GW Tech Collective Wi-Fi Notification System

Brandon Waller

April 14, 2018

Contents

1 Introduction

In this workshop, you will create a micro controller based system that interacts with If This Then That, and dweet.io to display incoming notifications using an RGB LED. The code provided can easily be extended to use other If This Then Thatservices.

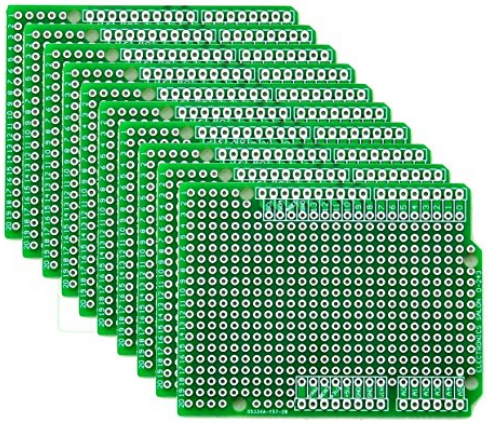
1.1 Materials

1.1.1 Micro Controller - ESP8266 (Wemos D1)

In this workshop, we will be using an ESP8266 micro controller on the same platform as the Arduino Uno. The ESP8266 is used over the Arduino Uno because of its built in Wi-Fimodule, and the Wemos D1 board makes it easier to work with over a normal ESP8266. Additionally, the ESP8266 can be programmed in the Arduino IDE using most of the standard Arduino library. There are a few differences in programming, including the pin name. In future versions of this workshop, the 8266 can be changed to the newer ESP32.

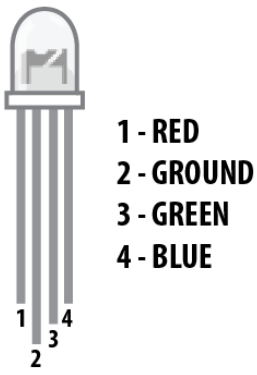
1.1.2 PCB

Because the Wemos D1 uses the Arduino Uno platform, we can use an Arduino PCB shield.



1.1.3 Components

To display notifications, an RGB led will be used, additionally other devices such as piezo buzzers can be used. A button will be used to dismiss all notifications. Finally, a temperature sensor will be used to demonstrate sending notifications from the micro controller to the web server.



1.2 Web Services

1.2.1 If This Then That

If This Then That[1] is a free service that connects to other services that causes something to happen if something happens on that service. In If This Then That, you create applets with an input condition, and an output. For example, if a video is posted on Youtube, send an Email. We will use this to send incoming notifications to a web server.

1.2.2 dweet.io

If This Then ThatCannot send notifications directly to the micro controller, so we will use dweet.io to create an interface between If This Then That, and the micro controller. dweet.io is a public web server that any interconnected device can use. If This Then Thatcan send notifications to dweet.io, that the micro controller can read. Anything posted to dweet will be public, so you should use an obscure thing name, and notification names.

2 dweet.io Setup

The first step to create your notification system is to find an ID dweet.io that isn't in use. To do this, choose a name, in this document, `ObscureThingName`. Goto `dweet.io/get/latest/dweet/for/[Thing Name]` (`dweet.io/get/latest/dweet/for/ObscureThingName`). If on the web page, the output is similar to `"this":"failed","with":404,"because":"we couldn't find this"`, then the thing name you chose is most likely not in use, and should be good to use. Note that all dweet.io pages are reset after a certain amount of time, so there is a chance that someone else is using the same thing name as you.

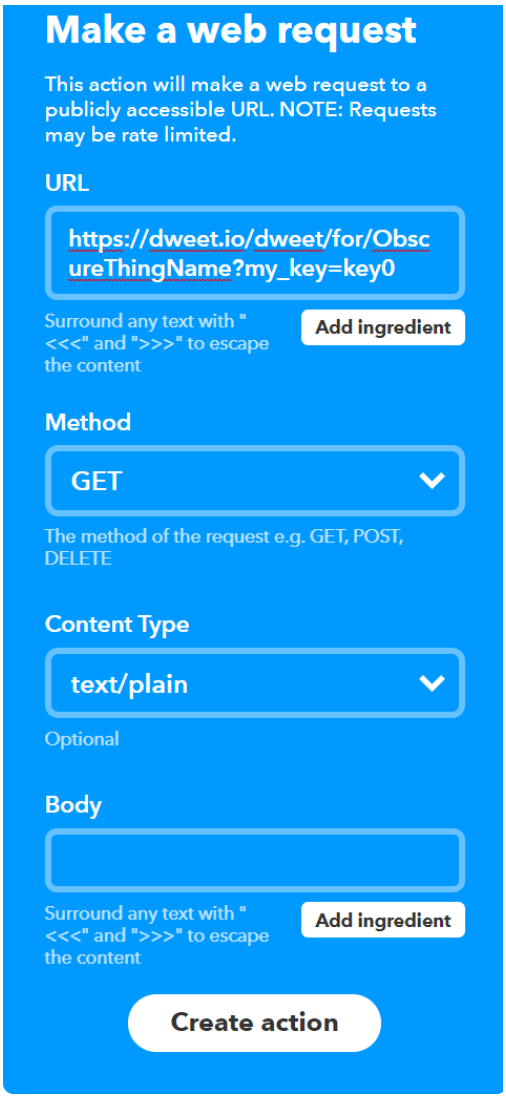
3 If This Then ThatSetup

After you have a dweet thing name, you can then set up the triggers for your notification system.
Setting up Gmail Trigger:

- Goto ifttt.com, and create an account. Press the 'x' in the top right corner when you get the **Welcome to IFTTT!** page.
- On the top, click on My Applets, then New Applet
- Click on +this > Gmail > Connect > Desired trigger

[1] ifttt.com

- Click on +that > Webhooks > Connect > Make a web request
  - URL: `https://dweet.io/dweet/for/[Thing Name]?my_key=[key name]`[2]
  - Method: Get
  - Content Type: text/plain
  - Leave Body Blank
- Create action



[2]key will be discussed later

4 Micro Controller Programming

4.1 Installing Arduino Libraries

Instructions for Windows, may vary for Mac
Steps to install the ESP8266 library for the Arduino IDE:

- In the Arduino IDE, File > Preferences > Additional Board Manager URLs: `http://arduino.esp8266.com/stable/package_esp8266com_index.json`
- Tools > Boards > Boards Manager > Search: ESP8266 > Install
- Tools > Boards > Wemos D1 (Retired)
- Optional, test one of the ESP8266 example sketches

Installing the `dweet-esp` Library:

- goto https://github.com/gamo256/dweet-esp
- Clone or download > Download ZIP > Save zip folder
- Sketch > Include Library > Add .zip Library > Choose zip folder

5 Code Explained

The code to read, and respond to notifications has 3 main components:

- Read dweet.io for posts
- Test if the post contains notification
- Respond
  - Do something the RGB LED
  - Send a post to dweet.io to dismiss the notification

5.1 Micro Controller Code

In this section, parts of the code will be explained, the full code is available on github.

Reading dweet

```
#include "dweetESP8266.h"

// param.h
#define NUM_NOTIFICATIONS 2
String keys[NUM_NOTIFICATIONS] = {"key0", "key1"}
#define KEY0 0
#define KEY1 1
#define THING "[thing_name]"
#define DELAY 3000

void run_dweet()
{
  // Poll the dweet when DELAY ms passes
```

```
  if(millis() - poll_time > DELAY)
  {
    // get value from dweet with key "my_key"
    String value = client.getDweet(THING, "my_key");
    if(value == keys[KEY0])
    {
      // check if key already has active notification
      if(!active[KEY0])
      {
        notifications++;
        active[KEY0] = 1;
      }
      // send message to dweet to dismiss the notification
      // on dweet
      reset_dweet();
    }
    ... // repeat for other keys

    poll_time = millis(); // reset poll time
  }
}
```

This code reads dweet, and checks if there is something after `my_key` in dweet's JSON. If there is, check if it matches any of the key values your checking for. If it does, check if you already processed a notification from that service. If not, set set the flag for the key to 1.

Putting Notifications on the RGB LED:

```
int j = 0;

void run_led()
{
  if(millis() - led_time) > 500)
  {
    // if the jth service has a notification, turn on the RGB LED
    if(active[j])
    {
      // RED, GREEN, BLUE are pin numbers for each color
      // red[], green[], and blue[] are rgb color arrays
      analogWrite(RED, red[j]);
      analogWrite(GREEN, green[j]);
      analogWrite(BLUE, blue[j]);
    }
    else if(notifications == 0)
    {
      // turn off all LEDs
    }

    j = (j + 1) % NUM_NOTIFICATIONS;
  }
}
```

Using the active array in the previous parts, the micro controller knows which notifications it should show on the RGB LED.