LoRaWAN implementation for acoustic localization of coyotes

Team Coyote1

CONTENTS

Team members

Hyemin Lim
Chung-Ang University
Computer Science and Engineering

Jaehui Boo
Dankook University
Computer Engineering

Justin Anderson
Purdue University
Network Engineering Technology

Hyeongjun Kim
Daegu Catholic University
Computer Engineering

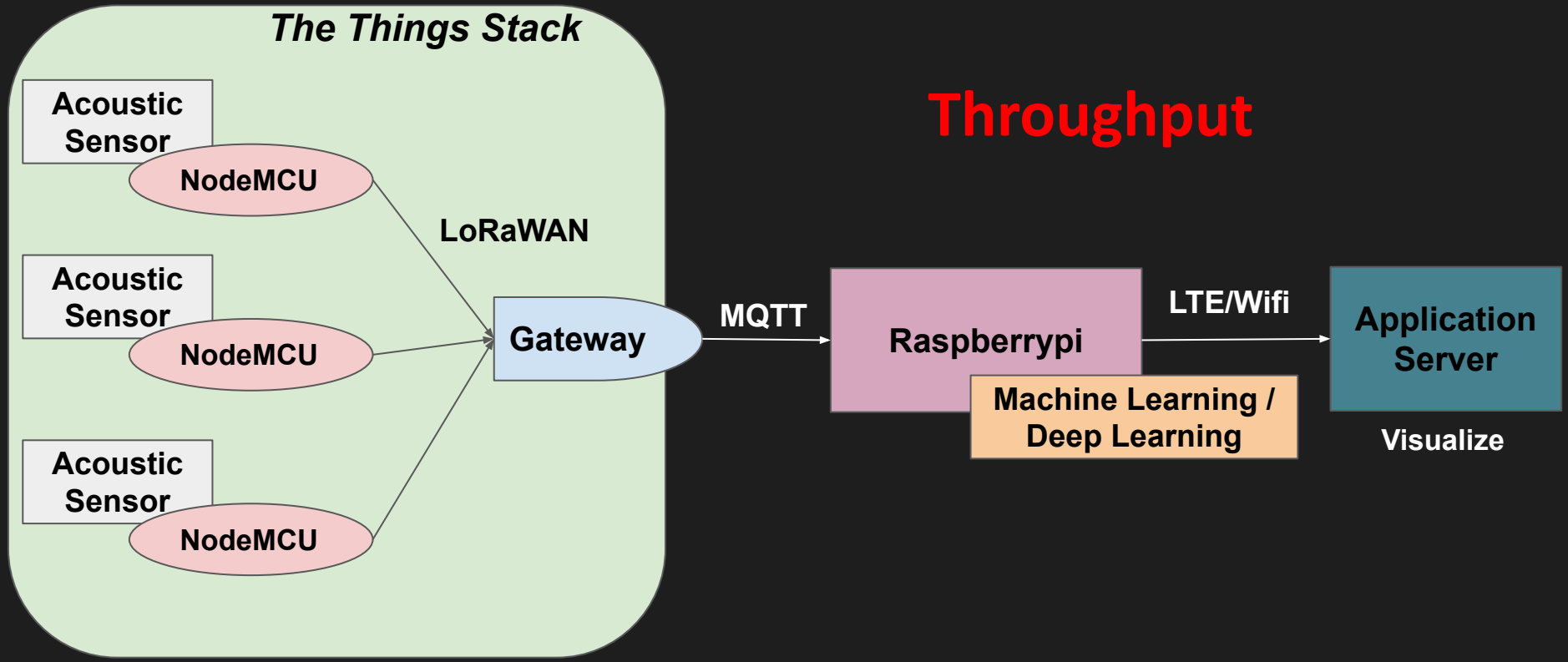
Nayoun Kim
Woosong University
Information Technology Convergence

Wei-Chieh Chin(Victor)
Purdue University
Computer & Information Technology

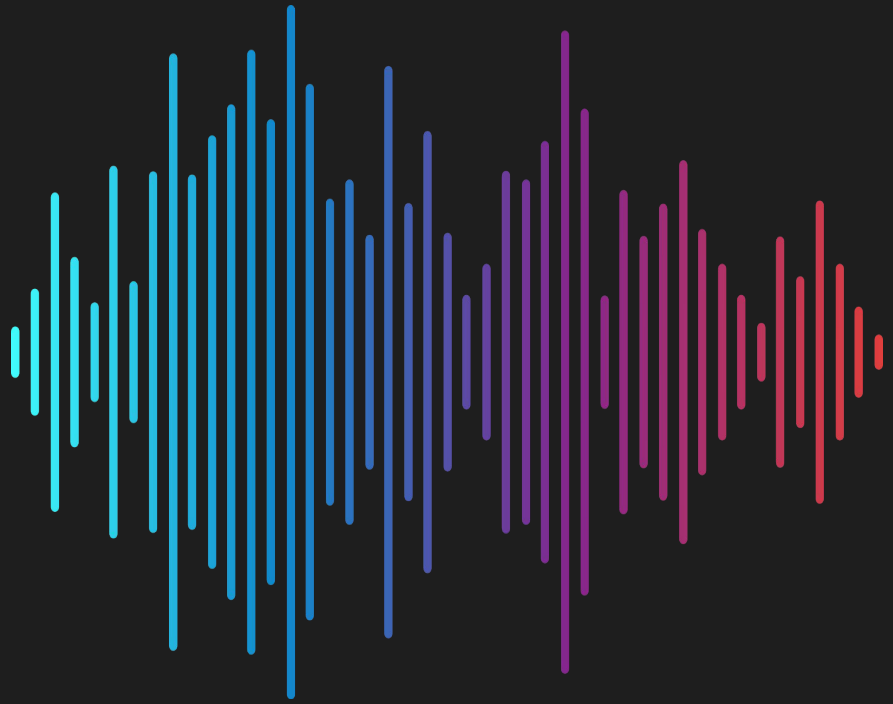
01

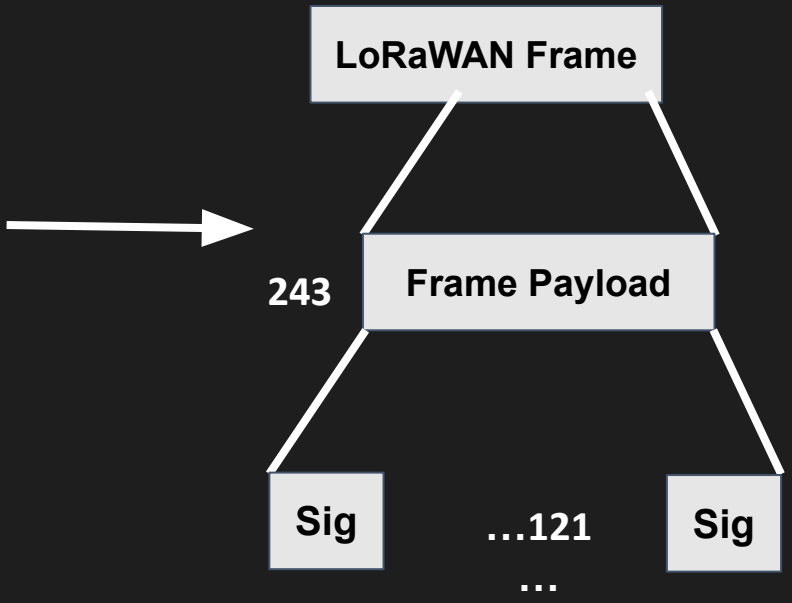
Introduction

1.1 What has changed since the Midterm



1.1 What has been changed since the Midterm

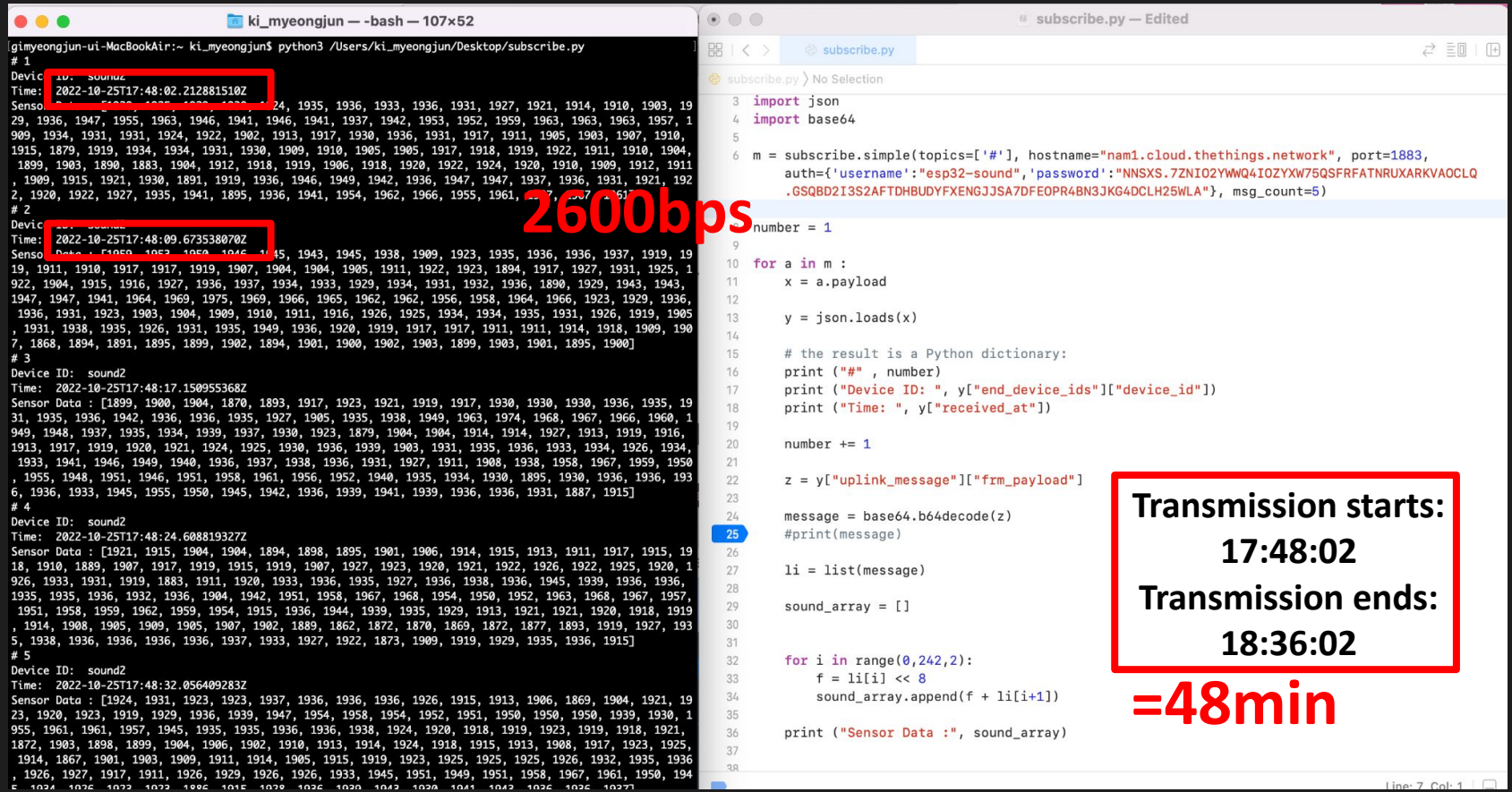




3 seconds of Audio * 16000 sampling rate
= 48000 audio signals

48000 / 121 = 400

1.1 What has been changed since the Midterm



1.1 What has been changed since the Midterm

[1] Hindawi., "Performance Evaluation of LoRaWAN Communication Scalaility in Large-Scale Wireless Sensor Networks", in *WCMC*, vol. 2018, June. 2018, Art no. 6730719

Table 2: Data rate codes defined by LoRaWAN.

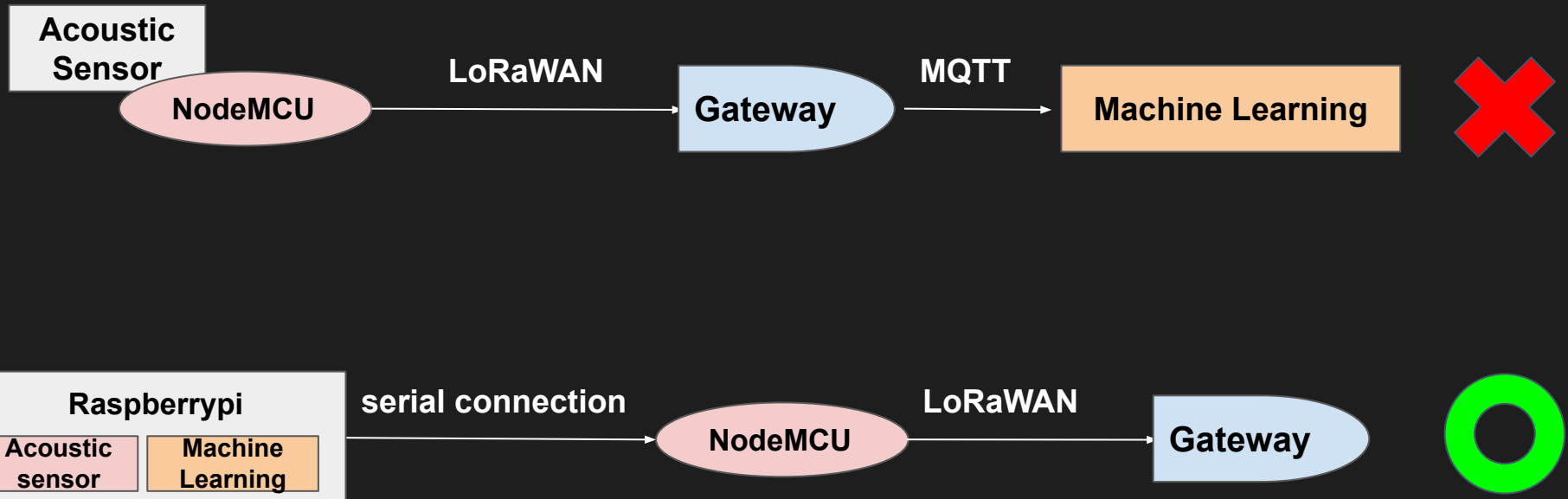
Data RateCode	Spreading Factor	Channel Width	Coding Rate	Data Rate
0	12	125 kHz	4/6	250 bps
1	11	125 kHz	4/6	440 bps
2	10	125 kHz	4/5	980 bps
3	9	125 kHz	4/5	1760 bps
4	8	125 kHz	4/5	3125 bps
5	7	125 kHz	4/5	5470 bps
6	7	250 kHz	4/5	11000 bps

7

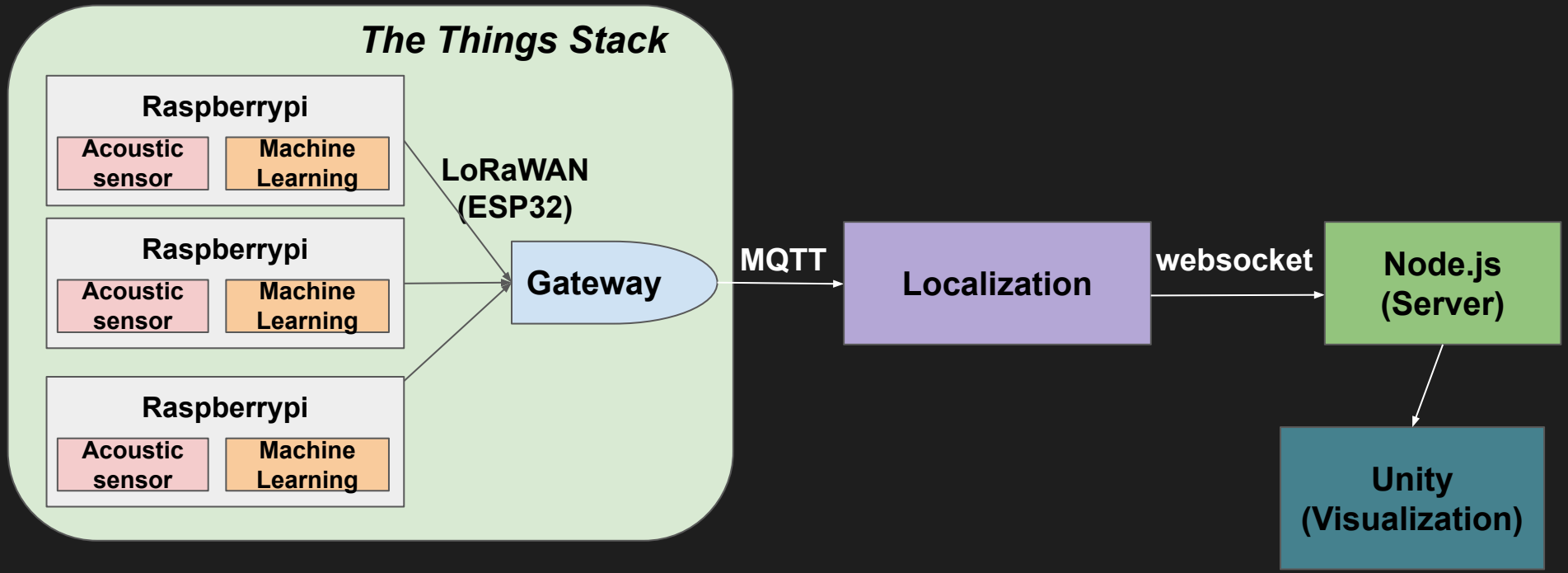
125kHz

4/5

1.1 What has been changed since the Midterm



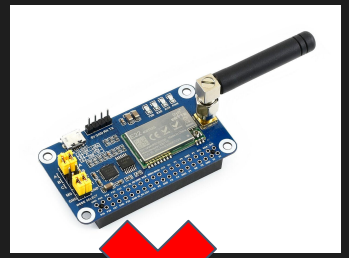
1.2 New Project Architecture

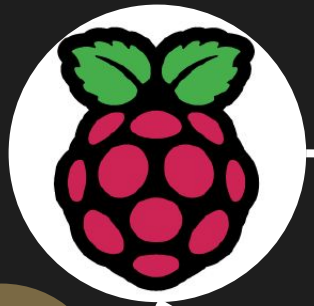


02

Implementation

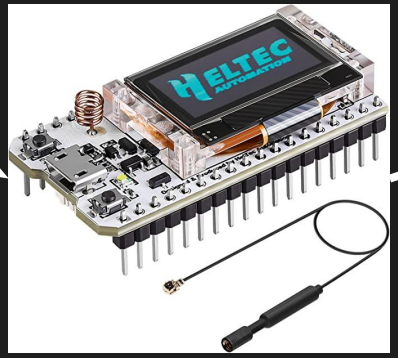
2.1 End node connection





THE THINGS STACK LoRaWAN®

USB



- Stable result
- easy to use
  (thanks to pyserial)

2.1 End node connection

```
ser = serial.Serial('/dev/ttyUSB_DEV1', 115200, timeout=1)
ser.reset_output_buffer()
```

```
output = output.strftime('%Y-%m-%d %H:%M:%S.%f')[14:26]
print(output)
output = output.encode('utf-8')
ser.write(output)
```

Raspberry Pi

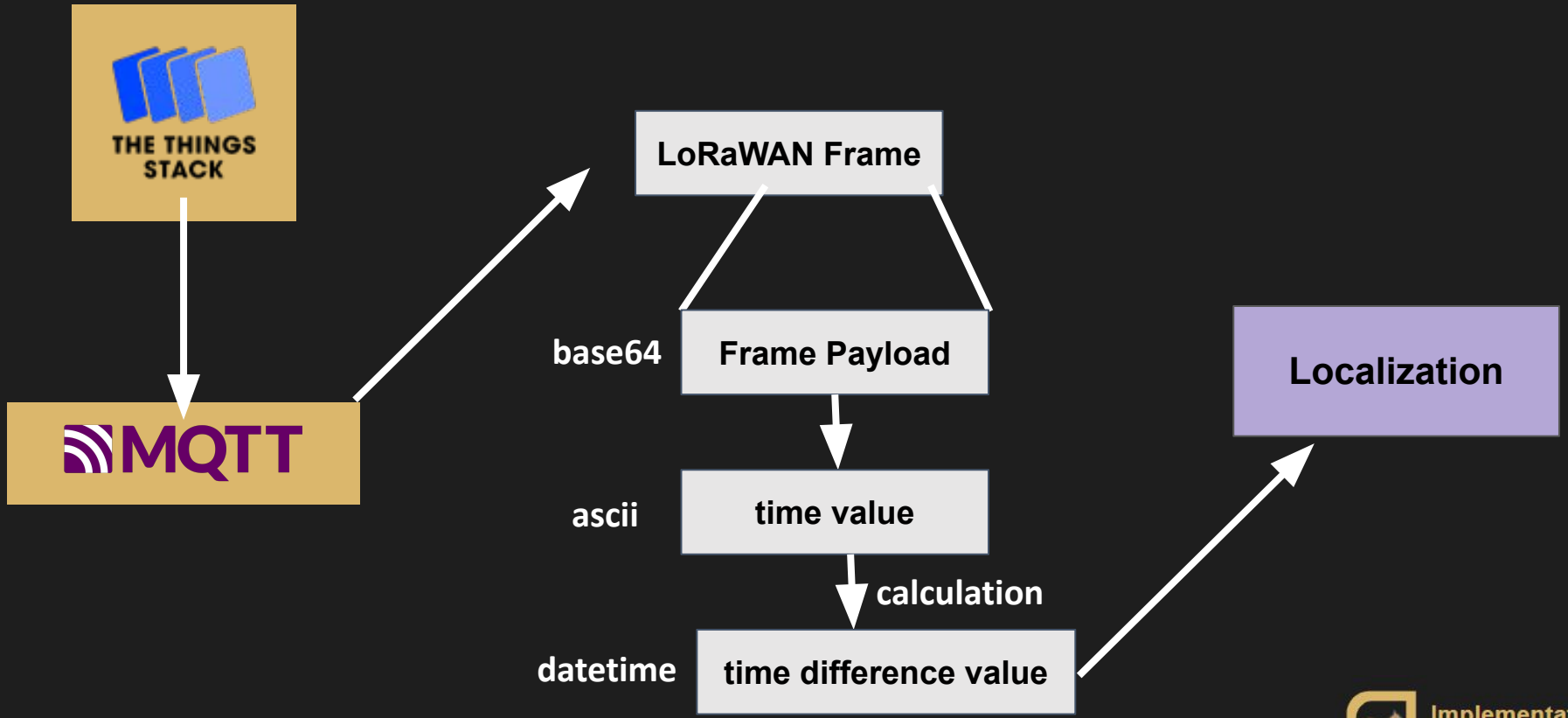
Timestamp

ESP 32

Payload(Timestamp)

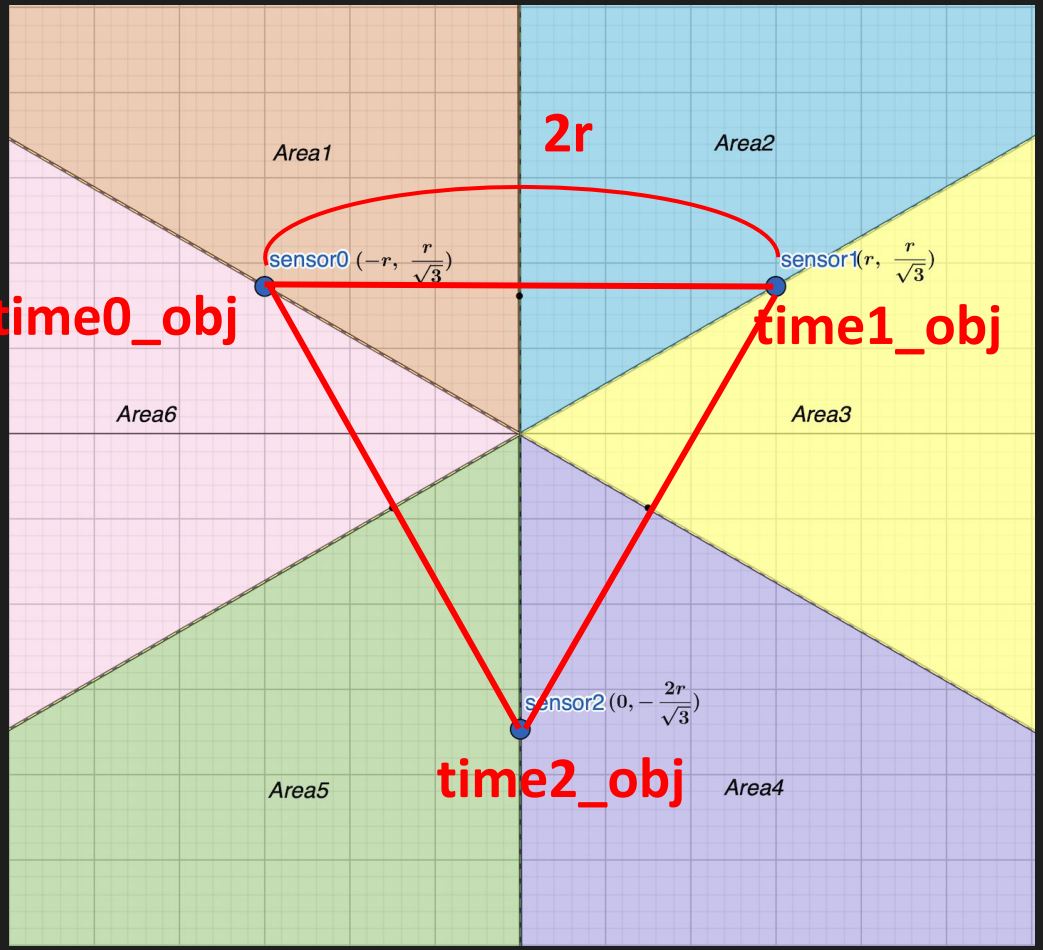
```
String data = Serial.readStringUntil('\n'); //read string from serial port sent by raspberry pi
data.getBytes(mydata, 30); //put into mydata array
do_send(&sendjob); //schedule next transmission
```

2.2 Raw frame data processing



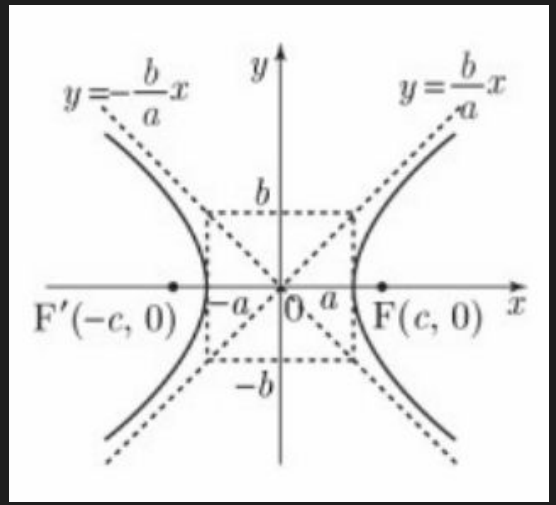
2.2 Localization

Unit: cm, ms



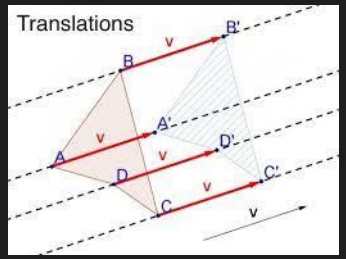
td0	td1	td2
v		
Area		

Hyperbolic function



$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

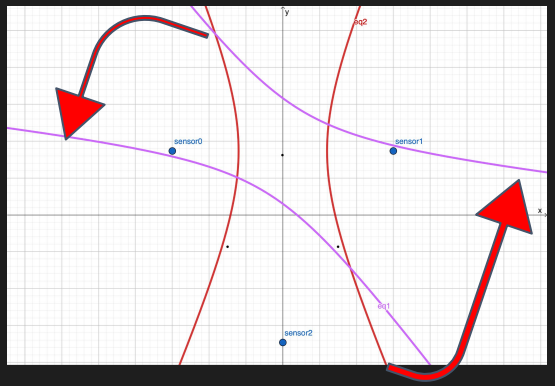
Parallel Translation



$$x' = x + m$$
$$y' = y + n$$

$$f(x - m, y - n) = 0$$

Rotation Transformation

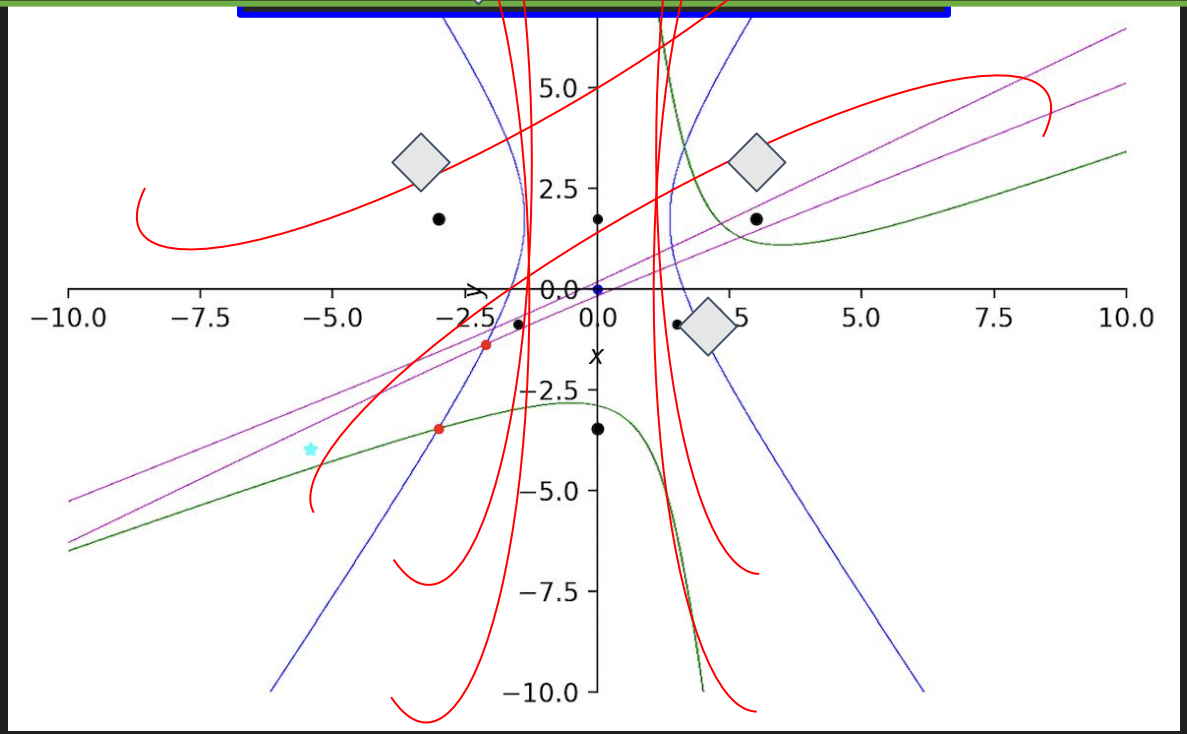


$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

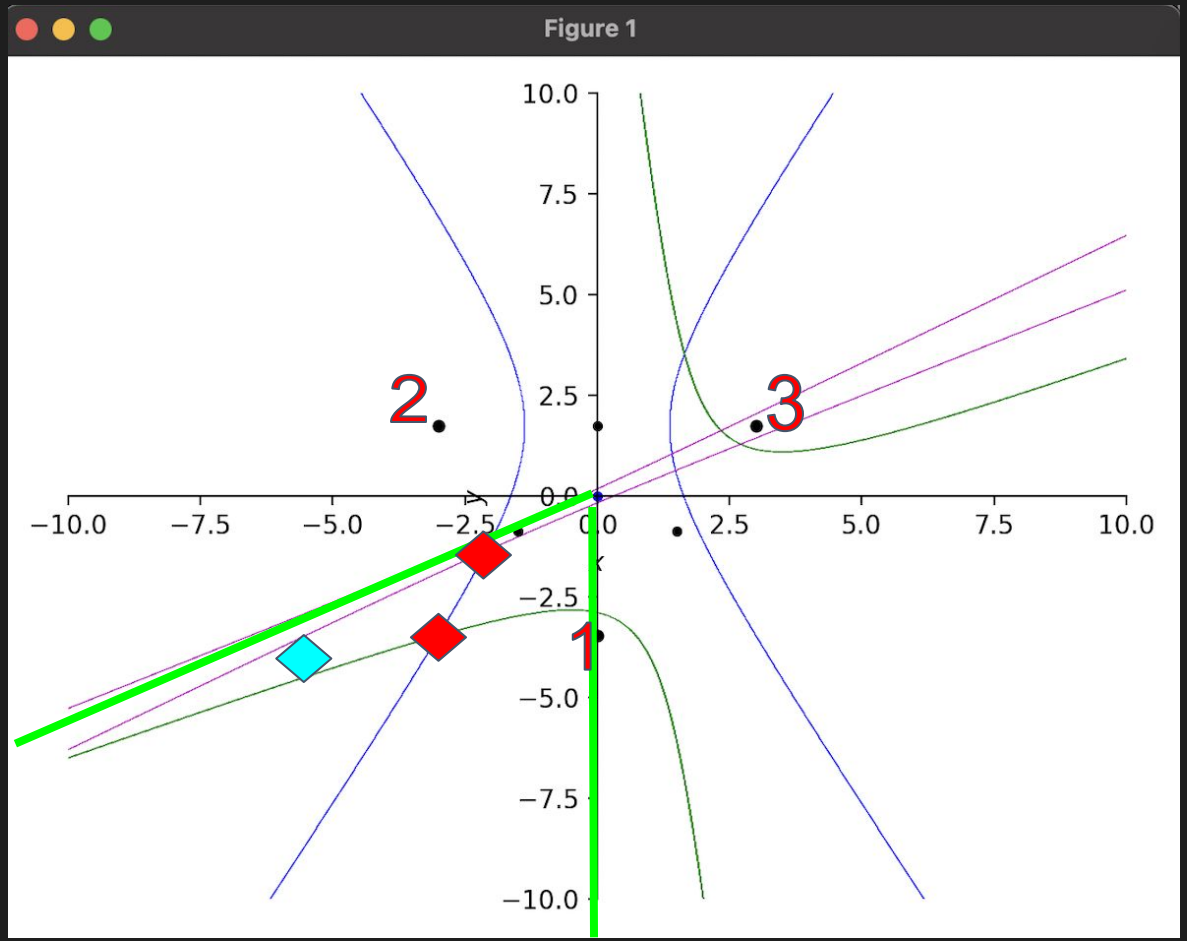
$$f(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta) = 0$$

2.2 Localization

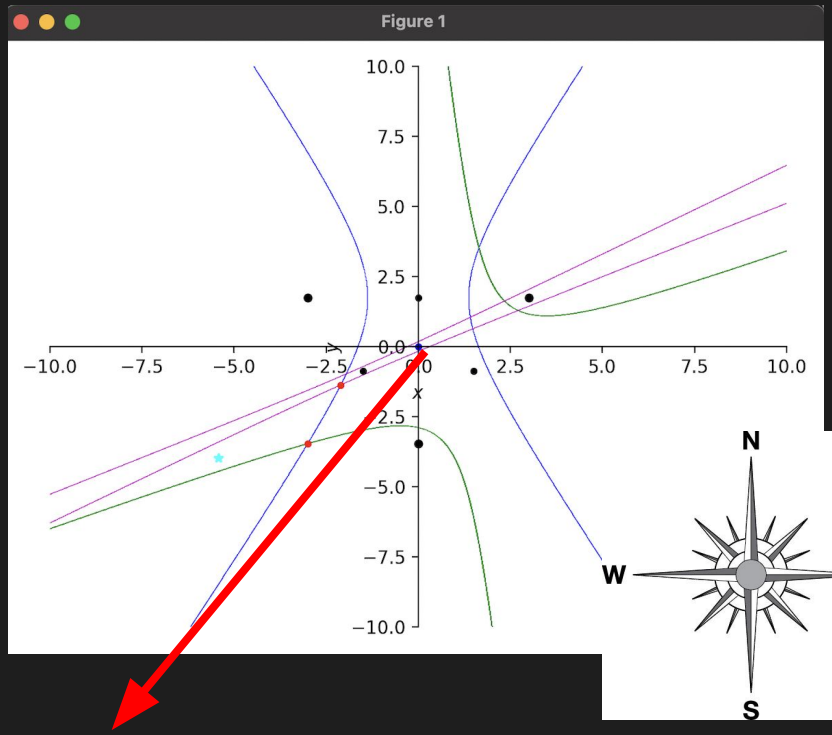
$$\frac{4\{(x+\frac{r}{2})\cdot\cos(-60)°+(y+\frac{r}{2\cdot\sqrt{3}})\cdot\sin(-60)°\}^2}{(v\cdot td_2)^2} - \frac{4\{-(x+\frac{r}{2})\cdot\sin(-60)°+(y+\frac{r}{2\cdot\sqrt{3}})\cdot\cos(-60)°\}^2}{4\cdot r^2-(v\cdot td_2)^2} = 1$$

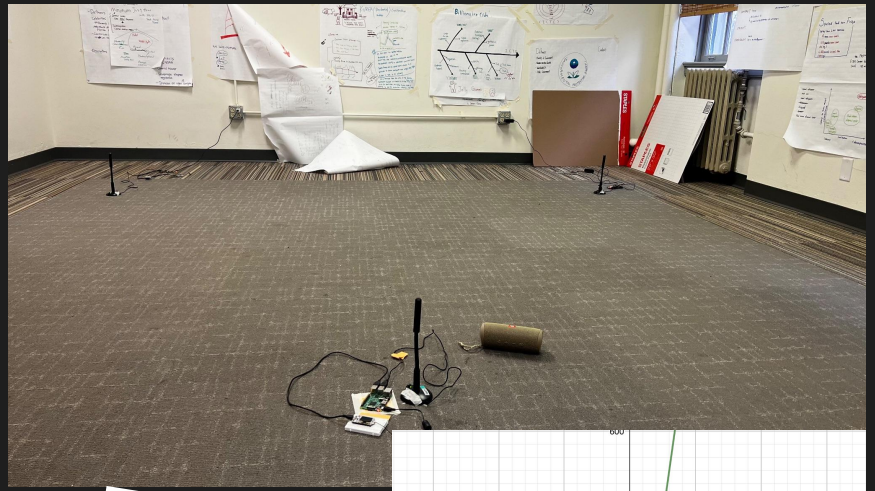


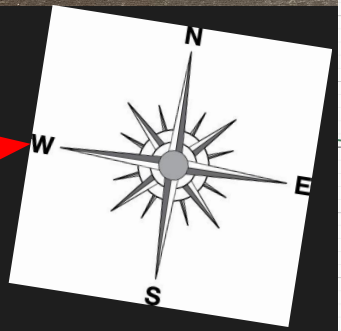
2.2 Localization

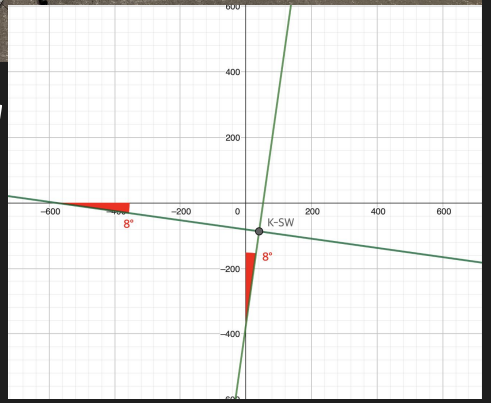


2.2 Localization







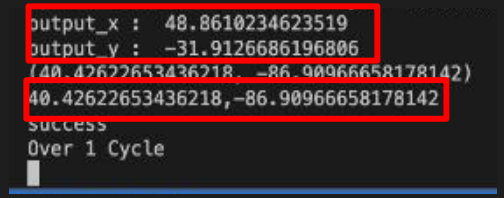


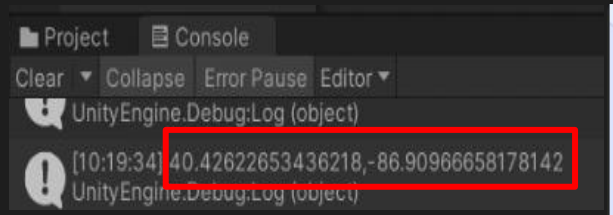
(m, n) (unit: cm)

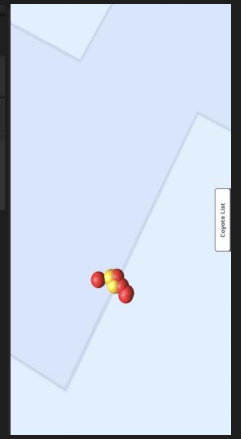
(40.426…, -86.909…) (unit: °)

Implementation

2.3 Visualization







Localization

coordinates

websocket

node

coyote detection history (recent 5)

sensor coordinates

http

real-time detection

websocket

unity

Mapbox SDK

mongoDB

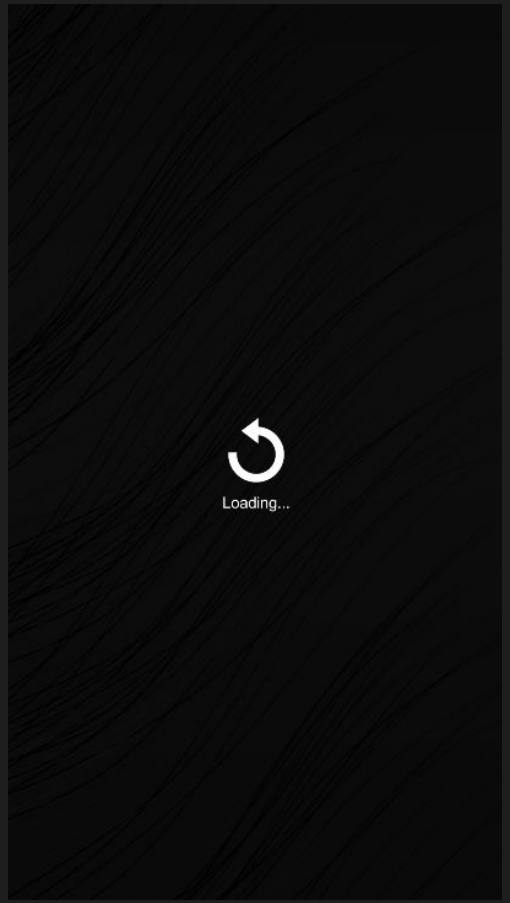
collections

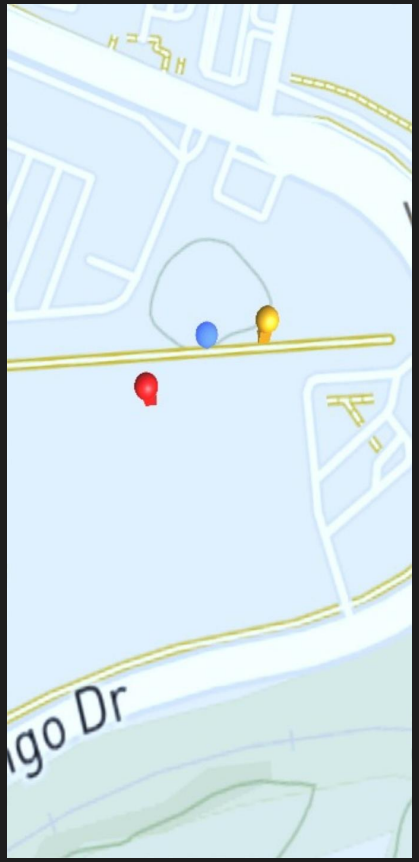
sensors

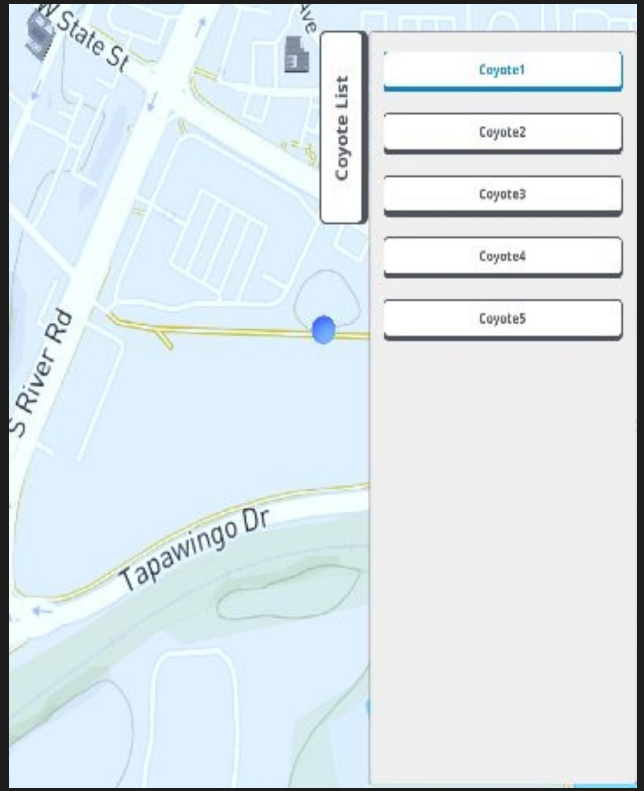
coyotes

Implementation

2.3 Visualization



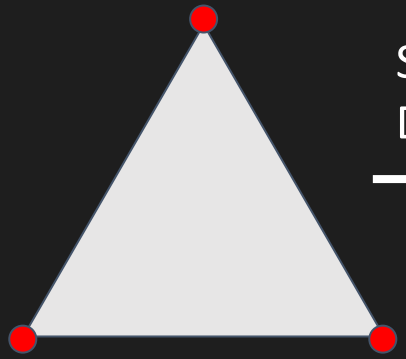




03

Experiment

3. Experiment



Sensors
(Each sensors distance is same)

Sound
Detect

ESP32

Packet

Gateway

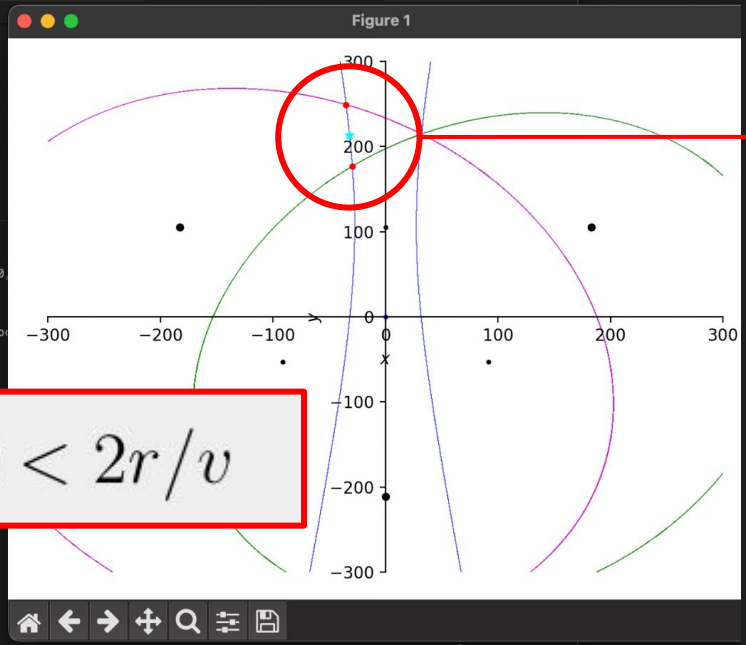
Localization

3. Experiment

```
ksw_localization > localization_combined_ver1.py > localization
219         r = 182.88 # cm
220         m = 0
221         n = 0
222         #r = float(input('Enter r: '))
223         #m = float(input('Enter m: '))
224         #n = float(input('Enter n: '))
225         '''
226         t0 = float(input('Enter t0: '))
227         t1 = float(input('Enter t1: '))
228         t2 = float(input('Enter t2: '))
229         area = int(input('Enter area: '))
```

문제 2 출력 디버그 콘솔 터미널 JUPYTER

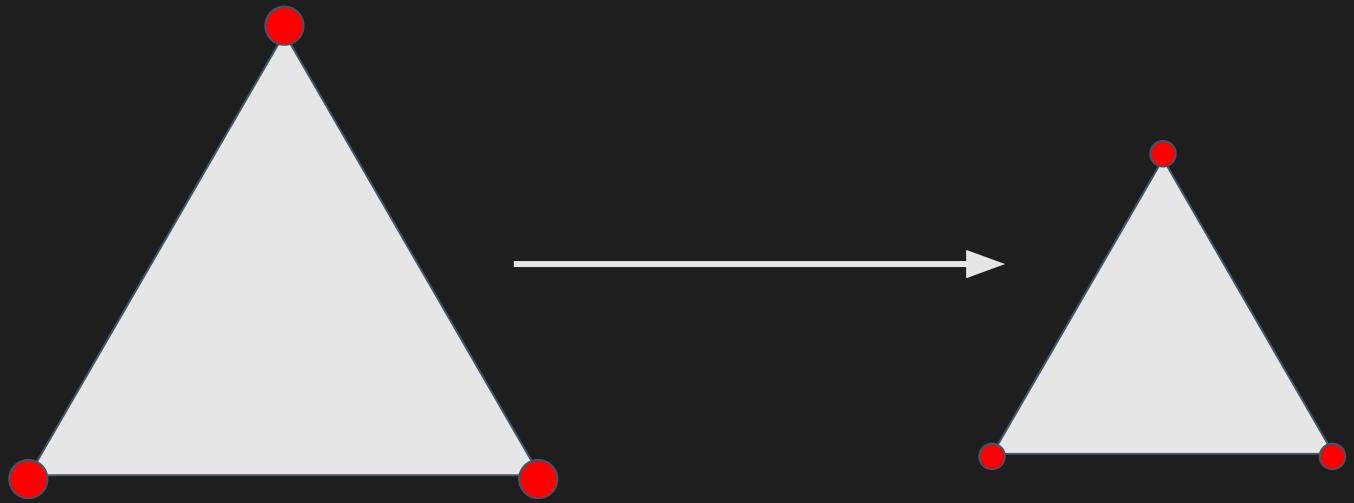
```
  File "/Users/jaehuiboo/Library/Python/3.9/lib/python/site-packages/paho/mqtt/client.py", line 1150,
    socklist = select.select(rlist, wlist, [], timeout)
KeyboardInterrupt

jaehuiboo@bujaehuiui-MacBookPro JaehuiBoo % python -u "/Users/jaehuiboo/Desktop/algorithm2022/ksw_lo
('sound3', datetime.datetime(1900, 1, 1, 0, 46, 9, 966409), 40.4203008430482, -86.90254211425781)
('sound1', datetime.datetime(1900, 1, 1, 0, 46, 9, 982375), 40.4203008430482, -86.90254211425781)
('sound2', datetime.datetime(1900, 1, 1, 0, 46, 10, 161152), 40.4203008430482, -86.90254211425781)
Receive Done
time0_obj
1900-01-01 00:46:09.966409
time1_obj
1900-01-01 00:46:09.982375
time2_obj
1900-01-01 00:46:10.161152
td0
0:00:00.015966
td1
0:00:00.178777
td2
0:00:00.194743
t0
1.5966
t1
17.8777
t2
19.4743
area is -> 1
result / len: 12
[{x: -69.1714832629622 - 7.17339988198475e-28*I, y: -313.884321713277 - 4.86074596514818e-28*I}, {x: -29.4362715040037 + 7.17340039915661e-28*I, y: 176.934699489415 + 4.86074596514832e-28*I}, {x: 31.9767176460774 + 7.17340037994243e-28*I, y: 214.646408552867 + 4.8607459651482e-28*I}, {x: 73.2865353179917 - 7.17340089715842e-28*I, y: -343.322997406844 - 4.86074596514834e-28*I}, {x: 31.9767176460777, y: 214.646408552868}, {x: 73.2865353179923, y: -343.322997406844}, {x: -20.1955153561235 - 452.7893640022*I, y: -646.05575407459 + 11.1742349379732*I}, {x: -20.1955153561235 + 452.7893640022*I, y: -646.05575407459 - 11.1742349379732*I}, {x: -77.3616628143088 - 9.6857943395625e-29*I, y: -372.211719324639 + 6.555548316618e-29*I}, {x: -34.9736662617896 + 7.99983131274333e-29*I, y: 249.266031150987 - 5.414449862280 1e-29*I}, {x: 31.9767176460774 - 8.99766665568336e-29*I, y: 214.646408552868 + 6.08980534337147e-29*I}, {x: 73.2865353179917 + 1.06836296827102e-28*I, y: -343.322997406845 - 7.23090367376146e-29*I}]
12
[{'x': -69.1714832629622, 'y': -313.884321713277}, {'x': -29.4362715040037, 'y': 176.934699489415}, {'x': 31.9767176460774, 'y': 214.646408552867}, {'x': 73.2865353179917, 'y': -343.322997406844}, {'x': 31.9767176460777, 'y': 214.646408552868}, {'x': 73.2865353179923, 'y': -343.322997406844}, {'x': -20.1955153561235, 'y': -646.05575407459}, {'x': -20.1955153561235, 'y': -646.05575407459}, {'x': -77.3616628143088, 'y': -372.211719324639}, {'x': -34.9736662617896, 'y': 249.266031150987}, {'x': 31.9767176460774, 'y': 214.646408552868}, {'x': 73.2865353179917, 'y': -343.322997406845}]
result2 / len: 2
[{'x': -29.4362715040037, 'y': 176.934699489415}, {'x': -34.9736662617896, 'y': 249.266031150987}]
```



$$\Delta t < 2r/v$$

Wrong Result

3. Experiment



Sensor Distance: 144ft

Sensor Distance: 60ft

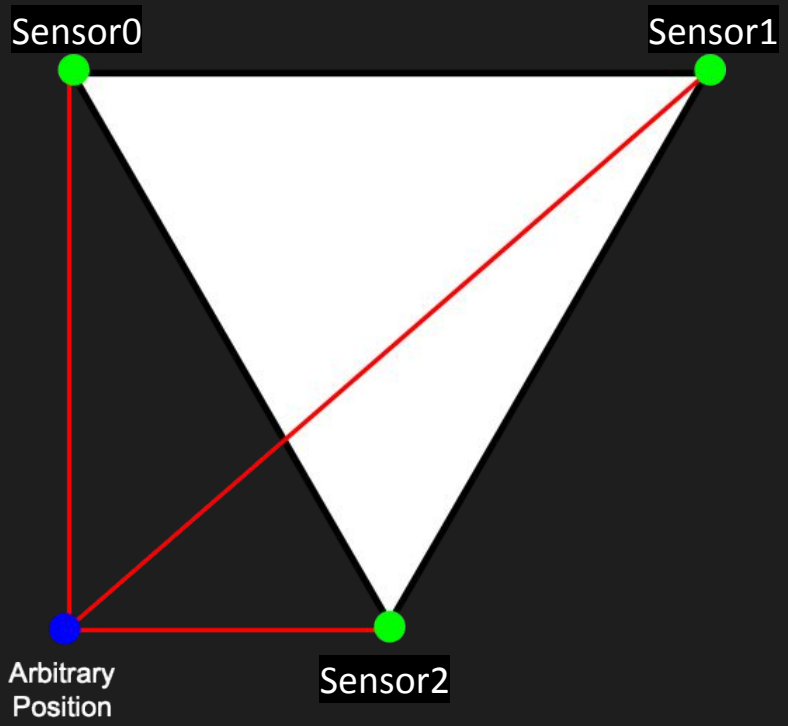
3. Experiment

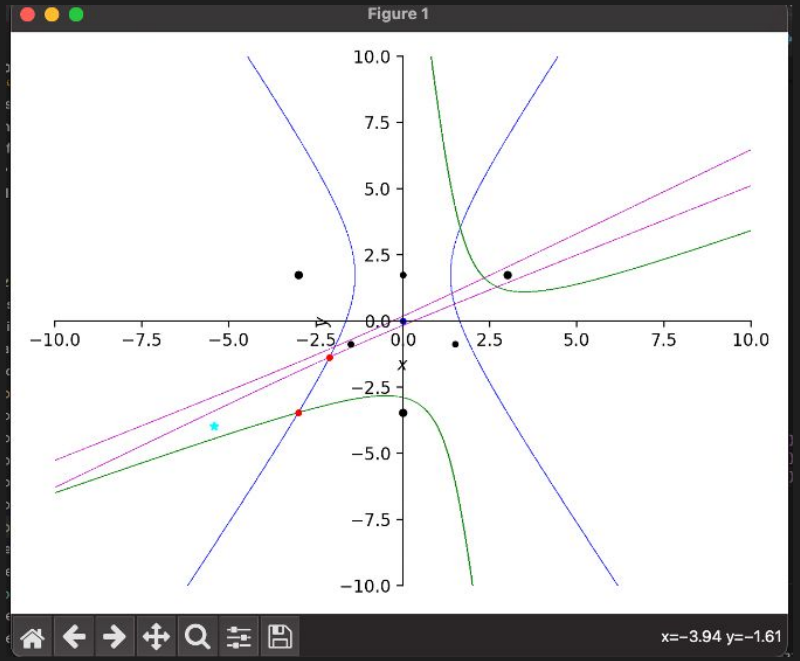
```
#real code
for arr in record_voice:
    if(arr[0]>0.057):
        output = datetime.now()
        output = output.strftime('%Y-%m-%d %H:%M:%S.%f')[14:26]
        print(output)
        output = output.encode('utf-8')
        ser.write(output)
        time.sleep(1)
        break
```

```
"f_port": 1,
"frm_payload": "NDk6NTMuMzY4NTQ2AAAAAAAAAAAAAAAAAAAAAAAA==",
```

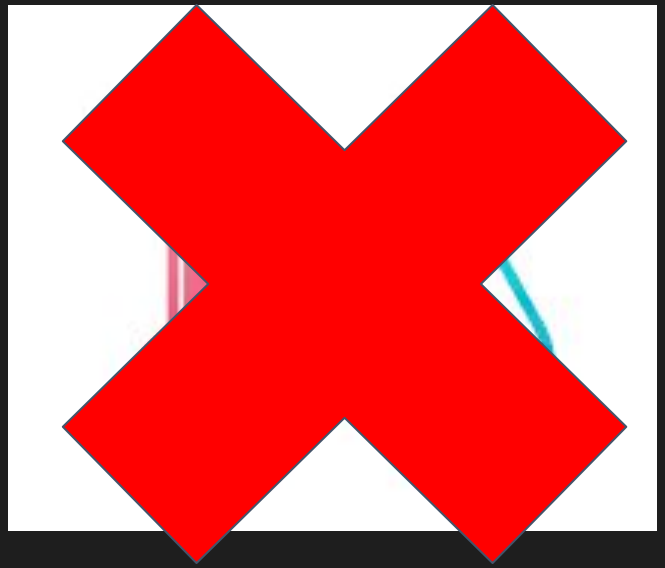
↑ 13:50:03	sound2	Forward uplink data message	DevAddr:	26 0C FD 82	Payload:	35 30 3A 30 32 2E 33 34 …	FPort:	1
↑ 13:49:54	sound2	Forward uplink data message	DevAddr:	26 0C FD 82	Payload:	34 39 3A 35 33 2E 33 36 …	FPort:	1

3. Experiment



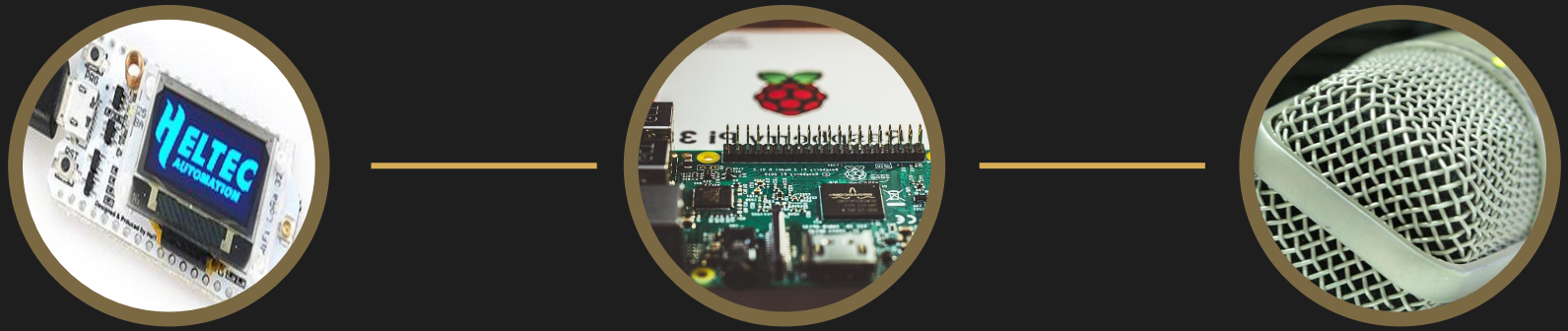


3. Experiment



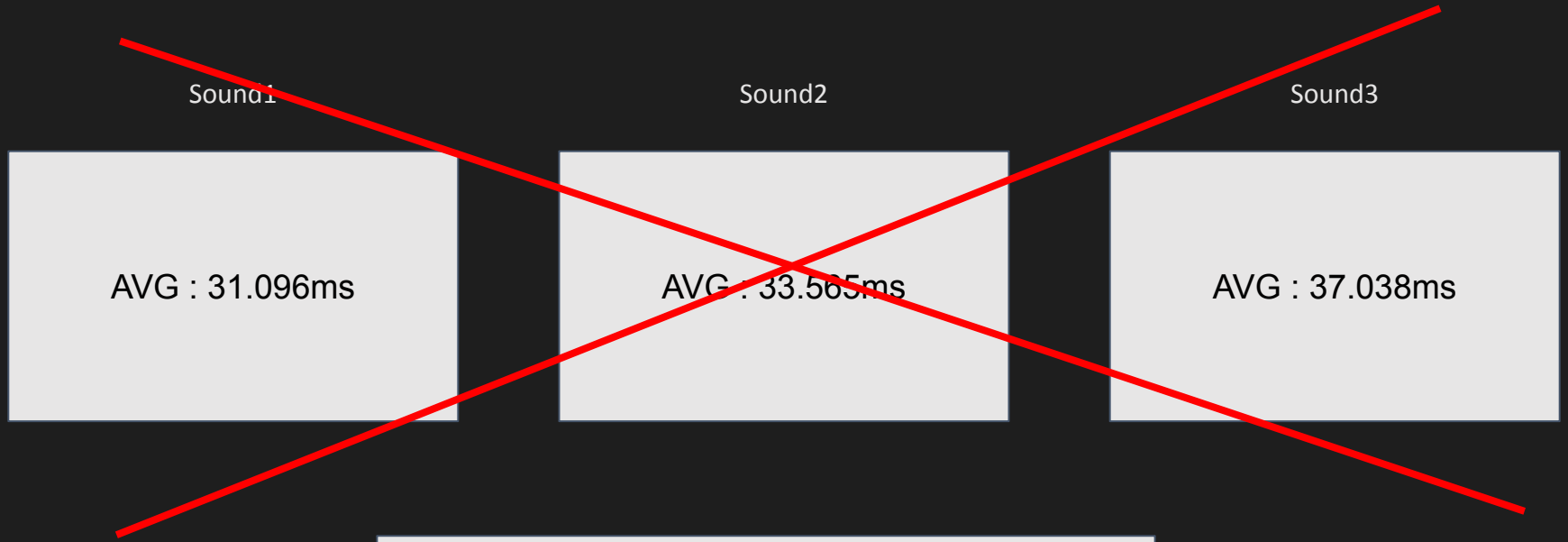
3. Experiment

Time Delay?



1m Length usb cable took 10ns to communicate

3. Experiment



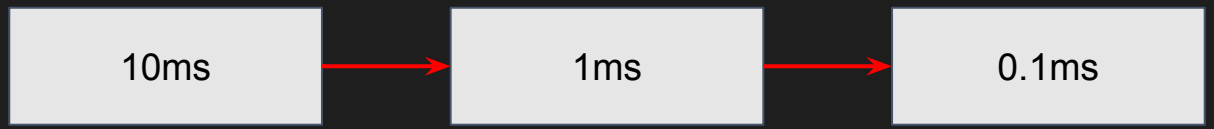
Biggest problem is recognition accuracy

3. Experiment

```
for arr in record_voice:
    if(arr[0]>0.057):
```

3. Experiment

```
fs = 44100
second = 0.001 Frame Value
i = 0
# FILE_PATH = '/home/coyote/code'

ser = serial.Serial('/dev/ttyUSB_DEV1', 115200, timeout=1)
ser.reset_output_buffer()

while(True):
    arr_high =[]
    # FILE_NAME = 'REC_FILE'+str(i)
    record_voice = sd.rec( int(second*fs), samplerate=fs, channels=2 ) # per 0.001 sec
    sd.wait()
```



3. Experiment

```
fs = 44100
second = 0.001
i = 0
# FILE_PATH = '/home/coyote/code'

ser = serial.Serial('/dev/ttyUSB_DEV1', 115200, timeout=1)
ser.reset_outpu

while(True):
    arr_high =[
    # FILE_NAM
    record_voi
    sd.wait()
```

Sampling Rate x 2

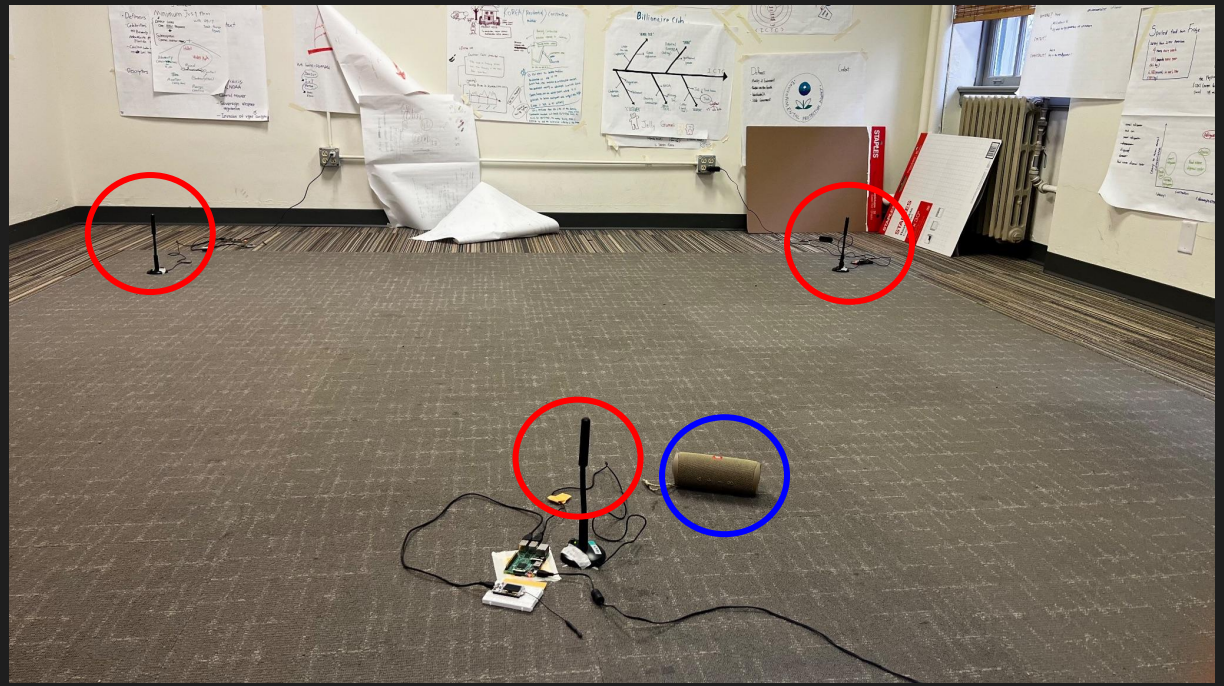
```
fs = 88200
second = 0.001
i = 0
# FILE_PATH = '/home/coyote/code'

ser = serial.Serial('/dev/ttyUSB_DEV1', 115200, timeout=1)
ser.reset_output_buffer()

while(True):
    arr_high =[]
    # FILE_NAME = 'REC_FILE'+str(i)
    record_voice = sd.rec( int(second*fs), samplerate=fs, channels=2 ) # per 0.001 sec
    sd.wait()
```

3. Experiment

```
# time difference
td0 = td0.total_seconds()  *1000 / 10
td1 = td1.total_seconds()  *1000 / 10
td2 = td2.total_seconds()  *1000 / 10
```

Calibration Value

3. Experiment



acoustic sensor

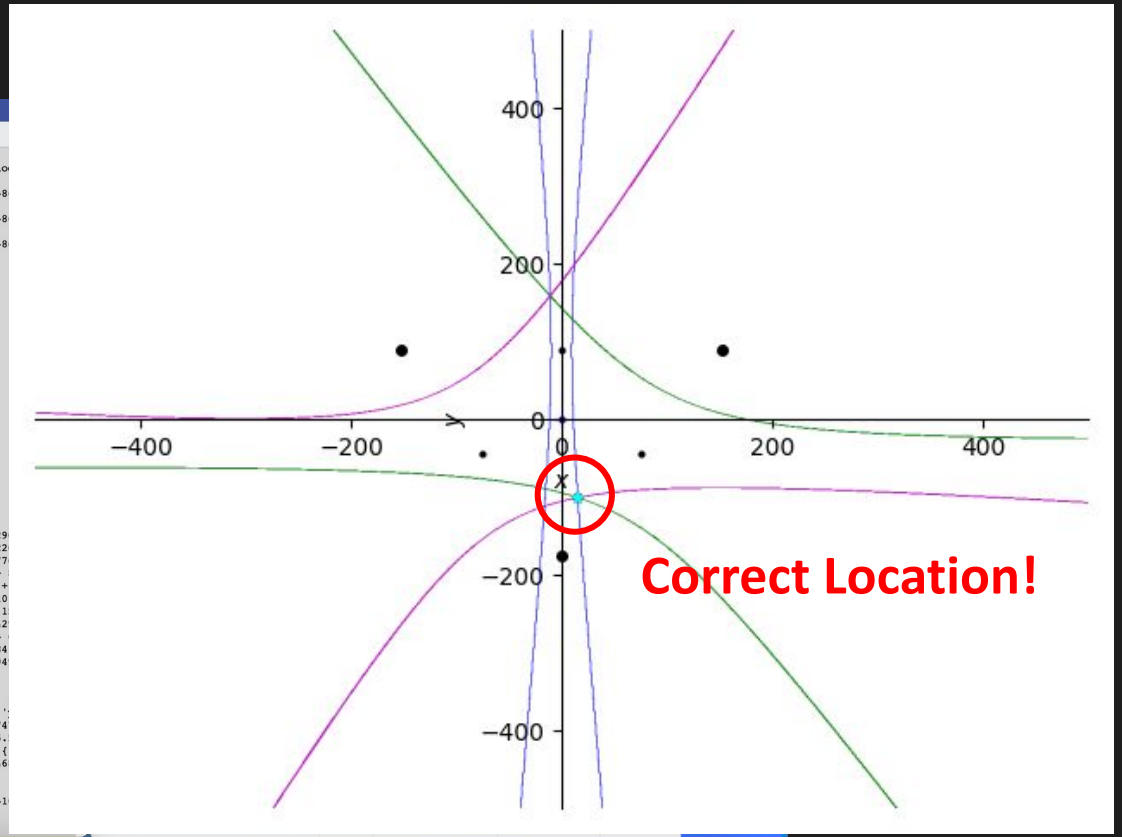
Sound Source

3. Experiment

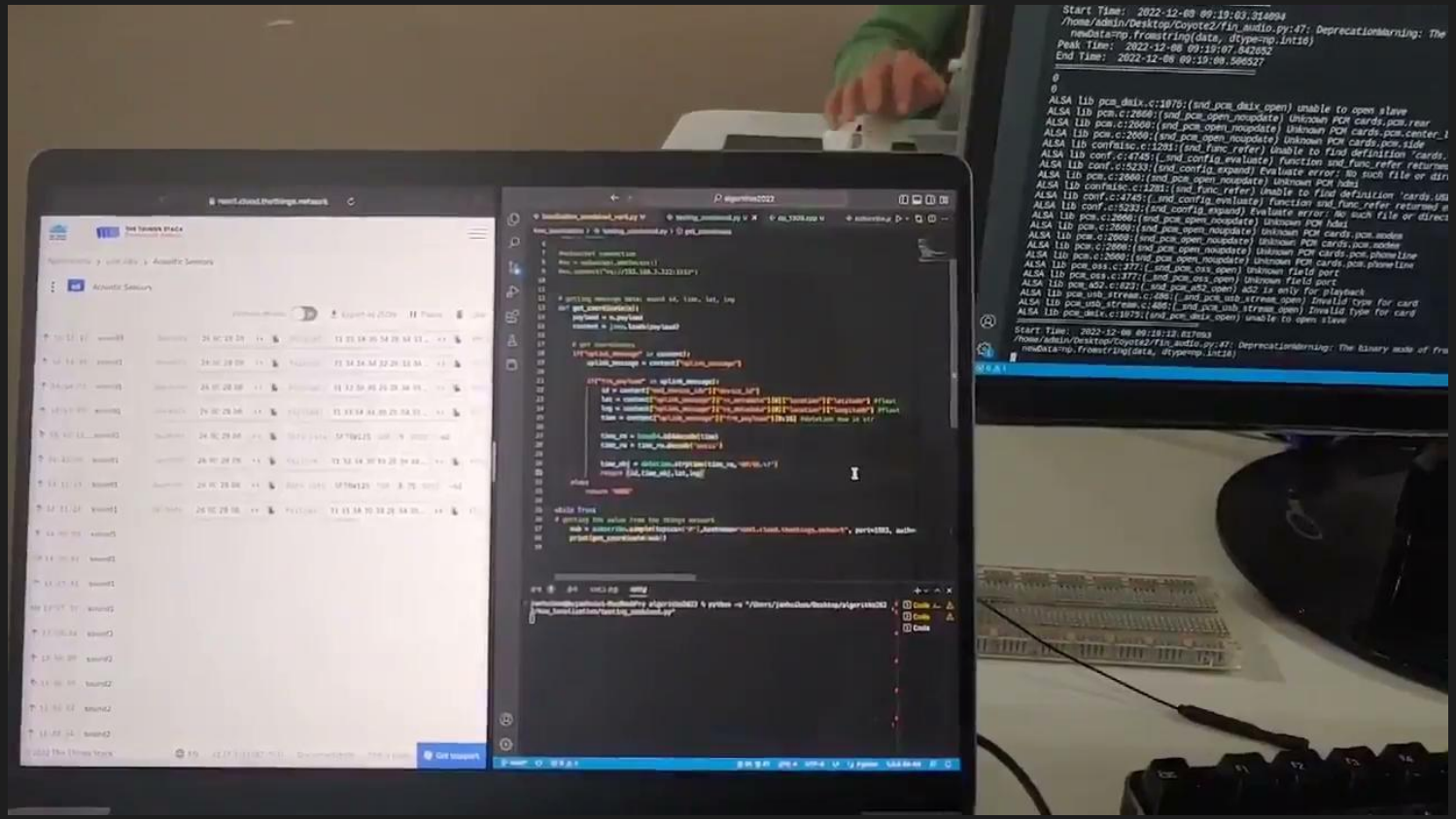
Python

ki_myeongjun — localization_combined_ver4.py — 91×54

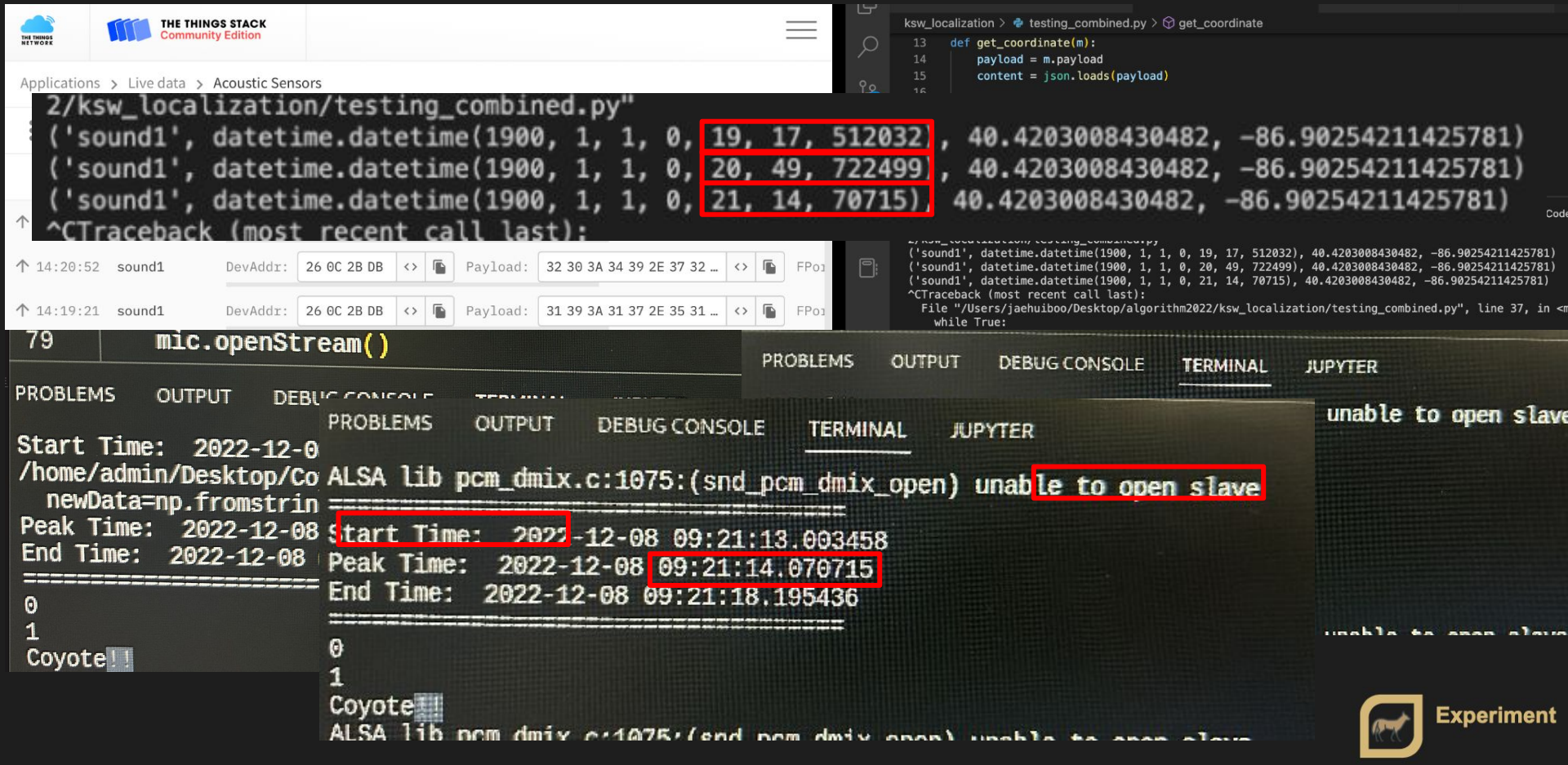
```
[ki_myeongjun@gimyeongjun-ui-MacBookAir ~ % python3 /Users/ki_myeongjun/Desktop/lo
_combined_ver4.py
('sound2', datetime.datetime(1900, 1, 1, 0, 10, 35, 822201), 40.4203008430482, -8
425781)
('sound1', datetime.datetime(1900, 1, 1, 0, 10, 35, 867918), 40.4203008430482, -8
425781)
('sound3', datetime.datetime(1900, 1, 1, 0, 10, 35, 873602), 40.4203008430482, -8
425781)
Receive Done
time0_obj
1900-01-01 00:10:35.873602
time1_obj
1900-01-01 00:10:35.867918
time2_obj
1900-01-01 00:10:35.822201
td0
0:00:00.005684
td1
0:00:00.045717
td2
0:00:00.051401
diff_td0
0.5684
diff_td1
4.5717
diff_td2
5.1401
area is -> 4
result / len: 12
[{x: -14.9983971765955 - 5.3199804704328e-29*I, y: -89.8506681555633 - 9.32584829
*I}, {x: -10.7396993053258 - 1.67897648827015e-29*I, y: 158.31170473219 - 2.94322
-29*I}, {x: 10.0909025388922 + 2.15441673702121e-29*I, y: 128.677774972911 + 3.77
75e-29*I}, {x: 15.5497636794982 + 4.84454021998331e-29*I, y: -101.026648273991 +
021934e-29*I}, {x: -28.9334674732088 - 302.978562327358*I, y: -51.3898378471993 +
743760*I}, {x: -28.9334674732088 + 302.978562327358*I, y: -51.3898378471993 - 10
7603*I}, {x: -10.7396993053258 + 0.e-24*I, y: 158.311704732191 + 0.e-19*I}, {x: 1
94983 + 0.e-25*I, y: -101.026648273991 + 0.e-19*I}, {x: -16.1113630215503 - 1.842
2e-29*I, y: -112.148753628244 + 4.30128438331022e-29*I}, {x: -10.7396993053258 -
407536e-30*I, y: 158.311704732191 + 1.43120495327257e-29*I}, {x: 12.0286252808784
288623657e-30*I, y: 197.937784514687 - 1.08148984248826e-29*I}, {x: 15.5497636794
275419353405e-29*I, y: -101.026648273991 - 4.65099949409453e-29*I}]
12
[{'x': -14.9983971765955, 'y': -89.8506681555633}, {'x': -10.7396993053258, 'y':
73219}, {'x': 10.0909025388922, 'y': 128.677774972911}, {'x': 15.5497636794982, '
26648273991}, {'x': -28.9334674732088, 'y': -51.3898378471993}, {'x': -28.9334674
': -51.3898378471993}, {'x': -10.7396993053258, 'y': 158.311704732191}, {'x': 15.
983, 'y': -101.026648273991}, {'x': -16.1113630215503, 'y': -112.148753628244}, {
396993053258, 'y': 158.311704732191}, {'x': 12.0286252808784, 'y': 197.9377845146
 15.5497636794982, 'y': -101.026648273991}]
result2 / len: 3
[{'x': 15.5497636794982, 'y': -101.026648273991}, {'x': 15.5497636794983, 'y': -1
73991}, {'x': 15.5497636794982, 'y': -101.026648273991}]
```



3. Experiment



3. Experiment



04

Future works

4. Future Works

```
# time difference
td0 = td0.total_seconds()  *1000 / 10
td1 = td1.total_seconds()  *1000 / 10
td2 = td2.total_seconds()  *1000 / 10
```

Calibration Value should be change

[References]

[1] Wave Share. “SX1262 868M LoRa HAT” Wave Share. Accessed: Nov, 2, 2022. [Online]. Available: https://www.waveshare.com/wiki/SX1262_868M_LoRa_HAT

[2] Spotpear. “Raspberry Pi SX1262 868M LoRa HAT User Guide” Spotpear. Accessed: Nov, 2, 2022. [Online]. Available: https://www.spotpear.com/index/study/detail/id/244.html

[3] sb Components. “Getting started with LoRa™ Hat for raspberry pi” sb Components. Dec, 10, 2021. [Online]. Available: https://shop.sb-components.co.uk/blogs/posts/getting-started-with-lora-hat-for-raspberry-pi

[4] Jac. K. “Gateway SX 1262” The Things Network Forum. Jun 13, 2021. [Online]. Available: https://www.thethingsnetwork.org/forum/t/gateway-sx-1262/48750

[5] viveknayyar007 “How To Change Default Internet Connection Sharing IP Address Range” Tom's Hardware Forum. Dec, 12, 2013. [Online]. Available: https://forums.tomshardware.com/faq/how-to-change-default-internet-connection-sharing-ip-address-range.1606758/

[6] Lydia. C. “Connecting a Raspberry Pi to a Laptop Display” Atomic Object. Jun, 9, 2019. [Online]. Available: https://spin.atomicobject.com/2019/06/09/raspberry-pi-laptop-display/

[7] Dexter Industries “Connecting to Raspberry Pi without a monitor for Beginners” Dexter Industries. Accessed: Nov, 2, 2022. [Online]. Available: https://www.dexterindustries.com/howto/connecting-raspberry-pi-without-monitor-beginners/

[8] jurosofish, “multilateration” Github.com, Nov, 4, 2022. [Online]. Available: https://github.com/jurasofish/multilateration/pulse

[9] automaticaddison. “Two Way Communication Between Raspberry Pi and Arduino” Automatic Addison. Jul, 6, 2020. [Online]. Available: https://automaticaddison.com/2-way-communication-between-raspberry-pi-and-arduino/

[References]

[10] Hopkins. J. “Understanding the difference between UART and USB” Totalphase. Accessed: Nov, 11, 2022. [Online]. Available: https://www.totalphase.com/blog/2022/01/understanding-differences-between-uart-and-usb/

[11] Press Start. “Unity - Mobile Panning with a Perspective Camera” Youtube. Nov, 9, 2018. [Online]. Available: https://www.youtube.com/watch?v=4_HUlAFlxwU

[12] Google. “Maps SDK for Unity Key Concepts” Googles map platform. Accessed: Nov, 9, 2022. [Online]. Available: https://developers.google.com/maps/documentation/gaming/concepts_musk

[13] Kristopher C. “A Guide to Using Google’s Maps SDK for Unity 3D - Part 1” Medium. Apr, 19, 2021. [Online]. Available: https://medium.com/everdevs-community/a-guide-to-using-googles-new-maps-sdk-for-unity-3d-ed1d68b2305e

[14] Ahmed. S. “WebSocket Client & Server (Unity & NodeJS)” Medium. Dec 6, 2020. [Online]. Available: https://medium.com/unity-nodejs/websocket-client-server-unity-nodejs-e33604c6a006

[15] SymPy Development Team. (2022). Accessed: Aug. 22, 2022. [Online]. Available: https://docs.sympy.org/latest/tutorials/intro-tutorial/index.html

[16] Python. “datetime” python software foundation. Accessed: Nov, 16, 2022 [online]. Available: https://docs.python.org/3/library/datetime.html

[17] mapbox. “Maps SDK for Unity” mapbox docs Guides. Accessed: Nov, 14, 2022 [online]. Available: https://docs.mapbox.com/unity/maps/guides/

[18] Mapbox. “Build a Unity 3D game Introduction to PocketDroids GO and the Mapbox SDK for Unity” Youtube. Jun, 14, 2018 [online]. Available: https://www.youtube.com/watch?v=RhG1kfDBhgM

Q&A

Thank you