```
Exalens/XDK/acceleration/x 257
Exalens/XDK/acceleration/y 314
Exalens/XDK/acceleration/z 4346
Exalens/XDK/acceleration/magnitude 4364
Exalens/XDK/gyroscope/x -9
Exalens/XDK/gyroscope/y 24
Exalens/XDK/gyroscope/z 11
Exalens/XDK/orientation/x 1068851641
pvgarage/HM-1500_OBEN/ch2/YieldTotal 338.338
iot-2/type/cMT2078X/id/cMT-Shak/evt/topic 2/fmt/json {
    "Сетевой" : 27.65,
    "ГВС" : 24.33,
    "Котловой" : 26.98,
    "Уличная" : 16.74
}

Exalens/XDK/orientation/y 1066514529
rt/DSL23002977/res/ 1713211904,24704,0
Exalens/XDK/orientation/z 1066588972
ruuvi/DC:B9:2C:80:0E:7C/EC:A9:07:01:90:1A {
        "gw_mac":       "DC:B9:2C:80:0E:7C",
        "rssi": -63,
        "aoa":  [],
        "gwts": "1713201149",
        "ts":   "1713201149",
        "data": "02011A020A0C0AFF4C0010050014885B77",
        "coords":       ""
}
Exalens/XDK/magnetic/x -461
ruuvi/DC:B9:2C:80:0E:7C/48:55:19:9A:E3:F2 {
        "gw_mac":       "DC:B9:2C:80:0E:7C",
        "rssi": -74,
        "aoa":  [],
        "gwts": "1713201149",
        "ts":   "1713201149",
        "data": "02010610FFA90B0105000B00100AF0E39A195548",
        "coords":       ""
}
Exalens/XDK/acceleration/x 251
ruuvi/DC:B9:2C:80:0E:7C/DC:56:E7:42:BA:E7 {
        "gw_mac":       "DC:B9:2C:80:0E:7C",
        "rssi": -67,
        "aoa":  [],
        "gwts": "1713201149",
        "ts":   "1713201149",
        "data": "02011A020A0C0AFF4C0010050514C184BF",
        "coords":       ""
}
ruuvi/DC:B9:2C:80:0E:7C/72:55:73:EB:CA:F3 {
        "gw_mac":       "DC:B9:2C:80:0E:7C",
```

```
pi@raspberrypi:~/lab4 $ mosquitto_sub -h test.mosquitto.org -t "Exalens/XDK/acceleration/x" -v
Exalens/XDK/acceleration/x 247
Exalens/XDK/acceleration/x 252
Exalens/XDK/acceleration/x 289
Exalens/XDK/acceleration/x 263
Exalens/XDK/acceleration/x 251
Exalens/XDK/acceleration/x 276
Exalens/XDK/acceleration/x 237
Exalens/XDK/acceleration/x 268
Exalens/XDK/acceleration/x 252
Exalens/XDK/acceleration/x 237
Exalens/XDK/acceleration/x 284
Exalens/XDK/acceleration/x 233
Exalens/XDK/acceleration/x 274
Exalens/XDK/acceleration/x 273
Exalens/XDK/acceleration/x 234
Exalens/XDK/acceleration/x 242
Exalens/XDK/acceleration/x 226
Exalens/XDK/acceleration/x 251
Exalens/XDK/acceleration/x 228
Exalens/XDK/acceleration/x 234
Exalens/XDK/acceleration/x 275
Exalens/XDK/acceleration/x 288
Exalens/XDK/acceleration/x 258
Exalens/XDK/acceleration/x 249
Exalens/XDK/acceleration/x 257
Exalens/XDK/acceleration/x 285
Exalens/XDK/acceleration/x 268
Exalens/XDK/acceleration/x 279
Exalens/XDK/acceleration/x 249
Exalens/XDK/acceleration/x 246
Exalens/XDK/acceleration/x 260
Exalens/XDK/acceleration/x 255
Exalens/XDK/acceleration/x 246
Exalens/XDK/acceleration/x 262
Exalens/XDK/acceleration/x 244
Exalens/XDK/acceleration/x 227
Exalens/XDK/acceleration/x 251
Exalens/XDK/acceleration/x 264
Exalens/XDK/acceleration/x 254
Exalens/XDK/acceleration/x 267
Exalens/XDK/acceleration/x 250
Exalens/XDK/acceleration/x 253
```

```
pi@raspberrypi:~/lab4 $ mosquitto_sub -h broker.hivemq.com -t "uiowa/iot/lab4/#" -v
uiowa/iot/lab4/cpu1 0.50779
uiowa/iot/lab4/cpu2 0.043995
uiowa/iot/lab4/cpu3 0.51825
uiowa/iot/lab4/cpu1 0.50618
uiowa/iot/lab4/cpu2 0.044945
uiowa/iot/lab4/cpu3 0.51557
uiowa/iot/lab4/cpu1 0.4893
uiowa/iot/lab4/cpu2 0.07882
uiowa/iot/lab4/cpu3 0.45022
uiowa/iot/lab4/cpu1 0.49238
uiowa/iot/lab4/cpu2 0.15145
uiowa/iot/lab4/cpu3 0.39777
uiowa/iot/lab4/cpu1 0.50714
uiowa/iot/lab4/cpu2 0.23964
uiowa/iot/lab4/cpu3 0.45732
uiowa/iot/lab4/cpu1 0.50543
uiowa/iot/lab4/cpu2 0.34578
uiowa/iot/lab4/cpu3 0.42916
uiowa/iot/lab4/cpu1 0.5207
uiowa/iot/lab4/cpu2 0.4743
uiowa/iot/lab4/cpu3 0.55273
```

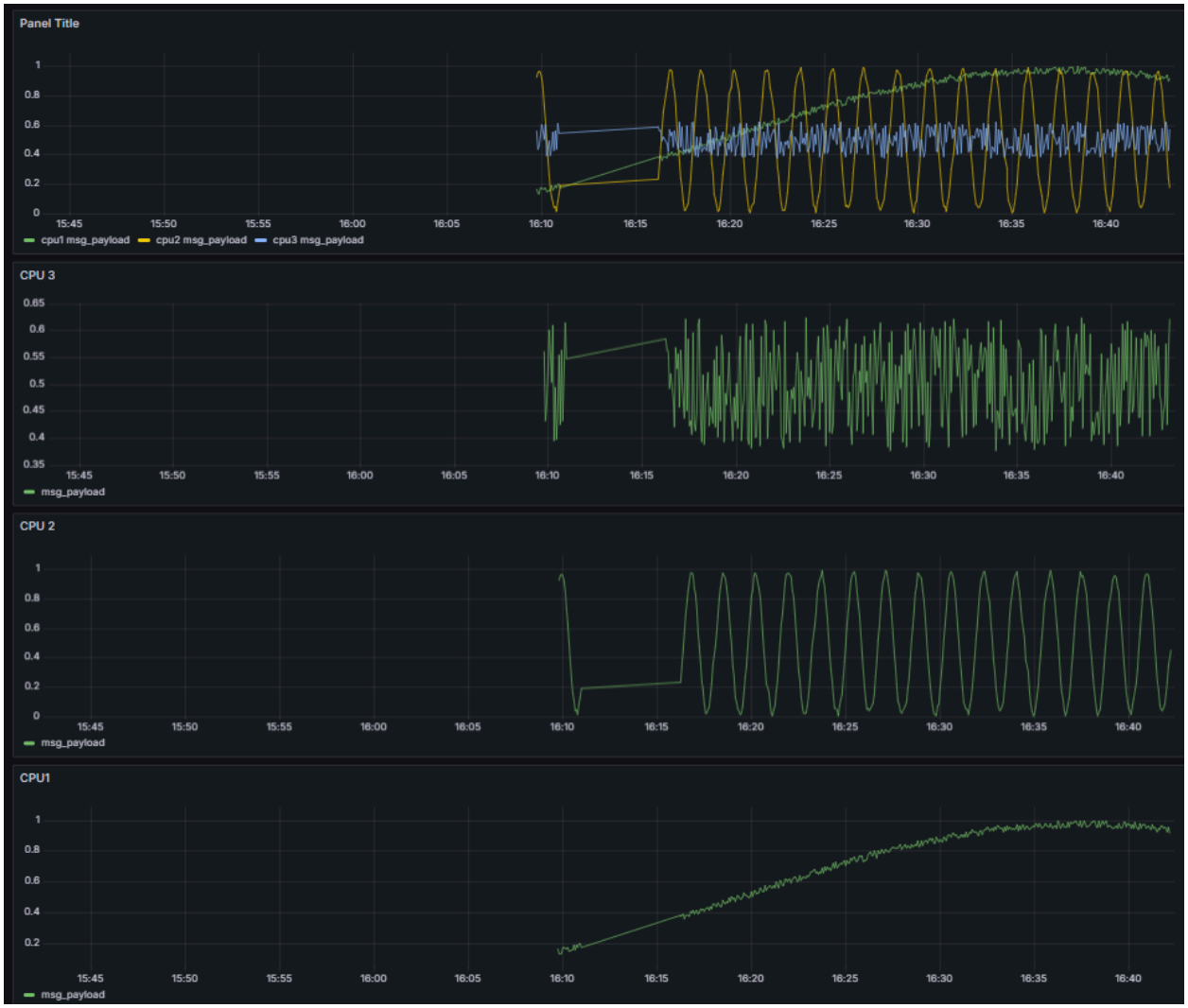
Interactive Table View (press q to exit mode, shift+up/down to navigate table

Name: cpu1

index	time	m_decode	msg_payload
1	1713203351669166848.0000000000	0.58273	0.5827300000
2	1713203355583540736.0000000000	0.57769	0.5776900000
3	1713203359687788544.0000000000	0.60238	0.6023800000
4	1713203363692257280.0000000000	0.6081	0.6081000000
5	1713203367721556480.0000000000	0.60045	0.6004500000
6	1713203371732662528.0000000000	0.57874	0.5787400000
7	1713203375827866368.0000000000	0.59149	0.5914900000
8	1713203379803847680.0000000000	0.60717	0.6071700000
9	1713203383730005760.0000000000	0.60428	0.6042800000
10	1713203387728236288.0000000000	0.5675	0.5675000000
11	1713203391752145408.0000000000	0.59586	0.5958600000
12	1713203395823816704.0000000000	0.58056	0.5805600000
13	1713203399860482048.0000000000	0.56657	0.5665700000
14	1713203403913619200.0000000000	0.57573	0.5757300000
15	1713203407937245440.0000000000	0.57531	0.5753100000
16	1713203412074538752.0000000000	0.55171	0.5517100000
17	1713203416341883392.0000000000	0.56987	0.5698700000
18	1713203420319585792.0000000000	0.57807	0.5780700000
19	1713203424335456256.0000000000	0.56572	0.5657200000
20	1713203428322886144.0000000000	0.55609	0.5560900000
21	1713203432325212928.0000000000	0.54063	0.5406300000
22	1713203436455439360.0000000000	0.54959	0.5495900000

4 Columns, 22 Rows, Page 1/1
Table 1/1, Statement 1/1

```
(venv) pi@raspberrypi:~/lab4 $ python3 lab4.py
/home/pi/lab4/lab4.py:35: DeprecationWarning: Callback API version 1 is deprecated, update to latest v
ersion
  mqtt_client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION1)
Connected with result code 0
Received a message on topic: uiowa/iot/lab4/cpu1
type of decoded message payload is: <class 'str'>
Type of msg_payload after json.loads() is: <class 'float'>
Contents of message payload:
0.97179
Received a message on topic: uiowa/iot/lab4/cpu2
type of decoded message payload is: <class 'str'>
Type of msg_payload after json.loads() is: <class 'float'>
Contents of message payload:
0.8445
Received a message on topic: uiowa/iot/lab4/cpu3
type of decoded message payload is: <class 'str'>
Type of msg_payload after json.loads() is: <class 'float'>
Contents of message payload:
0.43804
Received a message on topic: uiowa/iot/lab4/cpu1
type of decoded message payload is: <class 'str'>
Type of msg_payload after json.loads() is: <class 'float'>
Contents of message payload:
0.98482
Received a message on topic: uiowa/iot/lab4/cpu2
type of decoded message payload is: <class 'str'>
Type of msg_payload after json.loads() is: <class 'float'>
Contents of message payload:
0.93497
Received a message on topic: uiowa/iot/lab4/cpu3
type of decoded message payload is: <class 'str'>
Type of msg_payload after json.loads() is: <class 'float'>
Contents of message payload:
0.59035
^CTraceback (most recent call last):
  File "/home/pi/lab4/lab4.py", line 48, in <module>
    mqtt_client.loop_forever()
  File "/home/pi/lab4/venv/lib/python3.11/site-packages/paho/mqtt/client.py", line 2291, in loop_forev
er
    rc = self._loop(timeout)
         ^^^^^^^^^^^^^^^^^^^
  File "/home/pi/lab4/venv/lib/python3.11/site-packages/paho/mqtt/client.py", line 1657, in _loop
    socklist = select.select(rlist, wlist, [], timeout)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
KeyboardInterrupt
```


Panel Title
1
0.8
0.6
0.4
0.2
0
15:45
15:50
15:55
16:00
16:05
16:10
16:15
16:20
16:25
16:30
16:35
16:40
cpu1 msg_payload
cpu2 msg_payload
cpu3 msg_payload
CPU 3
0.65
0.6
0.55
0.5
0.45
0.4
0.35
msg_payload
CPU 2
msg_payload
CPU1
msg_payload

In part 1 of this lab we were supposed to use mosquito to connect to a MQTT broker and receive data about three CPU's on a specific port. Part 2 of the lab consisted of taking the data from part1 on putting it into a time series database. The time series database in question was influxdb. We used a python program to act as an MQTT subscriber client and as a client for the InfluxDB API. We needed to do this because influxDB only accepts data points in a JSON format, so we needed the python program to turn the info from the MQTT stream into a JSON format and put that info into the database. Part 3 of the lab consisted of using grafana to see all of the data we were putting into the influxDB. We had to log into grafana using the IP of our PI. Once in grafana we just had to make a proper query of the database and it was able to properly display the live information of the db.

We only ran into one major issue in the lab and that was properly querying the database in grafana. At first we were querying the data we transformed into a string in the python script. Since it was in string format, grafana was unable to properly show the graph and the live information. It was a simple fix; we just had to query for the proper information and it was up and running.

This lab included many aspects that would be considered IoT. In part one it was viewing live information from a MQTT stream, in part 2 it was taking the information from part 1 and transforming it to the proper usable format and putting it into a live time series based database, and finally part3 was using a different service to create graphs of our data.