

MAE 6291

Internet of Things for Engineers

Prof. Kartik Bulusu, MAE Dept.

Week 10 [04/02/2025]

- Guest lecture: Post model Edge AI for Fisheries management by Chris Rodley, CEO and Founder, Snap Information Technologies Ltd.
- Mosquitto – Open source MQTT broker
- The Eclipse Paho Project – Open source MQTT Implementation
- Experiments with senseHat
- Edge Compute Python codes
- In-class Raspberry Pi Lab - Mosquitto MQTT
 - Create the Edge-lab-messenger inspired by Facebook messenger
- Explore senseHat
- In-class Raspberry PI - paho MQTT
 - Publish and subscribe senseHat data

git clone https://github.com/gwu-mae6291-iot/spring2025_codes.git



School of Engineering
& Applied Science

Spring 2025

THE GEORGE WASHINGTON UNIVERSITY

Photo: Kartik Bulusu

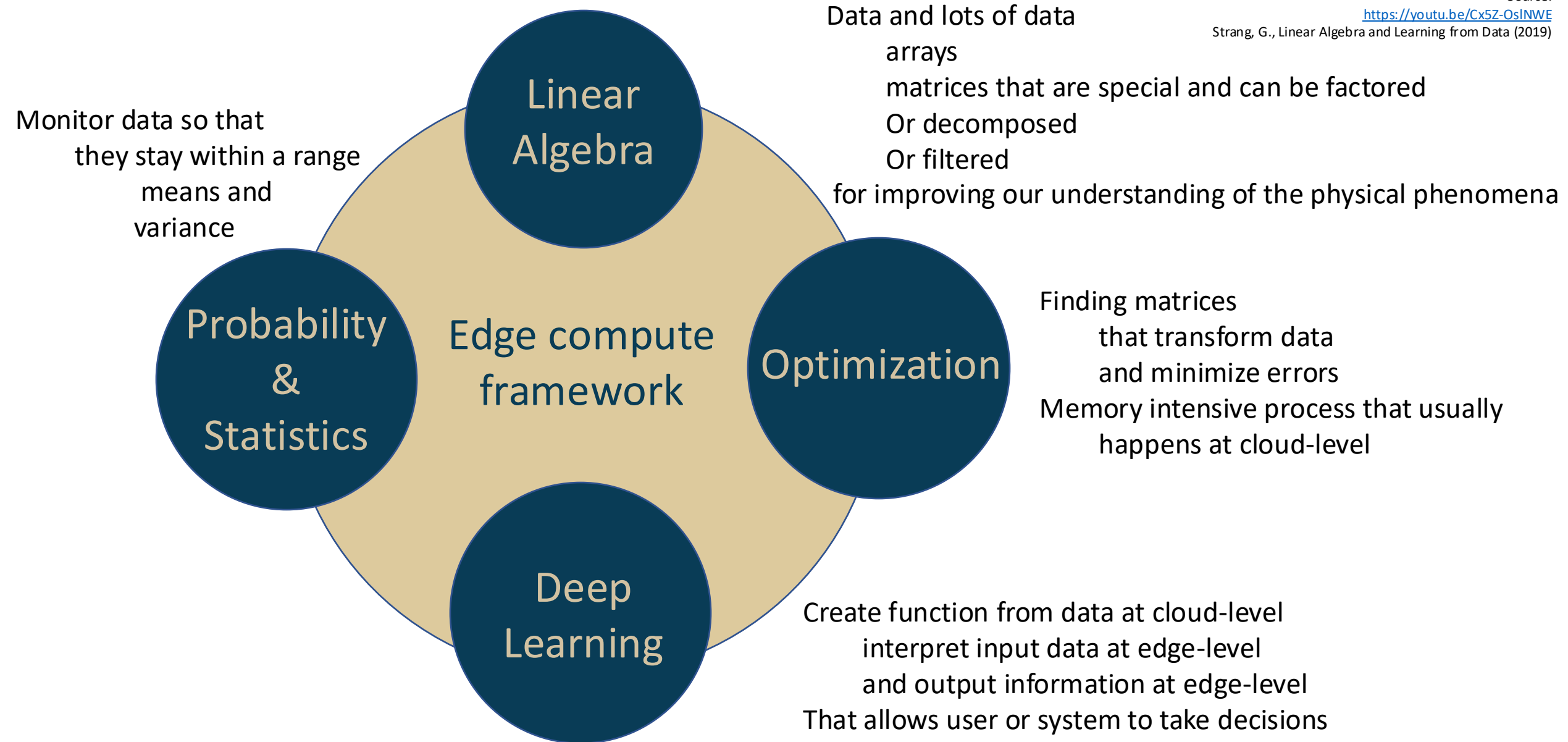
Final project proposals are all approved



Final project proposals

Name	Project title	March 26, 2025
Sumner Gubisch	Smart 3D Printer Filament Dryer and Desiccator	Approved
Eliot Hunter, Shota Kakiuchi	Autonomous 6-Axis Gimbal for Face Tracking and Web Upload	Approved
Dominic Savarino & Nathan Janssen	IoT Christmas Tree	Approved
Ben Sirota and Miya Liu	Music Motivator: Your New Fitness Coach	Approved
William Lynam	Smart Pet Feeder	Approved
Omar Nayfeh	LyftSync	Approved
Nick Neirotti	Brainfart	Approved
Aly Nguyen	Whack-A-Mole!	Approved
Yazan Sawalhi	Live Image Intensity Tracker	Approved
Alexandra Trotter	Small-scale automatic humidity regulation	Approved
Alex Vasilev	V Security System	Approved
Puchen Wang	Intelligent Forgotten Item Reminder System Based on Computer Vision and IoT	Approved
<i>Kartik Bulusu</i>	<i>Nothing to show !!!</i>	I am flunking this !





Explore Signal Processing with Scipy- Python library



SciPy (pronounced [/'saɪpaɪ/](#) "sigh pie"^[2]) is a [free and open-source Python](#) library used for [scientific computing](#) and technical computing.^[3]

SciPy contains modules for [optimization](#), [linear algebra](#), [integration](#), [interpolation](#), [special functions](#), [FFT](#), [signal](#) and [image processing](#), [ODE](#) solvers and other tasks common in science and engineering.



All activities today are a part of graded in-class lab
Download codes from github and demonstrate
[10 points]

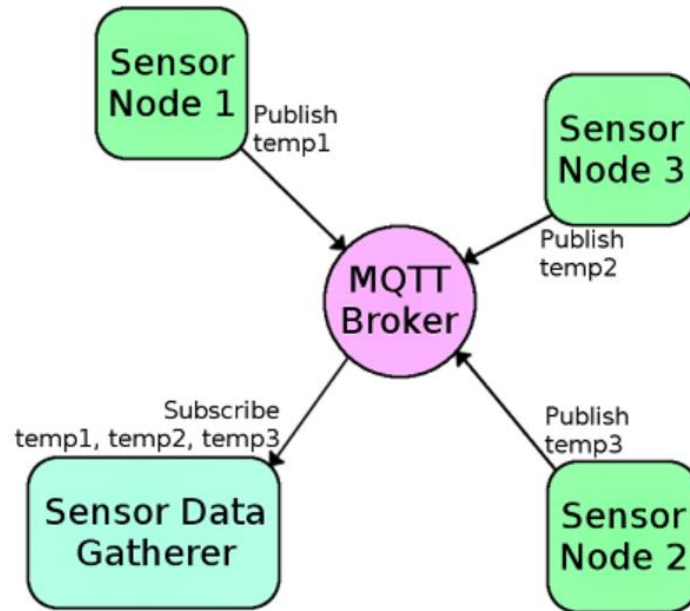


Redo MQTT Basics with Mosquitto [Message Queuing Telemetry Transport]

Goal: To demonstrate in groups of two how publishing and subscribing works practically – from various locations in SEH



Join me in creating Edge-Lab-Messenger



Inspiration



Created by Rolas Design
from Noun Project

Source:
<https://learn.sparkfun.com/tutorials/introduction-to-mqtt#the-basics>
<https://andre-benevides.medium.com/introduction-to-mqtt-and-configuration-of-a-mosquitto-broker-f0f7a7738bc8>
Health monitoring by Sinta Maulana from [Noun Project](#) (CC BY 3.0)
motion sensor by Tomas Knopp from [Noun Project](#) (CC BY 3.0)
fire sensor by Carolina Cani from [Noun Project](#) (CC BY 3.0)
chat by Rolas Design from [Noun Project](#) (CC BY 3.0)



Eclipse Mosquitto - An open source MQTT broker



Eclipse Mosquitto provides a lightweight server implementation of the MQTT protocol that is suitable for all situations from full power machines to embedded and low power machines.

Sensors and actuators, which are often the sources and destinations of MQTT messages, can be very small and lacking in power. This also applies to the embedded machines to which they are connected, which is where Mosquitto could be run.



Types of MQTT messages for your reference

CONNECT — Is the client request to connect to the broker

CONNACK — Acknowledgement of the connect

PUBLISH — Publishes a message to a topic

PUBACK — Acknowledgement of the publish with QoS level 1

PUBREC — Acknowledgement of the publish with QoS level 2 (2nd packet)

PUBREL — Response to the PUBREC. (3rd packet when using QoS level 2)

PUBCOMP — Response to PUBREL (4th and last packet when using QoS lvl 2)

SUBSCRIBE — Packet from the client to subscribe to topics

SUBACK — Acknowledgement of the subscribe packet

UNSUBSCRIBE — Packet from the client to unsubscribe from topics



Step-1: Eclipse Mosquitto - An open source MQTT broker

sudo apt-get update

sudo apt-get upgrade

sudo apt install mosquitto

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt install mosquitto
```

```
pi@raspberrypi:~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt install mosquitto  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libev4 libwebsockets8  
Suggested packages:  
  apparmor  
The following NEW packages will be installed:  
  libev4 libwebsockets8 mosquitto  
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.  
Need to get 241 kB of archives.  
After this operation, 543 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

```
pi@raspberrypi:~  
File Edit Tabs Help  
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.  
Need to get 241 kB of archives.  
After this operation, 543 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://mirror.pit.teraswitch.com/raspbian/raspbian stretch/main armhf libev4 armhf 1:4.22-1 [34.0 kB]  
Get:2 http://mirror.us.leaseweb.net/raspbian/raspbian stretch/main armhf mosquitto armhf 1.4.10-3+deb9u5 [122 kB]  
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf libwebsockets8 armhf 2.0.3-2+b1-rpt1 [85.2 kB]  
Fetched 241 kB in 5s (44.1 kB/s)  
Selecting previously unselected package libev4.  
(Reading database ... 97725 files and directories currently installed.)  
Unpacking libev4 (1:4.22-1) ...  
Selecting previously unselected package libwebsockets8:armhf.  
Unpacking libwebsockets8:armhf (2.0.3-2+b1-rpt1) ...  
Selecting previously unselected package mosquitto.  
Unpacking mosquitto (1.4.10-3+deb9u5) ...  
Setting up libev4 (1:4.22-1) ...  
Processing triggers for libc-bin (2.24-11+deb9u4) ...  
Setting up mosquitto (1.4.10-3+deb9u5) ...  
Processing triggers for systemd (232-25+deb9u14) ...
```



Step-2: restart Mosquitto

`sudo /etc/init.d/mosquitto restart`

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo /etc/init.d/mosquitto restart  
[ ok ] Restarting mosquitto (via systemctl): mosquitto.service.  
pi@raspberrypi:~ $
```

Step-3: Get your IP address

`ifconfig -a`

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ ifconfig  
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    ether b8:27:eb:3d:a8:1c txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 73 bytes 3847 (3.7 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 73 bytes 3847 (3.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.216 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 2600:4040:2db8:400:702:5225:99c5:a013 prefixlen 64 scopeid 0x0<g  
    local>  
    inet6 fe80::17d0:331:ba10:cde3 prefixlen 64 scopeid 0x20<link>  
    ether b8:27:eb:68:fd:49 txqueuelen 1000 (Ethernet)  
    RX packets 868 bytes 857162 (837.0 KiB)
```

Let's collect all your IP addresses



Prof. Kartik Bulusu, MAE Dept.

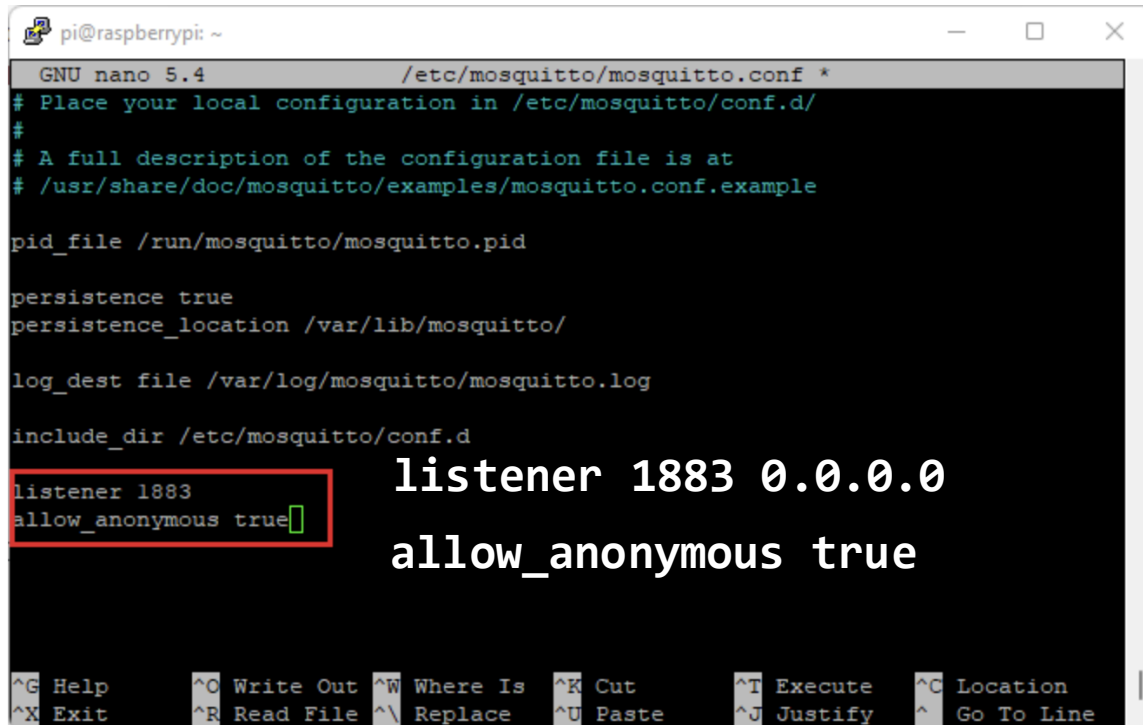
MAE 6291

Spring 2025

Internet of Things for Engineers

Step-4: Enable Remote Access to Mosquitto Broker (No Authentication)

sudo nano /etc/mosquitto/mosquitto.conf



```
pi@raspberrypi: ~
GNU nano 5.4 /etc/mosquitto/mosquitto.conf *
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1883
allow_anonymous true

listener 1883 0.0.0.0
allow_anonymous true

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace  ^U Paste     ^J Justify   ^_ Go To Line
```

CTRL x

- **Add a listener:** Include a line like **listener 1883 0.0.0.0** in your mosquitto.conf file. This instructs Mosquitto to listen for connections on port 1883 across all network interfaces.
- **Allow anonymous connections:** Add **allow_anonymous true** to the configuration file. This permits clients to connect without requiring authentication.
 - Note: This is the least secure option, so consider implementing authentication (usernames and passwords) for production environments.

sudo systemctl restart mosquitto

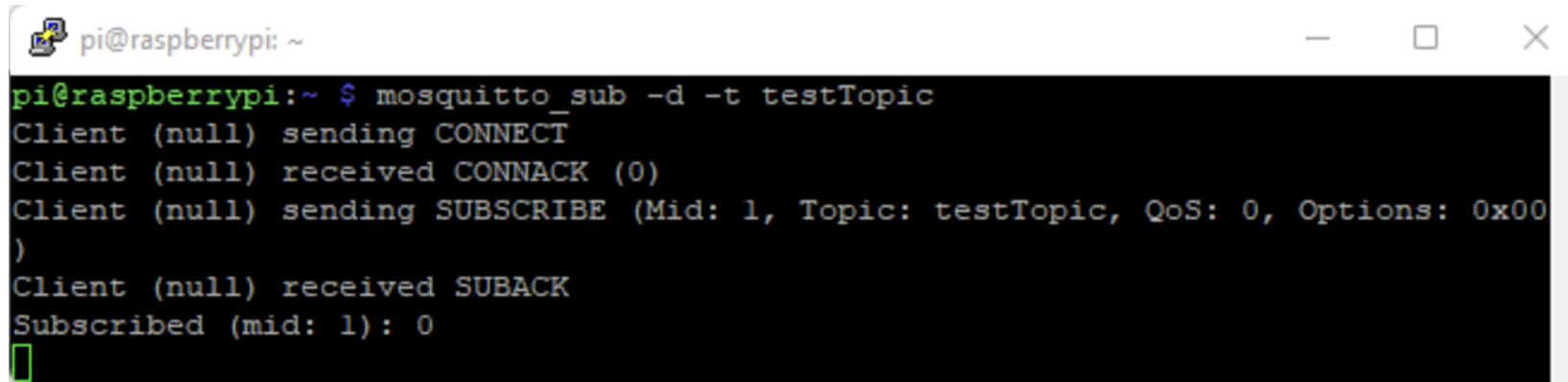
sudo systemctl status mosquito

CTRL z

Step-4: Subscribing “Hello World!” Message to *ProfBulusu* Topic

Install mosquitto-clients:

```
sudo apt install -y mosquitto mosquitto-clients  
clear
```



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ mosquitto_sub -d -t testTopic  
Client (null) sending CONNECT  
Client (null) received CONNACK (0)  
Client (null) sending SUBSCRIBE (Mid: 1, Topic: testTopic, QoS: 0, Options: 0x00  
)  
Client (null) received SUBACK  
Subscribed (mid: 1): 0  
█
```

Open a terminal window and type the following:

```
mosquitto_sub -v -t '#' -h <Prof.Bulusu's IP address>
```

Step-4: “Hello World!” Message with *ProfBulusu* Topic will be published

```
mosquitto_pub -d -t ProfBulusu -m "Hello world from ProfBulusu!"
```

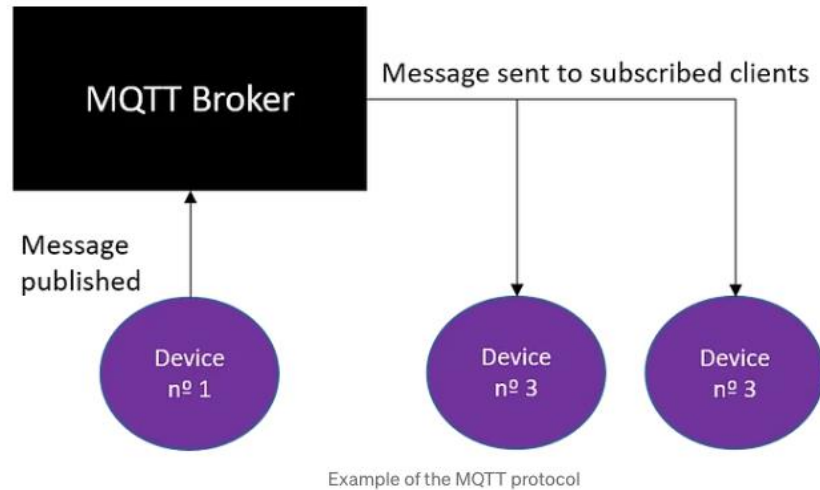
Step-5: Publish your topic *Your-Names* and “*YourMessage*”

You must follow the example-format below

```
mosquitto_pub -d -t Alex-Aly -m "Here is our IP Address: IP Address"
```



MQTT paradigm



Hardware

Broker

- The broker is the server
- It distributes the information to the interested devices connected to the server.

Client

- The device that connects to broker to send or receive information.

Messaging

Topic

- The name that the message is about.
- Clients publish, subscribe, or do both to a topic.

Publish

- Clients that send information to the broker to distribute to interested clients based on the topic name.

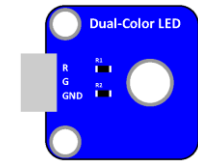
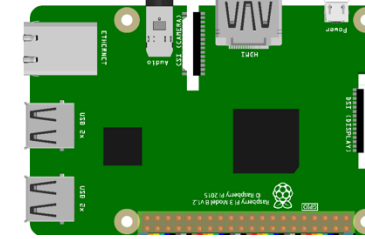
Subscribe

- Clients tell the broker which topic(s) they're interested in.

QoS

- Quality of Service to the broker
- Integer value ranging from 0-2.

Source: <https://andre-benevides.medium.com/introduction-to-mqtt-and-configuration-of-a-mosquitto-broker-f0f7a7738bc8>
<https://learn.sparkfun.com/tutorials/introduction-to-mqtt#the-basics>



Explore SenseHat

(The RPi companion sensor in the International Space Station)

Goal: Use paho-mqtt to publish and subscribe data



Example of who is using the sense HAT and where
- Astro Pi



Source: https://youtu.be/kk_7KNuRLrk

What we will do today

- **Co-work**
 - Observe, ask and try in groups
- **Write small program using Python**
- **Think about**
 - Challenges, Opportunities, Gaps and Surprises

What we will learn today

- Communicate with the Sense HAT using Python
- Access the outputs of the Sense HAT
- Use the Sense HAT library to display messages and images
- Use loops to repeat certain code blocks



Step-1: “Put on the Sense HAT”

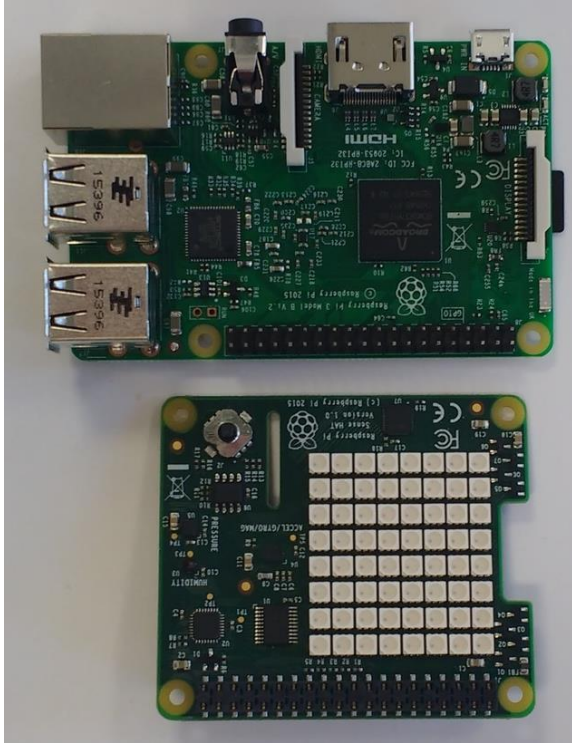


Image and animation source: <https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat/2>

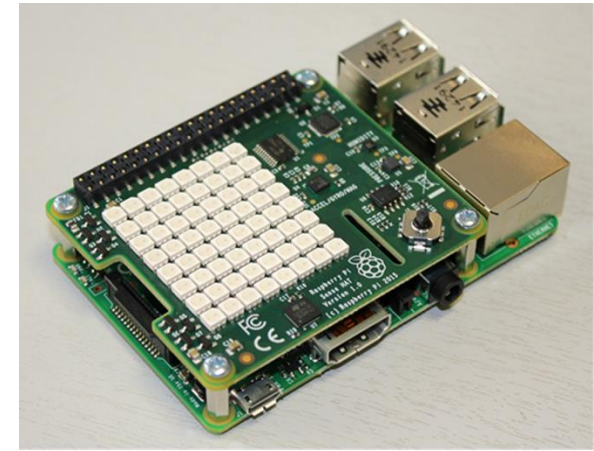
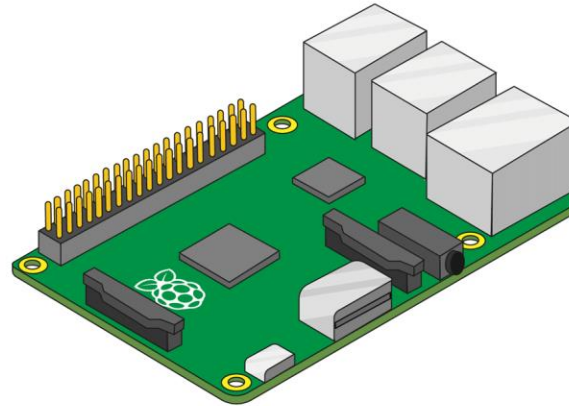


Image source: <https://reference.wolfram.com/language/ref/device/SenseHAT.html>



Source: <https://youtu.be/8NwWNOMqai4>

“and take a closer look” ...



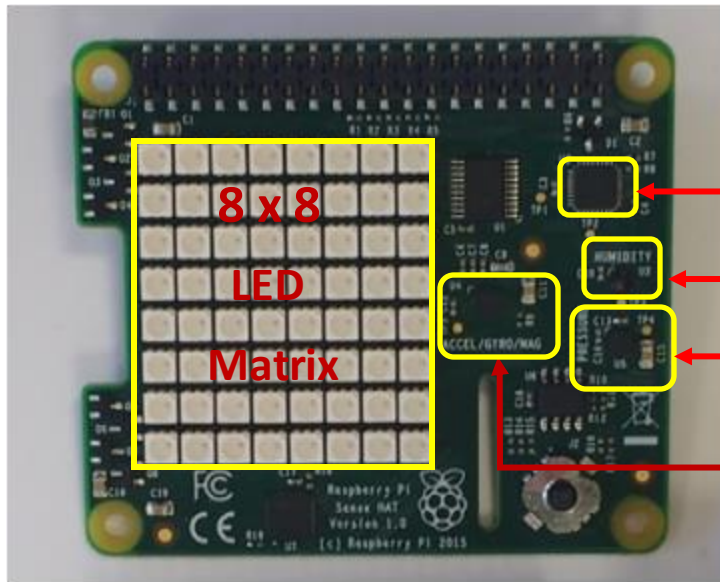
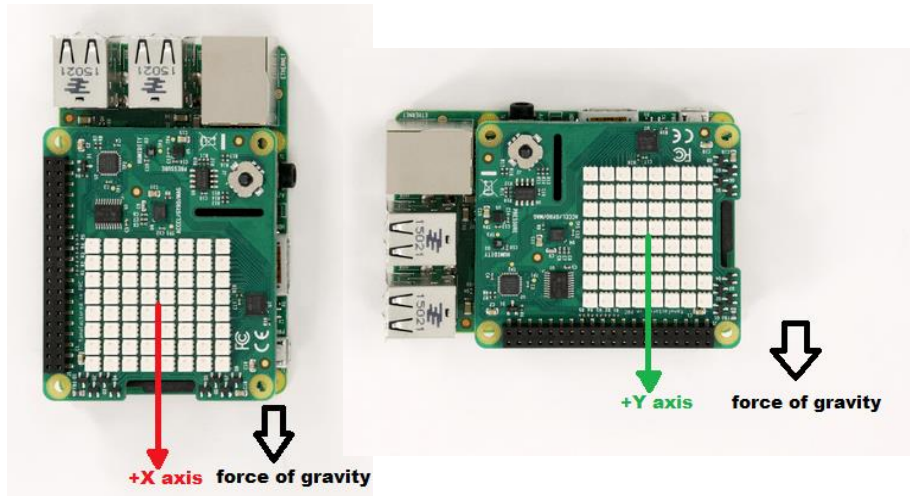


Image source: <https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat/2>

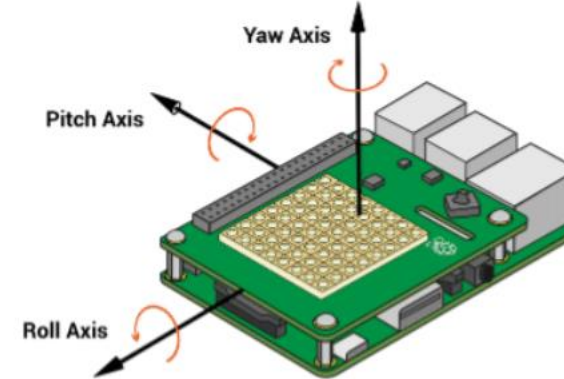


Source: <https://www.mathworks.com/help/supportpkg/raspberrypi/examples/auto-rotate-an-image-displayed-on-sense-hat-led-matrix.html>

▪ 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

▪ The gyroscope, accelerometer, and magnetometer sensors return a list of three values that corresponds to $\{roll, pitch, yaw\}$, as oriented according to the following image:



Starting point for further exploration:

[Link for “Getting started with the Sense HAT”](#)

Source: <https://reference.wolfram.com/language/ref/device/SenseHAT.html>

Eclipse paho - Another open source MQTT broker



paho-mqtt 1.6.1

```
pip install paho-mqtt
```



Paho is an



Eclipse-paho provides a client class which enable applications to connect to an [MQTT](#) broker to publish messages, and to subscribe to topics and receive published messages.

It also provides some helper functions to make publishing one off messages to an MQTT server very straightforward.



Step-2: Install paho-mqtt & psutil libraries

sudo apt-get update && sudo apt-get upgrade

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo apt-get update && sudo apt-get upgrade  
Get:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]  
Hit:2 http://archive.raspberrypi.org/debian stretch InRelease  
Fetched 15.0 kB in 5s (2,647 B/s)  
Reading package lists... Done  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
pi@raspberrypi:~$
```

pip install paho-mqtt

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo pip3 install paho-mqtt  
/usr/lib/python3/dist-packages/secretstorage/dhcrypto.py:15: CryptographyDeprecationWarning: Python 3.5 support will be dropped in the next release of cryptography. Please upgrade your Python.  
  from cryptography.utils import int_from_bytes  
DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. Please upgrade your Python as Python 3.5 is no longer maintained. pip 21.0 will drop support for Python 3.5 in January 2021. pip 21.0 will remove support for this functionality.  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: paho-mqtt in /usr/local/lib/python3.5/dist-packages (1.6.1)  
pi@raspberrypi:~$
```

pip install psutil

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo pip3 install psutil  
/usr/lib/python3/dist-packages/secretstorage/dhcrypto.py:15: CryptographyDeprecationWarning: Python 3.5 support will be dropped in the next release of cryptography. Please upgrade your Python.  
  from cryptography.utils import int_from_bytes  
DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. Please upgrade your Python as Python 3.5 is no longer maintained. pip 21.0 will drop support for Python 3.5 in January 2021. pip 21.0 will remove support for this functionality.  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: psutil in /usr/local/lib/python3.5/dist-packages (5.9.4)  
pi@raspberrypi:~$
```

paho-mqtt 2.0.0

pip install paho-mqtt

psutil 5.9.8

pip install psutil

psutil (process and system utilities) is a cross-platform library for retrieving information on **running processes** and **system utilization** (CPU, memory, disks, network, sensors) in Python



Prof. Kartik Bulusu, MAE Dept.

MAE 6291

Spring 2025

Internet of Things for Engineers

Works in groups of two to

- i. Acquire senseHat-data
- ii. publish and subscribe sensehat-data and
- iii. learn how to implement paho-mqtt in python programs



Someone should summarize what we learned today

