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

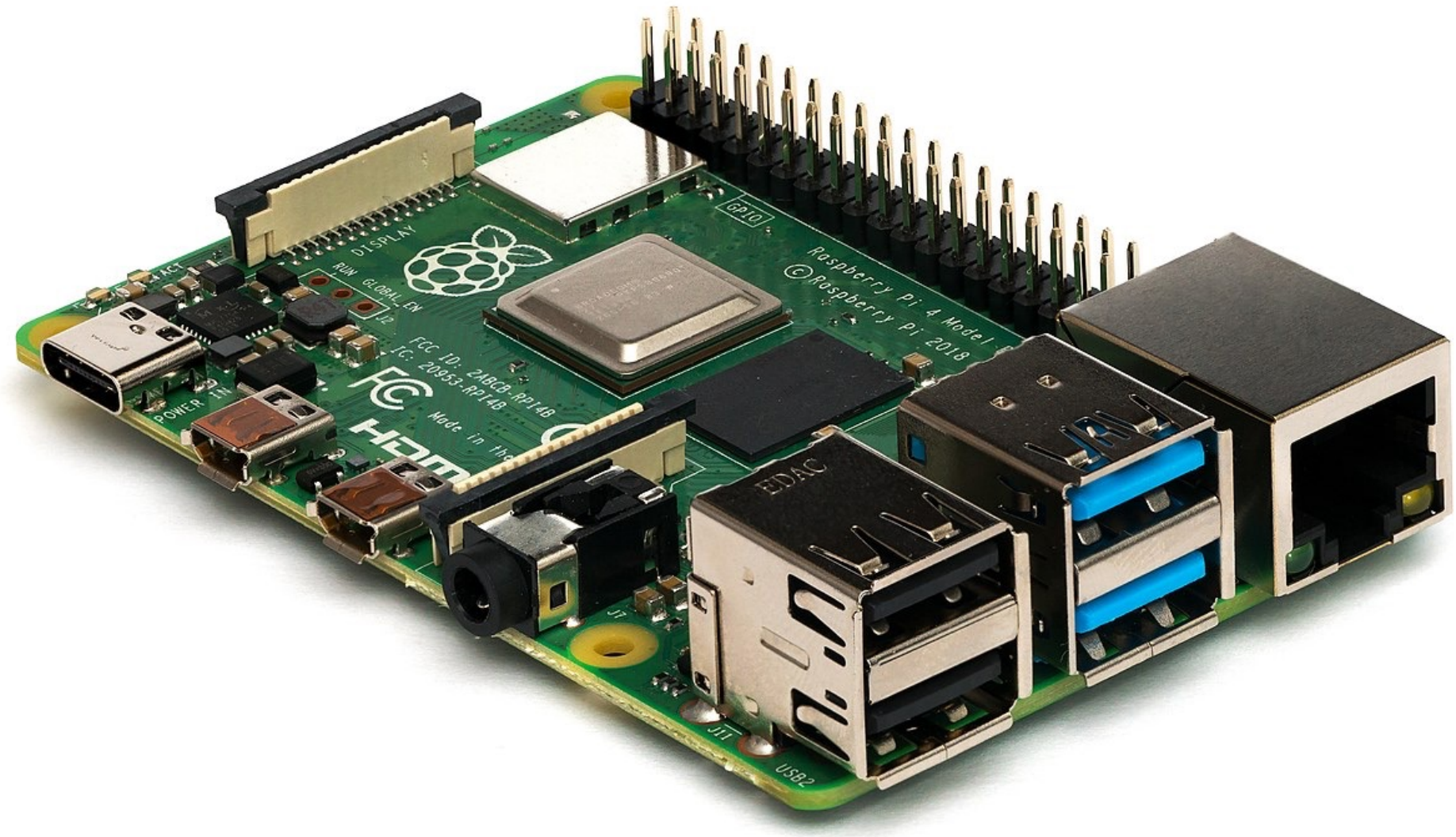
Internet of Things

- 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

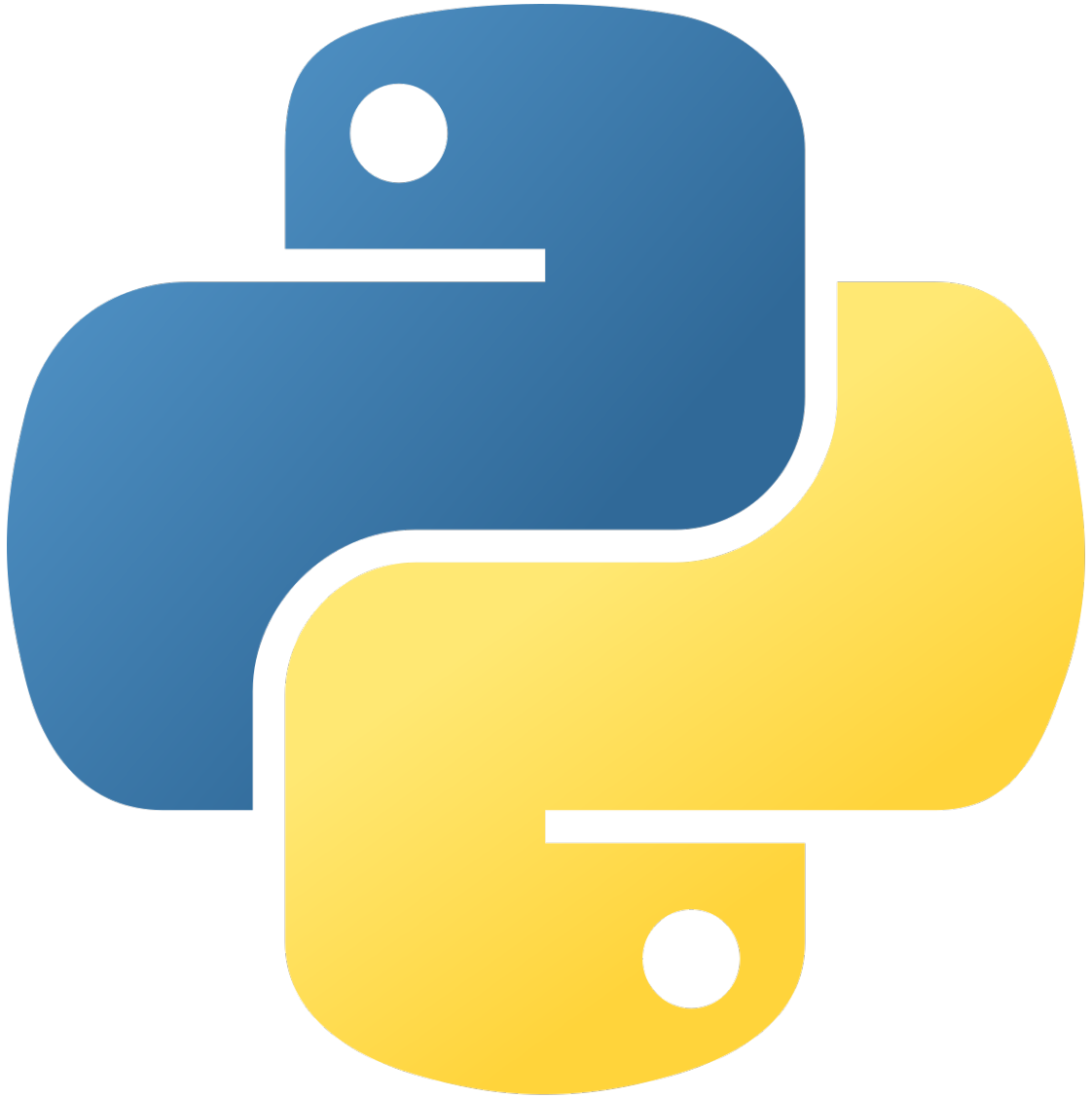


Python Programming

- Raspberry Icon/Programming/Thonny

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

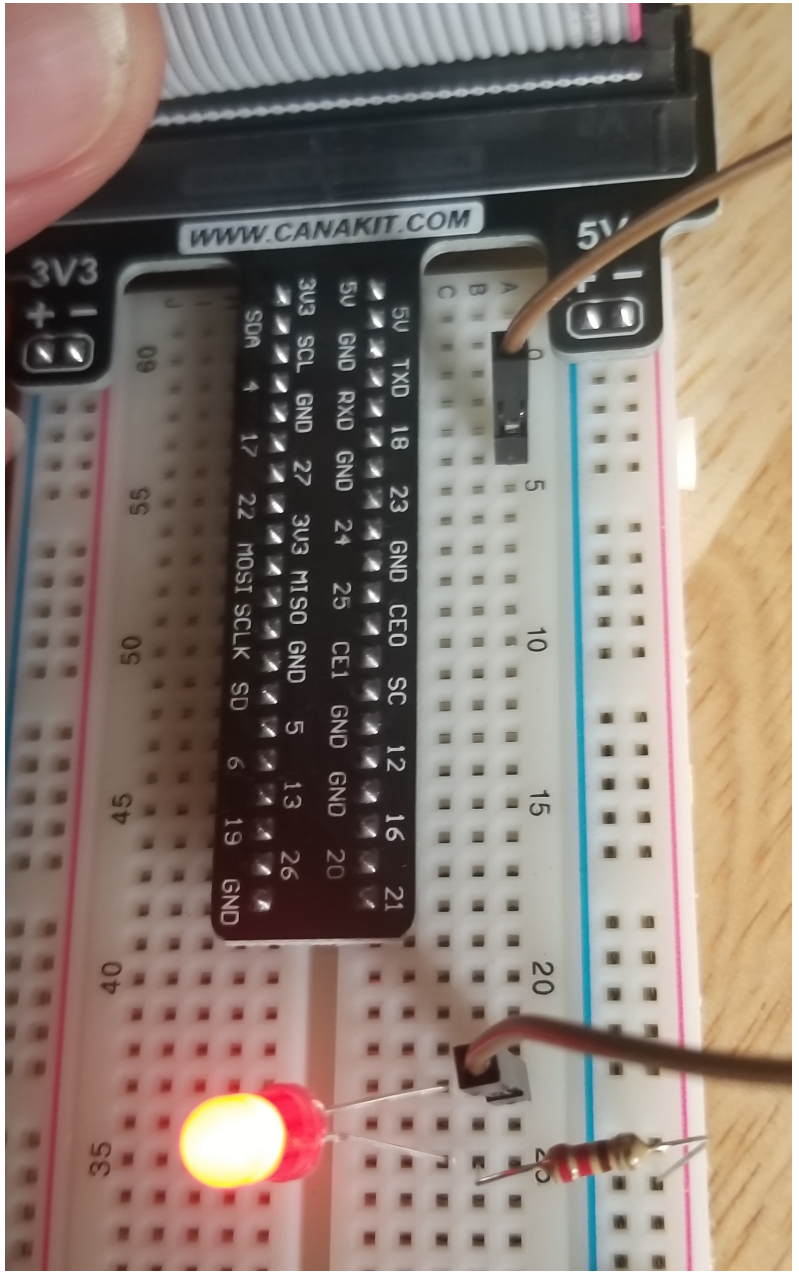
- Run Program



Blink LED - Hardware

- Digital Output

- Polarity matters on LED
  - Short leg is Cathode, connect to – (GND)
- Resistor 220Ω (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)
```

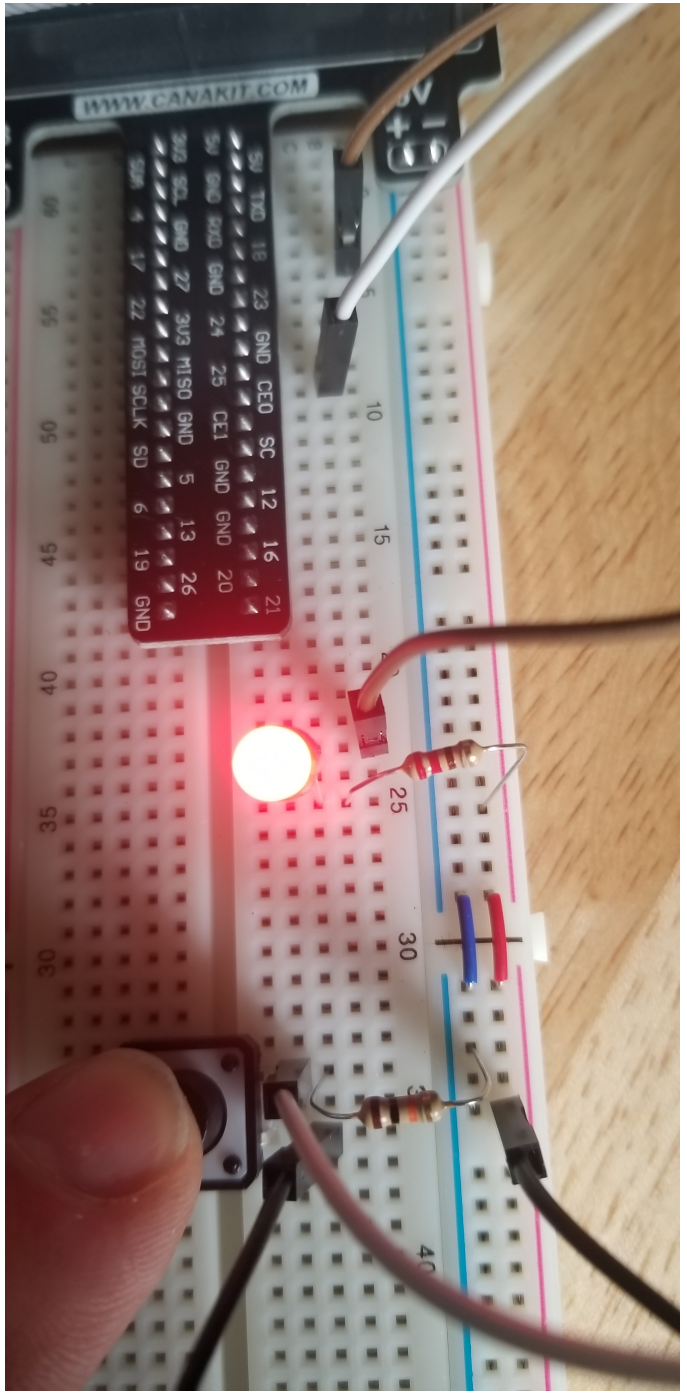
Make the LED blink faster

```
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 10kΩ (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

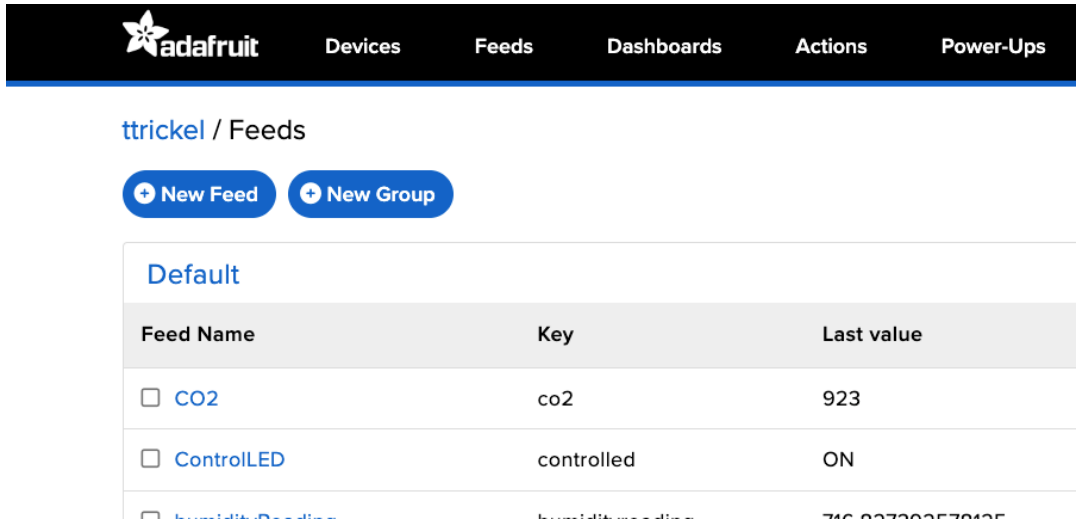
Feeds & Dashboards

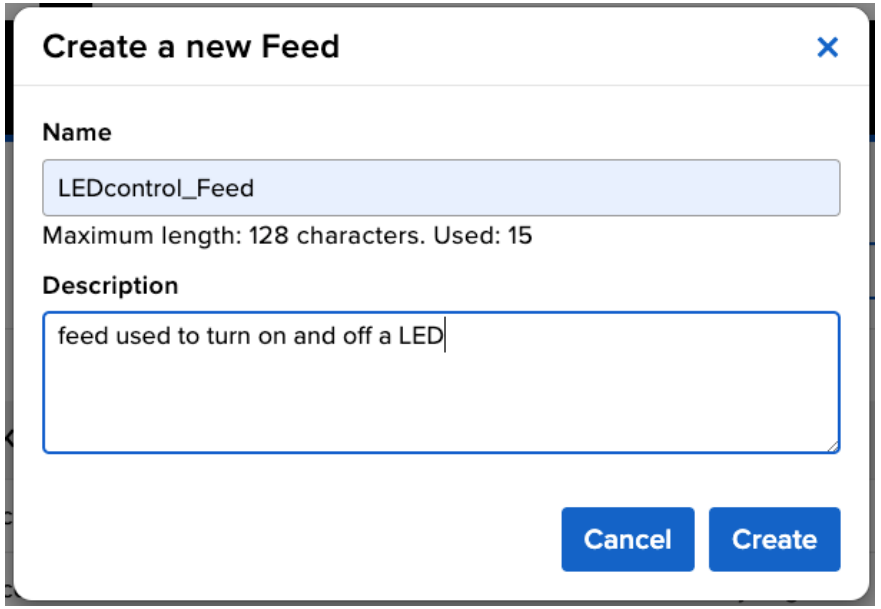
- Think of a Feed as property of an IoT device
  - Feeds are Features of an IoT device
    - Weather Station – temperature, humidity, wind speed, etc
    - Thermostat – set temperature, on/off
    - Door Lock
- Data is sent to and comes from Feeds
- Dashboards are used to
  - Display Feed data
  - Send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

Control LED from AIO

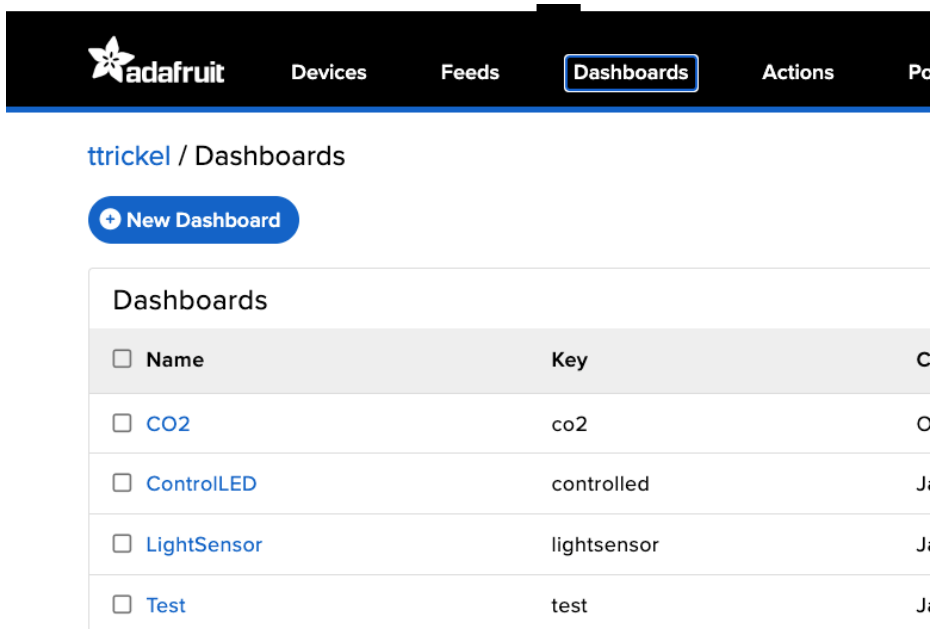
- Hardware is the same
- Create Feed
- Create Dashboard
- Write Python Code

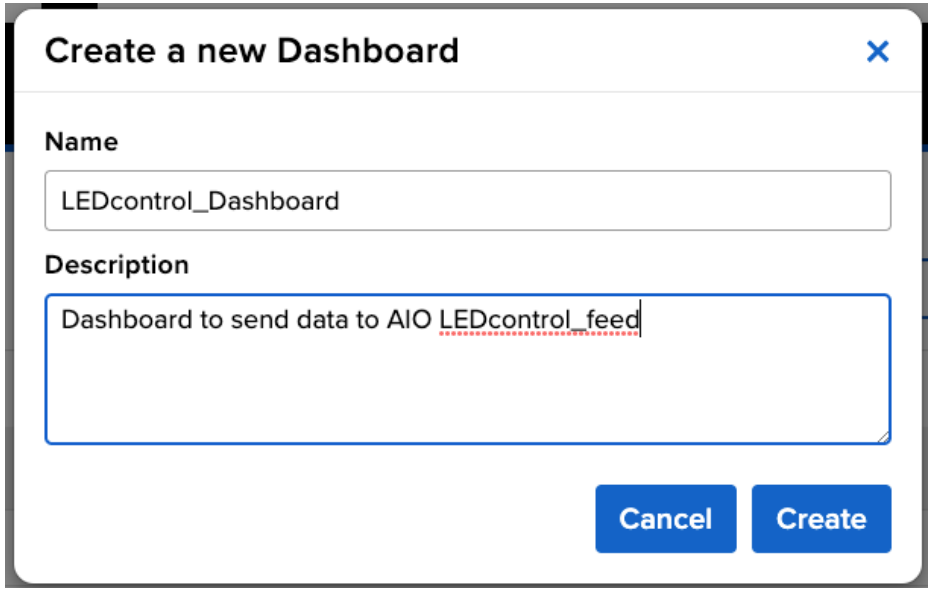
Create AIO Feed



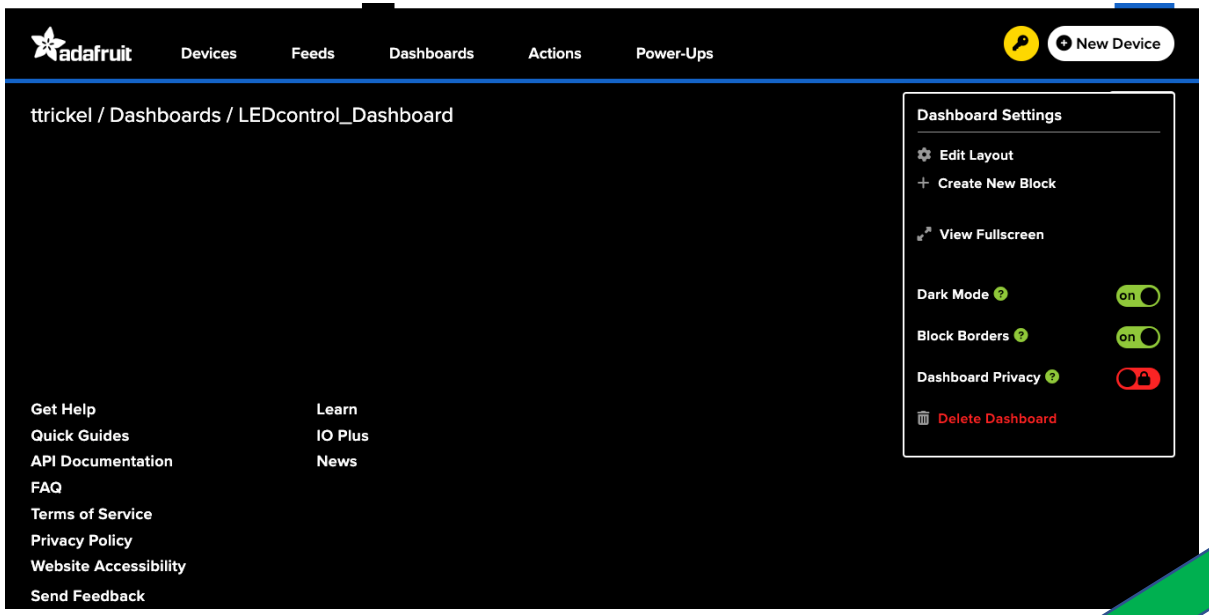


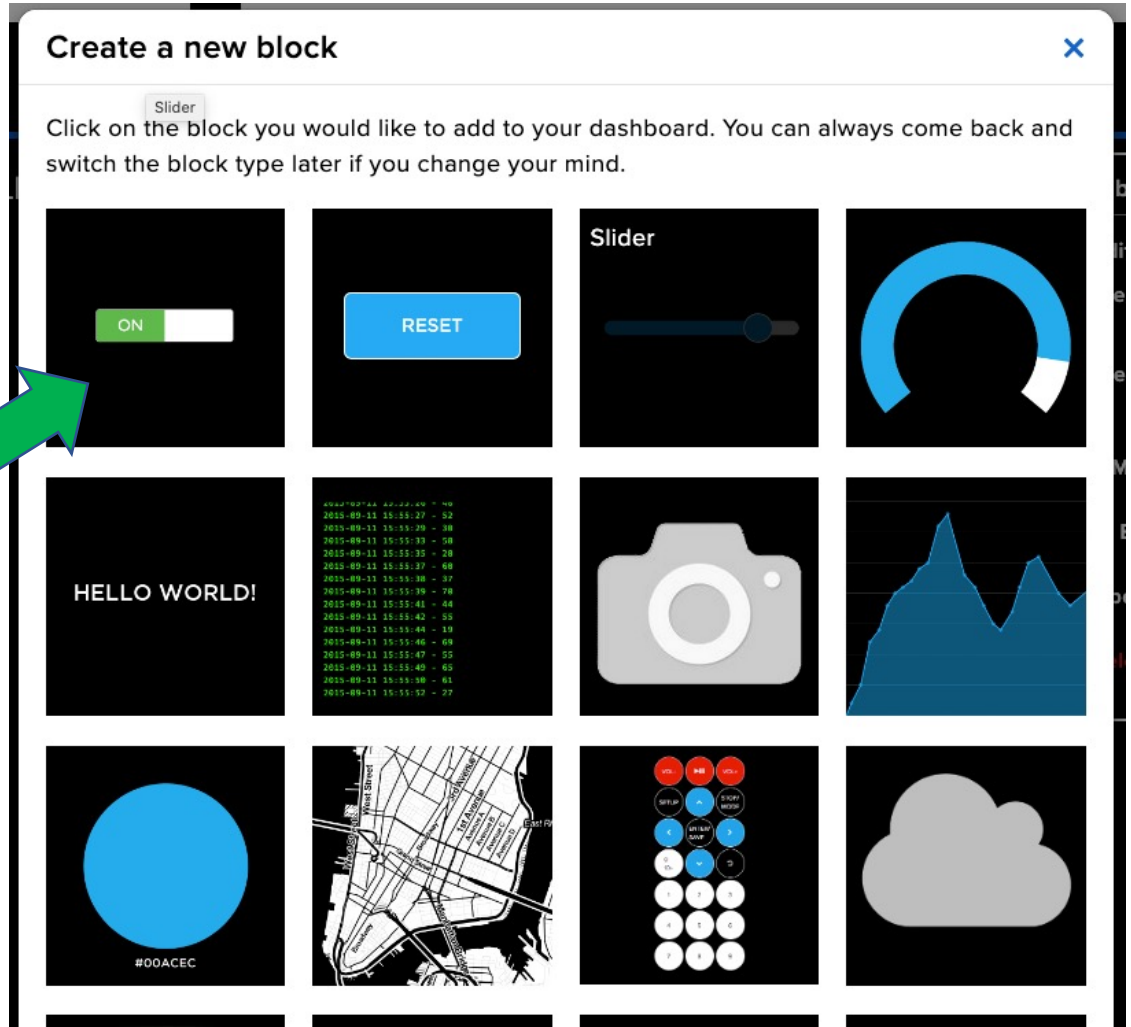
Create AIO Dashboard



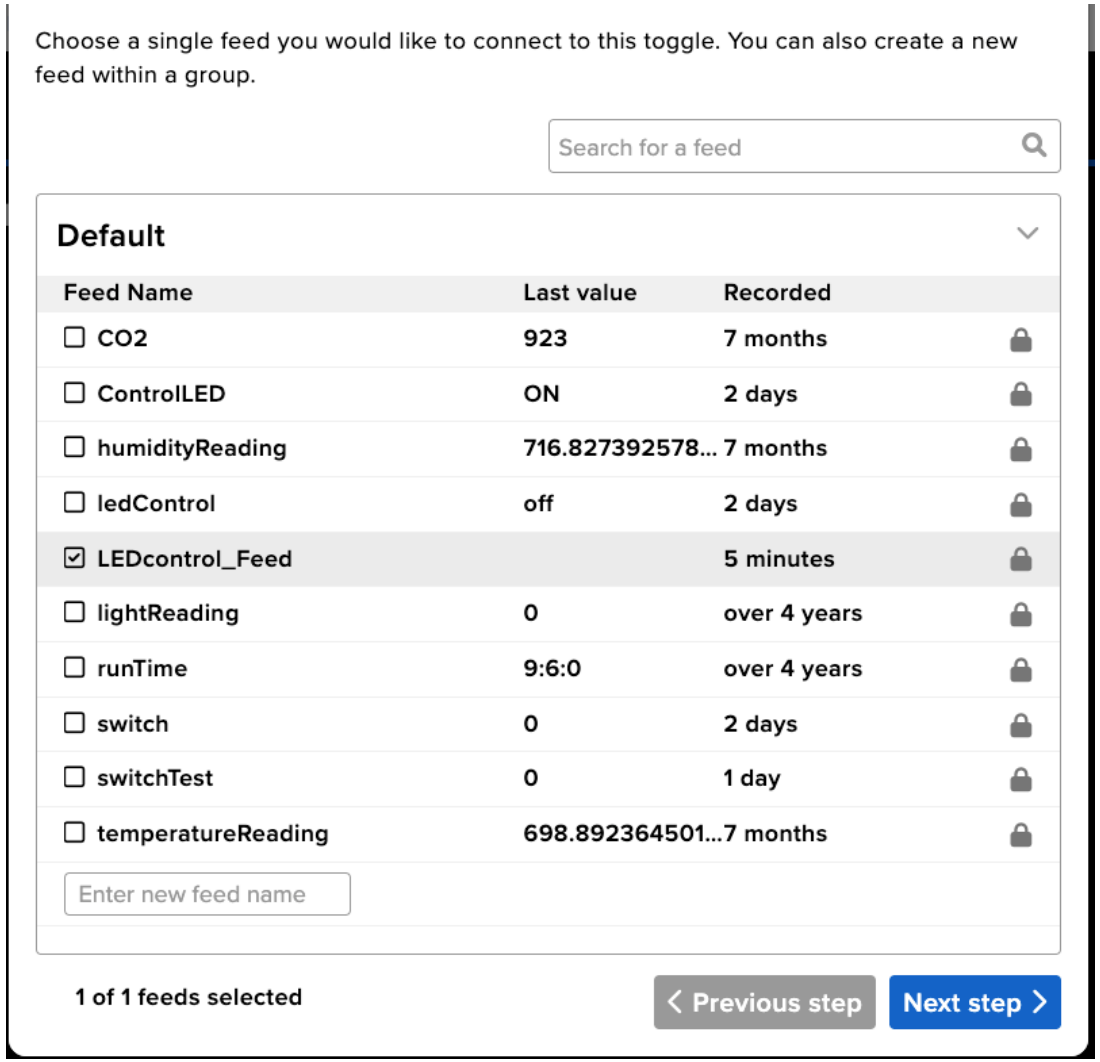


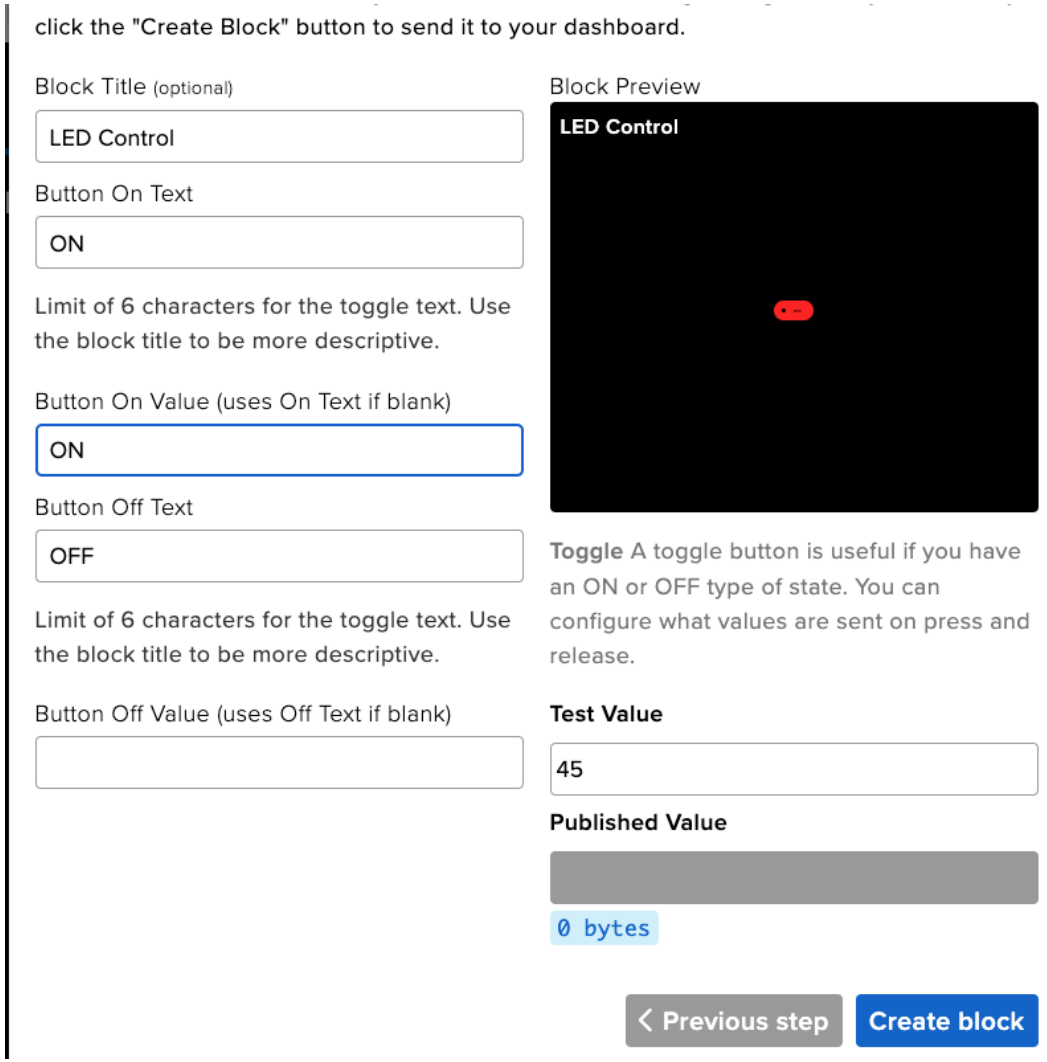
Create AIO Dashboard continued



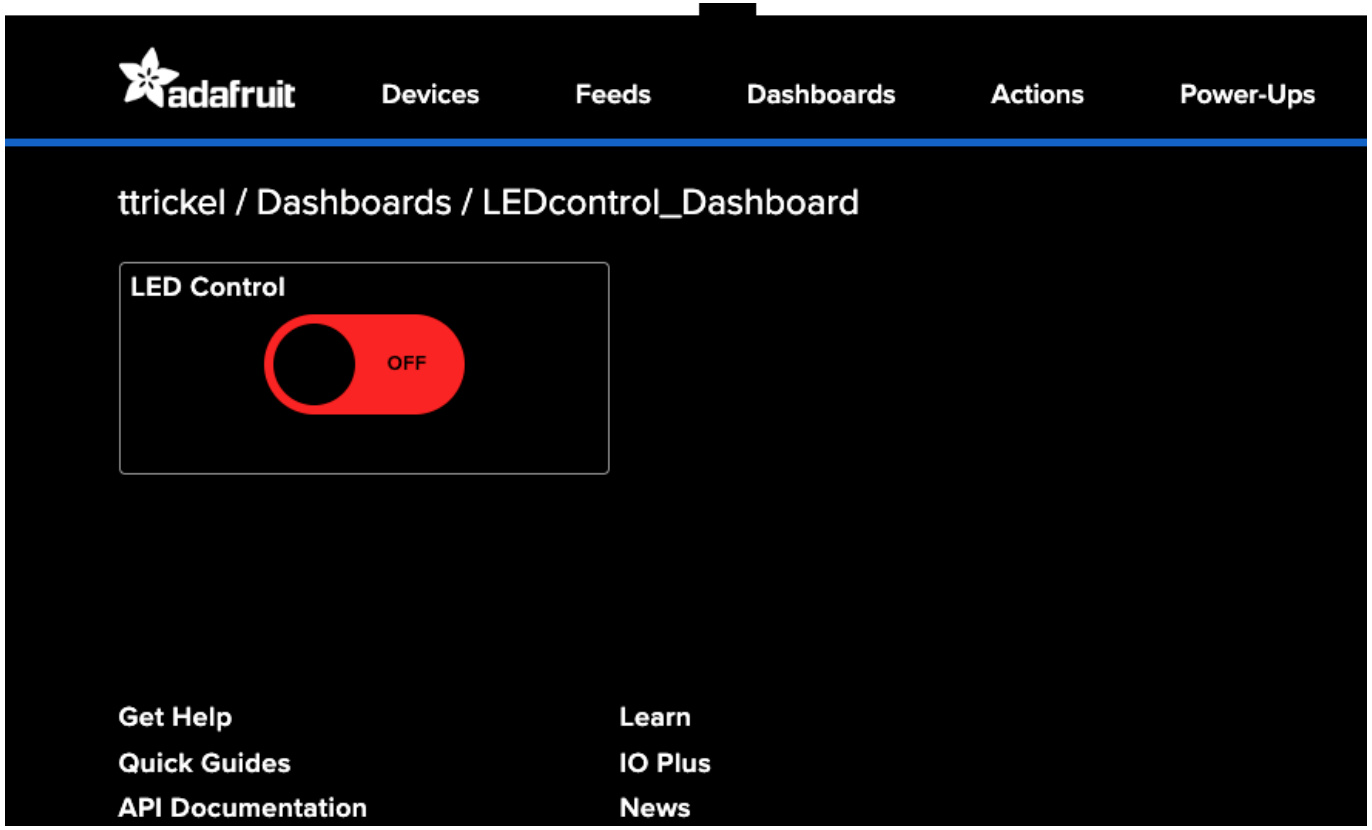


Create AIO Dashboard continued





Create AIO Dashboard continued



Control LED from AIO Python Code

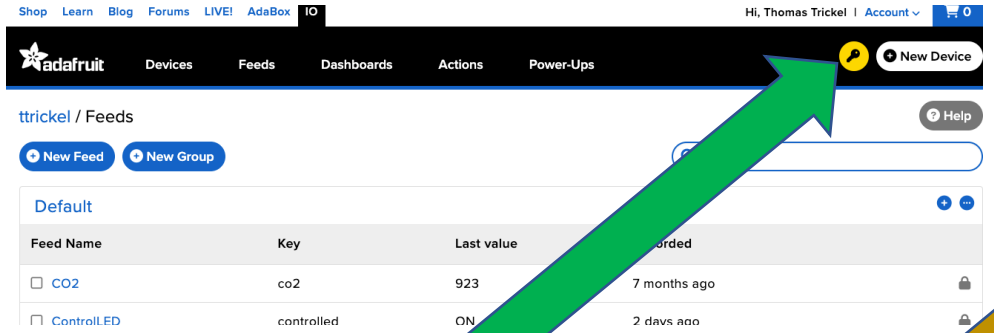
change

YOUR_AIO_KEY

YOUR_AIO_USERNAME

```
"""
'digital_out.py'
==================================
Example of turning on and off a LED
from the Adafruit IO Python Client
Author(s): Brent Rubell, Todd Treece
           with modifications by Thomas Trickel
"""
# Import standard python modules
import time
import RPi.GPIO as GPIO

# import Adafruit IO REST client.
from Adafruit_IO import Client, Feed, RequestError

# setup GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)


# Set to your Adafruit IO key.
# Remember, your key is a secret,
# so make sure not to publish it when you publish this code!
ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'

# Set to your Adafruit IO username.
# (go to https://accounts.adafruit.com to find your username)
ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME"

# Create an instance of the REST client.
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

try: # if we have a 'digital' feed
    digital = aio.feeds('ledcontrol-feed')
except RequestError: # create a digital feed
    feed = Feed(name="ledcontrol-feed")
    digital = aio.create_feed(feed)

while True:
    data = aio.receive(digital.key)
    if data.value == "ON":
        print('received <- ON\n')
        GPIO.output(18, True)
    elif data.value == "OFF":
        print('received <- OFF\n')
        GPIO.output(18, False)

# timeout so we dont flood adafruit-io with requests
    time.sleep(0.5)

```

AIO KEY & USERNAME



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       "a0da523dfa16502494a70e61e1f683ba8
a0b32f4"
```

Linux Shell

```
export IO_USERNAME="ttrickel"
export IO_KEY="a0da523dfa16502494a70e61e1f683ba8a0b32f
4"
```

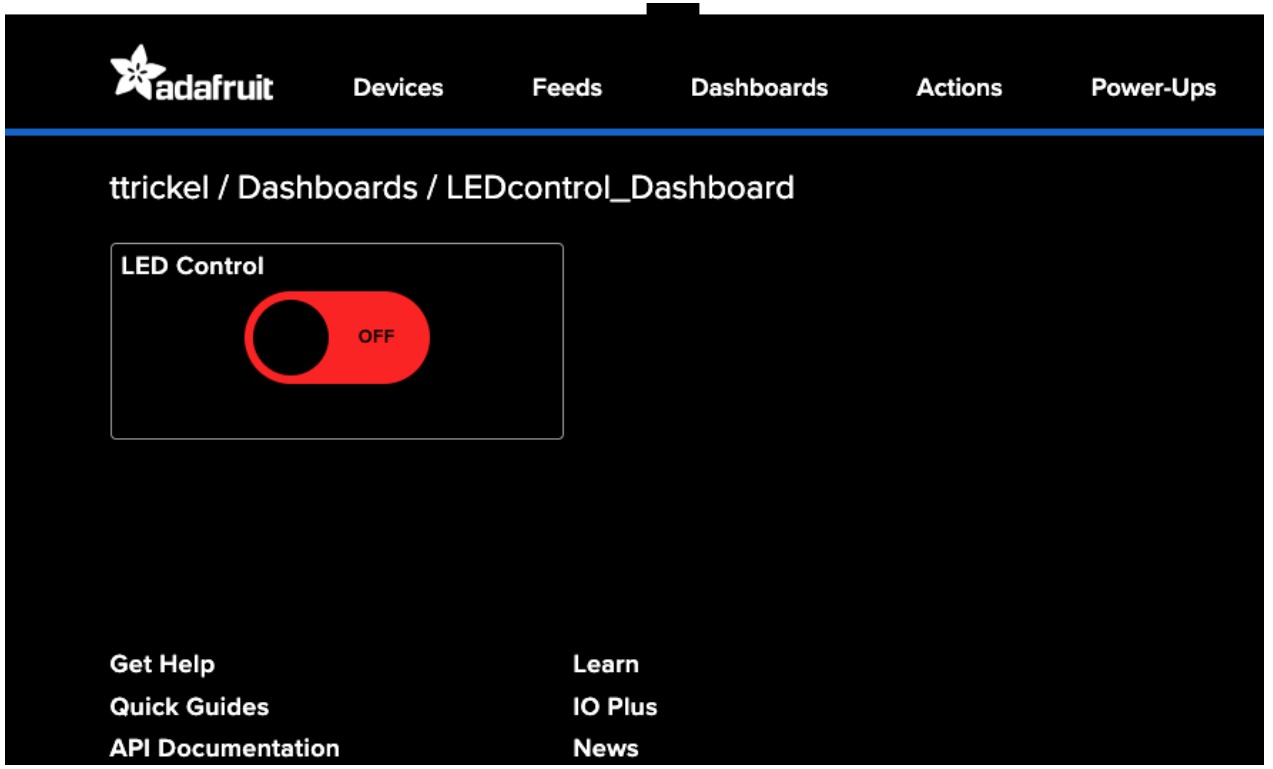
Scripting

```
ADAFRUIT_IO_USERNAME = "ttrickel"
ADAFRUIT_IO_KEY = "a0da523dfa16502494a70e61e1f683ba8a0b
32f4"
```

Test It

Run Python Code

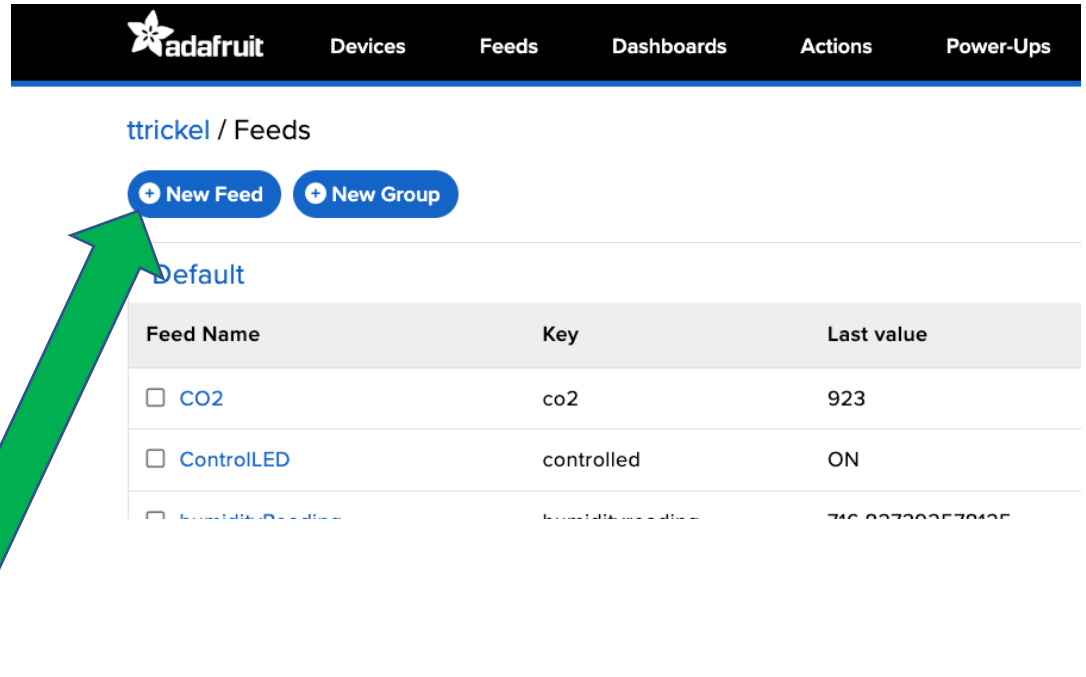
Click on LED Control icon

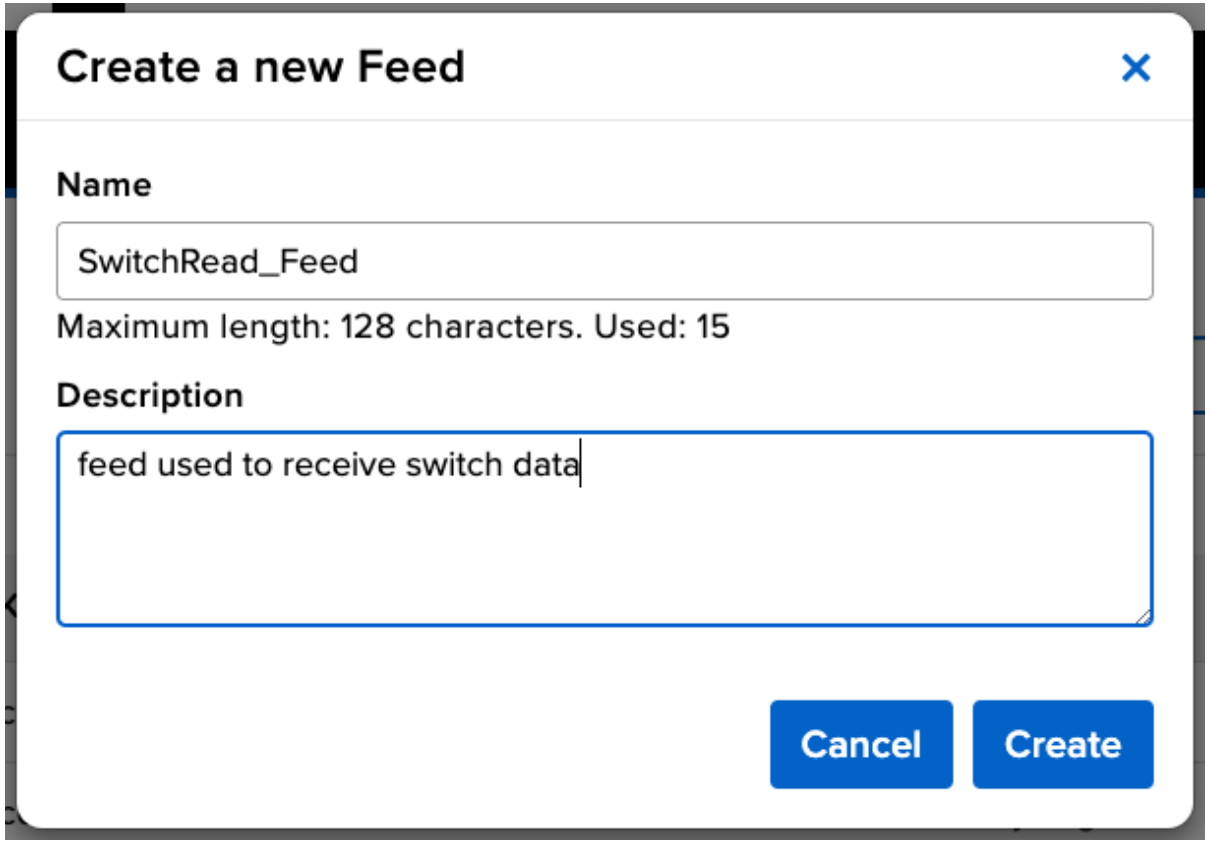


Send Switch Data to AIO

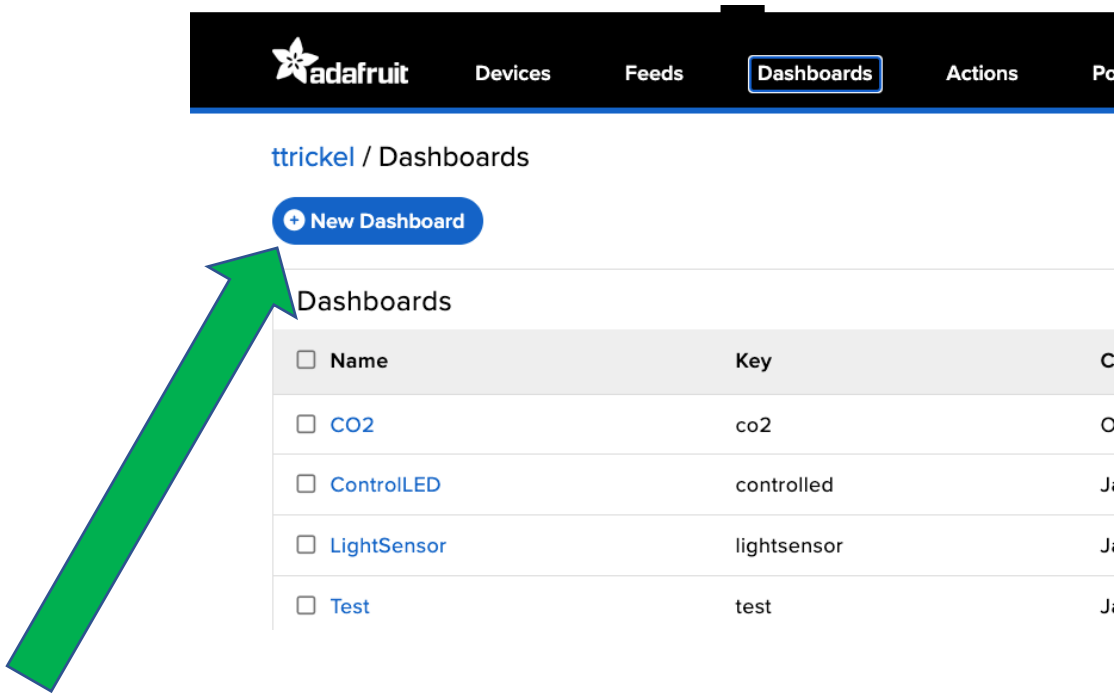
- Hardware is the same
- Create Feed
- Create Dashboard
- Write Python Code

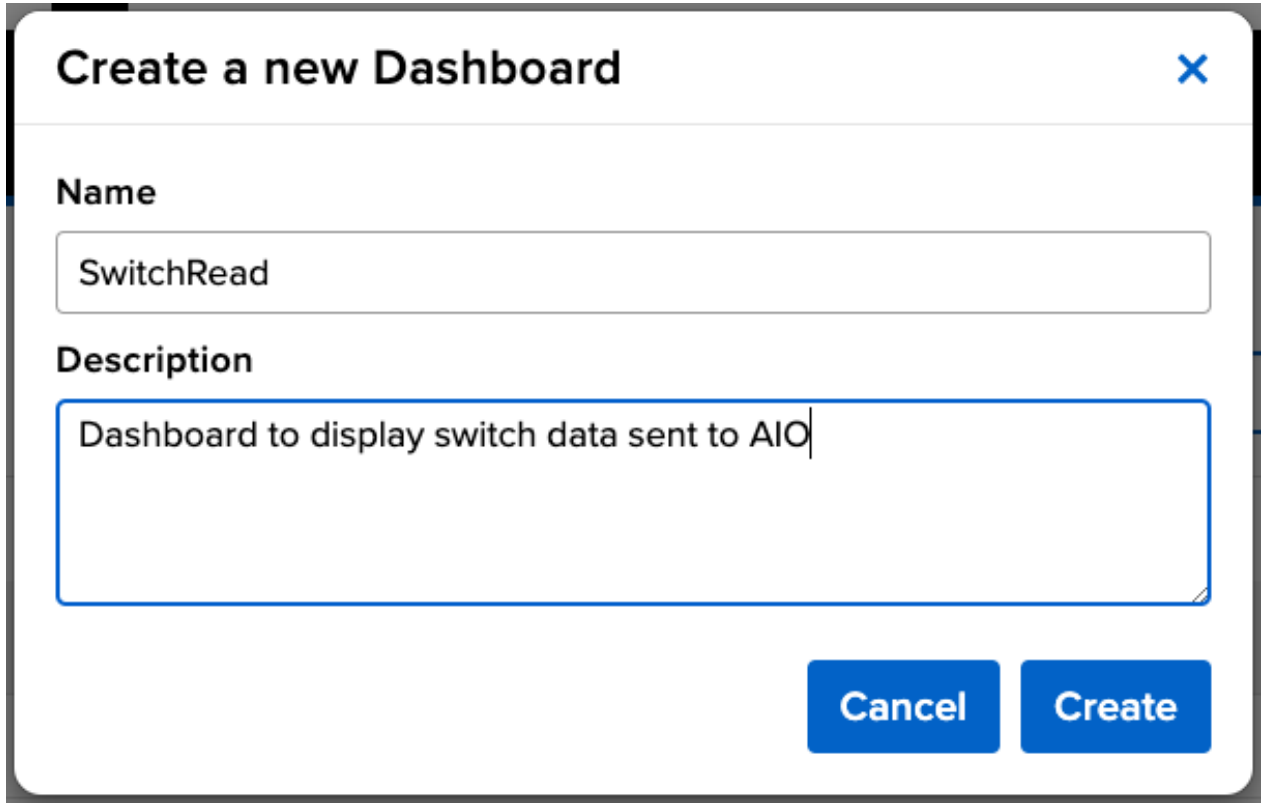
Create AIO Feed



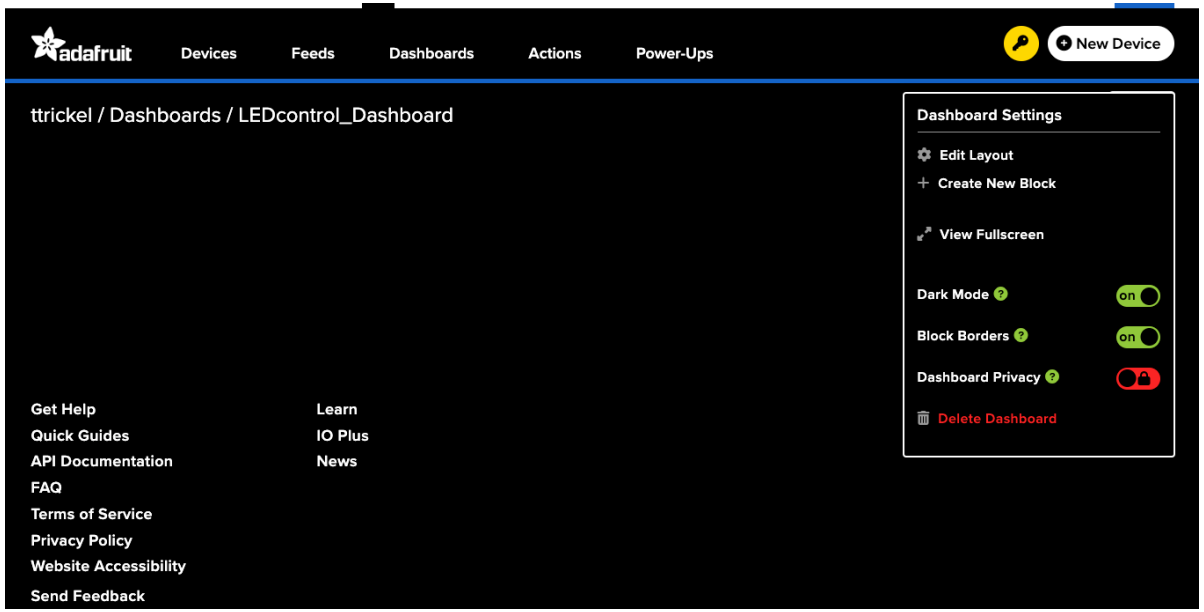


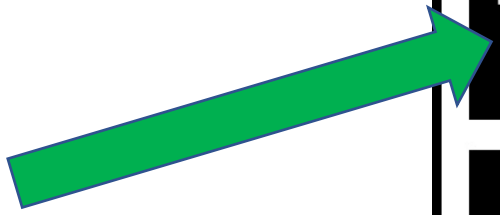
Create AIO Dashboard

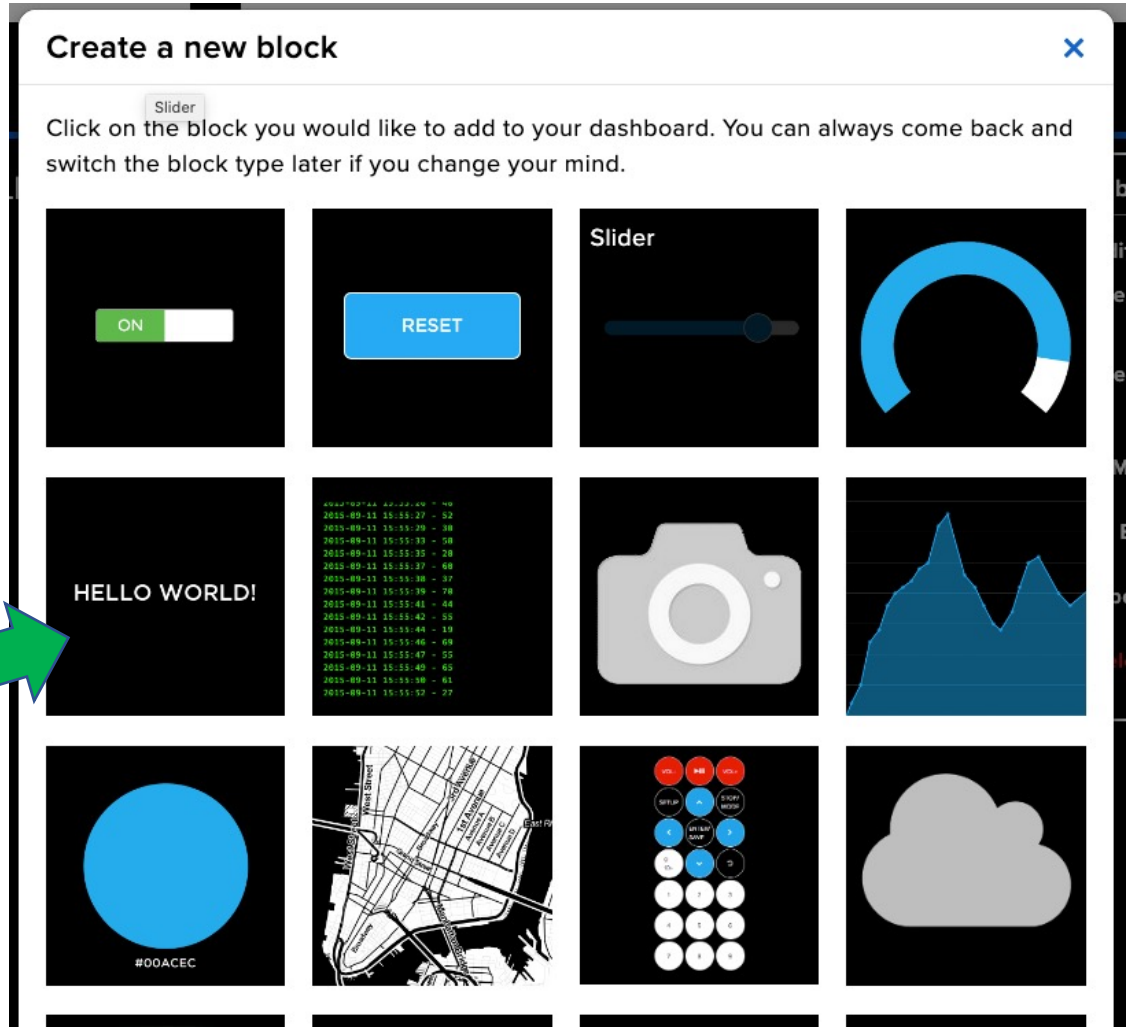




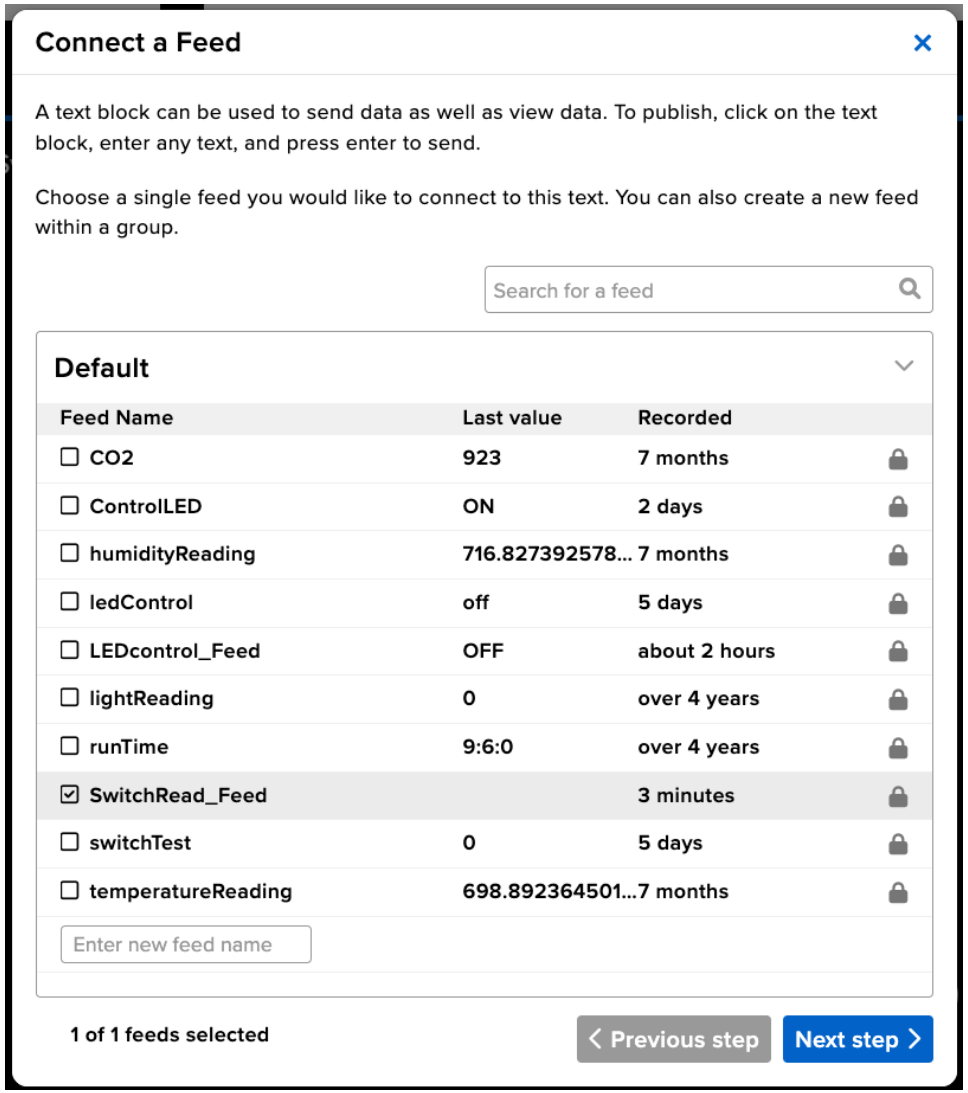
Create AIO Dashboard continued

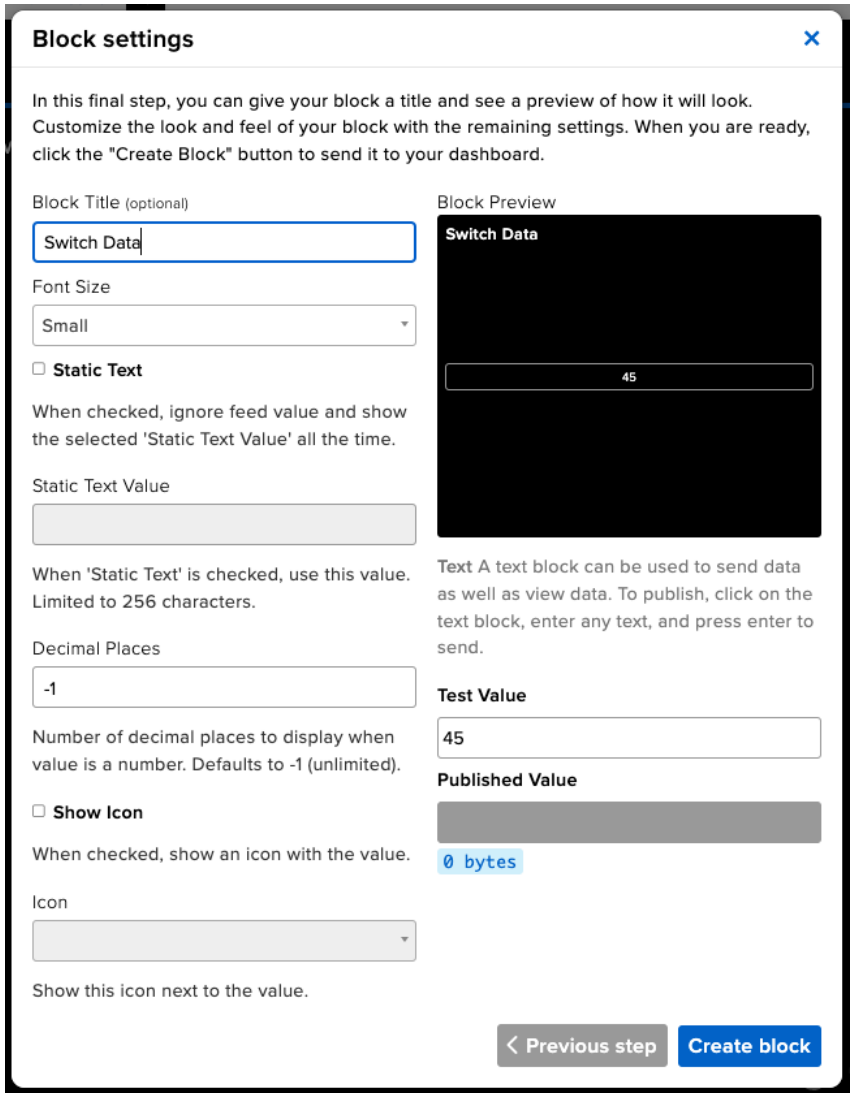




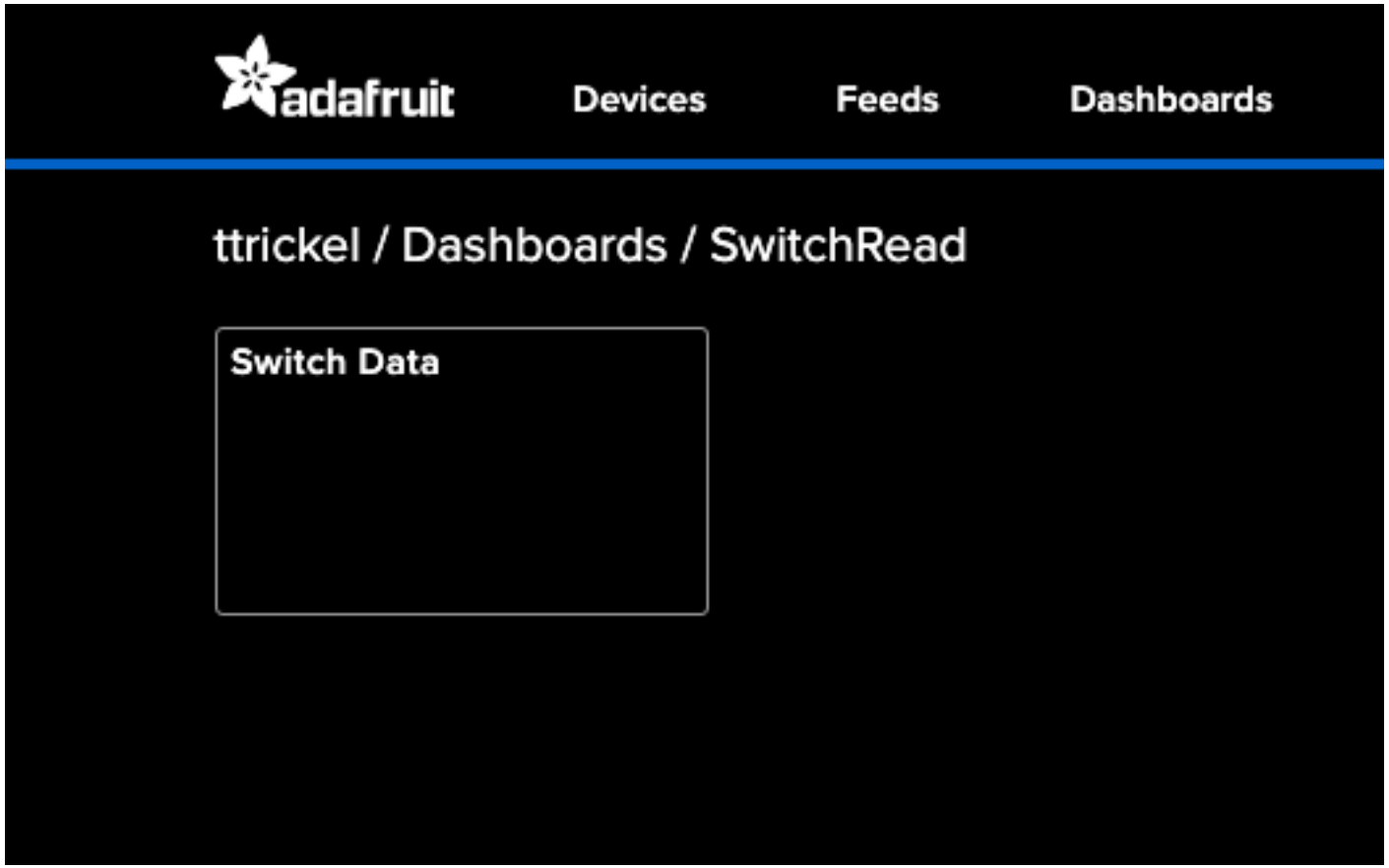


Create AIO Dashboard continued





Create AIO Dashboard continued



Send Switch Data to AIO Python Code

change

YOUR_AIO_KEY

YOUR_AIO_USERNAME

```
"""
'digitalIn.py'
==================================
Example of sending the status of a
switch to the Adafruit IO Python Client
"""
# Import standard python modules
import time
import RPi.GPIO as GPIO

# import Adafruit IO REST client.
from Adafruit_IO import Client, Feed, RequestError

# setup GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.IN)

# Set to your Adafruit IO key.
# Remember, your key is a secret,
# so make sure not to publish it when you publish this code!
ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'

# Set to your Adafruit IO username.
# (go to https://accounts.adafruit.com to find your username)
ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'

# Create an instance of the REST client.
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

try: # if we have a 'digital' feed
    digital = aio.feeds('switchread-feed')
except RequestError: # create a digital feed
    feed = Feed(name="switchread-feed")
    digital = aio.create_feed(feed)

while True:
    if GPIO.input(25):
        aio.send(digital.key, 0)
    else:
        aio.send(digital.key, 1)

# timeout so we dont flood adafruit-io with requests
    time.sleep(0.5)

```

Test It

Run Python Code

Press Switch on Breadboard