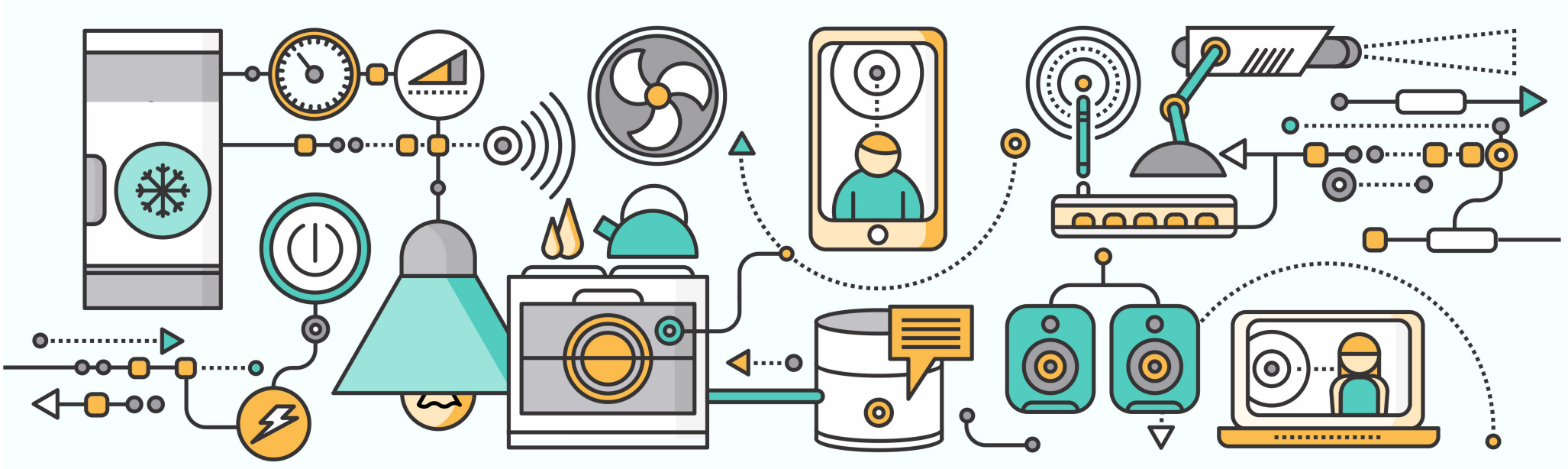
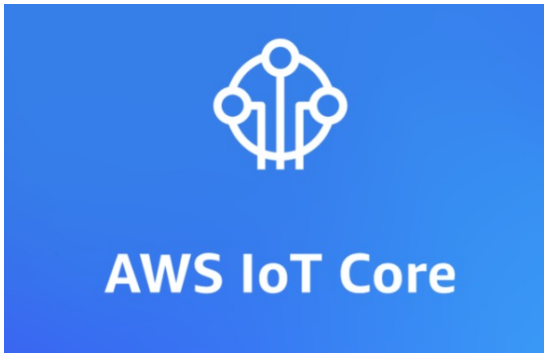
ENME 441
Mechatronics and the Internet of Things



IoT Platforms / Adafruit IO


AWS IoT Core


openremote


tuya


ThingWorx


ThingSpeak


B
Blynk


Ayla


deviceWISE


IFTTT


adafruit.io


Microsoft Azure
IoT Platform


PARTICLE


Telegraf
ctrlX
AUTOMATION

Message Queuing Telemetry Transport (MQTT)

MQTT is a communication protocol designed for IoT that uses TCP connections (same as sockets / Websockets), but with several advantages over sockets:

- Low overhead for transmitted packets
- Last known good data value for a device can be stored
- Notifications when client unexpectedly disconnects
- Publish/subscribe routing for one-to-many message distribution
- Secure connections are easy to make

Core MQTT commands:

- CONNECT
- SUBSCRIBE
- PUBLISH
- UNSUBSCRIBE
- DISCONNECT

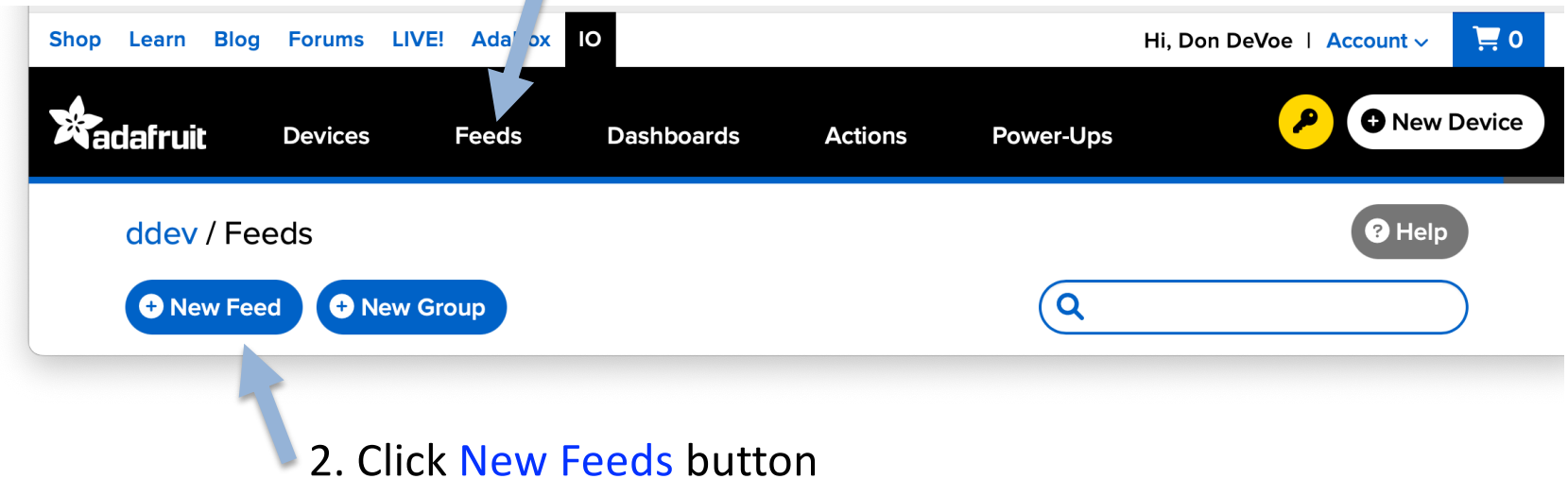
Adafruit IO Overview

Go to https://io.adafruit.com and sign up for a free account

- Terminology:
  - Dashboard = Adafruit IO user interface
  - Feed = single data stream (e.g. sensor values over time from an IoT device)
- Steps for creating a user interface:
  - Create a feed for each data stream to be sent from the ESP32 and displayed through the UI
  - Create a feed for each data stream to be send from the UI and received by the ESP32
  - Create a dashboard
  - Add UI elements to the dashboard, with each element associated with an appropriate feed

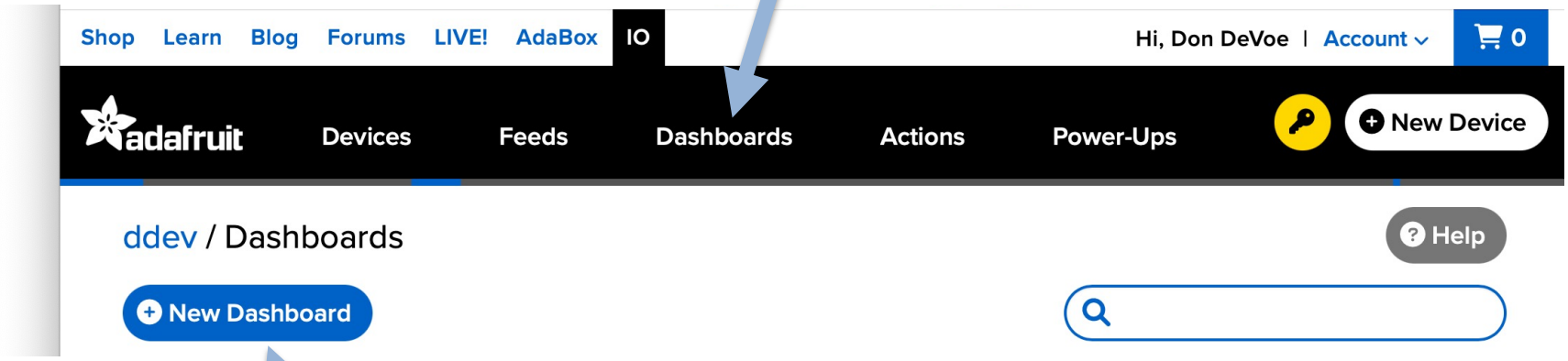
Creating a Feed & Dashboard

1. Click Feeds link



2. Click New Feeds button

3. Click Dashboards link



2. Click New Dashboard button

Lab 9: Motion Tracker

Save the mpu6886.py module to your ESP32

Goals:

- Display instantaneous acceleration of a single IMU axis
- Plot acceleration vs. time
- Flash the Matrix screen at each measurement time point
- Switch measurement axis between X and Y axes when user button held down during measurement
- Display the currently-selected axis (X or Y)
- Switch LED flash color between green and red when a button is pressed on the Web UI

Step 1: Create Feeds

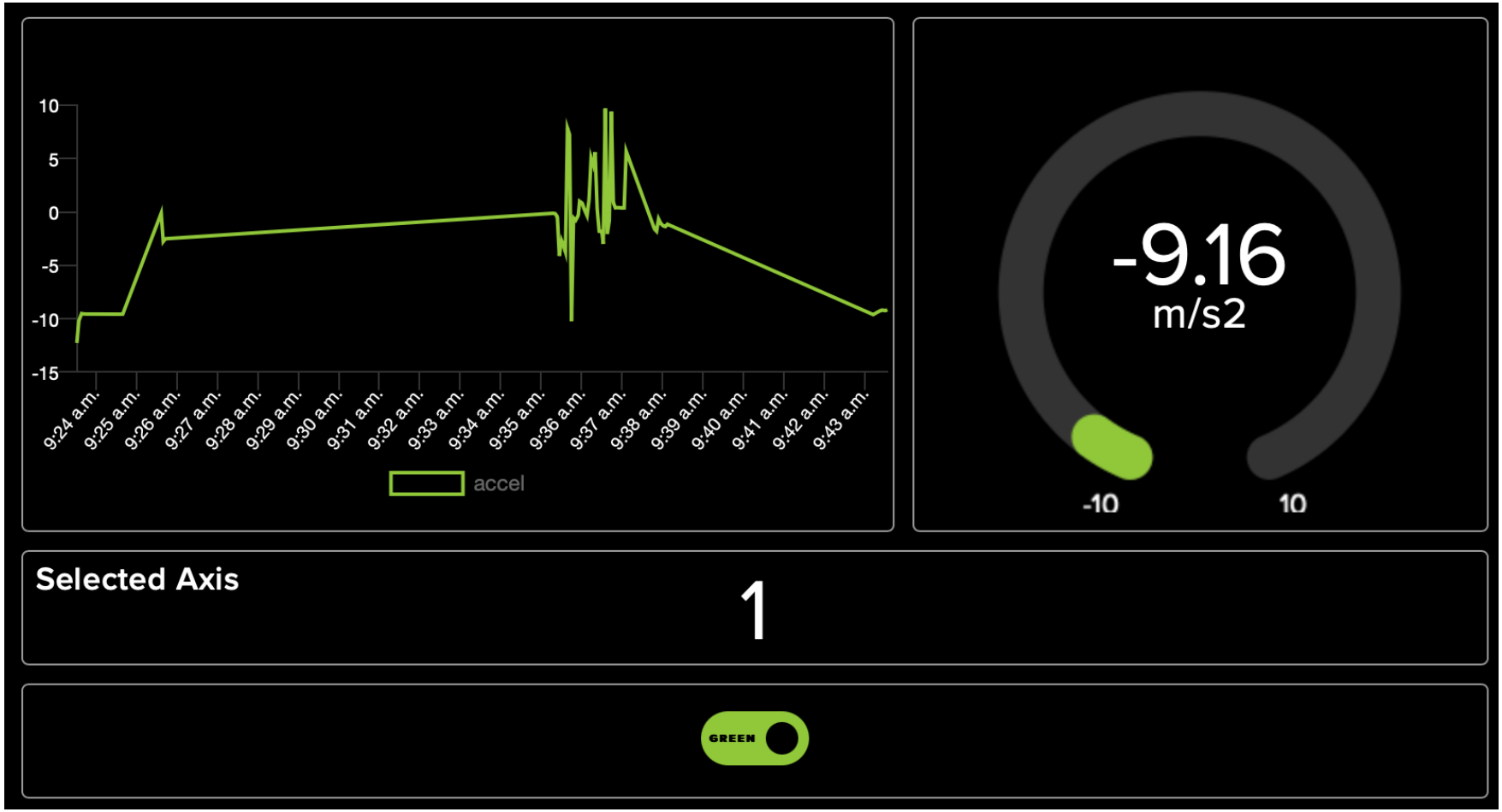
- From the problem statement, 3 feeds are required
- Create each feed in your Adafruit IO account, with the following names:
  - Acceleration measurement: accel
  - ESP32 button state: matrix_button
  - Web button state: color_button

Step 2: Design the Dashboard

1. Select UI elements:
- Instantaneous acceleration: gauge 
- Plot of acceleration vs. time: line chart 
- Button to select Matrix flash color: toggle 

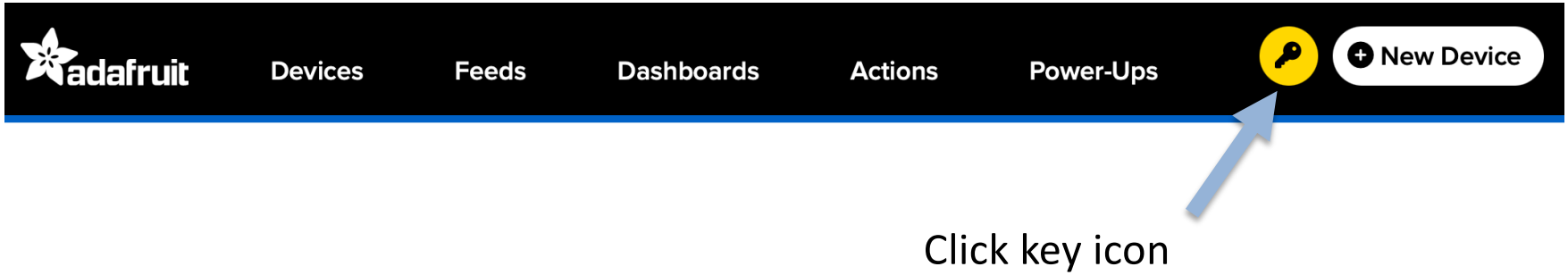
- Display of current acceleration axis: text 

2. Select a feed to connect to each UI element
3. Add UI element settings if needed
4. Click Create block


10
5
0
-5
-10
-15
9:24 a.m.
9:25 a.m.
9:26 a.m.
9:27 a.m.
9:28 a.m.
9:29 a.m.
9:30 a.m.
9:31 a.m.
9:32 a.m.
9:33 a.m.
9:34 a.m.
9:35 a.m.
9:36 a.m.
9:37 a.m.
9:38 a.m.
9:39 a.m.
9:40 a.m.
9:41 a.m.
9:42 a.m.
9:43 a.m.
accel
-9.16
m/s2
-10
10
Selected Axis
1
GREEN

Prepare for Code Development

- Get the Adafruit IO key for your account
  - A single key is used to update all feeds
  - Avoid sharing your key with others!
  - A new key can be generated if needed, but you will need to update all code after making the change



- You will also need your account user name and site URL (io.adafruit.com)

adafruit_io_imu.py