



LoRaWAN implementation for acoustic localization of coyotes

Team Coyote1



CONTENTS



01. Introduction



02. Implementation



03. Experiment



04. Future works

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



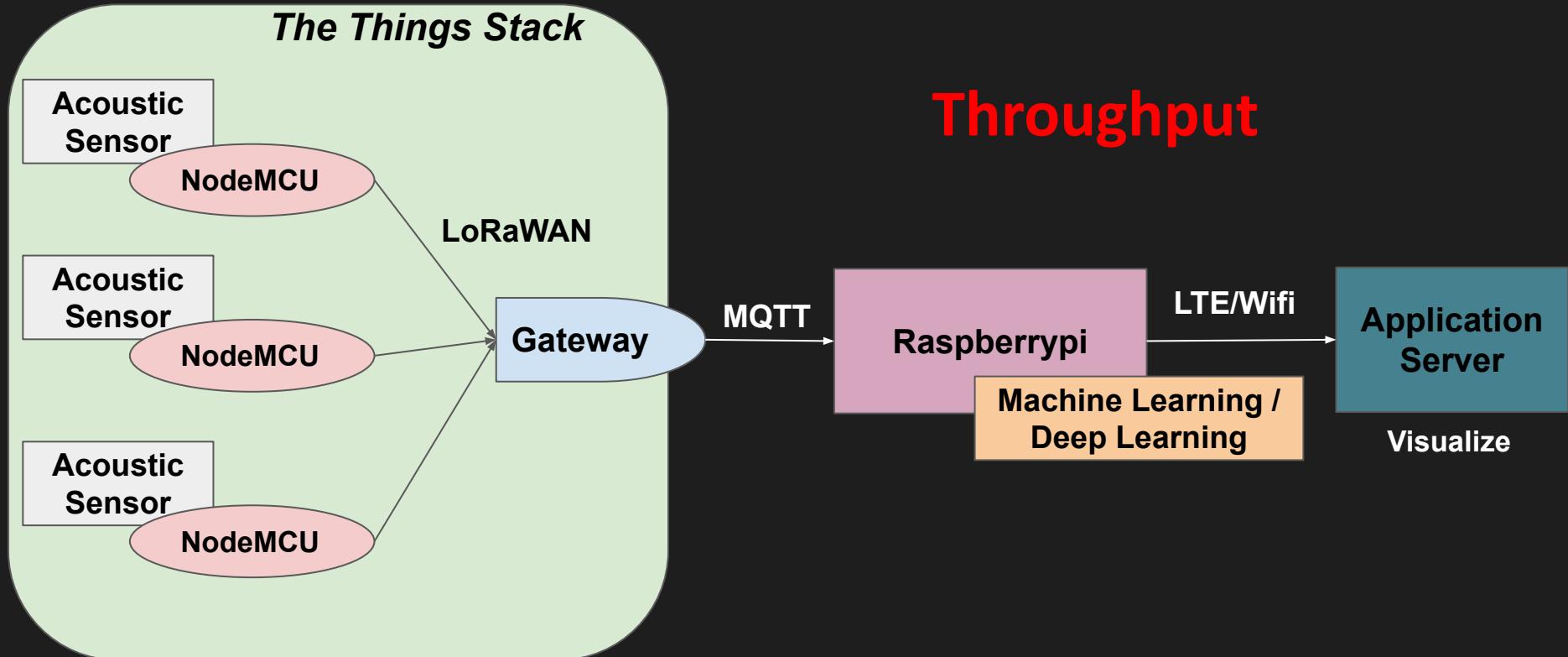


Introduction

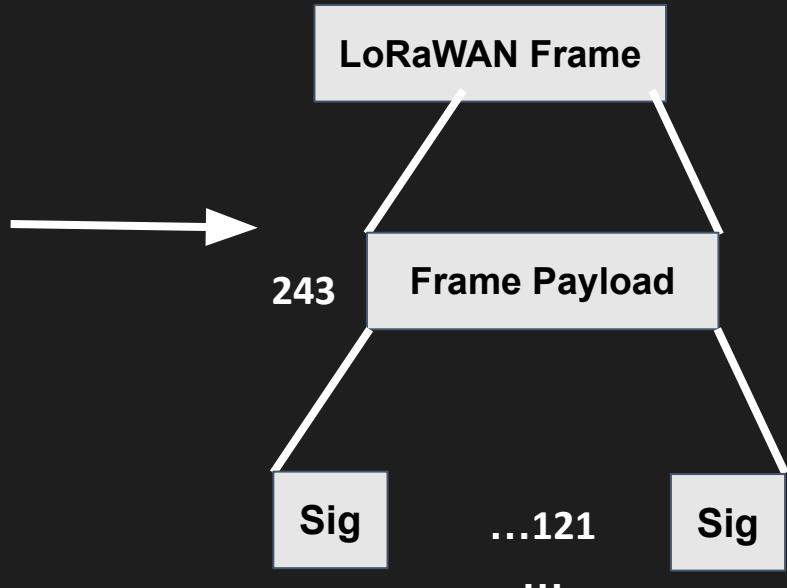
01

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

```
subscribe.py — Edited
subscribe.py No Selection
3 import json
4 import base64
5
6 m = subscribe.simple(topics=['#'], hostname="nam1.cloud.thethings.network", port=1883,
    auth={'username':'esp32-sound','password':'NNSXS.7ZN1O2YWWQ4IOZYXW75QSFRFATNRUXARKVAOCLQ
        .GSQBD2I3S2AFTDBUDYFXENGJJSA7DFEOPR4BN3JKG4DCLH25WLA"}, msg_count=5)
7
8 number = 1
9
10 for a in m :
11     x = a.payload
12
13     y = json.loads(x)
14
15     # the result is a Python dictionary:
16     print ("#" , number)
17     print ("Device ID: " , y["end_device_ids"]["device_id"])
18     print ("Time: " , y["received_at"])
19
20     number += 1
21
22     z = y["uplink_message"]["frm_payload"]
23
24     message = base64.b64decode(z)
25     #print(message)
26
27     li = list(message)
28
29     sound_array = []
30
31
32     for i in range(0,242,2):
33         f = li[i] << 8
34         sound_array.append(f + li[i+1])
35
36     print ("Sensor Data :", sound_array)
37
38
```

Transmission starts:
17:48:02

Transmission ends:
18:36:02

=48min

Transmission starts:
17:48:02
Transmission ends:
18:36:02
=48min



1.1 What has been changed since the Midterm

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

TABLE 2: Data rate codes defined by LoRaWAN.

Data Rate Code	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

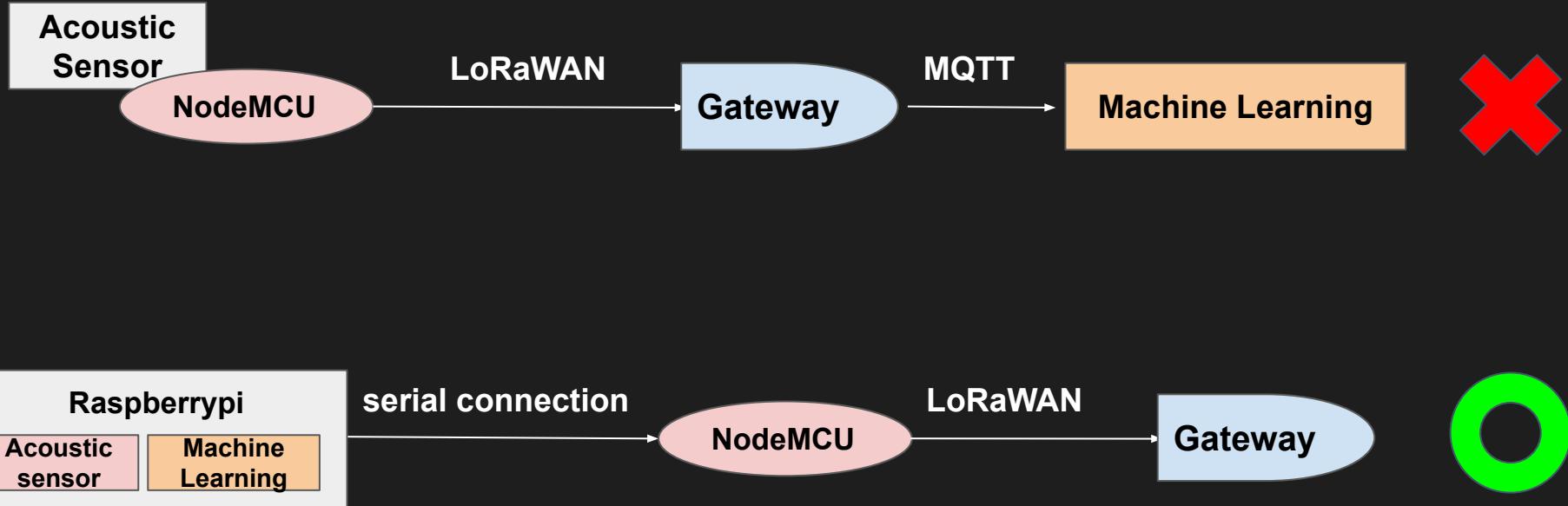
7

125kHz

4/5

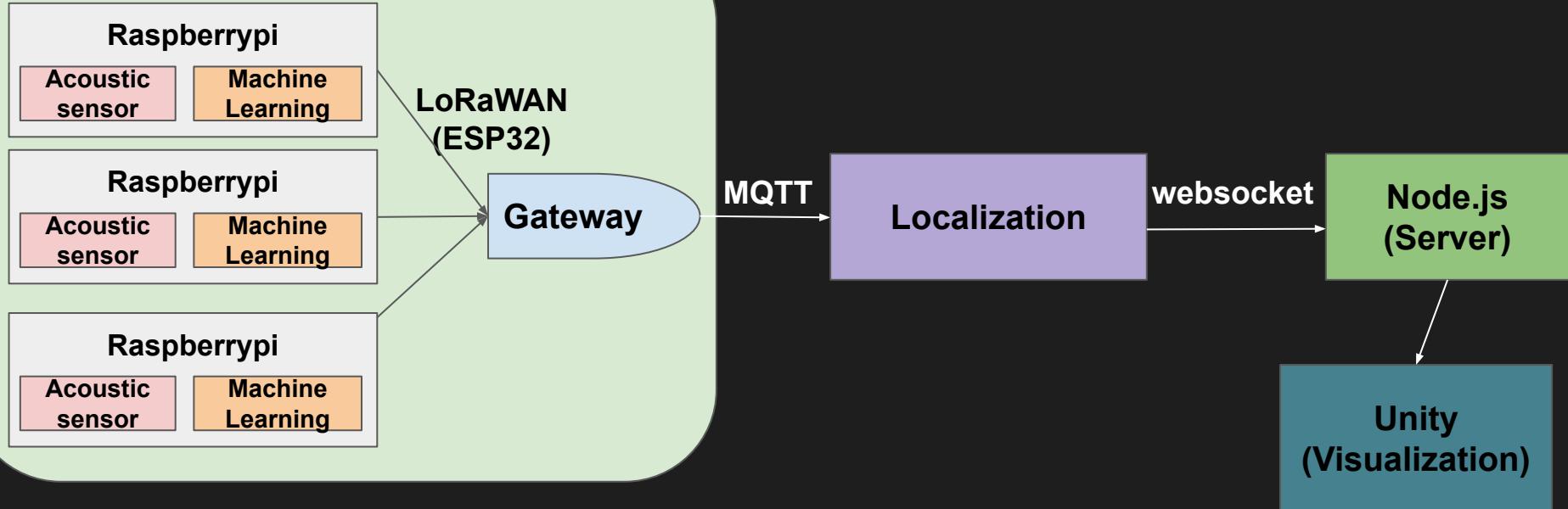


1.1 What has been changed since the Midterm



1.2 New Project Architecture

The Things Stack





02

Implementation



End node connection



Localization

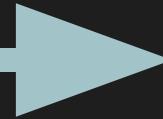
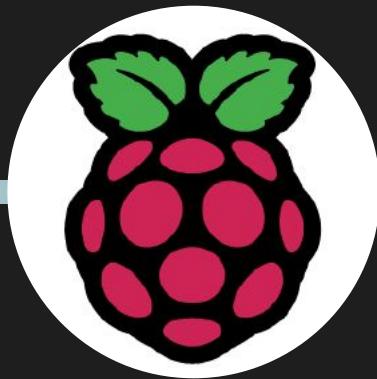


Visualization

2.1 End node connection



MAX4466

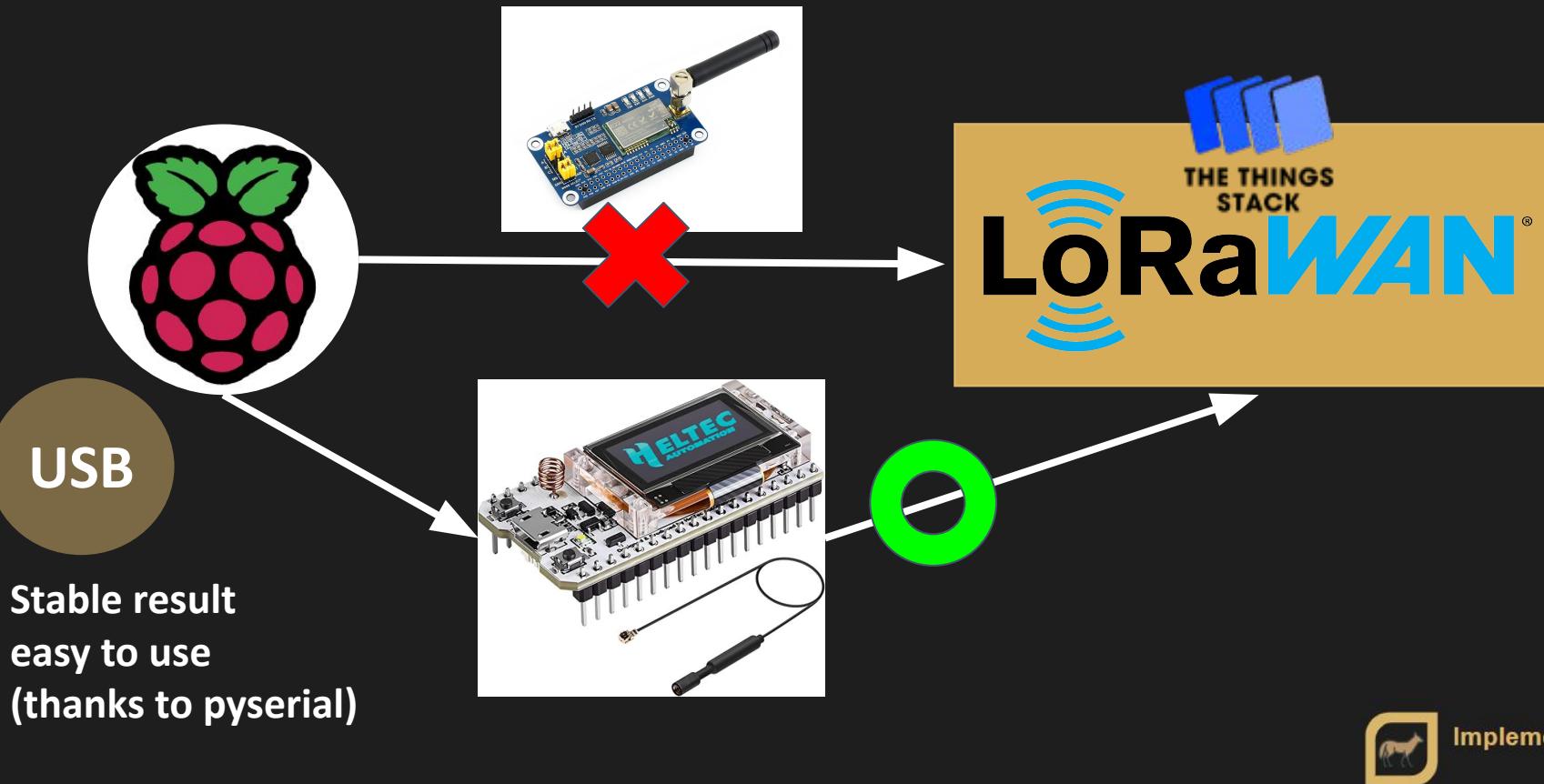


AK5371



Implementation

2.1 End node connection



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)

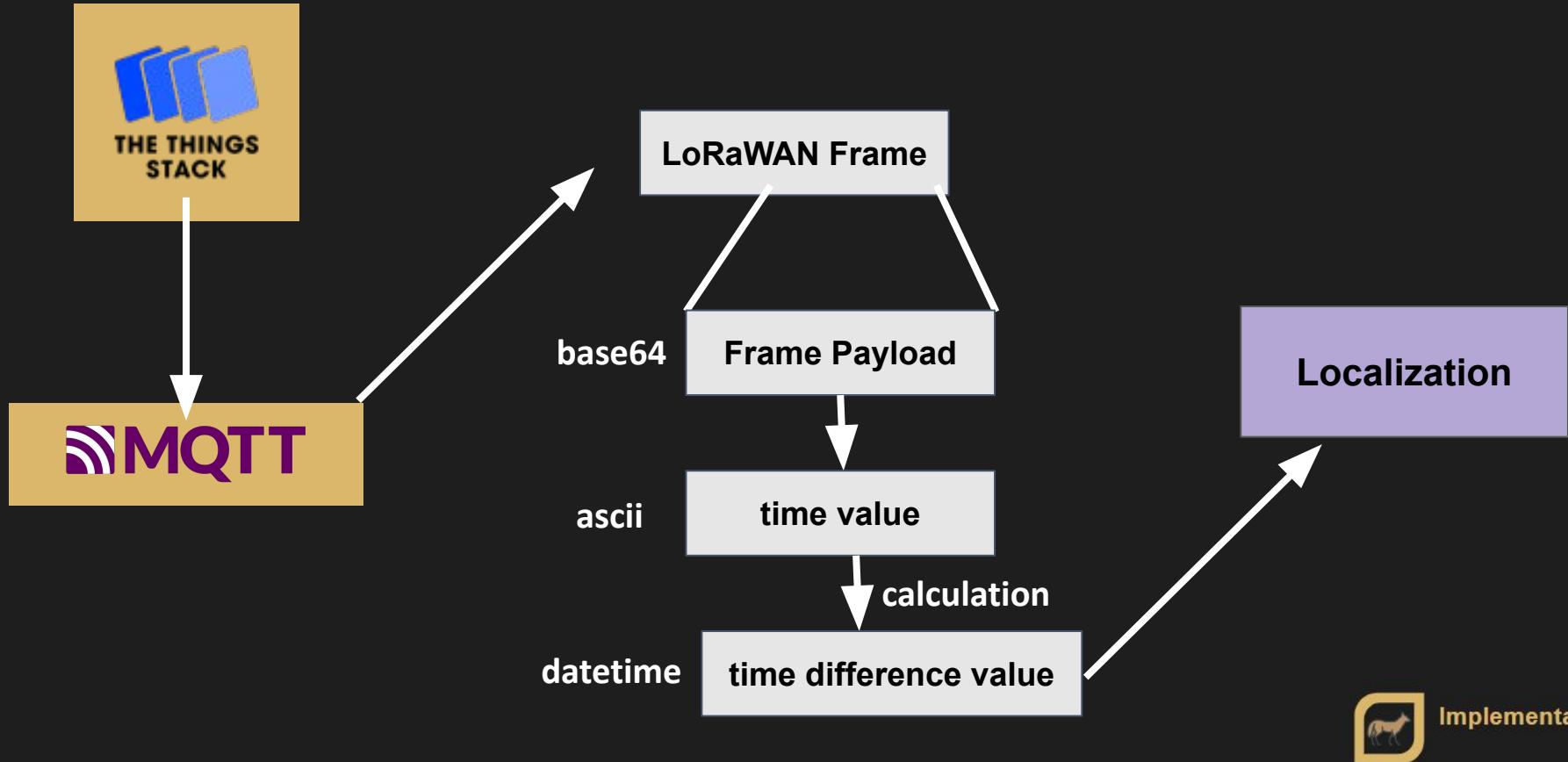


```
String data = Serial.readStringUntil('\n'); //read string from serial port sent by raspberry pi
data.getBytes(mydata, 30); //put into mydata array
do_send(&sendjob); //schedule next transmission
```



Implementation

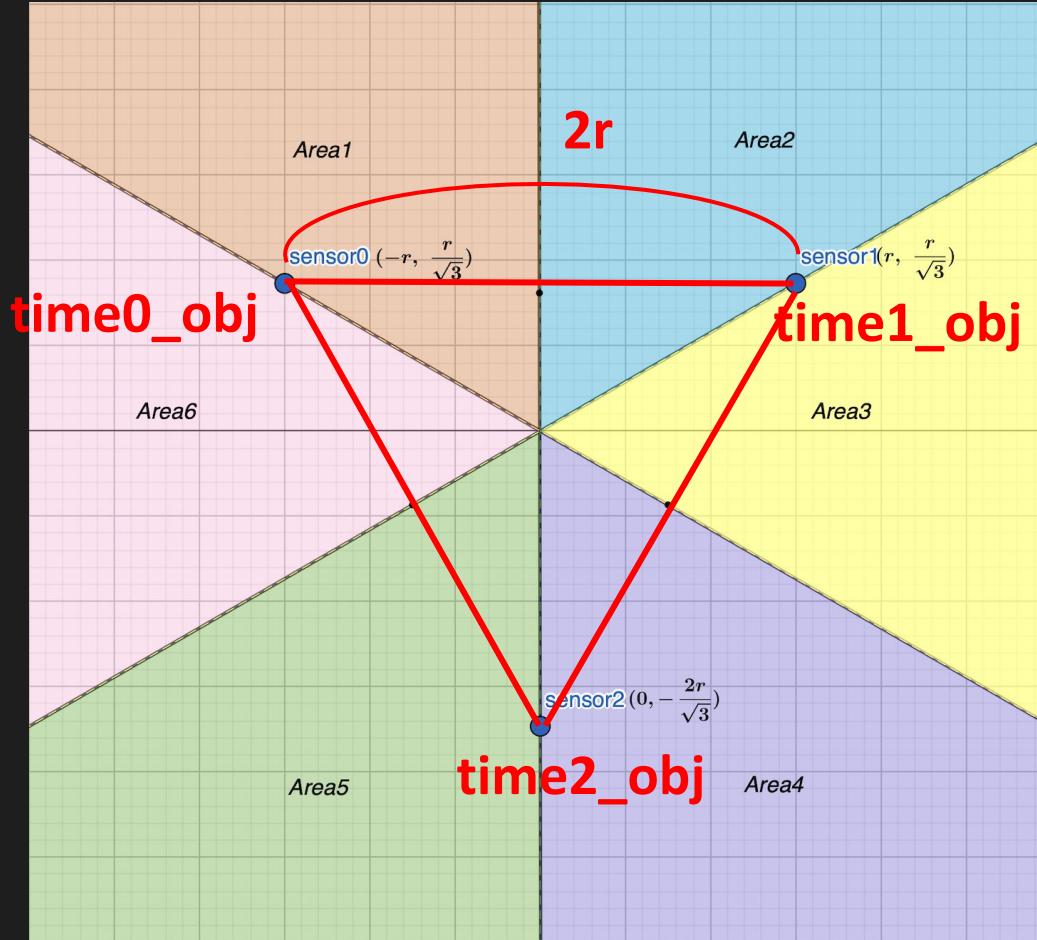
2.2 Raw frame data processing



Implementation

2.2 Localization

Unit: cm, ms



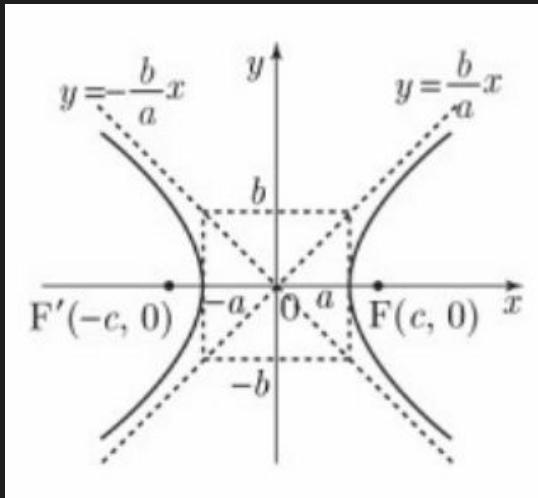
td0 td1 td2
v
Area



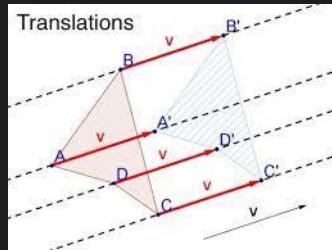
Implementation

2.2 Localization

Hyperbolic function Parallel Translation



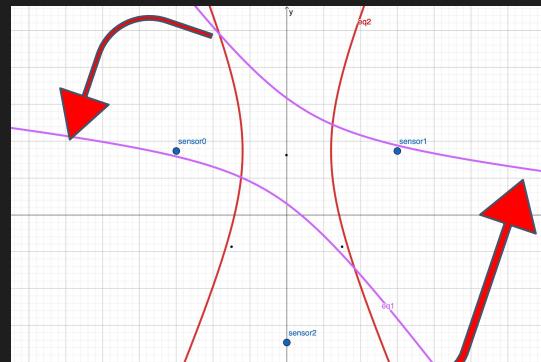
$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$



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

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

Rotation Transformation



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

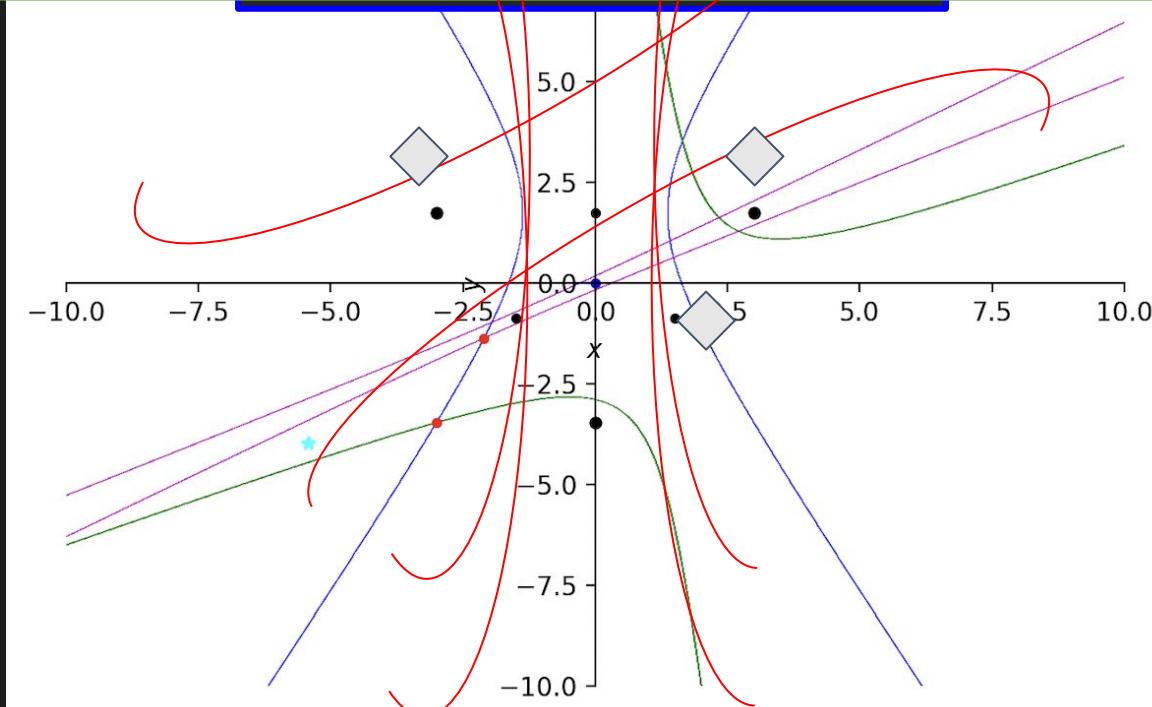
$$f(x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta) = 0$$



Implementation

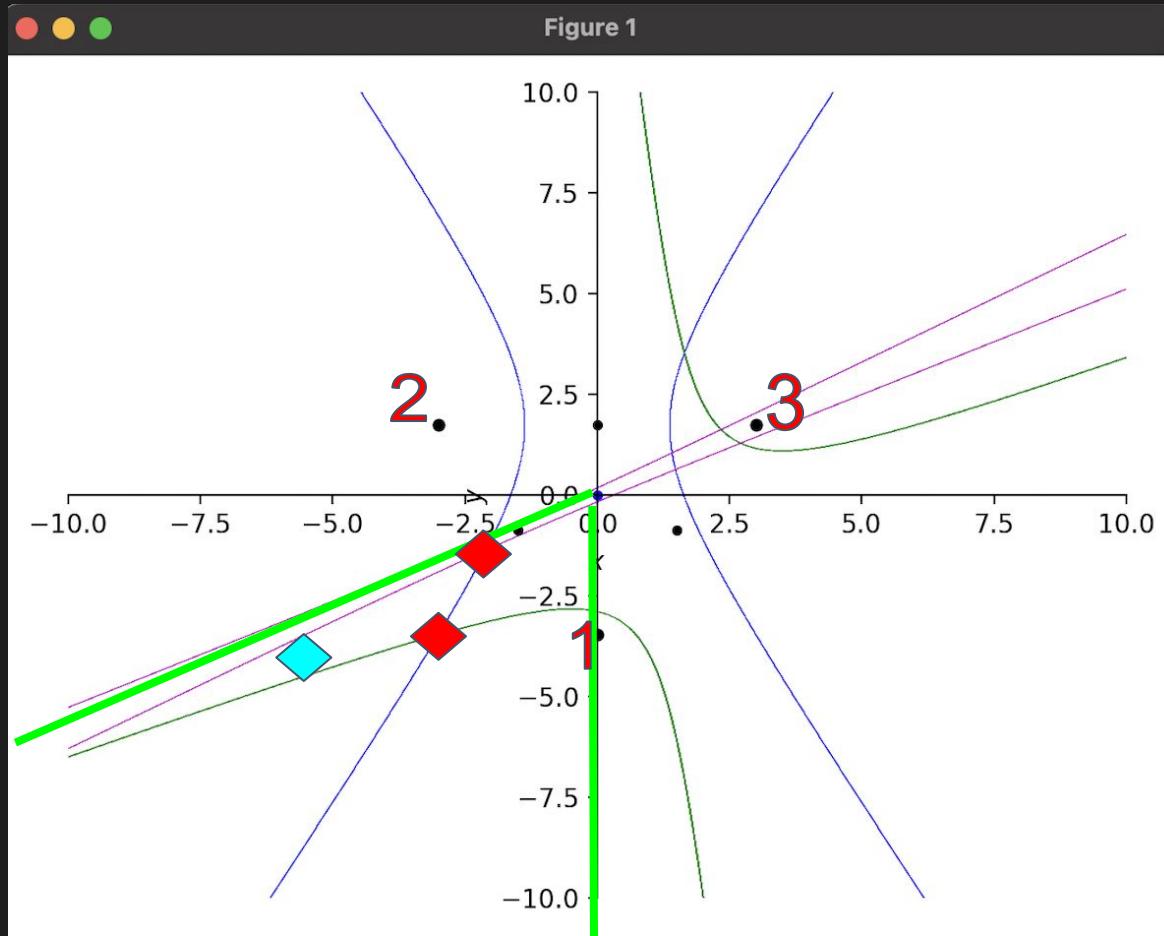
2.2 Localization

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



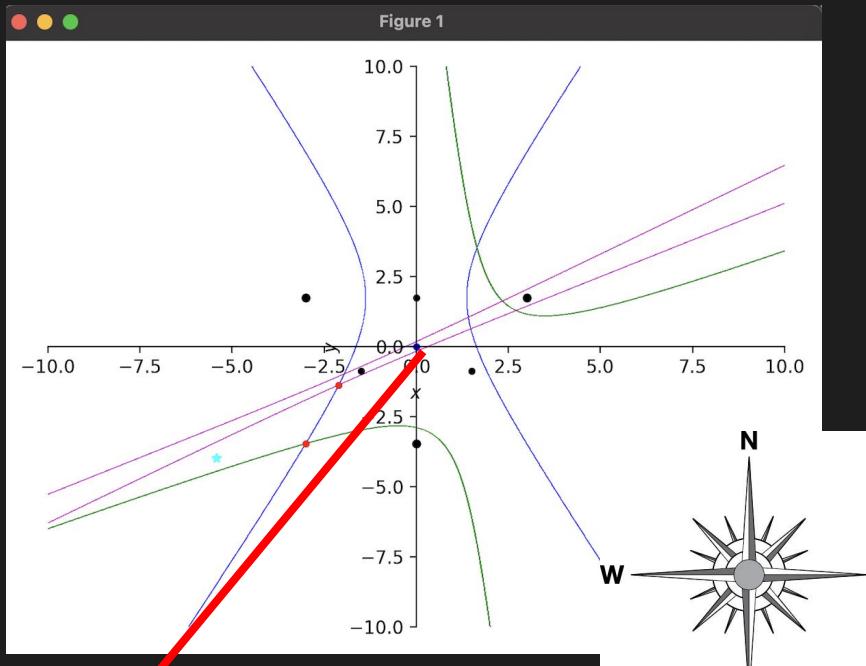
Implementation

2.2 Localization



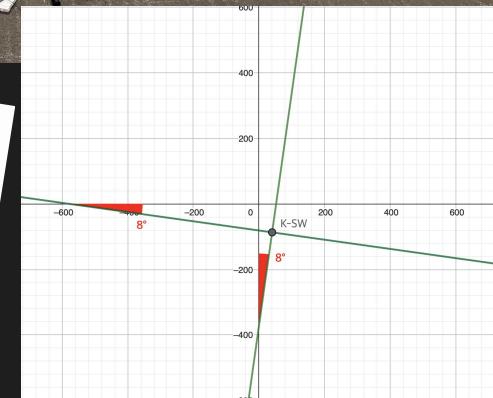
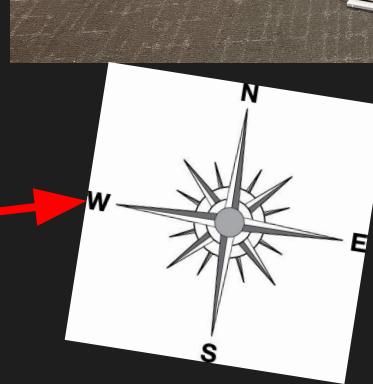
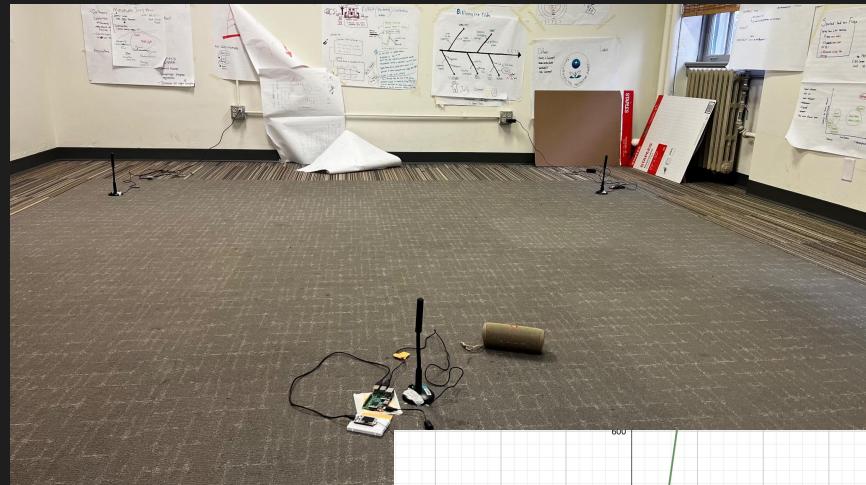
Implementation

2.2 Localization



(m, n)
(unit: cm)

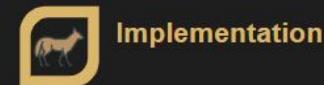
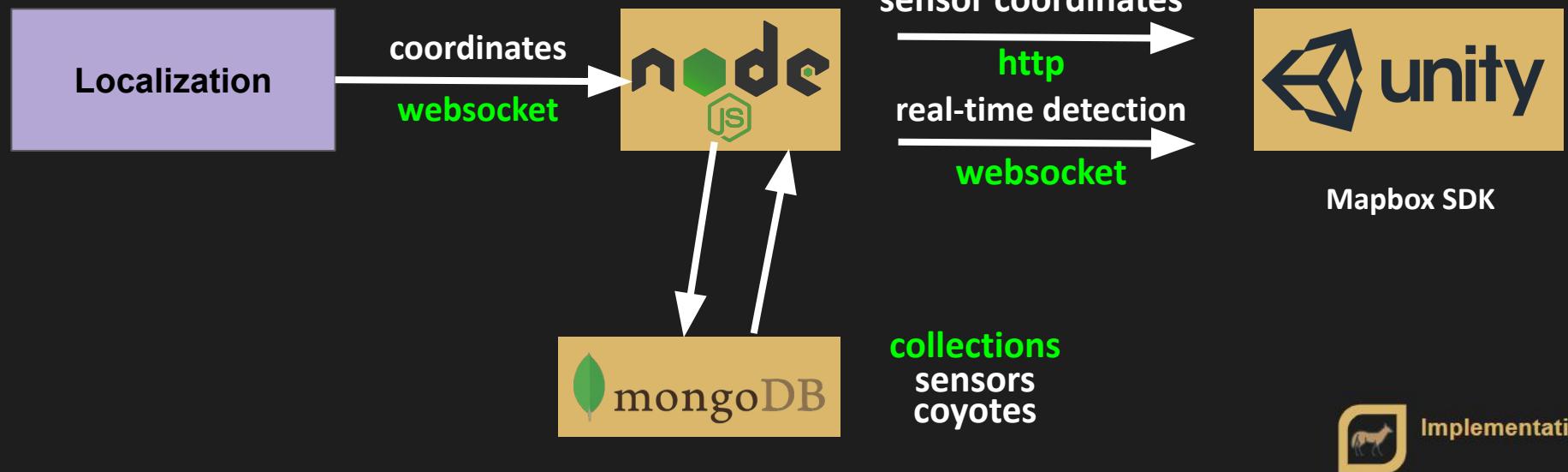
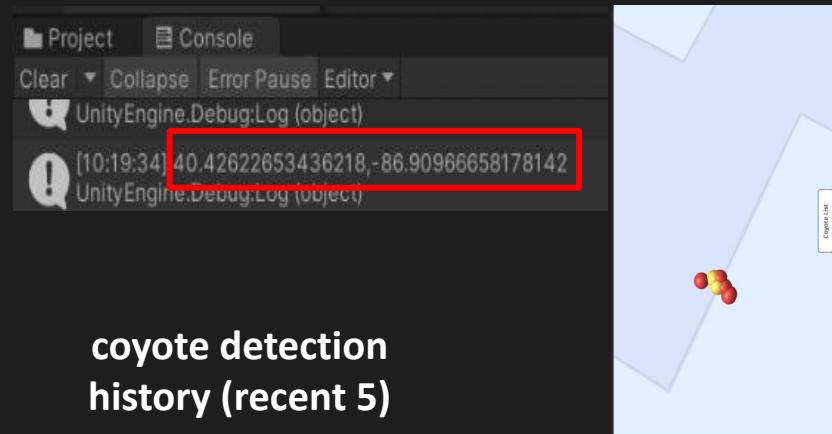
$(40.426\ldots,$
 $-86.909\ldots)$ (unit: $^{\circ}$)



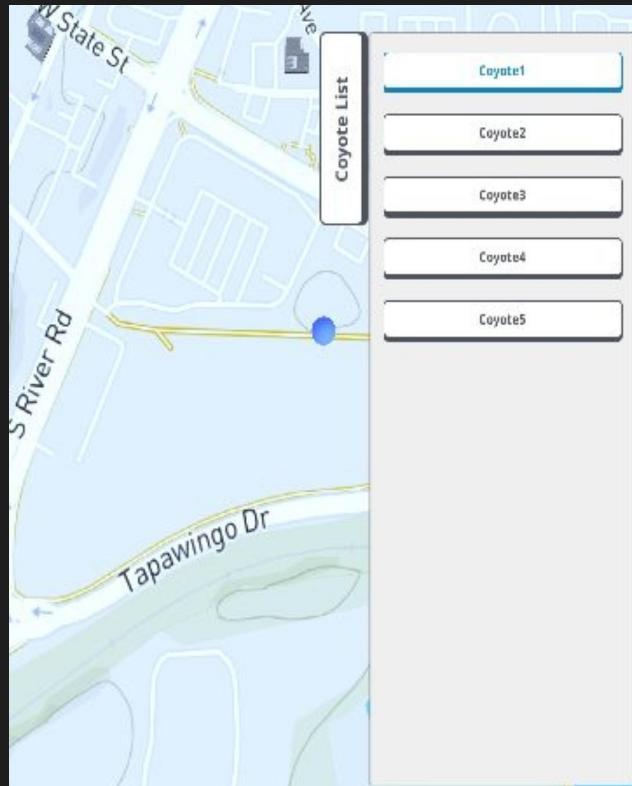
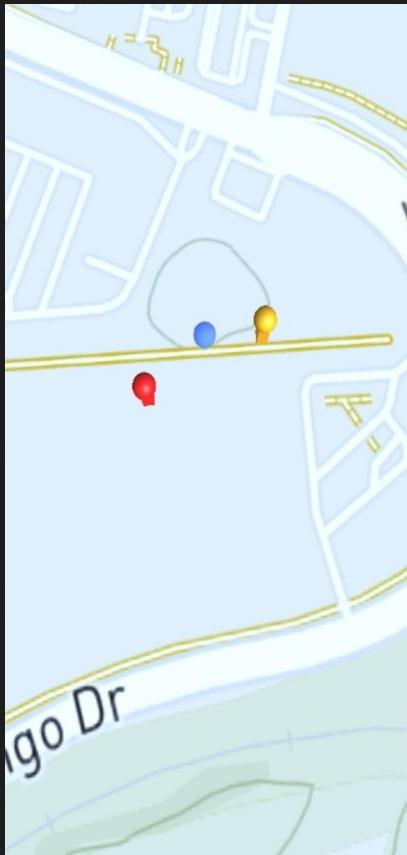
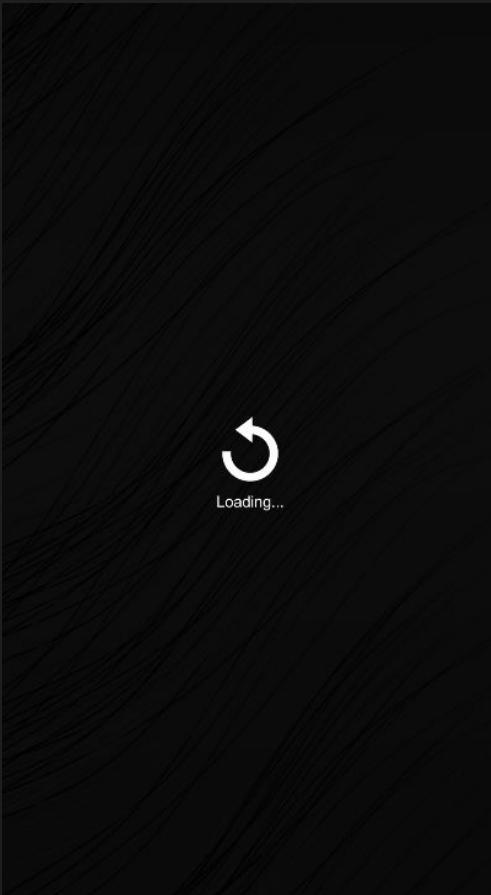
Implementation

2.3 Visualization

```
output_x : 48.8610234623519  
output_y : -31.9126686196806  
(40.42622653436218, -86.90966658178142)  
40.42622653436218,-86.90966658178142  
Success  
Over 1 Cycle
```



2.3 Visualization



Implementation

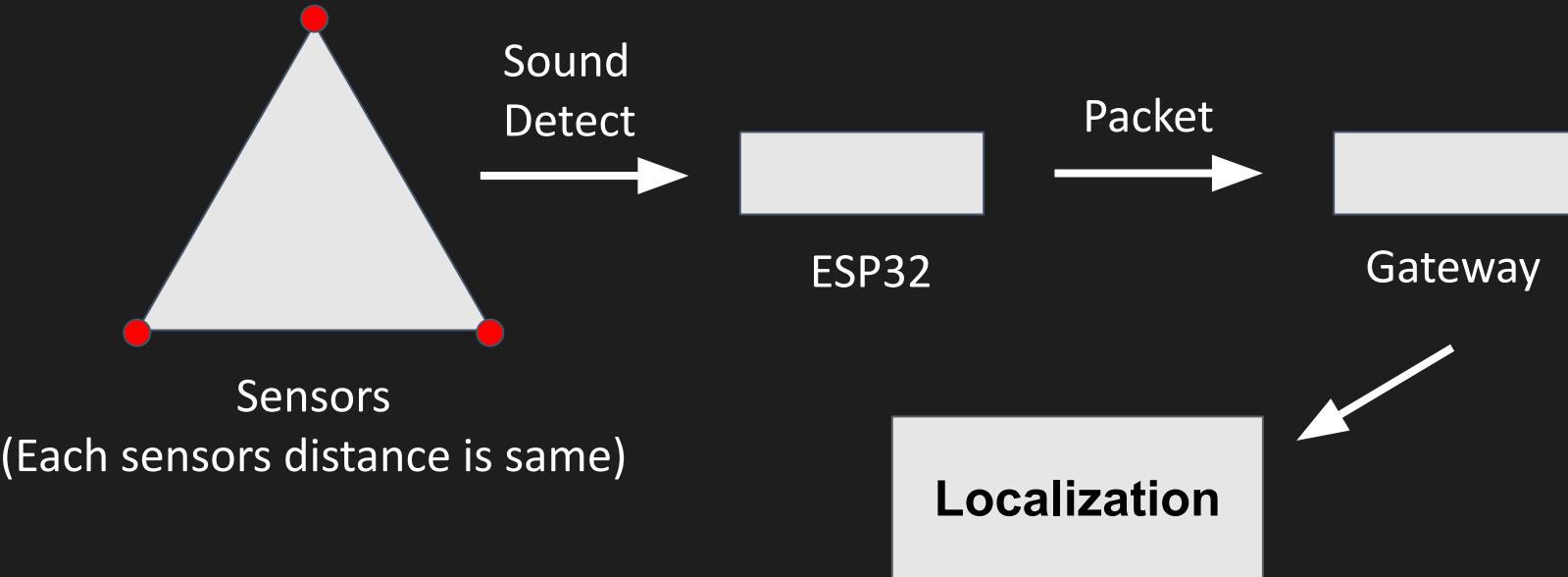


Experiment

03

Experiment

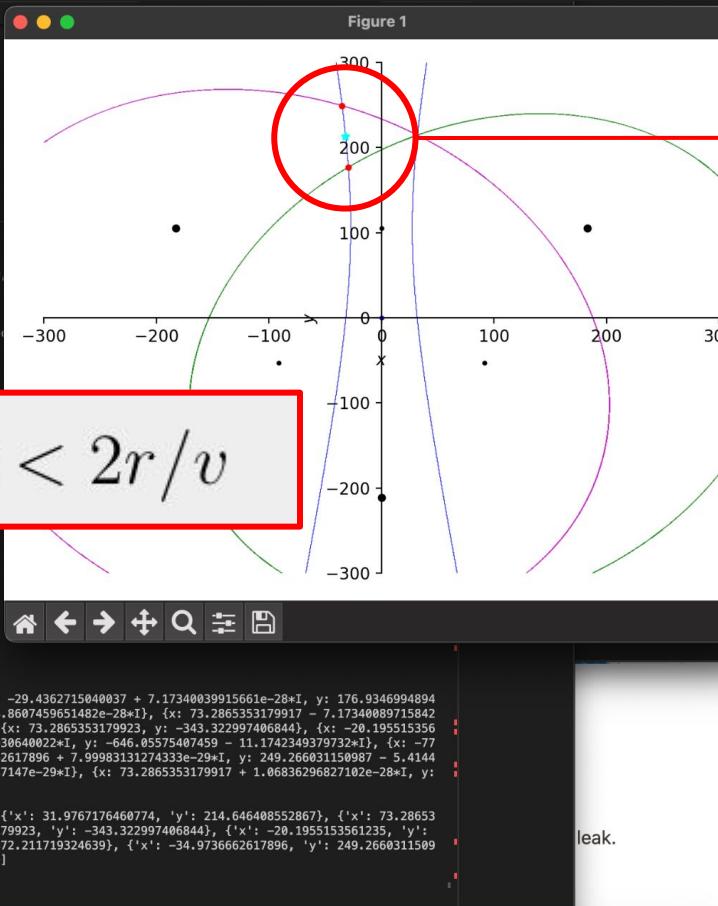
3. Experiment



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: '))
230
문제 ② 출력 디버그 콘솔 터미널 JUPITER
File "/Users/jaeuiboo/Library/Python/3.9/lib/python/site-packages/paho/mqtt/client.py", line 1150
    socklist = select.select(rlist, wlist, [], timeout)
KeyboardInterrupt
○ jaeuiboo@bujaehuiboo-MacBookPro Jaeuiboo % python -u "/Users/jaeuiboo/Desktop/algoritm2022/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': 176.9346994894
15 + 4.86074596514832e-28*I}, {'x': 31.9767176460774 + 7.173400399423e-28*I, 'y': 214.646408552867 + 4.8607459651482e-28*I}, {'x': 73.2865353179917 - 7.173400889715842
e-28*I, 'y': -343.322997406844 - 4.86074596514834e-28*I}, {'x': 31.9767176460774, 'y': 214.646408552868}, {'x': 73.2865353179923, 'y': -343.322997406844}, {'x': -20.195515356
1235 - 452.78930640022*I, 'y': -646.05575407459 + 11.1742349379732*I}, {'x': -28.1955153561235 + 452.78930640022*I, 'y': -646.05575407459 - 11.1742349379732*I}, {'x': -77
.3616628143088 - 9.68579433956294*I, 'y': -372.211719324639 + 6.55554831618e-29*I}, {'x': -34.9736662617896 + 7.9989313127433e-29*I, 'y': 249.266031150987 - 5.4144
49986228016-29*I}, {'x': 31.9767176460774 - 7.23890367376146e-29*I}, {'x': 214.646408552868 + 6.08980534337147e-29*I}, {'x': 73.2865353179917 + 1.06836296827102e-28*I, 'y':
-343.322997406845 - 7.23890367376146e-29*I}]
12
[{'x': -69.1714832629622, 'y': -313.884321713277}, {'x': -29.4362715040037, 'y': 176.934699489415}, {'x': 31.9767176460774, 'y': 214.646408552867}, {'x': 73.28653
53179917, 'y': -343.322997406844}, {'x': 31.9767176460774, 'y': 214.646408552868}, {'x': 73.2865353179923, 'y': -343.322997406844}, {'x': -20.1955153561235, 'y':
-646.05575407459}, {'x': -28.1955153561235, 'y': -646.05575407459}, {'x': -77.3616628143088, 'y': -372.211719324639}, {'x': -34.9736662617896, 'y': 249.266031150987
- 5.414449986228016}, {"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}]
```

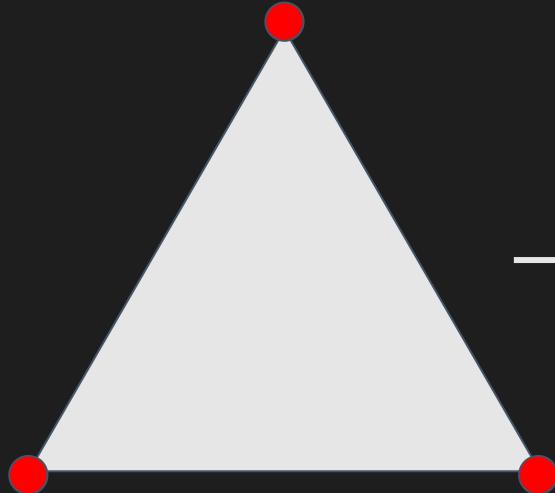


Wrong
Result

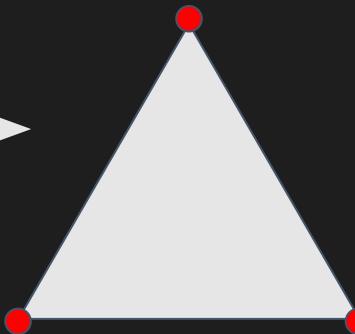
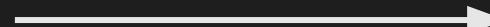


Experiment

3. Experiment



Sensor Distance: 144ft



Sensor Distance: 60ft

3. Experiment

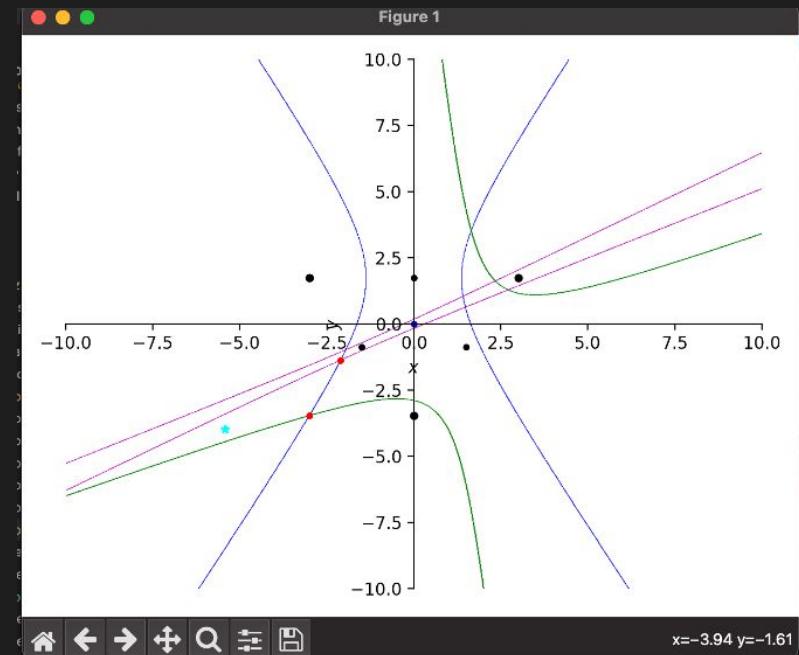
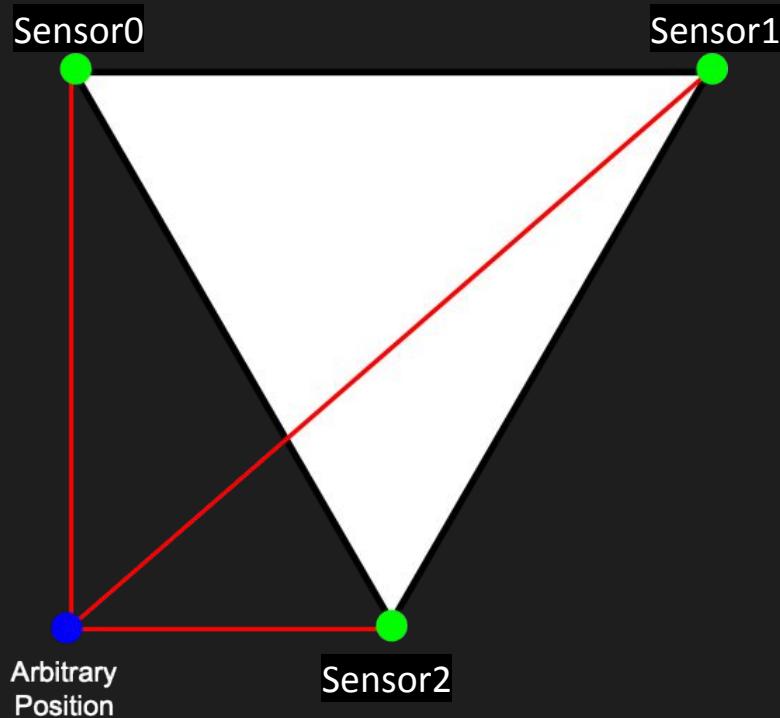
```
#real code
for arr in record_voice:
    if(arr[0]>0.057):
        output = datetime.now()
        output = output.strftime('%Y-%m-%d %H:%M:%S.%f')[14:26]
        print(output)
        output = output.encode('utf-8')
        ser.write(output)
        time.sleep(1)
    break
```

"f_port": 1,
"frm_payload": "NDk6NTMuMzY4NTQ2AAAAAAAAAAAAAAA==",
"frm_payload": "E

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



3. Experiment



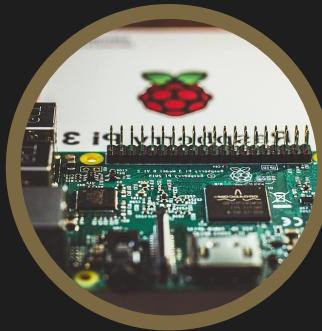
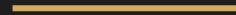
Experiment

3. Experiment



3. Experiment

Time Delay?

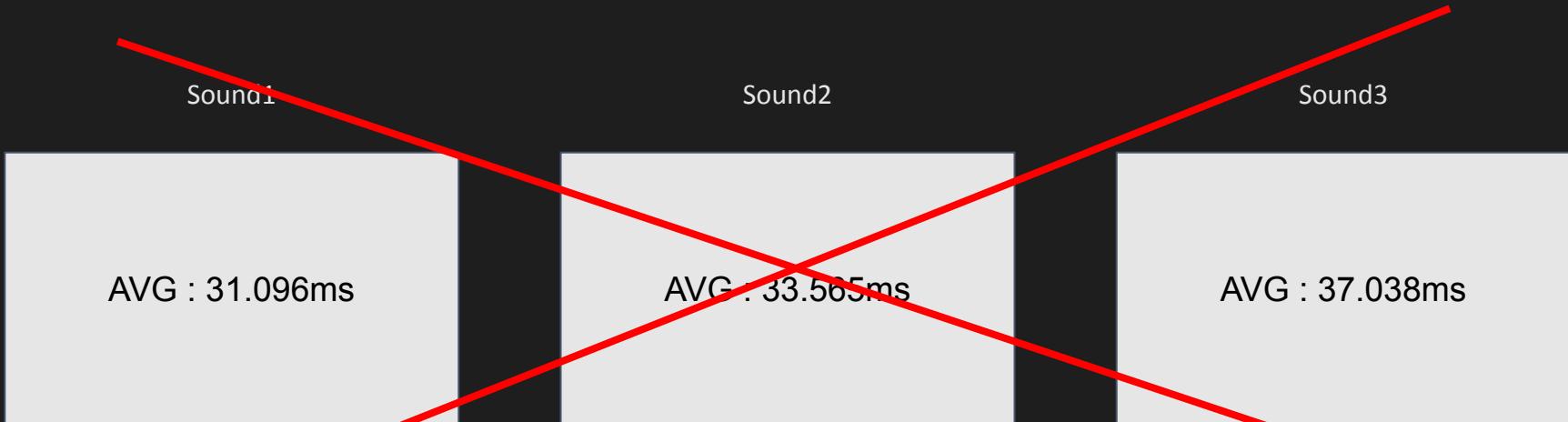


1m Length usb cable took 10ns to communicate



Experiment

3. Experiment



Biggest problem is
recognition accuracy



3. Experiment

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

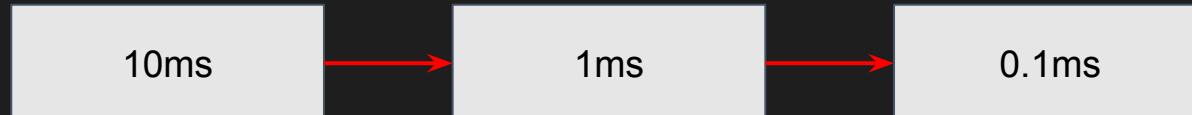


3. Experiment

```
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
Sampling Rate x 2
i = 0
# FILE_PATH = '/home/coyote/code'

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

while(True):
    arr_high = []
    i = 0
    # FILE_NAME = 'REC_FILE'+str(i)
    record_voice = sd.rec( int(second*fs), samplerate=fs, channels=2 ) # per 0.001 sec
    sd.wait()

    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()

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

        record_voice = sd.rec( int(second*fs), samplerate=fs, channels=2 ) # per 0.001 sec
        sd.wait()
```

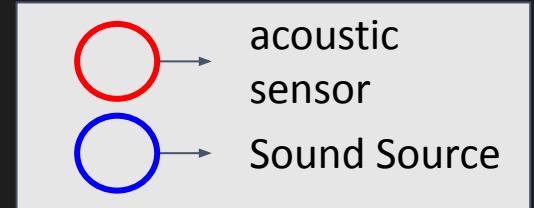
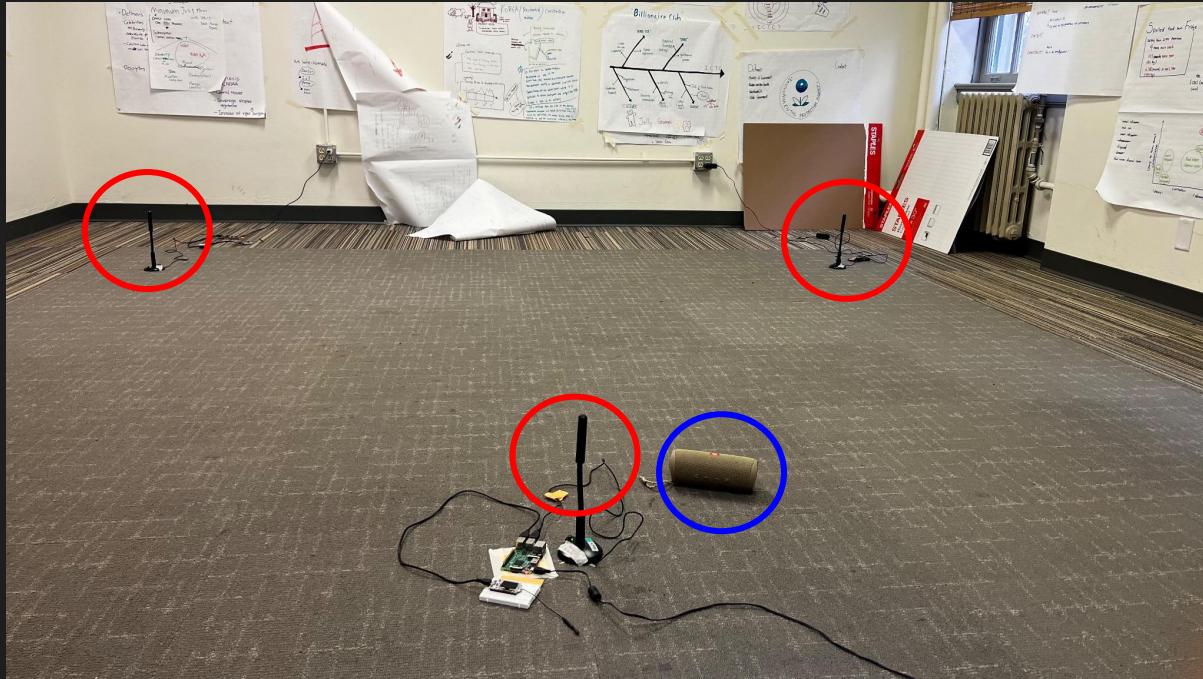


3. Experiment

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

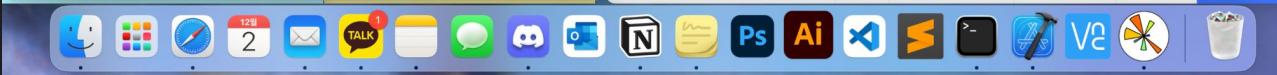
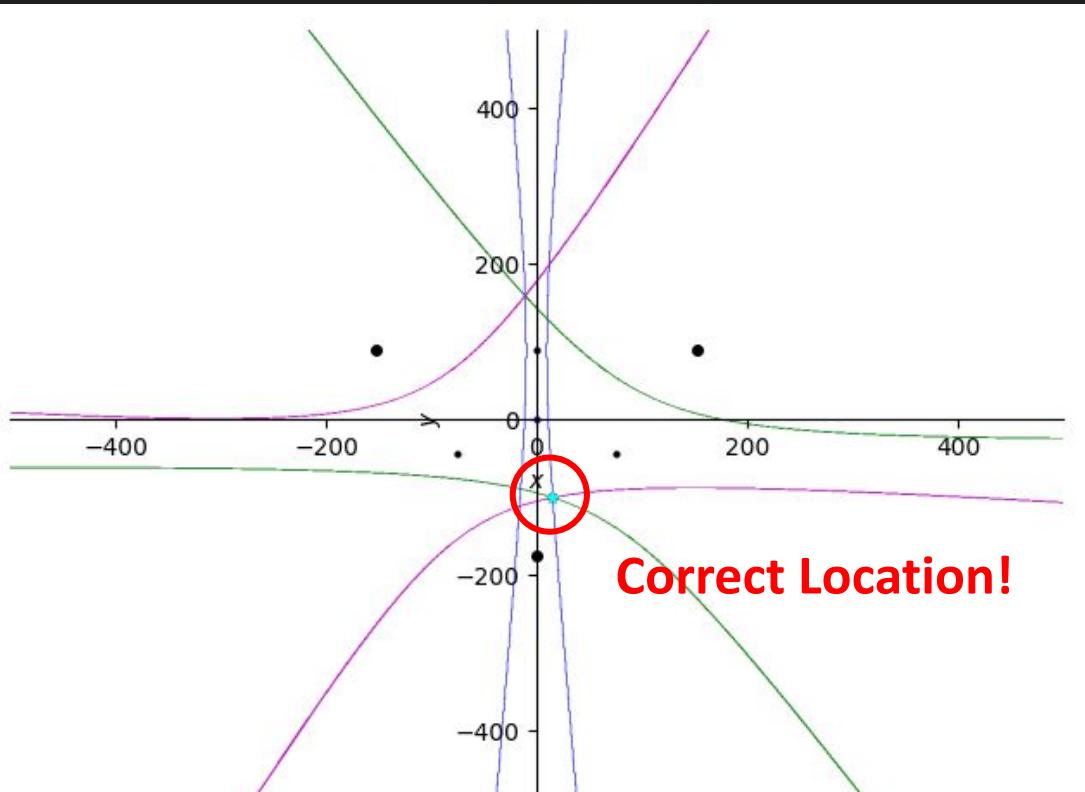


3. Experiment



3. Experiment

```
Python
ki_myeongjun@giemyeongjun-MacBookAir ~ % python3 /Users/ki_myeongjun/Desktop/localization_combined_ver4.py
('sound1', datetime.datetime(1900, 1, 0, 10, 35, 822201), 40.4203008430482, -8
425781)
('sound1', datetime.datetime(1900, 1, 0, 10, 35, 867918), 40.4203008430482, -8
425781)
('sound1', datetime.datetime(1900, 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
t0d
0:00:00.005684
t1d
0:00:00.045717
t2d
0:00:00.051401
diff_t0d
0.5684
diff_t1d
4.5717
diff_t2d
5.1401
area is -> 4
result : len: 12
[{'x': -14.9983971765955, 'y': -89.8506681555633, 'z': -9.32584829
*I}, {'x': -10.7396993053258, 'y': -1.678976488270156-29*I, 'z': 158.31170473219 - 2.94332
-29*I}, {'x': 10.0909025388922 + 2.15441673702121e-29*I, 'y': 128.67774972911 + 3.77
75e-29*I}, {'x': 15.5497636794982 + 4.8445402198331e-29*I, 'y': -101.026648273991 +
021934e-29*I}, {'x': -28.9334674732088 - 302.978562327358*I, 'y': -51.3898378471993 +
7437603*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': 128.67774972911 + 0.e-19*I}, {'x': -16.1113630215503 - 1.842
29-29*I, 'y': -12.148753628244 + 0128438231028e-29*I}, {'x': -10.7396993053258 -
407536e-30*I, 'y': 158.311704732191 - 1.43120495327257e-29*I}, {'x': 12.0286252808784
288623658e-30*I, 'y': 197.937784514687 - 1.08148984248826e-29*I}, {'x': 15.5497636794
2754193534058e-29*I, 'y': -101.026648273991 - 4.6509949409453e-29*I}]
12
[{'x': -14.9983971765955, 'y': -89.8506681555633}, {'x': -10.7396993053258, 'y':
73219}, {'x': 10.0909025388922, 'y': 128.67774972911}, {'x': 15.5497636794982, 'y':
266648273991}, {'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': -12.148753628244}, {'x': 396993053258, 'y': 158.311704732191}, {'x': 12.0286252808784, 'y': 197.9377845146
15.5497636794982, 'y': -101.026648273991}, {'x': 15.5497636794983, 'y': -101.026648273991}
12
result : len: 3
[{'x': 15.5497636794982, 'y': -101.026648273991}, {"x": 15.5497636794983, 'y': -101.026648273991}, {"x": 15.5497636794982, 'y': -101.026648273991}]
```

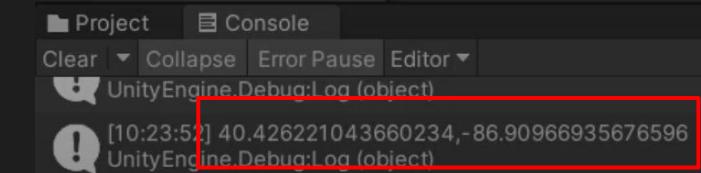


3. Experiment

```
output_x : -6.8874344139165835  
output_y : 7.38169010153424
```

```
(40.426221043660234, -86.90966935676596)
```

```
File Edit Assets GameObject Component Jobs Tools Mapbox Window Help
```



```
File Edit Assets GameObject Component Jobs Tools Mapbox Window Help
```

Code Jeah... Hierarchy Game Display 1 1920x1080 Portrait Scale 0.39x

Over 1 Cycle
('sound1', datetime.datetime(1900, 1, 0, 23, 4, 554989), 40.4203008430482, -86.9025421142578)
('sound2', datetime.datetime(1900, 1, 0, 23, 4, 676624), 40.4203008430482, -86.9025421142578)
None
Over 1 Cycle
('sound1', datetime.datetime(1900, 1, 0, 23, 24, 788329), 40.4203008430482, -86.9025421142578)
1)
('sound2', datetime.datetime(1900, 1, 0, 23, 24, 924866), 40.4203008430482, -86.9025421142578)
1)
('sound1', datetime.datetime(1900, 1, 0, 23, 24, 956708), 40.4203008430482, -86.9025421142578)
1)
===== Receive Done ======

td0
0:00:00.168371
td1
0:00:00.031834
td2
0:00:00.136537
diff_td0
1.68371
diff_td1
0.31834
diff_td2
1.36537
area is > 6
result =
[{'x': -32.4997647430863, 'y': 10.6997823292727 + 4.47196168089515e-29*I},
(x: -31.2358299968989 + 6.46661158967318e-29*I, y: 26.273876788751 - 1.9215631253726e-29*I), {
x: 34.4885619693581 + 7.10739574685069e-29*I, y: -13.4951140270117 - 4.80162854646729*I},
(x: 36.4885619693581 - 7.10739574685069e-29*I, y: -27.6125639071384 + 2.2512259943925e-29*I), (x:
-32.4997647430864 + 2.188063806699e-29*I, y: 10.6997823292727 + 7.12128224629291e-30*I), {
x: -19.7206299890802 + 1.6182876892831e-29*I, y: 19.37776842707 + 4.1230728328323e-30*I}, (x: 2.
0233689674781 + 3.58282105054253e-29*I, y: -19.4784581083147 + 6.5883588259696e-30*I}, {
x: 34.889782977819 + 3.79526254001988e-29*I, y: -13.4951140270118 + 9.67084529684537e-30*I}, (x: -39.
39767438863 + 2.3149983806699e-29*I, y: 10.6997823292727 + 5.68808869738446e-29*I), (x: -29.607
97764972 + 4.4862782720958e-29*I, y: 54.894646576159 - 34.8897823292727 + 34.8897823292727 + 2.56577325965283e-29*I, y: -13.4951140270118 - 8.52082134599155e-29*I}]
12
[{'x': -32.4997647430863, 'y': 10.6997823292727}, {'x': -31.2358299968989, 'y': 26.273876788751},
, {'x': 34.4885619693581, 'y': -13.4951140270117}, {'x': 36.4885619693581, 'y': -27.6125639071384},
{'x': -32.4997647430864, 'y': 10.6997823292727}, {'x': -19.7206299890802, 'y': 19.37776842707},
{'x': 2.0233689674781, 'y': -19.4784581083147}, {'x': 34.889782977819, 'y': -13.4951140270118},
{'x': -39.39767438863, 'y': 10.6997823292727}, {'x': -29.6079776892831, 'y': 54.894646576159},
{'x': -29.6079776892831, 'y': 54.894646576159}, {'x': 34.8897823292727, 'y': -13.4951140270118}]
type of sum_x: <class 'float'>
result2 / len: 3
[{'x': -32.4997647430863, 'y': 10.6997823292727}, {'x': -32.4997647430864, 'y': 10.6997823292727}]
output_x : 10.6997823292727
(40.426221043660234,-86.90966935676596)
40.4262210446573,-86.90966931673398
success
Over 1 Cycle



```
File Edit Assets GameObject Component Jobs Tools Mapbox Window Help
```

Console

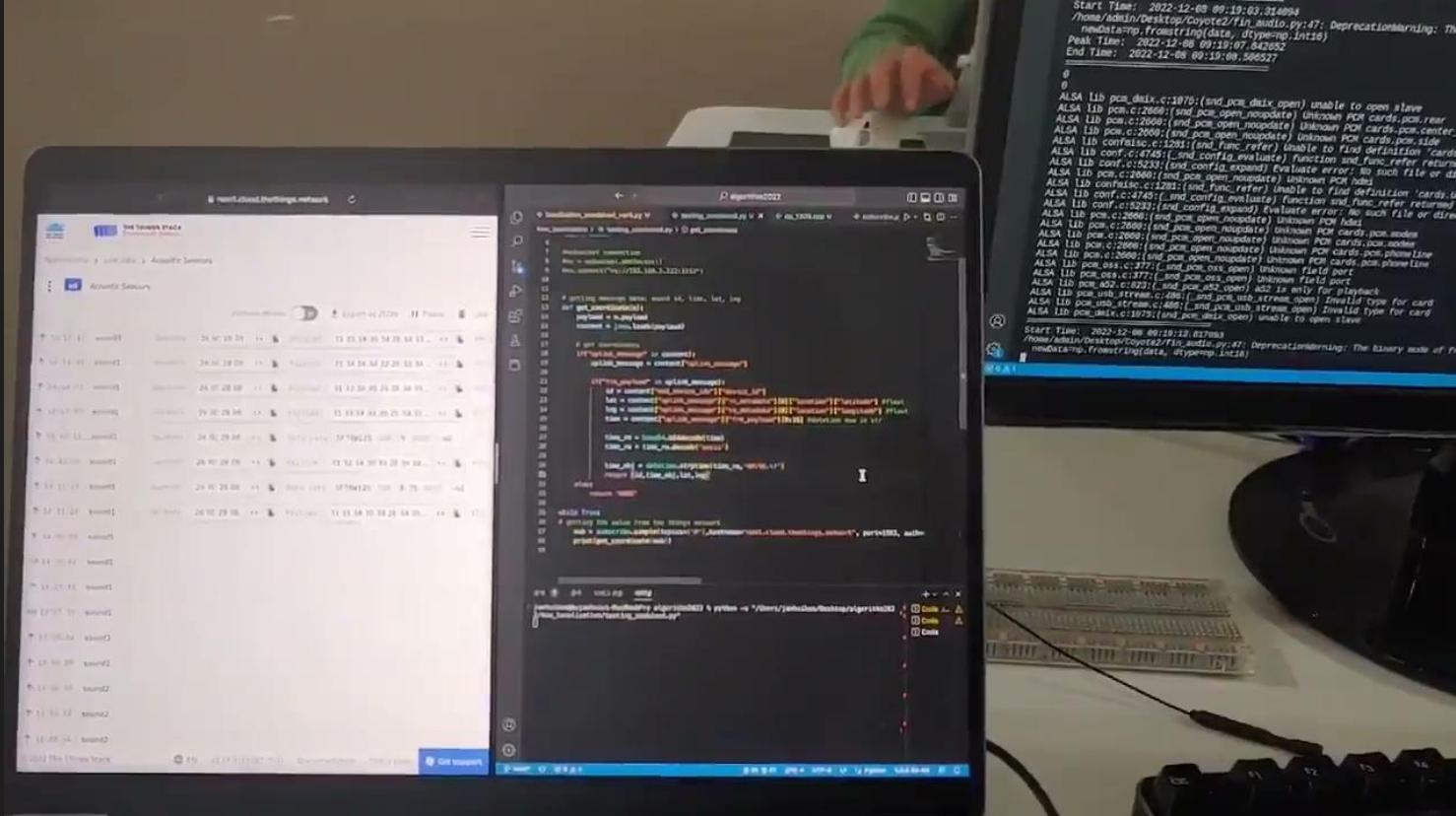
Clear Collapse Error Pause Editor

UnityEngine.Debug:Log (object)

[10:23:10] 40.42621871446573,-86.90966931673398

UnityEngine.Debug:Log (object)

3. Experiment



Experiment

3. Experiment

The screenshot shows a terminal window with several tabs and multiple panes of output.

Code Editor Tab: Displays the file `ksw_localization > testing_combined.py > get_coordinate`. The code defines a function `get_coordinate` that parses JSON payload data.

```
def get_coordinate(m):
    payload = m.payload
    content = json.loads(payload)
```

Terminal Tab: Shows the command `2/ksw_localization/testing_combined.py` running. It outputs three lines of data, each containing a timestamp, device ID, and coordinates. The coordinates (19, 17, 512032), (20, 49, 722499), and (21, 14, 70715) are highlighted with red boxes.

```
('sound1', datetime.datetime(1900, 1, 1, 0, 19, 17, 512032), 40.4203008430482, -86.90254211425781)
('sound1', datetime.datetime(1900, 1, 1, 0, 20, 49, 722499), 40.4203008430482, -86.90254211425781)
('sound1', datetime.datetime(1900, 1, 1, 0, 21, 14, 70715), 40.4203008430482, -86.90254211425781)
^CTraceback (most recent call last):
```

Output Tab: Shows two entries for device `sound1` with DevAddr 26 0C 2B DB. The first entry has Payload: 32 30 3A 34 39 2E 37 32 ... and the second has Payload: 31 39 3A 31 37 2E 35 31 ...

Bottom Terminal: Displays the command `mic.openStream()` at line 79. Below it, the terminal shows a series of timestamped log messages. The error message `unable to open slave` appears twice, once associated with `newData=np.fromstrin` and once with `Peak Time: 2022-12-08 09:21:14.070715`.

```
Start Time: 2022-12-08 09:21:13.003458
/home/admin/Desktop/Co ALSA lib pcm_dmix.c:1075:(snd_pcm_dmix_open) unable to open slave
newData=np.fromstrin ==
Peak Time: 2022-12-08 09:21:14.070715
End Time: 2022-12-08 09:21:18.195436
=====
0
1
Coyote!!
```

Bottom Right: A small icon of a horse and the word `Experiment`.



04

Future works

4. Future Works



Machine Learning Model Detection
=> 'Decibel' (as substitute)

Sensor
Sensitivity



Future Works

[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_HULAFIxwU
- [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