Course: EGDF20 Diploma in Electronic and Computer Engineering

Module: EGE356 IoT System Architecture & Technology

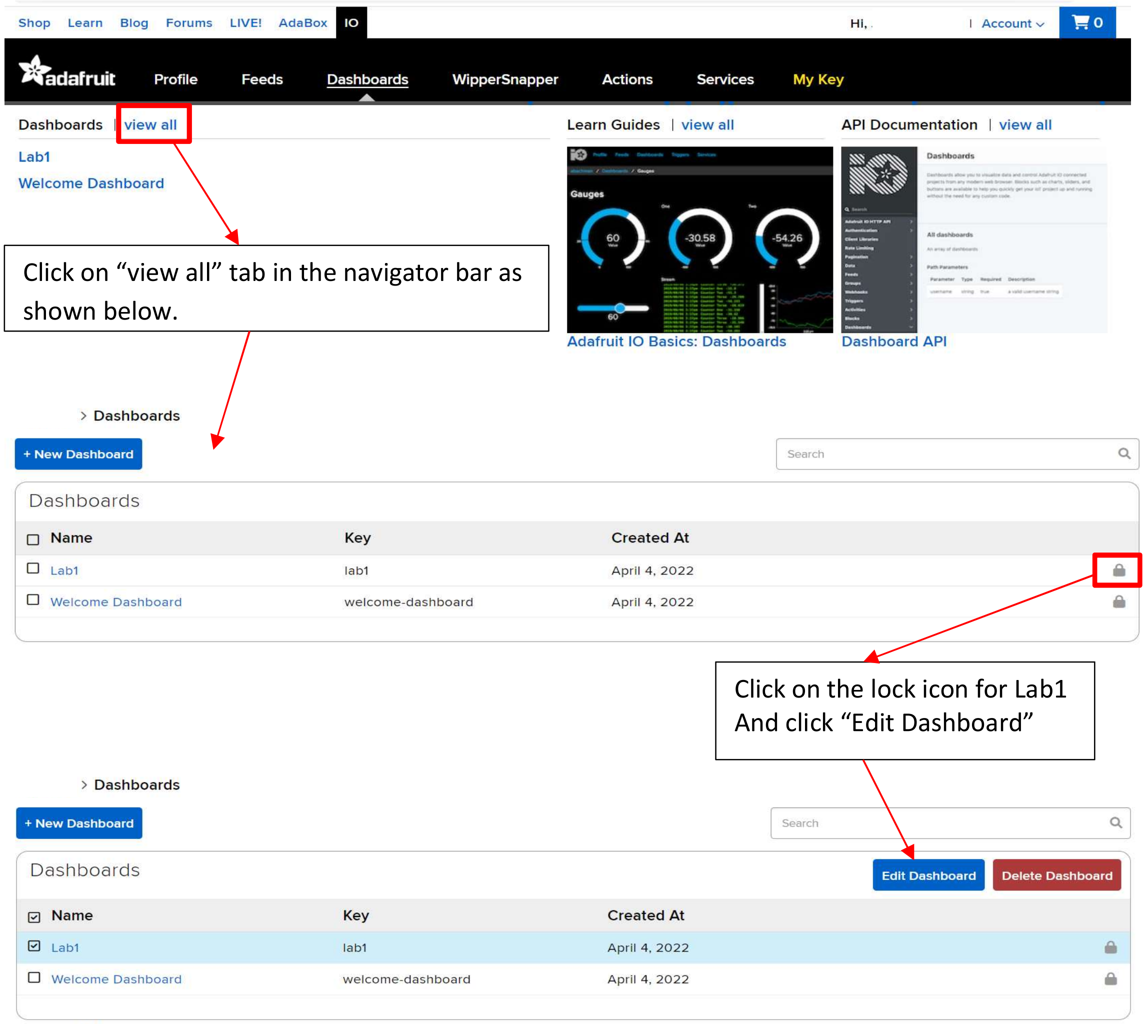
Lab 2: Endpoint Data to Cloud Platform Dashboard

Objectives:

1. Endpoint Data to Dashboard Gauge
2. Cloud and Endpoint Data Exchange

Part 1: Create Dashboard Gauge

1. Upon login to Adafruit IO, go to the Dashboard page.



Click on “view all” tab in the navigator bar as shown below.

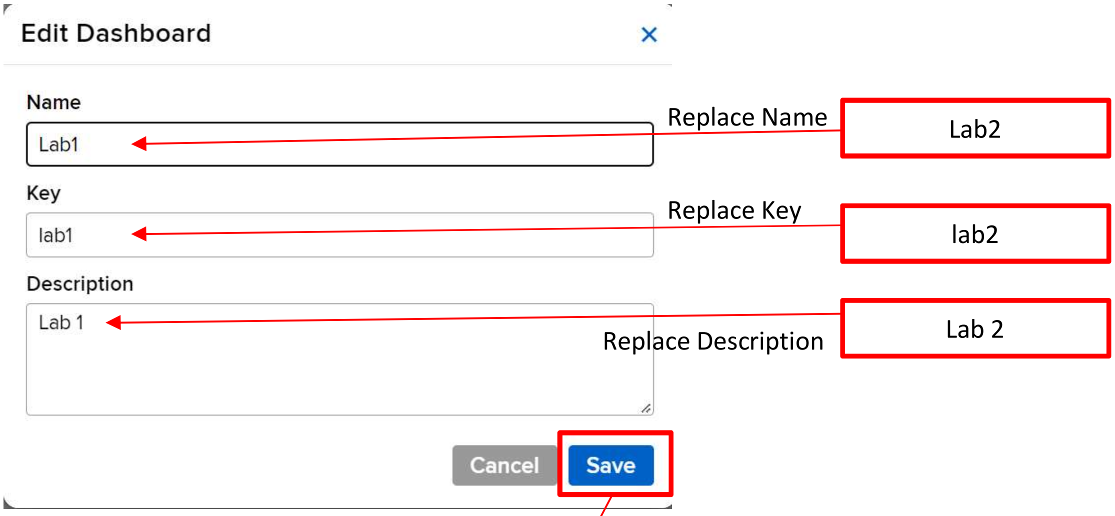
Dashboards

□ Name	Key	Created At
□ Lab1	lab1	April 4, 2022
□ Welcome Dashboard	welcome-dashboard	April 4, 2022

Click on the lock icon for Lab1 And click “Edit Dashboard”

Dashboards

☑ Name	Key	Created At
☑ Lab1	lab1	April 4, 2022
□ Welcome Dashboard	welcome-dashboard	April 4, 2022



Edit Dashboard

Name: Lab1

Key: lab1

Description: Lab 1

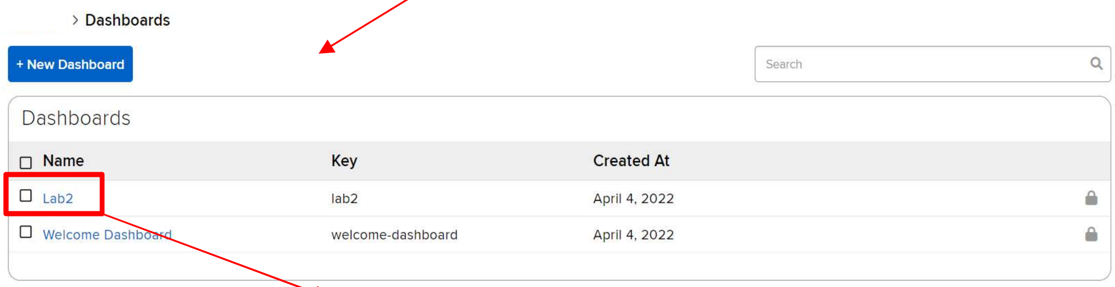
Cancel | Save

Replace Name → Lab2

Replace Key → lab2

Replace Description → Lab 2

Click “Save” and refresh the Dashboard to ensure changes updated.



› Dashboards

+ New Dashboard

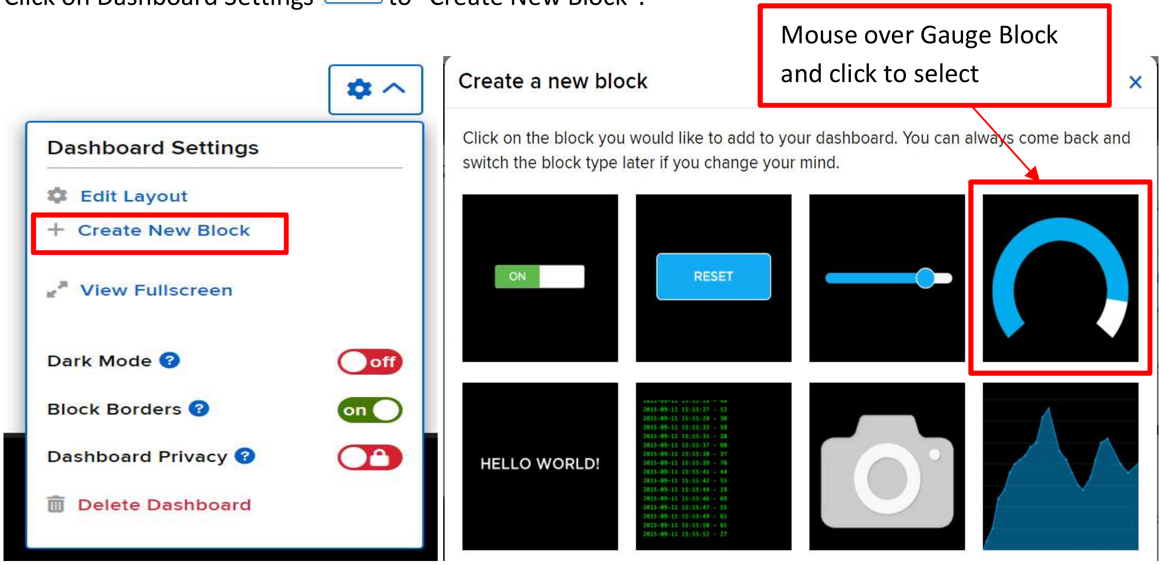
Dashboards

□ Name	Key	Created At
□ Lab2	lab2	April 4, 2022
□ Welcome Dashboard	welcome-dashboard	April 4, 2022

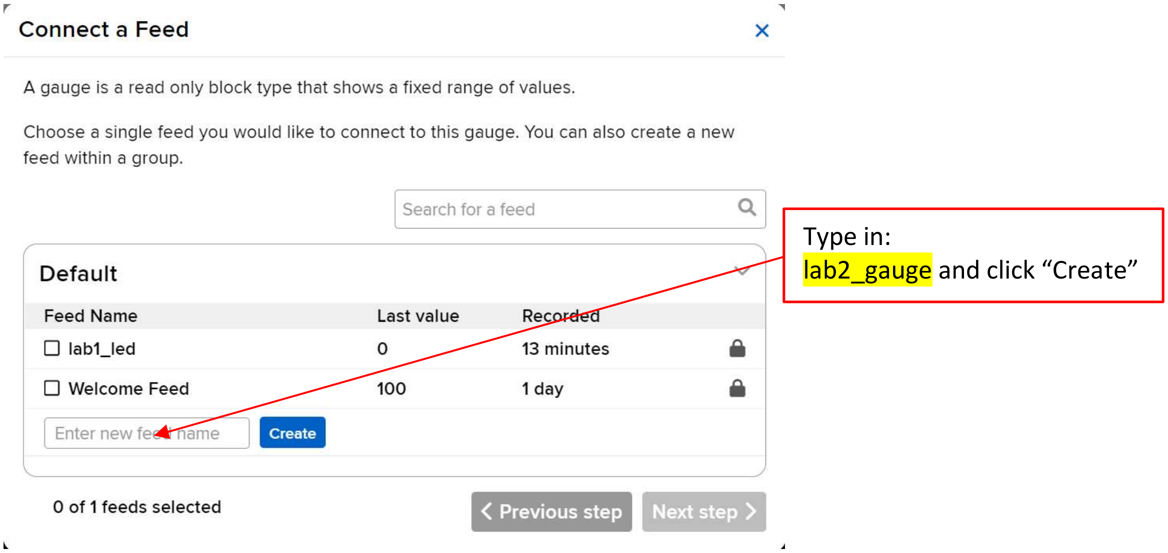
Click on Lab2 Dashboard to create the Gauge Block

2. Click on Dashboard Settings ⚙ ^ to “Create New Block”.

Mouse over Gauge Block and click to select



Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



Connect a Feed

A gauge is a read only block type that shows a fixed range of values.

Choose a single feed you would like to connect to this gauge. You can also create a new feed within a group.

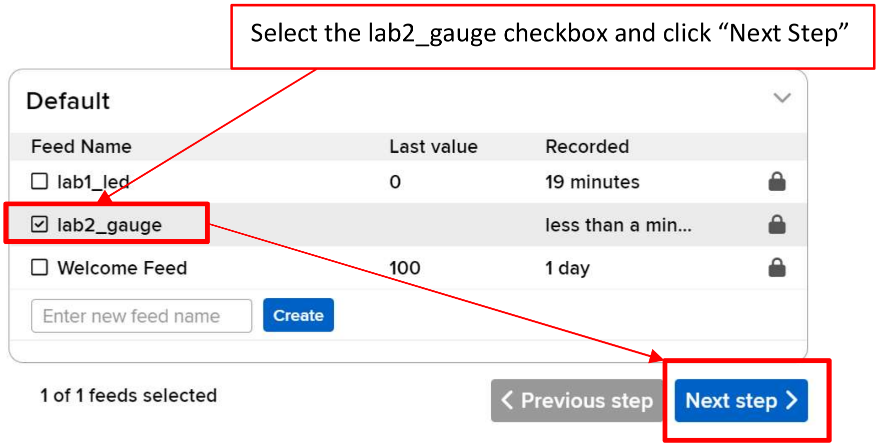
Default

Feed Name	Last value	Recorded
□ lab1_led	0	13 minutes
□ Welcome Feed	100	1 day

Enter new feed name — Create

0 of 1 feeds selected

Type in:
lab2_gauge and click "Create"



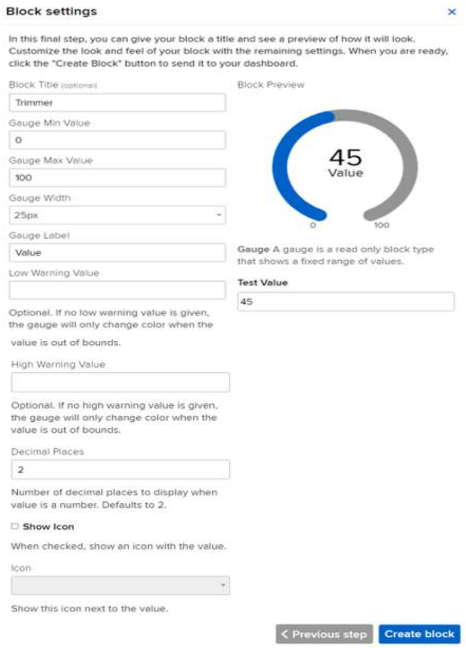
Select the lab2_gauge checkbox and click "Next Step"

Default

Feed Name	Last value	Recorded
□ lab1_led	0	19 minutes
☑ lab2_gauge		less than a min...
□ Welcome Feed	100	1 day

Enter new feed name — Create

1 of 1 feeds selected



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 (optional): Trimmer

Gauge Min Value: 0

Gauge Max Value: 100

Gauge Width: 25px

Gauge Label: Value

Low Warning Value:

Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds.

High Warning Value:

Optional. If no high warning value is given, the gauge will only change color when the value is out of bounds.

Decimal Places: 2

Number of decimal places to display when value is a number. Defaults to 2.

□ Show Icon

When checked, show an icon with the value.

Icon:

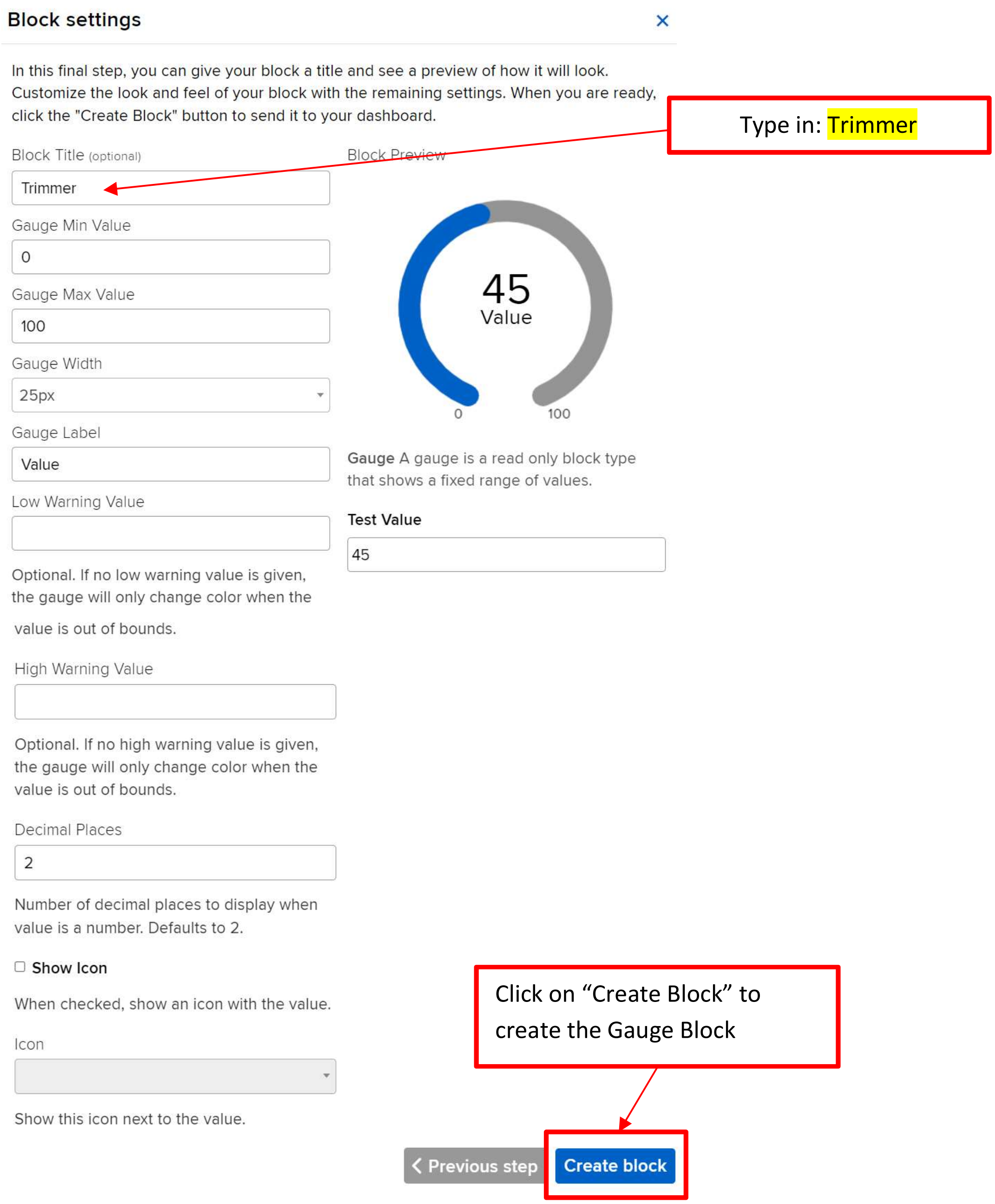
Show this icon next to the value.

Block Preview

Gauge A gauge is a read only block type that shows a fixed range of values.

Test Value: 45

Click Next Step, leave the default settings in tack. Note that the full details of settings for the Gauge is provided on the next page. You may use it for subsequent labs to configure gauge display where necessary. For now just make the following minor changes as shown on the next page.



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 (optional): Trimmer

Gauge Min Value: 0

Gauge Max Value: 100

Gauge Width: 25px

Gauge Label: Value

Low Warning Value:

Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds.

High Warning Value:

Optional. If no high warning value is given, the gauge will only change color when the value is out of bounds.

Decimal Places: 2

Number of decimal places to display when value is a number. Defaults to 2.

□ Show Icon

When checked, show an icon with the value.

Icon:

Show this icon next to the value.

Block Preview

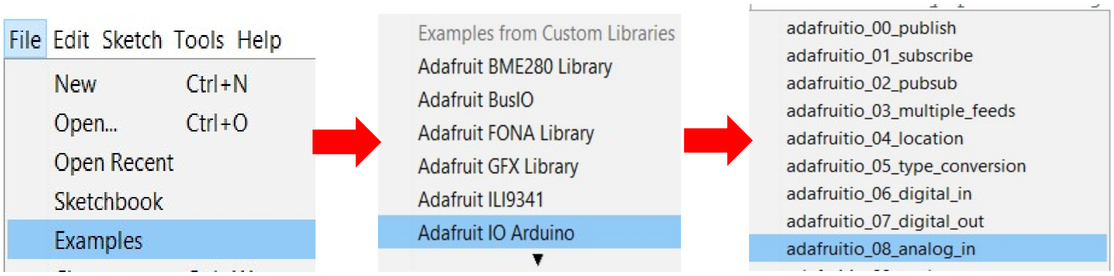
Gauge A gauge is a read only block type that shows a fixed range of values.

Test Value: 45

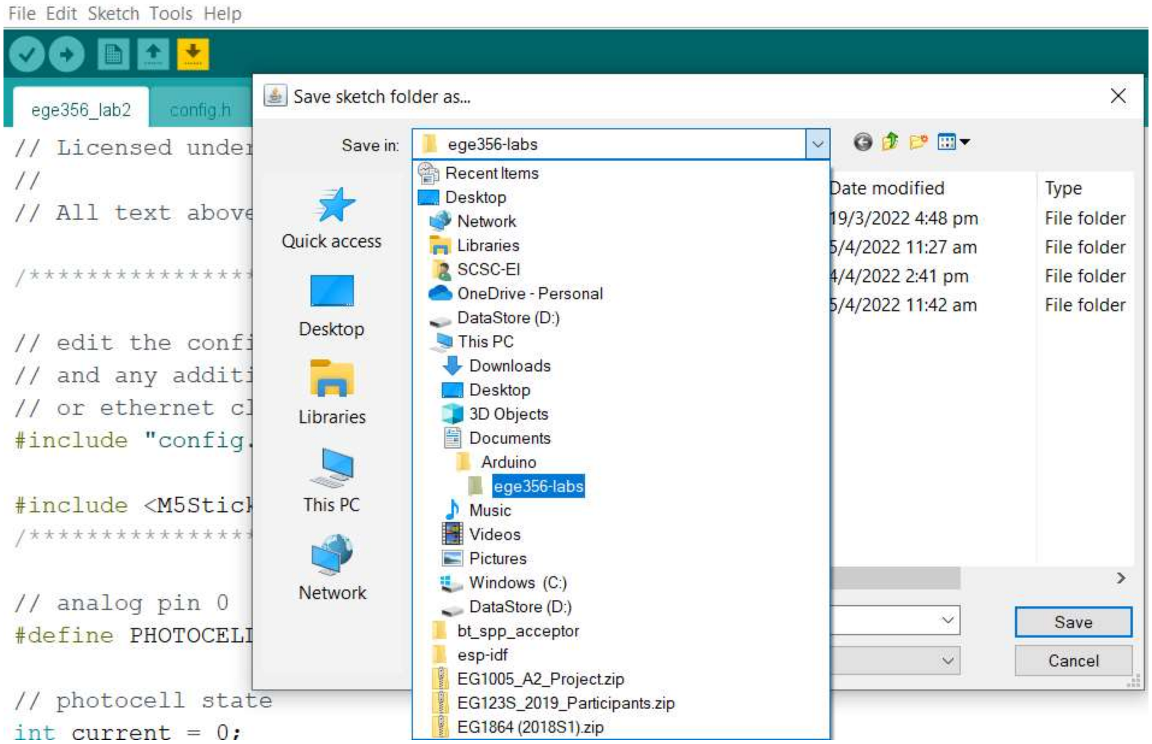
Type in: Trimmer

Click on “Create Block” to create the Gauge Block

3. Launch Arduino.exe, go to File → Examples → Adafruit IO Arduino → adafruitio_08_analog_in and select the example code as shown.



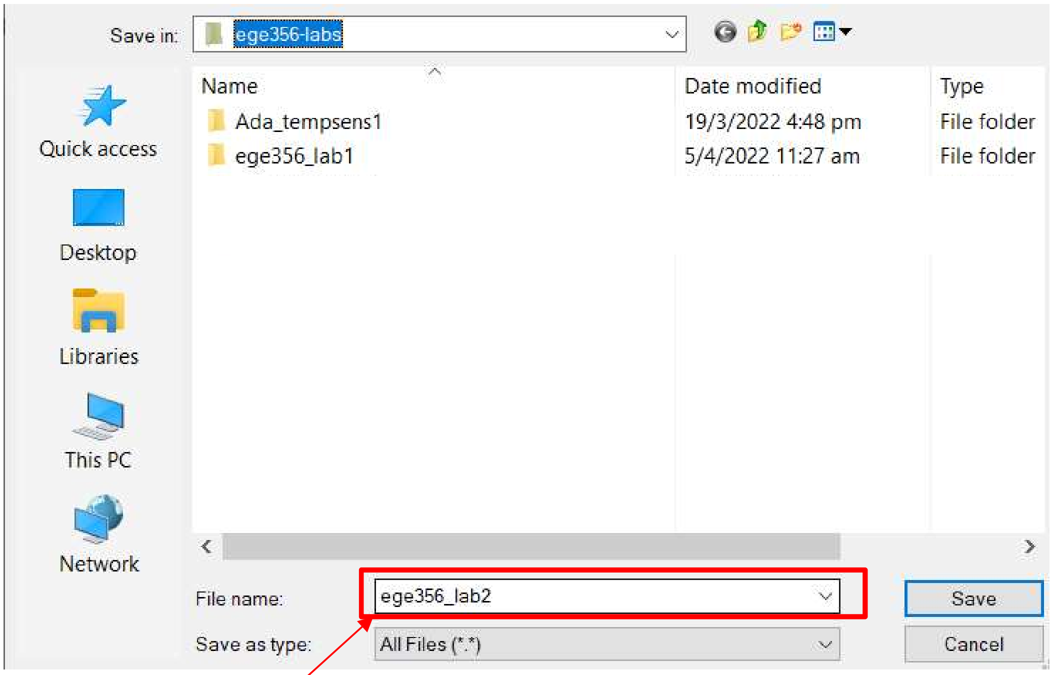
4. Click on File → Save As → ege356-labs →



Save in: ege356-labs

Date modified	Type
19/3/2022 4:48 pm	File folder
5/4/2022 11:27 am	File folder
4/4/2022 2:41 pm	File folder
5/4/2022 11:42 am	File folder

```
// Licensed unde
//
// All text above

/***************

// edit the confi
// and any additi
// or ethernet cl
#include "config.

#include <M5Stic
/***************

// analog pin 0
#define PHOTOCELI

// photocell state
int current = 0;
```



Save in: ege356-labs

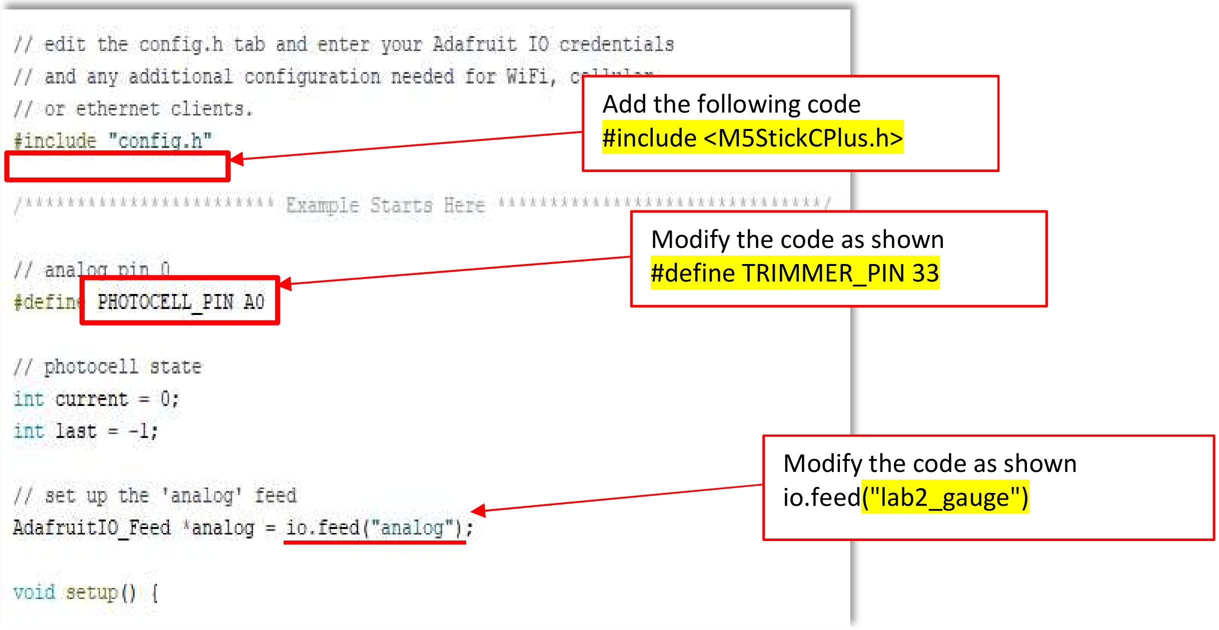
Name	Date modified	Type
Ada_tempsens1	19/3/2022 4:48 pm	File folder
ege356_lab1	5/4/2022 11:27 am	File folder

File name: ege356_lab2

Save as type: All Files (*.*)

Key in ege356_lab2 and click save.

5. Modify the codes as shown below.



```
// edit the config.h tab and enter your Adafruit IO credentials
// and any additional configuration needed for WiFi, cellular,
// or ethernet clients.
#include "config.h"

/************************ Example Starts Here *******************************/

// analog pin 0
#define PHOTOCELL_PIN A0

// photocell state
int current = 0;
int last = -1;

// set up the 'analog' feed
AdafruitIO_Feed *analog = io.feed("analog");

void setup() {
```

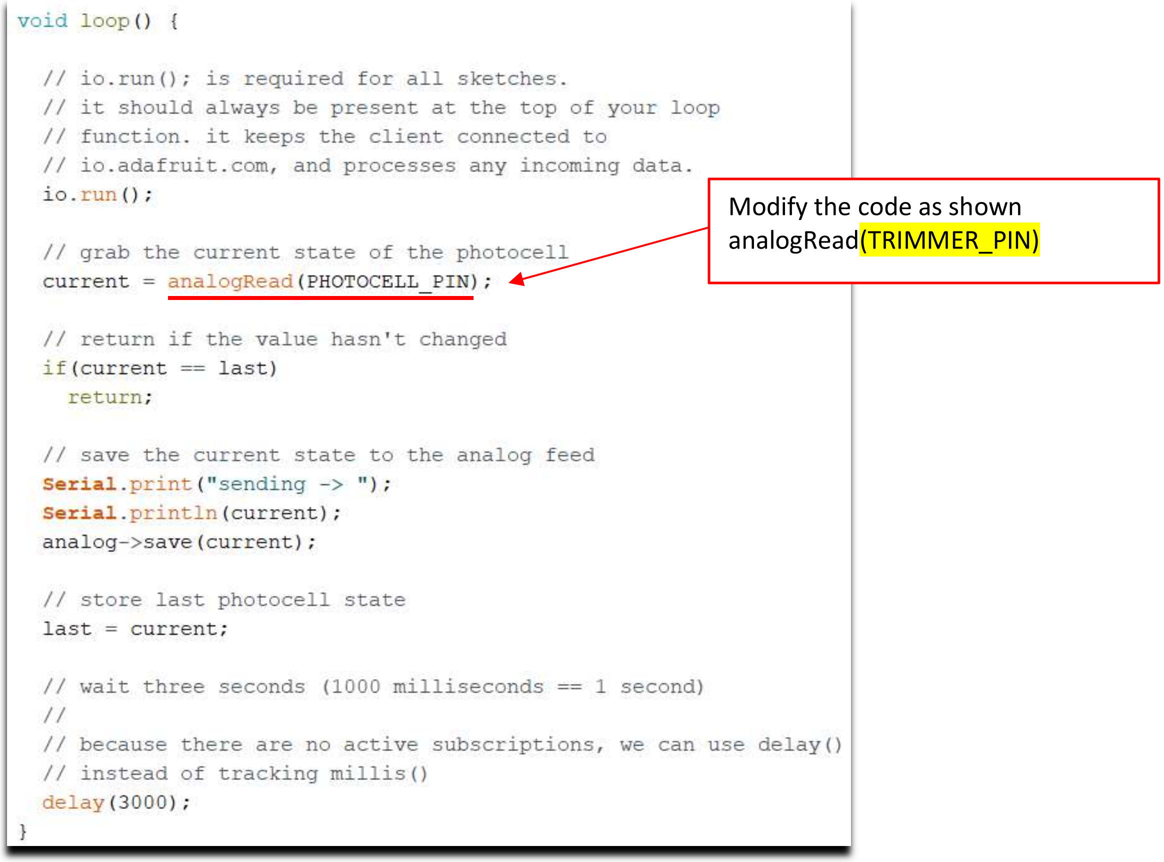
Add the following code
#include <M5StickCPlus.h>

Modify the code as shown
#define TRIMMER_PIN 33

Modify the code as shown
io.feed("lab2_gauge")

. . . .

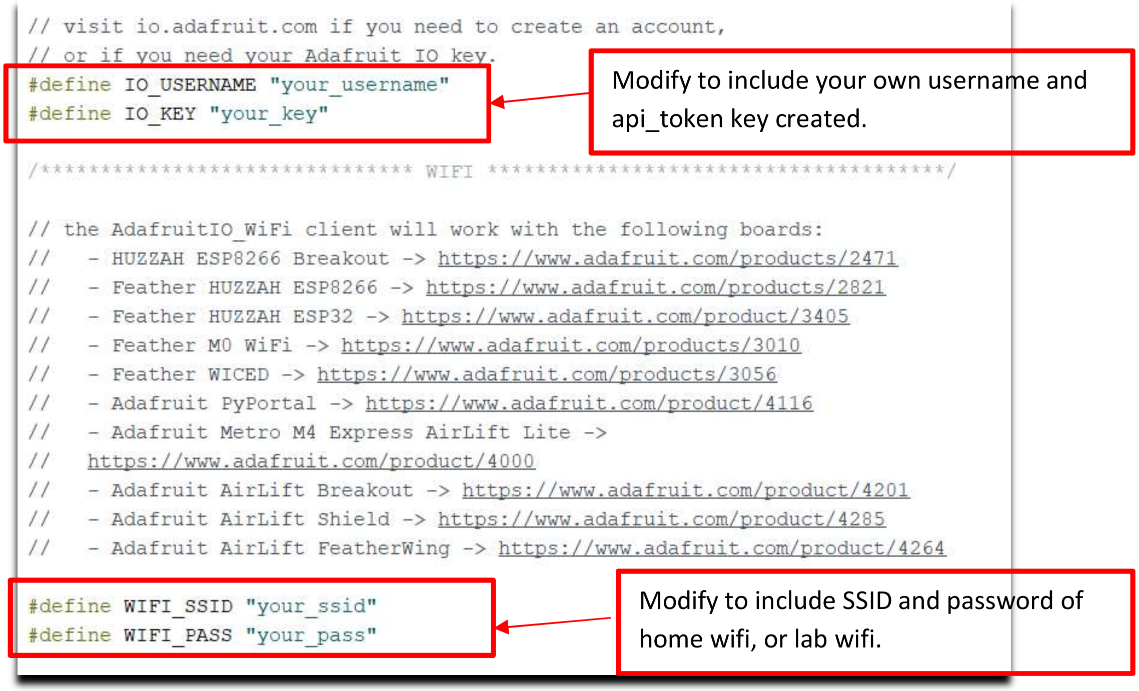
. . . .



```
void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

  // grab the current state of the photocell
  current = analogRead(PHOTOCELL_PIN);

  // return if the value hasn't changed
  if(current == last)
    return;

  // save the current state to the analog feed
  Serial.print("sending -> ");
  Serial.println(current);
  analog->save(current);

  // store last photocell state
  last = current;

  // wait three seconds (1000 milliseconds == 1 second)
  //
  // because there are no active subscriptions, we can use delay()
  // instead of tracking millis()
  delay(3000);
}
```

Modify the code as shown
analogRead(TRIMMER_PIN)

6. Make the required changes to config.h to include IO_USERNAME,IO_KEY, WIFI_SSID, AND WIFI_PASS with the correct username, api_token, ssid and password.

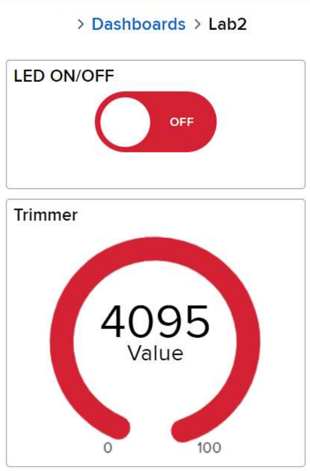


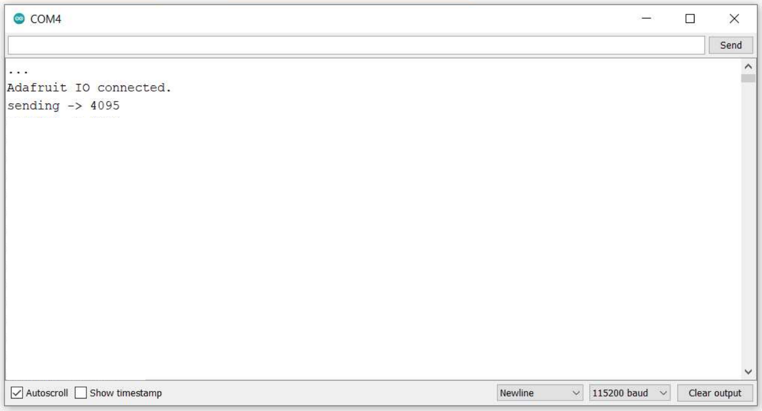
```
// visit io.adafruit.com if you need to create an account,
// or if you need your Adafruit IO key.
#define IO_USERNAME "your_username"
#define IO_KEY "your_key"

/******************************* 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 HUZZAH ESP32 -> https://www.adafruit.com/product/3405
//   - Feather M0 WiFi -> https://www.adafruit.com/products/3010
//   - Feather WICED -> https://www.adafruit.com/products/3056
//   - Adafruit PyPortal -> https://www.adafruit.com/product/4116
//   - Adafruit Metro M4 Express AirLift Lite ->
//   https://www.adafruit.com/product/4000
//   - Adafruit AirLift Breakout -> https://www.adafruit.com/product/4201
//   - Adafruit AirLift Shield -> https://www.adafruit.com/product/4285
//   - Adafruit AirLift FeatherWing -> https://www.adafruit.com/product/4264

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

Modify to include your own username and api_token key created.

Modify to include SSID and password of home wifi, or lab wifi.

7. Compile then load the code to M5StickC Plus. Connect the M5StickC Plus device with the ANGLE Unit (potentiometer) as shown in step 11.

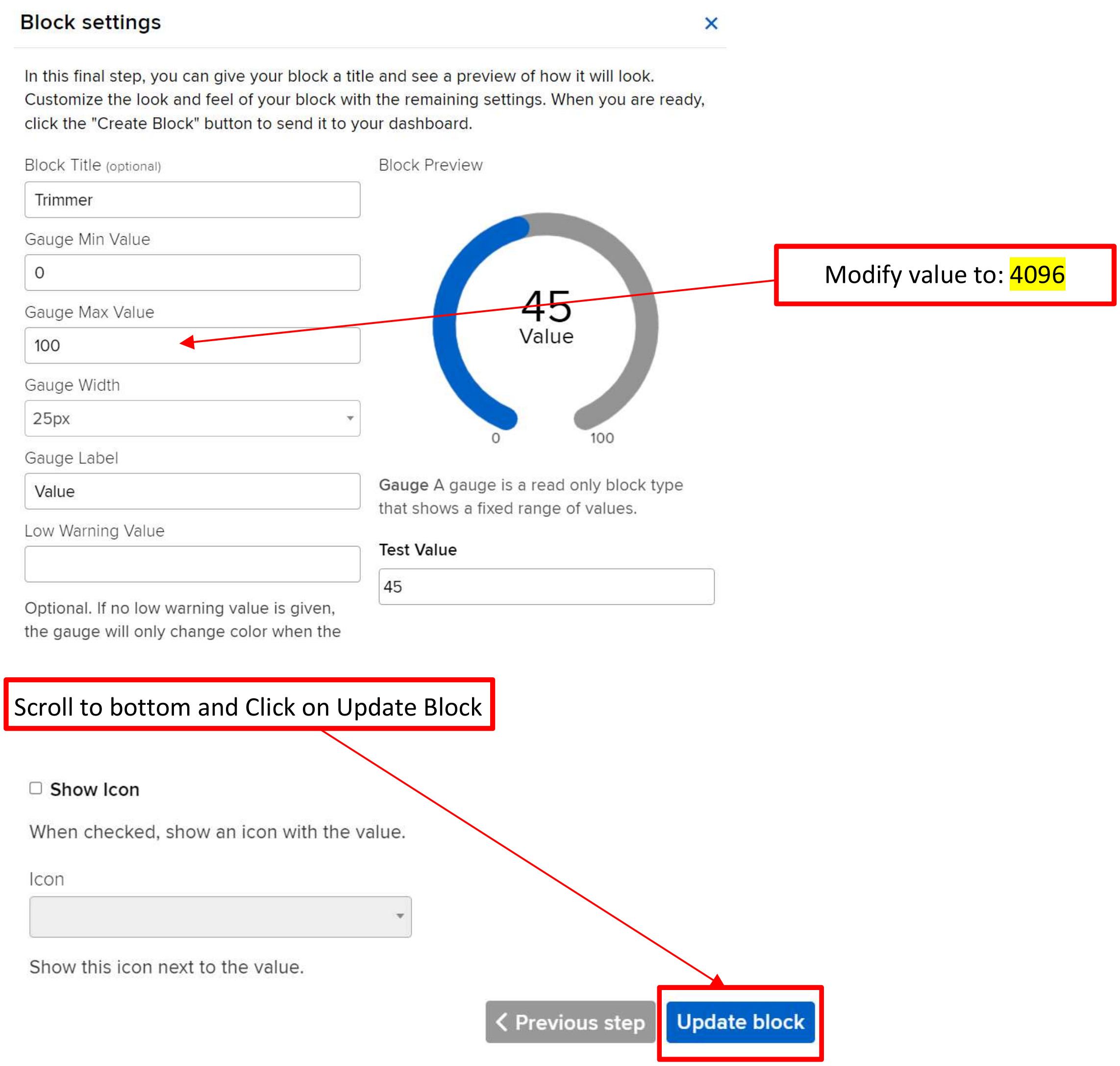




```
...
Adafruit IO connected.
sending -> 4095
```

8. Note that value has exceed the max value of 100, as such the color of the gauge is red.

9. Go to Dashboard edit control → edit the layout → edit Trimmer Block → click on “Next Step” in “Connect a Feed” → Modify Gauge Max Value in Block Settings as shown below.



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 (optional): Trimmer

Gauge Min Value: 0

Gauge Max Value: 100

Gauge Width: 25px

Gauge Label: Value

Low Warning Value:

Optional. If no low warning value is given, the gauge will only change color when the

Block Preview

Gauge A gauge is a read only block type that shows a fixed range of values.

Test Value: 45

☐ Show Icon

When checked, show an icon with the value.

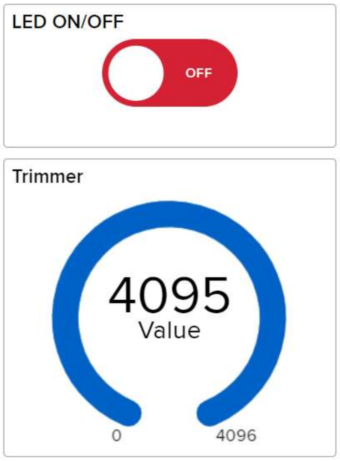
Icon

Show this icon next to the value.

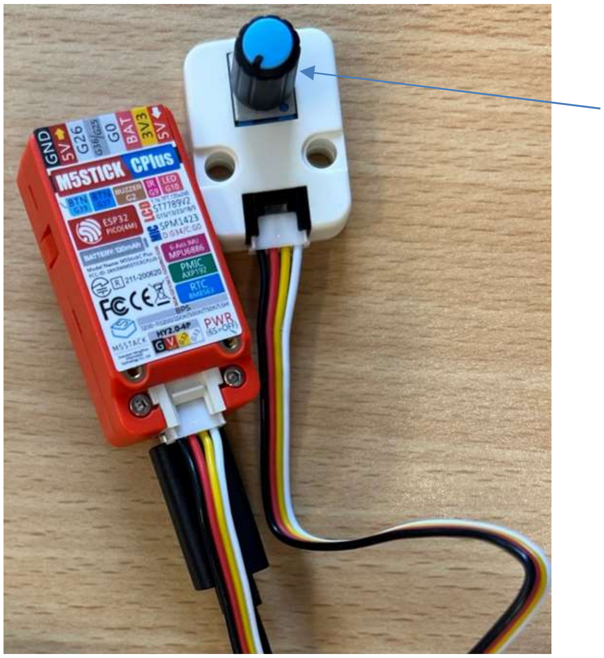
Modify value to: 4096

Scroll to bottom and Click on Update Block

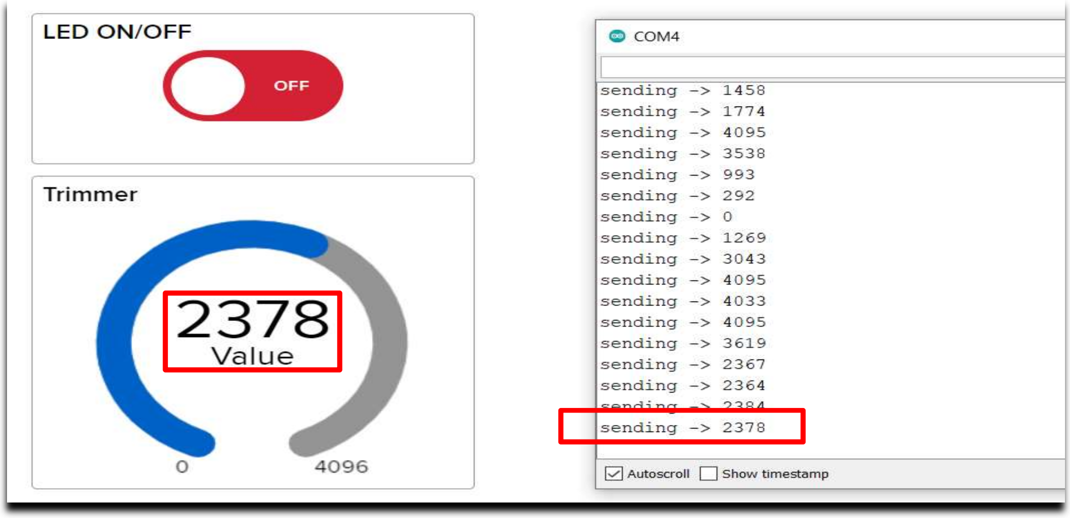
10. Click on “Save Layout” to update the Dashboard.



11. Trimmer and M5StickC device connection.



12. Note that the Trimmer Block value displayed corresponds to the serial output display.



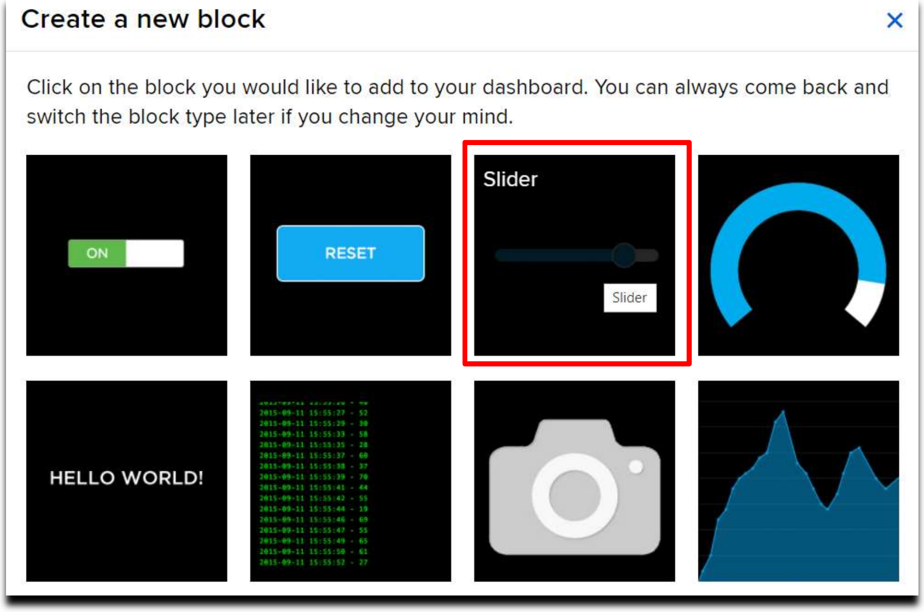
```
sending -> 1458
sending -> 1774
sending -> 4095
sending -> 3538
sending -> 993
sending -> 292
sending -> 0
sending -> 1269
sending -> 3043
sending -> 4095
sending -> 4033
sending -> 4095
sending -> 3619
sending -> 2367
sending -> 2364
sending -> 2384
sending -> 2378
```

☑ Autoscroll □ Show timestamp

13. Summary: Each time there is a change in value, the data will be sent from M5StickC device to Adafruit_IO. Note that the delay used is 3000ms, and that data is sent if there is a change in value.

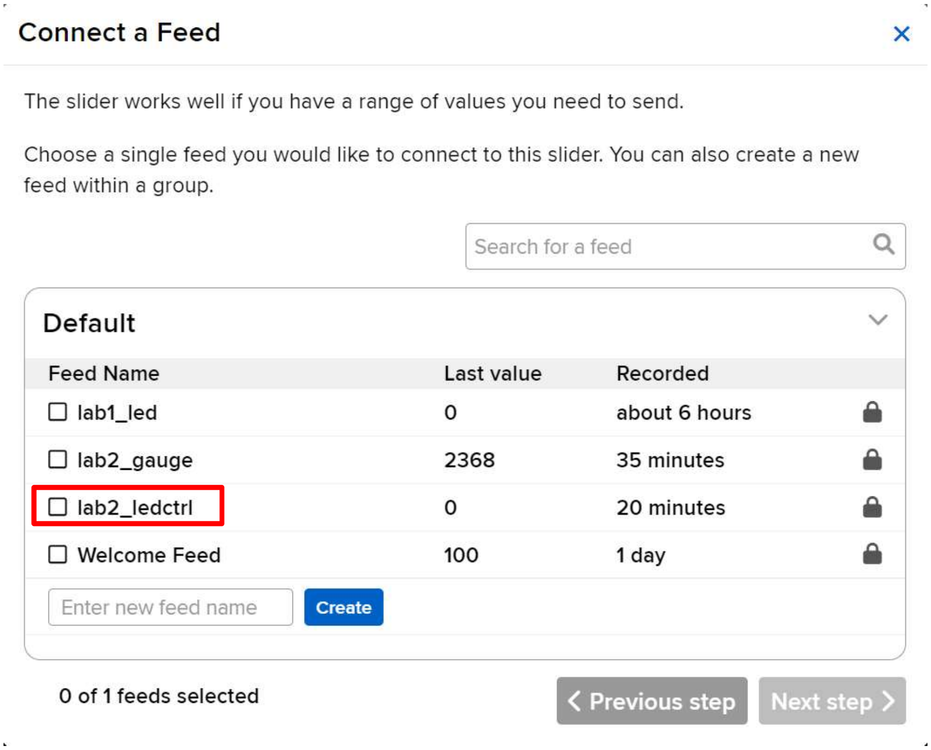
Part 2: Cloud and Endpoint Data Exchange

1. Add a **Slider Block** and the **lab2_ledctrl** feed as shown below.



Create a new block ✕

Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



Connect a Feed ✕

The slider works well if you have a range of values you need to send.

Choose a single feed you would like to connect to this slider. You can also create a new feed within a group.

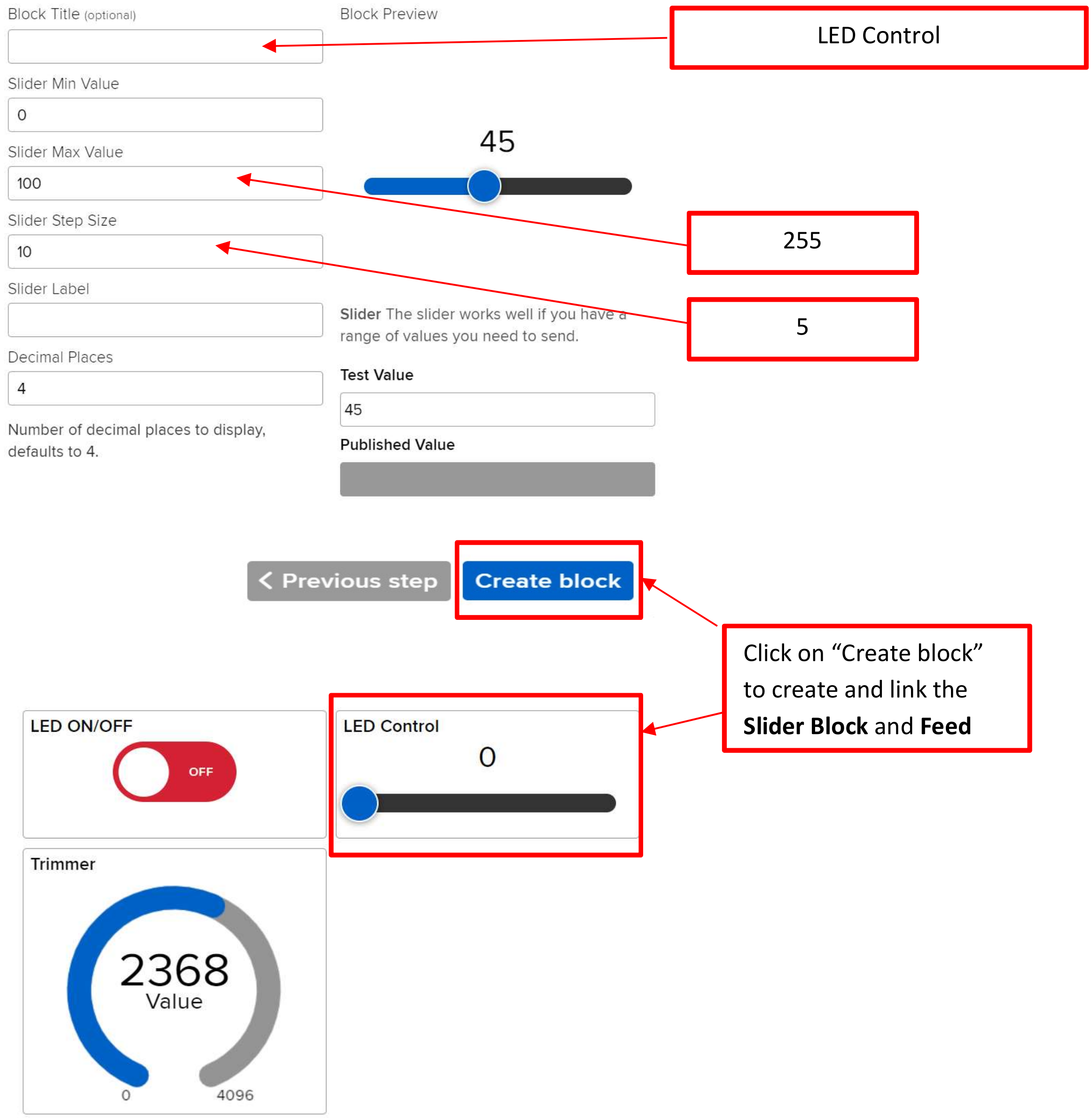
Search for a feed

Default

Feed Name	Last value	Recorded	
□ lab1_led	0	about 6 hours	🔒
□ lab2_gauge	2368	35 minutes	🔒
□ lab2_ledctrl	0	20 minutes	🔒
□ Welcome Feed	100	1 day	🔒

Enter new feed name | Create

0 of 1 feeds selected



Block Title (optional)

Slider Min Value

0

Slider Max Value

100

Slider Step Size

10

Slider Label

Decimal Places

4

Number of decimal places to display, defaults to 4.

Block Preview

45

Slider The slider works well if you have a range of values you need to send.

Test Value

45

Published Value

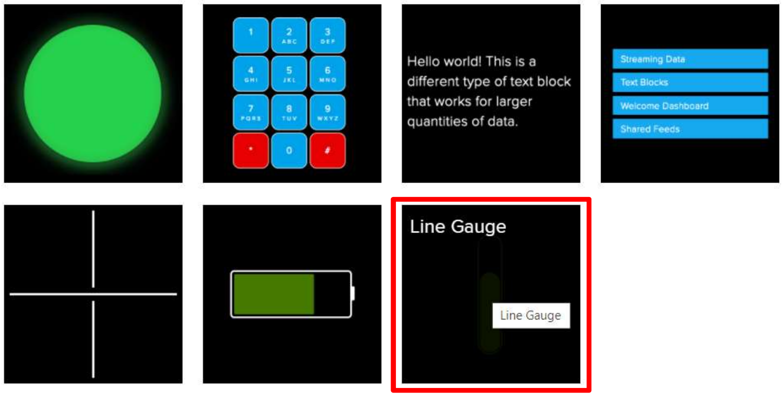
LED Control

255

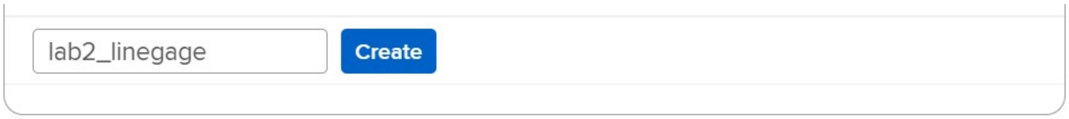
5

Click on "Create block" to create and link the **Slider Block** and **Feed**

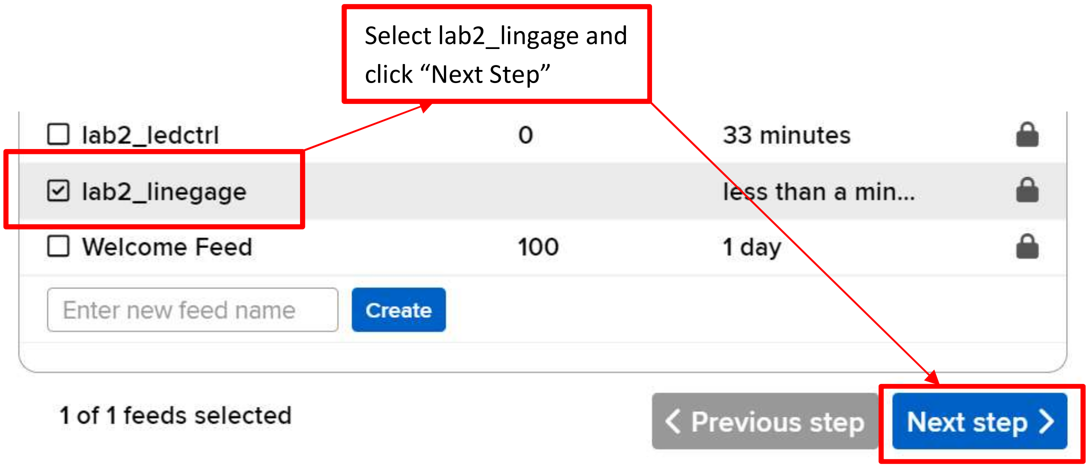
2. Create new Line Gauge Block as shown below with the corresponding Feed.



Hello world! This is a different type of text block that works for larger quantities of data.
Streaming Data
Text Blocks
Welcome Dashboard
Shared Feeds
Line Gauge
Line Gauge



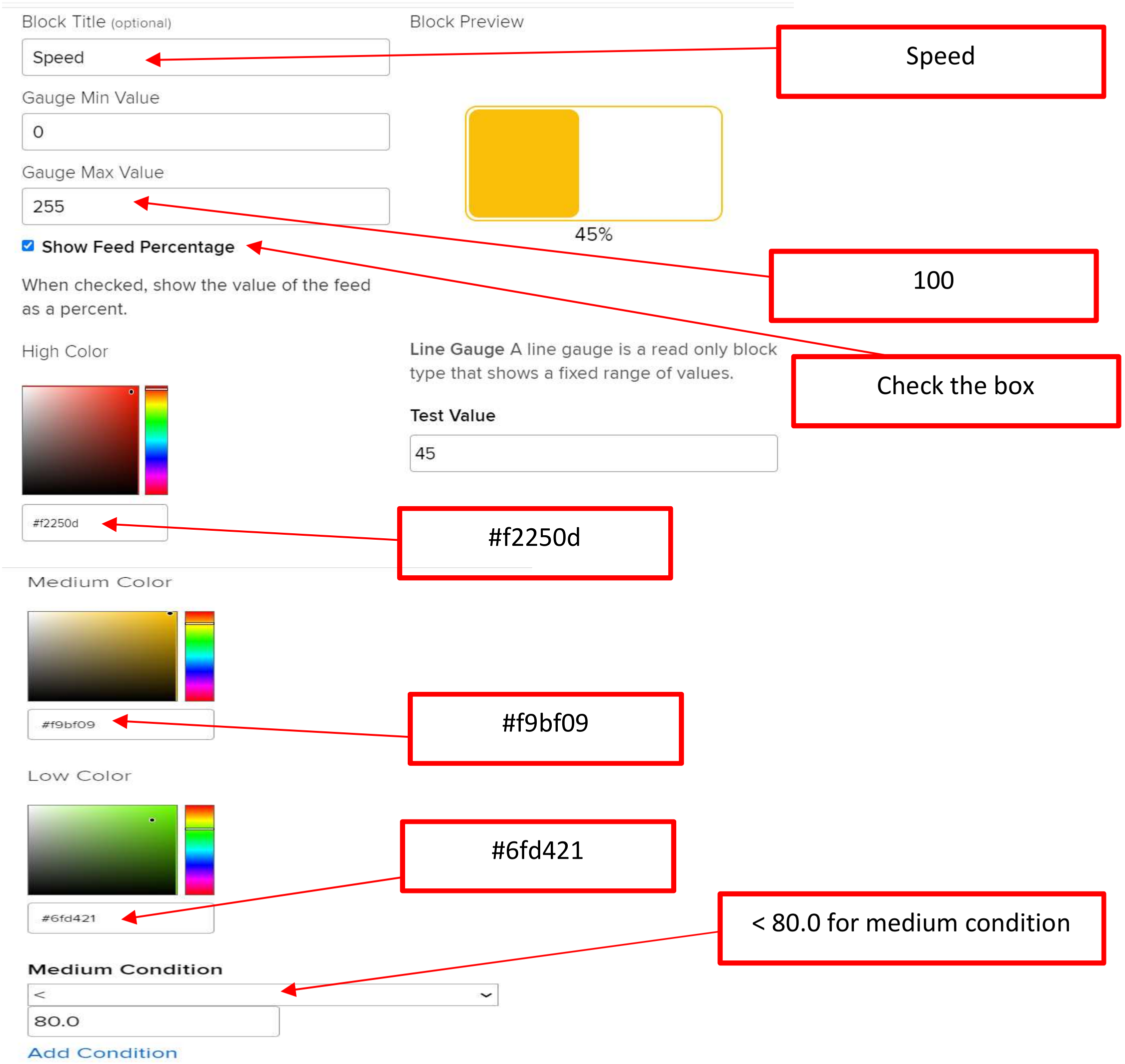
lab2_linegage



Select lab2_lingage and click "Next Step"

☐ lab2_ledctrl	0	33 minutes	🔒
☑ lab2_linegage		less than a min...	🔒
☐ Welcome Feed	100	1 day	🔒

1 of 1 feeds selected



Block Title (optional): Speed

Speed

Gauge Min Value: 0

Gauge Max Value: 255

100

☑ Show Feed Percentage

Check the box

When checked, show the value of the feed as a percent.

Line Gauge A line gauge is a read only block type that shows a fixed range of values.

Test Value: 45

High Color: #f2250d

#f2250d

Medium Color: #f9bf09

#f9bf09

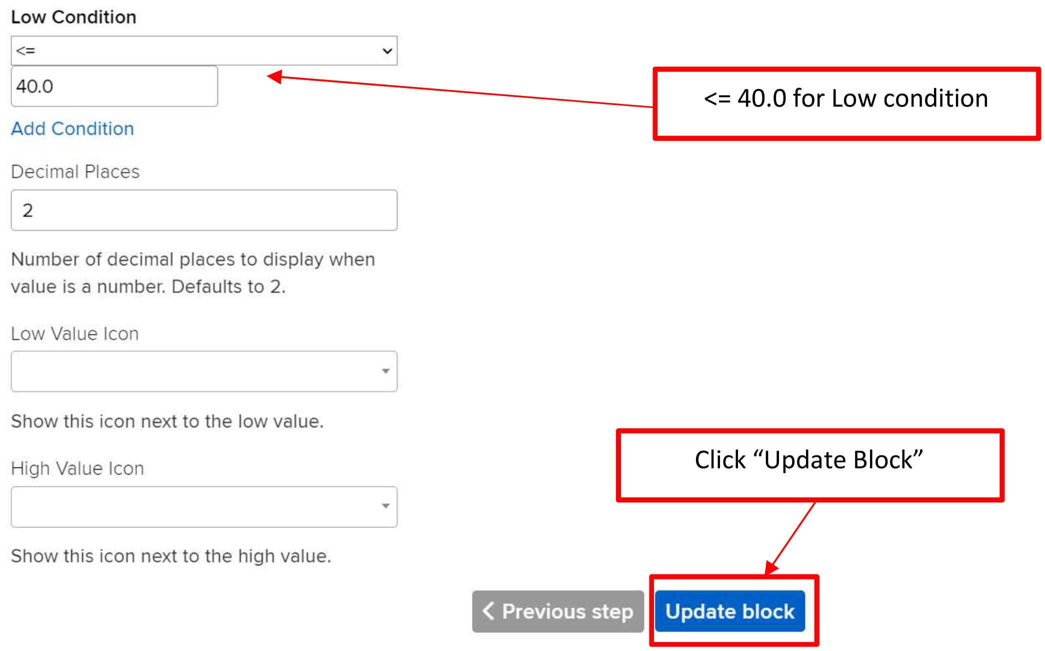
Low Color: #6fd421

#6fd421

Medium Condition: <

80.0

< 80.0 for medium condition



Low Condition

<=

40.0

Add Condition

Decimal Places

2

Number of decimal places to display when value is a number. Defaults to 2.

Low Value Icon

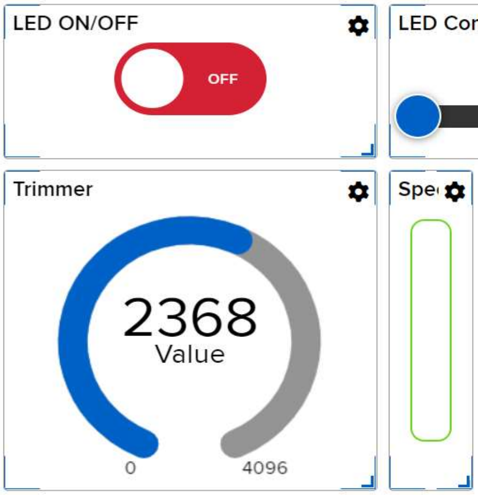
Show this icon next to the low value.

High Value Icon

Show this icon next to the high value.

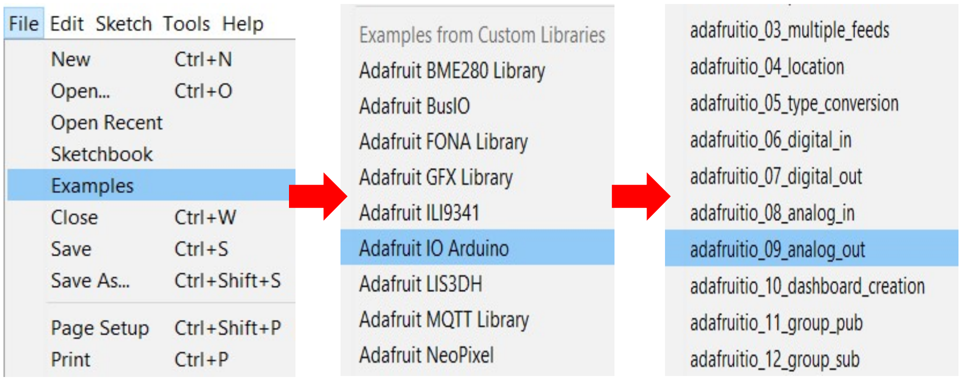
<= 40.0 for Low condition

Click “Update Block”

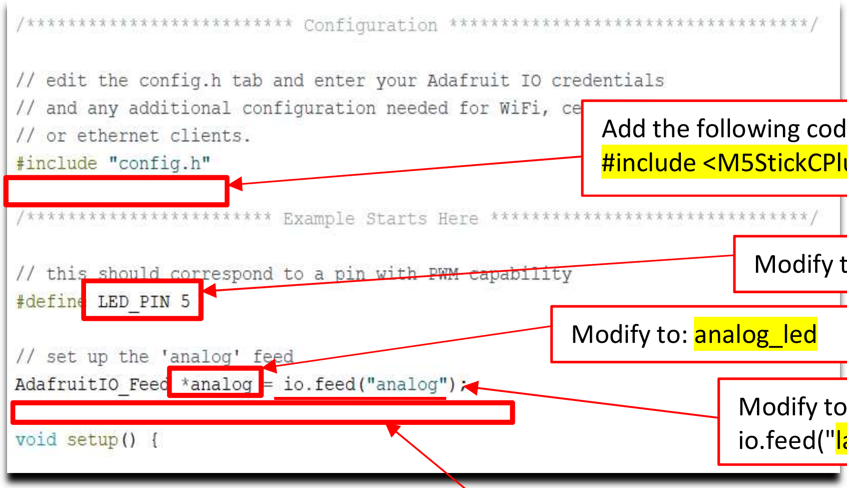


Click on Save Layout to save the Dashboard

3. Select the following example code adafruitio_09_analog_out from Arduino and save it as ege356_lab2_led in ege356_labs folder.



4. Add the following code to ege356_lab2_led.ino



```
/************************** Configuration ***********************************/

// edit the config.h tab and enter your Adafruit IO credentials
// and any additional configuration needed for WiFi, ce
// or ethernet clients.
#include "config.h"

/************************ Example Starts Here *******************************/

// this should correspond to a pin with PWM capability
#define LED_PIN 5

// set up the 'analog' feed
AdafruitIO_Feed *analog = io.feed("analog");

void setup() {
```

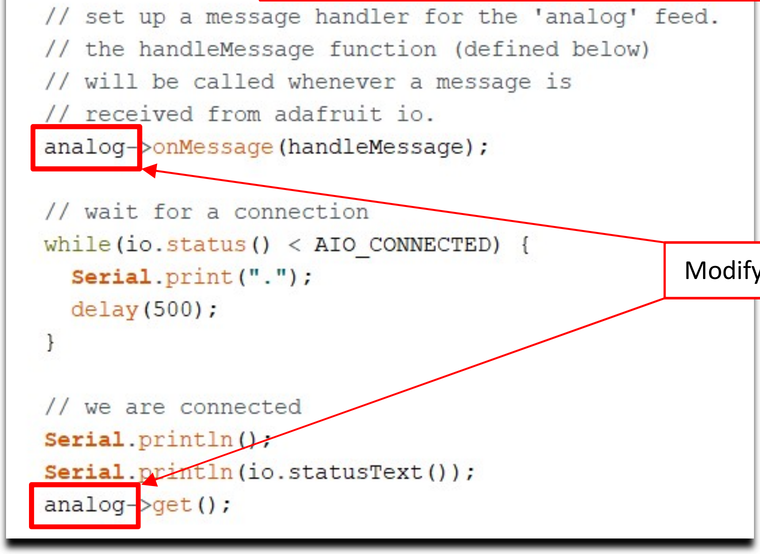
Add the following code
#include <M5StickCPlus.h>

Modify to: LED_PIN 26

Modify to: analog_led

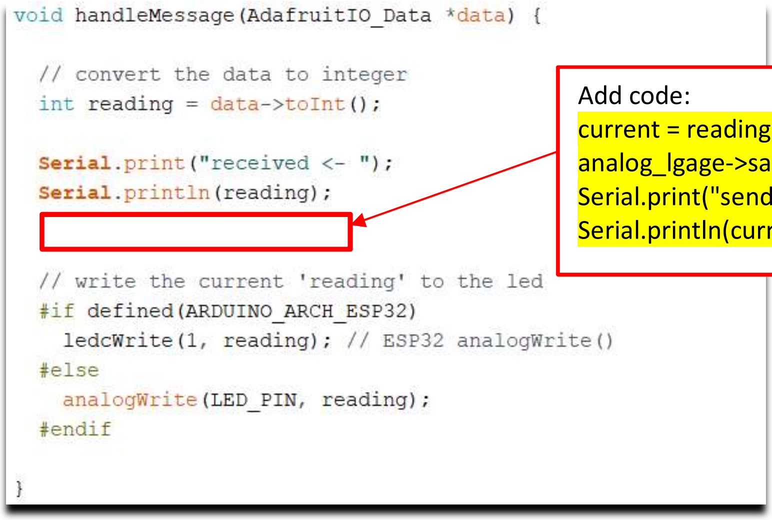
Modify to:
io.feed("lab2_ledctrl")

Add code: AdafruitIO_Feed *analog_lgage = io.feed("lab2_linegage");



```
  // set up a message handler for the 'analog' feed.
  // the handleMessage function (defined below)
  // will be called whenever a message is
  // received from adafruit io.
  analog->onMessage(handleMessage);

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  // we are connected
  Serial.println();
  Serial.println(io.statusText());
  analog->get();
```

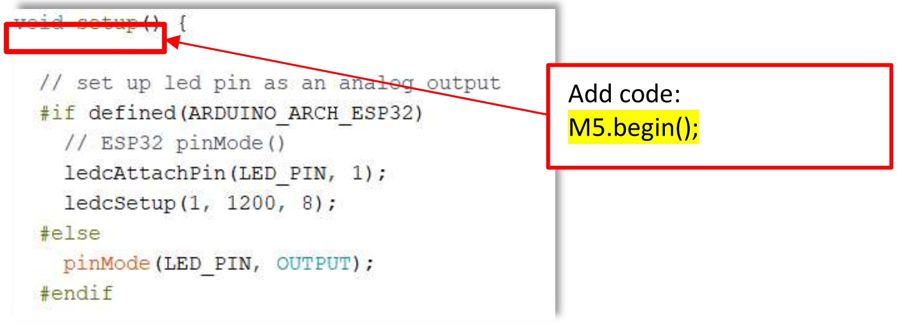
Modify to: analog_led



```
void handleMessage(AdafruitIO_Data *data) {

  // convert the data to integer
  int reading = data->toInt();

  Serial.print("received <- ");
  Serial.println(reading);

  // write the current 'reading' to the led
  #if defined(ARDUINO_ARCH_ESP32)
    ledcWrite(1, reading); // ESP32 analogWrite()
  #else
    analogWrite(LED_PIN, reading);
  #endif

}
```

Add code:
current = reading;
analog_lgage->save(current/255.0*100);
Serial.print("sending <- ");
Serial.println(current/255.0*100);

5. Add the following codes to **void setup()** display IP address to M5StickC Plus display screen:

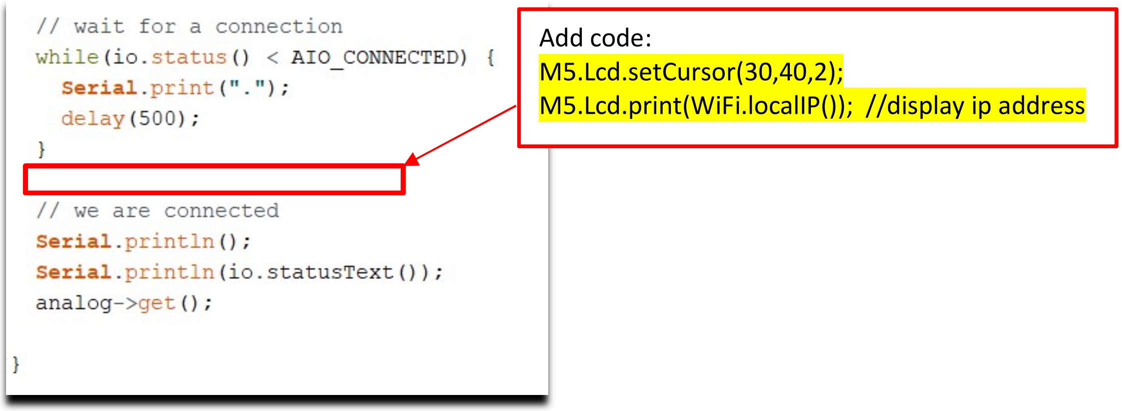


```
void setup() {

  // set up led pin as an analog output
  #if defined(ARDUINO_ARCH_ESP32)
    // ESP32 pinMode()
    ledcAttachPin(LED_PIN, 1);
    ledcSetup(1, 1200, 8);
  #else
    pinMode(LED_PIN, OUTPUT);
  #endif
```

Add code:
M5.begin();

. . . .

. . . .



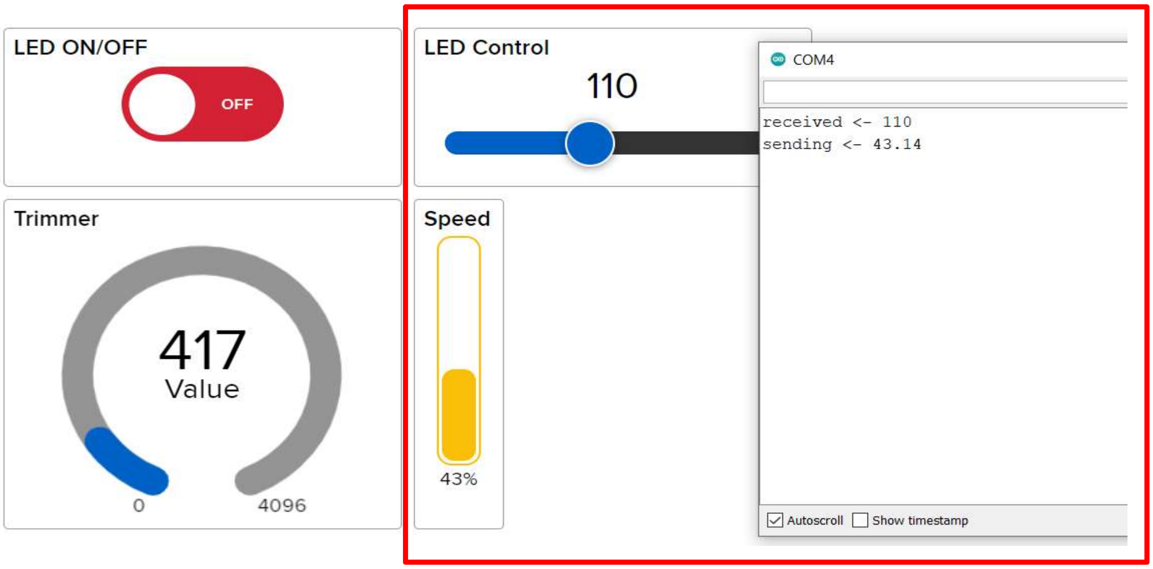
```
  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  // we are connected
  Serial.println();
  Serial.println(io.statusText());
  analog->get();

}
```

Add code:
M5.Lcd.setCursor(30,40,2);
M5.Lcd.print(WiFi.localIP()); //display ip address

6. Save the code and compile to ensure no errors. Load the code to the M5StickC device.

7. Once loading is completed and IP address is displayed on the M5StickC device, the device is connected to the network.

8. Drag the slider to obtain different values, and observe the Line Gauge changes in display.



```
received <- 110
sending <- 43.14
```

9. Note also that if an LED is connected to the PIN G26 of the M5StickC device, the LED will light up and gradually increase in brightness and as the slider value increases or decreases.

10. Why does the input reading for the Trimmer range from 0 to 4095 while the output for LED Control range from 0 to 255?

11. Summary: Data received from the cloud can be used to control the device. Data captured on device from cloud or from sensors can be processed and sent back to the cloud. Cloud will be able to display and alerts to users remotely.