

GOOGLE ASSISTANT WITH ESP8266

Using **Adafruit IO** and **IFTTT** we can establish a IoT system which controlled by both commands and voices of users.

We are using two services for the google assistant integration with **ESP8266**.

They are:

1. **Adafruit IO**
2. **IFTTT**

Adafruit IO is a platform designed to *display, respond, and interact* with your project's data. Display your data in real-time, online Make your project internet-connected: Control motors, read sensor data, and more! Connect projects to web services like Twitter, RSS feeds, weather services, etc. Connect your project to other internet-enabled devices. Adafruit MQTT is widely used in IoT (Internet of Things) embedded applications, where every sensor is connected to a server and we have access to control them over the internet. The MQTT Client module of NodeMCU is according to version 3.1.1 of the MQTT protocol. Make sure that your broker supports and is correctly configured for version 3.1.1. let's see the functions used for MQTT on NodeMCU.

IFTTT defines a clear and concise protocol which your service's API will implement. Each trigger and action for your service will map 1-to-1 to an API endpoint on your service built specifically for IFTTT. Trigger endpoints will be event streams that IFTTT will poll for new data. Conversely, action endpoints will be writable endpoints that IFTTT will send data to. For services that use OAuth, IFTTT is fully compatible with a couple of the common flavours of OAuth 2.0.

Reference link for:

Adafruit IO creation : <https://learn.adafruit.com/mqtt-adafruit-io-and-you/getting-started-on-adafruit-io>


IFTTT : <https://learn.adafruit.com/using-ifttt-with-adafruit-io/ifttt-to-adafruit-io-setup>

For **Adafruit IO**, please remember these two parameters along with your **username**:

- **Key** - this is a long, unique identifier that you use to authenticate any devices using your account. This is your password! Keep it safe! You get one key per account, but you can, at any time revoke and regenerate your key.
- **Feed** - this is basically a set of data that you can read or write from like a sequential file. There is some history stored with feeds, with MQTT you cannot access historical data (REST does support it) but you can add data and you can receive the latest added data. You can create two feeds for both **publish** and **subscribe**.

SOURCE CODE MODIFICATION:

Open the source code and do changes in **ada.h** with your Adafruit IO **USERNAME & KEY**.



```
1 #ifndef __ADA_H
2 #define __ADA_H
3
4 #define AIO_SERVER          "io.adafruit.com"
5 #define AIO_SERVERPORT     1883
6 #define AIO_USERNAME       "ADAFRUIT IO USERNAME"
7 #define AIO_KEY            "ADAFRUIT IO KEY"
8
```

Change the **ssid** and **password** in **global.h**



```
1 #ifndef __GLOBAL_H
2 #define __GLOBAL_H
3
4 #define WLAN_SSID "YOUR WIFI SSID"
5 #define WLAN_PASS "YOUR WIFI PASSWORD"
6
7 #endif
```

Paste the **publish** and **subscribe feed** name in **ada.h** to the marked space.:



```
1 #include <ESP8266WiFi.h>
2 #include "Adafruit_MQTT.h"
3 #include "Adafruit_MQTT_Client.h"
4 #include "ada.h"
5
6 /*+++++Obj declaration+++++*/
7 WiFiClient Ada;
8 Adafruit_MQTT_Client Amqtt(sAda, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
9 Adafruit_MQTT_Subscribe Rec = Adafruit_MQTT_Subscribe(sAmqtt, AIO_USERNAME"/feeds/SUBSCRIBE FEED NAME FROM IFITTT");
10 Adafruit_MQTT_Publish Txmss = Adafruit_MQTT_Publish(sAmqtt, AIO_USERNAME"/feeds/SUBSCRIBE FEED NAME TO ADAFRUIT");
11 /*+++++*/
12
13 /*+++++Feedbck declaration+++++*/
14 char Feedback_on[] = "Hey!! Light is ON";
```

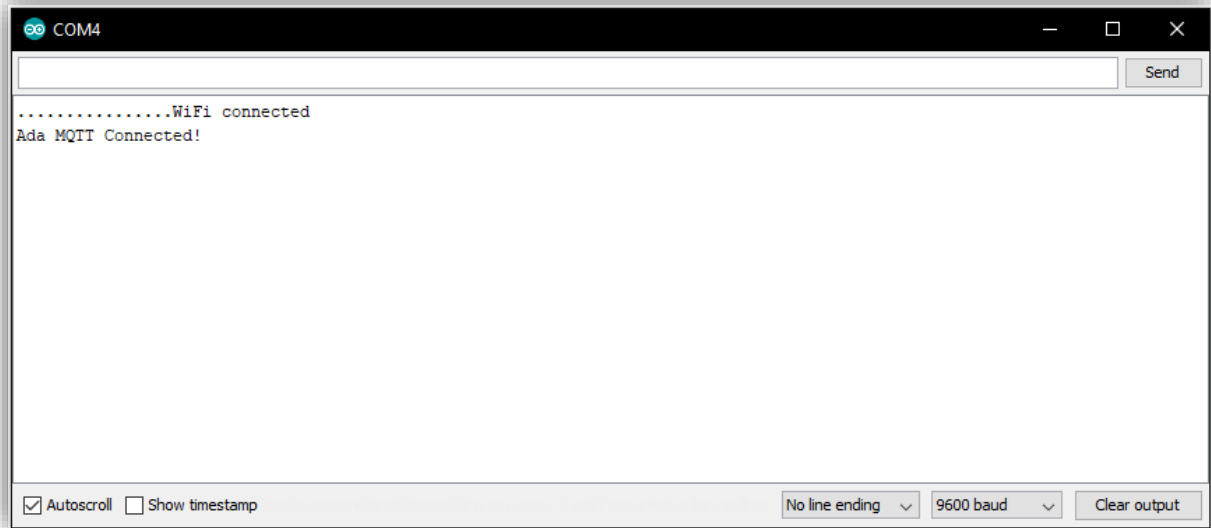
TESTING & DEBUGGING:

SERIAL MONITOR OUTPUT:

Power on the esp8266 and flash the source code with above mentioned changes

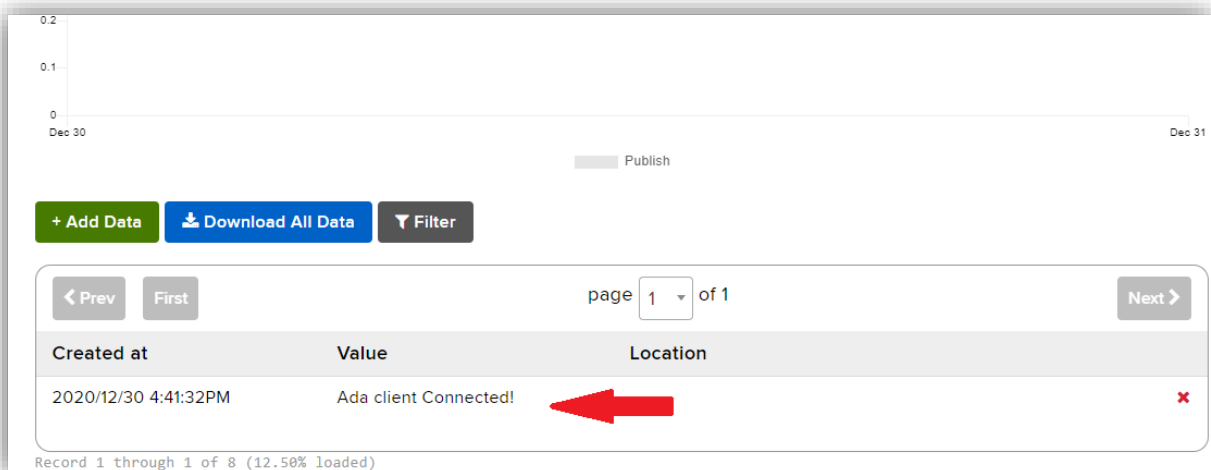
When WIFI gets connected we get a serial print: ***"WIFI connected"***

When ESP8266 establishes cloud connection with adafruit io then we get: ***"Ada MQTT Connected!"***



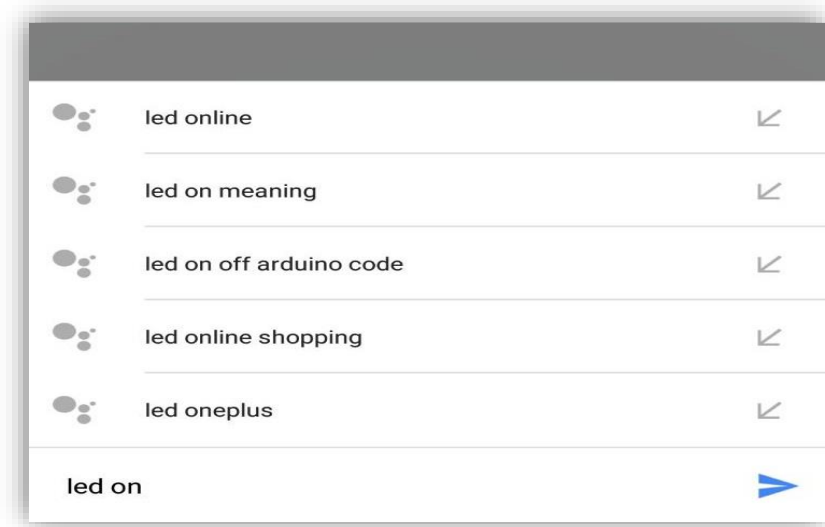
ADAFRUIT IO OUTPUT:

After ***WIFI*** gets connected we could see an acknowledgement message in the dashboard [On ***publish feed***].

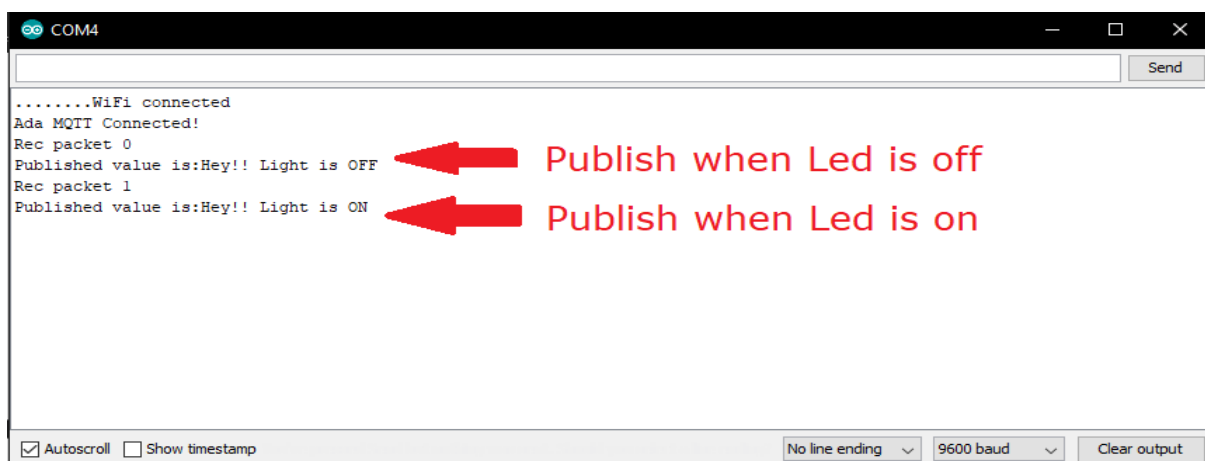


We can send commands to turn **on & off** lights, which connected to **esp8266** via **google assistant**:

When we type "**led on**" in the assistant window [**"led off**" can put like this]:



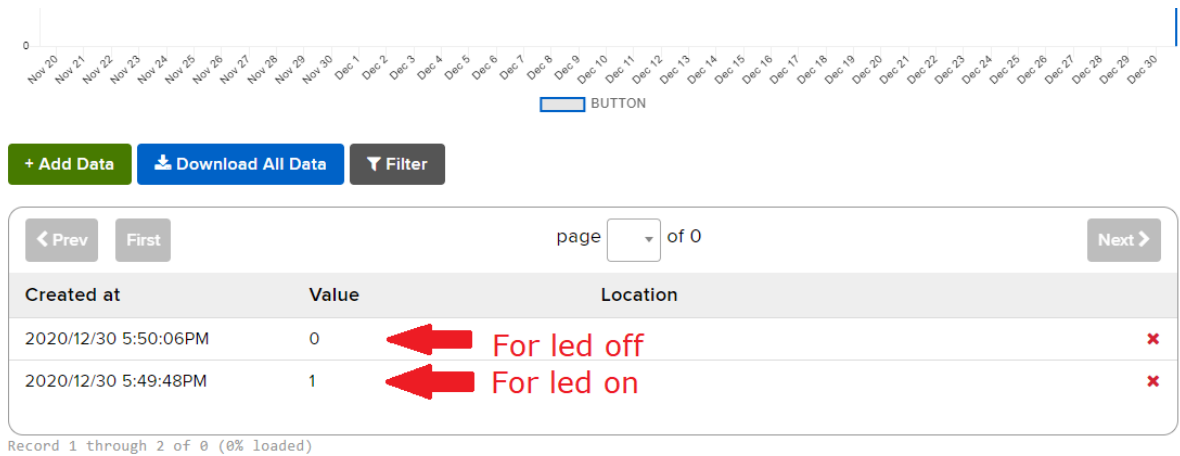
We can see the **received packet** in digital form (**0 or 1**) and its published message via serial monitor debugging.



At the same time **ADAFRUIT IO OUTPUT** on the publish feed is :

0	Dec 30	Dec 31
Publish		
+ Add Data Download All Data Filter		
page 1 of 1		
Created at	Value	Location
2020/12/30 5:53:21PM	Hey!! Light is ON	Acknowledgement from esp8266 on the basis of led status
2020/12/30 5:52:50PM	Hey!! Light is OFF	
2020/12/30 5:52:45PM	Ada client Connected!	

on the subscribe feed we can see the digital values for *on* and *off* from *google assistant* via *IFTTT* :



Thank you!!

≠