

# **Smart bus system of waiting time estimation and special passengers tracking**

Group members:  
Emmanuel Sedicol

Jingwen Jiang

Jianing Tan

Instructor: Dr. Kumar Yelamarthi

Dr. Frank Walsh



## Problems and demands around bus:

- Unknowing the specific position of bus, people waste lots of time waiting at bus stops.
- There are demands that people want to know whether the bus is full or not and whether there is an empty seat on bus.
- As for school bus, parents want to know if their children miss the bus or not.



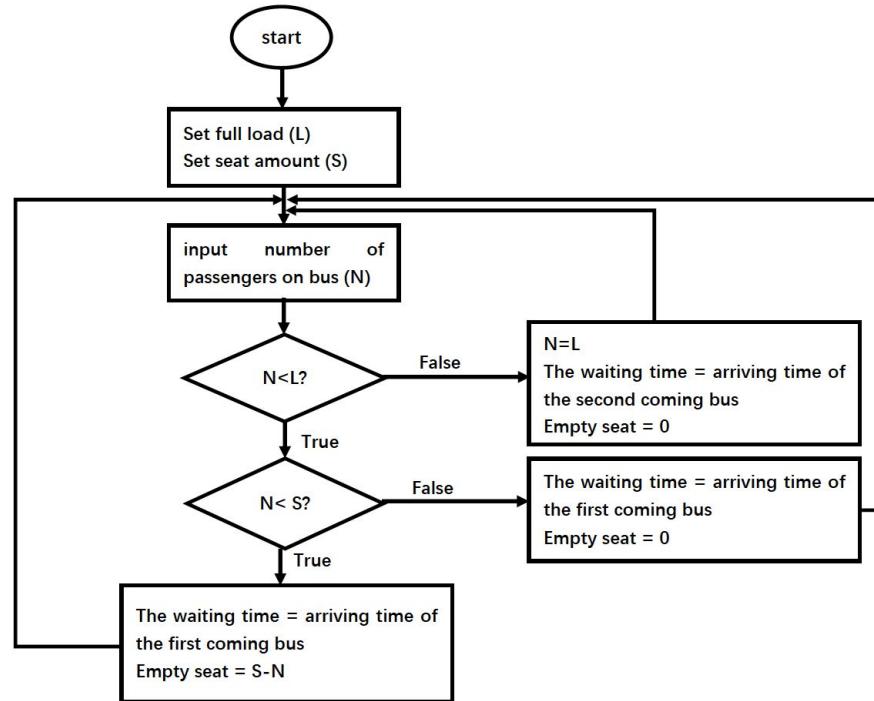
Smart Bus System: {  
Waiting time evaluation and empty seats detection system  
Parental Monitor Application

# Problem Statement



# Waiting time evaluation system and empty seats detection system:

- Using GPS module to track buses and transfer the data to the cloud.
- Regarding raspberry pi as a bluetooth receiver to detect the number of bluetooth signals from smart phone(number of passenger).
- Analysis on the cloud:



## **Parental Monitor Application:**

- Every bluetooth beacon has an unique address and store address in Database.
- Using raspberry pi as a bluetooth scanner to detect the specific beacon.
- Send text messages using twilio package (need to make twilio account online)



## **Rationale**



```
from pymongo import MongoClient
# ----- mongodb -----
client = MongoClient('mongodb://localhost:27017')

# -- connect to the db --
db = client['smart_bus']
post = db.children

# -- posting to db -- ##
def post_db():
    data = [
        {"name" : 'Jim Bob', "addr" : '8c:85:90:ab:ae:51', "miss" : 0, "momNo" : "0864785467"},
        {"name" : 'Mark Bob', "addr" : '98:28:a6:1d:71:30', "miss" : 0, "momNo" : "0864780455"}
    ]
    try:
        result = post.insert(data)
        print("Post Succesful")
        sense.show_message("Post Succesful")
    except Exception as error:
        print(error)
        sense.show_message("Post Failed")

def get_name(ad):
    try:
        result = post.find_one({ "addr" : ad })
        name = result.name
        print("name ",name)
    except Exception as error:
        print(error)
        sense.show_message("Failed")
    return name

def get_momsNo(ad):
    try:
        result = post.find({ "addr" : ad })
        number = result.momNo
    except Exception as error:
        print(error)
        sense.show_message("Failed")
    return number
```

# NoSQL

using NoSQL database to access data of a certain children

## Account details:

PROJECT NAME sedicolemanuel@gmail.com's Account [edit](#)

ACCOUNT SID ACc4b0a542f3e1d7d8b3ac261343c5bf14

AUTH TOKEN [hide](#) b0c3c9346758fd5fa9360081cd02113e

NUMBER	FRIENDLY NAME	CAPABILITIES
+353 76 620 6050 (national) IE	353766206050	 

Twilio link:

<https://www.twilio.com/>

# Twilio

**Install package in Pi**

**command - “sudo pip install twilio”**

**Create account**

**Add new number**

```

from pymongo import MongoClient
import subprocess
from sense_hat import SenseHat
from twilio.rest import Client
sense = SenseHat()

addr = ['8c:85:90:ab:ae:51', '98:28:a6:1d:71:30']

def arp_scan():
    missedBus = False
    db_size = post.count()
    output = subprocess.check_output("sudo arp-scan -l", shell = True)
    for i in range(db_size):
        if addr[i] in output:
            msg = get_name(addr[i]) + " yay! safely got on the Bus"
            missedBus = False
            sendMessage(int(get_momsNo(addr[i])), msg)
        else:
            msg = get_name(addr[i]) + " ohhhh no!! Jim missed bus "
            missedBus = True
            childrens[i].miss += 1
            sendMessage(int(get_momsNo(addr[i])), msg)

    print(msg)
    sense.show_message(msg)

# ----- sending text messages -----
def sendMessage(momsNumber):
    # Your Account SID from twilio.com/console
    account_sid = "ACc4b0a542f3e1d7d8b3ac261343c5bf14"
    # Your Auth Token from twilio.com/console
    auth_token = "b0c3c9346758fd5fa9360081cd02113e"

    client = Client(account_sid, auth_token)
    message = client.messages.create(
        to=momsNumber,
        from_="+353766206050",
        body=msg)

if __name__ == "__main__":
    arp_scan()

```

# Parental Monitor Application

Dictionary for every children.

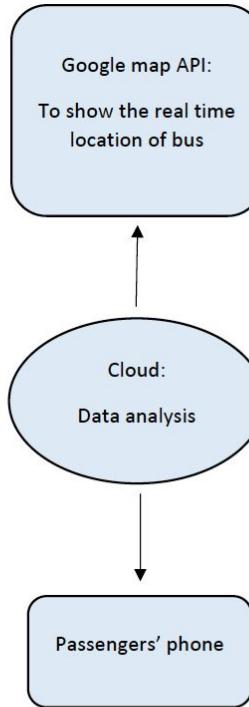
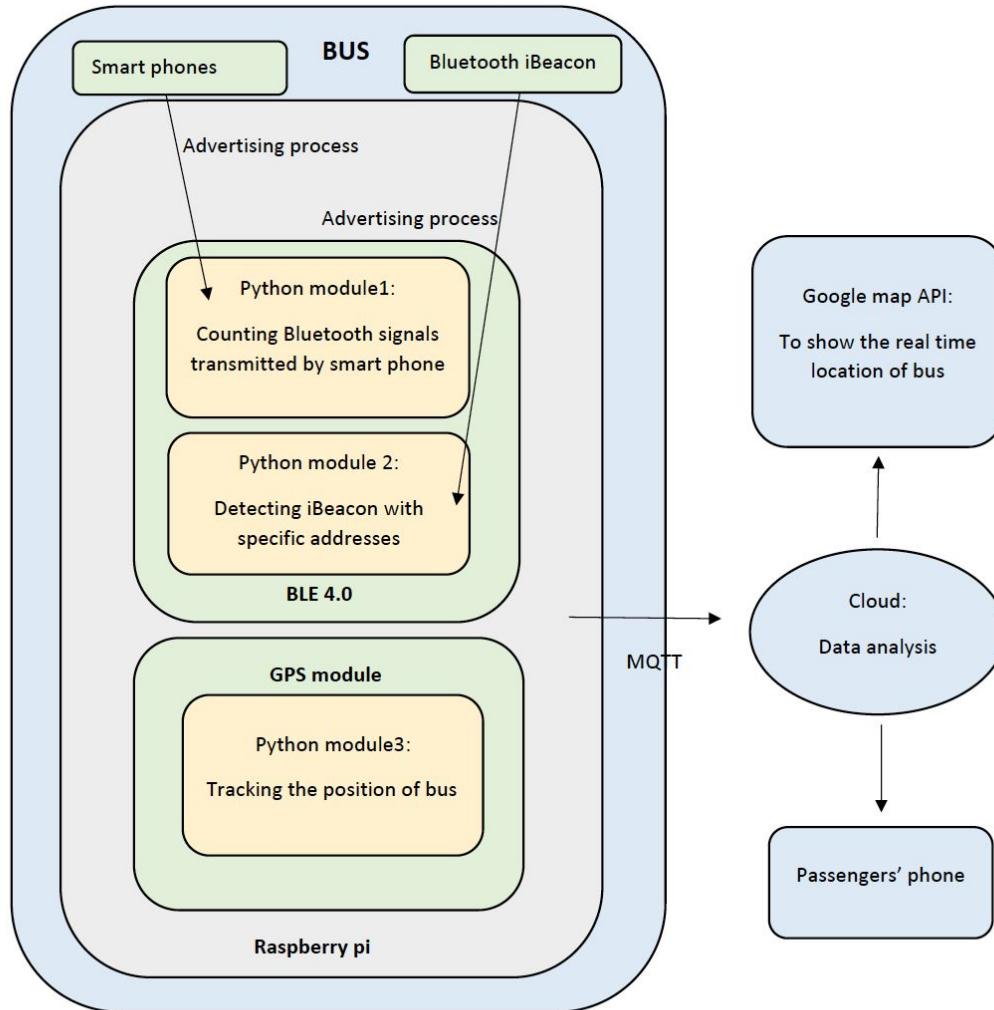
Do presence check

Send text message using Twilio

**Future add-ons:**

- send bus time in the next hour

(import datetime module)



# Diagram

## To build a bus tracking system

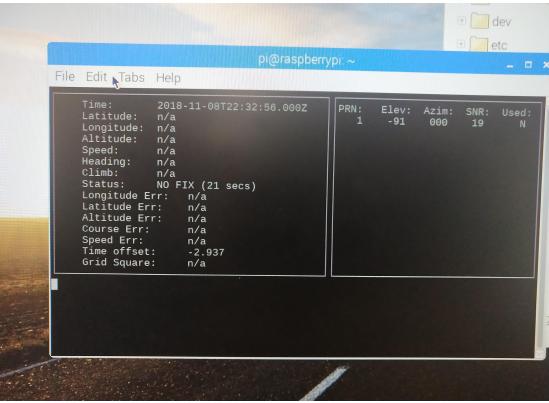
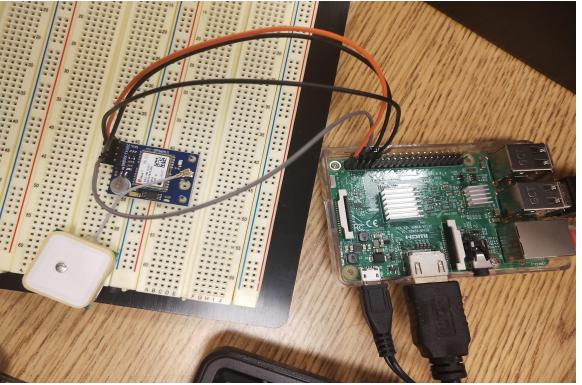
- GPS module and GSM module to establish a bus tracking system(Lee, Tewolde, and Kwon, 2014),
- Bluetooth for the bus tracking (A.Kumar, 2017).

## Number of people measurement in public place

- RFID to track bus cards (V. Kostakos 2010).
- Single camera fixed at an overhead position(T.Wang, 2016)
- BLE to measure the amounts of people.(P.Mathur, 2016)

**Previous  
work**





We set up GPS module(NEO 6m) on Raspberry pi , but some troubles were found on NEO 6M GPS module setting.

```
[CHG] Device 79:FA:1E:91:38:BD ManufacturerData Value: 0xb0  
[CHG] Device 79:FA:1E:91:38:BD ManufacturerData Value: 0x7b  
[CHG] Device 58:FF:F5:92:F8:D5 RSSI: -45  
[CHG] Device 66:44:02:9F:0C:2C RSSI: -47  
[CHG] Device 66:44:02:9F:0C:2C RSSI: -62  
[bluetooth]# devices  
Device 50:A4:62:7F:15:DD 50-AA-62-7F-15-DD  
Device EA:96:E2:F8:2C:55 Alta  
Device 60:6D:AC:FF:40:45 60-6D-AC-FF-40-45  
Device 79:2F:08:C4:C6:9A 79-2F-08-C4-C6-9A  
Device 79:FA:1E:91:38:BD 79-FA-1E-91-38-BD  
Device 5F:D5:A8:B1:6D:DC 5F-D5-A8-B1-6D-DC  
Device A4:5E:60:E3:9D:F0 A4-5E-60-E3-9D-F0  
Device 66:44:02:9F:0C:2C 66-44-02-9F-0C-2C  
Device 58:FF:F5:92:F8:D5 58-FF-F5-92-F8-D5  
Device 42:26:1D:6D:D7:F3 42-26-1D-6D-D7-F3  
Device 52:8A:04:1E:AF:25 52-8A-04-1E-AF-25  
Device 70:DA:4D:0A:D5:15 70-DA-4D-CA-D5-15  
[CHG] Device 79:FA:1E:91:38:BD RSSI: -67  
[CHG] Device 79:FA:1E:91:38:BD ManufacturerData Key: 0x004c
```

Bluetooth 4.0 on pi works as a bluetooth signal scanner.

# Current Status

1. GPS connection:

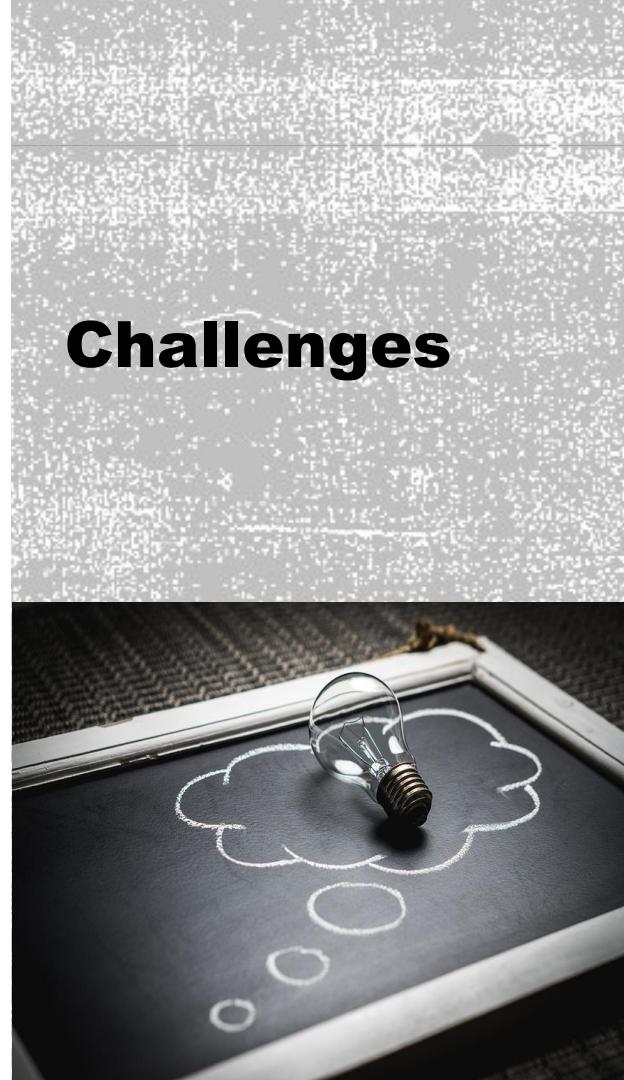
UART or USB

2 Package the data and send to the cloud

3 How to analyze those data on the cloud

NEXT:

- Finish setting the GPS module.
- Connect GPS module with Google map API.
- Start detect Bluetooth iBeacon signals on raspberry pi.



Thank You!