

# Maker Lab

---

## Adafruit IOT, Hats and getting data

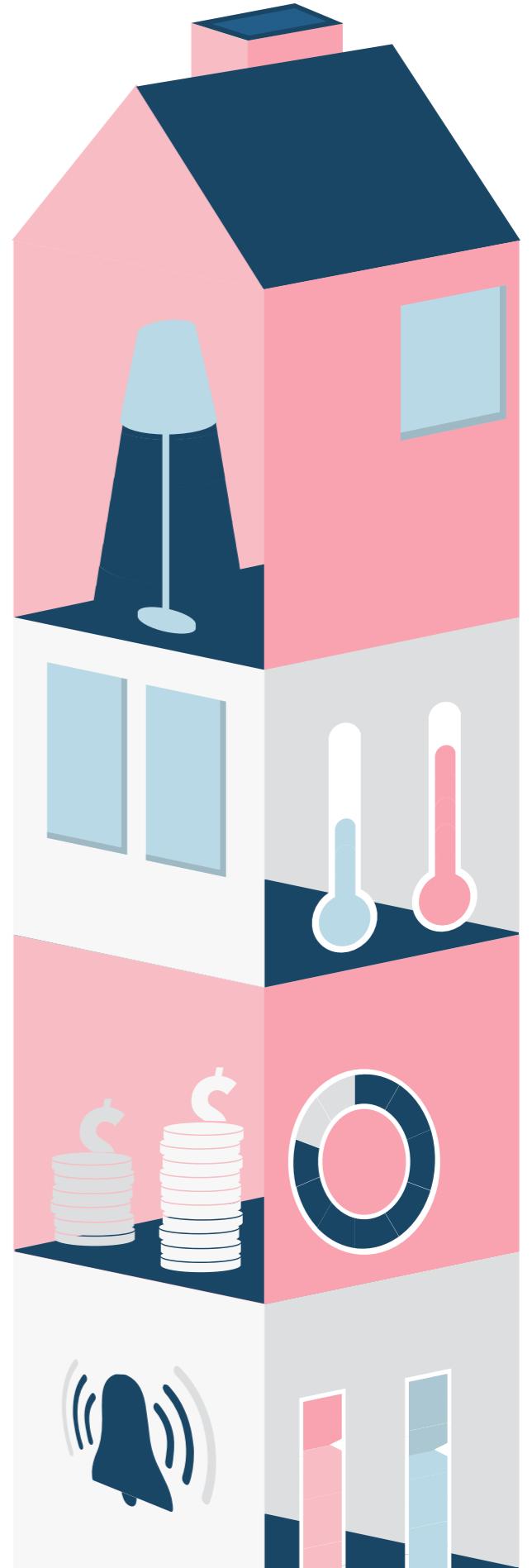
---

Rosella Gennari

Faculty of Computer Science

Free U. of Bozen-Bolzano

[www.inf.unibz.it/~gennari/makerlab.html](http://www.inf.unibz.it/~gennari/makerlab.html)



---

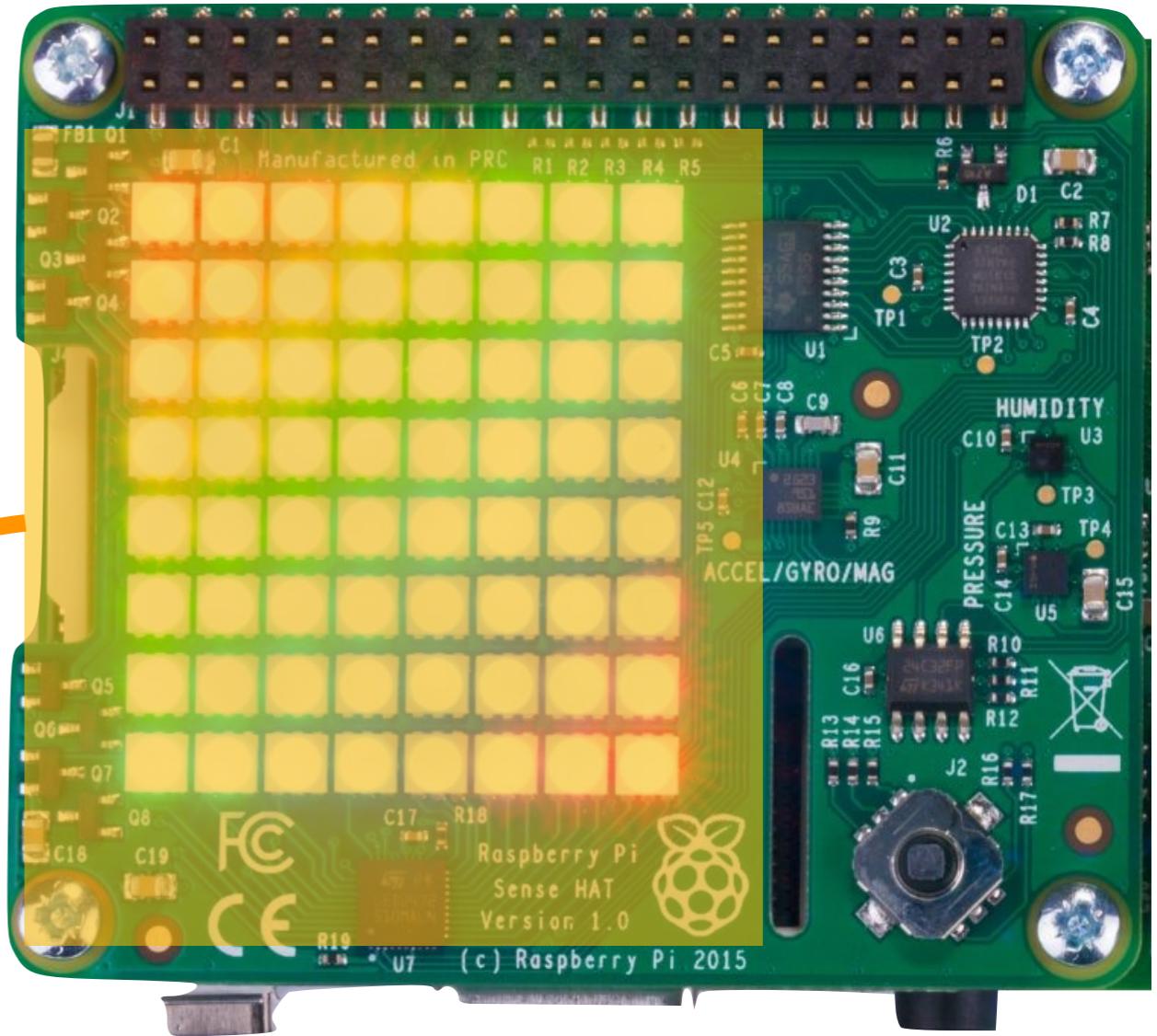
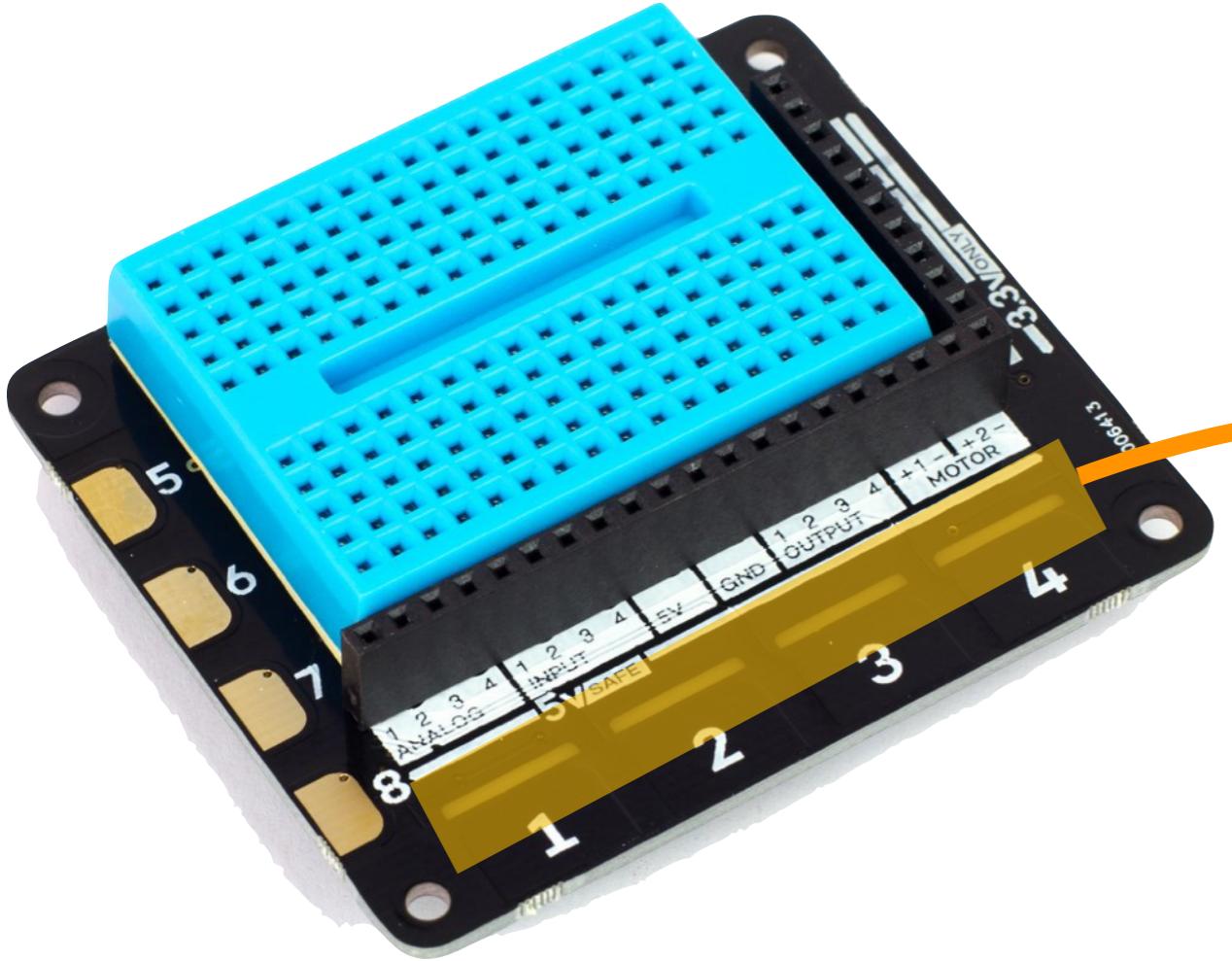
# Outline

---

Explorer Hat and Sense Emu "simulation": scrolling colours

Explorer Hat gets data from Adafruit IOT feeds and dashboards

Assignments: from Explorer Hat with Adafruit IOT to Sense Emu with Adafruit IOT



```

from sense_emu import SenseHat
from time import sleep

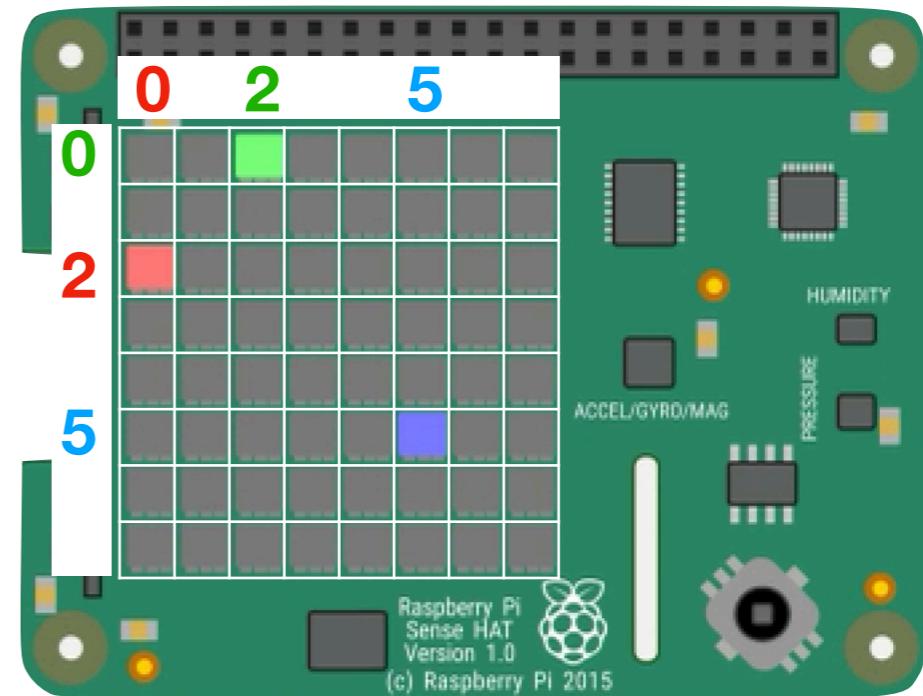
sense = SenseHat()

red = (255,0,0)
green = (0,255,0)
blue = (0,0,255)

sense.set_pixel(0,2,red)
sense.set_pixel(2,0,green)
sense.set_pixel(5,5,blue)

sleep(2)
sense.clear()

```



# LED MATRIX REMINDER: SET\_PIXEL VS SET\_PIXELS

See above.

```

from sense_emu import SenseHat
from time import sleep

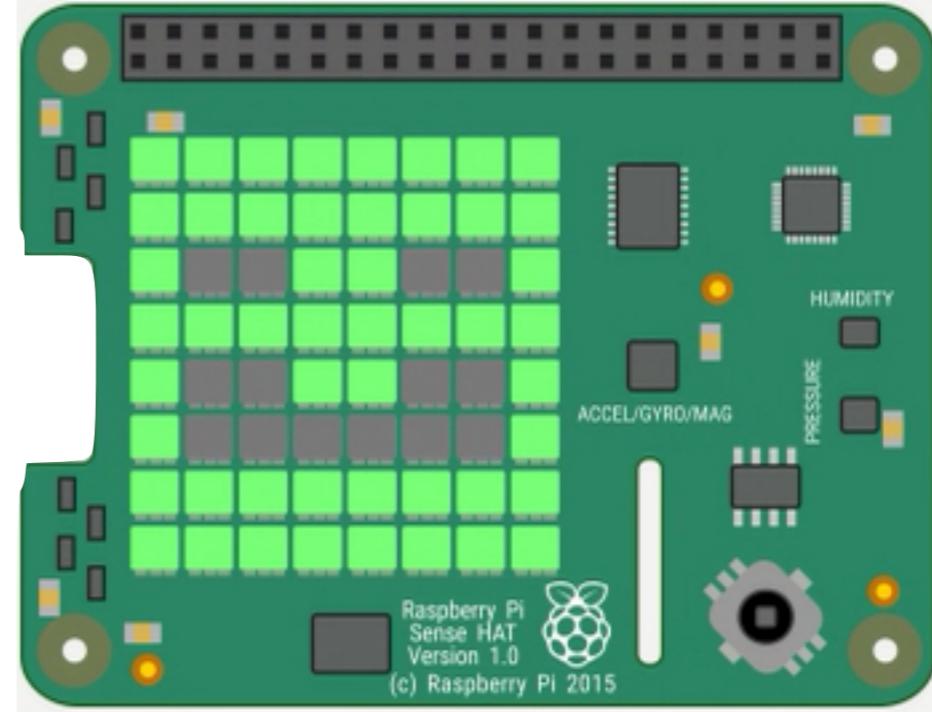
sense = SenseHat()

red = (255,0,0)
green = (0,255,0)
black = (0,0,0)

pattern = [
    #0,      #1,      #2,      #3,      #4,      #5,      #6,      #7
    green,   green,   green,   green,   green,   green,   green,   green,   #0
    green,   green,   green,   green,   green,   green,   green,   green,   #1
    green,   black,   black,   green,   green,   black,   black,   green,   #2
    green,   green,   green,   green,   green,   green,   green,   green,   #3
    green,   black,   black,   green,   green,   black,   black,   green,   #4
    green,   black,   black,   black,   black,   black,   black,   green,   #5
    green,   green,   green,   green,   green,   green,   green,   green,   #6
    green,   green,   green,   green,   green,   green,   green,   green,   #7
]
sense.set_pixels(pattern)

sleep(2)
sense.clear()

```



# LED MATRIX REMINDER: SET\_PIXEL VS SET\_PIXELS

See above.

```

from sense_emu import SenseHat
from time import sleep

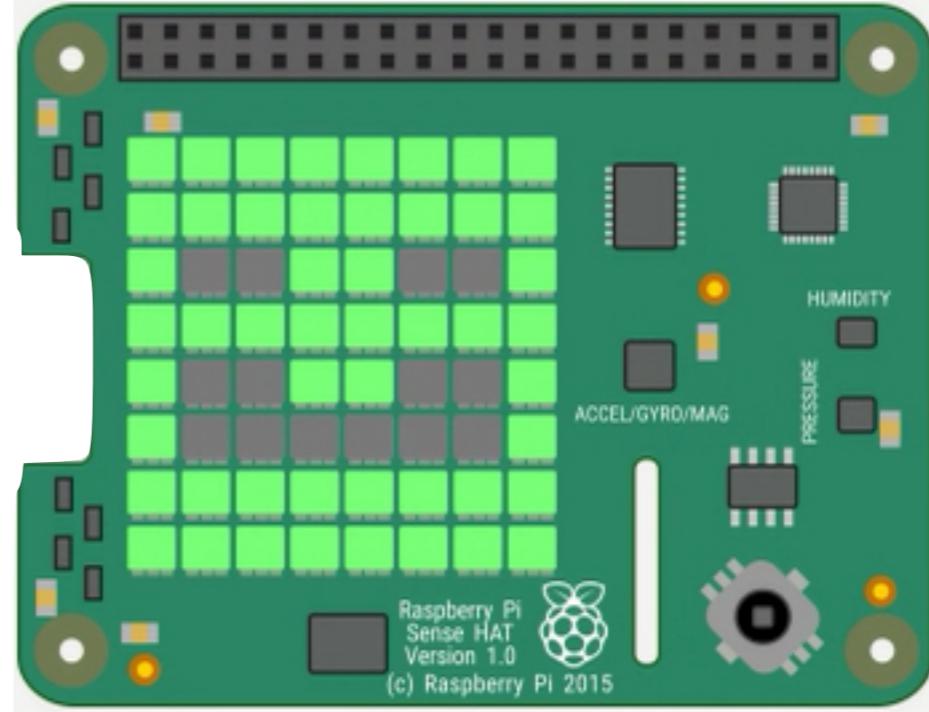
sense = SenseHat()
sense.clear()

red = (255,0,0)
green = (0,255,0)
black = (0,0,0)

greenmatrix = [ green for i in range(0,64) ]
sense.set_pixels(greenmatrix)
for j in range(2,6):
    if (j == 2 or j == 4):
        for i in range(1,7):
            sense.set_pixel(1,j, black) # only those
            sense.set_pixel(2,j, black) # only those
            sense.set_pixel(5,j, black) # only those
            sense.set_pixel(6,j, black) # only those
    elif j == 5:
        for i in range(1,7):
            sense.set_pixel(i,j, black) # all (1,5), ..., (6,5)
    else:
        pass

sleep(12)
sense.clear()

```



# LED MATRIX REMINDER: SET\_PIXEL VS SET\_PIXELS

See above.

```

from sense_emu import SenseHat
from time import sleep

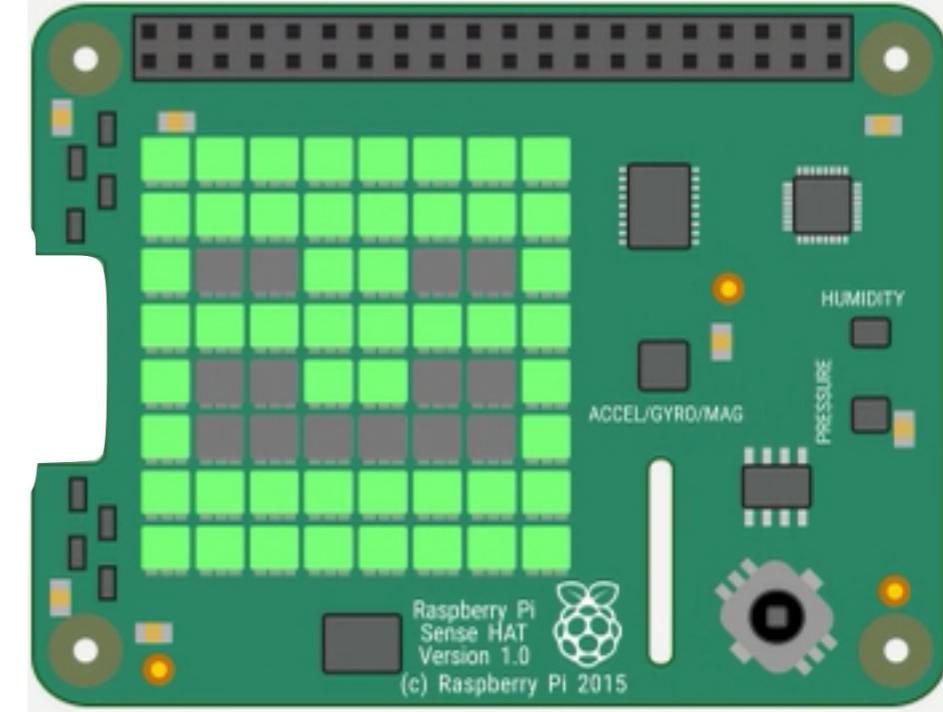
sense = SenseHat()
sense.clear()

red = (255,0,0)
green = (0,255,0)
black = (0,0,0)

sense.clear(green)
for j in range(2,6):
    if (j == 2 or j == 4):
        for i in range(1,7):
            sense.set_pixel(1,j, black) # only those
            sense.set_pixel(2,j, black) # only those
            sense.set_pixel(5,j, black) # only those
            sense.set_pixel(6,j, black) # only those
    elif j == 5:
        for i in range(1,7):
            sense.set_pixel(i,j, black) # all (1,5), ..., (6,5)
    else:
        pass

sleep(12)
sense.clear()

```



# LED MATRIX REMINDER: SET\_PIXEL VS SET\_PIXELS

See above.

```
from sense_emu import SenseHat
from time import sleep

sense = SenseHat()

red = (255, 0, 0)
green = (0, 255, 0)
blue = (0, 0, 255)
yellow = (0, 255, 255)

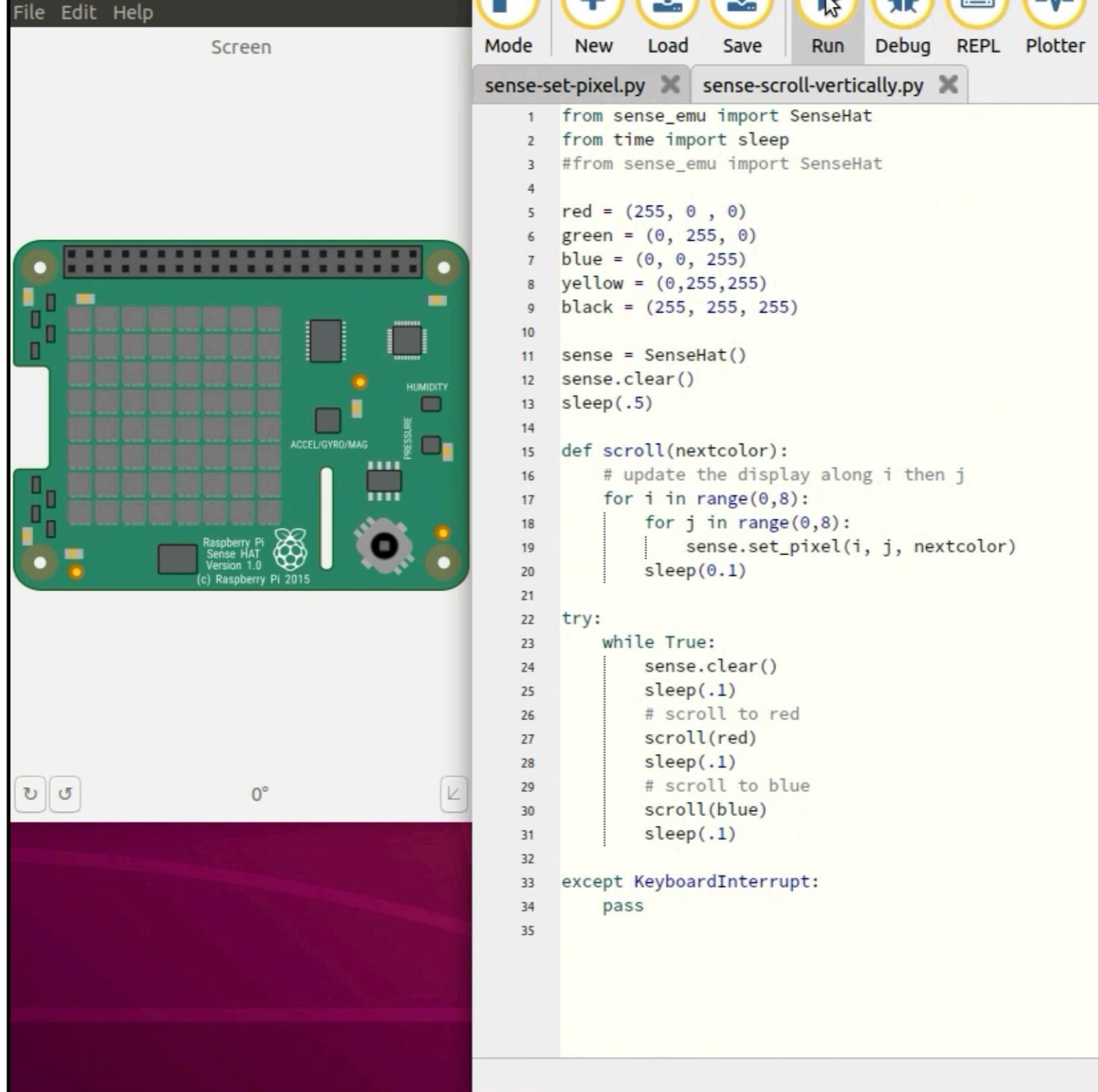
def scroll(RGBvalue):
    # clear the LED matrix
    sense.clear()
    sleep(.1)
    for i in range(0,8):
        # update the LED matrix along i
        for j in range(0,8):
            sense.set_pixel(i, j, RGBvalue)
        sleep(0.1)

try:
    while True:
        # scroll to red
        scroll(red)
        sleep(.1)
        # scroll to blue
        scroll(blue)
        sleep(.1)

except KeyboardInterrupt:
    pass
```

## SCROLLING COLOURS ON SENSE EMU LED MATRIX

We have a single LED matrix with 64 individual LEDs in Sense Hat (emu). We can use them all and scroll colours, e.g., **vertically**.



# SCROLLING COLOURS ON SENSE EMU LED MATRIX

We have a single LED matrix with 64 individual LEDs in Sense Hat (emu). We can use them all and scroll colours, e.g., **vertically**.

---

# Outline

---

Explorer Hat and Sense Emu "simulation": scrolling colours

Explorer Hat gets data from Adafruit IOT feeds and dashboards

Assignments: from Explorer Hat with Adafruit IOT to Sense Emu with Adafruit IOT

## LED switch

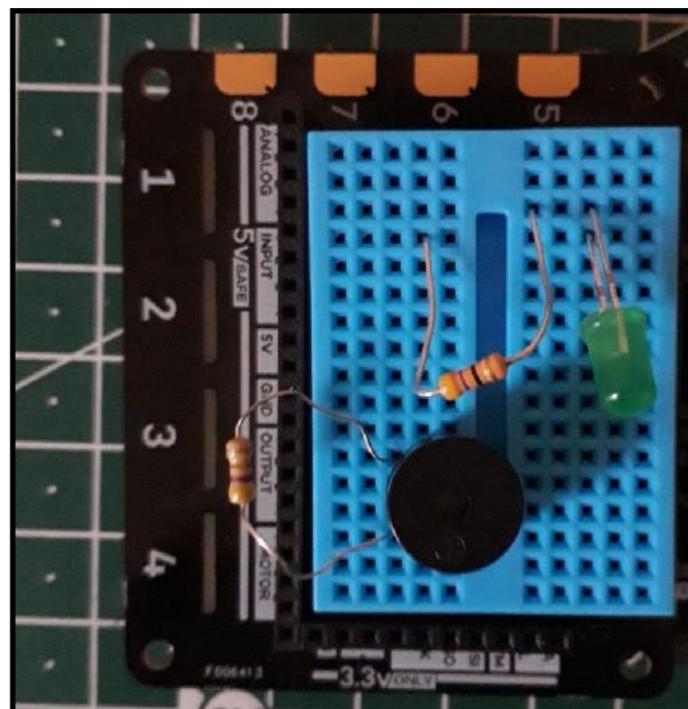


## OUTLOOK AND INTERACTION

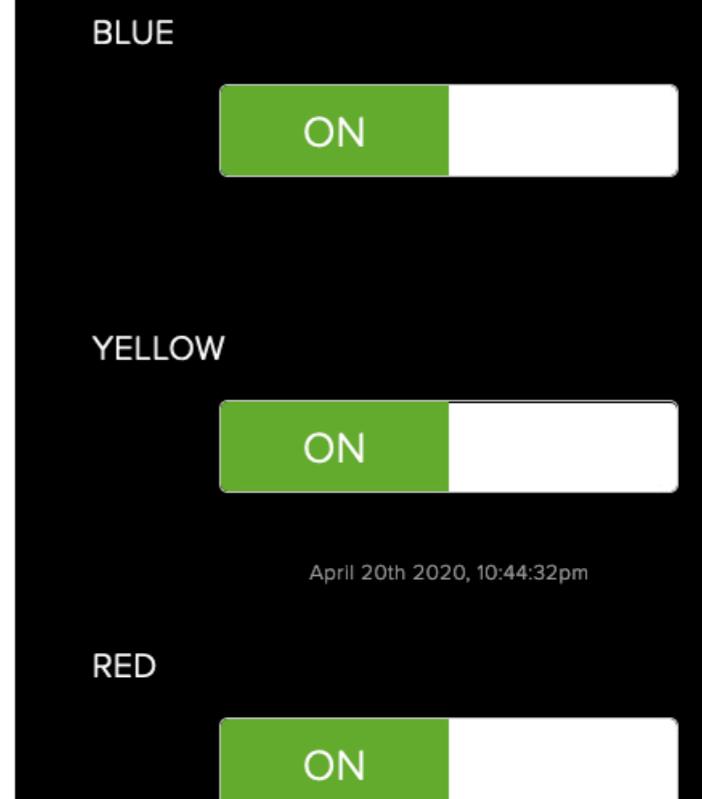
For the Explorer Hat, we are going to create a group of **toggle dashboards** (see right image), each connected to a feed for tracking the status of a LED of Explorer Hat.

We create a script which continuously (until **CTRL+C**):

- checks the color status of LEDs on Adafruit IO (careful: 'ON', 'OFF' are the values received by an Adafruit IO toggle dashboard)
- toggles the "physical" corresponding LED on Explorer Hat accordingly: on the right side, the **OFF status...**



## LED switch

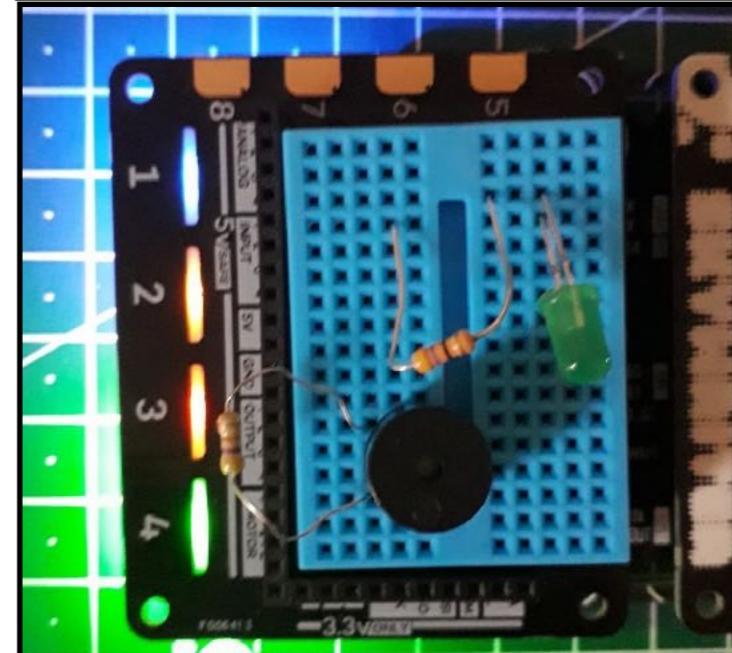


## OUTLOOK AND INTERACTION

For the Explorer Hat, we are going to create a group of **toggle dashboards** (see right image), each connected to a feed for tracking the status of a LED of Explorer Hat.

We create a script which continuously (until **CTRL+C**):

- checks the color status of LEDs on Adafruit IO (careful: 'ON', 'OFF' are the values received by an Adafruit IO toggle dashboard)
- toggles the "physical" corresponding LED on Explorer Hat accordingly: on the right side, the **OFF status...**





Actions ▾

[Create a New Dashboard](#)[Edit Selected Dashboard](#)[Remove Selected Dashboards](#) [eh led indicator](#) [graph](#) [LED switch](#) [ledtoggle](#) [rhyme](#) [sense matrix indicator](#) [sense matrix toggle](#) [servo\\_speed](#) [singleledindicator](#) [todayrhymes](#)

Loaded in 0.19 seconds.

**Help**

- [Get Help](#)
- [Quick Guides](#)
- [API Documentation](#)
- [FAQ](#)
- [Terms of Service](#)
- [Privacy Policy](#)
- [Send Feedback](#)

**Explore**

- [Learn](#)
- [IO Plus](#)
- [News](#)

*"Elegance is not a dispensable luxury but a quality that decides between success and failure" - Edsger Dijkstra*

```
import time
import explorerhat as eh
from Adafruit_IO import Client, Data, Feed, RequestError
from username_key import my_username, my_key

# Create a dictionary mapping colorled: color for each color of eh
# where color is a valid string for eh colours
# where colorled is the key of a feed, already created in AIOT
eh_color_dict = { 'redled': 'red' , 'greenled': 'green', 'blueled': 'blue', 'yellowled': 'yellow' }

# Create an instance of the REST client
aio = Client(my_username, my_key)

# Create the list feeds_keys with elements of the form colorled
# one per feed already created in AIOT
feeds_keys = ['blueled','yellowled','redled','greenled']

# Define a function that serves to
# access each colorled in feeds_keys and their latest_data.value and
# create a dictionary of pairs colorled : latest_data.value
def color_status(feeds_keys):
    color_status_dict = {}
    for colorled in feeds_keys:
        status = aio.feeds(colorled)
        color_status_dict[colorled] = aio.receive(status.key).value
    return color_status_dict
```

```
color_status_dict = {}
try:
    while True:
        # receive colored with datavalue from AIOT
        color_status_dict = color_status(feeds_keys)
        # create a list of colored with datavalue == ON in AIOT
        colorlist_on = [ colored for colored, status in color_status_dict.items() if
status == 'ON' ]
        # create a list of colored with datavalue == OFF in AIOT
        colorlist_off = [ colored for colored, status in color_status_dict.items() if
status == 'OFF' ]

        # iterate through colored ON in AIOT and toggle them ON in eh
        for colored in colorlist_on:
            color = eh_color_dict[colored]
            eh.light[color].on()
        # iterate through colored OFF in AIOT and toggle them OFF in eh
        for colored in colorlist_off:
            color = eh_color_dict[colored]
            eh.light[color].off()
        time.sleep(.3)

except KeyboardInterrupt:
    eh.light.off()
    pass
```

---

# Outline

---

Explorer Hat and Sense Emu "simulation": scrolling colours

Explorer Hat gets data from Adafruit IOT feeds and dashboards

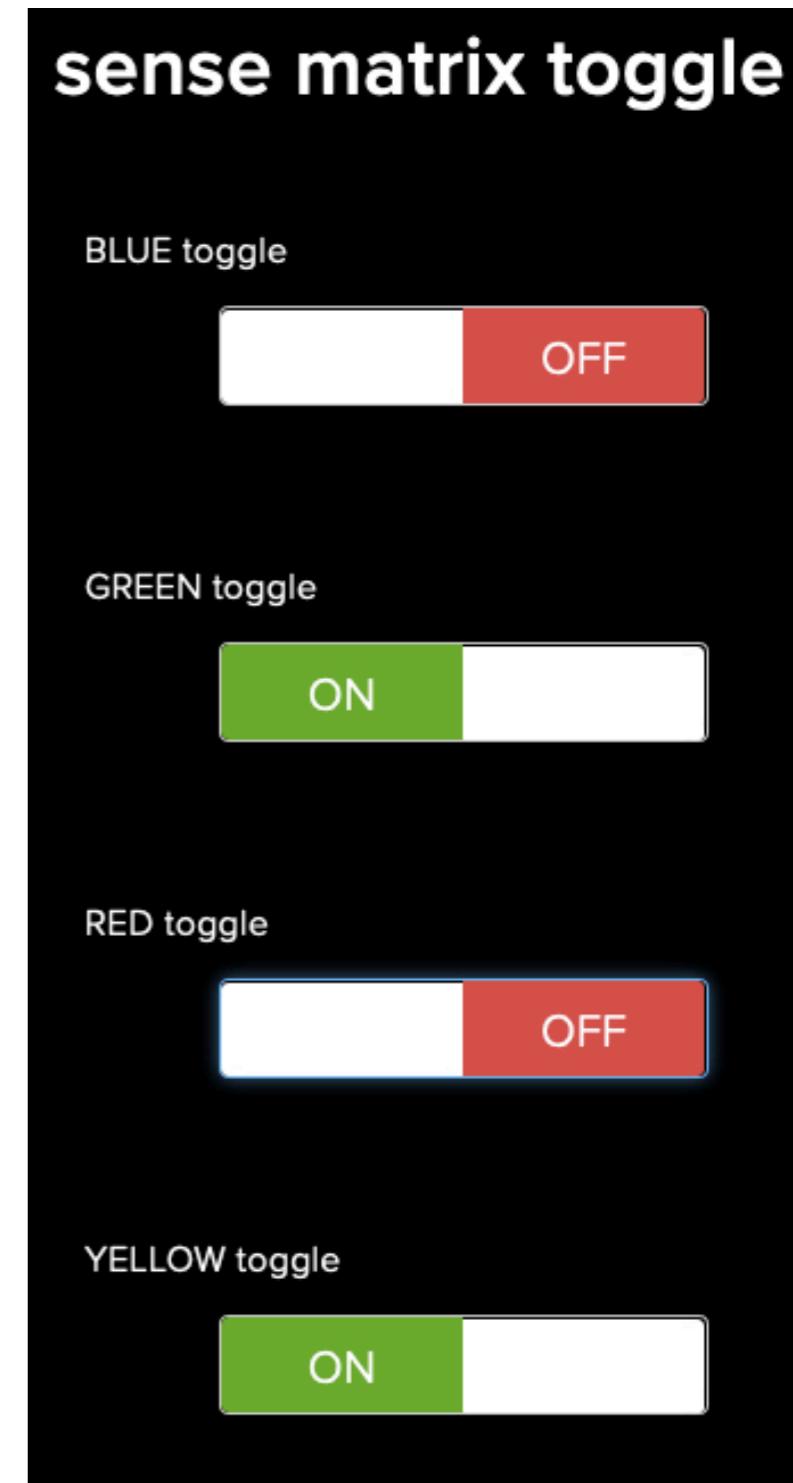
Assignments: from Explorer Hat with Adafruit IOT to Sense Emu with Adafruit IOT

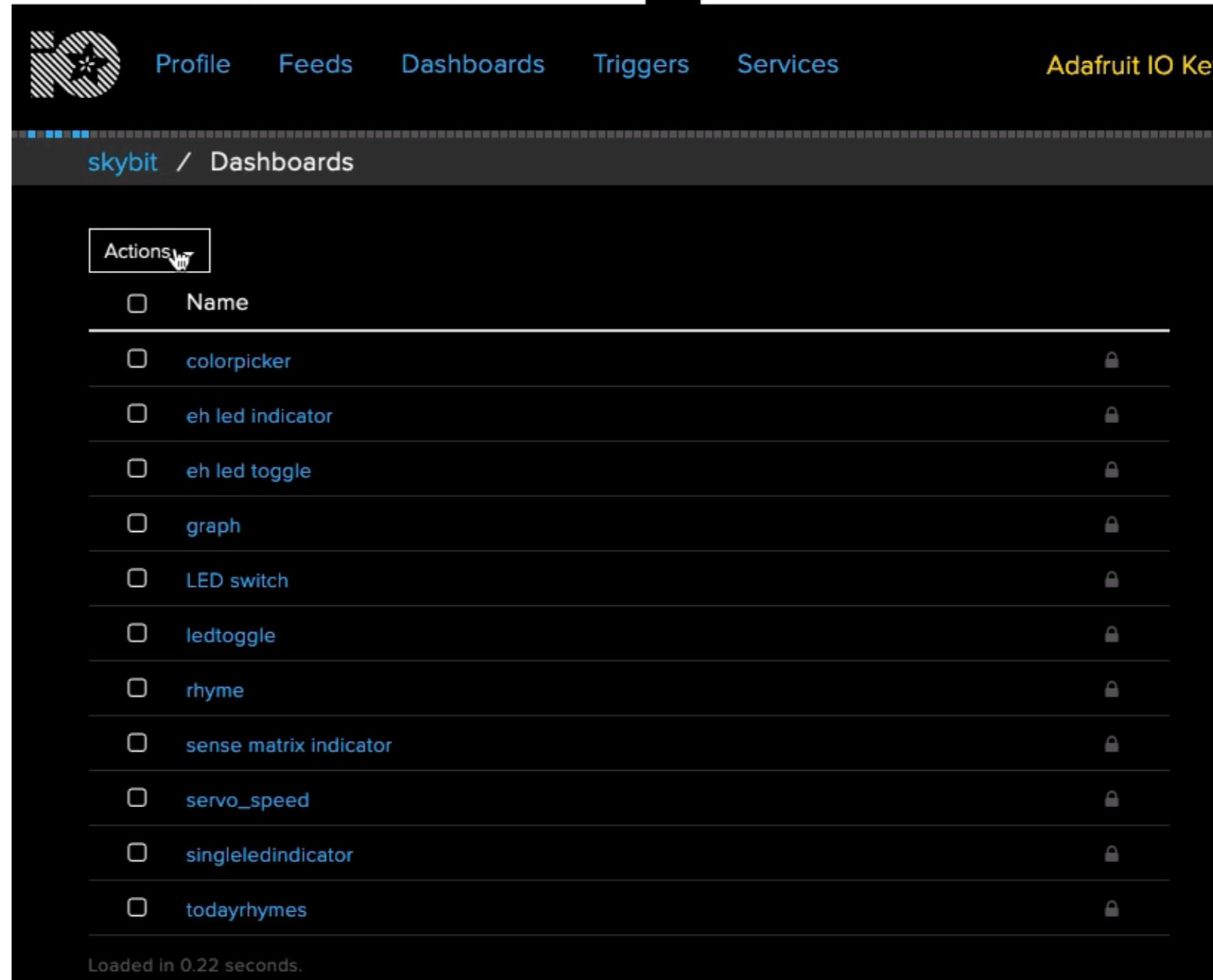
# Assignment 13

Start from the relevant code in slides for Explorer Hat and Adafruit IOT. Then, create a toggle dashboard for each different **color**matrix feed for blue, yellow, red, green.

Create a script which continuously (until CTRL+C):

- checks which colours in the dashboard are toggled 'ON'
- and scrolls colours in the sense-emu LED matrix accordingly, e.g., if blue and red are toggled on in the dashboard, then the LED matrix continues scrolling blue and red (e.g., vertically), and it becomes black if there is no color to scroll. See next.





The screenshot shows the Adafruit IO web interface. At the top, there's a navigation bar with links for Shop, Learn, Blog, Forums, Videos, Adabox, IO, and Adafruit IO Key. Below the navigation bar is a secondary header with links for Profile, Feeds, Dashboards, Triggers, and Services. The main content area is titled "skybit / Dashboards". A "Actions" button is highlighted with a yellow box. Below it is a list of ten items, each with a checkbox and a lock icon on the right:

- Name
- colorpicker
- eh led indicator
- eh led toggle
- graph
- LED switch
- ledtoggle
- rhyme
- sense matrix indicator
- servo\_speed
- singleledindicator
- todayrhymes

At the bottom of the list, it says "Loaded in 0.22 seconds."

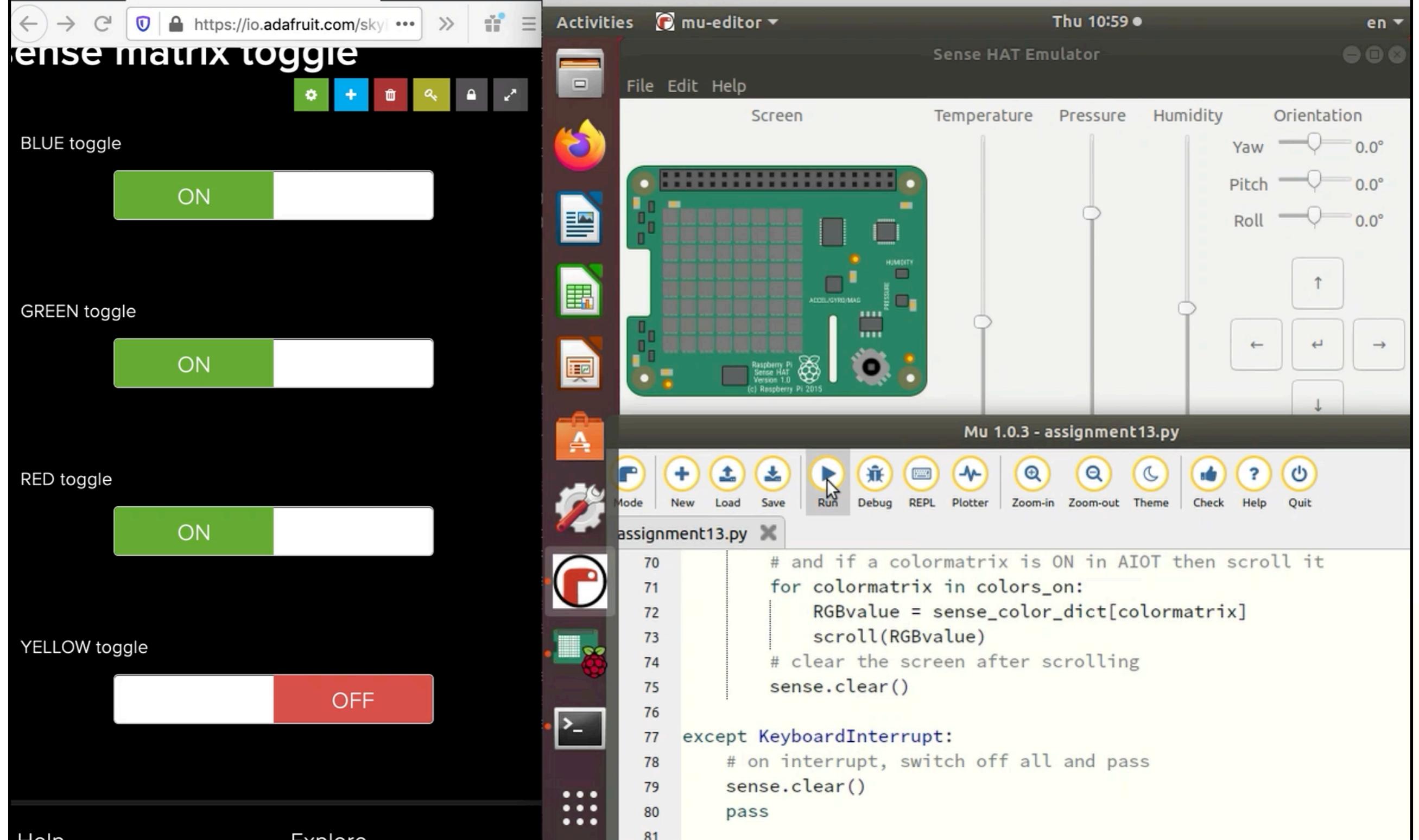
**Help**

[Get Help](#)  
[Quick Guides](#)  
[API Documentation](#)  
[FAQ](#)  
[Terms of Service](#)  
[Privacy Policy](#)  
[Send Feedback](#)

**Explore**

[Learn](#)  
[IO Plus](#)  
[News](#)

*"Elegance is not a dispensable luxury but a quality that decides between success and failure" - Edsger Dijkstra*



## Assignment 13: run the script

The script you create, when run, should indicate what you see above in the dashboard. See also <https://www.dropbox.com/s/onnmxl6fryv0fqk/assignment13.mov?dl=0>

---

# Summary

---

Explorer Hat and Sense Emu "simulation": scrolling colours

Explorer Hat gets data from Adafruit IOT feeds and dashboards

Assignments: from Explorer Hat with Adafruit IOT to Sense Emu with Adafruit IOT