

Cover sheet for submission of work for assessment

UNIT DETAILS

Unit name	IoT Programming		Class day/time	Tuesday
Unit code	SWE30011	Assignment	Group	Due date
Name of lecturer/teacher	Siti Hawa Mohamed Said			
Tutor/marker's name				

STUDENT(S)

	Family Name(s)	Given Name(s)	Student ID Number(s)
(1)	Liang	Weng Kang	103834532, J19030225
(2)	Tan	Yun Wei	104202321, J22038599
(3)	Liew	Rui Sheng	104209144, J22038592

Table of Contents

Introduction	2
Conceptual Design	3-5
Tasks Breakdown	5-10
Implementation	11-17
User Manual	18-22
Limitation	23
Resources	24
Software	24
Appendix	25-28

Video Link: <https://www.youtube.com/watch?v=bBuypgHzZwA>

Introduction

The aim of this group assignment was to cooperate, learn and extend the practical knowledge that we have governed and absorbed during our tutorial classes and the individual practical assignment to develop an IoT project. The project is required to incorporate at least 3/4 edge devices (based on the number of students in a group), able to store data into a database, consisting of IoT communication, IoT cloud computing, APIs or web servers, data analysis and so on. Based on the required implementations needed, our group (Prison Break) have come up with an idea to develop a smart prison. A smart prison was the topic that we had gone for because not many IoT projects were conducted to accommodate a smart prison. The most common topics covered by many were probably smart house, office, farm, and town. Hence, going with smart prison will be the most safest choice as it will be rather different and a breath of fresh air. Prisons have always been around since the medieval ages until the modern era that we currently live in. Before IoT or technologies that are as advanced as the ones we have today, the infrastructure of the prison may be strong and solid but the security of things may be lacking.



On the positive side, it is a strong facility built to lock-up prisoners who had committed crimes that shall not be spoken of. On the negative side, prisoners may still be able to escape from their cells and have successfully managed to escape from prison. As a result, we hope our IoT project is able to lock the door remotely and tightly and shall only be opened with the keycard of the guards. As there are 3 of us in a group, each person is responsible to handle at

least 1 system. In our smart prison, the 3 systems used are a RFID smart door, sound alarm system and a fire alarm system. The types of sensors and overall system design will be covered and explored further down in this report.

Conceptual Design

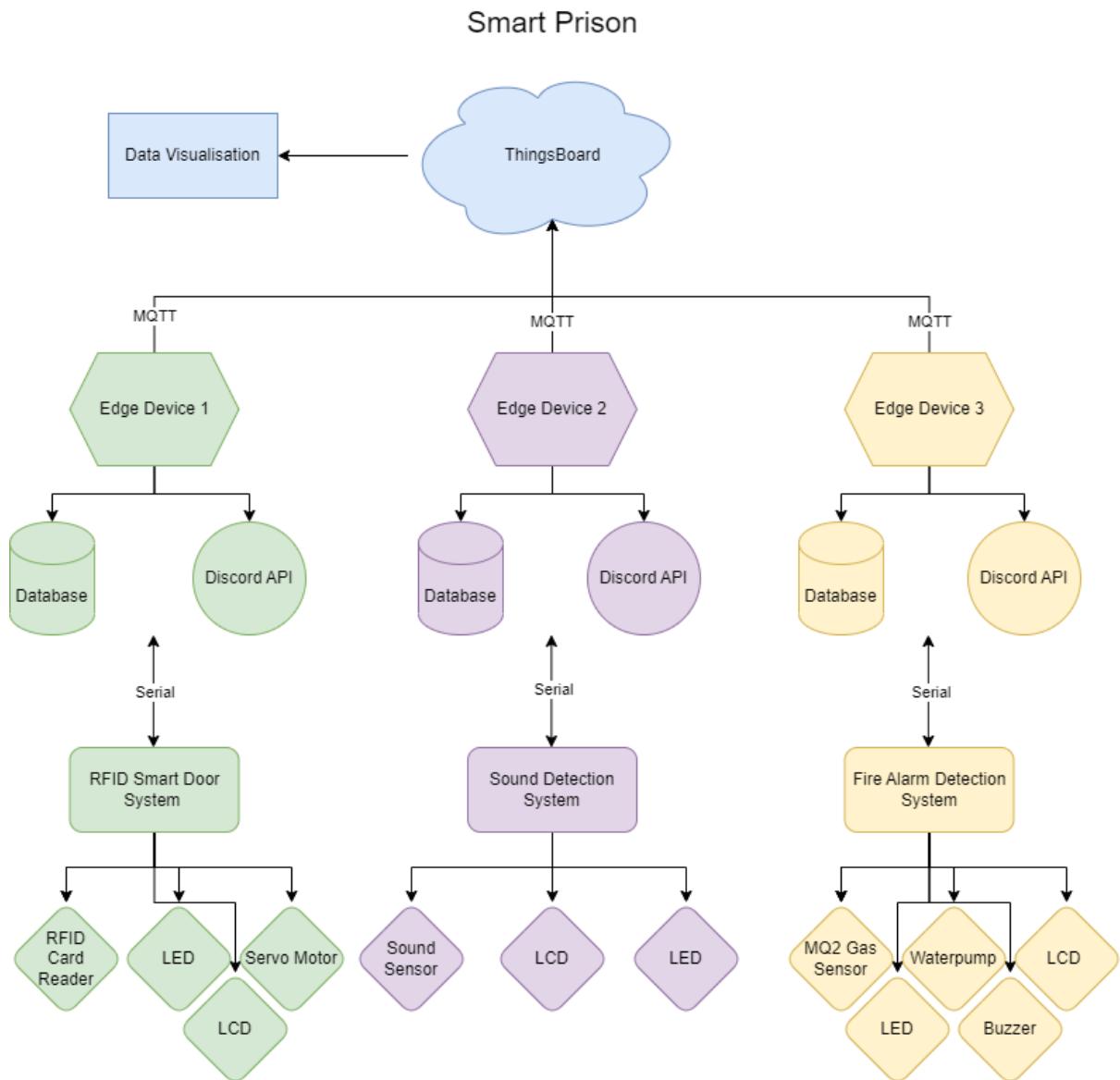


Figure 1: The overall flow of our system design.

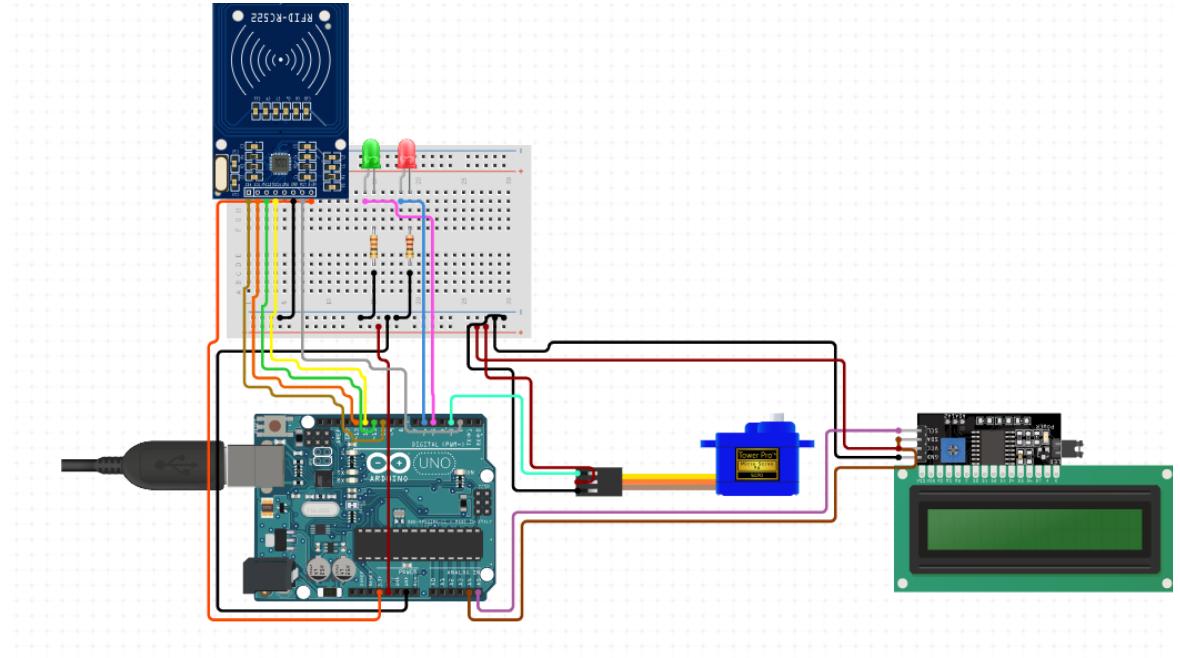


Figure 2: The concept of the circuit design of the **RFID Smart Door System**.

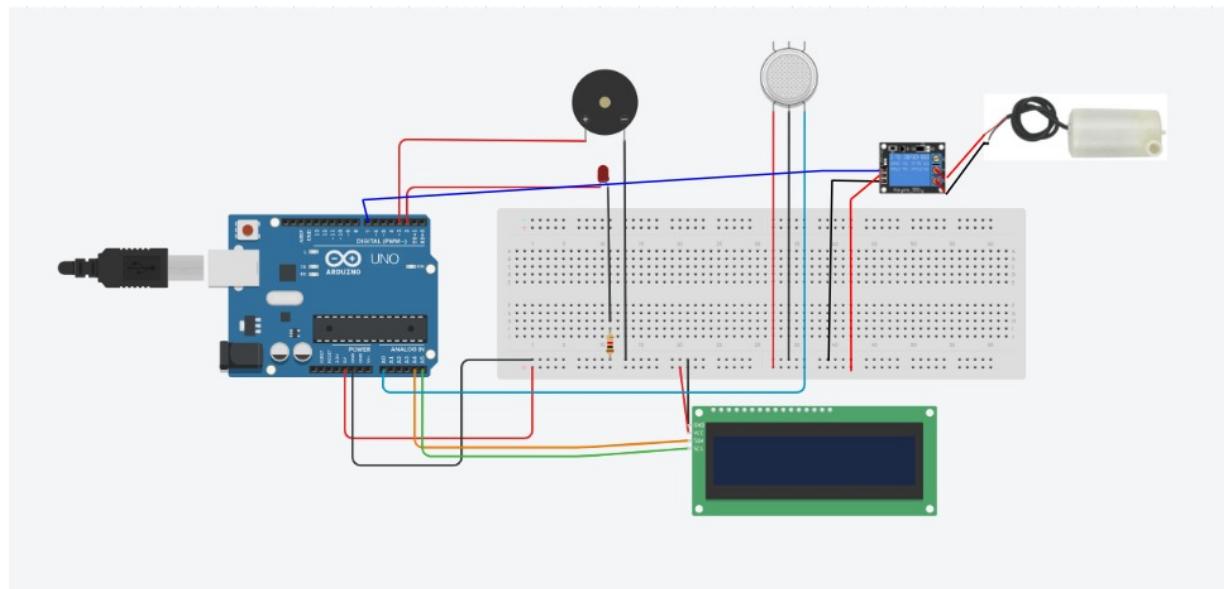


Figure 3: The concept of the circuit design of the **Fire Alarm System**.

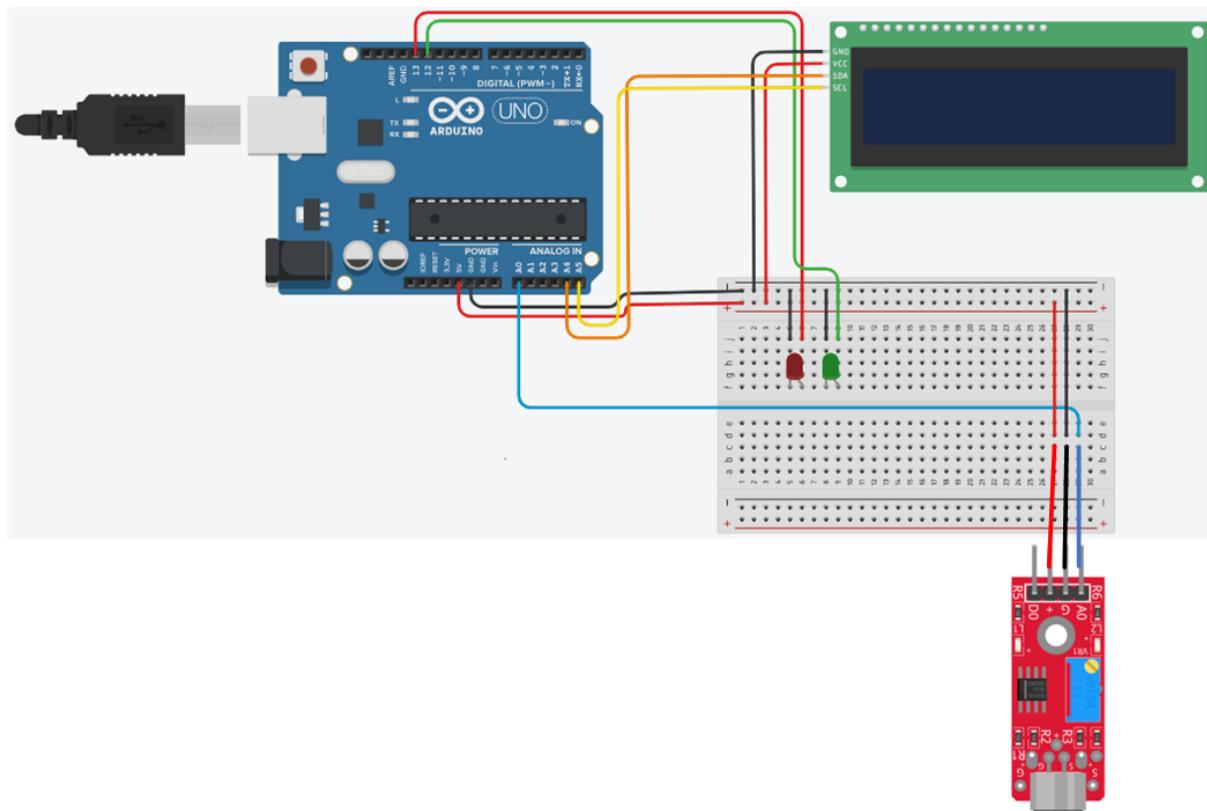


Figure 4: The concept of the circuit design of the **Sound Monitoring System**.

Tasks Breakdown

1. Tasks Breakdown and Work Report for Liew Rui Sheng

Task: Setup Hardware for RFID Smart Door System system

Description: Install and configure the smart door sensors and hardware components.

Task	Work Done
1. Sensor Setup	Completed
2. Hardware Test	Completed

Task: Implement Edge Server and RFID Smart Door system

Description: Set up the edge server using a Raspberry Pi and configure the database server.

Task	Work Done
1. Edge Server Setup	Completed
2. Database Setup	Completed

Task: Setup Discord Bot into Discord Channel

Description: Create a Discord bot account and set up a dedicated Discord channel for the RFID Smart Door system.

Task	Work Done
1. Discord Bot Setup	Completed
2. Discord Channel Creation	Completed

Task: Integrate Discord API into RFID Smart Door system

Description: Implement the Discord bot functionality using the Discord API library.

Task	Work Done
1. API Integration	Completed
2. Function Development	Completed

Task: Building a Smart Prison Model with All 3 Systems

Description: Integrate the RFID Smart Door system, Sound Monitoring system, and Fire Alarm system into a smart prison model.

Task	Work Done
1. Prototype	Completed

Task: Writing Report

Description: Prepare a comprehensive report documenting the project implementation.

Task	Work Done
1. Report Writing	Completed

2. Tasks Breakdown and Work Report for Tan Yun Wei

Task: Setup Hardware for Sound Monitoring system

Description: Install and configure the sound monitoring sensors and hardware components.

Task	Work Done
1. Sensor Setup	Completed
2. Hardware Test	Completed

Task: Implement Edge Server and Database for Sound Monitoring system

Description: Set up the edge server using a Raspberry Pi and configure the database server.

Task	Work Done
1. Edge Server Setup	Completed
2. Database Setup	Completed

Task: Setup Discord Bot into Discord Channel

Description: Create a Discord bot account and set up a dedicated Discord channel for the sound monitoring system.

Task	Work Done
1. Discord Bot Setup	Completed
2. Discord Channel Creation	Completed

Task: Integrate Discord API into Sound Monitoring system

Description: Implement the Discord bot functionality using the Discord API library.

Task	Work Done
1. API Integration	Completed
2. Function Development	Completed

Task: Connect Fire Alarm System and Sound Monitoring System into Thingsboard with MQTT Communication Protocol

Description: Establish communication between the fire alarm system, sound monitoring system, and Thingsboard using MQTT.

Task	Work Done
1. MQTT Integration	Completed
2. System Integration	Completed
3. Data Publishing to Thingsboard	Completed

Task: Building a Smart Prison Model with All 3 Systems

Description: Integrate the fire alarm system, sound monitoring system, and RFID smart door system into a smart prison model.

Task	Work Done
1. Prototype	Completed

Task: Writing Report

Description: Prepare a comprehensive report documenting the project implementation.

Task	Work Done
1. Report Writing	Completed

3. Tasks Breakdown and Work Report for Liang Weng Kang

Task: Setup Hardware for Fire Alarm system

Description: Install and configure the fire alarm sensors and hardware components.

Task	Work Done
1. Sensor Setup	Completed
2. Hardware Test	Completed

Task: Implement Edge Server and Database for Fire Alarm system

Description: Set up the edge server using a Raspberry Pi and configure the database server.

Task	Work Done
1. Edge Server Setup	Completed
2. Database Setup	Completed

Task: Setup Discord Bot into Discord Channel

Description: Create a Discord bot account and set up a dedicated Discord channel for the Fire Alarm system.

Task	Work Done
1. Discord Bot Setup	Completed
2. Discord Channel Creation	Completed

Task: Integrate Discord API into Fire Alarm system

Description: Implement the Discord bot functionality using the Discord API library.

Task	Work Done
1. API Integration	Completed
2. Function Development	Completed

Task: Connect Fire Alarm System and Sound Monitoring System into Thingsboard with MQTT Communication Protocol

Description: Establish communication between the fire alarm system, sound monitoring system, and Thingsboard using MQTT.

Task	Work Done
1. MQTT Integration	Completed
2. System Integration	Completed
3. Data Publishing to Thingsboard	Completed

Task: Building a Smart Prison Model with All 3 Systems

Description: Integrate the fire alarm system, sound monitoring system, and RFID smart door system into a smart prison model.

Task	Work Done
1. Prototype	Completed

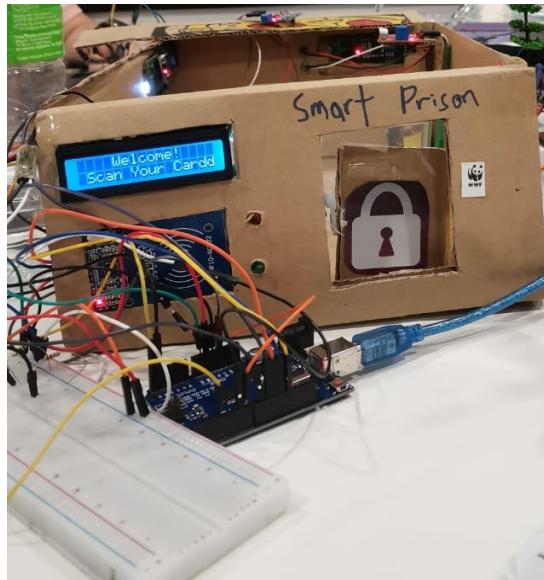
Task: Writing Report

Description: Prepare a comprehensive report documenting the project implementation.

Task	Work Done
1. Report Writing	Completed

Implementation

1. RFID Smart Door System



• Sensing System

In this system, the main sensor used will be the RFID card scanner. The RFID scanner is mainly used to read the UIDs of our student ID to represent that we're the guards of the smart prison. Normally the RFID card scanner in IoT comes together with 2 things. A RFID card and tag. But, in this case, I wanted to use the RFID scanner in a different way. Hence, I had used it to scan for the UIDs of our student ID. As for the actuator aspect, I had used LCD I2C display to show the result based on every scan whether it be "Access Granted" or "Access Denied". Similarly, two LEDs are used for the same purpose. Both being the red and green LEDs. The red LED will light up for "Access Denied" and the green LED will light up for "Access Granted". Lastly, the servo motor is being utilized as a way to unlock the door. Its default position is at 0 which indicates the door is locked. Once the door is unlocked, the servo motor position will be altered to 180. Hence, opening the door in a vertical way.

• Edge Servers

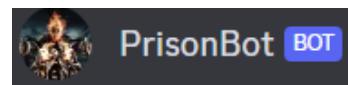
In the edge server, the main function used is to read and process the data from Arduino and to create a database that is mainly used to store the UIDs of our student ID. As a result, I had used the pymysql library to connect to the database to check and

process the data further. Moreover, the edge server will connect to the Discord API by connecting with the Discord bot by using the token and channel ID.

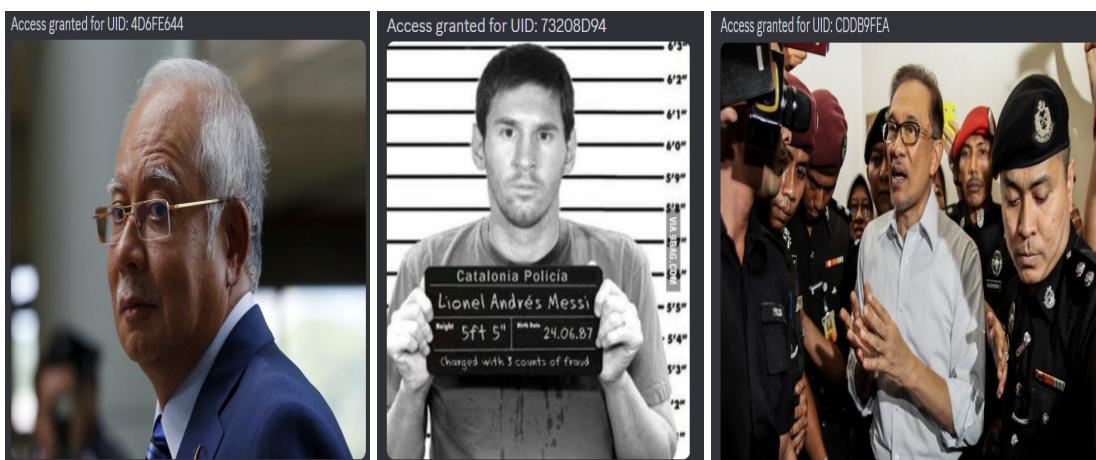
- **Communication Protocols**

As there are 3 of us in a group, the 3 UIDs stored in the database are the only UIDs that have the record of “Granted” in the access column. Hence, for security purposes, the edge server will connect to that database in order to check the access and uid columns. To showcase the edge processing, the edge server will read the UID of the student ID from the serial communication of the Arduino side. If the serial UID matches the UID in the database, it will send a message of “o” back to the Arduino to unlock the door. This will properly showcase the bidirectional communication between the Arduino and edge server.

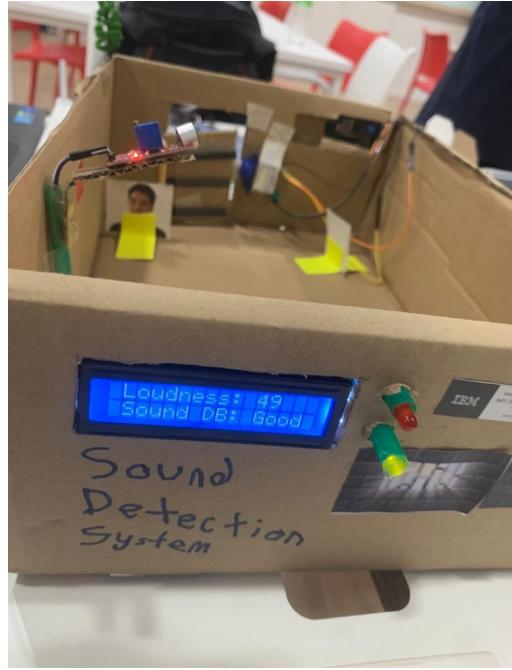
- **API**



As stated previously, the Discord API will be used and connected to the edge server if and only if the Discord token and channel ID matches in the python script. It has to be depending on both criteria. Otherwise, it won’t connect at all. The Discord bot used in our project is named PrisonBot. Furthermore, inside the Discord API, it is being used to display a message and an image for the granted UIDs as the result. For the UIDs that are not granted, it won’t be displayed in the Discord API at all. This way, it can demonstrate the proper usage and communication of the Discord API with the edge server. Examples of the results of the 3 guards UID in Discord will be shown below.



2. Sound Detection System



• Sensing System

The sensing system in this Sound Monitoring System has been used is KY-037 microphone sound detection system. This sensor senses the sound from the surrounding, when the sound of the threshold exceeds the threshold it will indicate a message. The sound sensor needs to put in the horizon way at the box with the straw holding like a light pole and also need to turn the potentiometer which is like a flat screw cap at the component, that is to adjust the sensitivity of the sound sensor to receive the incoming sound from the surrounding. The RED_LED represents the alarm, when the threshold is reached, the red light indicates and the LCD will display the condition in bad. The LCD also works as the serial monitor to show the output instead using the IDE serial monitor.

• Edge Servers

For the edge server site, the edge server is responsible to store the data and read the data bidirectional by using python Serial and MySQL Database. The edge server site allows the data to communicate with the web server.

- **Communication Protocols**

WebSocket : Establish a TCP connection between Discord Bot and the users, when the user has any request from the chatbox, the discord bot senses the related information is matched, the discord bot will respond to the related information.

MQTT (Message Queuing Telemetry Transport) : The MQTT protocols function as the messaging protocols from a machine to a machine communication. For example, Thingsboard cloud is able to get the data from the sensor which has been connected to the edge server with only minimal resources with high efficient transmission.

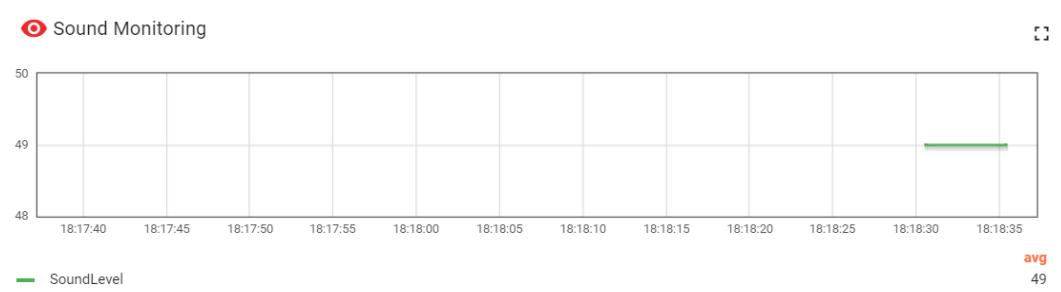
- **API**

Discord API is able to share the data, modify the threshold and send the alert when the threshold reaches the designated value. Inside the Discord API, the token needs to be implemented before the server can be run inside the python file.

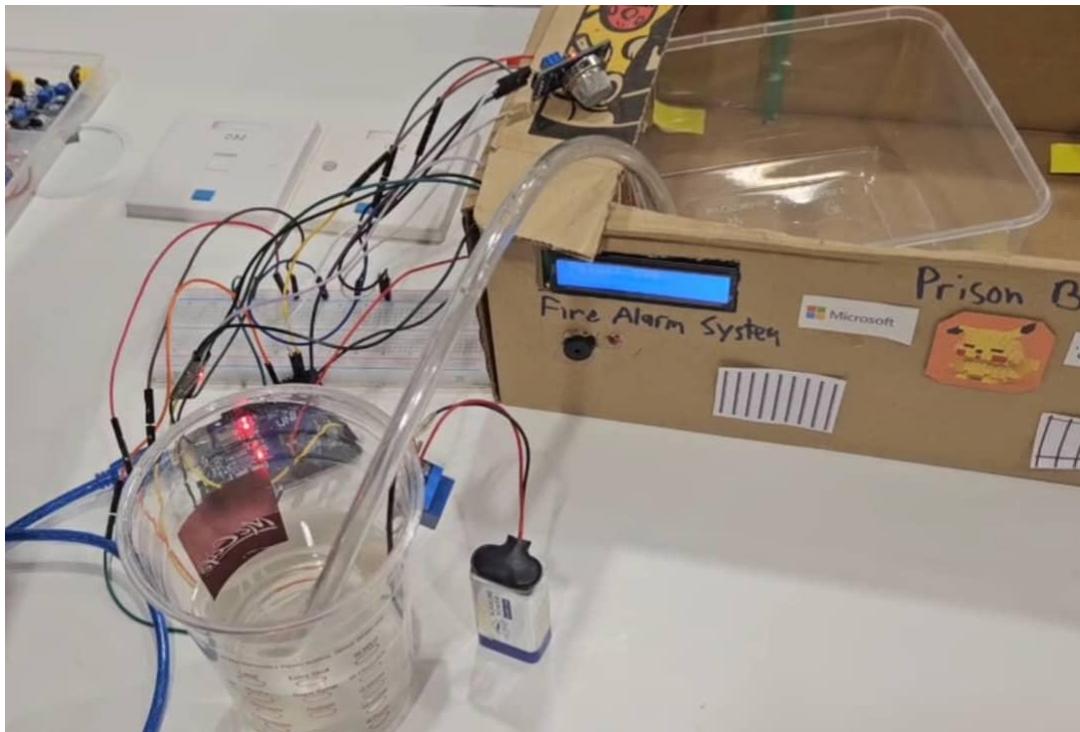
- **Cloud Computing**

The Application used to build the cloud server is Thingsboard. This cloud application is able to send the data to the cloud and do the real time monitoring. Data will visualize in the graph to let the user know how the trend is for the device for detection.

Real time monitoring for Sound in Thingsboard



3. Fire Alarm System



- **Sensing System**

In this system, the sensing system is responsible for measuring the smoke level in prison. The sensing system continuously reads the smoke level data and sends it to Edge Servers to do processing. The Fire Alarm System includes one sensor which is an MQ2 gas sensor. There are some actuators which are LCD for displaying the value received from the sensor, Red LED for indicating whether the alert system is triggered, Buzzer for indicating whether the alert system is triggered, Water Pump for dispense water automatically or manually when the alert system is triggered. When the smoke level surpasses a predefined threshold in the prison, the alert system will be triggered. The alert system consists of a Buzzer, LED and Water Pump.

- **Edge Servers**

The edge servers are responsible for processing the data received from the Arduino board and perform appropriate actions. The edge server connects to a MySQL database using pymysql library to store the collected data and retrieve threshold values. The edge server uses the Discord API and the discord library to integrate with Discord for sending alerts and notifications. For example, when smoke level exceed the

threshold value, the system will automatically send a notification to the Discord channel.



- **Communication Protocols**

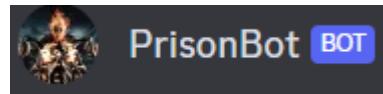
- Serial Communication

The Arduino board communicates with the edge server over a serial connection using the serial library. The edge serve have bidirectional communication allowing the

- MQTT (Message Queuing Telemetry Transport)

The edge server communicates with the ThingsBoard platform using the MQTT protocol. The paho.mqtt.client library is used to establish an MQTT connection and publish data to the ThingsBoard broker.

- **API**

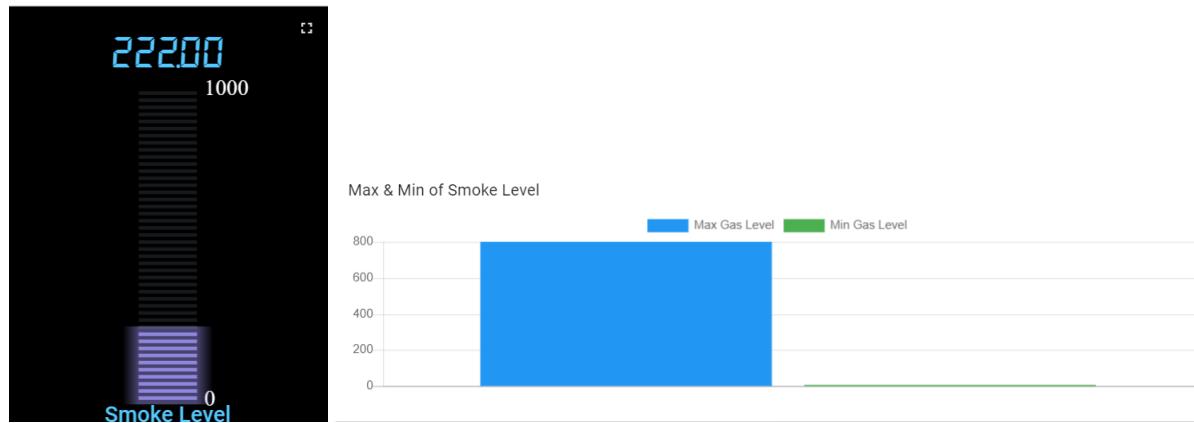


The system implements a Discord bot that can receive commands and send alerts to the specified channel. The Discord bot is named PrisonBot in our Discord server. Through integrating Discord API, there are commands that can be used to retrieve the latest data from the database, update and retrieve the threshold value of the fire alert system, and stop the Water Pump when the alert system is triggered. This allows the users to manually control the Water Pump when the fire alert system is triggered.

- **Cloud Computing**

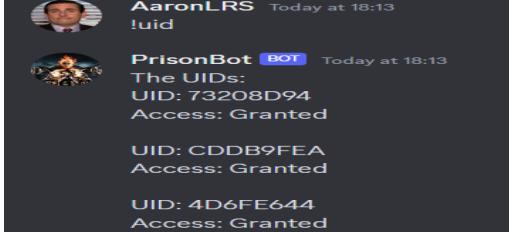
The implementation incorporates cloud computing through the use of the ThingsBoard platform. The edge server publishes the smoke level and maximum and minimum level of

smoke level to the ThingsBoard platform using MQTT. The ThingsBoard platform receives the data, processes it, and provides visualization and analytics capabilities through its website

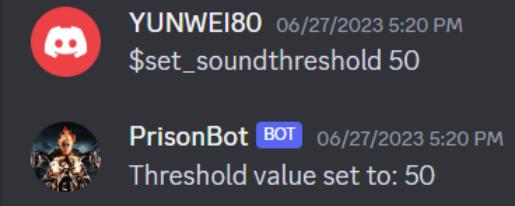
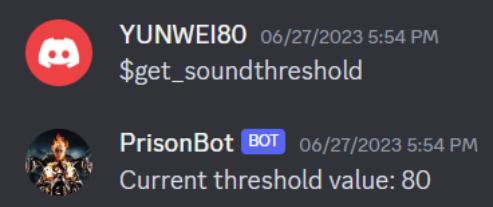


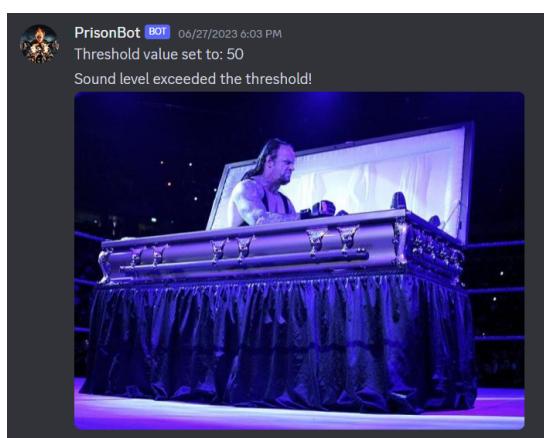
User Manual

1. User Manual for RFID Smart Door System in Smart Prison

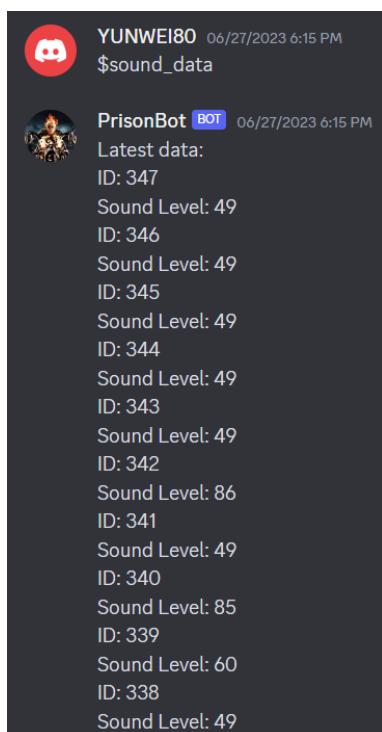
Steps	Tasks
Step 1	Connect and upload the Arduino code to the Raspberry Pi to form an edge device.
Step 2	Run the Raspberry Pi python script that connects to the database, Discord bot, and bidirectional communication with the Arduino.
Step 3	Use the discord command !uid to display the lists of guards' uid that are granted access to unlock the door. 
Step 4	Use and scan one of the Student ID of one of the group members as the security guard of the smart prison to unlock the prison door.
Step 5	View the LCD message and view the conditions of the LED lights. If it's "Access Granted" on the LCD, the green LED should light up. Vice versa, the red LED should light up for "Access Denied" on the LCD.
Step 6	Check the debug message in Raspberry Pi as well as the displayed message in Discord API.
Step 7	Repeat steps 1 - 4 but this time use another RFID card/tag to scan at the RFID scanner.
Step 8	Check the debug message in Raspberry Pi as well as the displayed message in Discord API again.

2. User Manual for Sound Monitoring System in Smart Prison

Steps	Tasks
Step 1	Connect the Arduino UNO to the machine with the fully functions sensors and actuators.
Step 2	Enable the Arduino IDE and run the code and run the Raspbian web server machine as an edge device.
Step 3	Run the SoundConnect.py and SoundCloud.py files which need to boot up the Discord Bot and Cloud Server (Thingsboard)
Step 4	<p>Run Discord which has set the group as PrisonBot Group and run the specific command which to Set and Read the threshold value from the database.</p> <ul style="list-style-type: none"> - \$set_soundthreshold {value} – Set the sound threshold (limit)  <pre> YUNWEI80 06/27/2023 5:20 PM \$set_soundthreshold 50 PrisonBot BOT 06/27/2023 5:20 PM Threshold value set to: 50 </pre> <ul style="list-style-type: none"> - \$get_soundthreshold - View the sound threshold  <pre> YUNWEI80 06/27/2023 5:54 PM \$get_soundthreshold PrisonBot BOT 06/27/2023 5:54 PM Current threshold value: 80 </pre> <p>Trigger Alert: When threshold exceed maximum</p>



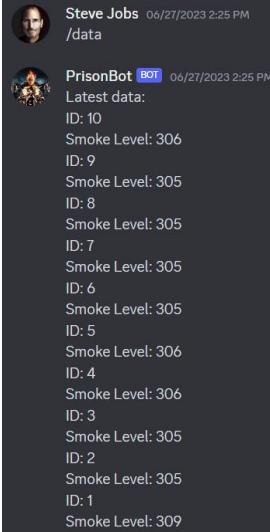
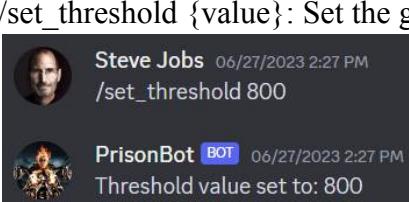
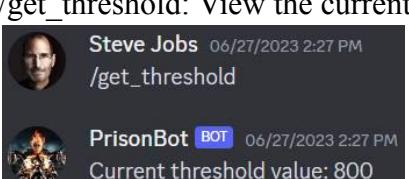
\$sound_data - Get the latest **10** input data.



Step 5	Both files are activated and then run Thingsboards cloud server on the web.
Step 6	Go to the dashboard and view the real time Graph.

3. User Manual for Fire Alarm System in Smart Prison

This user manual will guide you through the main operations of the system. Please follow the instructions below:

Steps	Tasks
Step 1	Connect and upload the Arduino board to Raspberry Pi
Step 2	Run the gas_discord.py and gas_cloud.py <ul style="list-style-type: none">● Execute the gas_discord.py to start reading and storing the sensor data and start Discord bot● Execute the gas_cloud.py to establish a connection with the Thingsboard cloud server.
Step 3	Run Discord and use the specific commands to interact with the fire alarm system through Discord. <ul style="list-style-type: none">● /data: Get the latest 10 data values. A screenshot of a Discord chat. Steve Jobs sends a message at 06/27/2023 2:25 PM with the command '/data'. PrisonBot responds with a list of 10 smoke level readings: ID: 10, Smoke Level: 306; ID: 9, Smoke Level: 305; ID: 8, Smoke Level: 305; ID: 7, Smoke Level: 305; ID: 6, Smoke Level: 305; ID: 5, Smoke Level: 306; ID: 4, Smoke Level: 306; ID: 3, Smoke Level: 305; ID: 2, Smoke Level: 305; ID: 1, Smoke Level: 309.● /set_threshold {value}: Set the gas threshold to the specified value A screenshot of a Discord chat. Steve Jobs sends a message at 06/27/2023 2:27 PM with the command '/set_threshold 800'. PrisonBot responds with a confirmation message: 'Threshold value set to: 800'.● /get_threshold: View the current gas threshold value. A screenshot of a Discord chat. Steve Jobs sends a message at 06/27/2023 2:27 PM with the command '/get_threshold'. PrisonBot responds with a confirmation message: 'Current threshold value: 800'.● /stop_alert: Stop the water pump when alert system is active

	 <p>Steve Jobs 06/27/2023 3:26 PM /stop_alert</p>  <p>PrisonBot BOT 06/27/2023 3:26 PM Alert has been turned off.</p>
Step 4	Go to https://demo.thingsboard.io/home
Step 5	Access the Thingsboard dashboard to view real-time graphs and analytics.

The Fire Alarm System offers an efficient way to monitor smoke levels and detect fire in a prison environment. By following the instructions provided in this user manual, you can effectively use and manage the system to ensure the safety and security of the prison facility.

Note: Ensure that the edge server, database, Thingsboard and Discord bot are running and properly configured for the system to function correctly.

Limitations

During the entire process of completing this IoT smart prison system, our implementations aren't exactly flawless as we would've imagined. There are without a doubt hiccups along the way. Even though there are solutions used to solve the problems related to the coding aspect of things or the designing of the model, there are still some limitations that hinders our IoT smart prison project from reaching its full capacity. The first limitation would be the lack of overall power supply to the systems. The reason for this limitation is because each system is being handled by different individuals amongst us. The workload was distributed to divide 1 system to 1 person. As a result, all the systems are being powered by our laptops separately in order to function. No external power supply was being used in this project. Thus, it provides a limitation as we have to always rely on our laptops individually in order to power up each related system in the smart prison.

Moreover, the second limitation is the non-existent website in our project. As we had already decided and had already demonstrated that our smart prison will be using Discord API, that may bring limitations to our system as users may face some difficulties if they're not familiar with the usage of Discord. Discord is a social platform in the modern era where it is accustomed to many younger individuals as their target users. The older generation of individuals may feel out of place or uneasy if they need to view the data or result through the Discord API. They may feel like viewing the data freely by browsing a website instead of typing the necessary Discord bot commands in the Discord API. Even if the bot commands are simple and user-friendly, this may still bring limitations to our system.

Resources

References

1. Mendizábal, I. (2022) *IOT communication protocols-IOT data protocols - technical articles, All About Circuits*. Available at: <https://www.allaboutcircuits.com/technical-articles/internet-of-things-communication-protocols-iot-data-protocols/> (Accessed: 01 July 2023).
2. amazon, amazon (1978) *What is ...*, Amazon. Available at: <https://aws.amazon.com/what-is/mqtt/> (Accessed: 01 July 2023).
3. Discord, D. (2023) *API docs for bots and developers*, *Discord Developer Portal*. Available at: <https://discord.com/developers/docs/reference> (Accessed: 01 July 2023).
4. kinsetsu, kinsetsu (2023) *IOT and the prisons of the future*, *Kinsetsu*. Available at: <https://kinsetsu.co.uk/iot-and-the-prisons-of-the-future/> (Accessed: 01 July 2023).
5. Blakinger, K. (2022) *Texas prisons defy fire safety standards. now, 2 men have died.*, *The Marshall Project*. Available at: <https://www.themarshallproject.org/2022/05/09/burned-to-death-in-a-prison-cell> (Accessed: 01 July 2023).
6. Emmet, E. (2022) *Setup a raspberry pi mysql database*, *Pi My Life Up*. Available at: <https://pimylifeup.com/raspberry-pi-mysql/> (Accessed: 01 July 2023).
7. Thingsboard, T. (2023) *MQTT device API reference*, *ThingsBoard*. Available at: <https://thingsboard.io/docs/reference/mqtt-api/> (Accessed: 02 July 2023).
8. Admin, A. (2022) *IOT smart farming using Blynk app: Control pump using IOT*, *Techatronic*. Available at: <https://techatronic.com/blynk-iot-smart-farming-using-water-pump/> (Accessed: 02 July 2023).

Software

Tinkercad & Circuito.io - For Circuit diagram design

Raspbian Virtual Machine - Edge Server

Arduino IDE - To compile the code to Arduino UNO

Discord Bot & API - To get automated message from command

ThingsBoard - Create a cloud server to monitoring the real time data

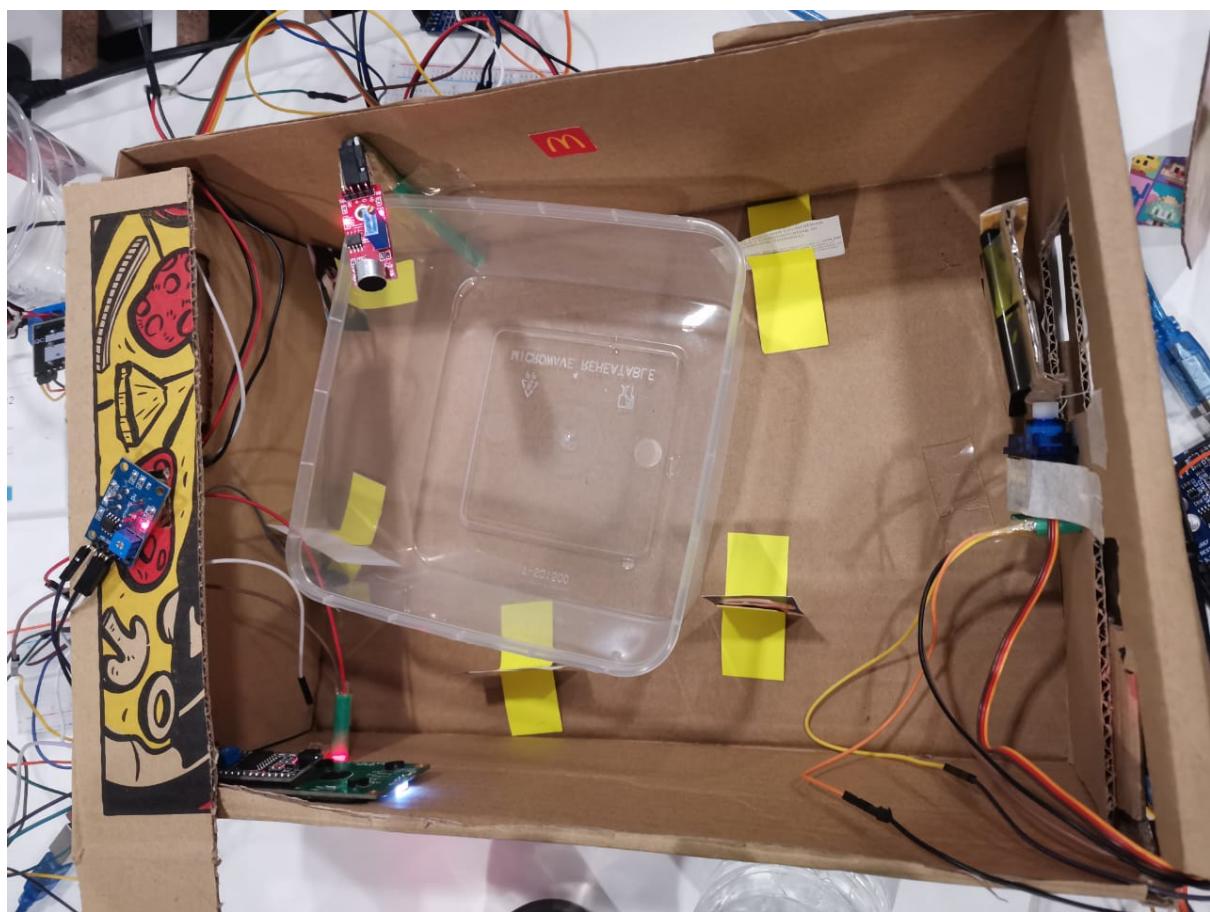
Appendix

Prototype of the Smart Prison

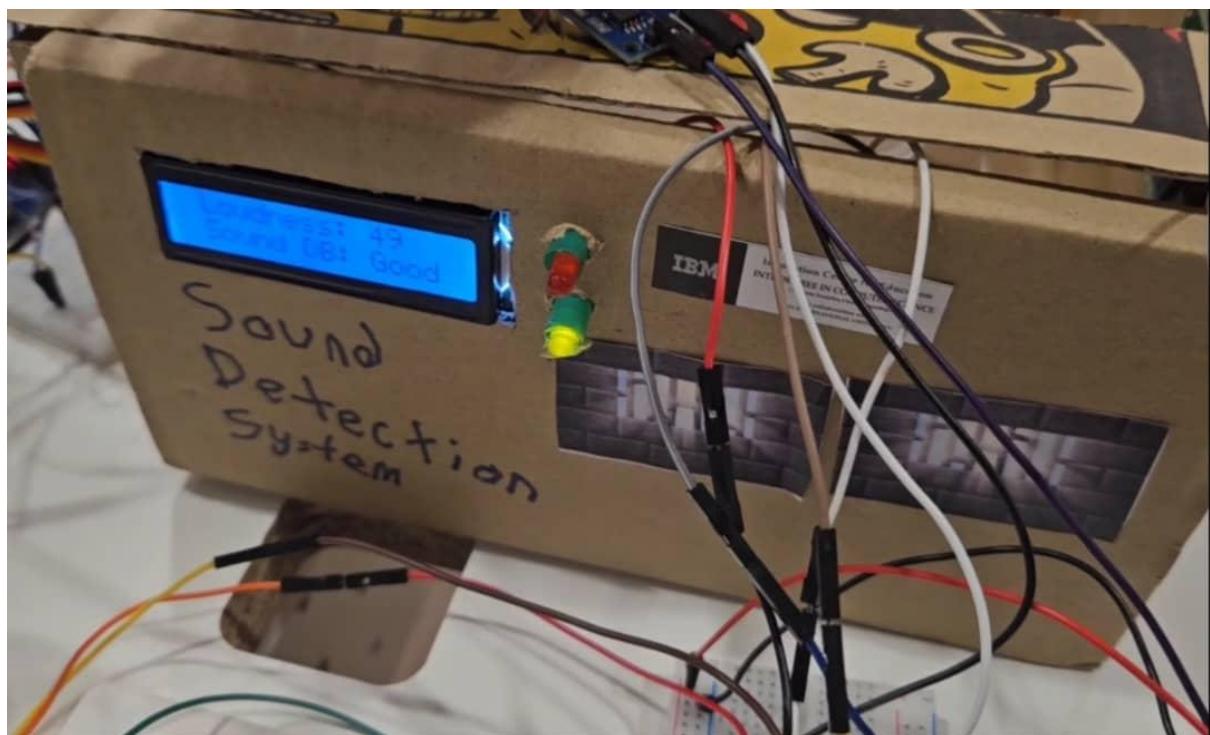
Front View



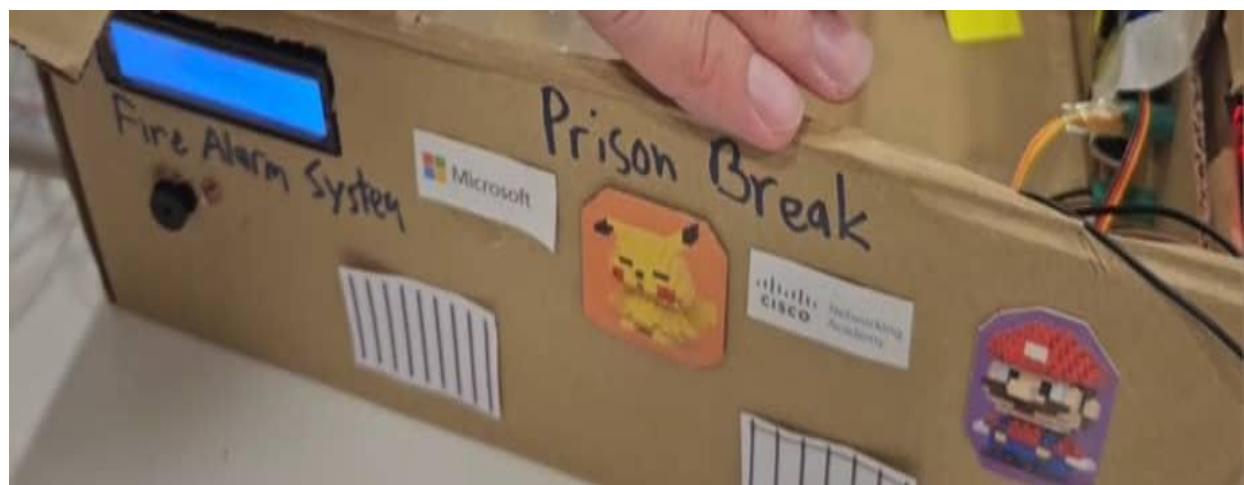
Top View



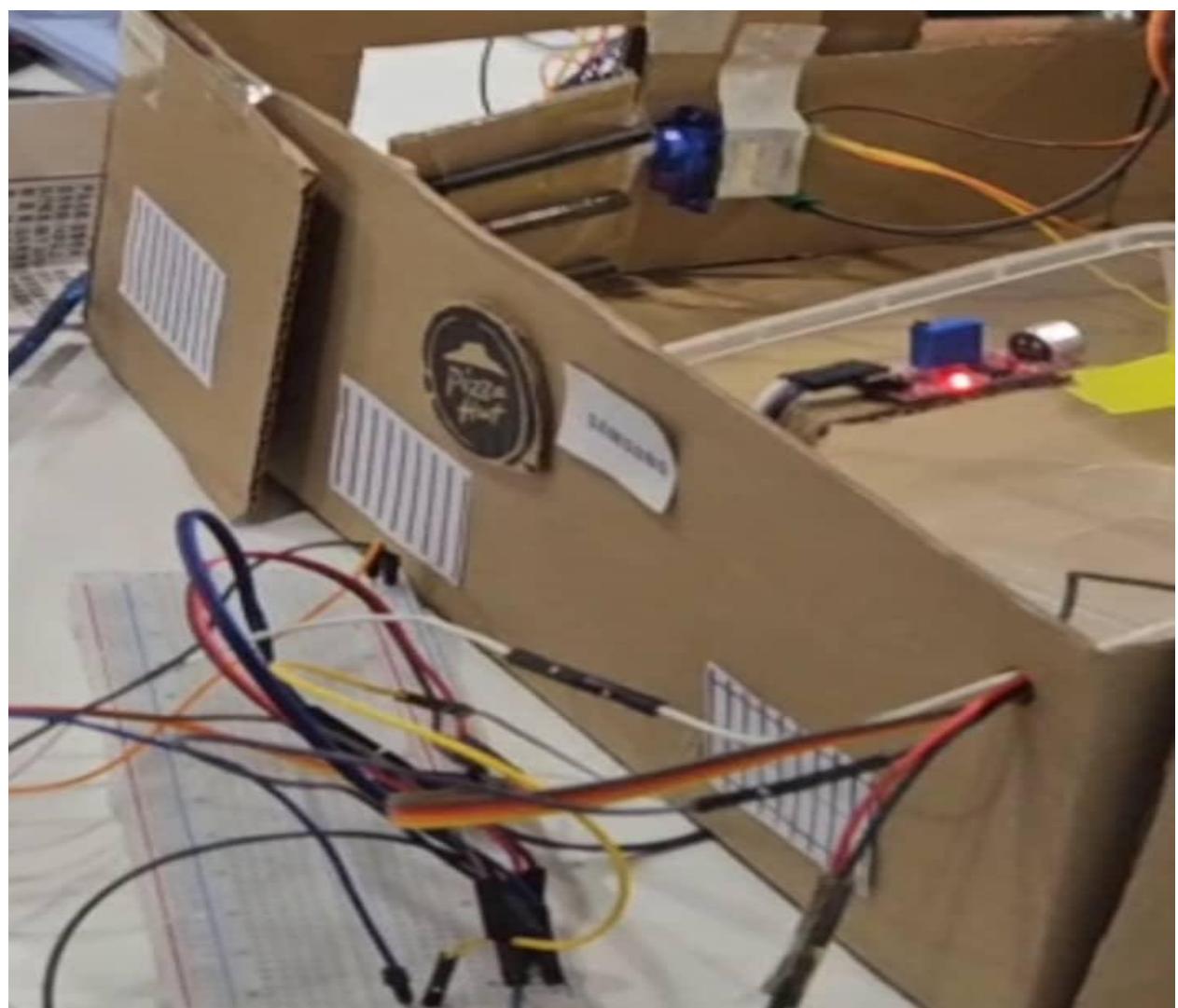
Back View



Left View



Right View



Thingsbaord Login Credentials

The image shows a login interface with a dark blue header and a white form area. At the top left is a logo consisting of a white 'T' inside a blue rounded square. To its right is the text 'Thingsbaord' in white, with 'Things' in a smaller font above 'baord'. Below this is a subtext 'Your Personal Dashboard' in a smaller, lighter blue font. The main form area has a light gray background. It contains two input fields: a 'Username (email)*' field with a mail icon and the value 'kangwengliang6818@gmail.com', and a 'Password' field with a lock icon and the value 'prisonbreak'. To the right of the password field is a visibility toggle icon. Below the form is a 'Forgot Password?' link. At the bottom is a large orange button with the word 'Login' in white.

Username (email)*
kangwengliang6818@gmail.com

Password
prisonbreak

Forgot Password?

Login