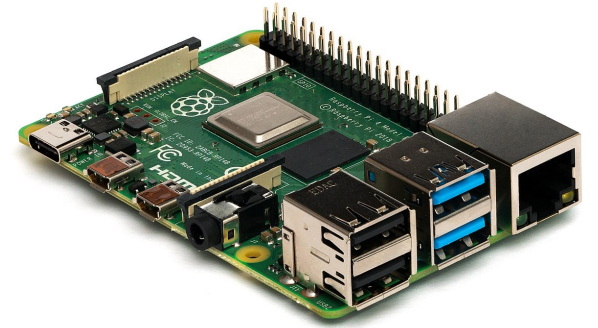


# 2023 STiCS Raspberry Pi Session

# Raspberry Pi

- Setup
- Python Hello World
- Blink LED
- Switch Controlled LED
- Internet of Things (IoT)
- Turn on LED
- Send Switch Data



<https://trickel.org/thomas/skc/RPi.html>

## 2023 Spring Technology in the Classroom Showcase

### Raspberry Pi

- [CanaKit Raspberry Pi Quick Start Guide](#)
- [Blink LED Python code](#)
- [Switch Controlled LED Python code](#)

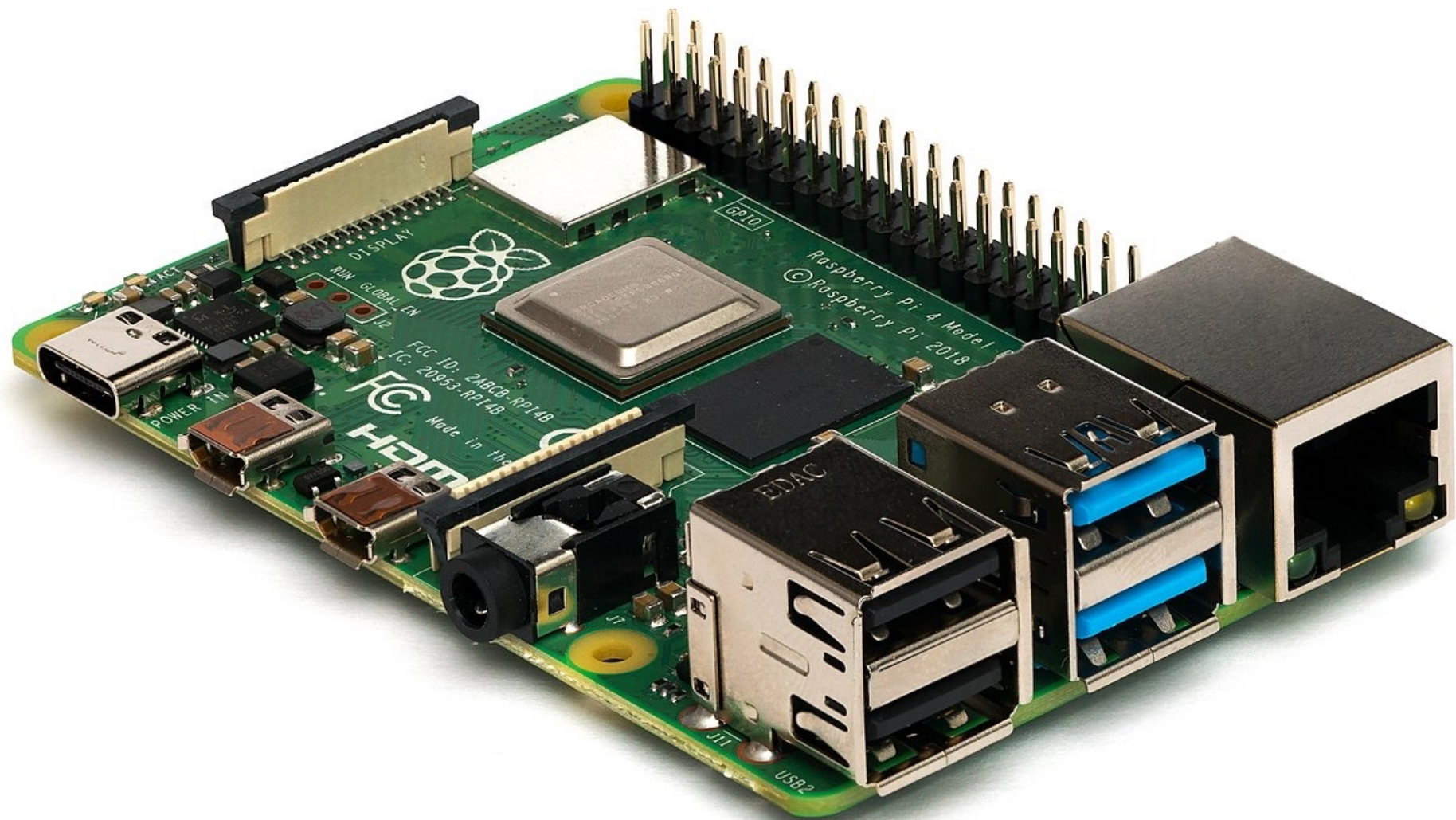
### Internet of Things

- [Welcome to Adafruit IO](#)
- [Digital Output from adafruit IO](#)
- [Digital Input to adafruit IO](#)

### Useful Links

- [Raspberry Pi Foundation Web Site](#)
- **Adafruit IO**
  - [Adafruit IO Basics: Feeds](#)
  - [Adafruit IO Basics: Dashboards](#)
  - [Adafruit IO Blocks](#)

# Raspberry Pi

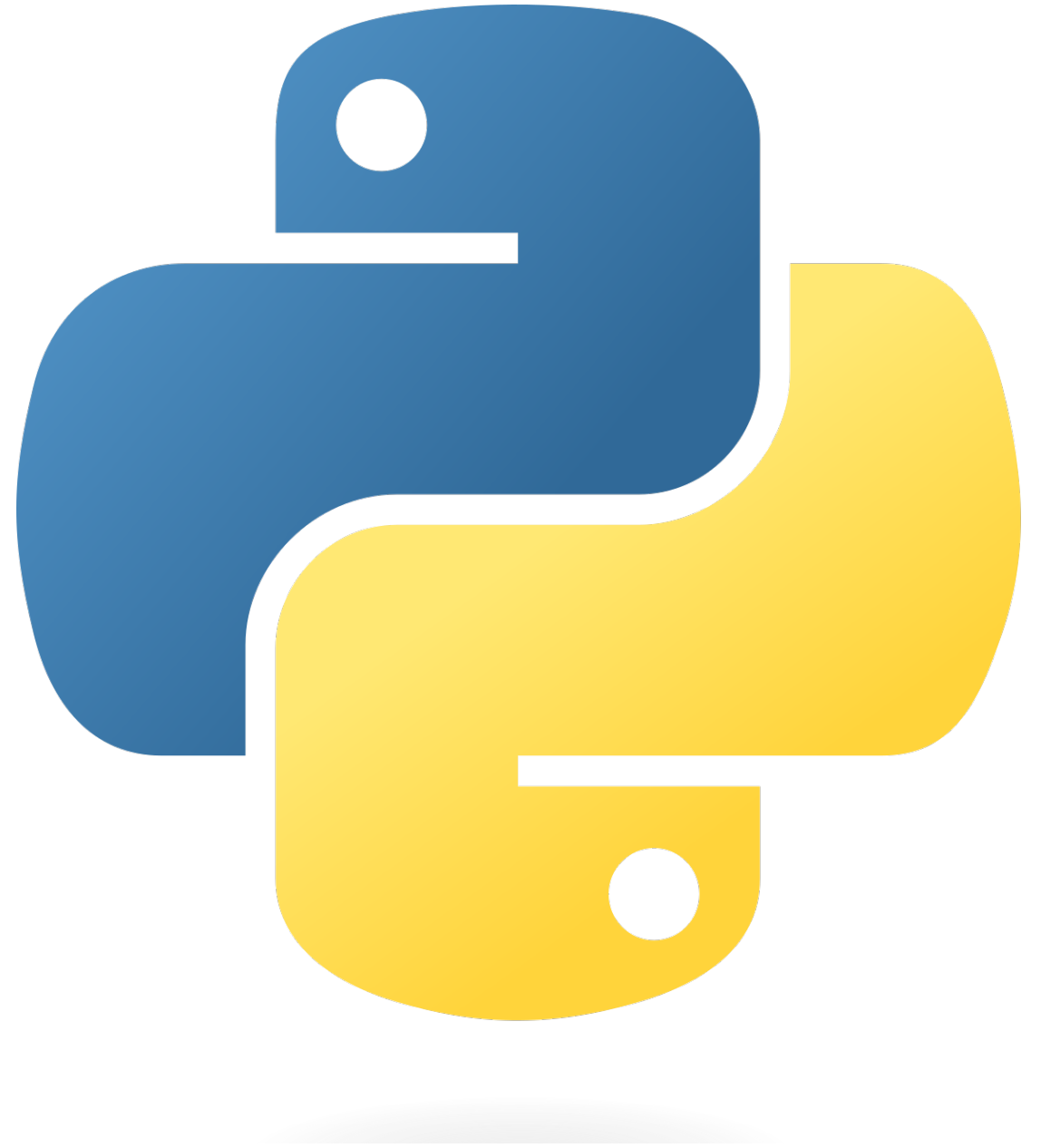


# Python Programming

- Raspberry Icon/Programming/Thonny

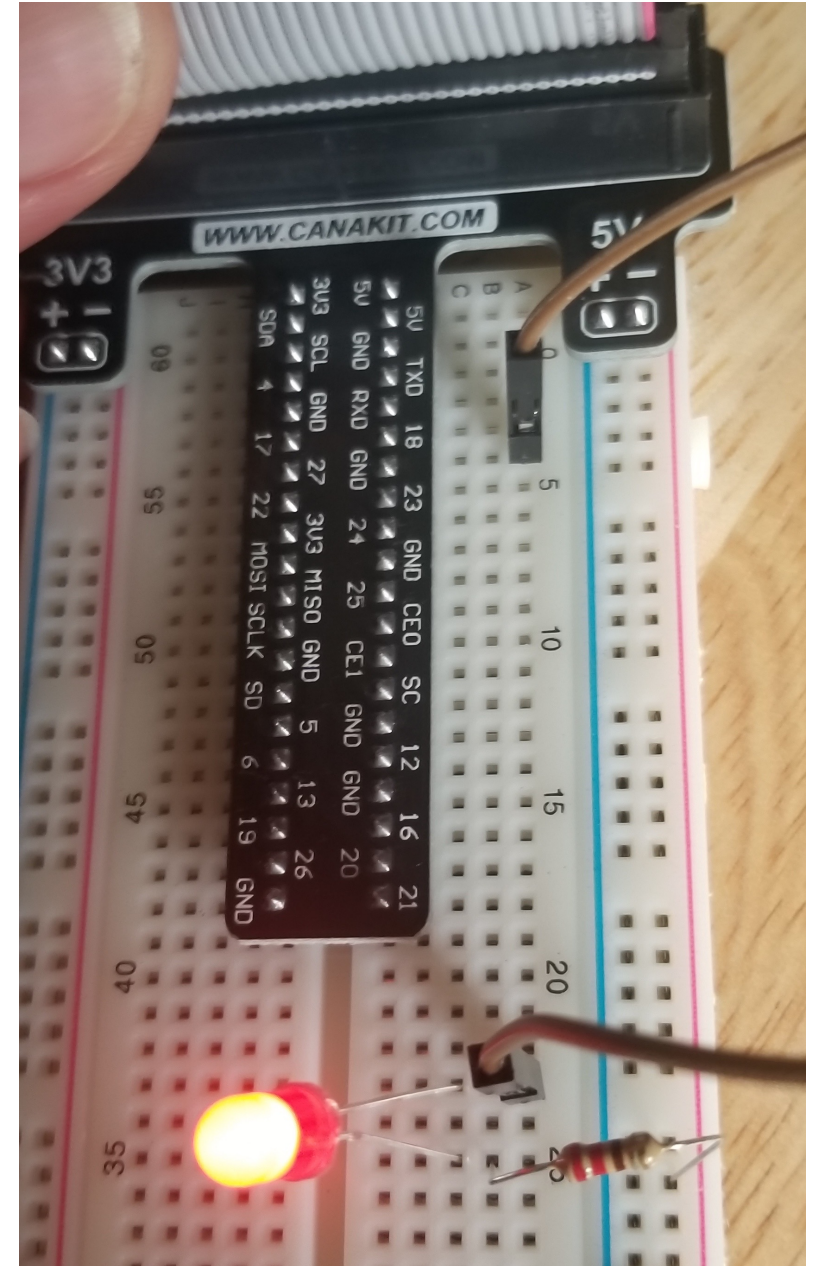
```
print( "Hello World" )
```

- Run Program



# Blink LED - Hardware

- Digital Output
- Polarity matters on LED
  - Short leg is Cathode, connect to – (GND)
- Resistor 220 $\Omega$  (Red Red Brown)
- Wire
  - LED Anode to GPIO 18



# Blink LED – Python Code

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setwarnings(False)
5 GPIO.setmode(GPIO.BCM)
6 GPIO.setup(18, GPIO.OUT)
7
8 while True:
9     GPIO.output(18, True)
10    time.sleep(1)
11    GPIO.output(18, False)
12    time.sleep(1)
```

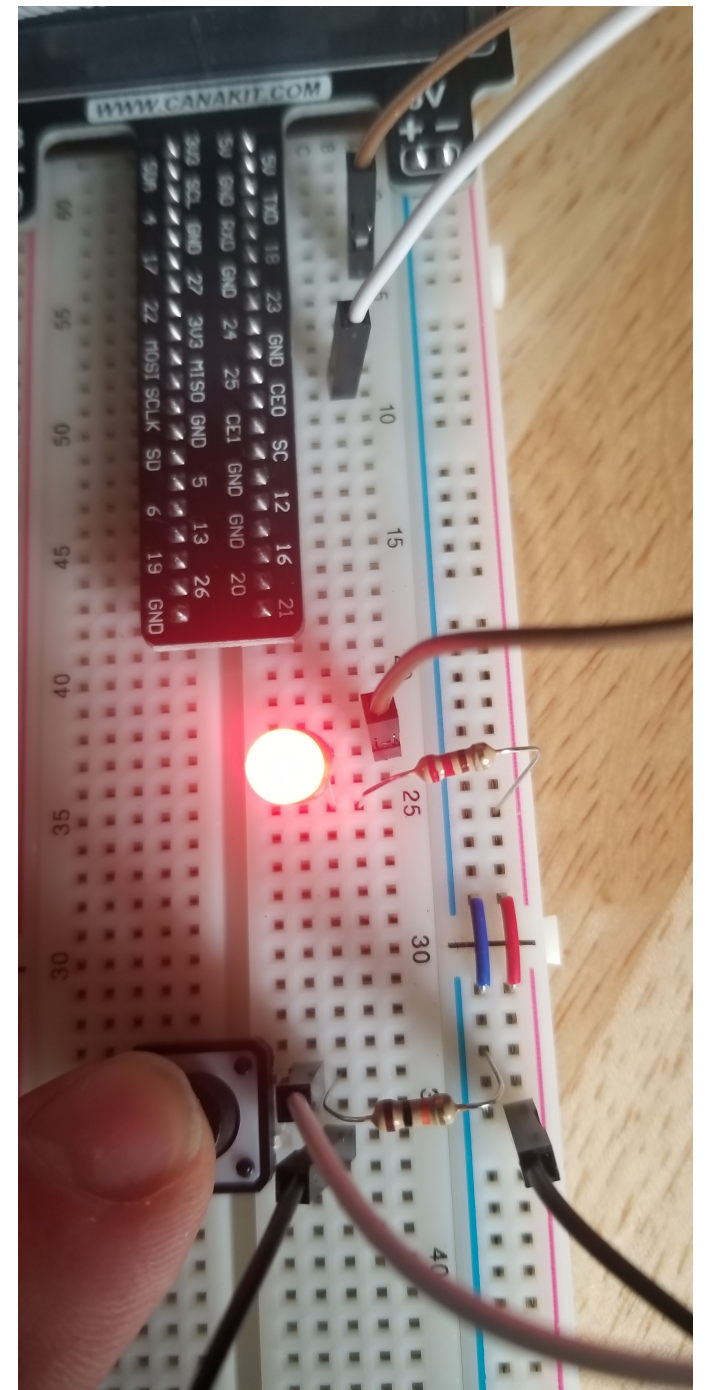
# Make the LED blink faster



```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setwarnings(False)
5  GPIO.setmode(GPIO.BCM)
6  GPIO.setup(18, GPIO.OUT)
7
8  while True:
9      GPIO.output(18, True)
10     time.sleep(1)
11     GPIO.output(18, False)
12     time.sleep(1)
```

# Switch Controlled LED - Hardware

- Digital Input
- Resistor 10k $\Omega$  (Orange Black Brown)
- Wire
  - Switch to GPIO 25



# Switch Controlled LED

## - Python Code

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setwarnings(False)
5 GPIO.setmode(GPIO.BCM)
6 GPIO.setup(18, GPIO.OUT)
7 GPIO.setup(25, GPIO.IN)
8
9 ▼ while True:
10 ▼     if GPIO.input(25):
11 └         GPIO.output(18, False)
12 ▼     else:
13 └         GPIO.output(18, True)
```

Make the LED  
blink when the  
switch is pressed



```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setwarnings(False)
5  GPIO.setmode(GPIO.BCM)
6  GPIO.setup(18, GPIO.OUT)
7  GPIO.setup(25, GPIO.IN)
8
9  ▼ while True:
10 ▼     if GPIO.input(25):
11  └         GPIO.output(18, False)
12 ▼     else:
13  └         GPIO.output(18, True)
```

Internet of things

# IOT



# Adafruit IO

- Internet of Things (IoT)
- Feeds and Dashboards
- Control LED from AIO (digital output)
- Send Data to AIO (digital input)
- Send CO2 data to AIO (analog input)

# Setup AIO Account

- Sign Up
- Two factor authentication

# Feeds & Dashboards

- Think of a Feed as property of an IoT device
  - Feeds are Features of an IoT device
    - Weather Station – temperature, humidity, wind speed, etc
    - Thermostat – set temperature, on/off
    - Door Lock
- Data is sent to and comes from Feeds
- Dashboards are used to
  - Display Feed data
  - Send data to a Feed

# Control LED from AIO

- Hardware is the same
- Create Feed
- Create Dashboard
- Write Python Code

# Create AIO Feed



ttrickel / Feeds

[+ New Feed](#) [+ New Group](#)

Default

Feed Name	Key	Last value
<input type="checkbox"/> CO2	co2	923
<input type="checkbox"/> ControlLED	controlled	ON
<input type="checkbox"/> ...	...	...

### Create a new Feed ✕

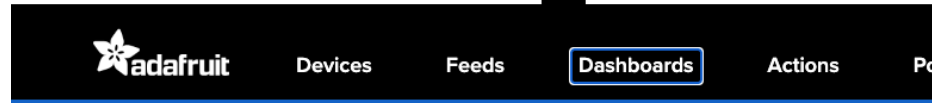
**Name**

Maximum length: 128 characters. Used: 15

**Description**

[Cancel](#) [Create](#)

# Create AIO Dashboard



ttrickel / Dashboards

+ New Dashboard

## Dashboards

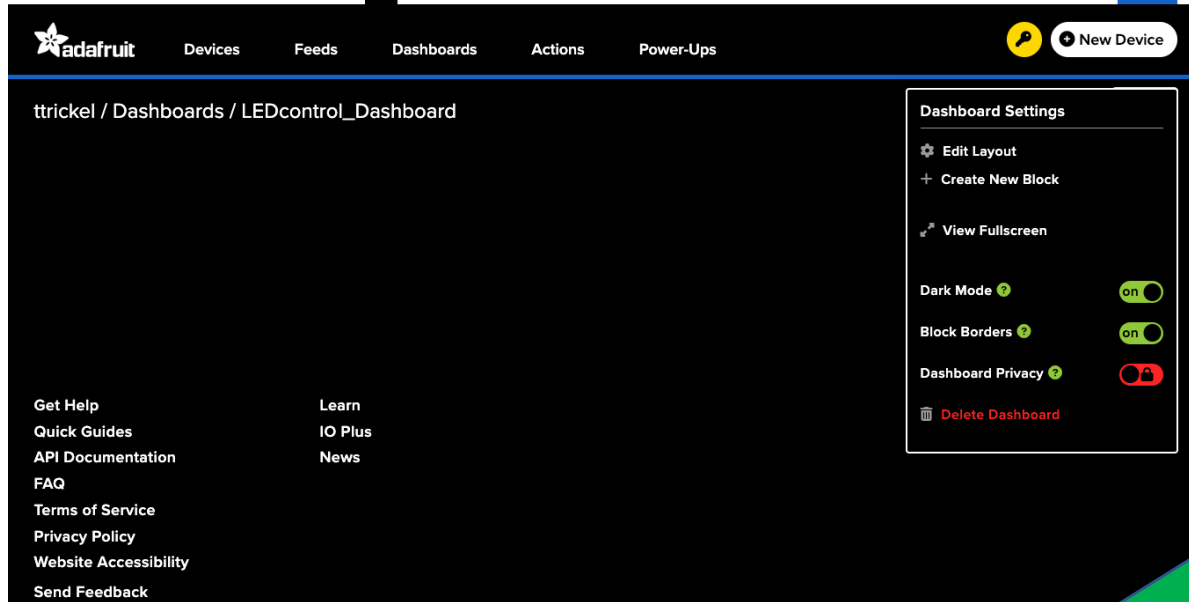
<input type="checkbox"/> Name	Key	C
<input type="checkbox"/> CO2	co2	O
<input type="checkbox"/> ControlLED	controlled	Ji
<input type="checkbox"/> LightSensor	lightsensor	Ji
<input type="checkbox"/> Test	test	Ji

### Create a new Dashboard ✕

**Name**

**Description**

# Create AIO Dashboard continued

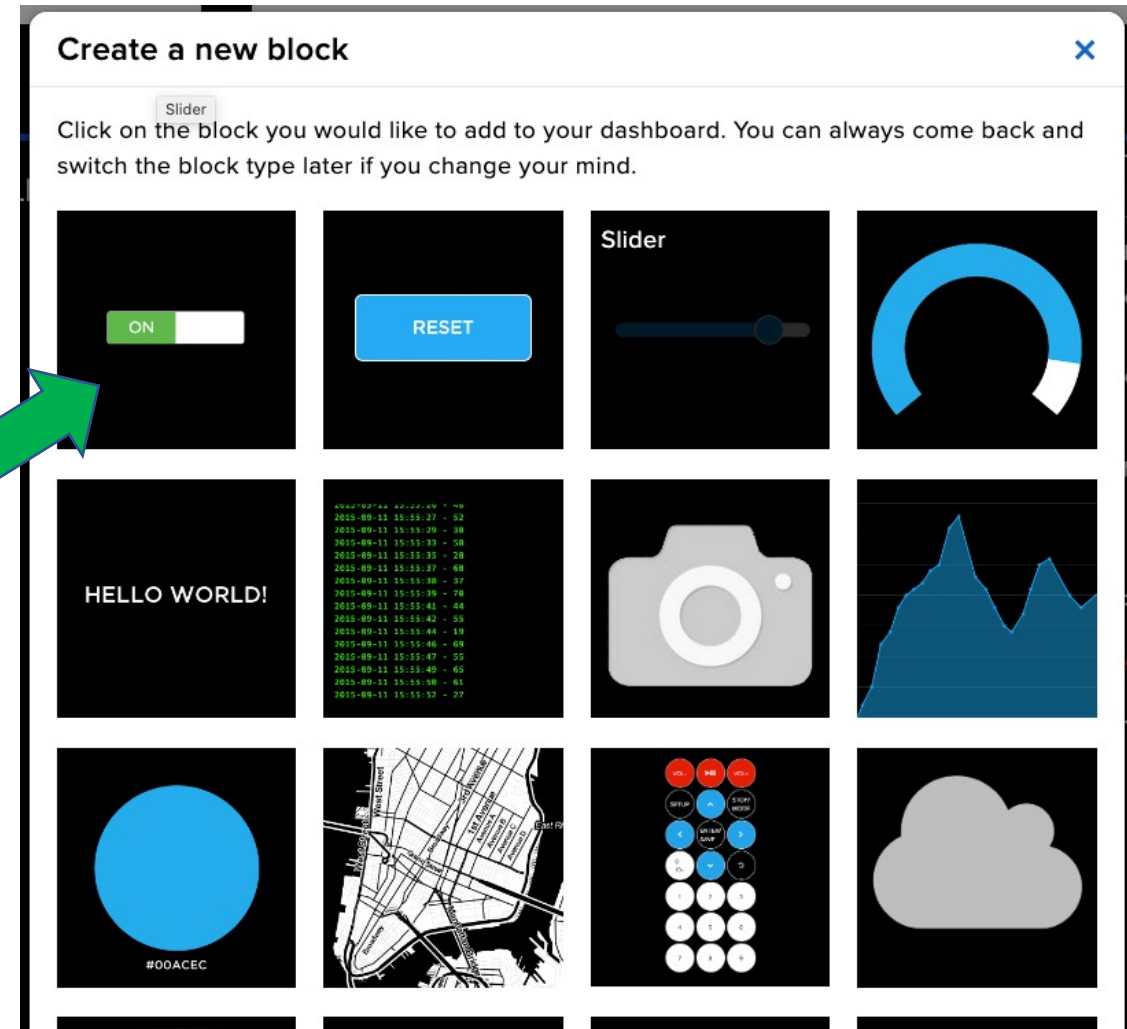


The screenshot shows the top navigation bar of the Adafruit dashboard with the following items: [Devices](#), [Feeds](#), [Dashboards](#), [Actions](#), [Power-Ups](#), and a [New Device](#) button. The breadcrumb trail reads "ttrickel / Dashboards / LEDcontrol\_Dashboard". A "Dashboard Settings" panel is open on the right, containing the following options:

- [Edit Layout](#)
- [Create New Block](#)
- [View Fullscreen](#)
- Dark Mode:
- Block Borders:
- Dashboard Privacy:
- [Delete Dashboard](#)

On the left side of the dashboard, there is a sidebar menu with the following links:

- [Get Help](#)
- [Quick Guides](#)
- [API Documentation](#)
- [FAQ](#)
- [Terms of Service](#)
- [Privacy Policy](#)
- [Website Accessibility](#)
- [Send Feedback](#)
- [Learn](#)
- [IO Plus](#)
- [News](#)



The "Create a new block" dialog box displays a grid of block types. A green arrow points from the "ON" block in the first row to the "Slider" block in the second row. The dialog includes the following text and options:

**Create a new block** [Close]

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.

Slider

The grid contains the following blocks:

- ON (toggle)
- RESET (button)
- Slider (horizontal slider)
- Progress indicator (circular gauge)
- HELLO WORLD! (text)
- Terminal (code output)
- Camera (camera icon)
- Area chart (line graph)
- #00ACEC (color swatch)
- Map (street map)
- Keypad (numeric keypad)
- Cloud (cloud icon)

# Create AIO Dashboard continued

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

## Default

Feed Name	Last value	Recorded	
<input type="checkbox"/> CO2	923	7 months	🔒
<input type="checkbox"/> ControlLED	ON	2 days	🔒
<input type="checkbox"/> humidityReading	716.827392578...	7 months	🔒
<input type="checkbox"/> ledControl	off	2 days	🔒
<input checked="" type="checkbox"/> LEDcontrol_Feed		5 minutes	🔒
<input type="checkbox"/> lightReading	0	over 4 years	🔒
<input type="checkbox"/> runTime	9:6:0	over 4 years	🔒
<input type="checkbox"/> switch	0	2 days	🔒
<input type="checkbox"/> switchTest	0	1 day	🔒
<input type="checkbox"/> temperatureReading	698.892364501...	7 months	🔒

1 of 1 feeds selected

< Previous step   Next step >

click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Button On Text

Limit of 6 characters for the toggle text. Use the block title to be more descriptive.

Button On Value (uses On Text if blank)

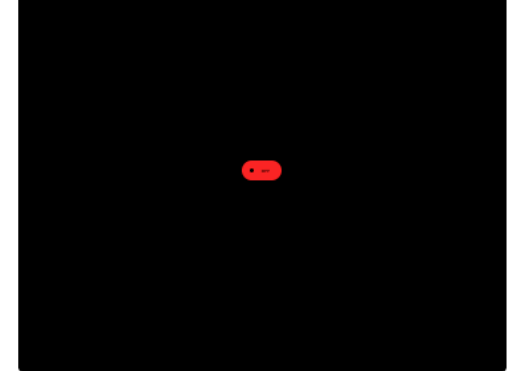
Button Off Text

Limit of 6 characters for the toggle text. Use the block title to be more descriptive.

Button Off Value (uses Off Text if blank)

Block Preview

LED Control



Toggle 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.

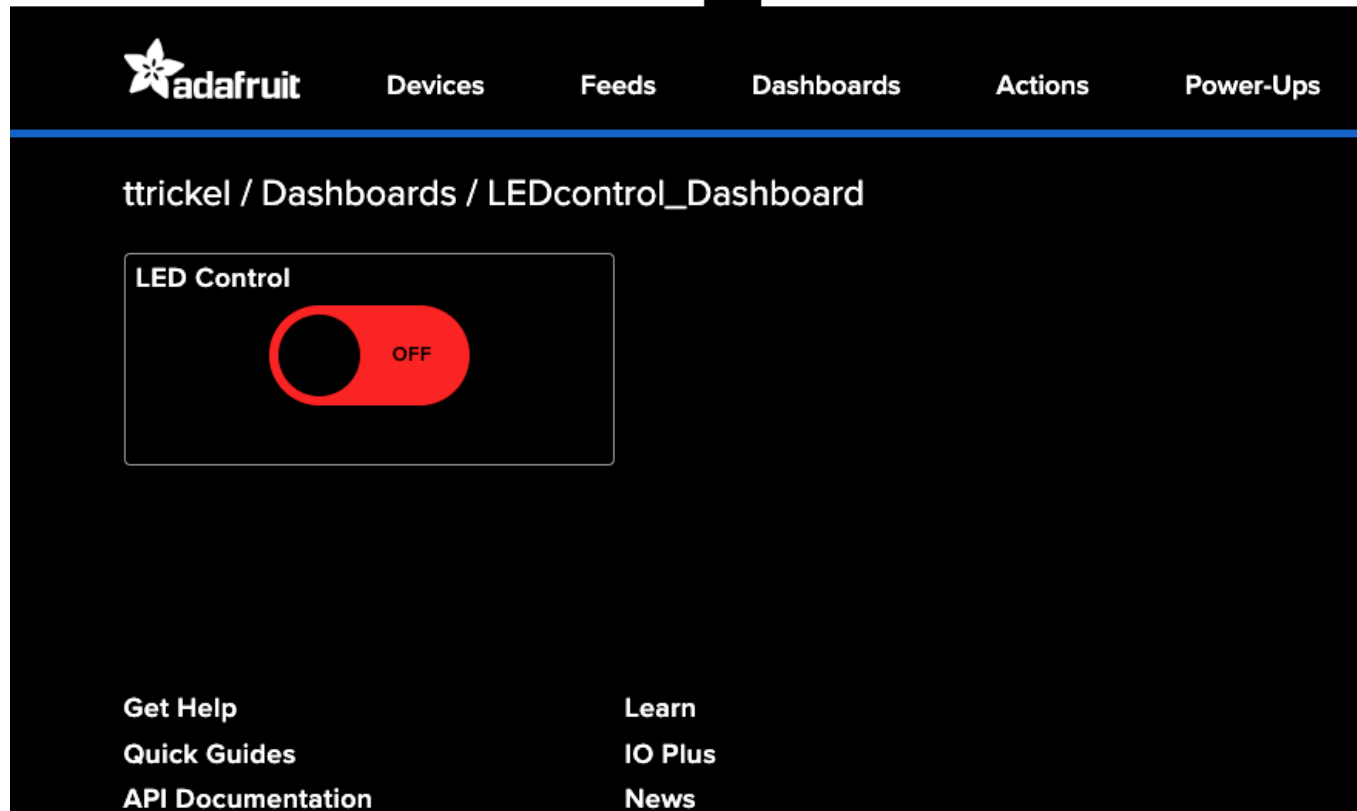
Test Value

Published Value

0 bytes

< Previous step   Create block

# Create AIO Dashboard continued



# Control LED from AIO Python Code

change

YOUR\_AIO\_KEY

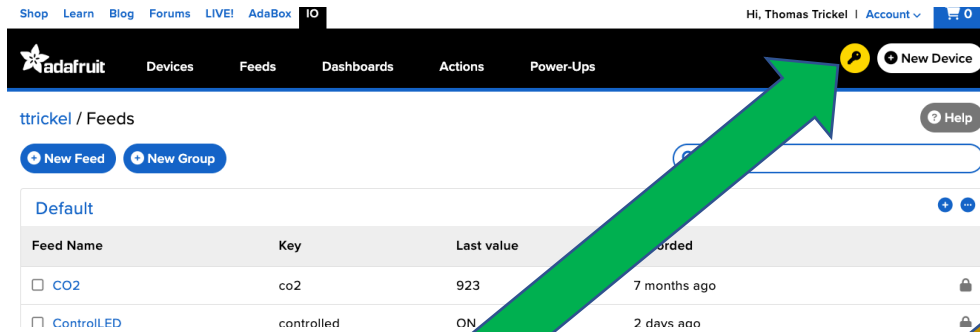


YOUR\_AIO\_USERNAME



```
1  """
2  'digital_out.py'
3  =====
4  Example of turning on and off a LED
5  from the Adafruit IO Python Client
6  Author(s): Brent Rubell, Todd Treece
7             with modifications by Thomas Trickel
8  """
9  # Import standard python modules
10 import time
11 import RPi.GPIO as GPIO
12
13 # import Adafruit IO REST client.
14 from Adafruit_IO import Client, Feed, RequestError
15
16 # setup GPIO
17 GPIO.setwarnings(False)
18 GPIO.setmode(GPIO.BCM)
19 GPIO.setup(18, GPIO.OUT)
20
21
22 # Set to your Adafruit IO key.
23 # Remember, your key is a secret,
24 # so make sure not to publish it when you publish this code!
25 ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'
26
27 # Set to your Adafruit IO username.
28 # (go to https://accounts.adafruit.com to find your username)
29 ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'
30
31 # Create an instance of the REST client.
32 aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)
33
34 try: # if we have a 'digital' feed
35     digital = aio.feeds('ledcontrol-feed')
36 except RequestError: # create a digital feed
37     feed = Feed(name="ledcontrol-feed")
38     digital = aio.create_feed(feed)
39
40 while True:
41     data = aio.receive(digital.key)
42     if data.value == "ON":
43         print('received <- ON\n')
44         GPIO.output(18, True)
45     elif data.value == "OFF":
46         print('received <- OFF\n')
47         GPIO.output(18, False)
48
49 # timeout so we dont flood adafruit-io with requests
50 time.sleep(0.5)
51
```

# AIO KEY & USERNAME



Shop Learn Blog Forums LIVE! AdaBox IO Hi, Thomas Trickel | Account

adafruit Devices Feeds Dashboards Actions Power-Ups New Device

ttrickel / Feeds Help

New Feed New Group

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> CO2	co2	923	7 months ago
<input type="checkbox"/> ControlLED	controlled	ON	2 days ago

## YOUR ADAFRUIT IO 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 Adafruit IO 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 Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.

Username

Active Key

REGENERATE KEY

[Hide Code Samples](#)

Arduino

```
#define IO_USERNAME "ttrickel"
#define IO_KEY      "a0da523dfa16502494a70e61e1f683ba8a0b32f4"
```

Linux Shell

```
export IO_USERNAME="ttrickel"
export IO_KEY="a0da523dfa16502494a70e61e1f683ba8a0b32f4"
```

Scripting

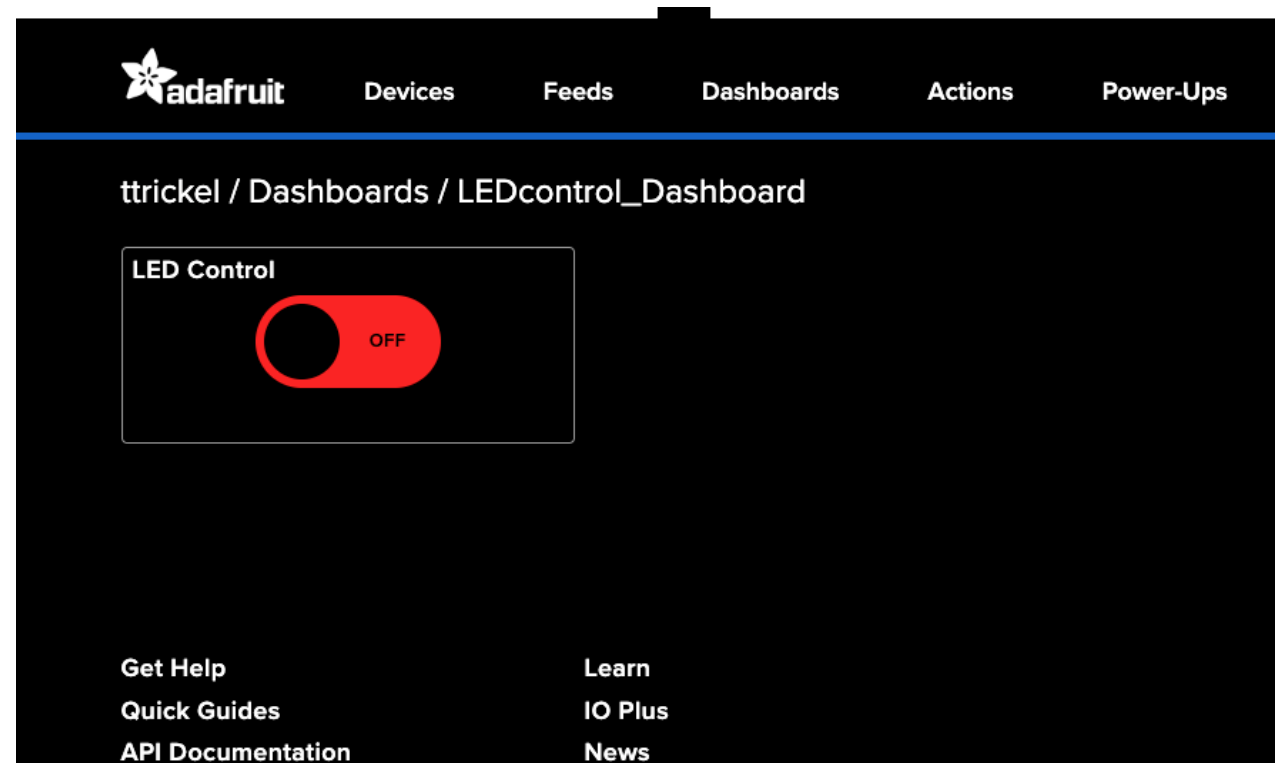
```
ADAFRUIT_IO_USERNAME = "ttrickel"
ADAFRUIT_IO_KEY = "a0da523dfa16502494a70e61e1f683ba8a0b32f4"
```

# Test It

## Run Python Code



## Click on LED Control icon



# Send Switch Data to AIO

- Hardware is the same
- Create Feed
- Create Dashboard
- Write Python Code

# Create AIO Feed

ttrickel / Feeds

[+ New Feed](#) [+ New Group](#)

Default

Feed Name	Key	Last value
<input type="checkbox"/> CO2	co2	923
<input type="checkbox"/> ControlLED	controlled	ON
<input type="checkbox"/> ...	...	...



### Create a new Feed ✕

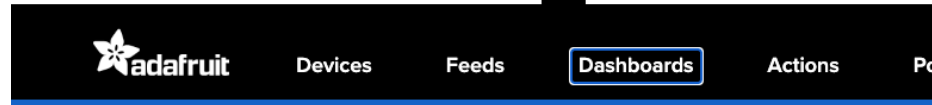
**Name**

Maximum length: 128 characters. Used: 15

**Description**

[Cancel](#) [Create](#)

# Create AIO Dashboard



ttrickel / Dashboards

+ New Dashboard

Dashboards

<input type="checkbox"/> Name	Key	
<input type="checkbox"/> CO2	co2	O
<input type="checkbox"/> ControlLED	controlled	Ji
<input type="checkbox"/> LightSensor	lightsensor	Ji
<input type="checkbox"/> Test	test	Ji



### Create a new Dashboard

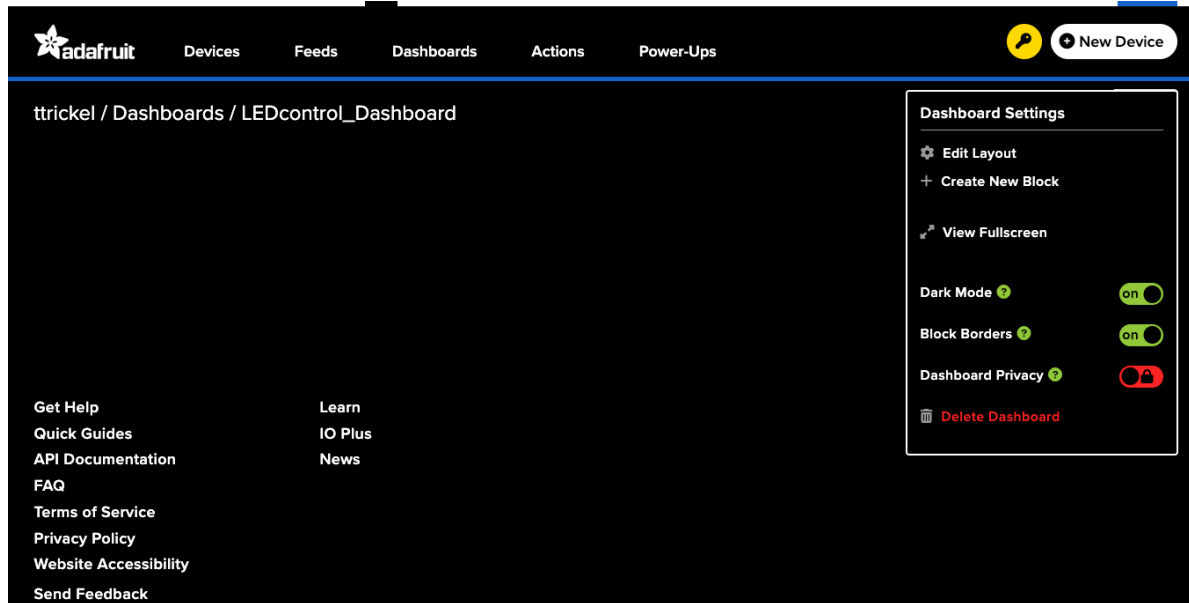
Name

SwitchRead

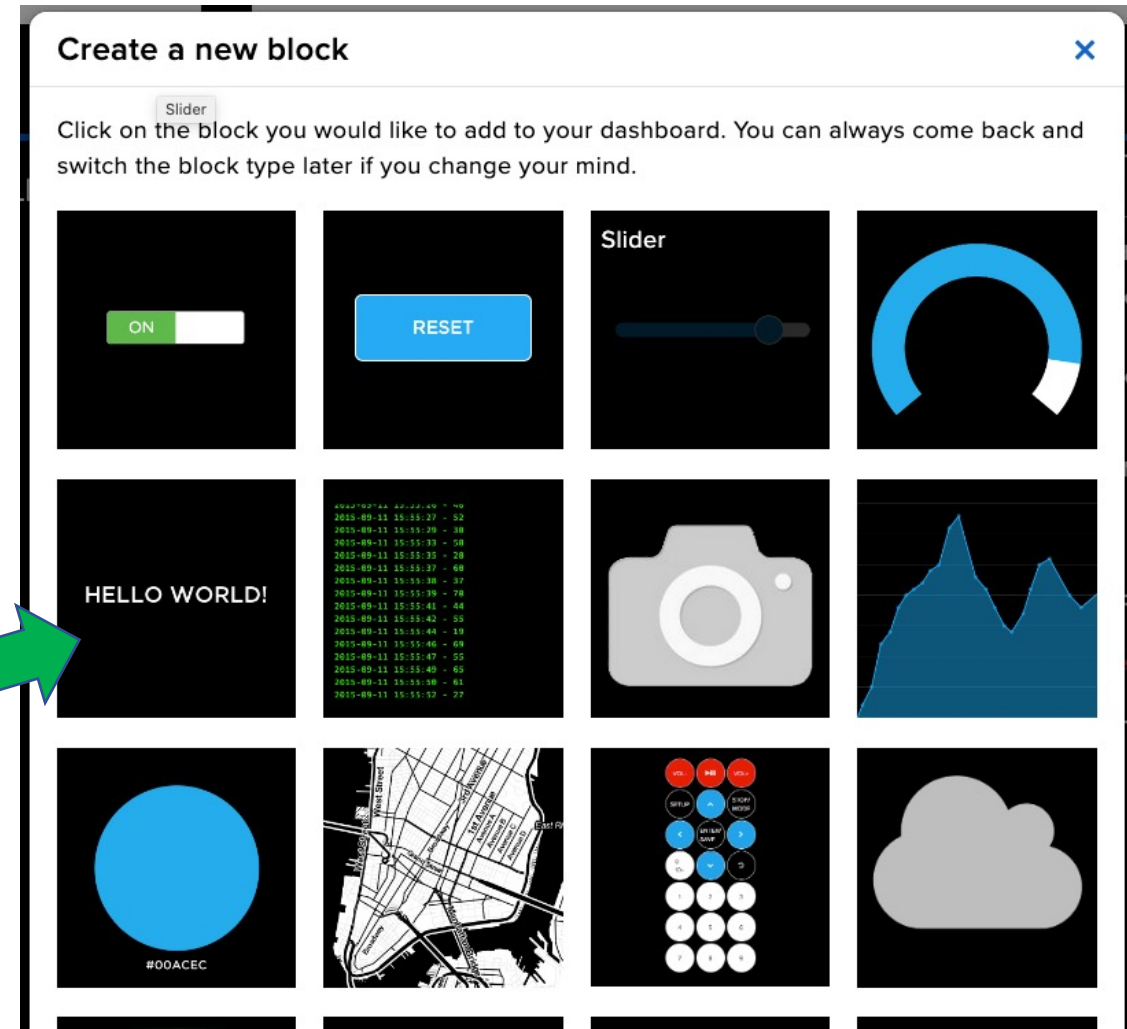
Description

Dashboard to display switch data sent to AIO

# Create AIO Dashboard continued



The screenshot shows the top navigation bar of the Adafruit dashboard with the following items: [Devices](#), [Feeds](#), [Dashboards](#), [Actions](#), [Power-Ups](#), and a [New Device](#) button. The breadcrumb path is [ttrickel / Dashboards / LEDcontrol\\_Dashboard](#). The **Dashboard Settings** menu is open, showing options: [Edit Layout](#), [Create New Block](#), and [View Fullscreen](#). Below these are three toggle switches: **Dark Mode** (on), **Block Borders** (on), and **Dashboard Privacy** (off). At the bottom of the settings menu is a [Delete Dashboard](#) option. On the left side of the dashboard, there is a vertical list of links: [Get Help](#), [Quick Guides](#), [API Documentation](#), [FAQ](#), [Terms of Service](#), [Privacy Policy](#), [Website Accessibility](#), and [Send Feedback](#). On the right side, there are links for [Learn](#), [IO Plus](#), and [News](#).



The screenshot shows a dialog box titled "Create a new block" with a close button (X) in the top right corner. Below the title, there is a "Slider" label and a paragraph of text: "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." The dialog displays a grid of block options. A green arrow points to the "HELLO WORLD!" block. The blocks shown include: a toggle switch labeled "ON", a blue "RESET" button, a slider control, a blue circular progress indicator, a "HELLO WORLD!" text block, a terminal window showing a list of timestamps and values, a camera icon, a blue area chart, a blue circle with the text "#00ACEC" below it, a map of a city street grid, a numeric keypad, and a grey cloud icon.

# Create AIO Dashboard continued

## Connect a Feed ✕

A text block can be used to send data as well as view data. To publish, click on the text block, enter any text, and press enter to send.

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

Search for a feed

Default			
Feed Name	Last value	Recorded	
<input type="checkbox"/> CO2	923	7 months	🔒
<input type="checkbox"/> ControlLED	ON	2 days	🔒
<input type="checkbox"/> humidityReading	716.827392578...	7 months	🔒
<input type="checkbox"/> ledControl	off	5 days	🔒
<input type="checkbox"/> LEDcontrol_Feed	OFF	about 2 hours	🔒
<input type="checkbox"/> lightReading	0	over 4 years	🔒
<input type="checkbox"/> runTime	9:6:0	over 4 years	🔒
<input checked="" type="checkbox"/> SwitchRead_Feed		3 minutes	🔒
<input type="checkbox"/> switchTest	0	5 days	🔒
<input type="checkbox"/> temperatureReading	698.892364501...	7 months	🔒

Enter new feed name

1 of 1 feeds selected

< Previous step Next step >

## 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)

Block Preview

Font Size

**Static Text**

When checked, ignore feed value and show the selected 'Static Text Value' all the time.

Static Text Value

When 'Static Text' is checked, use this value. Limited to 256 characters.

Decimal Places

Number of decimal places to display when value is a number. Defaults to -1 (unlimited).

**Show Icon**

When checked, show an icon with the value.

Icon

Show this icon next to the value.

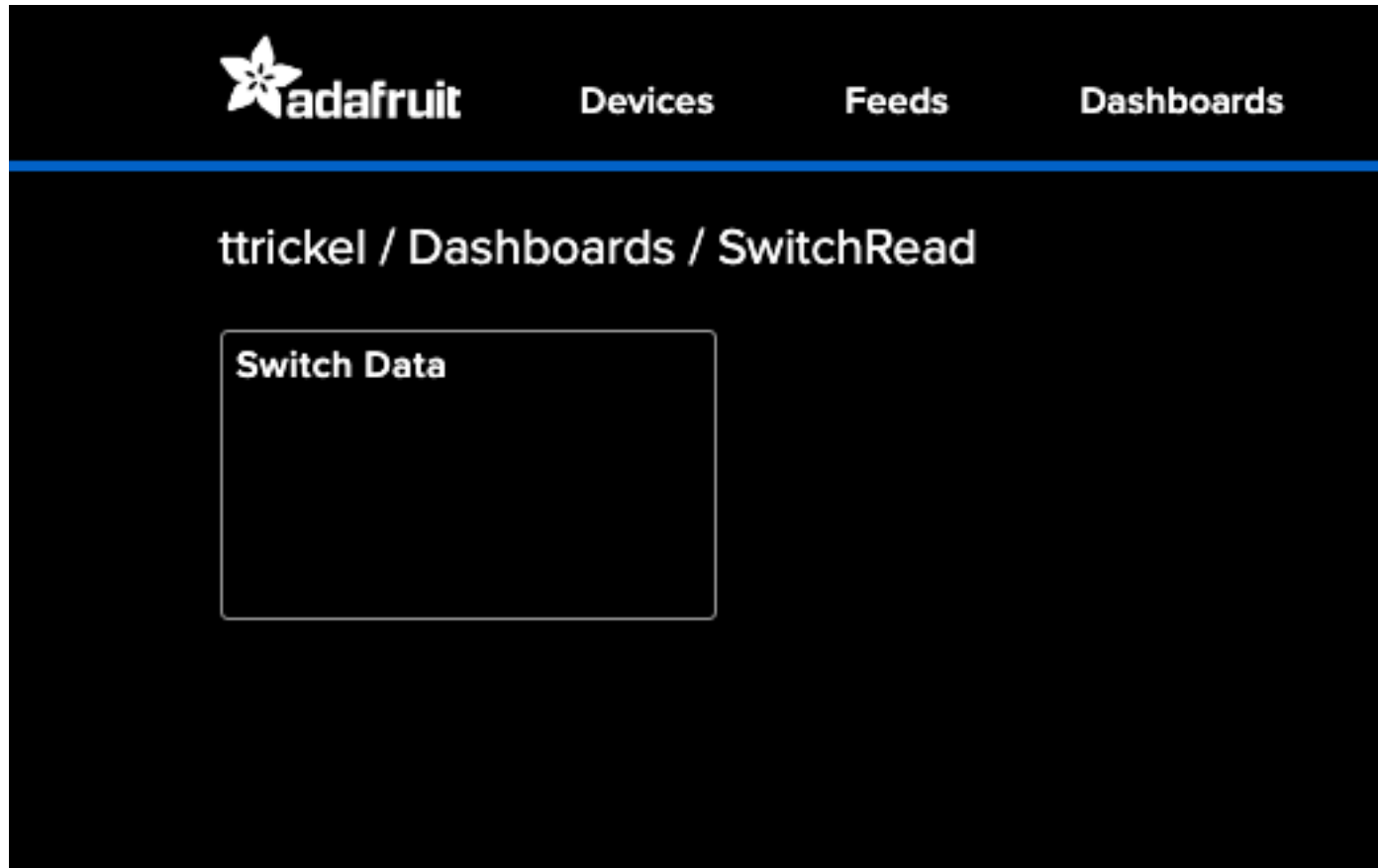
Text A text block can be used to send data as well as view data. To publish, click on the text block, enter any text, and press enter to send.

Test Value

Published Value

< Previous step Create block

# Create AIO Dashboard continued



# Send Switch Data to AIO Python Code

change

YOUR\_AIO\_KEY



YOUR\_AIO\_USERNAME



```
1  """
2  'digitalIn.py'
3  =====
4  Example of sending the status of a
5  switch to the Adafruit IO Python Client
6  """
7  # Import standard python modules
8  import time
9  import RPi.GPIO as GPIO
10
11 # import Adafruit IO REST client.
12 from Adafruit_IO import Client, Feed, RequestError
13
14 # setup GPIO
15 GPIO.setwarnings(False)
16 GPIO.setmode(GPIO.BCM)
17 GPIO.setup(25, GPIO.IN)
18
19 # Set to your Adafruit IO key.
20 # Remember, your key is a secret,
21 # so make sure not to publish it when you publish this code!
22 ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'
23
24 # Set to your Adafruit IO username.
25 # (go to https://accounts.adafruit.com to find your username)
26 ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'
27
28 # Create an instance of the REST client.
29 aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)
30
31 try: # if we have a 'digital' feed
32     digital = aio.feeds('switchread-feed')
33 except RequestError: # create a digital feed
34     feed = Feed(name="switchread-feed")
35     digital = aio.create_feed(feed)
36
37 while True:
38     if GPIO.input(25):
39         aio.send(digital.key, 0)
40     else:
41         aio.send(digital.key, 1)
42
43 # timeout so we dont flood adafruit-io with requests
44     time.sleep(0.5)
45
```

# Test It

## Run Python Code



## Press Switch on Breadboard

