Smart Garbage In-house Segregation Plant, GARB-I

Group 10: Akshay Kumar Jaju and Vedant Girish Dalvi

Service Computing Department, IAAS, University of Stuttgart
st180311@stud.uni-stuttgart.de
st179677@stud.uni-stuttgart.de

Abstract. Today, huge amounts of waste/garbage are generated every day, which has further piled pressure on landfills and caused eco- logical concerns worldwide. Recycling seems to be the best long-term a solution to this problem. In this project, we plan to implement a "Smart Garbage In-house Segregation Plant - GARB-I" which can give impetus to garbage recycling in hotels and help them in fulfilling the mandatory recycling requirements. This project plans to execute garbage segregation into Biodegradable and Non-Biodegradable waste and also provide a comfortable, energy-efficient, and safe working environment for the hotel staff. This project has been implemented with a Raspberry Pi 3 B+ board and sensors like humidity, temperature, air quality, ultrasonic sensor, and actuators like motors and relays. The data from these sensors is processed by the Raspberry Pi and AI planner. The data published via Sensor and Actuator topics would be published via indirect communication through MQTT and subscribed for visualization by a dashboard.

Keywords: Waste/Garbage Segregation · Ultrasonic Sensor · Safety · Comfort · Energy saving · Hotels.

1 System Introduction

1.1 Background information

On average, hotels generate around 1 kg of waste per guest per night. Any product that cannot be reused and becomes waste should be sorted into its component fractions so that, as much as possible, it can be recovered for recycling. It is estimated that at least 70 percent of waste generated at hotels can be recycled, provided that there is a functional and effective separation and collection system in-situ. In order to achieve these results, it is essential to consider waste separation in the hotels and establish an appropriate sorting system.

1.2 Problem Statement

Keeping in mind that comfort in rooms is a main objective in hotels, there are different environmental practices which are being implemented without reducing well-being of guests while generating environmental benefits. In most cases, hotel rooms only include a couple of waste bins, located in the bathroom and bedroom, where waste fractions are mixed. While bins located in the bathroom are intended for toilet waste, the one in the bedroom is used to collect all types of litter generated by guests (i.e., plastics, magazines, bio-waste, etc.). This is the bin that holds the largest potential to be adapted to a more sophisticated waste sorting and collection system. Currently, individual small-sized bins adapted for separation of different fractions (i.e., paper, plastic, glass and food waste) are presented as a solution [1]. An alternative could be the placement of several bins for different fractions in the room, although it would require more space and therefore it is less recommended. The hotel is then responsible for waste sorting and management and will make sure that all waste fractions are properly separated in their respective container. Later, an authorised waste manager takes care of the waste generated at the hotel and collects it periodically from the respective facilities. However, this solution leads to increased costs and labour. Even if we gave labour, because of uncomfortable and unsafe working environment, labour refuse to work in such environment. We propose a smart garbage sorting plant situated in the hotel which can segregate the waste in a planned and structured way and prevent this hassle. Also, it would provide a good and safe working conditions for the staff in the segregation plant and also would not disturb the hotel's regular operation.

1.3 Project Goals

The system aims to be a smart waste segregation plant which is energy-efficient, safe and comfortable for it's employees. It is achieved by,

1. Monitoring the segregation machine bin status, which gives the real-time bio and non-bio bin's capacity(i.e. whether the bins are filled or empty). This ensures time efficient operation of the working staff.

2. Temperature/humidity levels, air quality levels, human presence in the garbage segregation facility are continuously monitored. Automation of the room lighting and room ventilation is controlled by using this data. In addition, the plant lighting is controlled on the basis of human room presence. This ensures power efficient operation of the plant.

3. The real-time monitoring of the HVAC conditions and air quality levels in the plant also helps in alerting the facility manager as well as workers present in the plant in case of fire emergencies and poor air quality thus, accounting for their safety.

All three points reflect a smart work environment.

2 System Analysis

2.1 Functional Requirements

2.1.1 Mandatory Requirements

2.1.1.1 User Story: Hotel Guest waste disposal
As a **Guest**,
I want to **dispose my waste in one can**,
So that **I don't have to sort it manually in my room**

2.1.1.2 User Story: Energy/power Efficiency
As a **Hotel Manager**,
I want to **automate the HVAC and lighting in the plant**,
to have **energy savings and efficiency**

2.1.1.3 User Story: Safety alert and Security
As a **Hotel Manager**,
I want to **get alerts in case of an accidental fire or poor air quality related emergencies**,
to prevent **damage to property and lives.**

2.1.1.4 User Story: Manager notified when sorted bins are getting full
As a **Facility Manager**,
I want to **be notified when the sorted bins in the plant are getting full**,
So that **I can instruct the working staff to collect the separated materials**.

2.1.1.5 User Story: Room Service Trash Collection
As a **Room Service Personnel**,
I want to **collect trash from one can and put it in one can**,
So that **I can save time and cover more rooms**.

2.1.1.6 User Story: Hotel Worker having comfort working environment in sorting plant
As a **hotel worker**,
I want to **work in a clean and safe work environment**,
So that **I don't be uncomfortable and risk my health**.

2.1.2 Desirable Requirements

2.1.2.1 User Story: Hotel Worker managing bulk sorting
As a **hotel worker**,
I want to **just put all trash collected into the sorting machine**,
So that **the machine can sort them into the allocated bins**.

2.2 Non-Functional Requirements

2.2.1 Mandatory Requirements

2.2.1.1 User Story: Hotel Certification
As a **Facility Manager**,
I want to **increase waste separation and recycling rates**,
So that **the hotel can get certified with ISO 14001, EMAS, etc.**.

2.2.1.2 User Story: Hotel Infrastructure
As a **Hotel Manager**,
I want to **employ cost-effective and sustainable waste management**,
So that **the hotel doesn't have to hire more workers and only invest in infrastructure once and reap the rewards for years**.

3 System Architecture Design

Fig. 1 shows the System Architecture for the Smart Garbage In-house Segregation Plant, GARB-I.

The system architecture includes various elements. Generally, the system structure can be divided into four (4) layers, which are user layer, reasoning layer, ubiquitous layer and physical layer.

Sensors and actuators are part of the physical layer. The sensors collect information about various parameters in the environment, which can be converted into data for analysis or for actuation under certain predetermined conditions. The actuators perform the necessary actions to attain the desired goals. The interaction between the sensors and actuators is orchestrated by the raspberry pi.

In the ubiquitous layer, the data management and computing of the data from the sensor to the intelligent system happens. Also, the data from sensors (Ultrasonic sensors, IR sensors, Temperature humidity(DHT) sensors) is easily computed by the intelligent computing system in the reasoning layer. The reasoning layer is responsible for AI planning, intelligent waste recognition, indirect communication, and giving feedback to the client UI. AI planning is employed to actuate the HVAC equipment i.e. exhaust fans. Whenever the temperature and air quality increase beyond normal range in the plant area, AI planner controls the HVAC, which could manage the foul smell/high temperature/bad air quality in the plant and make the hotel premises comfortable for the employees and the customers. The plant should also give safety alerts in case of any hazardous situations. It also plans the emergency safety systems such as fire-alarm mitigation systems to counter the fatal situation and it actuates safety alert alarm buzzer and also displays a alert on the LCD display in the plant. It also makes decisions accordingly and fetches them to the plan for execution, that would

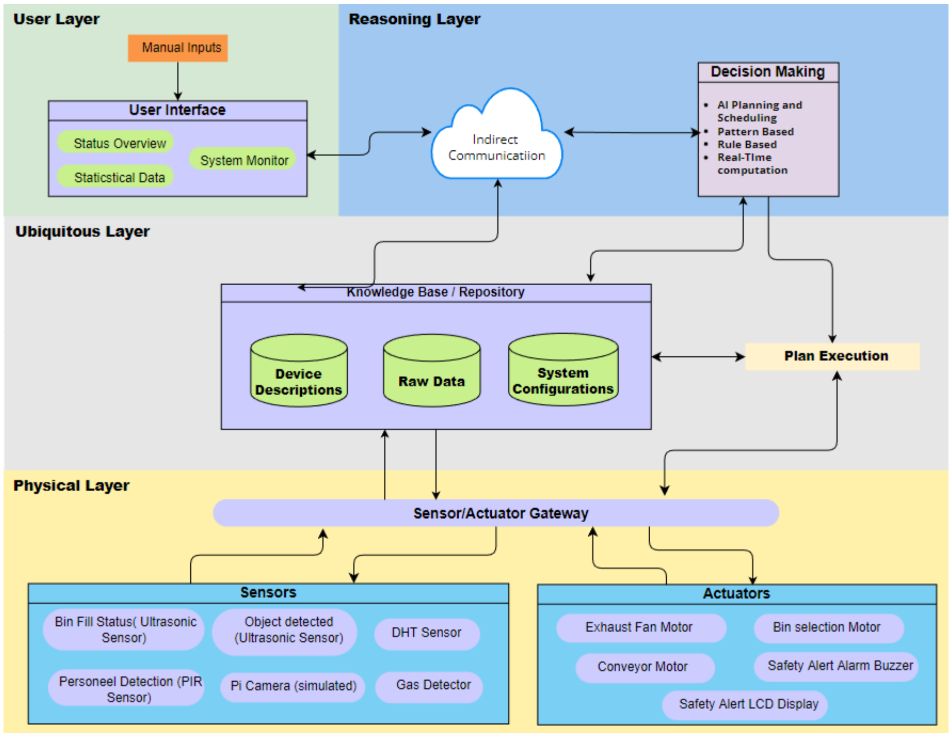

Fig. 1. Smart Garbage In-house Segregation Plant, GARB-I

actuate the responsible task to be done such as exhaust fan control and human presence detection based lighting control. By doing so, we can achieve the goal of power-saving and energy management.

In the user layer, the user can overview the current status of the Garbage segregating process and take further steps. It can also look at the past statistical data and do the planning/scheduling based on the peak segregation period (During Check-out and Check-in hours, Hose-keeping hours).

4 System Implementation

4.1 Hardware

Fig. 2 is the hardware assembly of the the GARBI.

4.1.1 Raspberry pi B3+:

The newest model in the Raspberry Pi 3 line, the Raspberry Pi 3 Model B+(RPi), features a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and

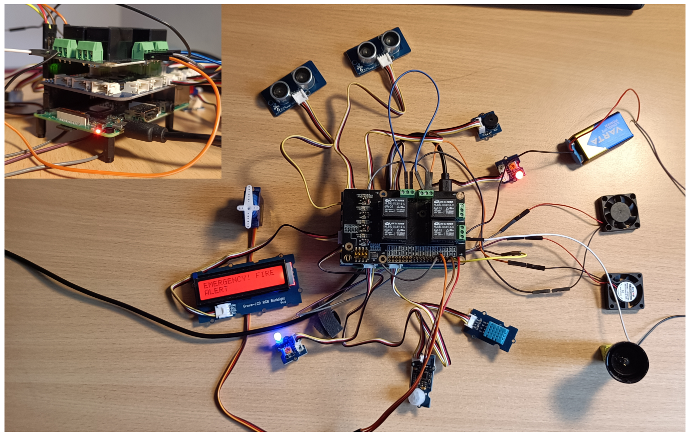

Fig. 2. Hardware assembly of Smart Garbage In-house Segregation Plant, GARB-I

5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE functionality using a separate PoE HAT. A group of single-board computers known as the RPi can be used for a variety of tasks, including as a general-purpose computer, a robot, or even to connect Internet of Things devices. Any display could be plugged into an RPi. It's a potent little tool that may be used to instruct students in Python or bash coding. It can perform almost any task that users can think of for a computer. Furthermore, RPi is also known to connect to the outside world through sensors and actuators.

Main Advantages of using Raspberry Pi for IoT. It is an exceptionally useful device due to its:

1. The list of languages using which it can connect to the internet is remarkable. Some widely used programming languages like Python, Java, and JavaScript are all compatible with the Raspberry Pi. It can also work as standalone server for smart home automation.

2. The RPi's GPIO pins allow sensors to be directly attached, making it highly efficient in system deployment.

3. It is a cheap and full-featured device with a Linux OS.

4.1.2 Grovepi+:

Grove is a modular, plug-n-play technology platform developed by SeeedStudio. Much like Lego, it takes a building block approach and applies it to assembling electronics. The Grove system consists of a base shield and various modules and sensors with standardized connectors. The base shield allows you to easily connect any microprocessor input or output from the Grove modules, each with their own function. Grove modules and sensors range from simple buttons to complex heart rate sensors. Each one comes with clear documentation and demo code to help you get started quickly.

The GrovePi+ Starter Kit gets you up and running with the Grovepi+ quickly. The starter kit bundles the most popular sensors for education and hobbyists, and lets you start playing and prototyping hardware with Raspberry Pi.

The GrovePi+ Starter Kit package includes:
1. GrovePi+ Board
2. 12 different Grove sensors and modules
3. Grove cables for connecting the sensors to the Grovepi+ board.

4.1.3 Grovepi Relay Shield:

The 4 Channel Relay Breakout is an easy way to use your Arduino, Raspberry Pi, or other micro-controller to switch high voltages and high current loads. The board is both 3.3V and 5V logic compatible and uses 4 digital outputs to control 4 individual relays. Each relay has the common, normally open, and normally closed pin broken out to a convenient 5.0mm pitch screw terminal. Grovepi relay shield is used to control the start/stop operation of conveyor motor and the exhaust fans.

4.1.4 GrovePi Sensors and Actuators:

The sensors in the system includes:

a. Bin Fill Status Sensor: Ultrasonic sensor is required to measure the percentage of load/the occupied area in the each garbage sorted bin i.e., Biodegradable and Non-Biodegradable waste.

b. Waste(object) Detection Sensor: An Ultrasonic Sensor is used to detect the item dropping onto the conveyor belt, which is vented out from any hotel room/area via a common duct.

c. Temperature Humidity(DHT) Sensor: Temperature Humidity sensor is used to measure the temperature and moisture content in the segregation area and to monitor the temperature and humidity levels in the plant such that the garbage does not make foul smell which would annoy any customer and does not cause uncomfortable place for the hotel staff, waste management staff to work.

d. Personnel Detection PIR sensor: PIR motion detector is used to sense the human motion and operate the lighting in the GARBI plant.

e. Air Quality Detector: This sensor used to monitor the air quality levels in the plant and alert the hotel staff and manager in case of poor AQI values.

f. Pi Camera: A simulated algorithm is used to detect and sort the garbage into Bio waste and Non-bio waste bin .

The actuators in the system include:

a. Conveyor Motor: Conveyor motor operates the conveyor belt, on to which the garbage from the common venting duct is collected. The type of garbage is then recognised and placed on the respective segregation bins.

b. Bin Selection Motor (Servo): A Servo motor is used to perform the bin selection i.e. biodegradable or non-biodegradable based on an algorithm. Once the type of garbage is recognised, the motor rotates and the respective type of segregation bin faces the conveyor belt to collect the garbage.

c. Exhaust Fans: Two Exhaust fans are deployed in the plant. One of which actuates depending on temperature and the other based on air quality values. Whenever temperature and air quality value in the in-house segregation plant GARB-I increases beyond a specified threshold temperature and air quality set-point, the respective exhaust motor is actuated and vents out the air from the plant and intakes fresh air from the outside environment.

d. Safety Alert Alarm buzzer / LED: If there is any kind of emergency related to fire or bad air quality in the plant,then a buzzer alarm and red light are actuated to alert the employees in the garbage segregation plant.

e. Safety Alert Warning LCD Display: Whenever there is any kind of emergency fire or bad air quality situation then a fire alert(in red) or bad air quality(in yellow) alert are displayed to alert the employees in the garbage segregation plant.

4.2 Software

4.2.1 Git Repository

All the codes relating to our project can be found in our Git repository at the following link: https://github.com/AKJ2497/Garbi.git

4.2.2 Setting up OS on Raspberry pi B3+

The first step for setting up the Raspberry Pi is the flashing of the correct OS. A standard installation of "wheezy" or Raspbian will not work with GrovePi; there are some specific modifications that must be made to successfully operate the GrovePi. Out of the three options given in the GrovePi manual to install the OS, we implemented the second option, i.e. to download custom "Raspbian for Robots" Image.The GrovePi+ module and sensors are compatible with this version of Raspbian. Also, Raspbian for Robots has all the needed libraries provided by Dexter industries for the smooth functioning of all the GrovePi+ board, relay board, GrovePi sensors and actuators. The Os was flashed on to the SD card using Win32 disk imager software for Windows.

For a headless setup, we created a file named ssh without any extension on the boot partition of the SD card. On boot, the Pi checks for the ssh file, enables SSH and then deletes the file. This enables us to access and control the Raspberry Pi using third party Linux software like Putty.

Then we setup the IP address of the Pi using the cmdline.txt file in the root of the boot partition of the OS. Then after inserting the SD card on the Pi, we powered on the Pi.

Now it was time for the Pi network configurations. We used the Linux command 'sudo raspi-config' to access the Localization options to change the Wi-Fi country and timezone. Then we changed the wpa-supplicant.conf file from the boot section of the OS to add Wifi network information of our room Wifi and Eduroam. Then after rebooting the Pi, a successful network connection was checked by pinging google.com.

Then, we encrypted the network password and updated the Pi to it's latest firmware using the commands 'sudo apt update sudo apt upgrade'. In addition, we had to install paho MQTT client for python using 'sudo apt install paho-mqtt'. The latest version of GrovePi was installed using 'sudocurl -kLdexterindustries.com/update-grovepi— bash' command.

4.2.3 Initial testing of GrovePi sensors, actuators and GrovePi Relay board

After setting up the Pi, we had to test the provided GrovePi sensors, LCD display, relay module and our purchased actuators i.e. DC motor (for conveyor), servo motor (for bin selection). The dummy codes for testing the GrovePi sensors and relay module were already provided in the GrovePi manual and all the GrovePi components were found to be working properly. Then, we wrote basic python codes for testing the operation of the motors and they were also working flawlessly. We then complied all the sensor and actuator codes into a single file to test the simultaneous working of our components.

4.2.4 Indirect Communication using MQTT

Our system is designed to make use of 2+ machines, the raspberry pi, a remote-client dashboard and another machine for remote access to pi. The Raspberry Pi is the brains of the project and performs majority of the actions like reading and publishing the sensor data. Indirect communication between the devices is achieved using the MQTT protocol. MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. It is also easy to code and implement. Raspberry Pi is acting as the broker publishing the sensor data. The Visualisation dashboard on the client end, subscribes to this code from a laptop/PC to display useful real-time information of the working of the plant to the facility manager. The code for the Actuators is also publishing its data. The Sensor and Actuator topics are being subscribed by the Visualization code on the client side.

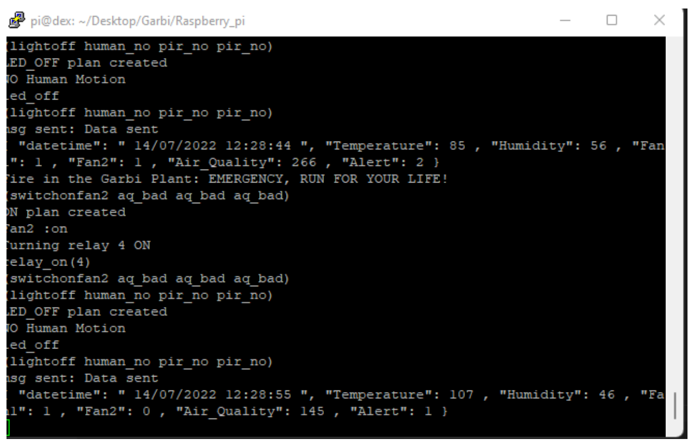

Fig. 3. Sensor Topic Publisher code output publishing sensor data and AI planning values

The topics "Sensor_data" and "Actuator" publisher data is shown in Fig. 3, 4. The subscribed topics by the subscriber is shown in Fig. 5, 6.

4.2.5 AI planning using PDDL

AI Planning is a branch of artificial intelligence that studies the use of autonomous methods to the resolution of planning and scheduling issues. A planning problem is one in which, via the application of a series of activities, we aim to change an initial beginning state into a desired goal state.

AI planning problem can be defined using the Planning Domain Definition Language (PDDL) family of languages. The language used to explain planning has

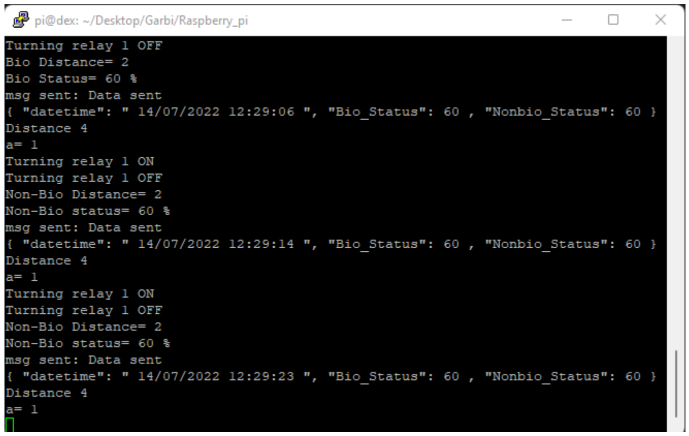

Fig. 4. Actuator Topic Publisher code output

grown along with it, and as a result, there are now numerous versions of PDDL accessible with various levels of expressivity. First, we made separate domain files for our three use cases. But, this was quite cumbersome and unnecessary. So, we created a single domain file containing the actions for all use cases.

Linux-based compilation is taken into consideration while creating almost all AI planners. Other planners have been successfully transferred to Windows and Mac, and in some cases, the source code's structure makes the porting process simple (such as the Java based planner JavaFF). Planners frequently need to be compiled because they aren't frequently given in binary form. There are no methods for assembling plans in this guide. Any compilation instructions supplied are for informational purposes only; for assistance with compilation, get in touch with the planner's designer. Most planners follow one of two syntaxes for inputting a domain and problem file.

$$./ < planner >< domain >< problem >$$

$$./ < planner > -o < domain > -f < problem >$$

The difference is mostly caused by the distinction between "operators" and "facts" in the domain and problem files, which led to the employment of the "o" and "f" command line options. When used as research tools, planners frequently offer feedback on the problem decomposition and search process rather than simply passing out a plan onto the output.

Most planners are made to be executed from the command line after you have a compiled binary for it.

The PDDL domain file named "masterdomain.pddl" contains the code for planning the action of the exhaust fans and lighting in the plant. One exhaust fan is

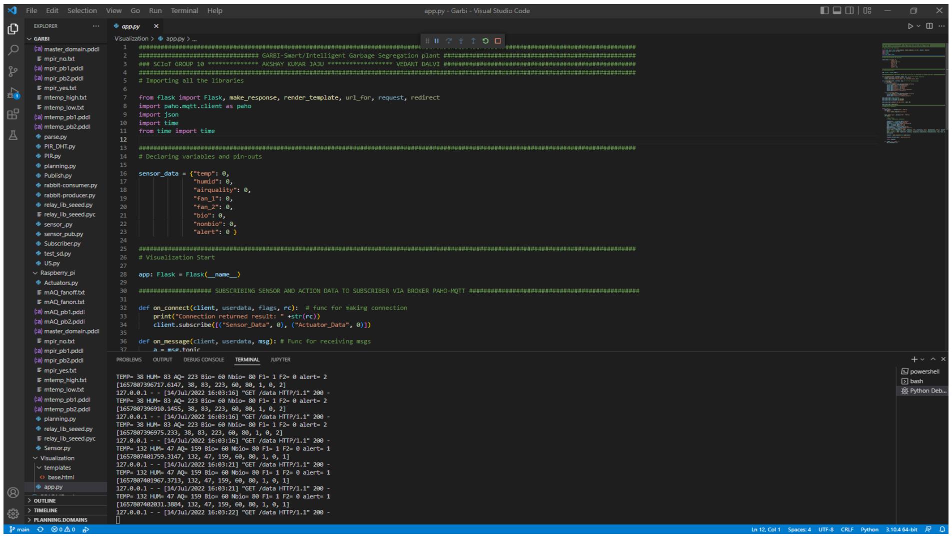

Fig. 5. Topic Subscriber code output

controlled by the temperature values and the other fan by the air quality values. For temperature in ambient conditions, the fan remains off and once the temperature crosses a set-point of 75 degrees, the exhaust fan turns on. On the other hand, the exhaust fan remains off when AQI is below 100 which are moderate values as per the AQI index. Once AQI crosses 100, which is in the unhealthy range, the second exhaust fan turns on to mitigate this issue. The PIR sensor detects human presence and switches on the lights in the plant. After a brief period of human inactivity in the plant, the lights are turned off which leads to power savings.

Goal 1: Switch OFF Fan 1
This goal is to perform the action of switching off fan1 when temperature is low or in ambient range.
PDDL Output: (switchofffan1 temp_low t_low t_low)

Goal 2: Switch ON Fan 1
This goal is to perform the action of switching on fan1 when temperature is above ambient conditions.
PDDL Output: (switchonfan1 temp_high t_high t_high)

Goal 3: Switch ON Fan2
This goal is to perform the action of switching on fan2 when air quality values are in bad AQI value range.
PDDL Output: (switchonfan2 aq_bad aq_bad aq_bad)

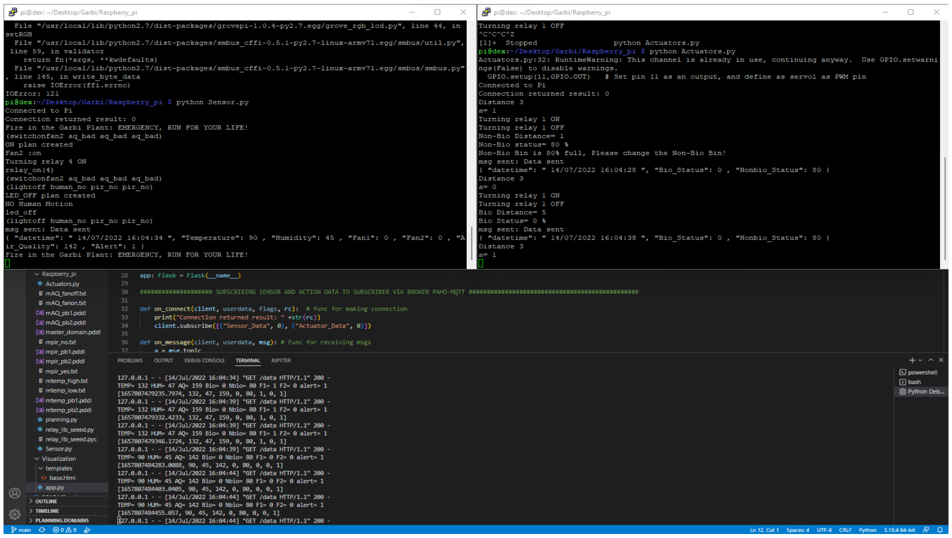

Fig. 6. Combined Publisher and Subscriber code output

Goal 4: Switch OFF Fan2
This goal is to perform the action of switching off fan2 when air quality values are in good AQI value range.
PDDL Output: (switchofffan2 aq_good aq_good aq_good)

Goal 5: Switch ON LED
This goal is to perform the action of switching on LED when PIR sensor detects a human presence.
PDDL Output: (lighton human_yes pir_yes pir_yes)

Goal 6: Switch OFF LED
This goal is to perform the action of switching off LED when PIR sensor does not detect a human presence for a period of time.
PDDL Output: (lightoff human_no pir_no pir_no)

The other component of the adage about planning is an AI planner. An AI Planner enables us to make an attempt at solving an AI Planning problem, whereas PDDL only allows us to define one. An AI planner uses the PDDL it has read in to break the problem down and solve it. Different AI planners support various syntax's to varying degrees since AI planners have developed with the languages they employ. In this project, we have utilized the online solver. The AI planner function in the code is called whenever the corresponding precondition for temperature/air quality/human presence is true and then it reads the respective PDDL code and performs the necessary actuation.

4.2.6 Visualization

The visualization dashboard has been created using Python Flask and HTML using Bootstrap. The template from the Bootstrap template was modified per the scope of our project. On the dashboard, we get the required real-time environmental data of the plant like temperature, humidity levels and air quality value along with the bio and non-bio bin fill status, exhaust fan operating state and safety alerts. User can also access the graphical data of the temperature, humidity and air-quality on scrolling down. This information enables remote monitoring of the plant. Thus, Garbi requires human intervention only during changing of the waste bins. Fig. 7 and Fig. 8 shows the dashboard with all the real-time parameter values and graphs.

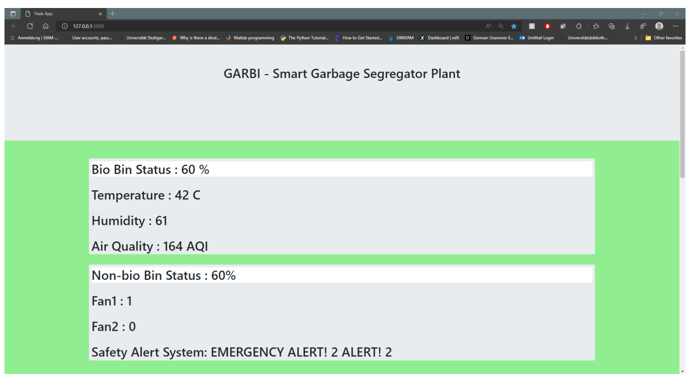

Fig. 7. Dashboard data

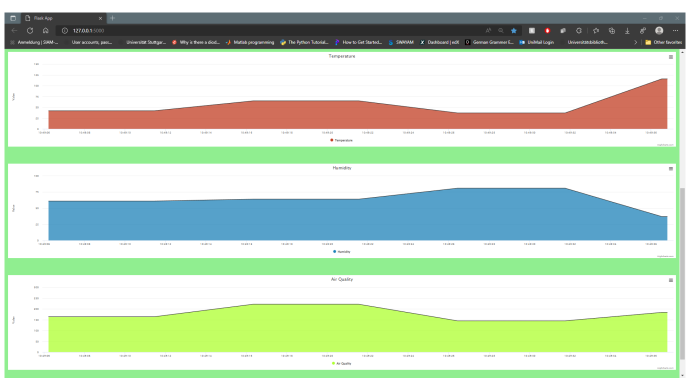

Fig. 8. Dashboard graphs

The exhaust fan status is represented in 2 binary states '1' and '0' showing 'On' and 'Off' states respectively. When each of the garbage segregation bins fill up to 80 percent of their capacity, it gets notified instantly on the dashboard with a red background as shown in Fig. 9, 10, 11.

- If the Safety Alert is 0, it indicated the normal working condition.
- If the Safety Alert is 1, it indicates an accidental fire.
- If the Safety Alert is 2, it indicates bad air quality in the plant.

4.3 Project Operation

The hardware and software components are combined in the working of the project. All the GrovePi sensors are connected GrovePi+. The Conveyor motor and the two exhaust fans are controlled by the GrovePi relay board and Bin Selection motor (servo) is controlled using the PWM GPIO pins of the Raspberry Pi.

The GrovePi DHT sensor continuously reads the temperature and humidity values in the plant. When the temperature vales are in ambient conditions i.e. below 25 degrees Celsius, the AI planner keeps the exhaust fan OFF. When the temperature vales are in between 25 and 70 degrees Celsius, the AI planner switches the exhaust fan ON. When temperature values are greater than 70 degree Celsius, the emergency fire alert is displayed on the LCD display and dashboard.

The simulated air quality values are continuously checked and when they are in the normal to moderate range i.e. between 0 and 100 as per AQ Index, the AI planner keeps the exhaust fan in the OFF state. When the AQ values are in the range 100 and 200, the AI planner switches ON the exhaust. If the AQ values exceed 200, a bad air quality alert is displayed on the LCD display and dashboard. To simulate the emergencies, we are simulating the temperature and air quality values.

Whenever garbage drops on the conveyor, it is detected by the Waste detection sensor and a simulated Pi camera algorithm detects the waste type (i.e. bio or non-bio) which executes the Bin Selection accordingly. The conveyor is then turned on till the waste falls in the respective segregation bin and then Bin Fill status sensor measures the filled capacity of the bin and sends the data to the dashboard.

- If the bin capacity is less than 80 percent, it is displayed on the respective section on dashboard.
- If the bin capacity exceeds 80 percent, it is alerted on the dashboard with a red highlight on the respective bin status i.e. bio bin status and non-bio bin status.

The MQTT topic 'Sensor-Data' is created after running the Sensor.py code. Within this code, the temperature, humidity values from DHT sensor, the human motion detection values from the PIR sensor and the simulated air quality values are read and published in json format.

The topic 'Actuator-Data' is created after running the Actuators.py code. This code reads the Waste(object) Detection Sensor values and starts the conveyor belt. The Bin Selection Motor (Servo) actuates and the garbage falls into the respective bin. Then, the Bin Fill Status Sensor reads the bin capacity and publishes the data in the json format.

The Visualization dashboard subscribes to the two topics of 'Sensor-Data' and 'Actuator-Data' and displays all the vital information.

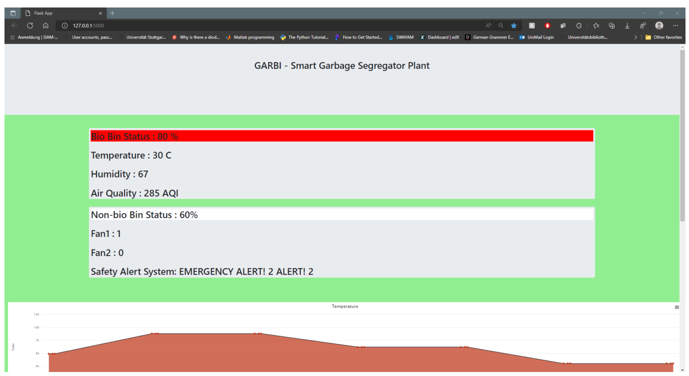

Fig. 9. Bio bin fill status alert

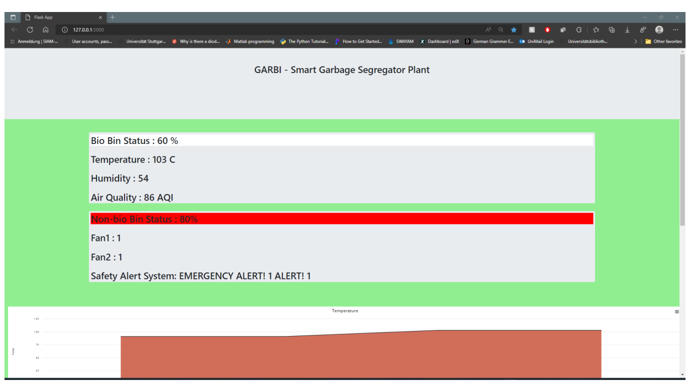

Fig. 10. Non-Bio bin fill status alert

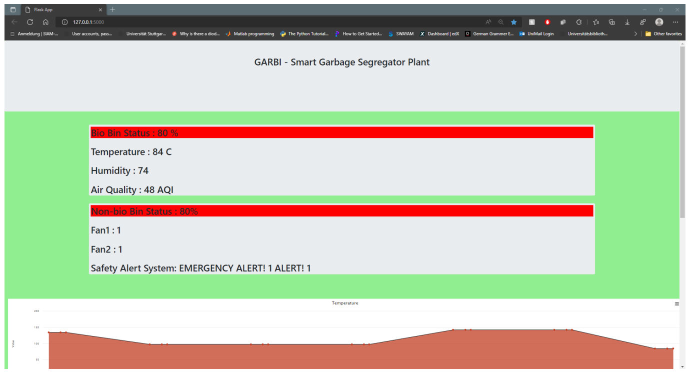

Fig. 11. Both bins status alert

5 Discussion and Conclusions

5.1 Issues faced:

5.1.1 Raspberry pi

This was our first interaction with the Raspberry Pi board and naturally, we faced plenty of issues. We also had limited initial experience with Linux. Initially, we found it difficult to find the correct raspbian image which is compatible with the GrovePi+ board and sensors. Then, we struggled with the headless pi setup, enabling SSH and initial network configurations of the Pi. We learned about static and dynamic IP's in the process and deployed a static IP for the Pi. Later, we encountered the name resolution error which occurs due to change in DNS name-server. This issue was very frustrating and it was a bit more difficult to debug as it took away a huge chunk of our time. While successfully solving the errors by searching for the solutions and practising trial and error method, we got to learn more about how the Raspberry Pi and Linux functions.

5.1.2 MQTT

Initially, we implemented RabbitMQ server in our MQTT Publisher code. However, we were not able to accurately extract the json data on the Subscriber side. This was due to a shortcoming in the default RabbitMQ code. Hence, we decided to use Paho python client for our MQTT code. We expected the MQTT code to get quickly implemented. But, we faced a few difficulties in getting the Publisher-Subscriber code to run successfully.

5.1.3 PDDL

The PDDL coding was not as challenging as we expected but we struggled with writing the code for calling the AI planner as per our use cases. We couldn't come up with a way to trigger the AI planner and the corresponding PDDL plan to implement the actuations. Eventually, after a lot of research, we came across the concept of parsing files and were able to write a code for it and integrate it with the rest of the code. We created a function for AI planner and invoked it when certain conditions were true. This helped us to solve this problem which took a lot of our time.

5.1.4 Visualisation

We wanted to deploy pop-up alerts on the dashboard whenever the fire alert or poor air quality emergency condition was aroused. We found it difficult to implement this in the code as we had no prior knowledge of HTML. After a lot of trial and error, we were able to successfully deploy this in our code.

5.1.5 Model

After all the coding, testing and debugging, it was finally time to make our physical Garbi model. We placed all the sensors and actuators properly in the plant. It was always going to be difficult to make a working conveyor belt model and it proved to be exactly as expected. Initially, we couldn't find the necessary components to make the conveyor belt mechanism. We used a bit of creativity to come up with a functioning small scale conveyor model.

Fig. 12. GARBI Model

Fig. 13. GARBI alert state

5.2 Future Scope:

The scope of this project can be extended to any type of establishment as waste is generated everywhere and recycling is mandatory in many countries. Further, the garbage segregation could be centralised in a way that there is a centralised duct system where an interconnected pipeline system would have an opening in each room, where the user can just drop the waste and the segregation would be totally autonomous.

5.3 Conclusion:

Today, with the increasing emphasis on environmental conversation and Smart Cities as a way of achieving sustainability, projects like Garbi are going to be very important in the distant future. Many research organizations and companies are working in this domain. With this project, we have tried to solve a problem which is of prime importance for recycling and waste disposal. Government norms are already in place for Hotels to meet recycling standards and this project is one of the efficient solutions. In this project, we have accomplished the goals of Safety, Energy Efficiency/Power Savings and a Smart comfortable work environment.

1. Safety

Safety of the plant was one of the important aspects to fulfill as the plant would be located in the basement or in the underground of the actual Hotel. So the HVAC, air quality should be reliably maintained and was successfully implemented by using Exhaust fans and software actuators. Emergencies arising from unforeseen circumstances have been accounted for in the plant sending instant alerts to the Facility manager/Hotel manager or security room in the hotel. fig. 13

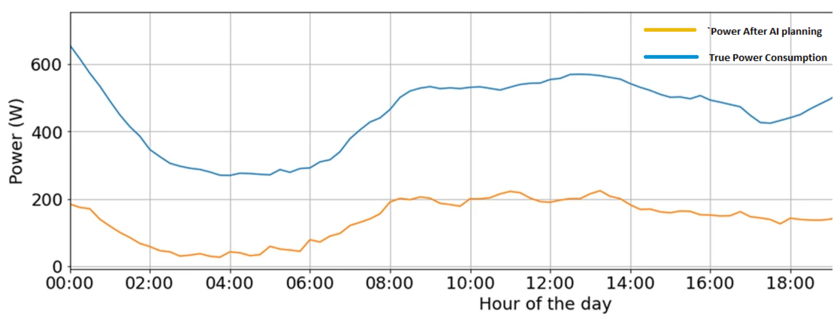

Fig. 14. power consumption comparison with and without AI planner

2. Energy Efficiency/Power Savings

Today, Smart Buildings have a prime focus on power savings. With increasing demand on power grids, efficient consumption is the way forward and many establishments are focusing their efforts on executing this aspect. Garbi's smart lighting ensures power savings and the conveyor's operation is also controlled by the detection of garbage on it, which avoids continuous operation of the conveyor and bin selection machine. These aspects save a good amount of energy which contributes to a significant amount in the long run.fig. 14

3. Smart comfortable work environment

Human interaction in the plant is limited by choice as we wanted to make it as autonomous as possible. Despite this, the continuous monitoring of the bins capacity and HVAC values allows for time-efficient operation of the working staff in a comfortable environment. fig. 15

In addition, we had many learning outcomes from this project. This project helped us to gain insights into a plethora of new topics which would surely help us in the future.

Fig. 15. GARBI normal state

References

1. Tanya Gupta, Rakshit Joshi, Devarshi Mukhopadhyay, Kartik Sachdeva, Nikita Jain, Deepali Virmani, Laura Garcia-Hernandez (2021) A deep learning approach based hardware solution to categorise garbage in environment, Springer
2. Aziz F, Arof H, Mokhtar N, Mubin M, Talip MSA (2015) Rotationinvariant bin detection and solid waste level classification. ElsevierMeas 65:19–28
3. Bhatia A (2019) Fighting India's garbage crisis: what are the chal-lenges India faces in managing its waste. New Delhi TelevisionLimited. Retrieved from https://swachhindia.ndtv.com/
4. Chahine K, Ghazal B (2021) Automatic sorting of solid wastesusing sensor fusion. Int J Eng Technol 9(6):4408–4414