Smart Room

INTERNET OF THINGS

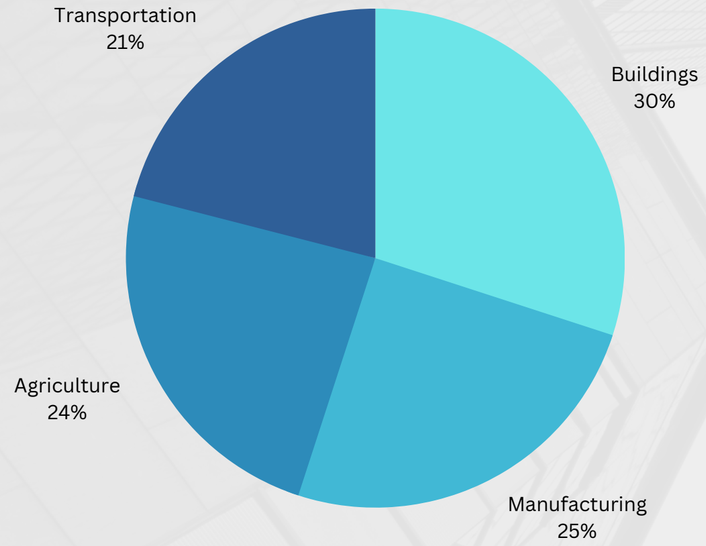
FINAL PRESENTATION

Team Members:

1. Paranut Prasittipap 6338139521
2. Pattadon Naksuwan 6338153221
3. Nathamon Kongsawat 6338116021
4. Aphiphu Sirimangkalalo 6338250421
5. Thanasit Pakkaananchai 6338101621
6. Theekadhas Chantarasanarm 6338089821

PROBLEM STATEMENT

30% of Carbon emission in Thailand comes from buildings, more specifically the electricity used in the buildings.



นโยบาย ›

คาร์บอนคุมได้ กทม.ปลอดคาร์บอน (BMA Net Zero)

สิ่งแวดล้อมดี

Focus on unnecessary electricity usage in hotels and lecture rooms

↓

Reduce carbon dioxide produced

The problems of unnecessary electricity of hotel and lecture rooms occur primarily from users forgetting to turn the lights and air-conditioning off.

INSIGHTS

1. Tubtim Siam River Kwai Resort - Karnchanaburi
2. Tubtim Siam Suvarnabhumi Resort - Bangkok
3. DRMIS Lab Chulalongkorn University - Bangkok

- Guests tend to leave the lights even when they are leave the room
- Customers bypass the keycard system by inserting a card to enable the electricity
- Students and professors tend to forget to turn the lights and the air conditioner off before leaving the room
- Currently data-logging is done by hand and is susceptible to error

What our customers will gain

- Lower electrical bill
- Clear record of customers entering and leaving time
- An improvement to the management system
- Dashboard that can track data based on sensors immediately

IDEA AND VALUE PROPOSITION

Our Idea

Automatic system to help reduce unnecessary electricity usage.

Using STM32 and Raspberry Pi to send information/ receive information from Sensors & Actuators to control the devices in the room.

Value Proposition

Customer Problems

- **Unnecessary energy consumption**
- **Unknown customer activity**
- **Untrackable operation and data**

Solution

- Automatic system to help reduce unnecessary electricity usage
- People Occupancy Detection
- Smart Dashboard

COMPARISON BETWEEN OTHERS

Using 5 criteria to evaluate the invention compared to other existing products and each metric.

Criteria	Keycard Access	Camera	Our Project
Easy to access data		✓	✓
Notification			✓
Privacy	✓		✓
Flexibility		✓	✓
Scalability			✓

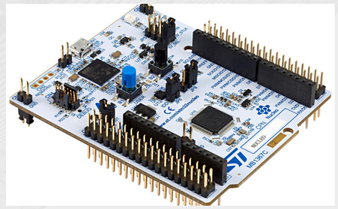
WHY IOT?

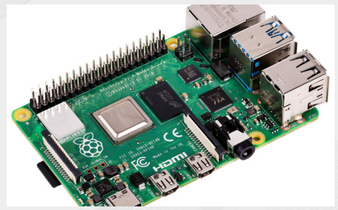
- **Real time indicator of the people activity inside any closed room**
- **Optimize the way of dealing with unnecessary electricity usage**
- **Smart devices that automate daily tasks allow humans to do other activities**

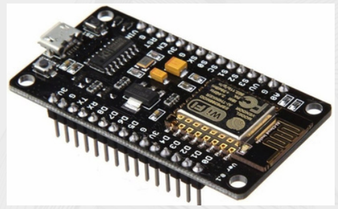
HARDWARE LIST

- BOARDS

-STMicroelectronics Board (411RE)
-Raspberry Pi 4 Model B
-ESP-8266





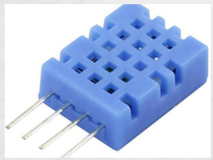


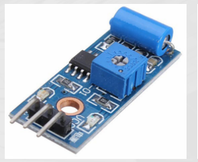
- OUTPUT

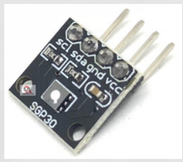
- Yeelight Bulb
- Sonoff switch

- SENSORS

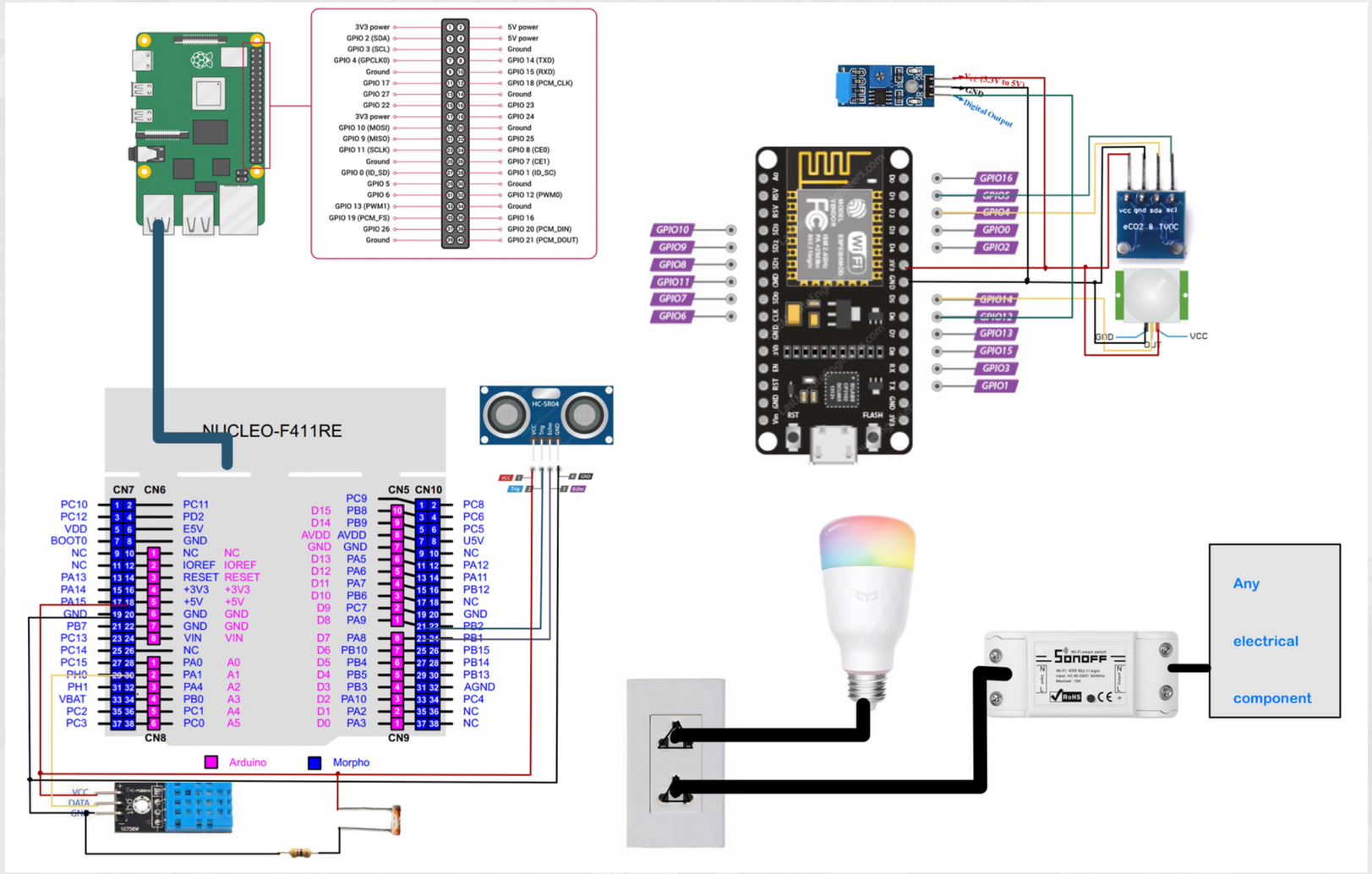
- Temperature Sensor (DHT 11)
- Ultrasonic Sensor (HC-SR04)
- Light Sensor (LDR)
- Vibration Sensor (SW420)
- CO2 Sensor (SGP30)
- IR & PIR Sensor (HC-SR501)







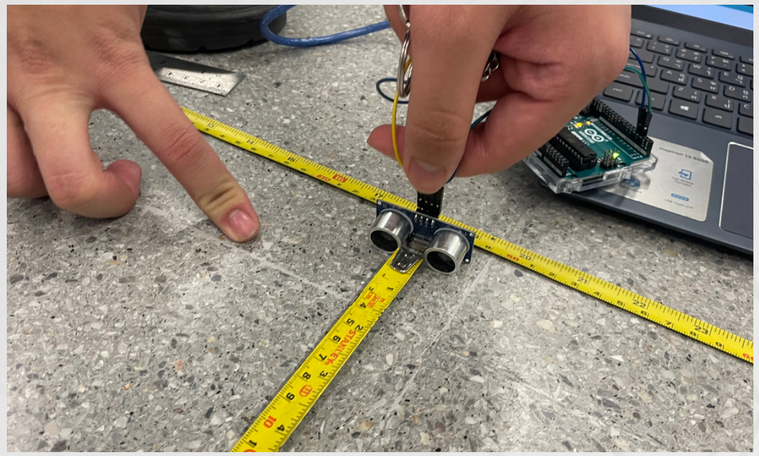
WIRING DIAGRAM



SENSORS CALIBRATION

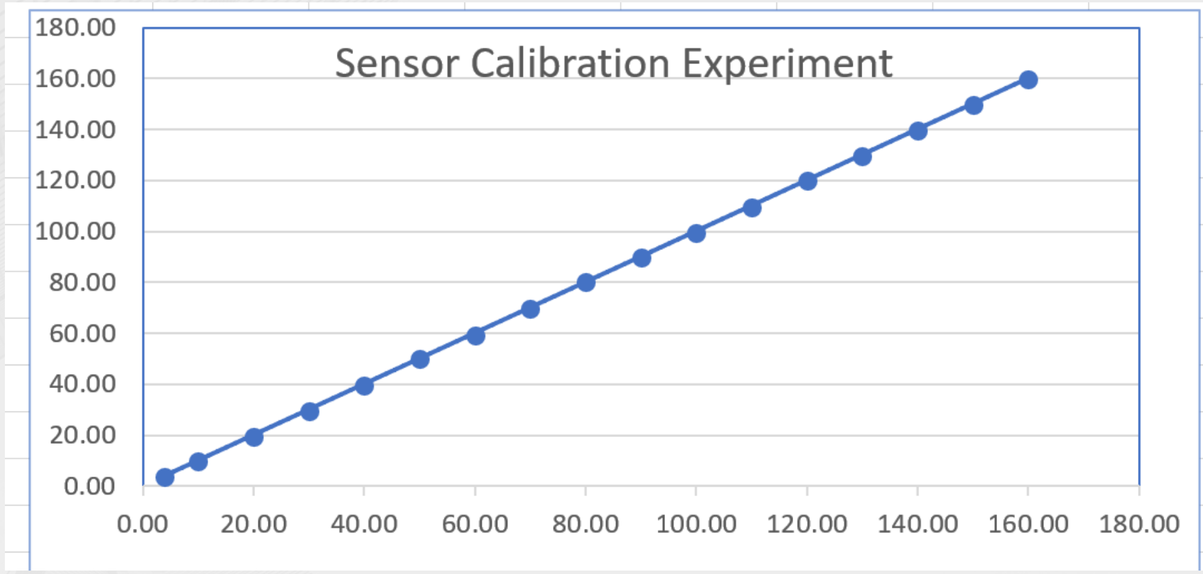
HC-SR04 Ultrasonic sensor

Have the value recorded by the sensor compared with the data from real environment. Then, adjust the sensor data with an equation in the STM32CubeIDE to match the real value.



	From Sensor(cm
4.00	4.00
10.00	10.00
20.00	20.00
30.00	30.00
40.00	40.00
50.00	50.00
60.00	59.90
70.00	70.10
80.00	80.00
90.00	90.00
100.00	100.00
110.00	110.10
120.00	119.90
130.00	129.90
140.00	140.00
150.00	150.00
160.00	160.00

Real Distance (cm.)



Distance read from Sensor (cm.)

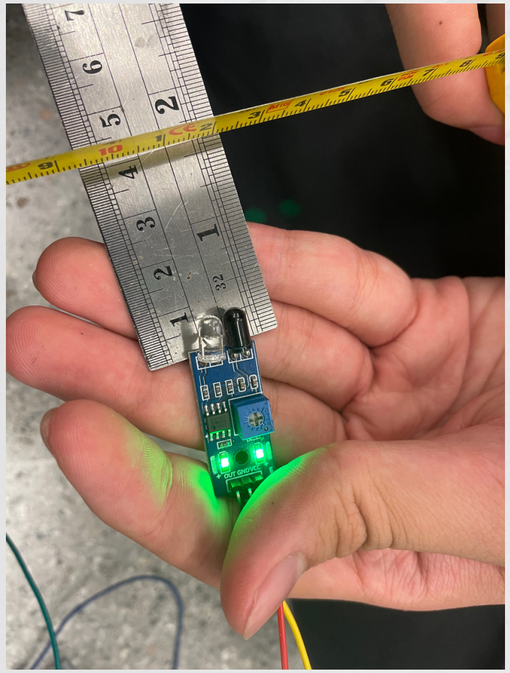
Distance = 0.9999d + 0.0041

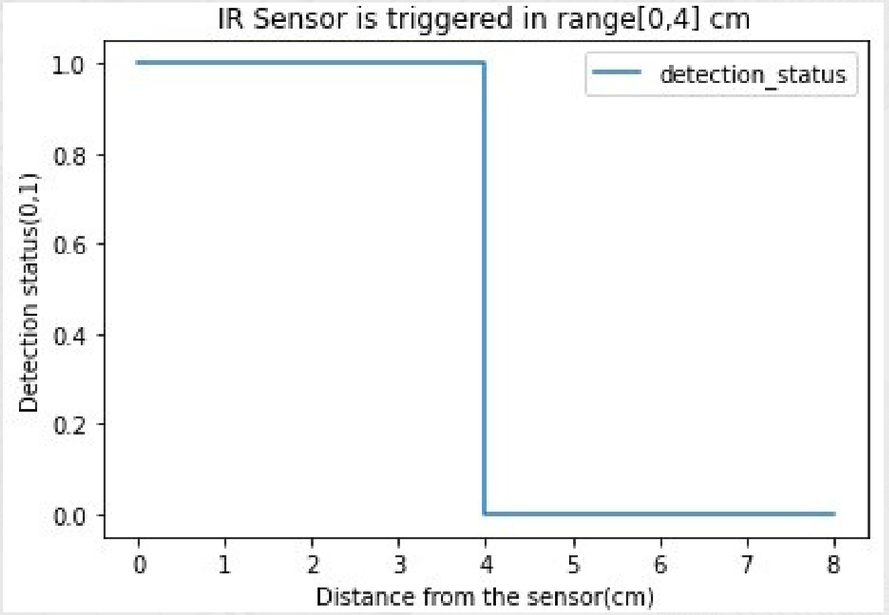
FINDINGS

HC-SR04 Ultrasonic sensor

The sensor readings are consistent and accurate.
When rounded up, real distance equals distance read from sensor.

SENSORS CALIBRATION





HC-SR501 Infrared sensor

Have the value recorded from the sensor compared with the real data from environment. Then, change the calculation equation in the STM32CubeIDE according to the real value.

FINDINGS

The sensor can detect at the maximum distance of 4 cm.

DHT11 LDR Light sensor, and SW420 Vibration sensor

They were initially calibrated from the factory.

CO2 SGP30 sensor

They were initially calibrated from the factory. However, it has to be running for 1 hour.

SENSORS CALIBRATION

Sensitivity Adjustment

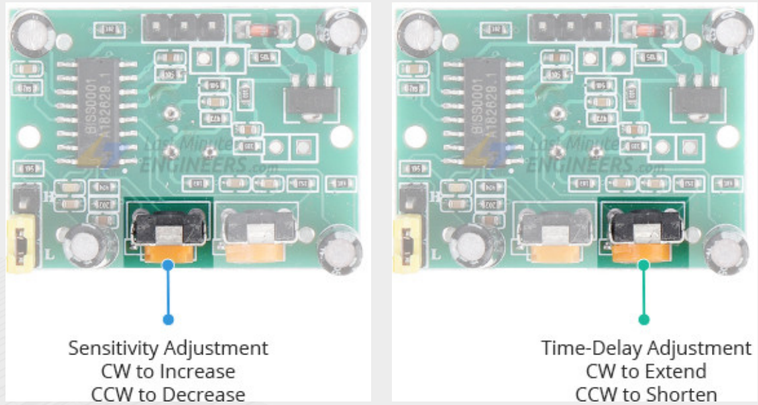
Rotating the knob clockwise increase the distance, turning it counter-clockwise will decrease the distance. The range of sensitivity is 3-7 meter.

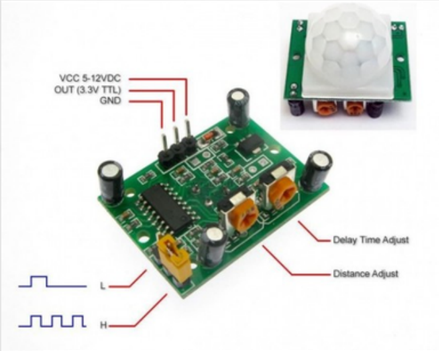
Time-Delay Adjustment

Rotating the knob clockwise increases the delay, turning it counter-clockwise decreases the delay. The range of the delay is 1 second to about 3 minutes.

HC-SR01 Pir sensor

Adjust by rotating the knob clockwise and counter-clockwise

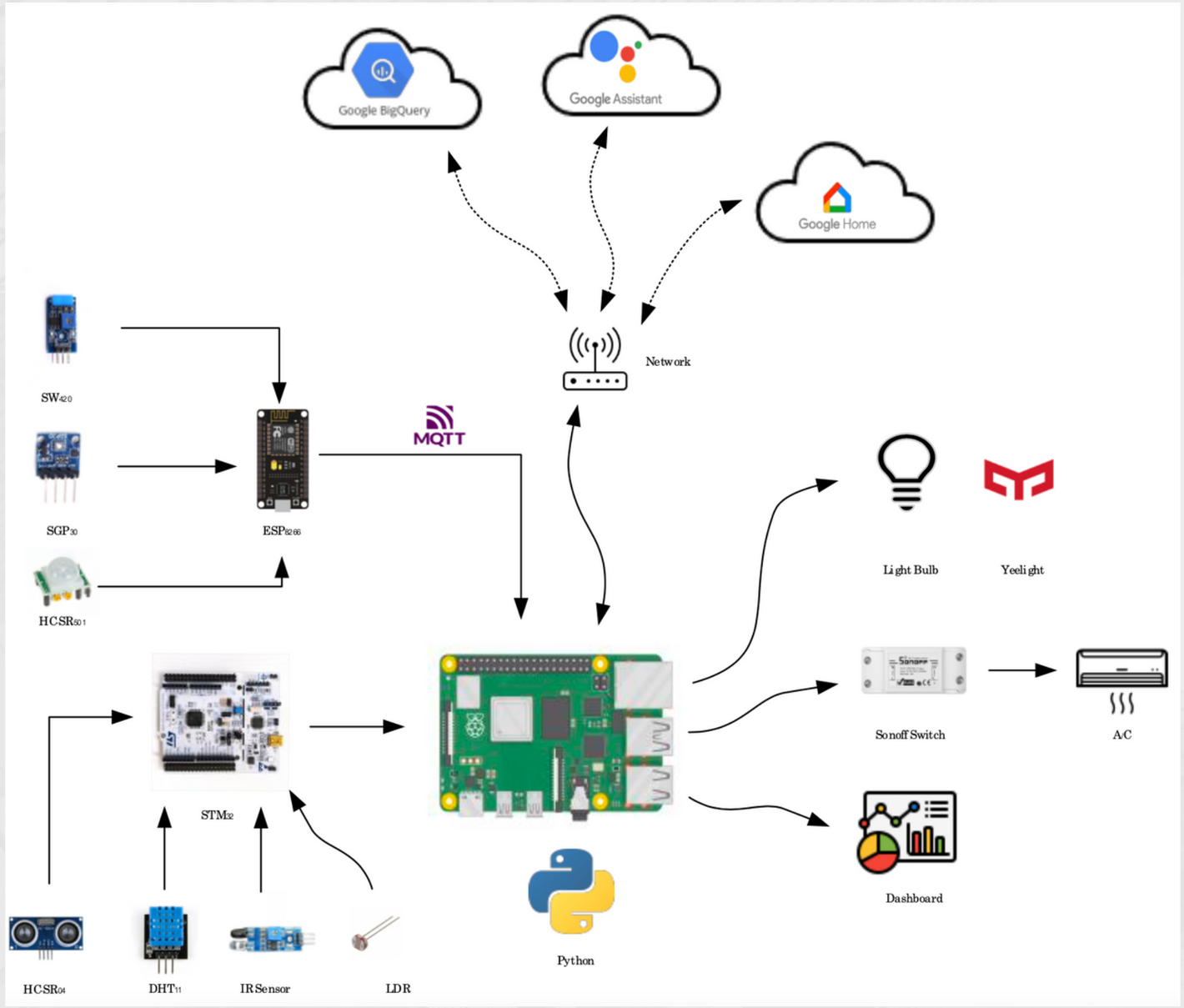




HC-SR01 Pir sensor

Run the sensor and then use the screwdriver to tighten or loosen the 2 screw which are the time delay and distance. By adjusting the time-delay until the sensor can detect human presence more precisely.

DESIGN OF EXPERIMENT - SYSTEM DIAGRAM



DESIGN OF EXPERIMENT TESTING METHODOLOGY

Requirement Analysis

- Unnecessary energy consumption
- Unknown customer activity
- Untraceable operation and data

Design

- Users can use dashboard easily and conviniently
- The communication between devices communicates through network.

Develop

Should be able to do the following tasks

- visualize data from sensors with the dashboard
- Automatically turn off the light with Google Home if the model can predict the absence of human within 5 minutes

Test

Functional Tests

- Can sensors read data correctly?
- Can sensors detect human absence?

Problem Aspects

- Can dashboard keep track of the real-time data?
- Can dashboard query from the database?

Deploy

Regularly run the system in the experiment room and monitor the data received through dashboard and network for further improvement.

DESIGN OF EXPERIMENT

EXPERIMENTING

By shutting off unnecessary lights, the design aims to assist the policy of reducing the consumption of CO2 through lowering the unnecessary power used. By concentrating on the guests who stayed at the resort and students at the university since they forget to turn off the lights before leaving.

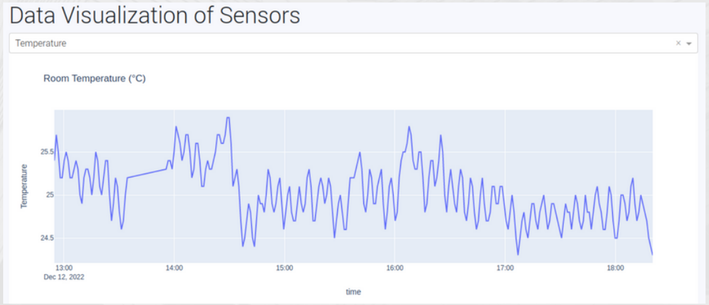
The DRMIS room is used as the testing site for our project on 12 Dec 2022. By leaving the sensors in the room for a couple of hours, letting the raspberry pi continuously running to save the data. After that, convert the data into csv file format in which the change of data will indicate activities of people within the room.

EXPECTED RESULT

The data from sensors will be sent to the dashboard and visualized into a graph. By having the dashboard, the users can determine whether or not someone is in the room.

The Lights and A/C will be automatically turned off via Google Home when there is no person detected with the machine learning model for 5 mins

User can command the light and A/C through the Google Home application anywhere



EXPERIMENT KPI - FUNCTIONALITY TEST

1

Light, A/C and other electrical components can be turned off without noticeable delay.

The light is immediately switched off by giving a command on the google assistant with less than 10 seconds.

2

The specificity (true negative rate) of the test set data is higher than 80%

Our model's true negative rate on our test set data is 85%

3

The dashboard can visualize the real time data, with non-noticeable delay. Moreover, historical data will update daily and the data can be kept up to 7 days

The dashboard can visualize the real time data, updated every minute with a delay less than 10 seconds. Moreover, historical data will update at 23.59 pm everyday and will be kept and visualize for more than a week.

EXPERIMENT KPI - RESOLVING PROBLEM

4

User can have access to all devices remotely and be able to control them from anywhere.

Users simply install the "Google Home" app and link their google account to control the system. Once sync, user can control all devices both automatically and manually.

5

The user can view the historical statistics of their rooms by viewing the past sensor data to determine anything that needs inspection

The historical data is kept for 7 days and is shown to the user as a graph. The user can then determine and track all unnatural rising or falling trends to inspect

6

Decrease the usage of unnecessary power consumption by 20%

For one hour of unnecessary power consumption, our system lowers consumption by 50%, which only increases as the number of hours that the room wastes energy

DATA COLLECTION CODE FROM STM32

```
DHT11_Start())
{
  RHI = DHT11_Read(); // Relative humidity integral
  RHD = DHT11_Read(); // Relative humidity decimal
  TCI = DHT11_Read(); // Celsius integral
  TCD = DHT11_Read(); // Celsius decimal
  SUM = DHT11_Read(); // Check sum
  if (RHI + RHD + TCI + TCD == SUM)
  {
    // Can use RHI and TCI for any purposes if whole number only
    tCelsius = (float)TCI + (float)(TCD/10.0);
    tFahrenheit = tCelsius * 9/5 + 32;
    RH = (float)RHI + (float)(RHD/10.0);
    // Can use tCelsius, tFahrenheit and RH for any purposes
  }
```

DHT 11

On lines 177-179, the data from DHT11 is converted to float type, and is sent to plot the graph and is used to measure the temperature and humidity in the room.

```
114  while (1)
115  {
116      HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_SET);  // pull the TRIG pin HIGH
117          __HAL_TIM_SET_COUNTER(&htim1, 0);
118          while (__HAL_TIM_GET_COUNTER (&htim1) < 10);  // wait for 10 us
119          HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET);  // pull the TRIG pin low
120
121          pMillis = HAL_GetTick(); // used this to avoid infinite while loop  (for timeout)
122          // wait for the echo pin to go high
123          while (!(HAL_GPIO_ReadPin (ECHO_PORT, ECHO_PIN)) && pMillis + 10 >  HAL_GetTick());
124          Value1 = __HAL_TIM_GET_COUNTER (&htim1);
125
126          pMillis = HAL_GetTick(); // used this to avoid infinite while loop (for timeout)
127          // wait for the echo pin to go low
128          while ((HAL_GPIO_ReadPin (ECHO_PORT, ECHO_PIN)) && pMillis + 50 > HAL_GetTick());
129          Value2 = __HAL_TIM_GET_COUNTER (&htim1);
130
131          Distance = (Value2-Value1)* 0.034/2;
132          HAL_Delay(50);
```

Ultrasonic

The code at line 116 turns the pin high, making the ultrasonic send the pulse until it hit the object. When the pulse hit the object it will send back the echo to the sensor then the data can be used to calculate the distance. But if the pulse did not hit the object for 10 microseconds, the pin will switch back to low, making the sensor detect the farthest distance that pulse can go

```
92   /* USER CODE END 2 */
93
94   /* Infinite loop */
95   /* USER CODE BEGIN WHILE */
96   while (1)
97   {
98     /* USER CODE END WHILE */
99
100    /* USER CODE BEGIN 3 */
101      if (HAL_GPIO_ReadPin(SENSOR_GPIO_Port, SENSOR_Pin)==GPIO_PIN_RESET) // Sensor Active
102      {
103        HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_SET); // Led On
104    }
105      else
106      {
107        HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, GPIO_PIN_RESET); // Led Off
108    }
109    /* USER CODE END 3 */
110  }
111
112 /**
113   * @brief System Clock Configuration
```

IR Sensor

This block of code receives the signal from infrared sensor. If it detects something, the LED on the sensor will be on. If not, the LED on the sensor will be off.

```
/* Light Sensor */
HAL_ADC_Start(&hadc1);
HAL_ADC_PollForConversion(&hadc1,HAL_MAX_DELAY);
uint16_t light = HAL_ADC_GetValue(&hadc1);
```

Light Sensor

This block of code receives the data from the light sensor, ranging from 0 to 65535 as the value is saved as an 16 bit ADC value

sending uart

```
sprintf(buf, "%.2f, %.2f, %d, %s, %d\r\n", tCelsius, RH, Distance, "Obstacle detected", light);
HAL_UART_Transmit(&huart2, buf, strlen(buf), 100);
HAL_Delay(1000);
```

DATA COLLECTION CODE FROM ESP8266

```
#define Vibrator D6
int SENSOR_OUTPUT_PIN = 14;
Adafruit_SGP30 sgp;
int motion_detected = LOW;
motion_detected = digitalRead(Vibrator);
int sensorvalue = digitalRead(SENSOR_OUTPUT_PIN);
if (mqtt.connect(MQTT_NAME, MQTT_USERNAME, MQTT_PASSWORD)) {
 sprintf(msg,"%d",motion_detected);
 sprintf(msg1,"%d",(int) sgp.TVOC);
 sprintf(msg2,"%d",(int) sgp.eCO2);
 sprintf(msg3,"%d",sensorvalue);
 s = String(msg);
 s = s+"/"+msg1+"/"+msg2+"/"+msg3;
 mqtt.publish("esp8266/sensors",s.c_str());
```

Vibration, SGP30 and pir sensor

This block of code read data from 3 sensors, which are vibration, sgp30 and pir. Moreover, it publishes messages through mqtt protocol tp topic "esp8266/sensors"

More Details in our group's github: https://github.com/gear-patt/Smart-Room/blob/main/MQTT_sensors/MQTT_sensors.ino

DATA COLLECTION CODE ON PI

```python
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to MQTT Broker!")
    else:
        print("Failed to connect, return code %d\n", rc)

client = mqtt_client.Client(client_id)
client.username_pw_set(username, password)
client.on_connect = on_connect
client.connect(broker, port)

def on_message(client, userdata, msg):
    print(f"Received `{msg.payload.decode()}` from `{msg.topic}` topic")
    temp = pd.DataFrame()
    temp['data'] = [msg.payload.decode()]
    temp.to_csv('esp_temp.csv',index=False)

client.subscribe(topic)
client.on_message = on_message
client.loop_start()
```

```python
with serial.Serial('/dev/ttyACM0', 9600, timeout=1) as ser:
    counter=0
    while(True):
        line = ser.readline()
        try:
            sensor_dt = pd.read_csv('esp_temp.csv')
            sensor_dt = sensor_dt['data'][0]
            vibration = sensor_dt.split('/')[0]
            tvoc = sensor_dt.split('/')[1]
            co2 = sensor_dt.split('/')[2]
            pir = sensor_dt.split('/')[3]
            if line.decode('UTF-8').strip() != "":
                data = pd.read_csv('sensors_data.csv')
                real = pd.read_csv('real_time.csv')
                now = datetime.now()
                my_dates = pd.date_range("now", periods=1)
                current_time = now.strftime("%Y-%m-%d %H:%M:%S")
                current_date = now.strftime("%Y-%m-%d")
                counter = counter+1
                print(counter)
```

This is the code for receiving the data from stm32IDE and ESP8266 every 60 seconds through serial communication and MQTT protocol.

More Details in our group's github: https://github.com/gear-patt/Smart-Room/blob/main/sensor_data_serial.py

DATABASE (GOOGLE BIG QUERY)

sensor_db QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Row	time	date	Temperature	RH	Distance	Detection	Light	Vibration	TVOC	CO2	PIR
1	2022-12-12 13:57:45	2022-12-12	25.4	52.0	296.0	0	1107.0	0	30	400	0
2	2022-12-12 14:13:53	2022-12-12	25.4	52.0	229.0	0	581.0	0	80	400	0
3	2022-12-12 14:03:08	2022-12-12	25.6	52.0	230.0	0	255.0	0	67	400	0
4	2022-12-12 13:31:14	2022-12-12	24.6	52.0	228.0	1	1151.0	0	118	507	0
5	2022-12-12 13:24:47	2022-12-12	25.0	52.0	230.0	1	1139.0	0	73	436	0
6	2022-12-12 13:14:04	2022-12-12	25.2	52.0	230.0	1	270.0	0	123	539	0
7	2022-12-12 12:57:59	2022-12-12	25.2	52.0	229.0	1	234.0	0	95	468	0
8	2022-12-12 13:03:20	2022-12-12	25.2	52.0	228.0	1	1010.0	0	127	529	1
9	2022-12-12 14:21:21	2022-12-12	25.4	54.0	5.0	0	378.0	0	148	496	0
10	2022-12-12 14:22:24	2022-12-12	25.5	54.0	5.0	0	221.0	0	210	588	0
11	2022-12-12 14:10:40	2022-12-12	25.3	54.0	230.0	0	645.0	0	133	512	0
12	2022-12-12 14:12:49	2022-12-12	25.6	54.0	229.0	0	230.0	0	157	499	0
13	2022-12-12 13:34:27	2022-12-12	25.2	54.0	229.0	1	278.0	0	195	616	1
14	2022-12-12 13:29:05	2022-12-12	25.1	54.0	230.0	1	407.0	0	110	494	0
15	2022-12-12 13:18:21	2022-12-12	25.4	54.0	228.0	1	342.0	0	180	601	0
16	2022-12-12 13:02:16	2022-12-12	25.4	54.0	229.0	1	306.0	0	81	448	0

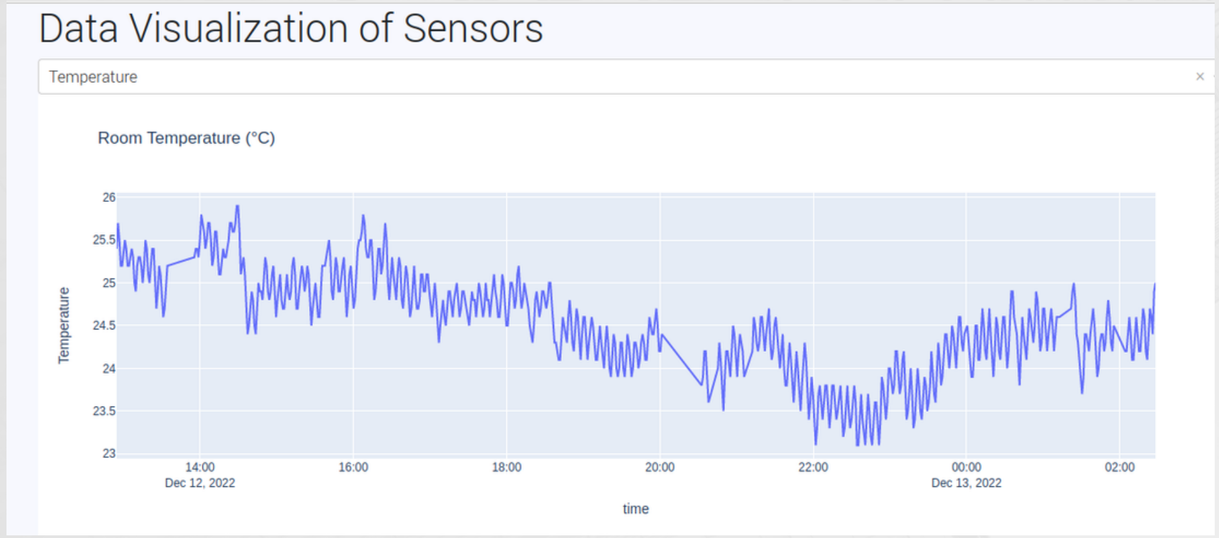
```
def push_db():
    data = pd.read_csv('sensors_data.csv')
    data.to_gbq(destination_table='sensor_data.sensor_db',project_id='smartroom-db',
if_exists='append')
    print('pushed')

push_db()
data_temp = pd.read_csv('data_temp.csv')
data_temp.to_csv('sensors_data.csv', index=False)
print('done push')
```

Push data to database

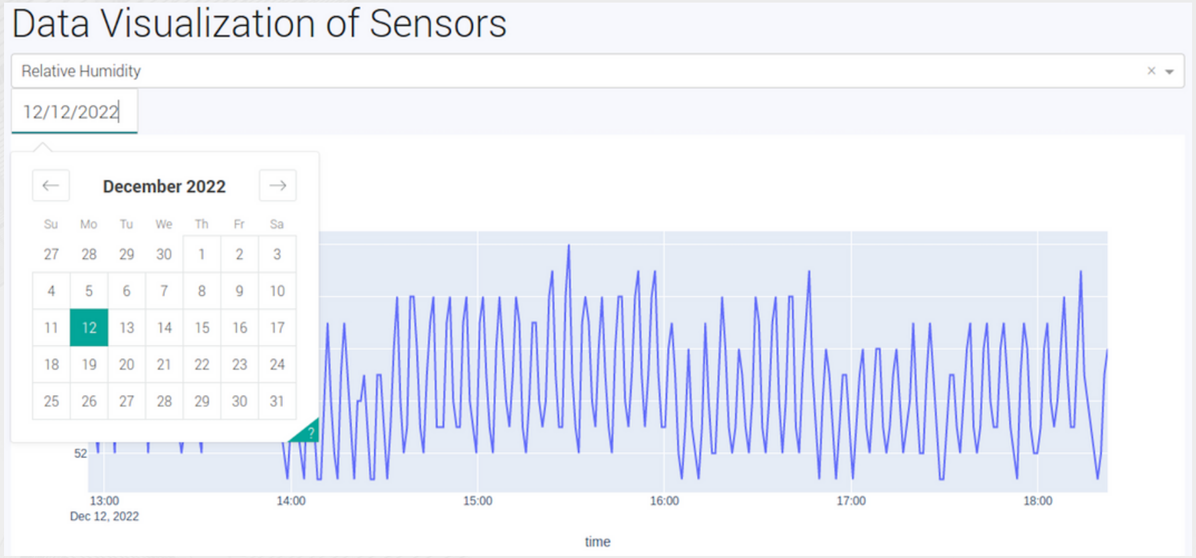
This block of code push the data from sensor_data.csv to the database table in Google Big Query

More Details in our group's github: https://github.com/gear-patt/Smart-Room/blob/main/sensor_data_serial.py

DASHBOARD (PLOTLY)



real-time dashboard



historical dashboard

More Details in our group's github:

- https://github.com/gear-patt/Smart-Room/blob/main/index.py
- https://github.com/gear-patt/Smart-Room/blob/main/app.py
- https://github.com/gear-patt/Smart-Room/blob/main/apps/historical.py
- https://github.com/gear-patt/Smart-Room/blob/main/apps/home.py
- https://github.com/gear-patt/Smart-Room/blob/main/apps/real_time.py

Model Training

```
# Initialze the estimators
clf1 = RandomForestClassifier(random_state=42)
clf2 = SVC(probability=True, random_state=42)
clf3 = LogisticRegression(random_state=42)
clf4 = DecisionTreeClassifier(random_state=42)
clf5 = KNeighborsClassifier()
clf6 = MultinomialNB()
clf7 = GradientBoostingClassifier(random_state=42)
```

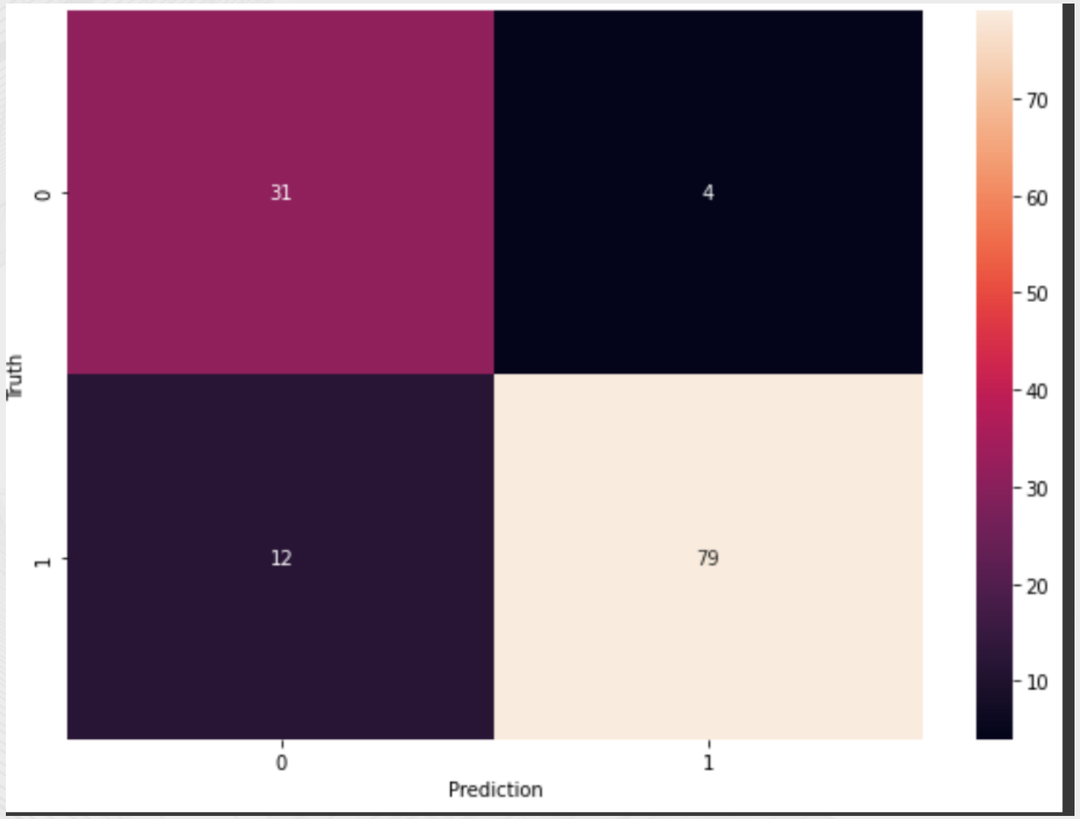
```
%%time
# Train the grid search model
gs = GridSearchCV(pipeline, params, cv=5, n_jobs=-1, scoring='precision').fit(X_train, y_train)
```

```
gs.best_params_

{'classifier': KNeighborsClassifier(n_neighbors=10),
 'classifier__n_neighbors': 10}
```

```
[83] gs.best_score_

0.9111377344460051
```

For the model selection the KNeighboursClassifer has the most precision of 91.11%.

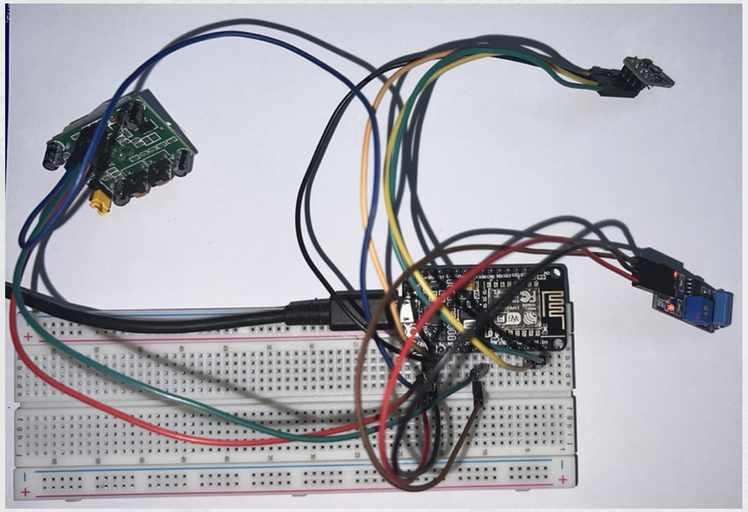
Confusion Matrix

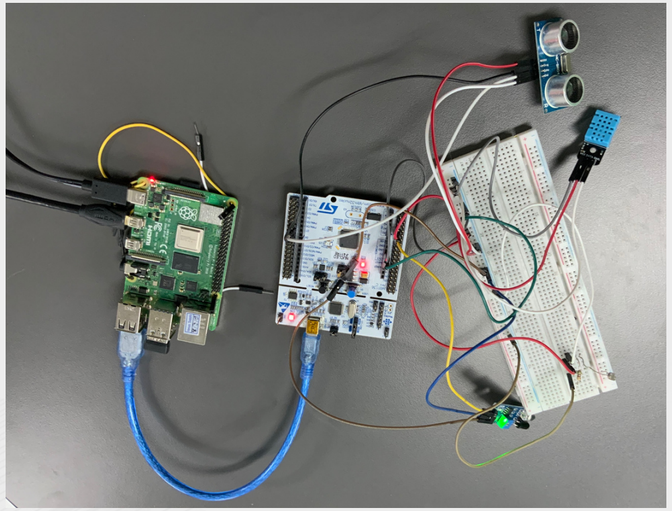


https://colab.research.google.com/drive/1qvvG5qmjm0jHKnrpbPue3Gtbo7XUXBgc#scrollTo=bb6jCOCQiAmP

EXPERIMENT SETUP

Sensor setup





Experiment Room

Demo Video

https://youtu.be/kicNKAMUOFI

CONCLUSION OF KPI

1. The light is immediately switched off by giving a command on the google assistant within 4 seconds

2. The specificity (true negative rate) of our test set data is at 88.6%. Meaning that it is able to accurately detect the absence of human presence 88.6% of the time.

3. The dashboard can visualize the real time data, updated every minute with 4-5 seconds delay. Historical data can be viewed every minutes in everyday and will be kept and visualize for more than a week.

4. By syncing device with Google Home, users can control the light both automatically and manually

5. The historical data is kept for 7 days and is shown to the user as a graph. The user can then determine and track all unnatural rising or falling trends to inspect

6. The system turns off when the prediction is made, which on average takes 10 minutes. For one hour, our system will save 50 mins on average lowering the consumption by at least 50%.

This project passes all three KPIs for functional tests and passes all three KPIs for problem aspects.

CONCLUSION

From our experimental result, the microcontrollers can work perfectly in both receiving and transmitting the sensor data. The light bulb and A/C are connected and controlled by google home. For a room with 3 hours of wasted energy requiring 64 KWH, our model specificity of 88.6% will saves on average 56 KWH per room of the 64 KWH. Extrapolated to the entirety of Bangkok, from the 1588 tons produced in the 3 hours that is typically wasted, this system saves 1397 tons of CO2 produced.

https://www.worldometers.info/co2-emissions/thailand-co2-emissions/

FUTURE WORKS

- Collect more data, to train more accurate models under different environments. The more dataset we have, the more accurate the model will be
- Implement the system with real A/C
- Decorate the Dashboard

THANK YOU