

Home-Automation-Using-Google-Assistant

INSTRUCTIONS

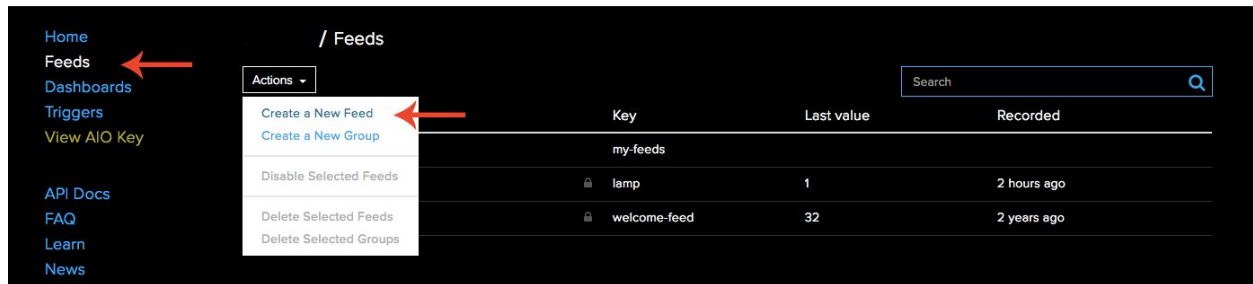
1- Creating Adafruit IO Account, Dashboard and feeds

Adafruit IO is an IOT platform built around the Message Queue Telemetry Transport (MQTT) Protocol. MQTT is a lightweight protocol that allows multiple devices to connect to a shared server, called the MQTT Broker, and subscribe or write to user defined topics. When a device is subscribed to a topic, the broker will send it a notification whenever that topic changes. MQTT is best suited for applications with low data rates, strict power constraints, or slow Internet connections.

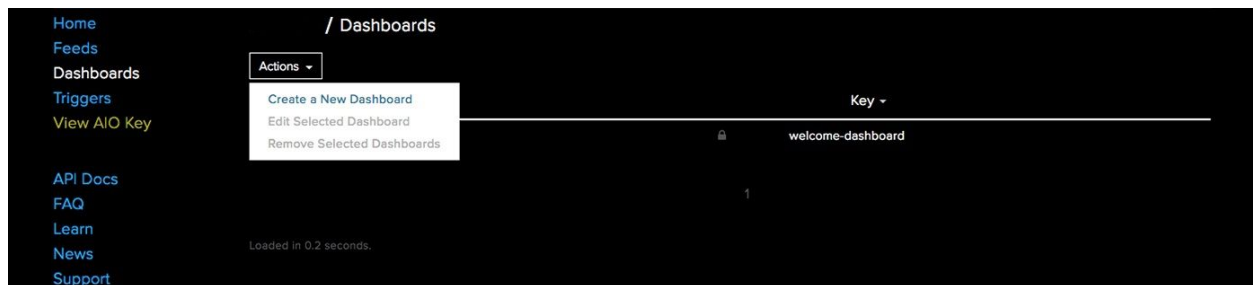
In addition to providing the MQTT Broker service, Adafruit IO also allows you to set up dashboards that let you directly manipulate or view the current value of each topic. Since it can be accessed from a web browser, it makes it the ideal hub for monitoring and controlling all of your various IOT projects.



create your Adafruit IO account after that you should be taken to the home screen. Select “Feeds” from the left-hand menu. Click the Actions drop-down menu, and create a new feed. I called mine “Lamp”.



Next, go to Dashboards in the left-hand menu. Click the Actions drop-down menu, and create a new dashboard. I called mine “My Room”.



Open the new dashboard, and you should be taken to a mostly blank page.



Pressing the blue + button will let you add new UI components to the dashboard. For now, all we'll need is a toggle button, which should be the first option.

Choose feed

A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

Create

Group / Feed	Last value	Recorded
<input type="checkbox"/> Welcome Feed	32	2 years ago
<input checked="" type="checkbox"/> Lamp	OFF	3 minutes ago

< Previous step Next step >

When prompted to choose a feed, select the one you just made, and keep the settings of button as following.

Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title

Button On Text

Button Off Text
































Block Preview

Lamp

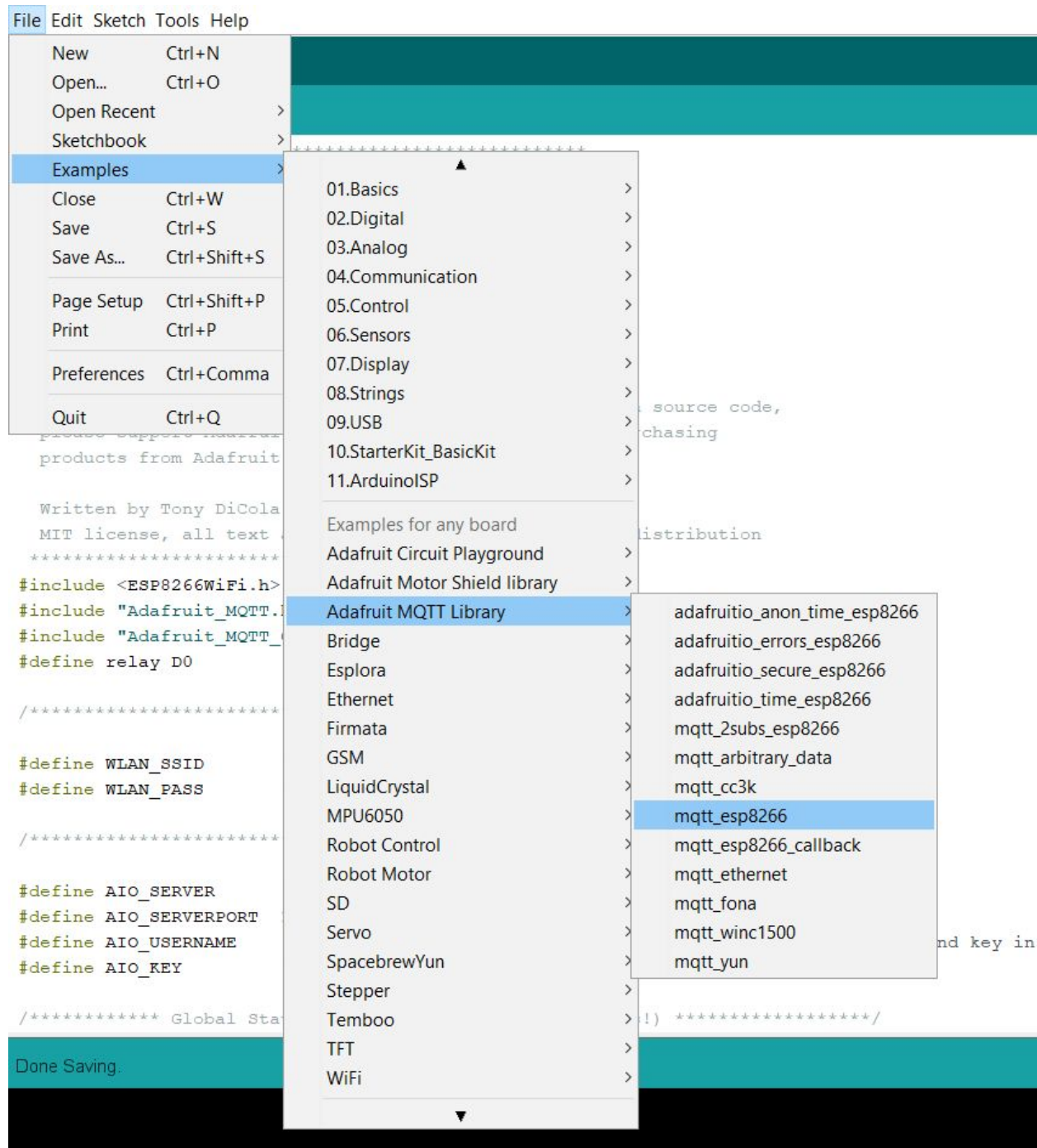


That's all for now on the Adafruit IO end of things

Download this repo and extract it to the arduino library folder

Name	Date modified	Type	Size
 Adafruit_Circuit_Playground	10/13/2018 11:29	File folder	
 Adafruit_MQTT_Library-master	10/13/2018 11:29	File folder	
 AFMotor	10/13/2018 11:29	File folder	
 Bridge	10/13/2018 11:29	File folder	
 Esploira	10/13/2018 11:29	File folder	
 Ethernet	10/13/2018 11:29	File folder	
 Firmata	10/13/2018 11:29	File folder	
 GSM	10/13/2018 11:29	File folder	
 I2Cdev	10/13/2018 11:29	File folder	
 Keyboard	10/13/2018 11:29	File folder	
 LiquidCrystal	10/13/2018 11:29	File folder	
 LMotorController	10/13/2018 11:29	File folder	
 Mouse	10/13/2018 11:29	File folder	
 MPU6050	10/13/2018 11:29	File folder	
 PID_v1	10/13/2018 11:29	File folder	
 Robot_Control	10/13/2018 11:29	File folder	
 Robot_Motor	10/13/2018 11:29	File folder	
 RobotIRremote	10/13/2018 11:29	File folder	
 SD	10/13/2018 11:29	File folder	
 self	10/13/2018 11:29	File folder	
 Servo	10/13/2018 11:29	File folder	
 SpacebrewYun	10/13/2018 11:29	File folder	
 Stepper	10/13/2018 11:29	File folder	
 Temboo	10/13/2018 11:29	File folder	
 TFT	10/13/2018 11:29	File folder	
 WiFi	10/13/2018 11:29	File folder	
 I2Cdev	4/25/2018 7:30 PM	C++ Source File	56 KB
 I2Cdev	4/25/2018 7:30 PM	C Header File	12 KB
 keywords	4/25/2018 7:30 PM	Text Document	1 KB
 library.json	4/25/2018 7:30 PM	JSON File	1 KB
 SoftwareSerial	9/10/2018 3:53 PM	C++ Source File	14 KB

Open this program from arduino example program



You need to edit this code

In this tutorial i will show you how to control one relay. You can control multiple relays by modifying this code

Firstly you need to define a variable for the output pin

```
#define relay D0 // I am using D0 as my output pin
```

Secondly add your WiFi SSID and password

Then add your AIO username and AIO key

```
Written by Tony DiCola for Adafruit Industries.
MIT license, all text above must be included in any redistribution
*****

#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#define relay D0

/***** WiFi Access Point *****/
#define WLAN_SSID      "Your WiFi SSID" // Repalce with your WiFi SSID
#define WLAN_PASS      "Your password" // Repalce with your WiFi password

/***** Adafruit.io Setup *****/

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883 // use 8883 for SSL
#define AIO_USERNAME    "*****" //you can find your AIO username and key in your adafruit account
#define AIO_KEY         "*****"

/***** Global State (you don't need to change this!) *****/

// Create an ESP8266 WiFiClient class to connect to the MQTT server.
WiFiClient client;
// or... use WiFiClientSecure for SSL
//WiFiClientSecure client;
```

In this line of code

```
Adafruit_MQTT_Subscribe onoffbutton = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/light1");
```

// Replace light1 with the feed name that you created

You can copy paste this line if you have more feeds(relays to control)

In the main function i edited the highlighted lines

```
void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(relay , OUTPUT);

  Serial.println(F("Adafruit MQTT demo"));

  // Connect to WiFi access point.
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);
```

```

WiFi.begin(WLAN_SSID, WLAN_PASS);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println();

Serial.println("WiFi connected");
Serial.println("IP address: "); Serial.println(WiFi.localIP());

// Setup MQTT subscription for onoff feed.
mqtt.subscribe(&onoffbutton);
}

uint32_t x=0;

void loop() {
  // Ensure the connection to the MQTT server is alive (this will make the first
  // connection and automatically reconnect when disconnected). See the MQTT_connect
  // function definition further below.
  MQTT_connect();

  // this is our 'wait for incoming subscription packets' busy subloop
  // try to spend your time here

  Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(5000))) {
    if (subscription == &onoffbutton) {
      Serial.print(F("Got: "));
      Serial.println((char *)onoffbutton.lastread);
      uint16_t state = atoi((char *)onoffbutton.lastread);
      digitalWrite(relay,state);
    }
  }
}

```

THIS IS PIECE OF CODE FROM THE MAIN CODE

THAT'S ALL

NOW YOU CAN BURN THE CODE IN TO YOUR NODE MCU

3- Connecting to Google Assistant Through IFTTT

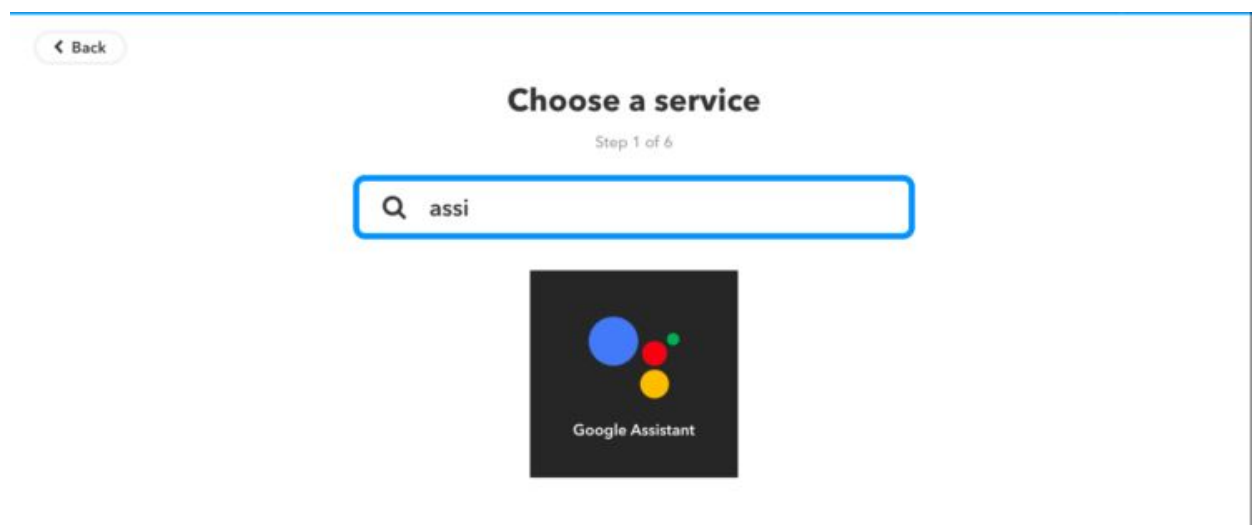
New Applet

if this then that

[Want to build your own service? Build on the platform](#) 

Now we'll connect our Google Assistant to the Adafruit IO MQTT Broker to allow us to control the lights with voice commands. To do this, we'll use the [IFTTT](#) (If This Then That) platform, which allows hundreds of different services to trigger actions in a variety of other services.

After you've set up your account and taken a look around, Select "[My Applets](#)" from the left hand menu, then click the blue "New Applet" button. This will take you to the applet editor, where you choose triggers ("If This") and the subsequent actions ("Then That").



For your trigger, choose "Google Assistant" as the service then select "Say a simple phrase" from the menu of specific triggers.



Complete trigger fields

Step 2 of 6

Say a simple phrase

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.

What do you want to say?

What's another way to say it? (optional)

And another way? (optional)

What do you want the Assistant to say in response?


Step 6 of 6



If You say "turn lamb on", then
send data to Lamp feed

54/140

by sobhydo

works with 

Receive notifications
when this Applet runs



Finish

This will bring up a new list of fields to fill in, including variations of the activation phrase, the Google Assistant's response, and the language. For my activation phrases, I chose "Turn lamp off" and "Switch lamp off" and you can add "Turn off lamp" too then click Next and Finish.

[< Back](#)

if  then  that

Your First Trig assigned to google assistant as you can see above .

[< Back](#)



Choose action

Step 4 of 6

Send data to Adafruit IO

This Action will send data to a feed in your Adafruit IO account.

[< Back](#)

Choose action service

Step 3 of 6

Q ad



Adafruit

The final part of your applet is the Action, what your applet does in response to the Trigger. For the service, choose “Adafruit”, and for the specific Action, choose “Send data to Adafruit IO”.



Complete action fields

Step 5 of 6

Send data to Adafruit IO

This Action will send data to a feed in your Adafruit IO account.

Feed name

Lamp



The name of the feed to save data to.

Data to save

0

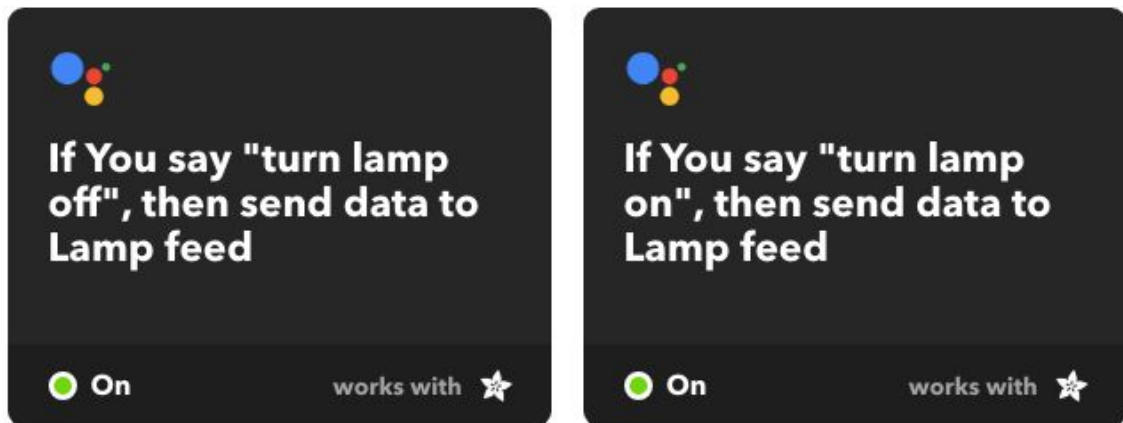
The data to be saved to your feed.

Add ingredient

Create action

This will bring up two fields that you need to fill in. The first should be replaced with the name of the Adafruit IO feed you want to send data to, in this case “Lamp”. The second field is the data to send. For this applet, we’ll send “0” Zero, which is the string our ESP8266 is waiting for.

Once you have that applet finished, create a second one for turning the lights “ON”. You should now see two applets on your IFTTT Platform page. To activate them, go to the [My Applets](#) page on the main IFTTT site, click on the applet card, and click set the on-off toggle switch to “On”. If you haven’t already, IFTTT will ask to connect to your Adafruit IO and Google Assistant accounts. Allow the accounts to be linked, then turn on the second applet as well.



Once both applets are turned on, the setup should be complete!