Lab 8
BLE on the Adafruit Feather Bluefruit Sense with an MQTT gateway to AdafruitIO

Part 1 - Prepare your AdafruitIO account

As a class you should have gone through the steps to create Adafruit IO accounts.

Create three new “feeds” in your AdafruitIO account. Create one called random_value. Create another called tx and a final feed called output.

You should also create a new dashboard - Lab 8. On the dashboard create two new blocks. The first block will be a Stream block to show the random_value feed. The second block will be a Line Chart. It will also visualize the random_value feed. The third block will be a Text block to show the current state of the output feed.

Show your instructor your feeds and your dashboard. ☐ ______

Part 2 - BLE connect to Bluefruit app

Using your code from Lab 7 as a starting point, program your Feather sense to send random integer values between 25 and 50 to the Bluefruit LE app using the UART connection. You should send a continuous stream of random data values with a delay of 10 seconds between each data point. Do not start sending the data until the Feather is connected to the Bluefruit LE app.

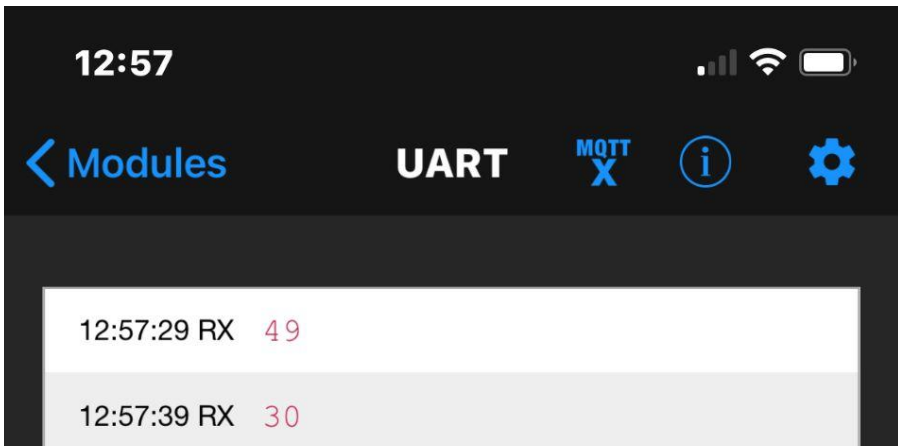
Show your instructor your streaming data. ☐ ______

Save your code as lastname_lab8_part2.py. Add header information and inline comments to your code.

Possible solution code lab8_part2.py

Part 3 - Configure the Bluefruit LE Connect MQTT gateway

Configure MQTT on the Bluefruit LE app. While connected to the Feather click on UART. You should see an MQTT option with an X below it at the top of the app screen.



```
12:57:29 RX   49
12:57:39 RX   30
```

Click on the MQTT option and the app will bring up a new screen.

The Address and the Port should be filled already.

With PUBLISH switched on,

Set `Uart RX:  your_AIO_ username/f/random_value`

Set `Uart TX:  your_AIO_username/f/tx`

With SUBSCRIBE switched on,

Set `Topic: your_AIO_username/f/output`

Set `Action: Transmit`

Scroll down to ADVANCED.

Set `Username: your_AIO_username`

Set `Pass/Key: your_AIO_KEY`

What have you just done? Here it is again with some descriptions…

With PUBLISH switched on, "publish, i.e. send data from the apps Uart RX+TX"

Set `Uart RX:  your_AIO_ username/f/random_value`

Publish Uart RX to the `your_AIO_ username/f/random_value`

Set `Uart TX:  your_AIO_username/f/tx`

Publish Uart TX to the `your_AIO_ username/f/tx`

Anything that the Bluefruit LE apps Uart receives or transmits is published to the feeds.

With SUBSCRIBE switched on, "the app subscribes to the chosen topic"

Set `Topic: your_AIO_username/f/output`

The App subscribes to the `your_AIO_username/f/output` topic and will receive new values sent from the AIO MQTT broker.

Set `Action: Transmit`

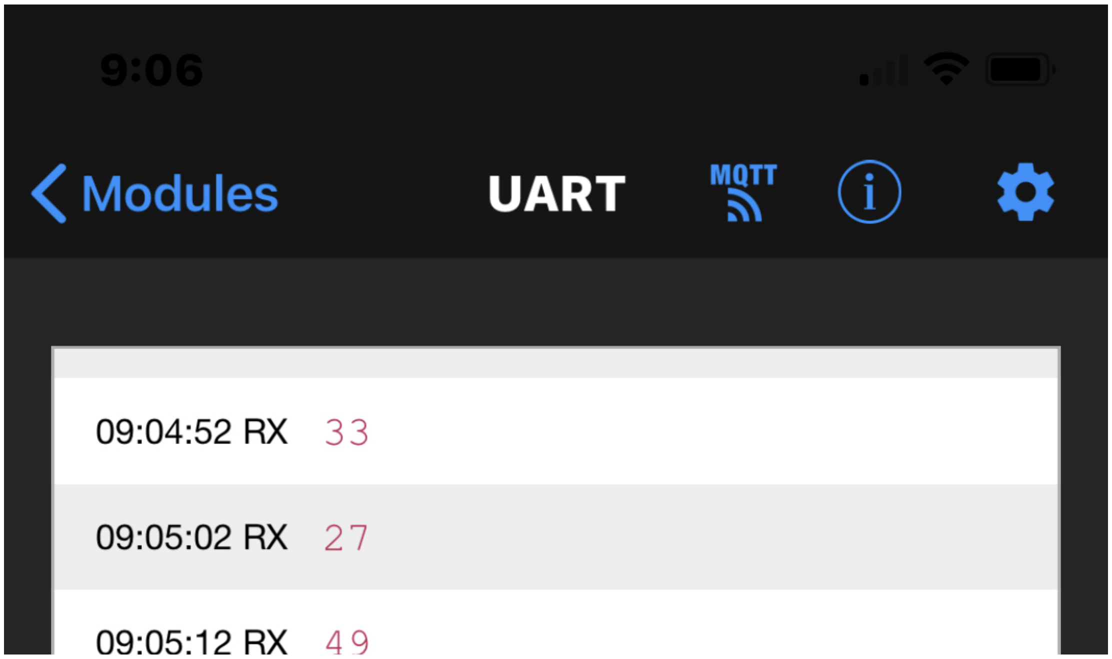
The App is set up to transmit the values it receives from the above topic subscription. The value will be transmitted by the Bluefruit LE app over BLE to the Feather for its use as well.

Scroll down to ADVANCED.

Set `Username: your_AIO_username`

Set `Pass/Key: your_AIO_KEY`

Scroll back to the top and select connect… The Bluefruit LE app should connect to the AIO broker. Sometimes you need to exit the application and start it up again.



```
09:04:52 RX   33
09:05:02 RX   27
09:05:12 RX   49
```

If everything is configured correctly, the Bluefruit app should change to show MQTT is transmitting and your dashboard on AdafruitIO should start showing updated values from the stream of data coming from the Feather.

Show your instructor your working AIO dashboard.

☐ ________

Part 4 - Add a Trigger on AdafruitIO and pause the data stream

Have AdafruitIO monitor the random_values feed for values above a setpoint. You should be streaming data between 25 and 50, so pick a setpoint of 45 or above. We want to have AIO monitor the feed and then respond back with the "output" feed. (Remember , we had the Bluefruit app subscribe to that feed.)

AdafruitIO allows you to set up an Action to do just this. You should set up a new Action, a Reactive Trigger that will update the output feed to the command PAUSE every time the random_value feed is greater than your setpoint.

With the Bluefruit app subscribed to the output feed, the app will receive the command and then Transmit it over BLE to the Feather.

You need to modify your Feather code to check for any UART input. Read the UART input if it exists and then act on the command. The following code will check for data on the uart stream and read it into a string called `command`.

```
if uart.in_waiting:
     command = uart.readline().decode("utf-8")
```

Use the command to halt the data stream for one minute. While the stream is paused, blink the RED_LED on the Feather Sense. After the required minute pause, continue with the previous program, streaming random data every 10 seconds.

Show your instructor your working Feather.

☐ ______

Save your code as lastname_lab8_part4.py. Modify header information and inline comments of your code.

Possible solution code lab8_part4.py

Part 5 - Add some new "commands" to the Feather/AIO

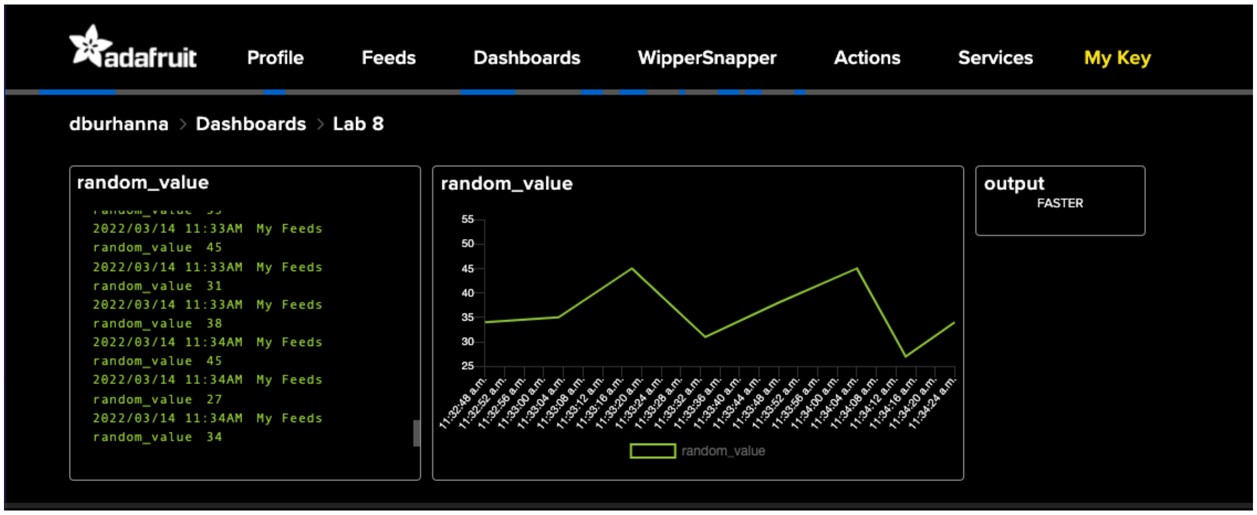
Can you write another "command" for the Feather? The PAUSE command was simply the string PAUSE placed on the "output" feed in AIO.   Create two new commands that change the delay in the data stream. With the following possible delay constraints:

min_delay = 5
max_delay = 30

Create the command, FASTER, which will reduce the current delay by five seconds.
Create the command, SLOWER, which will increase the current delay by five seconds.

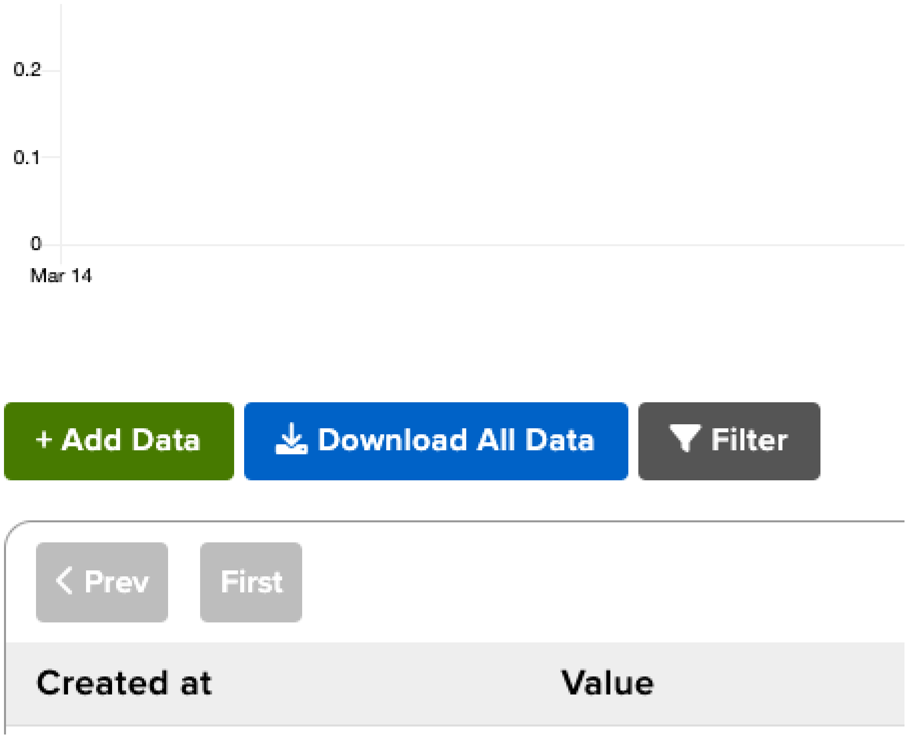
You can test these commands by manually entering data to the output feed on AIO.

First, select Feeds on the AIO window menu bar. This will drop down your list of feeds. Select the output feed. See image on next page.



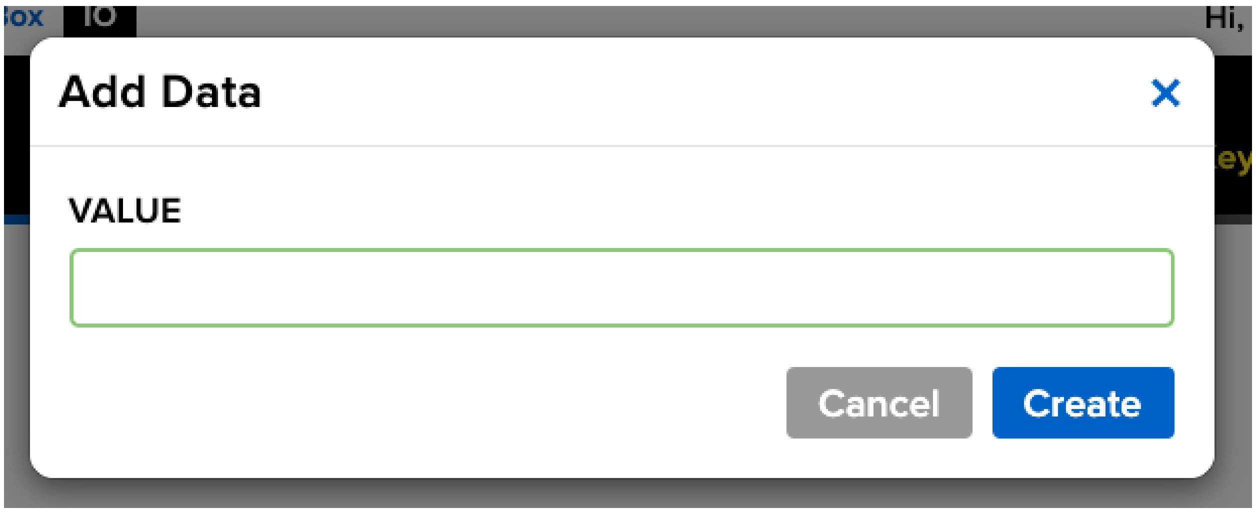
```
2022/03/14 11:33AM  My Feeds
random_value  45
2022/03/14 11:33AM  My Feeds
random_value  31
2022/03/14 11:33AM  My Feeds
random_value  38
2022/03/14 11:34AM  My Feeds
random_value  45
2022/03/14 11:34AM  My Feeds
random_value  27
2022/03/14 11:34AM  My Feeds
random_value  34
```

After you have selected the output feed, locate the green "+ Add Data" button in the middle left of the screen.



Created at	Value

After clicking the Add Data button, a pop up will appear where you can manually enter a value to the output feed. Try sending “FASTER” and “SLOWER” to the Feather Sense.



Show your instructor your working Feather/AIO commands. ☐ ________

Save your code as lastname_lab8_part5.py. Modify header information and inline comments of your code.

Possible solution code lab8_part5.py

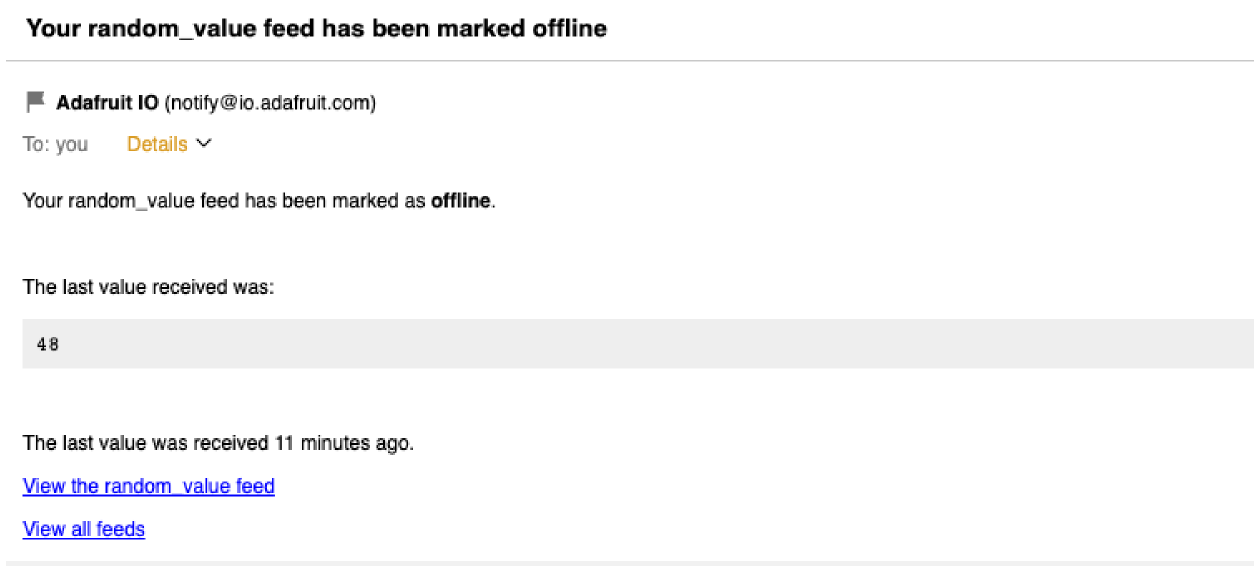
Part 6 - Explore some AIO feed options

In Part 6 you will explore some other features of the AIO service. Select the random_value feed and look at the right side of the screen. You will see a list of feed options: Feed Info , Privacy, Sharing, Notifications, Webhooks, Disable Feed, and License are the current options.

You can click on the gear to the upper right of each of the options to change the current settings. Most of these options are fairly obvious to use.

Click on Notifications settings and create a new active notification for the random_value feed. Have a “notification” sent to your email if the feed is offline. The default time is three days but you can set this notification to as little as 10 minutes to check to see how it works. Turn your Feather off and you should receive a notification email after the selected time has passed.

You should receive a notification email similar to the following:



Your random_value feed has been marked offline

Adafruit IO (notify@io.adafruit.com)

To: you   Details

Your random_value feed has been marked as **offline**.

The last value received was:

```
48
```

The last value was received 11 minutes ago.

View the random_value feed

View all feeds

A notification like this may be important if you are using an IOT device to monitor a mission critical device in the field.

IDEAS to talk about -

MQTT
Data aggregation
Data visualization
Actions/Triggers based on feed data
Feed notifications

This lab would be a great time to allow students to plug their Feathers into battery power and go truly wireless. They could test the range of the ble connectivity between the desktop and Feather or between the Feather and the Bluefruit app.

Project idea
Use an old cellphone with Wifi connection to connect to AIO? Old phones are plentiful. Connect Feathers via BLE to phone and use Bluefruit App to get to the AIO MQTT broker.