# NODEMCU

By Muhammad Tabish

# What is IOT?

> “The Internet of Things is the concept of connecting any device to the Internet and to other connected devices

# Internet of Things

# What is NodeMCU?

- NodeMCU is an open source IoT platform.
- It is built around the ESP8266 Module
- Programmed using Lua or Python in addition to Arduino

# PROS AND CONS

## PROS

- Open-source
- Programmable
- Inexpensive
- All IO pins are PWM enabled
- Compact

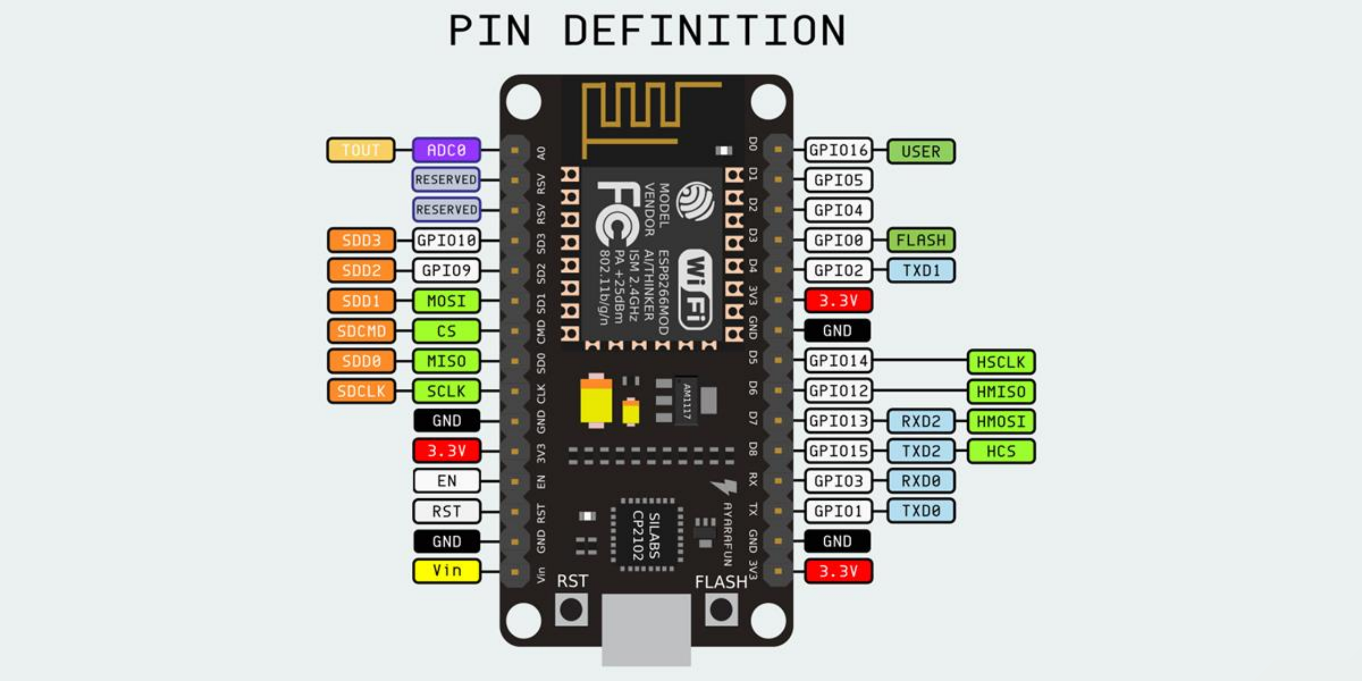
## CONS

- Can only produce 3.3v output
- Slightly longer upload time

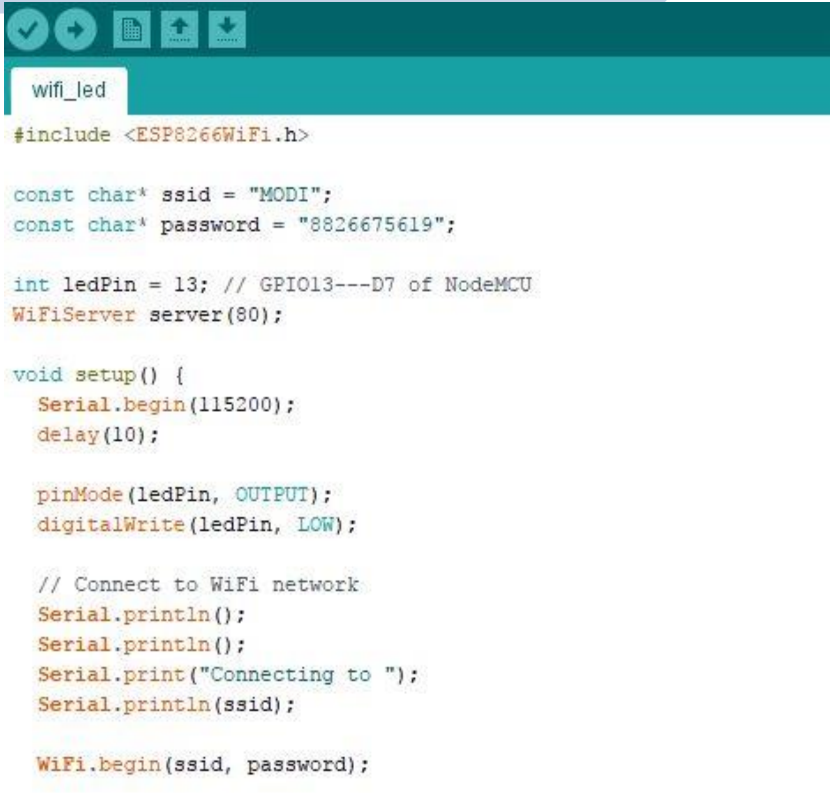
# Getting Started

- Download required drivers
- Download NodeMCU packages and ESP8266 libraries in Arduino IDE
- You can now program the NodeMCU

# Getting Started

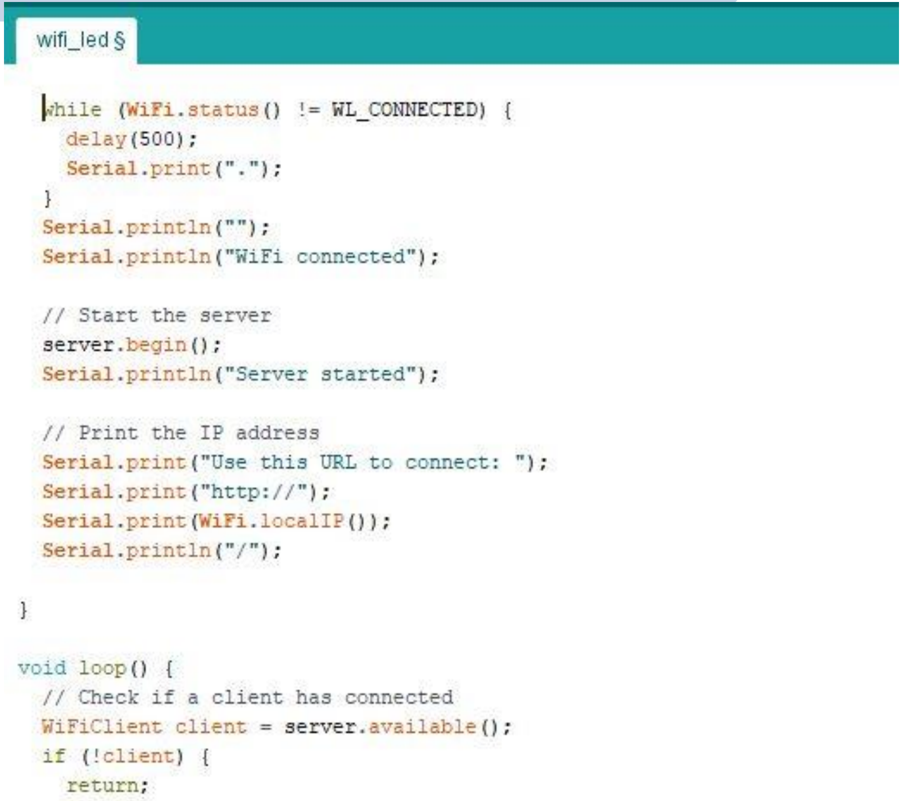


# Turn On/Off Led from the Internet



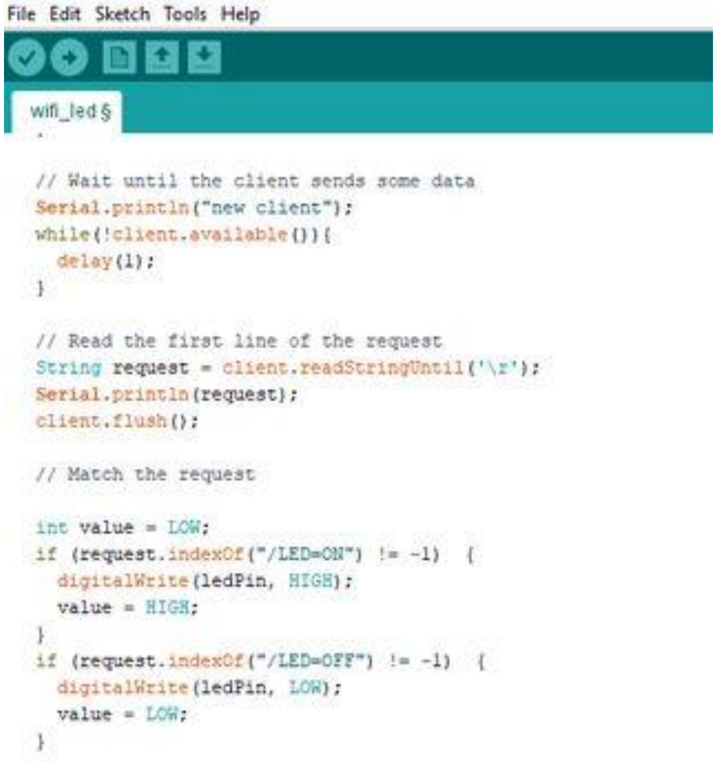
```
#include <ESP8266WiFi.h>

const char* ssid = "MODI";
const char* password = "8826675619";

int ledPin = 13; // GPIO13---D7 of NodeMCU
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
```

# Turn On/Off Led from the Internet



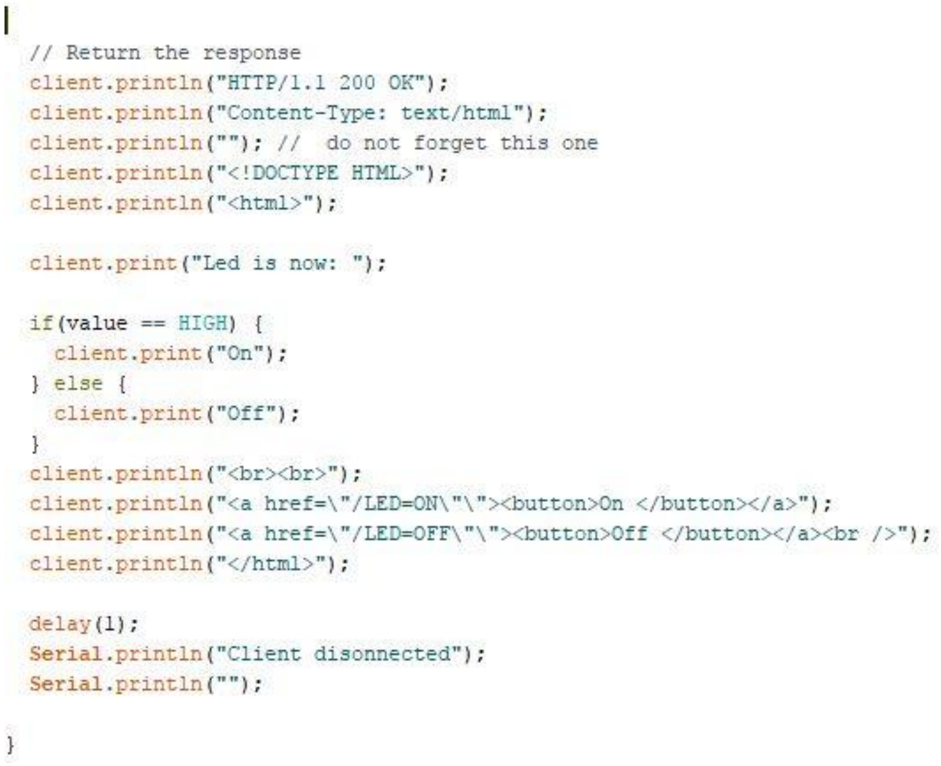
```
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
```

# Turn On/Off Led from the Internet



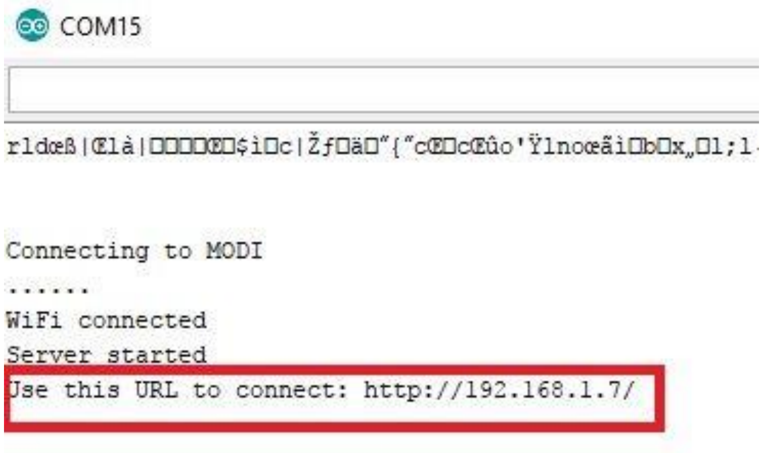
```
// Wait until the client sends some data
Serial.println("new client");
while(!client.available()){
  delay(1);
}

// Read the first line of the request
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();

// Match the request

int value = LOW;
if (request.indexOf("/LED=ON") != -1)  {
  digitalWrite(ledPin, HIGH);
  value = HIGH;
}
if (request.indexOf("/LED=OFF") != -1)  {
  digitalWrite(ledPin, LOW);
  value = LOW;
}
```

# Turn On/Off Led from the Internet

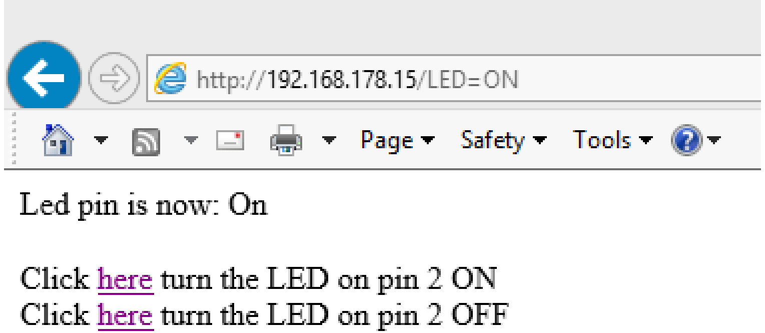


```
  // Return the response
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println(""); //  do not forget this one
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");

  client.print("Led is now: ");

  if(value == HIGH) {
    client.print("On");
  } else {
    client.print("Off");
  }
  client.println("<br><br>");
  client.println("<a href=\"/LED=ON\"\"><button>On </button></a>");
  client.println("<a href=\"/LED=OFF\"\"><button>Off </button></a><br />");
  client.println("</html>");

  delay(1);
  Serial.println("Client disonnected");
  Serial.println("");

}
```

# Turn On/Off Led from the Internet



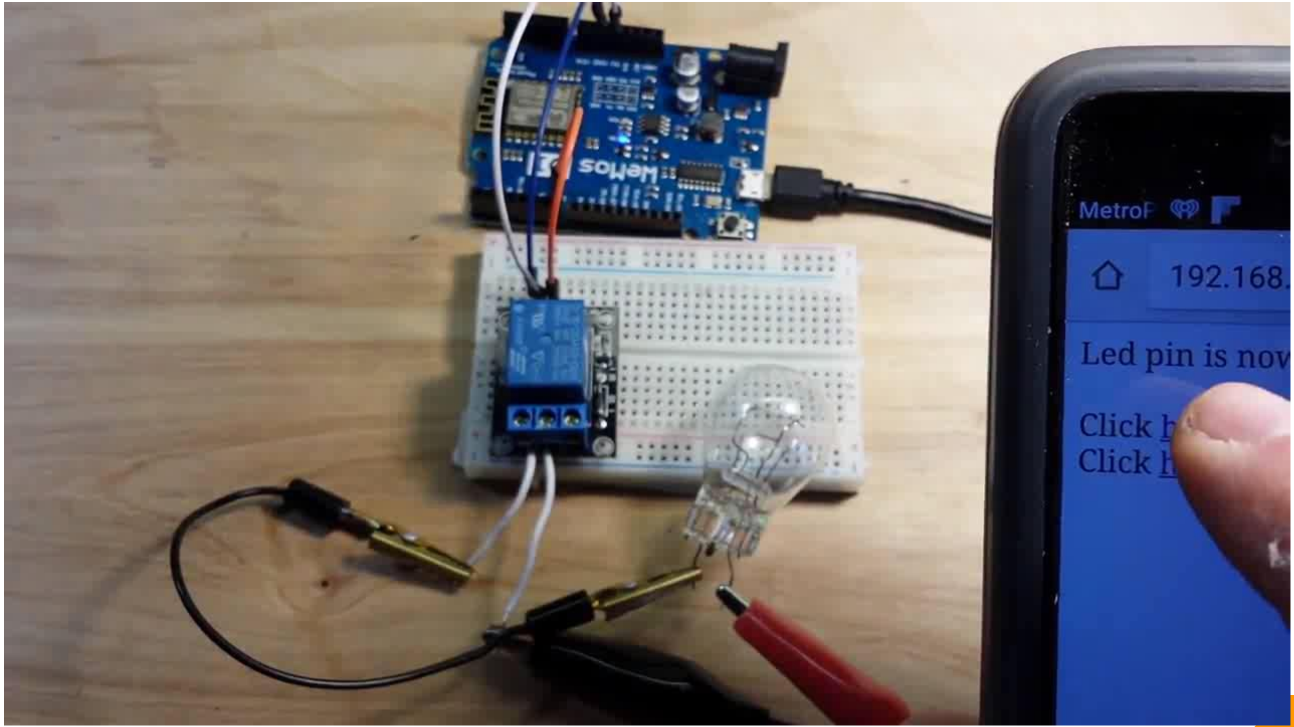
```
rldœß|Œlà|□□□□Œ□$ì□c|Žƒ□ä□"{"cŒ□cŒûo'Ÿlnoœãi□b□x„□l;l

Connecting to MODI
.......
WiFi connected
Server started
Use this URL to connect: http://192.168.1.7/
```



Led pin is now: On

Click here turn the LED on pin 2 ON
Click here turn the LED on pin 2 OFF

# Turn On/Off Led from the Internet

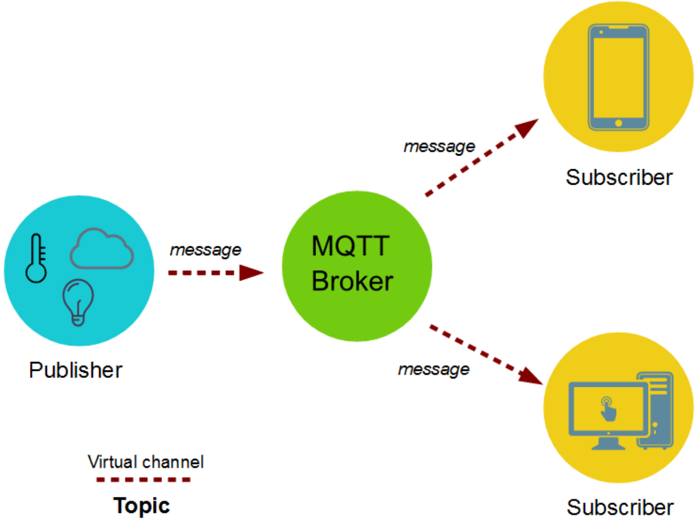


# MQTT Protocol

- MQTT (Message Queue Telemetry Transport) is a publish-subscribe messaging protocol widely used in IoT applications.
- It is highly recommended for microcontroller projects that sends data over the internet.

# MQTT Protocol

- Communication through MQTT needs a "broker" which is responsible for distributing messages to clients. A publisher sends data to the broker while subscribers reads data from the broker



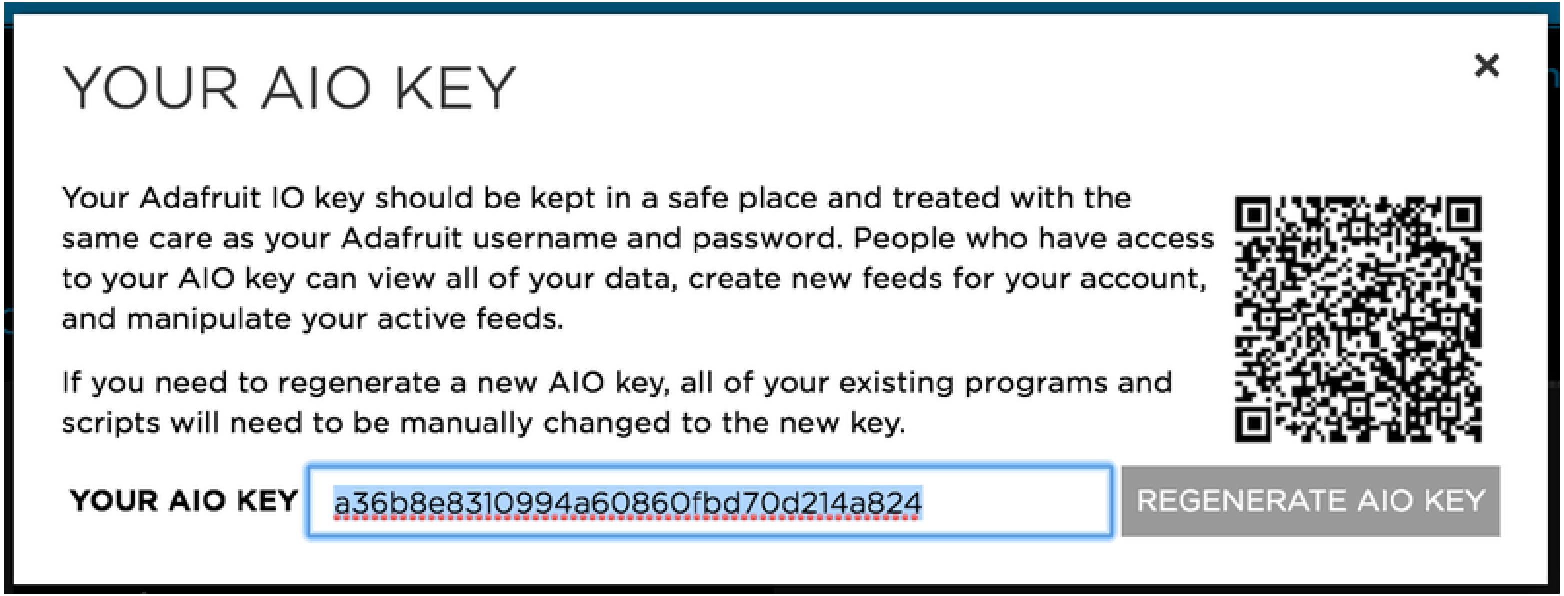
# MQTT Broker Apps

- Thingspeak
- Adafruit IO
- Mosquitto
- HiveMQ
- shiftr.io

# Getting Started with Adafruit IO

- Install required Adafruit IO, Adafruit MQTT and ArduinoHttpClient Libraries in Arduino IDE
- Get AIO Key from your Adafruit IO account
- Create a new feed and add a toggle block
- Open digital out example sketch in Arduino IDE and edit your wifi and Adafruit IO credentials.

# Adafruit IO



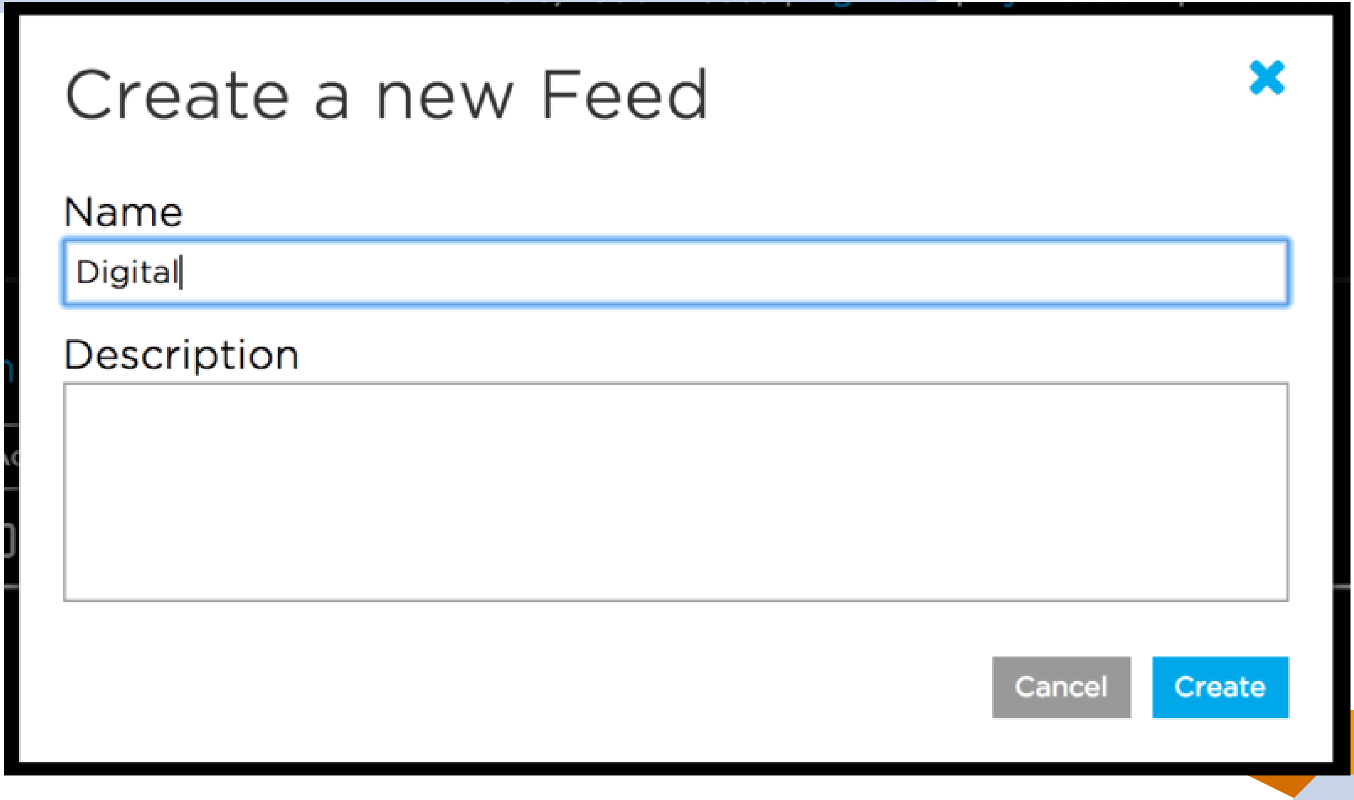
YOUR AIO KEY

Your Adafruit IO key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your AIO key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new AIO key, all of your existing programs and scripts will need to be manually changed to the new key.

**YOUR AIO KEY** a36b8e8310994a60860fbd70d214a824

# Adafruit IO



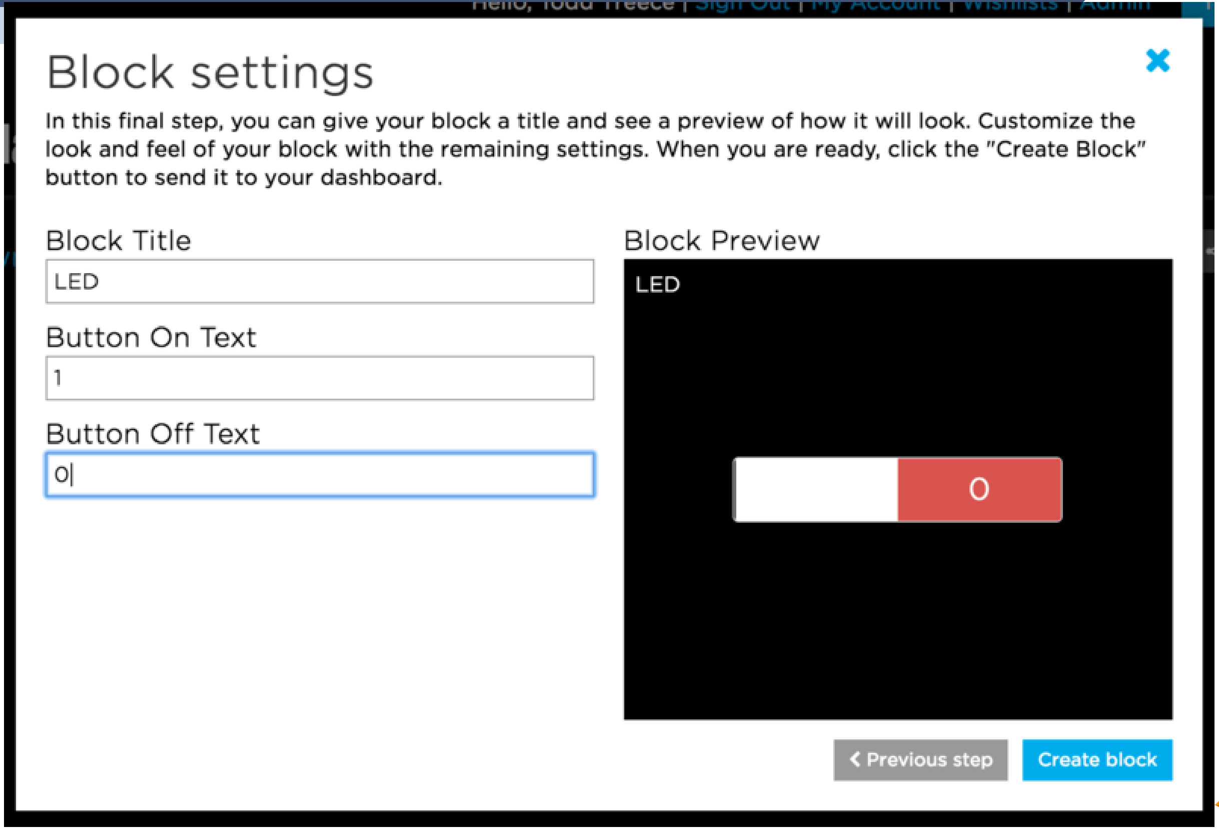
Create a new Feed

Name

Digital

Description

# Adafruit IO



Hello, Todd Treece | Sign Out | My Account | Wishlists | Admin

## Block settings

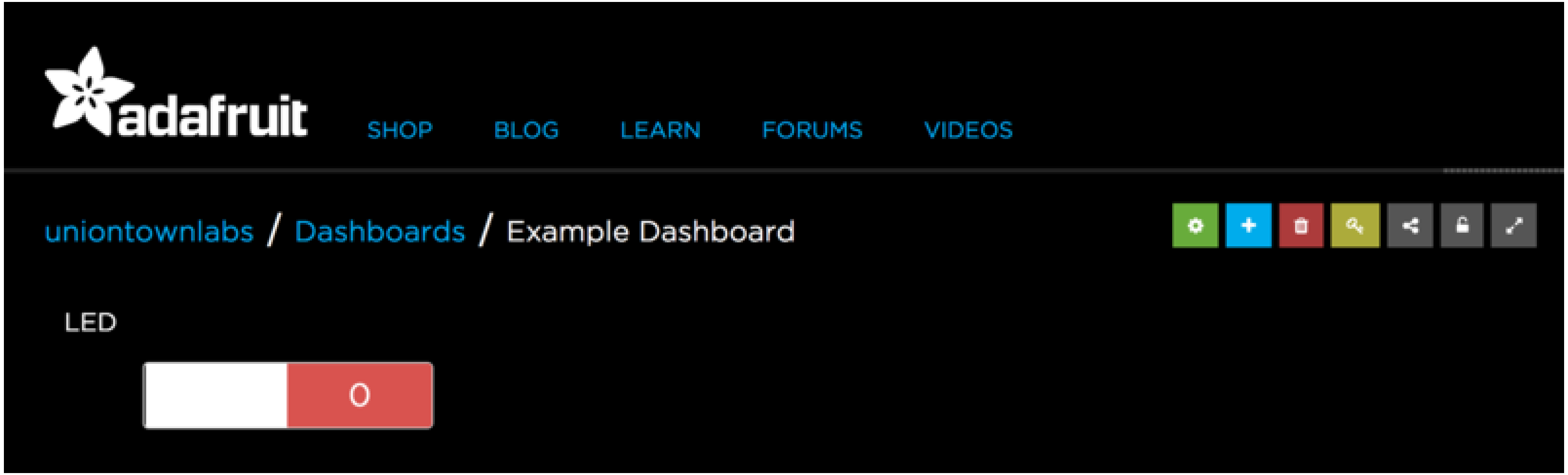
In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title: LED

Button On Text: 1

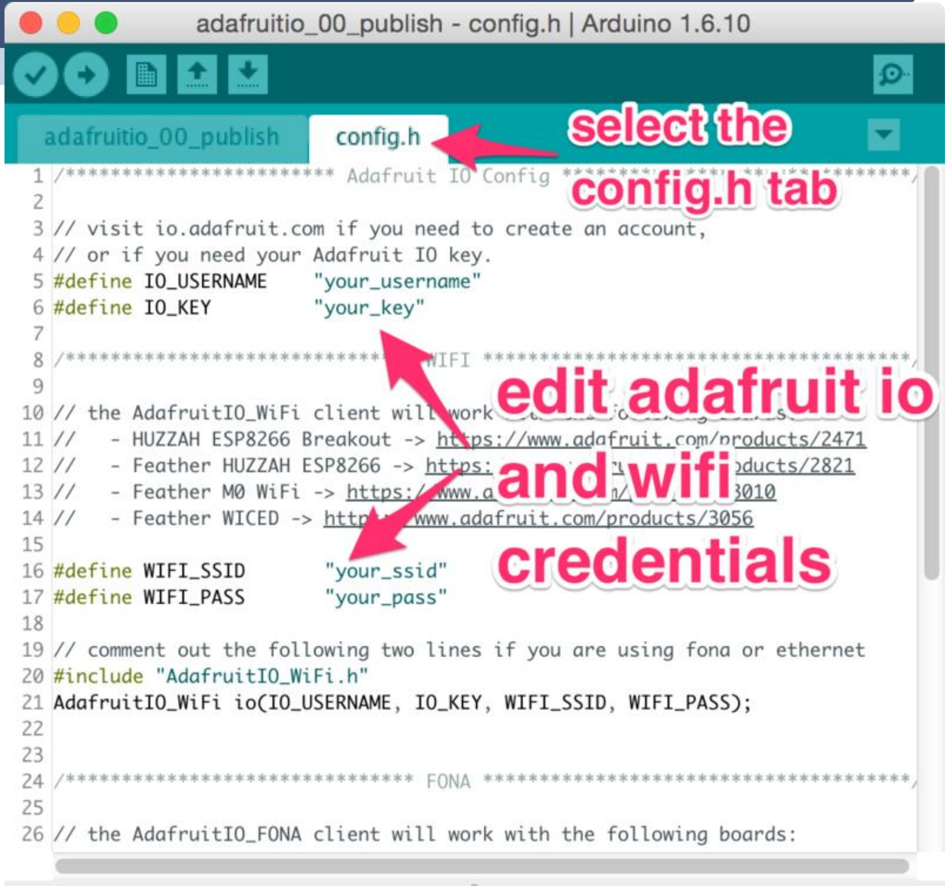
Button Off Text: 0

# Adafruit IO



adafruit
SHOP
BLOG
LEARN
FORUMS
VIDEOS
uniontownlabs / Dashboards / Example Dashboard
LED
0

# Adafruit IO

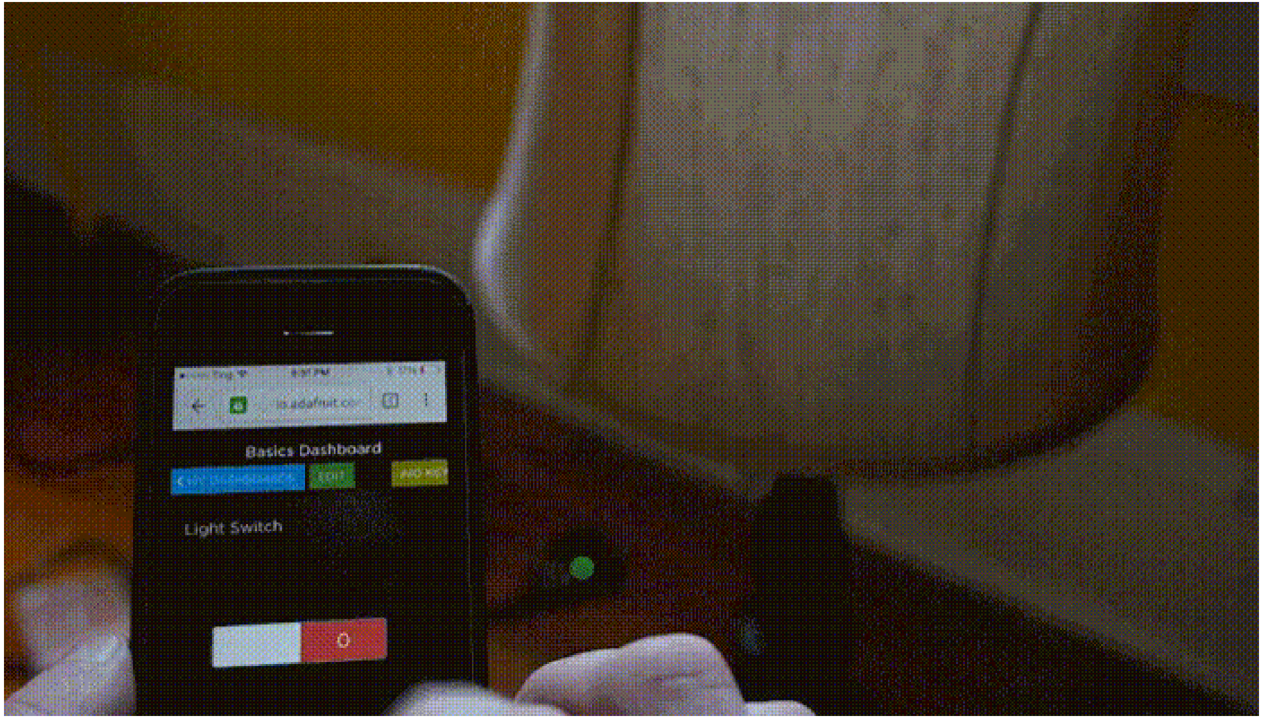


```
/************************ Adafruit IO Config *******************************/

// visit io.adafruit.com if you need to create an account,
// or if you need your Adafruit IO key.
#define IO_USERNAME    "your_username"
#define IO_KEY         "your_key"

/******************************* WIFI **************************************/

// the AdafruitIO_WiFi client will work with the following boards:
//   - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471
//   - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
//   - Feather M0 WiFi -> https://www.adafruit.com/products/3010
//   - Feather WICED -> https://www.adafruit.com/products/3056

#define WIFI_SSID       "your_ssid"
#define WIFI_PASS       "your_pass"

// comment out the following two lines if you are using fona or ethernet
#include "AdafruitIO_WiFi.h"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);


/******************************* FONA **************************************/

// the AdafruitIO_FONA client will work with the following boards:
```

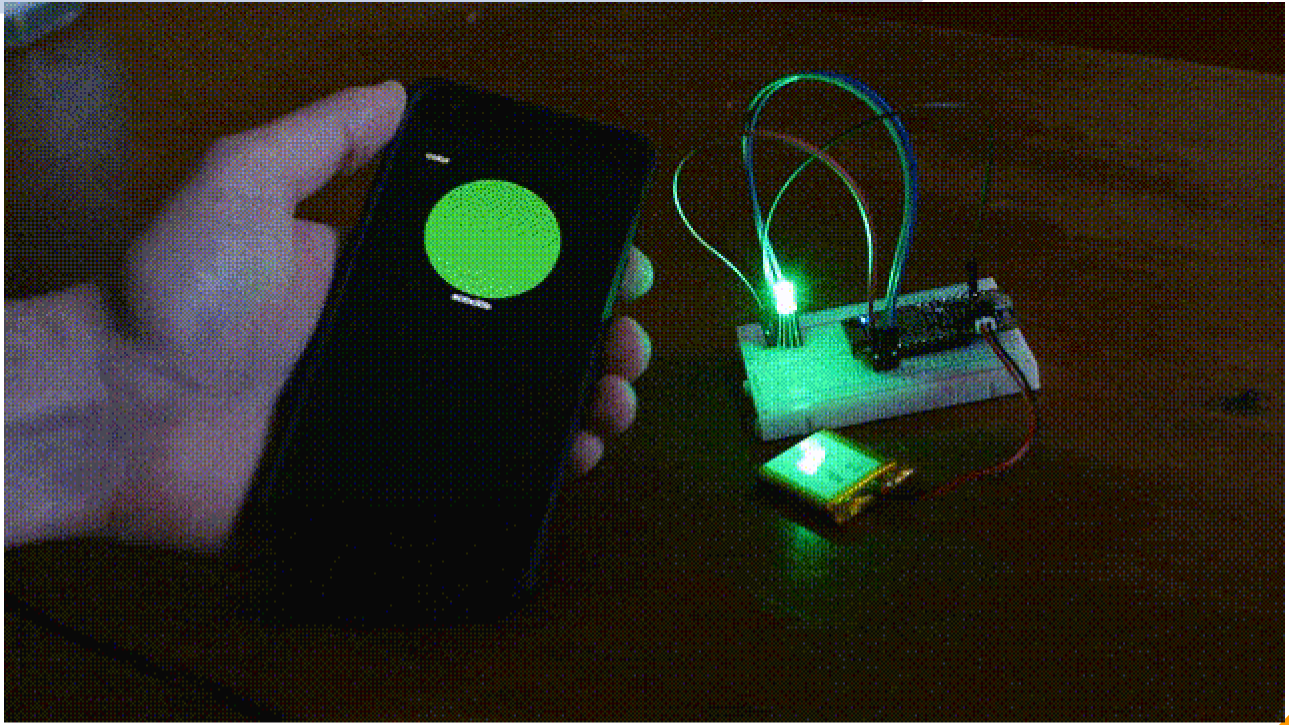
select the config.h tab

edit adafruit io and wifi credentials

# Adafruit IO

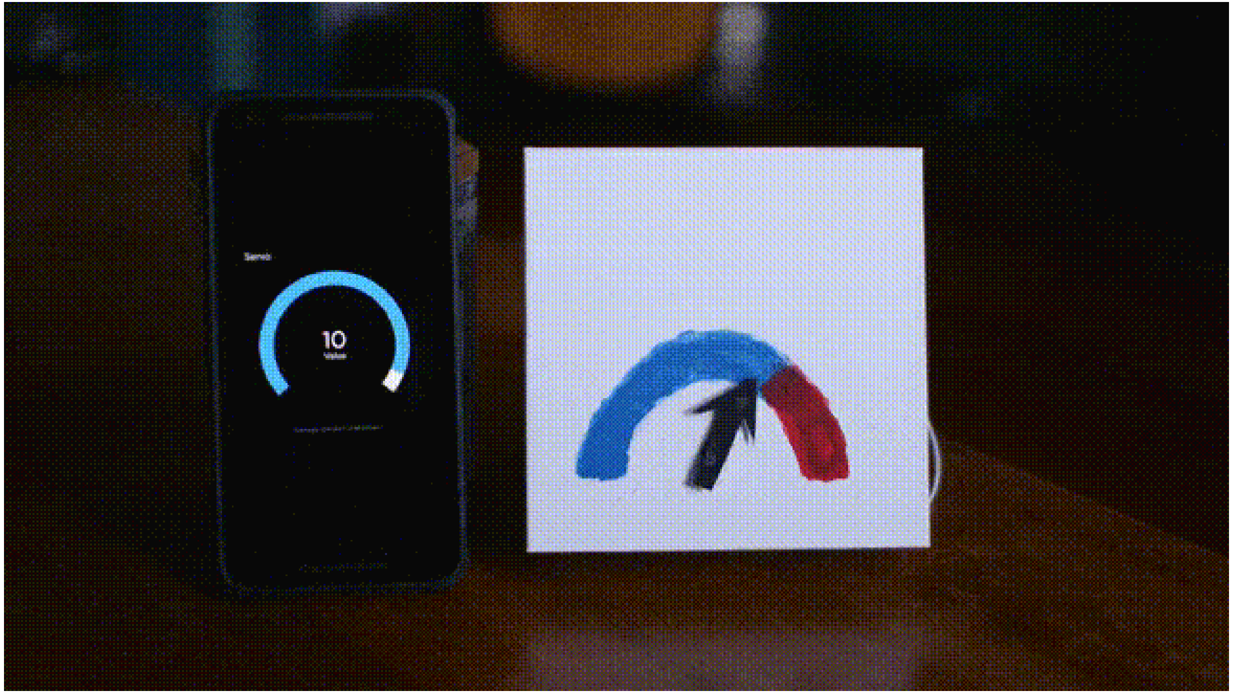


# Adafruit IO



# Adafruit IO

# Adafruit IO



# THANKS!