

Assignment 3 - A Smart Home with Two Devices to a Cloud

Introduction to IoT

Fall 2021



Outline

- Lab Objectives
- Thingspeak
- How to use Thingspeak?
 - Collect data in a channel
 - Analyze data (Computation in Thingspeak)
 - Act on data
- System Specification
 - Topology
 - Section 1 – Temperature Monitoring
 - Section 2 – Automatic LED
 - Section 3 – Intruder Detection
- Assignment 3 Specifications

Lab Objectives

- Simulate the smart home system
 - Use 2 Raspberry Pi's
 - Connect 4 types of sensors
 - Connect 3 types of actuators
 - Collaborate with Thingspeak
- Capable to collaborate with more than one Raspberry Pi
- Capable to make the complex IoT system

What is Thingspeak?

- Cloud based data service platform for IoT devices.



Collect

Send sensor data privately to the cloud.



Analyze

Analyze and visualize your data with MATLAB.



Act

Trigger a reaction.

ThingSpeak Features

- Collect data in private channels
- Share data with public channels
- RESTful and MQTT APIs
- MATLAB® analytics and visualizations
- Event scheduling
- Alerts
- App integrations

Works With

- MATLAB® & Simulink®
- Arduino®
- Particle devices
- ESP8266 and ESP32 Modules
- Raspberry Pi™
- LoRaWAN®
- Things Network
- Senet
- Libelium
- Beckhoff

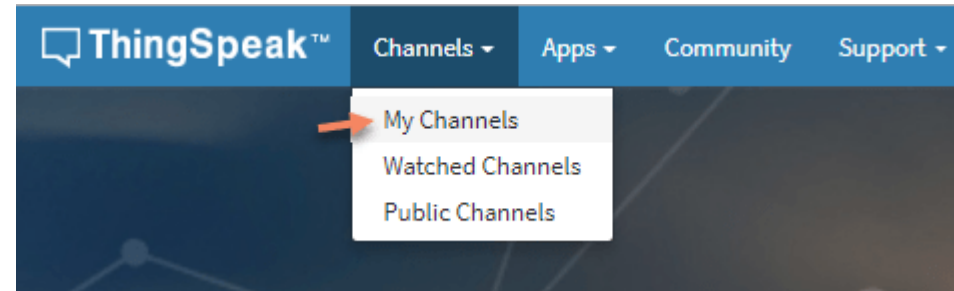
How to Use Thingspeak ?

- Register an account (<https://thingspeak.com>).
- Verify and complete the registration process
- Configure several things:
 - Collect data in a channel
 - Add the channels
 - Get API Keys
 - Connect the device to Thingspeak
 - Set the environment
 - Analyze data (Computation in Thingspeak)
 - Read the sensors data
 - Do the computation
 - Trigger the action
 - Act on data
 - Schedule the computation in Thingspeak
- Detail documentation about Thingspeak
 - <https://www.mathworks.com/help/thingspeak/getting-started-with-thingspeak.html>

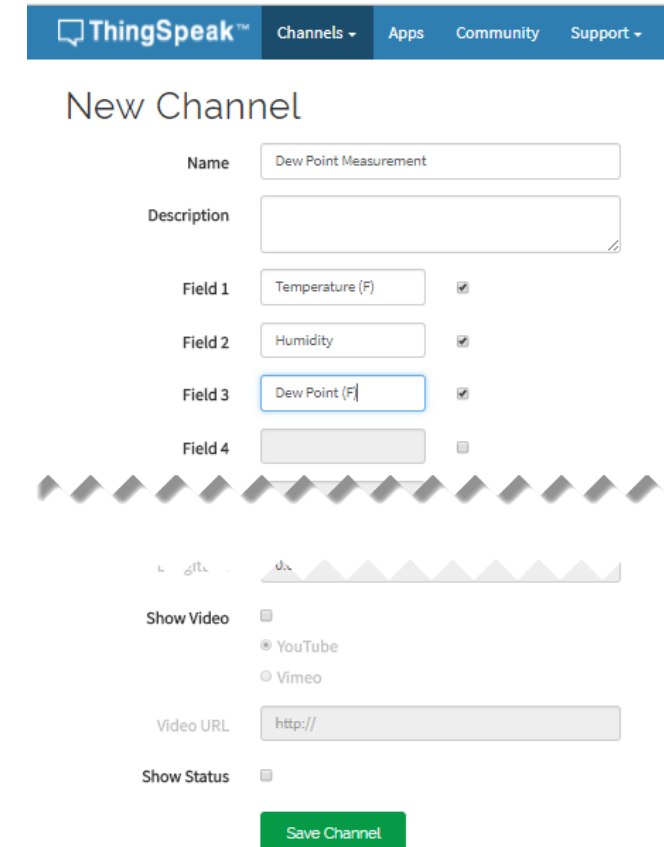
Thingspeak - Collect Data in a Channel

- **Create a Channel**

- Sign in to Thingspeak
- Click **Channels** > **MyChannels**.



- On the Channels page, click New Channel
- Check the boxes next to Fields 1–3
- Fill the boxes with your channel

A screenshot of the 'New Channel' form in the Thingspeak web interface. The form has a blue header with the Thingspeak logo and navigation links. The main form area is white. It includes a 'Name' field with 'Dew Point Measurement' entered, a 'Description' field, and four 'Field' sections. Field 1 is 'Temperature (F)' with a checked checkbox. Field 2 is 'Humidity' with a checked checkbox. Field 3 is 'Dew Point (F)' with a checked checkbox. Field 4 is empty with an unchecked checkbox. Below the fields are sections for 'Show Video' (with radio buttons for YouTube and Vimeo) and 'Show Status' (with a checkbox). A green 'Save Channel' button is at the bottom.

Thingspeak - Analyze Data

- Go to the **Apps** tab, and click **MATLAB Analysis**.
- Click **New**. Select the **Custom** template, and click **Create**.
- Write your MatLab code in MATLAB Code Box

Apps / MATLAB Analysis / Update LED based on Temperature / Edit

Name

Update LED based on Temperature

MATLAB Code

```
1 % Template MATLAB code for reading data from a private channel, analyzing
2 % the data and storing the analyzed data in another channel.
3
4 % Prior to running this MATLAB code template, assign the channel variables.
5 % Set 'readChannelID' to the channel ID of the channel to read from. Since
6 % this is a private channel, also assign the read API Key to the 'readAPIKey'
7 % variable. You can find the read API Key on the right side pane of this page.
8
9 % To store the analyzed data, you will need to write it to a channel other
10 % than the one you are reading data from. Assign this channel ID to the
11 % 'writeChannelID' variable. Also assign the write API Key to the
12 % 'writeAPIKey' variable below. You can find the write API Key in the right
13 % side pane of this page.
14
15 % TODO - Replace the [] with channel ID to read data from:
16 readChannelID = 1575130;
17 % TODO - Enter the Read API Key between the '' below:
18 readAPIKey = 'LA2OWORT3AFLBRDA';
19
20 % TODO - Replace the [] with channel ID to read data from:
21 readChannelID2 = 1575191;
22 % TODO - Enter the Read API Key between the '' below:
23 readAPIKey2 = 'LPFJ3W7V07DL2J8I';
```

MATLAB in ThingSpeak

[Apps](#) / [MATLAB Analysis](#) / New

Templates:

- ☒ Custom (no starter code)
- ☐ Get data from a private channel
- ☐ Get data from a public channel
- ☐ Get data from a webpage

Examples: Sample code to analyze and transform data

- ☐ Calculate and display average humidity
- ☐ Calculate wind chill and update channel
- ☐ Remove outliers from wind speed data
- ☐ Convert temperature units
- ☐ Calculate high and low temperatures
- ☐ Read Channel to Trigger Email
- ☐ Replace missing values in data
- ☐ Analyze text for the most common color
- ☐ Read live web data for vessels at the port of Boston
- ☐ Scrape web temperature data

Create

Help

Templates

MATLAB Analysis templates provide sample MATLAB code for analyzing data and writing it to a ThingSpeak channel. If you are new to MATLAB, you can learn interactively at [MATLAB Academy](#).

Examples

To see MATLAB Analysis in action, select the example and click **Create**.

These examples read data from public ThingSpeak channels:

- [Calculate and display average humidity](#) over the last 60 minutes, and write the results to a new channel.
- [Calculate wind chill and update channel](#) by writing to a new channel at regular intervals.
- [Remove outliers from wind speed data](#) over the past six hours, and write data to a new channel.
- [Convert temperature units](#) from Fahrenheit to Celsius, and write the results to a new channel.
- [Calculate high and low temperatures](#) over the past 24 hours, and write the data with timestamps to a new channel.
- [Trigger Email](#) by analyzing daily soil moisture values.
- [Replace missing values in data](#) of a weather channel, and clean the data using a list-wise deletion algorithm. Then display the missing values, or write data to a new channel.
- [Analyze text for the most common color](#) requested on the public Cheerlights channel, and write the data to a new channel.

These examples scrape data from websites:

- [Read live web data for vessels at the port of Boston](#) from the MarineTraffic website, count the number of vessels, and write the data to a new channel.
- [Scrape web temperature data](#) from the National Weather Service website, and write the data to a new channel.

New to MATLAB?

- Watch [Getting Started with MATLAB](#) video tutorial.
- Learn interactively at [MATLAB Academy](#).

Important MATLAB Syntax in ThingSpeak

- **thingSpeakRead**

- humidity1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 2);

- **thingSpeakWrite**

- thingSpeakWrite(writeChannelID, ldrData, 'WriteKey', writeAPIKey, 'Fields', 1);
 - thingSpeakWrite(writeChannelID, [ldrData, analyzedData], 'WriteKey', writeAPIKey, 'Fields', [1,2]);

- **MATLAB's syntax is similar to Python's. Want to know little bit more?**

- <https://www.mathworks.com/videos/getting-started-with-matlab-1564521672719.html>

Thingspeak - Act on Data

- Go to **Apps > React**, and click **New React**.
- In this example, Thingspeak will run the MatLab code if the value is greater than 0

Apps / React / Temperature_LED / Edit

React Name	<input type="text" value="Temperature_LED"/>
Condition Type	<input type="text" value="Numeric"/>
Test Frequency	<input type="text" value="On Data Insertion"/>
Condition	<div>If channel <input type="text" value="Raspberry Pi (1575130)"/></div> <div>field <input type="text" value="3 (LDR)"/></div> <div><input type="text" value="is greater than"/></div> <div><input type="text" value="0"/></div>
Action	<input type="text" value="MATLAB Analysis"/>
Code to execute	<input type="text" value="Update LED based on Temperature"/>
Options	<div><input type="radio"/> Run action only the first time the condition is met</div> <div><input checked="" type="radio"/> Run action each time condition is met</div>
<input type="button" value="Save React"/>	

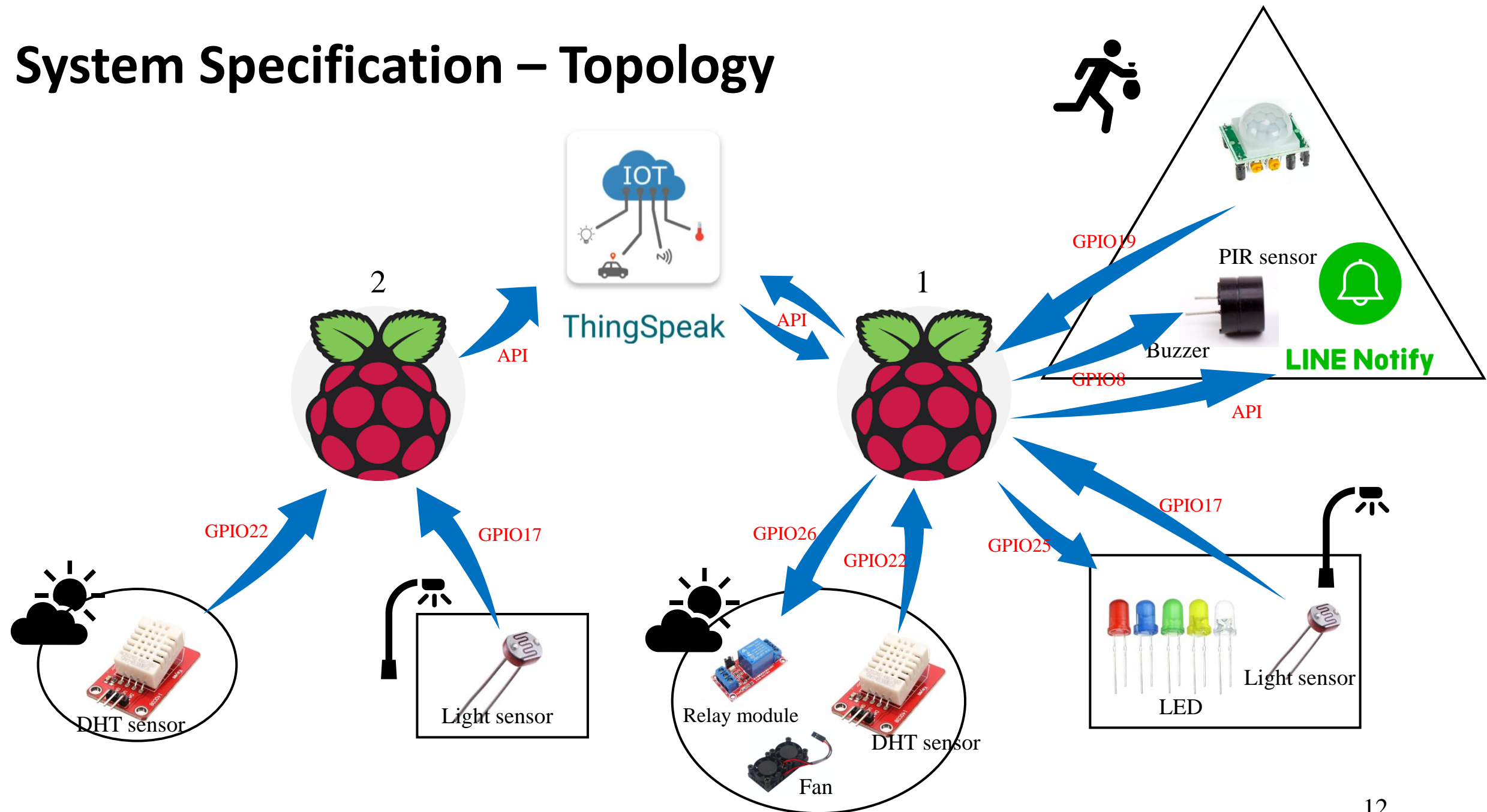
System Specification - How to Build it?

Section 1 : Temperature/Humidity is too high → Turn on the fan.

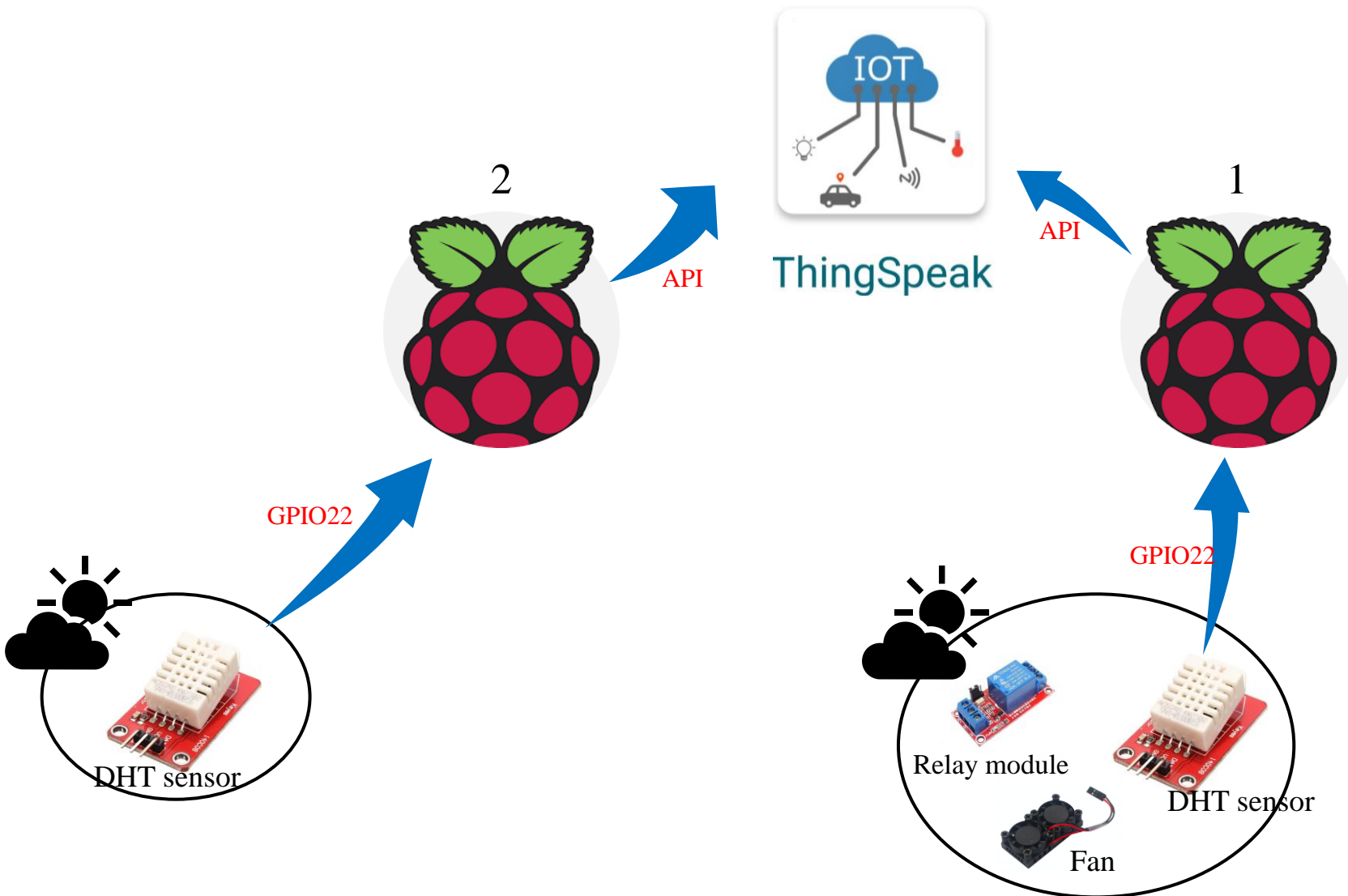
Section 2 : The room is dark → Turn ON the LED.

Section 3 : Intruder is detected → Send a Line Notification + Play the buzzer.

System Specification – Topology



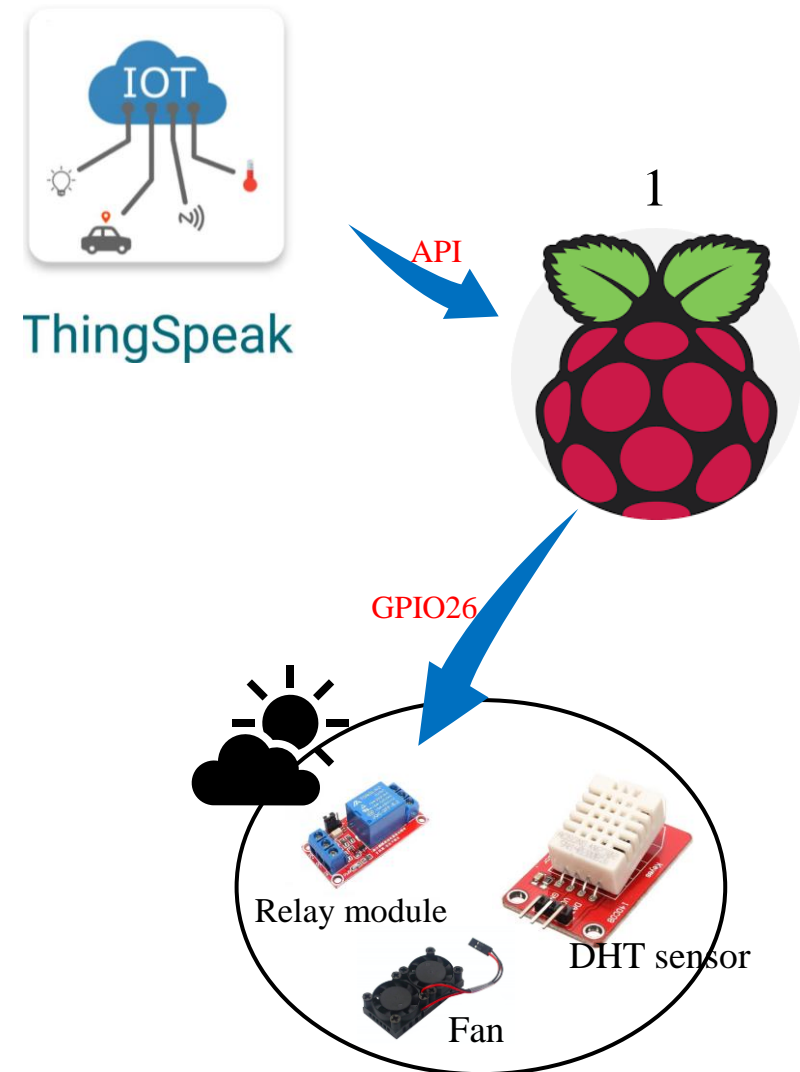
System Specification – Section 1 (1/3)



1. Measure temp/humidity by DHT sensors.
2. 1st and 2nd Raspberry Pi send the data to Thingspeak.

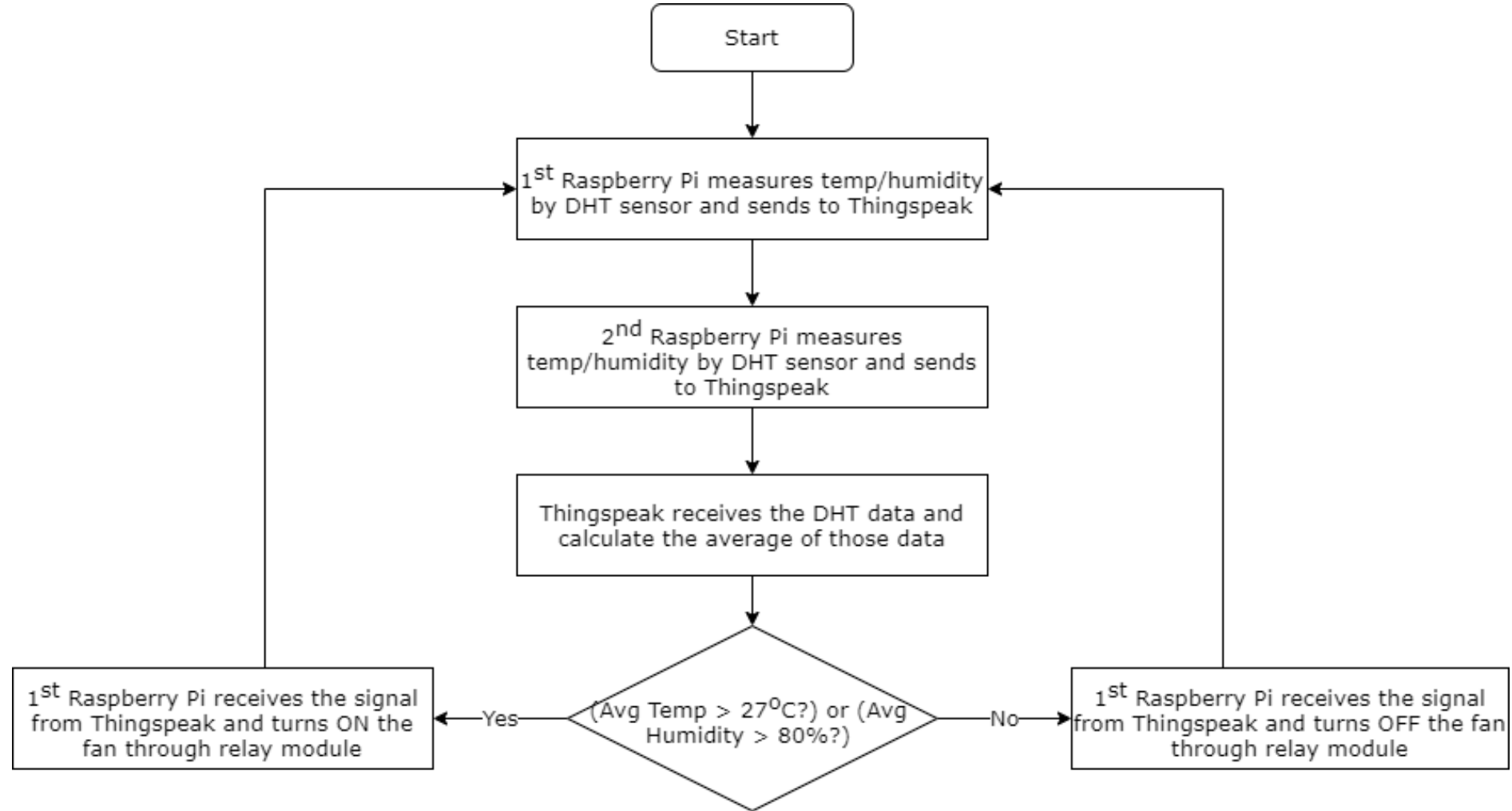
System Specification – Section 1 (2/3)

1. If (Avg Temp > 27°C) or (Avg Humidity > 80%), Thingspeak sets ON the fancontrol data channel.
2. 1st Raspberry Pi receives the signal, turns ON the fan through relay module.



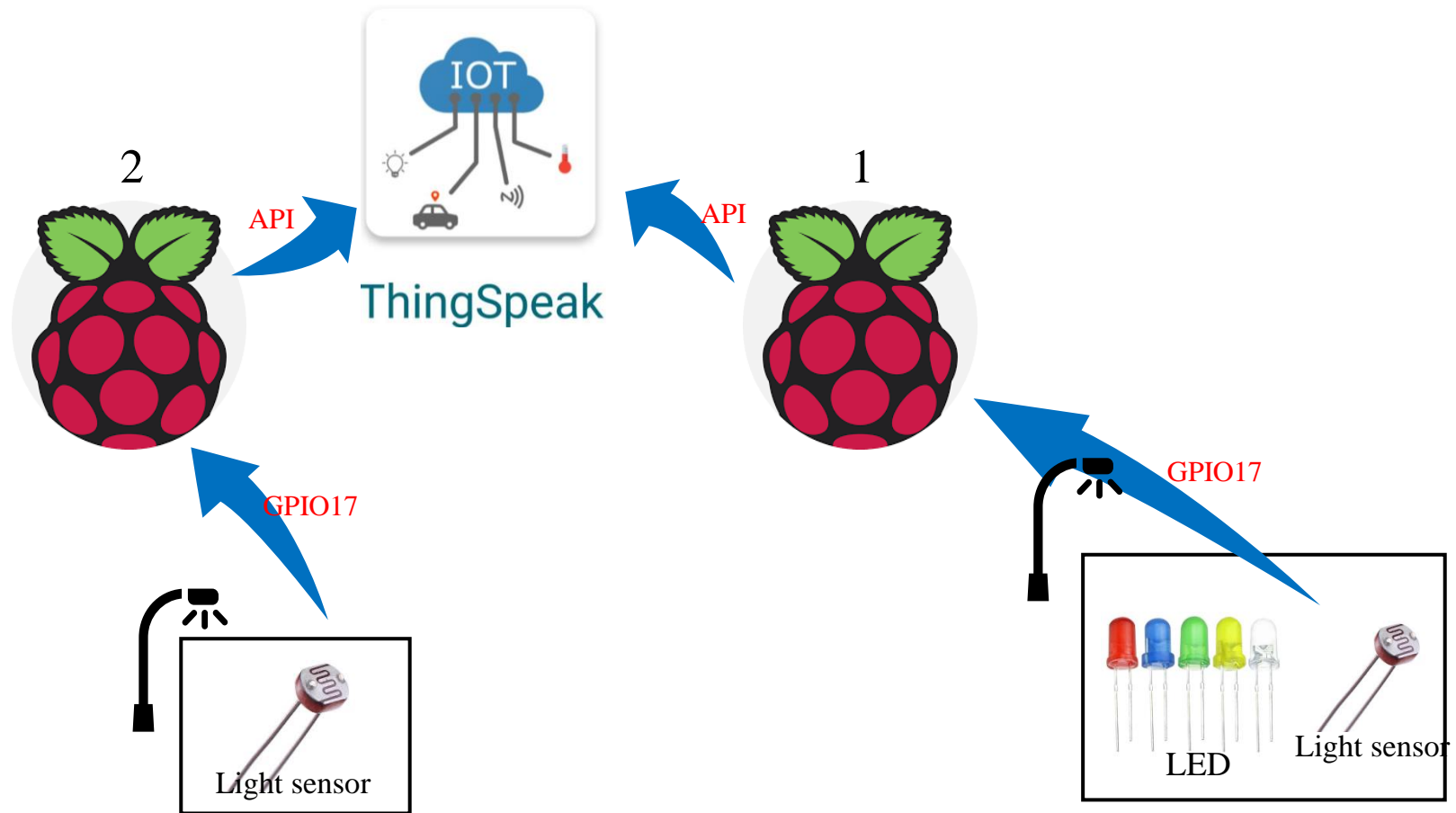
System Specification – Section 1 (3/3)

Section 1 Process Flowchart



System Specification – Section 2 (1/3)

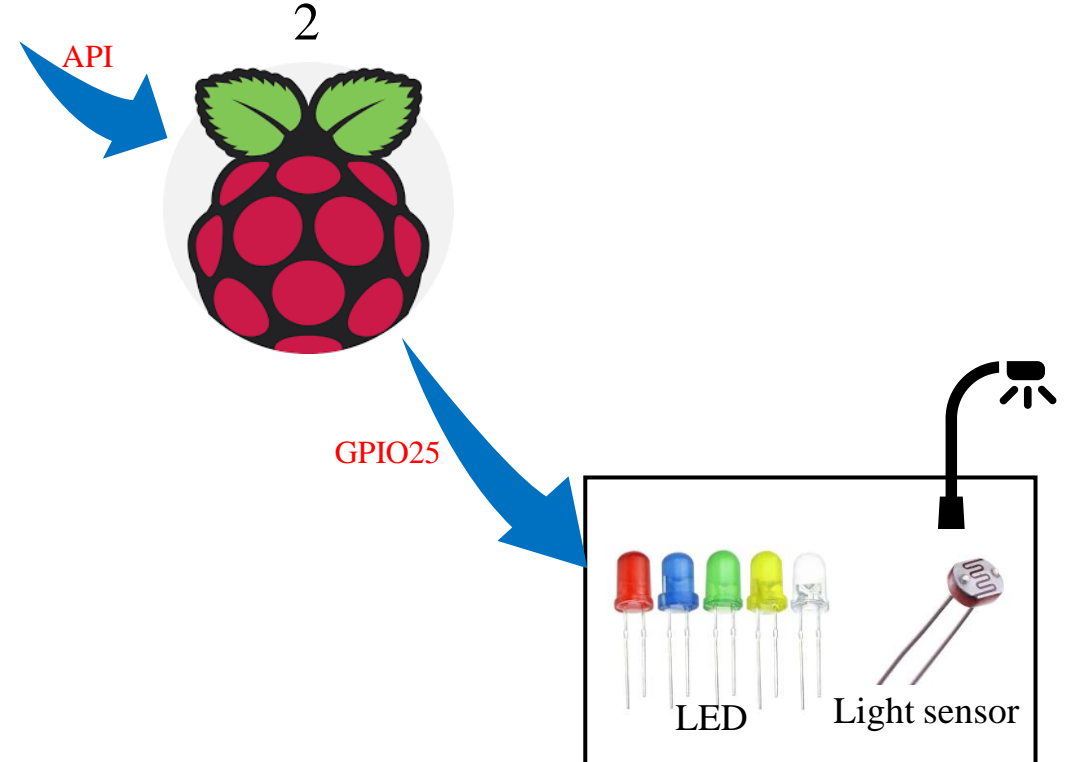
1. Measure light level by light sensors in 1st and 2nd Raspberry Pi.
2. 1st and 2nd Raspberry Pi send the data to Thingspeak.



System Specification – Section 2 (2/3)

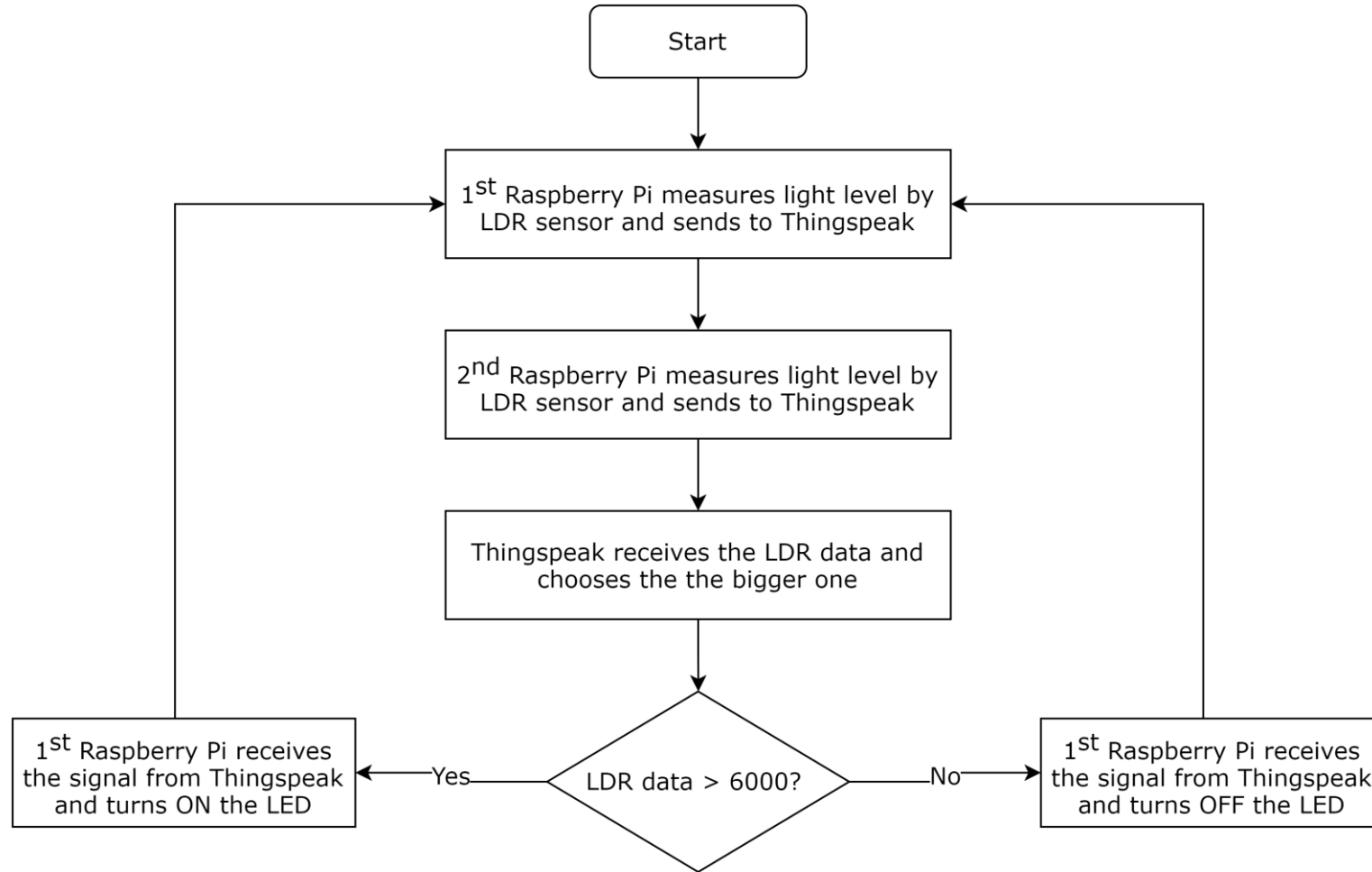


Thingspeak receives those data and chooses the darker one's value, then turn ON the LED through 1st Raspberry Pi.



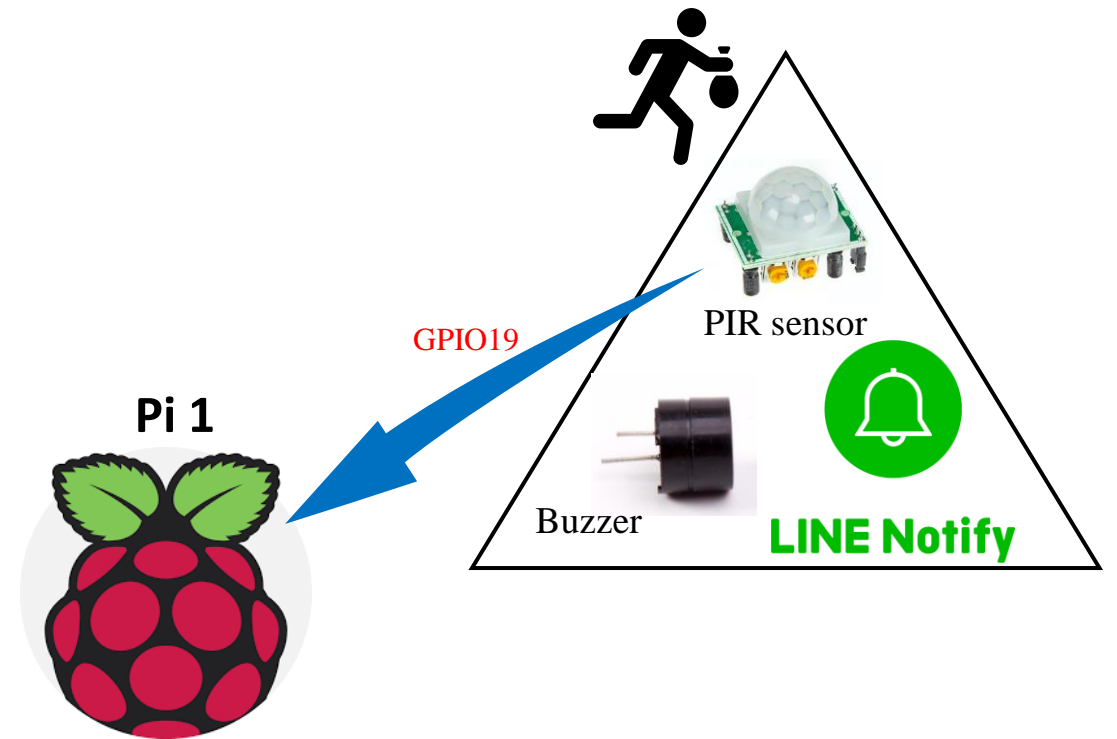
System Specification – Section 2 (3/3)

Section 2 Process Flowchart



System Specification – Section 3 (1/3)

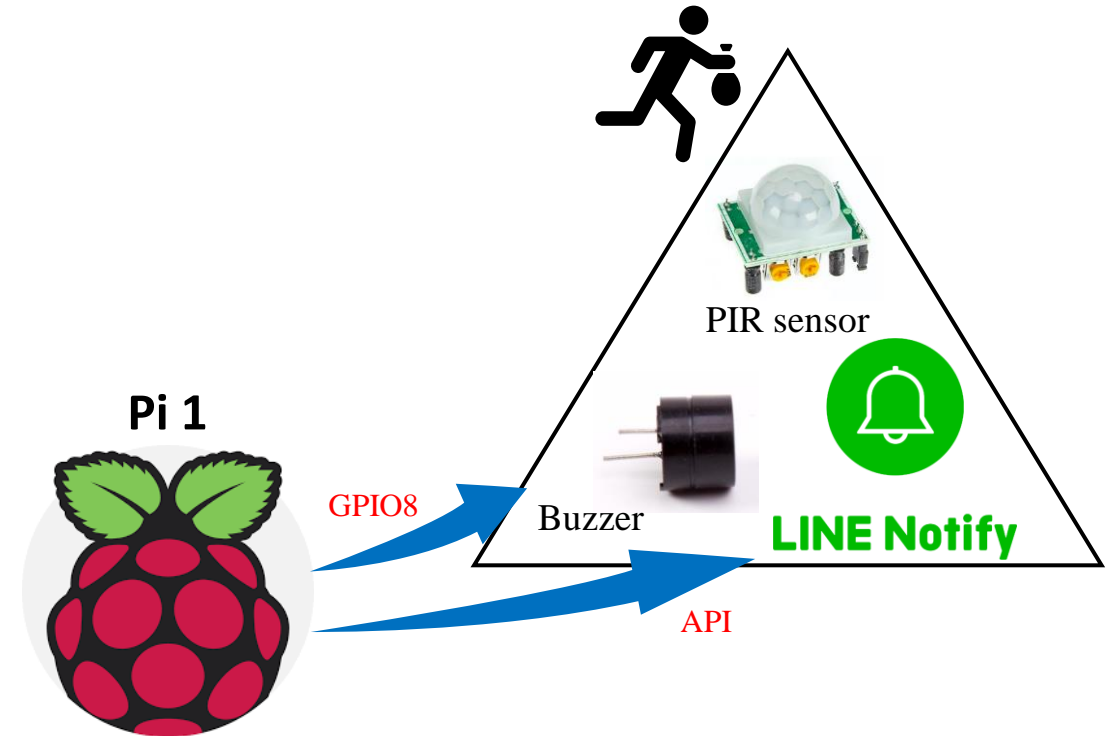
1. This alarm system keeps running.
2. Detect an intruder by PIR sensor.



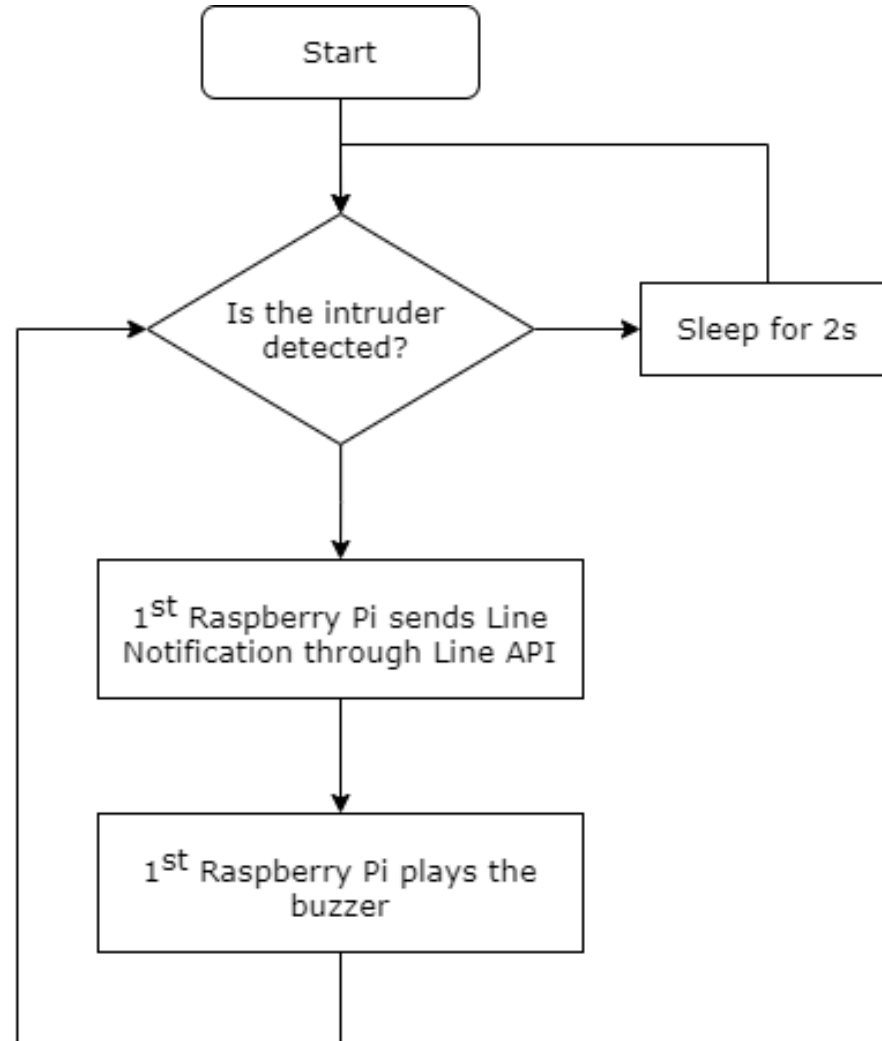
System Specification – Section 3 (2/3)

If an intruder is detected,

1st Raspberry Pi plays the buzzer to give a warning and sends a line notification to the owner immediately.



System Specification – Section 3 (3/3)



System Components

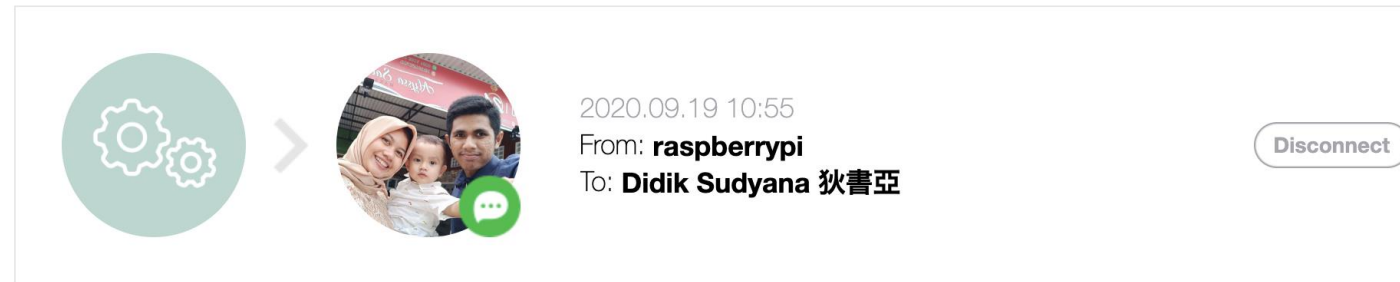
Hardware		
Component Name	Model	Functionality
2x Raspberry Pi 4	Model B, 1 GB RAM	
1x LED	Red	Visible LED
2x LDR Sensor	GL55	Measure the light level
2x Capacitor	1 μ F, 50 V	Construct RC circuits
3x Resistor	1 k Ω	Adjust the voltage
1x PIR sensor	HW-416-B	Detect the motion
1x Buzzer	G-S&S	Alarm the intruder
2x DHT sensor	AM2302	Measure temp/humidity
1x Fan	Raspberry Pi cooling fan	Cool down the room
1x Relay module	4-channel 5V	Adjust the voltage
Bread board	EIC-1104	Build electronic circuits
Dupont wire	Male to Male/Male to Female/Female to Female	Construct circuits

How to Send Notification via Line

- Register at <https://notify-bot.line.me/my/>
- Log in by using your Line account
- Click “Generate Token”

Connected services

Your connected services. Press the Disconnect button to disconnect any service.



Generate access token (For developers)

By using personal access tokens, you can configure notifications without having to add a web service.

Generate token

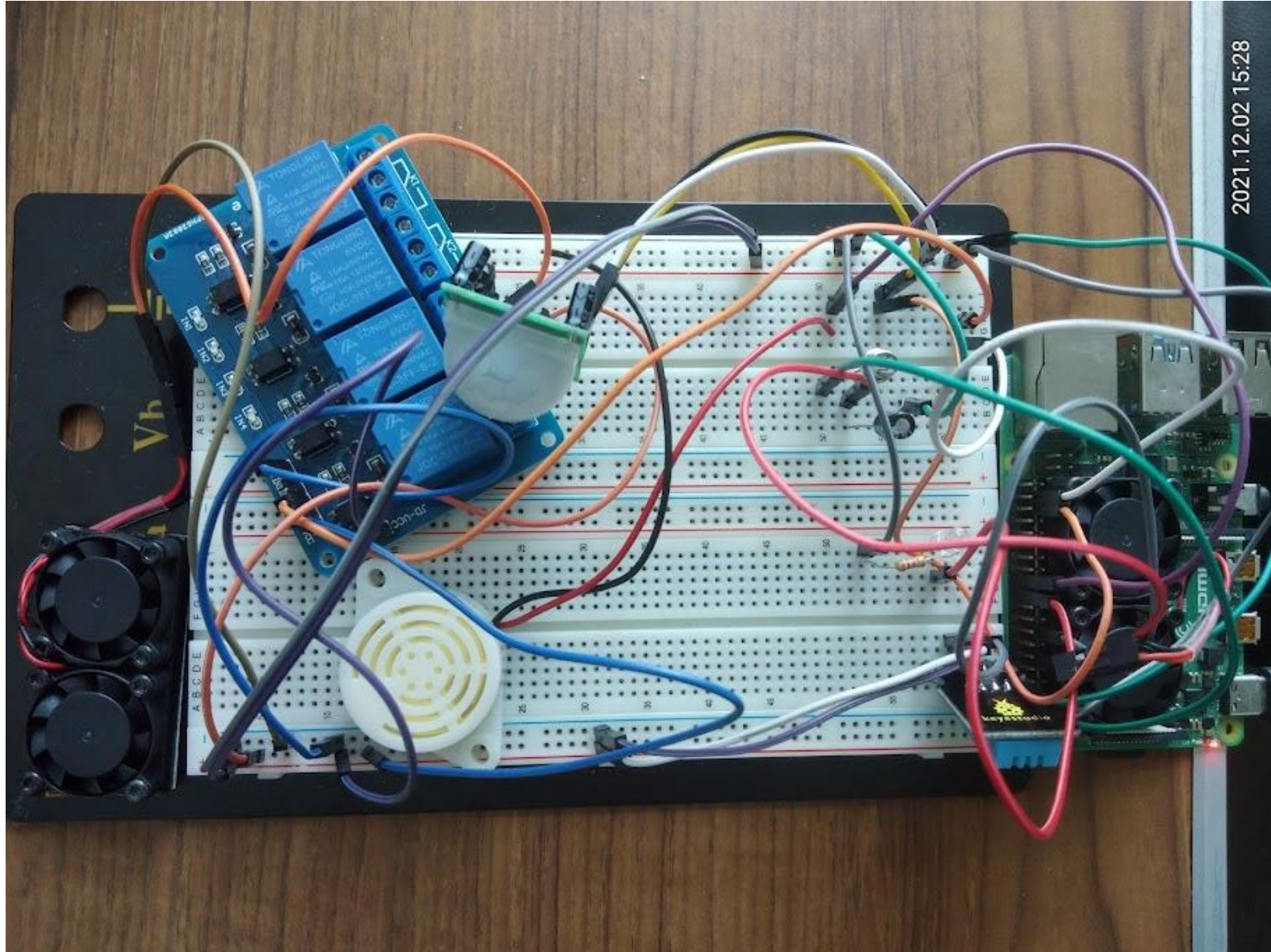
LINE Notify API Document

Thingspeak Restriction

- “Requests are too frequent” when Thingspeak runs MatLab code
 - Solution : Raspberry Pi sends the sensor data at once every 20 seconds.

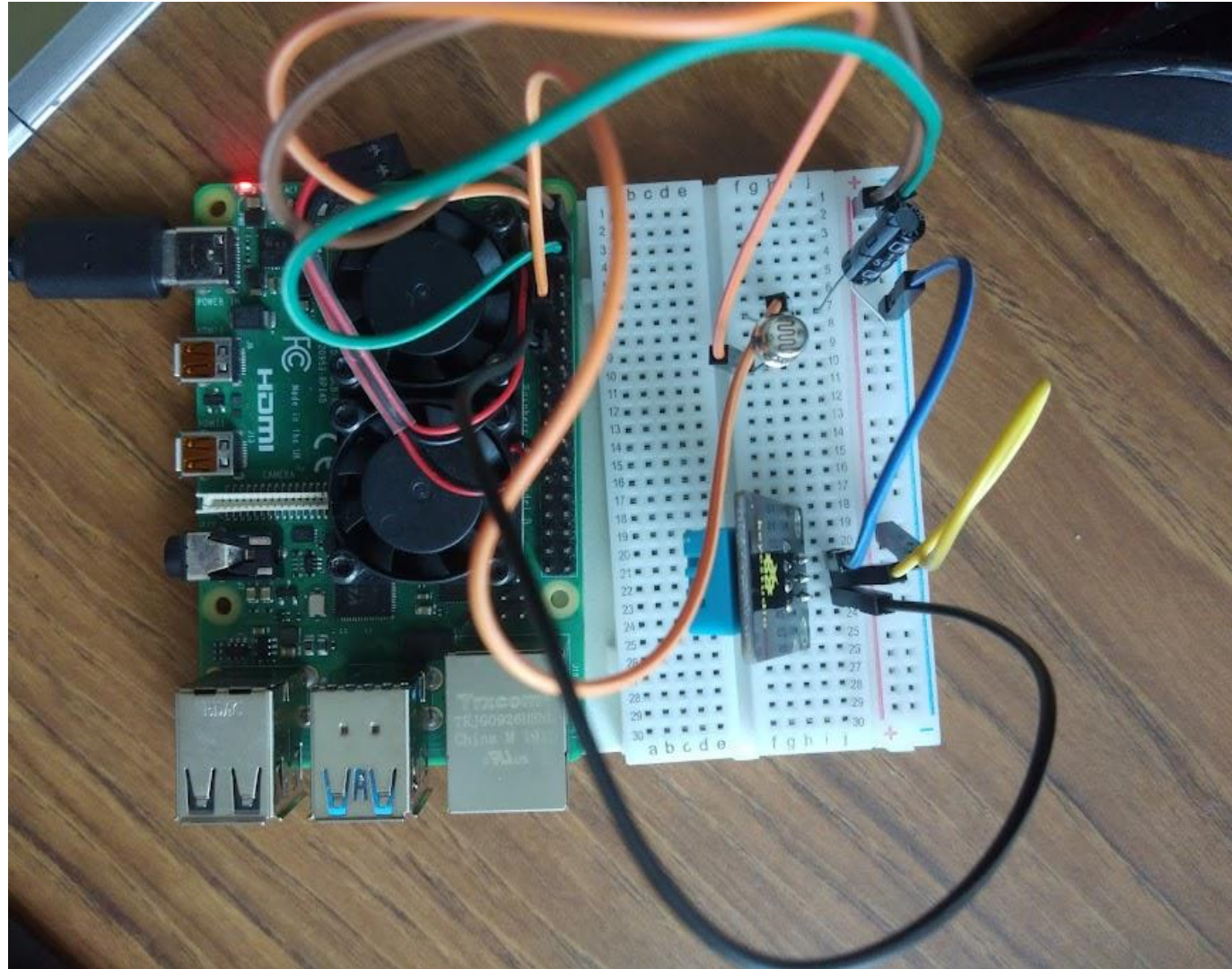
Implementation Photos (1/2)

1st Raspberry Pi



Implementation Photos (2/2)

2nd Raspberry Pi



1st Raspberry Pi (Source Code) (1/3)

```
import Adafruit_DHT
import RPi.GPIO as GPIO
from time import sleep
import sys
import random
import urllib.request
import requests
import threading
import lineTool

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

fan_pin = 26
LIGHT_PIN = 25
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 22
ldr_threshold = 30000
LDR_PIN = 17
PIR_pin = 19
GPIO.setup(PIR_pin, GPIO.IN)

token="INSERT_YOUR_TOKEN_HERE"
msg="Intruder detected!"

GPIO.setup(fan_pin, GPIO.OUT)
```

Copy and paste
your LINE token
here

```
def thingspeak_post_temphumd(temp, humd):
    URL='https://api.thingspeak.com/update?api_key='
    KEY= INSERT_YOUR_KEY_HERE '
    HEADER='&field1={}&field2={}'.format(temp,humd)
    NEW_URL=URL+KEY+HEADER
    print(NEW_URL)
    data=urllib.request.urlopen(NEW_URL)
    print(data)
```

Publish
temperature
data to the
cloud

```
def thingspeak_post_ldr(ldr):
    URL='https://api.thingspeak.com/update?api_key='
    KEY='INSERT_YOUR_KEY_HERE'
    HEADER='&field3={}'.format(ldr)
    NEW_URL=URL+KEY+HEADER
    print(NEW_URL)
    data=urllib.request.urlopen(NEW_URL)
    print(data)
```

Publish LDR data
to the cloud

```
def read_data_fan():
    URL='https://api.thingspeak.com/channels/1591839/fields/2.json?api_key='
    KEY='INSERT_YOUR_KEY_HERE'
    HEADER='&results=2'
    NEW_URL=URL+KEY+HEADER
    get_data=requests.get(NEW_URL).json()
    channel_id=get_data['channel']['id']

    feild_1=get_data['feeds']
    t=[]
    for x in feild_1:
        t.append(x['field2'])

    return(t[1])
```

Receive the
signal to control
fan from the
cloud

1st Raspberry Pi (Source Code) (2/3)

```
def read_data_led():
    URL='https://api.thingspeak.com/channels/1591839/fields/1.json?api_key='
    KEY=INSERT_YOUR_KEY_HERE
    HEADER='&results=2'
    NEW_URL=URL+KEY+HEADER

    get_data=requests.get(NEW_URL).json()
    channel_id=get_data['channel']['id']

    feild_1=get_data['feeds']

    t=[]
    for x in feild_1:
        t.append(x['field1'])

    return(t[1])

def temp_humd():
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
    if humidity is not None and temperature is not None:
        print("Temp={0:0.1f}*C Humidity={1:0.1f}%".format(temperature, humidity))
    else:
        print("Failed to retrieve data from humidity sensor")

    return humidity, temperature

def turnonFAN(fan_input):
    GPIO.output(fan_input, False)

def turnoffFAN(fan_input):
    GPIO.output(fan_input, True)
```

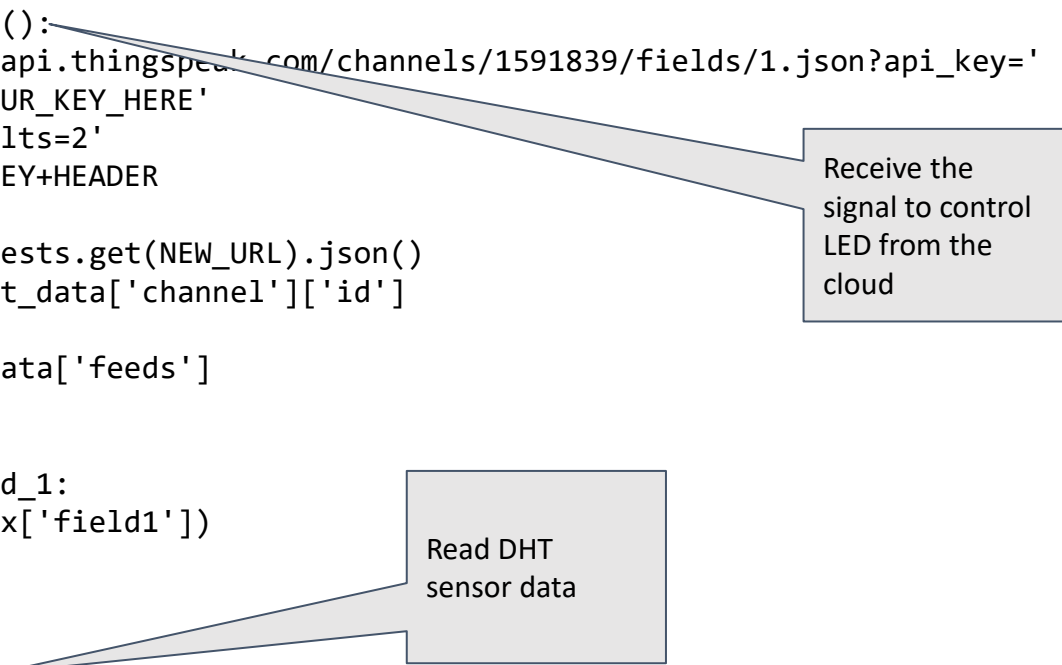


Diagram illustrating the code flow and sensor data retrieval:

- Callout: Receive the signal to control LED from the cloud (points to the URL and KEY assignment in `read_data_led()`)
- Callout: Read DHT sensor data (points to the `temp_humd()` function)

```
def switchOnLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, True)

def switchOffLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)

def readLDR(PIN):
    reading=0
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)
    time.sleep(0.1)
    GPIO.setup(PIN, GPIO.IN)
    while (GPIO.input(PIN)==False):
        reading=reading+1
    return reading

def linenotify(a,b):
    lineTool.lineNotify(a,b)

def play(p, frequency, tempo):
    p.ChangeFrequency(frequency)
    time.sleep(0.5 * tempo)
```

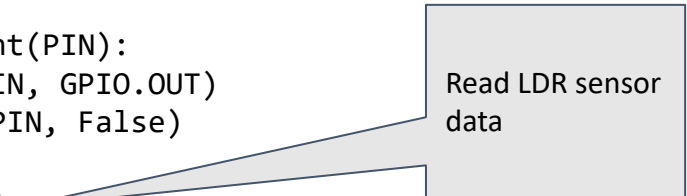
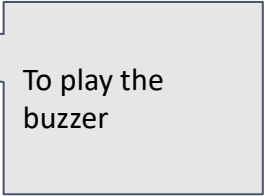


Diagram illustrating the code flow and sensor data retrieval:

- Callout: Read LDR sensor data (points to the `readLDR(PIN)` function)

1st Raspberry Pi (Source Code) (3/3)

```
def bell():
    C4 = 262 # Do
    E4 = 330 # Mi
    music = [C4, E4]
    M_1 = C4
    M_3 = E4
    GPIO.setup(8, GPIO.OUT)
    p = GPIO.PWM(8, 50)
    p.start(15) # 0 <= DV <= 100
    play(p, M_3, 1)
    play(p, M_1, 1)
    p.stop
```



To play the
buzzer

```
while True:
    try:
        humidity, temperature = temp_humd()
        thingspeak_post_temphumd(temperature, humidity)
        ldr_reading = readLDR(LDR_PIN)
        thingspeak_post_ldr(ldr_reading)
        fan = read_data_fan()
        if(fan == '0'):
            turnoffFAN(fan_pin)
            print("Turns Off")
        elif(fan == '1'):
            turnonFAN(fan_pin)
            print("Turns On")
        time.sleep(20)
        led = read_data_led()
        if (led == '1'):
            print("Turns On LED")
            switchOnLight(LIGHT_PIN)
        else:
            print("Turns Off LED")
            switchOffLight(LIGHT_PIN)

        i=GPIO.input(PIR_pin)
        if(i==0):
            print("No intruder.")
            time.sleep(2)
        elif(i==1):
            print("Intruder detected.")
            linenotify(token,msg)
            bell()
            time.sleep(1)

    except KeyboardInterrupt:
        exit()
```

2nd Raspberry Pi (Source Code)

```
import Adafruit_DHT
import RPi.GPIO as GPIO
import time
from time import sleep
import sys
import random
import urllib.request
import requests
import threading
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
```

```
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 22
ldr_threshold = 30000
LDR_PIN = 17
```

```
def thingspeak_post_temphumd(temp, humd):
    URL='https://api.thingspeak.com/update?api_key='
    KEY=' INSERT_YOUR_KEY_HERE '
    HEADER='&field1={}&field2={}'.format(temp, humd)
    NEW_URL=URL+KEY+HEADER
    print(NEW_URL)
    data=urllib.request.urlopen(NEW_URL)
    print(data)
```

Publish
temperature
data to the
cloud

```
def thingspeak_post_ldr(ldr):
    URL='https://api.thingspeak.com/update?api_key='
    KEY=' INSERT_YOUR_KEY_HERE '
    HEADER='&field3={}'.format(ldr)
    NEW_URL=URL+KEY+HEADER
    print(NEW_URL)
    data=urllib.request.urlopen(NEW_URL)
    print(data)
```

Publish LDR data
to the cloud

```
def read_data_thingspeak():
```

```
    URL='https://api.thingspeak.com/channels/1575191/fields/1.json?api_
    _key='
    KEY=' INSERT_YOUR_KEY_HERE '
    HEADER='&results=2'
    NEW_URL=URL+KEY+HEADER
```

```
    get_data=requests.get(NEW_URL).json()
    channel_id=get_data['channel']['id']
```

```
    feild_1=get_data['feeds']
```

```
    t=[]
    for x in feild_1:
        t.append(x['field1'])
```

```
    return(t[1])
```

Read DHT
sensor data

```
def temp_humd():
    print("Test")
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR,
    DHT_PIN)
```

```
    if humidity is not None and temperature is not None:
        print("Temp={0:0.1f}*C
    Humidity={1:0.1f}%".format(temperature, humidity))
```

```
    else:
        print("Failed to retrieve data from humidity sensor")
```

```
    return humidity, temperature
```

2nd Raspberry Pi (Source Code)

```
def readLDR(PIN):
    reading=0
    print("Test")
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)
    time.sleep(0.1)
    GPIO.setup(PIN, GPIO.IN)
    while (GPIO.input(PIN)==False):
        reading=reading+1
    return reading

while True:
    try:
        humidity, temperature = temp_humd()
        thingspeak_post_temphumd(temperature, humidity)
        ldr_reading = readLDR(LDR_PIN)
        thingspeak_post_ldr(ldr_reading)
        print(ldr_reading)
        time.sleep(1)

    except KeyboardInterrupt:
        exit()
```

Source Code in Thingspeak

```
% TODO - Replace the [] with channel ID to read data from:
readChannelID = 1575130;
% TODO - Enter the Read API Key between the '' below:
readAPIKey = ' INSERT_YOUR_KEY_HERE ';

% TODO - Replace the [] with channel ID to read data from:
readChannelID2 = 1575191;
% TODO - Enter the Read API Key between the '' below:
readAPIKey2 = ' INSERT_YOUR_KEY_HERE ';

% TODO - Replace the [] with channel ID to write data to:
writeChannelID = 1591839;
% TODO - Enter the Write API Key between the '' below:
writeAPIKey = ' INSERT_YOUR_KEY_HERE ';

%% Read Data %%
humidity1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 2);
temp1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 1);
ldr1 = thingSpeakRead(readChannelID, 'ReadKey', readAPIKey, 'Fields', 3);

%% Read Data 2%%
humidity2 = thingSpeakRead(readChannelID2, 'ReadKey', readAPIKey2, 'Fields', 2);
temp2 = thingSpeakRead(readChannelID2, 'ReadKey', readAPIKey2, 'Fields', 1);
ldr2 = thingSpeakRead(readChannelID2, 'ReadKey', readAPIKey2, 'Fields', 3);
```

```
%% Analyze Data %%
% Add code in this section to analyze data and store the
result in the
% 'analyzedData' variable.
avg_humd = (humidity1 + humidity2) / 2
avg_temp = (temp1 + temp2) / 2

if(avg_humd>80 | avg_temp > 27)
    analyzedData = 1
else
    analyzedData = 0
end

if(ldr1 > 6000 | ldr2 > 6000)
    ldrData = 1
else
    ldrData = 0
end

%% Write Data %%

thingSpeakWrite(writeChannelID, [ldrData, analyzedData],
'WriteKey', writeAPIKey, 'Fields', [1,2]);
```


Demo video:

<https://youtu.be/Y2yaceIULRw>

Assignment 3 - Specification

- Objectives:
 - IoT with Complex Sensors and Actuators
 - Computing in Thingspeak
 - Collaborate 2 Raspberry Pi's with Thingspeak
- Upload to E3 before 1/5 at 23:55PM
 - Assignment 3 – deliverables
 - Report (2-4 pages) using our template
 - Explain the objectives
 - Explain the specification of sensors and actuators used
 - Explain the system design
 - Explain your source code and the detail of how your script can post and get the data from Thingspeak to your sensors and actuators, respectively
 - Explain your matlab code in Thingspeak platform
 - Link to a 3-minute demo video on YouTube
 - Source Codes
 - Zip the above files into one compressed file and upload
 - Q&A? Post on E3 discussion board

Assignment 3 - Specification

- Note for Assignment 3:
 - You must use different combinations of sensors and actuators than the ones we present
 - If using the same combinations, your maximum score is only 65.
 - The report can be written in Chinese or English, but the video must be delivered in English.
 - In video, explain how you assemble your additional sensors and actuators, how to store and retrieve data from cloud, and show the results.
 - Upload your video to YouTube and put the link into your report. Don't upload your video to E3.
 - In your report, make sure you have a diagram of connected sensors, actuators, and cloud.

Thank You