker service. We will call our temperature_low.
- 4) Once you have entered the name, click on the Create trigger button.
- 5) 5. Now proceed to create the action to be taken when the trigger is fired. To do that, click on the that link:

IFTTT

1) Include the ESP8266 library and the DHT library:

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

```
#include "DHT.h"
```

```
//Set time between each trigger. Currently set to 30 seconds:
```

```
#define timeInterval 30000          // time between each trigger
```

IFTTT

2) Define the temperature sensor signal pin and sensor type, and create an object of the DHT library:

```
#define DHTPIN 2          // what digital pin we're connected to
#define DHTTYPE DHT11     // DHT 11

DHT dht(DHTPIN, DHTTYPE);

//Set Wi-Fi network credentials:
const char* ssid = "ssid";
const char* password = "pass";

//Set IFTTT requirements:
const char* host = "maker.ifttt.com";
const char* privateKey = "bqT9lyXcojqIexd8DiYk1F";
const char* event = "temperature_low";
```

IFTTT

3) Variable to hold previous time trigger was fired:

```
long lastTime = 0; // holds previous time trigger was sent
                    // Initialize serial communication and the dht sensor and connect
                    // ESP8266 module to the Wi-Fi network:

void setup() {
  Serial.begin(115200); // initialize serial communication
  dht.begin();          // initialize DHT11 sensor
  delay(100);

  // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

IFTTT

4) Get the temperature measurement from the dht sensor:

```
void loop() {  
  // Read sensor inputs  
  // get temperature reading in Celsius  
  float temperature = dht.readTemperature();  
  
  // Check if any reads failed and exit early (to try again).  
  while (isnan(temperature)) {  
    Serial.println("Failed to read from DHT sensor!");  
    delay(2000); // delay before next measurements  
    //get the measurements once more  
    temperature = dht.readTemperature();  
  }  
}
```

IFTTT

5) Check if the temperature has gone down by 30 degrees Celsius and if the last time the trigger was fired was 30 seconds ago. If those conditions are both true, the trigger can be fired:

```
if(temperature < 30 && millis()- lastTime > timeInterval){ //temperature is less than 30 deg celsius
    Serial.print("connecting to ");
    Serial.println(host);
```

6) Connect to the host server:

```
// Use WiFiClient class to create TCP connections
WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
}
```

IFTTT

7) Create the URL that will be used to fire the trigger:

```
// We now create a URI for the request
String url = "/trigger/";
url += event;
url += "/with/key/";
url += privateKey;
url += "?value1=";
url += String(temperature);
Serial.print("Requesting URL: ");
Serial.println(url);
```

IFTTT

8) Send HTTP request to fire the trigger and read the incoming response from the server:

```
// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}
// Read all the lines of the reply from server and print them to Serial
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
}
Serial.println();
Serial.println("closing connection");
lastTime = millis(); // save time of last trigger
}
```