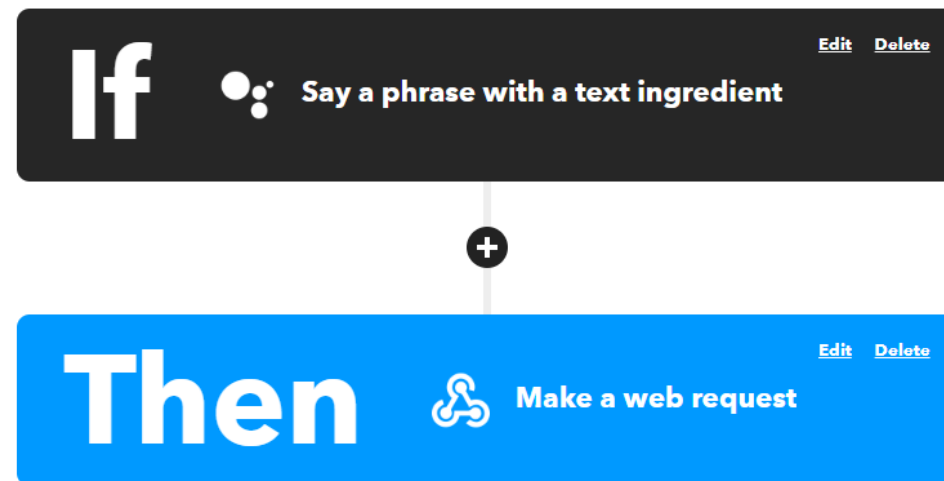


Lab 3

Motion Detector

IFTTT Design

- The motion sensor applet receives a google voice command from google assistant and transmit the data to thingspeak



IFTTT Design

- The google assistant setup is shown below
- The user can say 'motion sensor' + 'activate/deactivate'
- The google assistant will respond with 'ok, motion sensor activate/deactivate'

Say a phrase with a text ingredient

What do you want to say?

motion sensor \$

Enter a \$ where you'll say the text ingredient

What's another way to say it? (optional)

motion system \$

Enter a \$ where you'll say the text ingredient

And another way? (optional)

Enter a \$ where you'll say the text ingredient

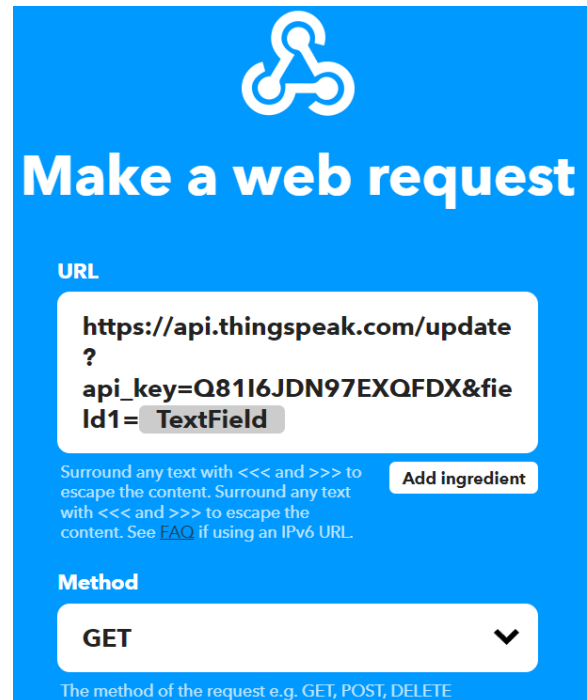
What do you want the Assistant to say in response?

ok, motion sensor \$

You can enter a \$ where you want to hear the text ingredient in the response

IFTTT Design

- The webhooks can receive the data from the google assistant and send the data to thingspeak through a URL



The image shows the 'Make a web request' configuration screen in IFTTT. It has a blue background with the IFTTT logo at the top. The title 'Make a web request' is in white. Below it, the 'URL' section contains a text box with the URL 'https://api.thingspeak.com/update?api_key=Q81I6JDN97EXQFDX&field1=' followed by a 'TextField' input. A small text block explains escaping characters with <<< and >>>. To the right of this text is an 'Add ingredient' button. The 'Method' section has a dropdown menu currently set to 'GET'. At the bottom, a small text line says 'The method of the request e.g. GET, POST, DELETE'.

Make a web request

URL

`https://api.thingspeak.com/update?`
`api_key=Q81I6JDN97EXQFDX&fie`
`ld1=`

Surround any text with <<< and >>> to escape the content. Surround any text with <<< and >>> to escape the content. See [FAQ](#) if using an IPv6 URL.

Add ingredient

Method

GET ▼

The method of the request e.g. GET, POST, DELETE

ThingSpeak Design

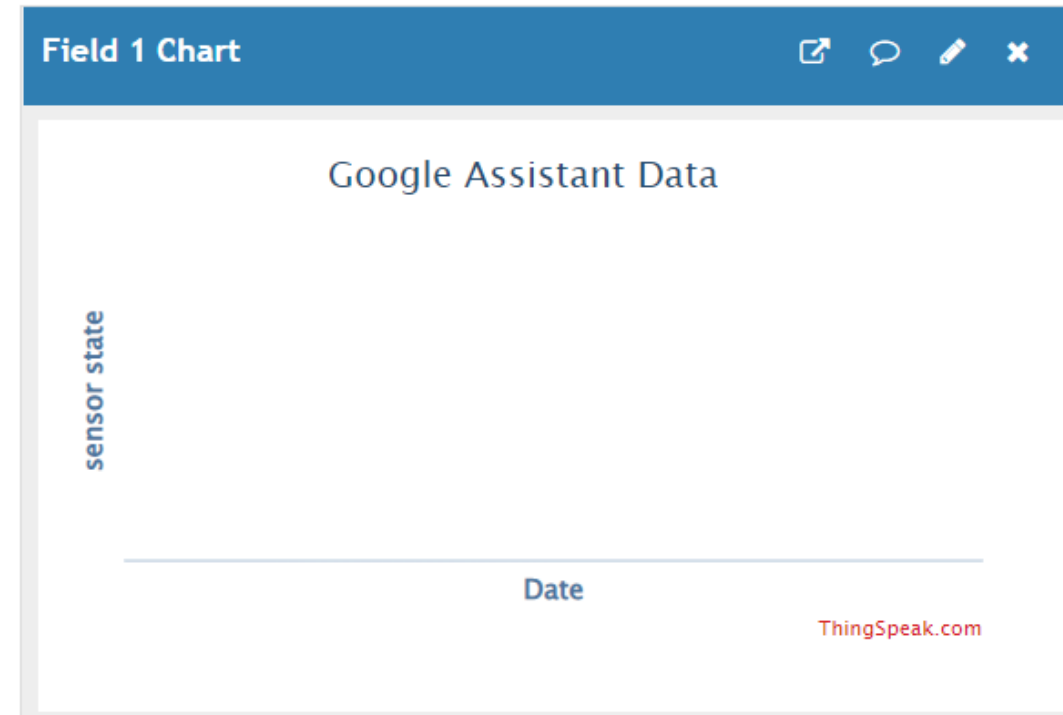
- I created a channel in ThingSpeak
- There is a Field 1 to receive data from IFTTT

Channel Stats

Created: [2 days ago](#)

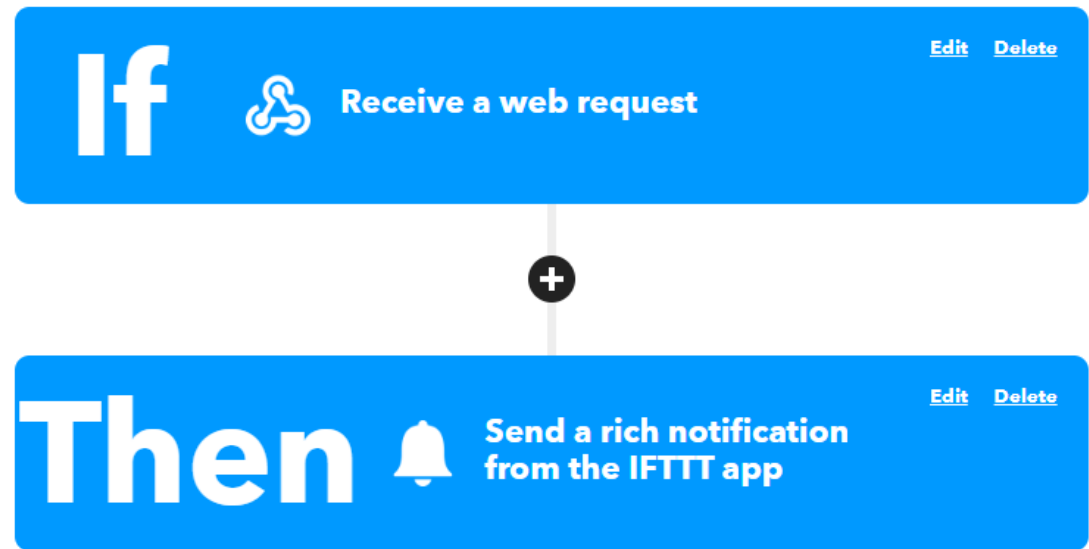
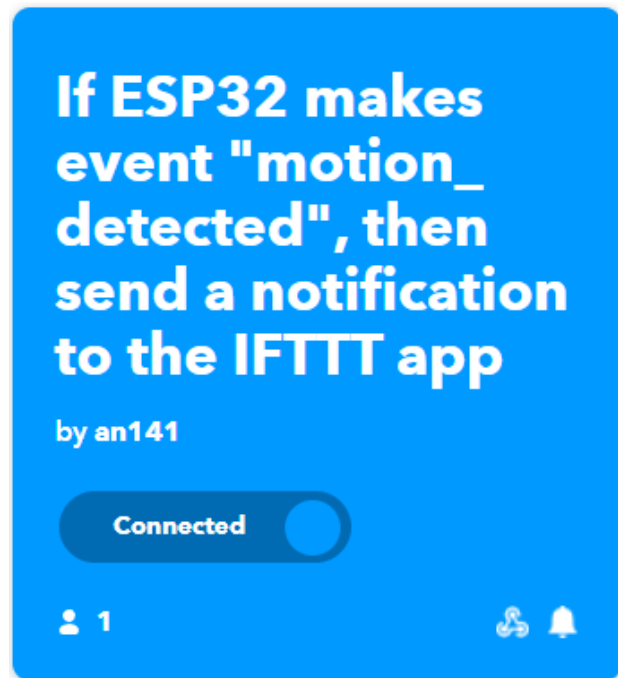
Last entry: [about 5 hours ago](#)

Entries: 10



IFTTT Design


- The notification applet uses webhooks to received data
- The notification applet uses google notification to notify my phone



IFTTT Design

- The webhooks set up is shown below
- The event name is 'motion_detected'
- The webhooks has a special url. ESP32 can use this url to send data to the IFTTT

Edit trigger fields



Receive a web request

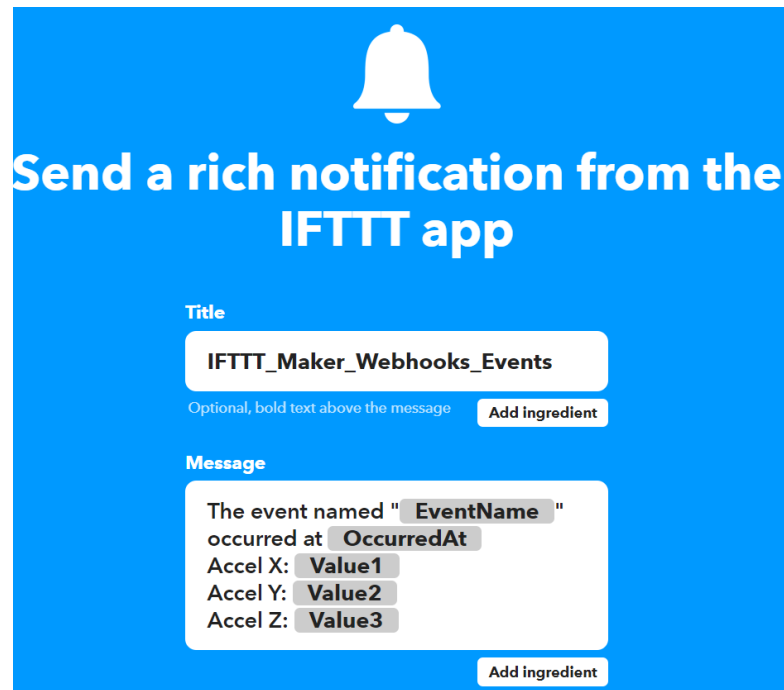
Event Name

motion_detected

The name of the event, like "button_pressed" or "front_door_opened". Use only letters, numbers, and underscores

IFTTT Design

- The notification set up is shown below
- The phone notification will show the current sensor reading, the event name, and the time when the notification is triggered



The image shows a notification setup interface for IFTTT. It features a blue background with a white bell icon at the top. The main heading is "Send a rich notification from the IFTTT app". Below this, there are two sections: "Title" and "Message". The "Title" section has a text input field containing "IFTTT_Maker_Webhooks_Events" and a small "Add ingredient" button. The "Message" section has a text input field containing a template message: "The event named 'EventName' occurred at OccurredAt Accel X: Value1 Accel Y: Value2 Accel Z: Value3". There is also an "Add ingredient" button for the message section.

Send a rich notification from the IFTTT app

Title

IFTTT_Maker_Webhooks_Events

Optional, bold text above the message [Add ingredient](#)

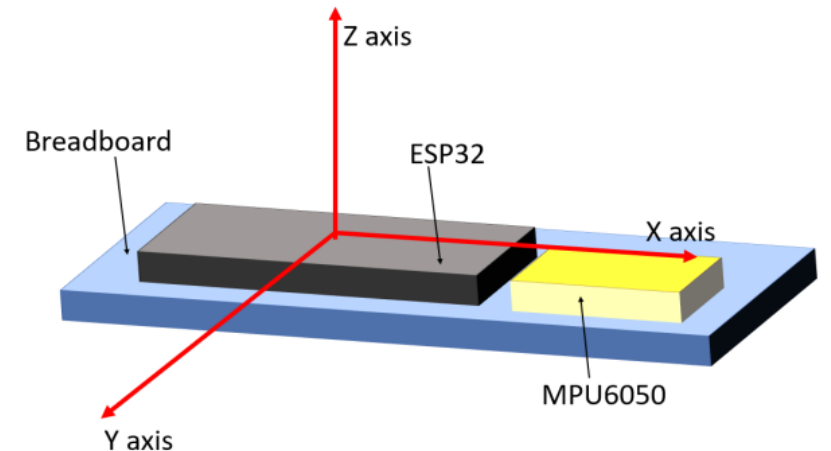
Message

The event named "EventName" occurred at OccurredAt
Accel X: Value1
Accel Y: Value2
Accel Z: Value3

[Add ingredient](#)

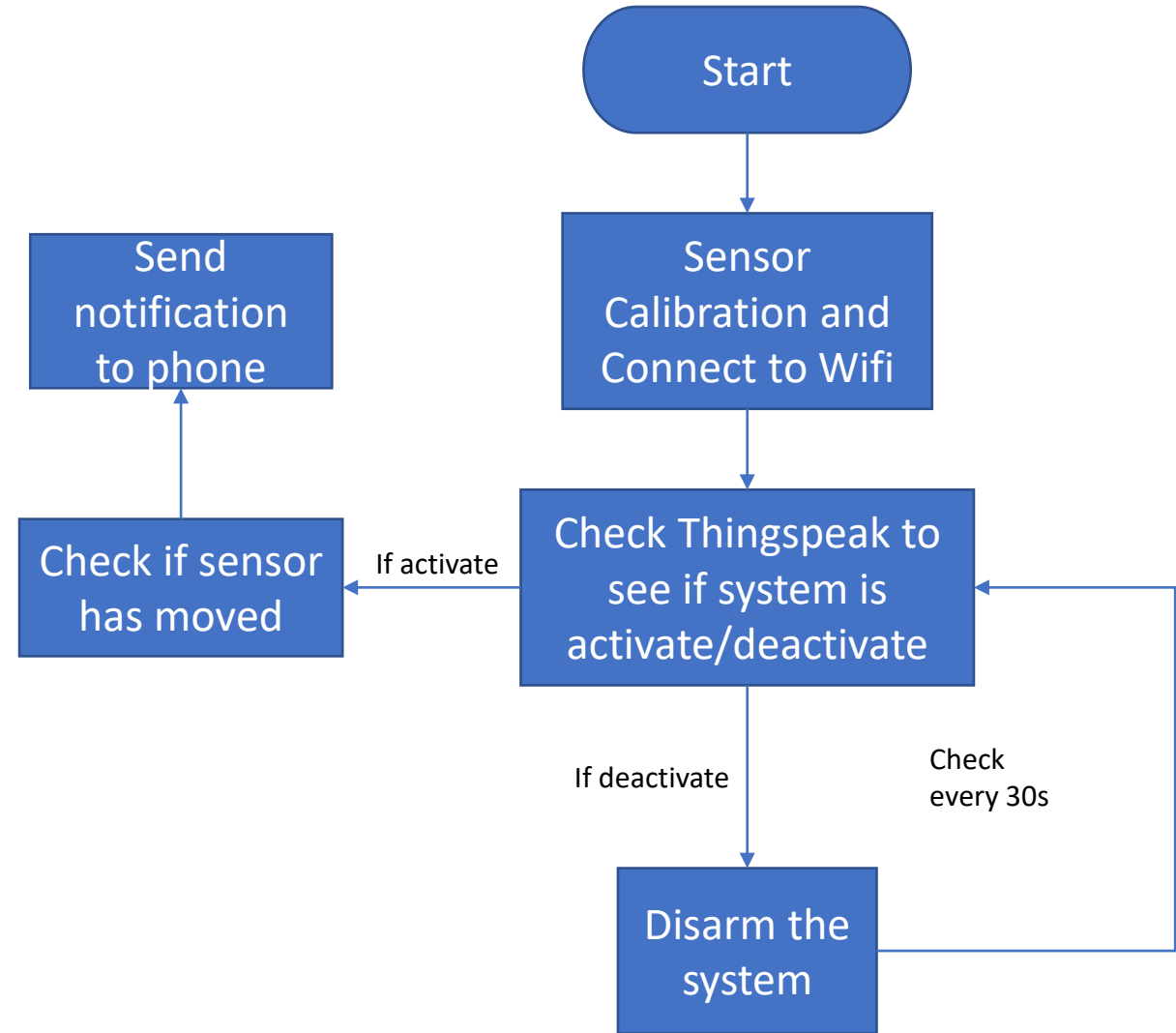
Sensor Calibration

- I read sensor data when the sensor facing upwards
 - In this case, AcZ_faceup should be -9.8 and the other readings should be zero
- Then, I read sensor data when the sensor facing downwards
 - In this case, $AcZ_facedown$ should be 9.8 and the other reading should be zero
- Therefore, the sensor gain $K = 9.8 * 2 / (AcZ_facedown - AcZ_faceup)$
- The sensor reading at zero acceleration is also known
 - $AcX_zero = (AcX_faceup + AcX_facedown) / 2$
 - $AcY_zero = (AcY_faceup + AcY_facedown) / 2$
 - $AcZ_zero = (AcZ_faceup + AcZ_facedown) / 2$
- The real acceleration reading is
 - $AcX_real = (AcX - AcX_zero) * K$
 - $AcY_real = (AcY - AcY_zero) * K$
 - $AcZ_real = (AcZ - AcZ_zero) * K$



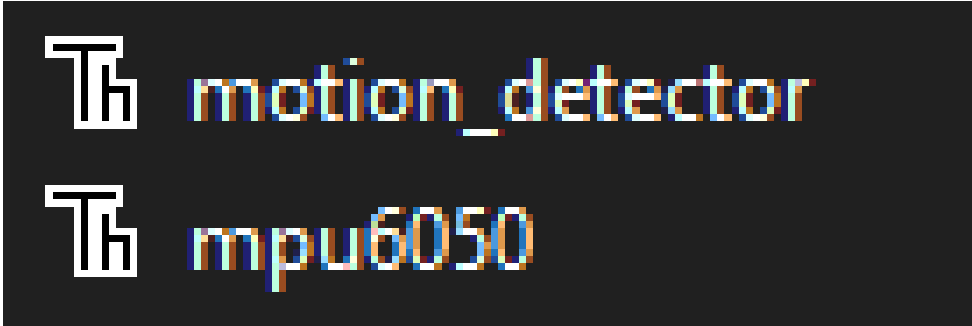
Software Flowchart

- At the start point, ESP32 will calibrate the sensor
- Once the sensor is calibrated and Wifi is connected, ESP32 will check with Thingspeak to see if the system should be activated or not.
- If the system should be activated, ESP32 will use MPU6050 to check the acceleration
- If the acceleration has changed, ESP32 will send a phone notification



File Structure

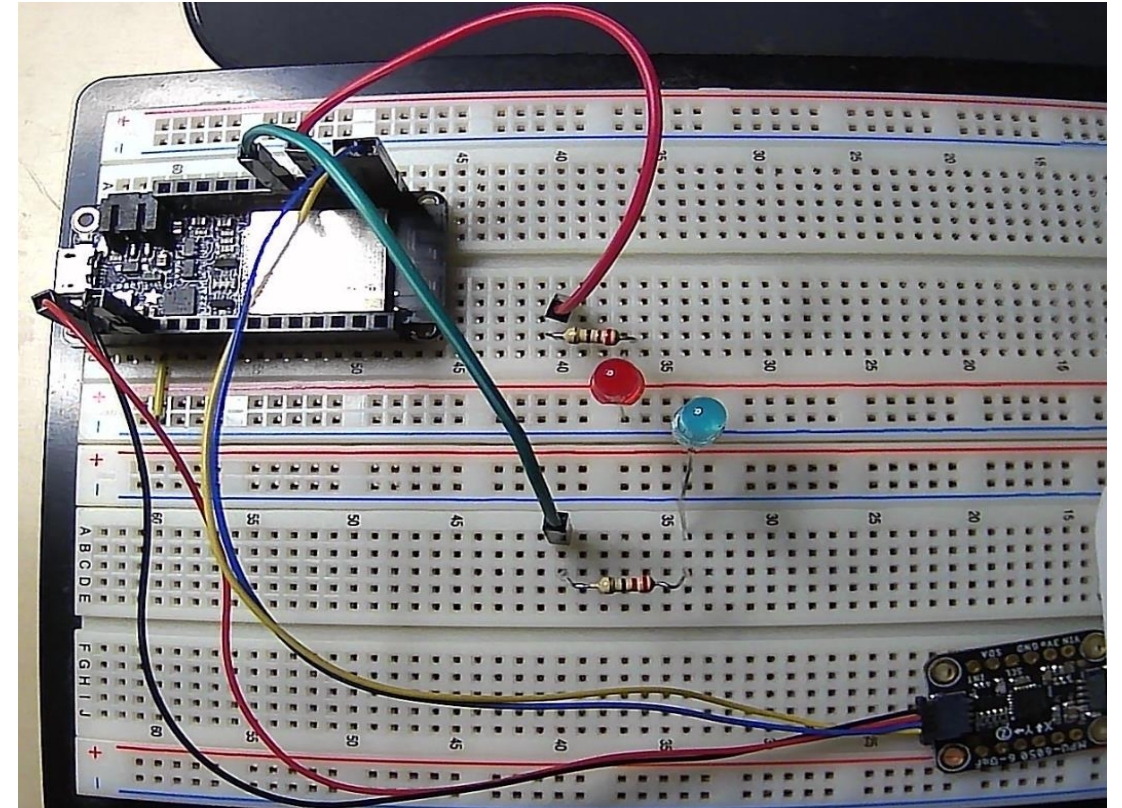
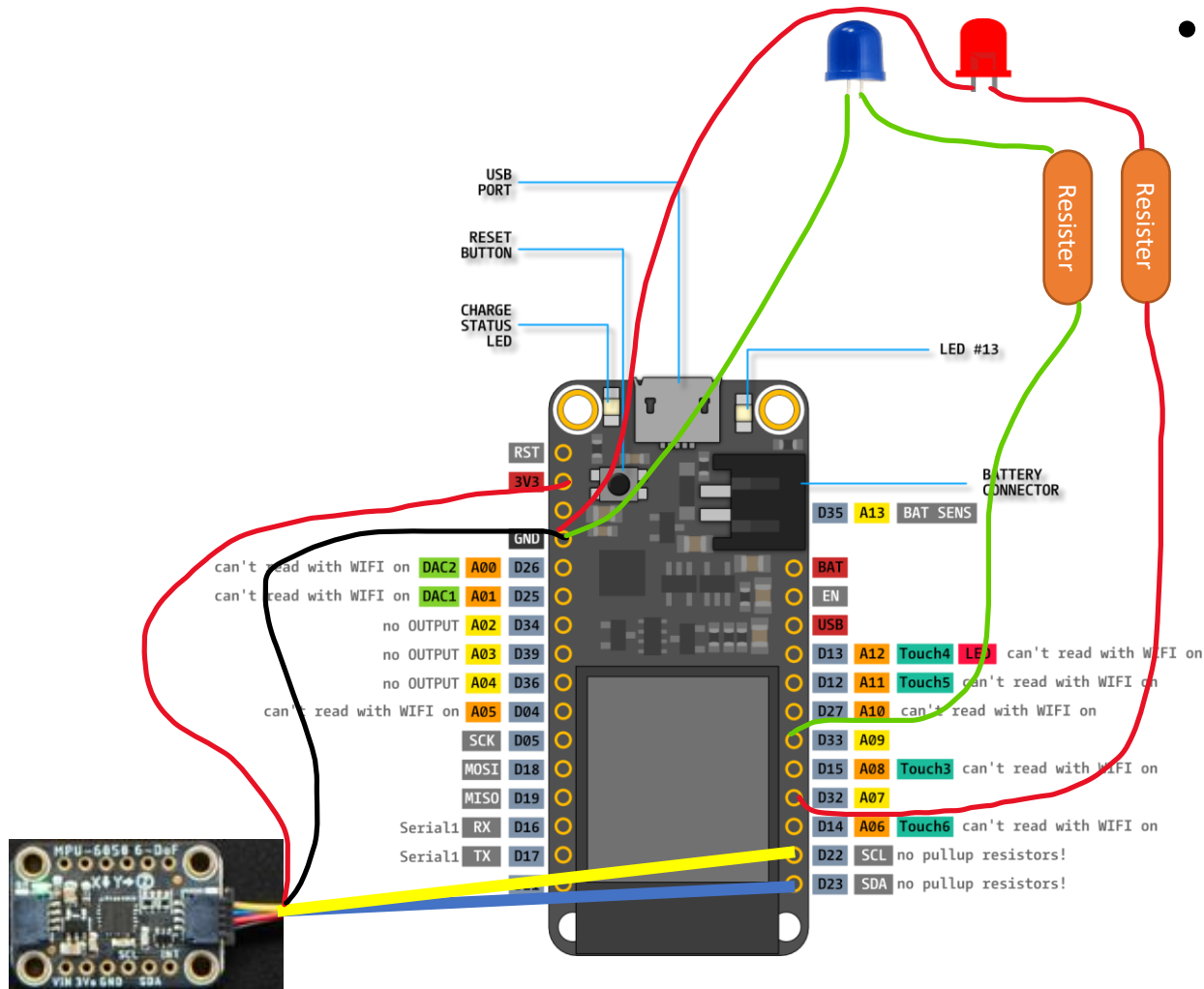
- motion_detector.py is the main code to run on the ESP32
- mpu6050.py is the MPU 6050 driver library that ESP32 used to communicate with MPU6050 through I²C

A terminal window with a dark background and light-colored text. It displays two lines of text, each preceded by a white icon of a folder with a document inside. The first line shows 'motion_detector' and the second line shows 'mpu6050'.

```
Th motion_detector
Th mpu6050
```

Hardware Connection

- The red led is connected to pin 32
- Blue led is connected to pin 33
- I²C connection is shown blow
 - Yellow goes to SCL
 - Blue goes to SDA
 - Red and black is 3V and ground



Video

- <https://youtu.be/zLkY6vPAUUY>
- In the video, I showed that
 - I calibrated the sensor with ESP32
 - I can use google assistant from my phone to activate and deactivate the system
 - I can receive a notification with the sensor reading sent from ESP32 when the sensor detects motion
- Below are some external reference I used for my design
 - <https://esp32io.com/tutorials/esp32-ifttt>
 - https://www.youtube.com/watch?v=X-_25tzo8Cw
 - https://makeblock-micropython-api.readthedocs.io/en/latest/public_library/Third-party-libraries/urequests.html