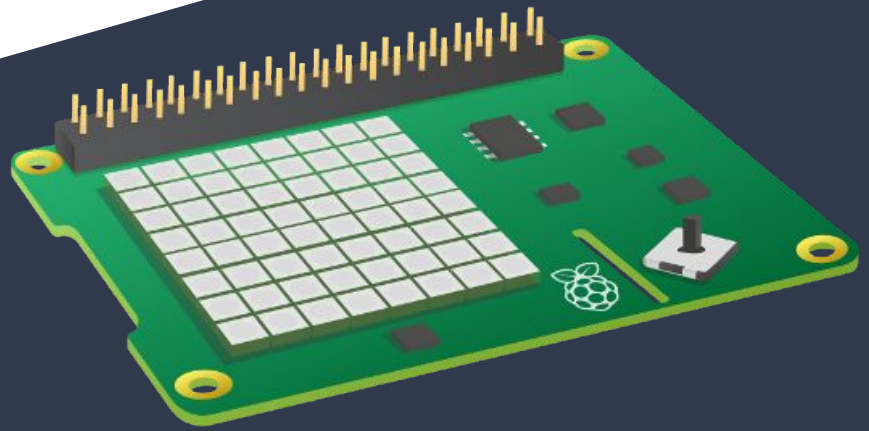
Week 2: Sense HAT

CSCI 3907: IoT using Raspberry Pi



The Basics

- The Sense HAT is an add-on board that sits on top of the Raspberry Pi.

- It has a variety of input sensors:
  - Temperature
  - Humidity
  - Pressure
  - Orientation
- Also has an 8x8 LED matrix for outputting information.

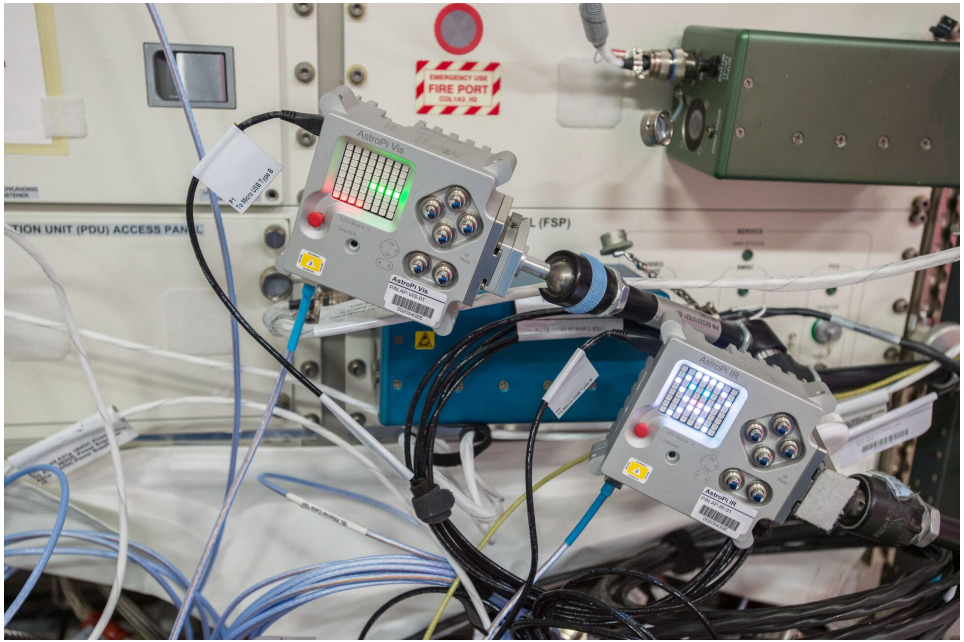
- Import Sense HAT library:

```
from sense_hat import SenseHat
```

- Initialize a Sense HAT object:

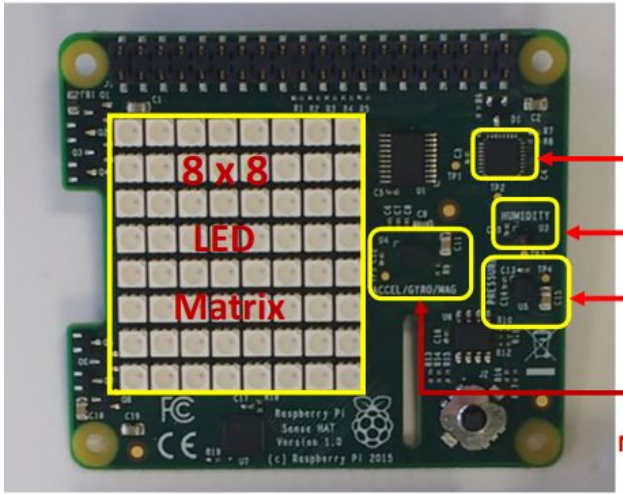
```
sense = SenseHat()
```

Astro Pi



Source: https://www.esa.int/Education/AstroPI/What_is_an_Astro_Pi

Sense HAT Diagram

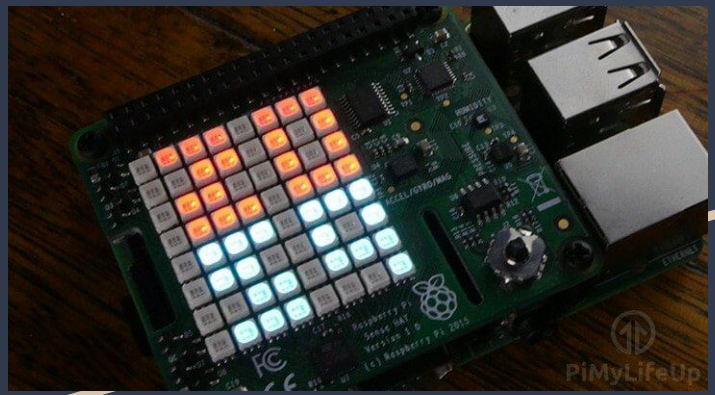


- The Sense HAT has a variety of sensors that can be read from:

"Temperature"	reads temperature in degrees Celsius
"Humidity"	reads humidity in % RH
"Pressure"	reads atmospheric pressure in millibars
"Rotation"	reads gyroscopic motion in revolutions per second
"Acceleration"	reads acceleration in terms of standard accelerations due to gravity on Earth's surface
"Orientation"	reads orientation relative to magnetic north in degrees
"Magnetic Field"	reads strength and direction of a magnetic field around the sensor in microteslas

Image Source: Professor Bulusu's CS 1010 / APSC 1001 Slides

Working with the LED Matrix



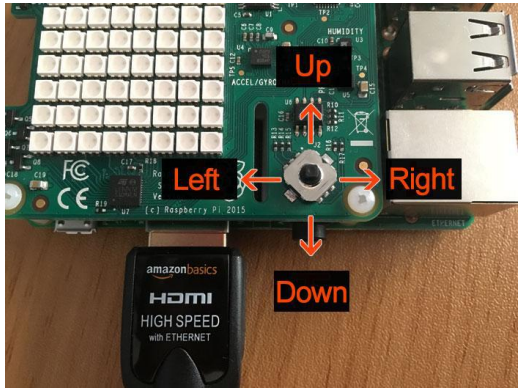

```
import sense_hat from SenseHat

sense = SenseHat()

# Set specific pixel
sense.set_pixel(<x>, <y>, (<r>, <g>, <b>))

# Set entire matrix
sense.set_pixels(<2d array>)

# Display a character
sense.show_letter(<chr>)

# Display a string
sense.show_message(<str>)

# Clear the display
sense.clear()
```

Reading Data from the Sensors

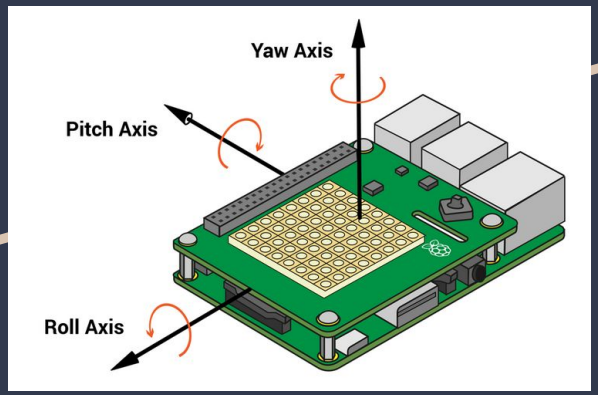
```
import sense_hat from SenseHat

sense = SenseHat()

# Read pressure
sense.get_pressure()

# Read temperature
sense.get_temperature()

# Read humidity
sense.get_humidity()
```

Reading Input from the Joystick

```
import sense_hat from SenseHat

sense = SenseHat()

# Call function foo anytime stick is moved
sense.stick.direction_any = foo

def foo():
     if event.direction == ‘up’:
          # Do something
     elif event.direction == ‘down’:
          # Do something
```



Detecting movement with the Inertial Measurement Unit (IMU)




```
import sense_hat from SenseHat

sense = SenseHat()

# Get angle of each axis in degrees
orientation = sense.get_orientation()

pitch = orientation[‘pitch’]
roll = orientation[‘roll’]
yaw = orientation[‘yaw’]

# Get amount of G-force acting on each axis
acceleration = sense.get_accelerometer_raw()

x = acceleration['x']
y = acceleration['y']
z = acceleration['z']
```

Assignment #1

Continuously pole the sense HAT for its orientation using the IMU. Then, using the LED matrix, display an arrow in the direction needed to “level-out” the Pi.

Part A:

- Start with only worrying about the ‘pitch’ axis
- Display an arrow in the direction that’s *closest* to levelling out the Raspberry Pi
    - *Hints:*
        - *When pitch=0, the Pi is “levelled-out”.*
        - *If pitch = 1, arrow should point to USB ports*
        - *If pitch = 359, arrow should point opposite to the USB ports*

Part B (*If time allows*):

- Now account for both pitch and roll
- Display an arrow in the direction that’s *closest* to levelling out the Pi on the axis that’s *furthest* from being level
- *i.e. If pitch > roll, display arrow along pitch axis*