Garbage Dumping Reminder System

Yuedi Mi
Department of Electrical & System Engineering
Washington University, St. Louis, MO, USA
m.yuedi@wustl.edu

Hongyi Xu
Department of Computer Science and Engineering
Washington University, St. Louis, MO, USA
hongyi.x@wustl.edu

Fake Zhao
Department of Computer Science and Engineering
Washington University, St. Louis, MO, USA
fake@wustl.edu

ABSTRACT

There are many things that need to be monitored in real time. However, people don't realize the benefits that monitoring them brings to their lives. This technical report describes how we have selected a small object in our lives, trash cans, and monitored them to bring convenience to people. Garbage Dumping Reminder System provides users the ability to monitor trash cans in real time. We will remind the user to the status of the trash can and remind the user to empty it. This paper also compares the running time and running efficiency with different techniques and algorithms to reduce the latency.

1 Introduction

Environmental problems are raised by modern cities for waste collection and disposal.[1]

Therefore, smart waste management systems became essential for cities that aim to reduce cost and manage resources and time.[2] Currently, the trend is shifting towards smart devices and internet of things (IoT) solutions to overcome common problems such as waste management issues. [3]

There is a widespread belief in the real-time system thought that real-time can only be applied to some large or emergency braking equipment. Adding a real-time system to a simple device is often considered by the public to be an expensive and extremely uneconomical thing. However, we can also apply some simple and affordable real-time systems to some of life's smaller objects and achieve good results. This project shows how to implement real-time monitoring on daily garbage cans for better efficient management and cleaning at a small cost.

2 Goals and Requirements

Imaging that you went to a tourist attraction, there are full of people, and they have generated a lot of garbage. At that time, you want to throw away your garbage, but then you find that all the garbage cans within this area are full, and because the distance between two garbage cans are very far, it can also take the cleaners a lot of time to dump all the garbage cans.

To avoid this situation and help the cleaners be able to dump the garbage can in time, we decided to design a reminder system to inform the users when the garbage cans need to be dumped.

3 Design

3.1 System design

This project uses Raspberry Pi and various sensors to collect environmental data from the trash cans. Construct a model to analyze these data, then sent the result into the database of the remote server, The server then analyzes these data and visualizes it.

3.2 Judgment criteria

Our team set 3 judgment criteria of whether a trash can needs to be dumped or not, which is ‘Smell’, ‘Height’, and ‘Time’.

‘Smell’: Food spoilage produces gases such as hydrogen sulfide, ammonia, methane, and carbon dioxide, those gases may cause harm to human-beings, we plan to use gas sensors to detect the concentration of the mentioned gases and send a reminder to users if the concentration of these gases has reached the threshold.

‘Height’: We want to keep the state that people can always throw garbage into the trash can, and we plan to place an infrared sensor at the appropriate height of the trash can, so when the amount of garbage reaches the critical value, users will receive a remind.

‘Time’: We don't advocate not dumping garbage for a long time, so if a trash can hasn't been dumped for a long time, let’s say over 7 days, users will be reminded that this trash can should be dumped.

4 Implementation

4.1 Raspberry pi

We install a raspberry pi on the trash can to monitor the environment. The raspberry pi uses the GPIO pin to connect the sensors or other IoT modules. the row of GPIO (general-purpose input/output) pins are shown below:

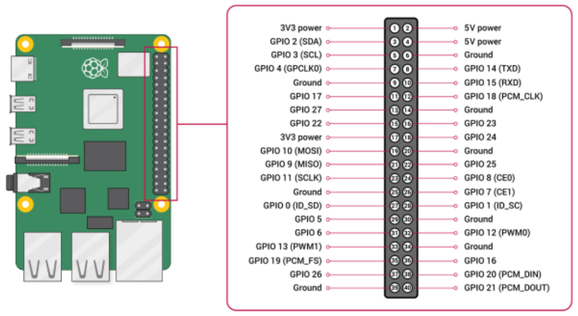


Figure 1. the row of GPIO (input/output) pins

4.2 Gas Sensor (BME680)

The BME680 is the first gas sensor that integrates high-linearity and high-accuracy gas, pressure, humidity, and temperature sensors. It is especially developed for mobile applications and wearables where size and low power consumption are critical requirements. The BME680 guarantees - depending on the specific operating mode - optimized consumption, long-term stability, and high EMC robustness. To measure air quality for personal wellbeing the gas sensor within the BME680 can detect a broad range of gases such as volatile organic compounds (VOC). [4]

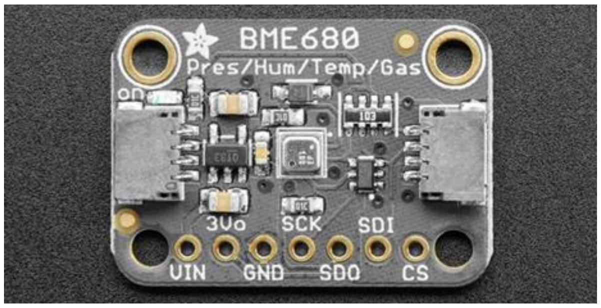


Figure 2. BME680 gas sensor

We use the BME680 sensor to detect if there is any rotting food in trash can. We will use it to detect humidity, air pressure and other environmental elements. Then we will train a model to compute the indoor air quality (IAQ) to evaluate the environment condition of trash can. This BME680 has power pins (Vin, 3Vo, GND), SPI Logic pins (SCK, SDO, SDI, CS), I2C Logic pins (SCK, SDI). We connected it to the specific Raspberry pi's pins separately.

4.3 Infrared sensor (HC-SR04)

At its core, the HC-SR04 Ultrasonic distance sensor consists of two ultrasonic transducers. The one acts as a transmitter which converts electrical signal into 40K Hz ultrasonic sound pulses. The receiver listens for the transmitted pulses. If it receives them, it produces an output pulse whose width can be used to determine the distance the pulse travelled. As simple as pie!

The sensor is small, easy to use in any robotics project and offers excellent non-contact range detection between 2 cm to 400 cm (that's about an inch to 13 feet) with an accuracy of 3mm. Since it operates on 5 volts, it can be hooked directly to an Arduino or any other 5V logic microcontrollers. [5]

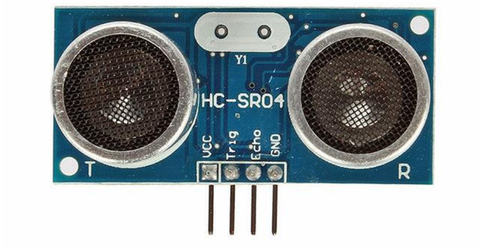


Figure 3. HC-SR04 infrared sensor

We use the HC-SR04 to detect the trash height in the trash cans. The HR-SR04 has four pins: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground). We connected these pins to the Raspberry pi.

4.4 Resistor

Figure 4. Resistors

We also need some resistors to protect our circuit. BME680 and HC-SR04 has different input and output volts limit with the Raspberry pi. So, we must make a protection on them. The schematic diagram of the circuit voltage divider is as follows**:**

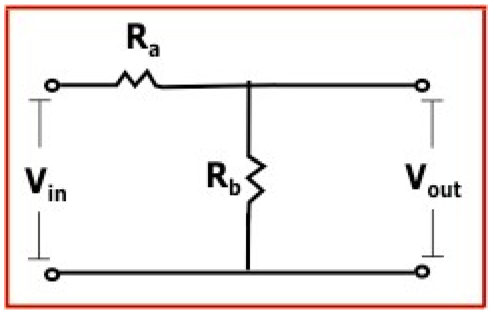


Figure 5. The schematic diagram of the circuit voltage divider

4.5 Data transmission and display

We chose the AWS server to run the database and Apache server. Raspberry pi will write the data to the database in real time. And we create a backend PHP component to get this data and connect to the front-end website to display the status of trash cans.

5 Experiments

We use a cardboard box and drop debris to simulate the appearance and condition of a trash can.

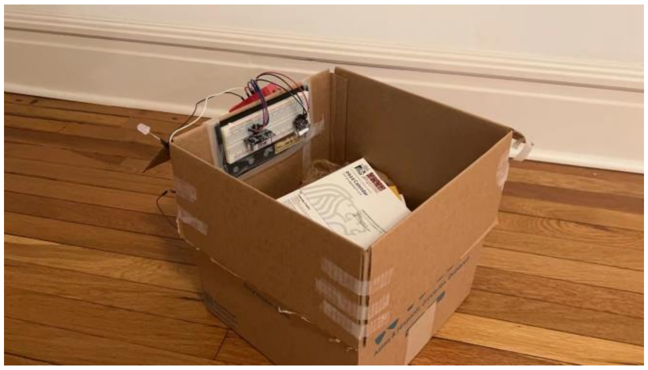


Figure 6. Simulation of the appearance of garbage cans

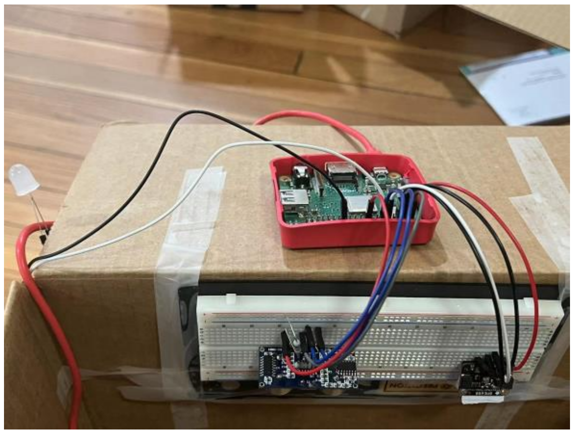


Figure 7. Overall system connection

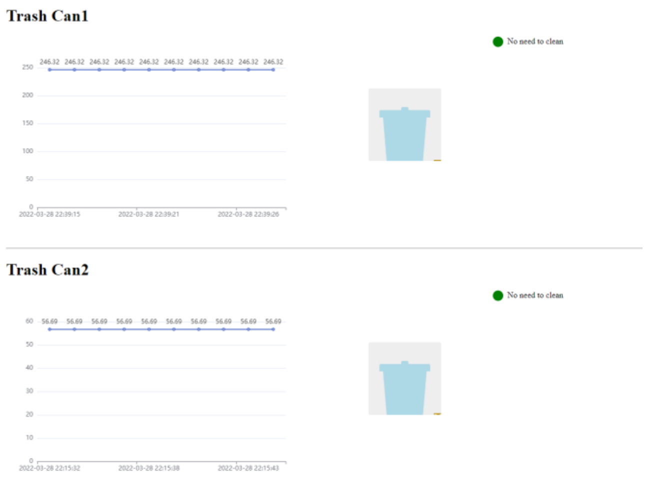


Figure 8. Data visualization website

Our website can update all the trash cans' status, they are listed by their id. We also created a chart to show the trend of all the trash cans' current IAQ. There is a trash bin's picture to show the current garbage height. If the IAQ or the height doesn't meet our criteria, then the notification light will turn from green to red and show the corresponding reminder messages.

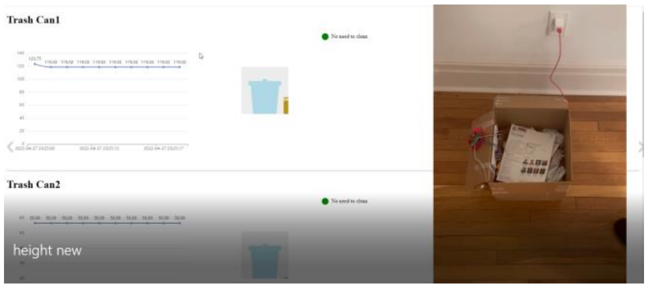


Figure 9. Website display in normal situation

In our experiments, when the trash can isn't full, the website will show the exact trash height and IAQ value.

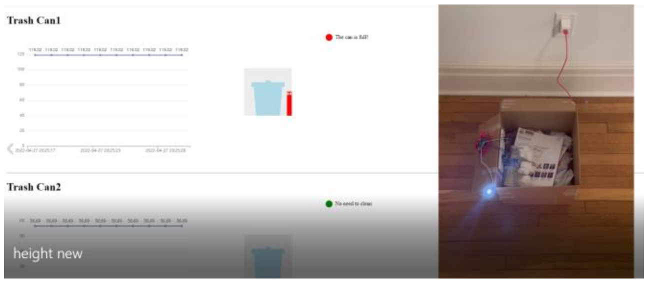


Figure 10. Website display when the trash can is full

If we throw some items into the trash can and make this can full, then the LED light of this garbage can turn on and the website's trash height status turns into red. The notification light turns into red as well and the message shows "The can is full".

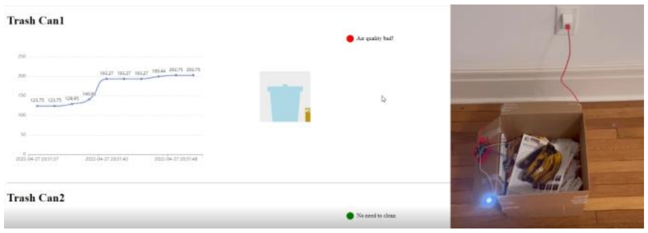


Figure 11. Website display when there is rotten food in trashes

If we let the height of trash normal and throw a Rotting food (banana in this experiment) into the trash can, the chart of this trash can on the website increases continuously and quickly. The notification light turns into red as well and the message shows "Air quality bad".

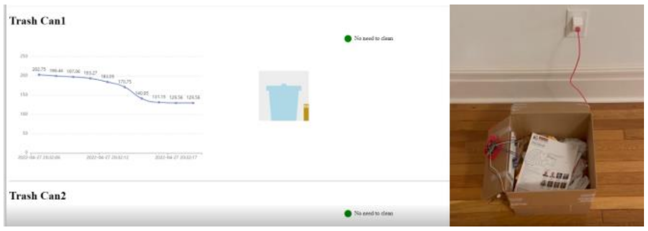


Figure 12. Website display after rotten food is cleaned

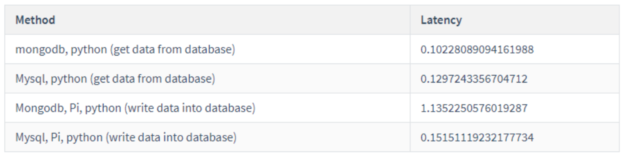
Then if we remove the rotting food from the trash can. The IAQ value will gradually return to the original level. The LED light of this garbage turns off. Notification light and message turn to the original status.

6 Related work

This chapter calculates and compare the latency time performance of different database techniques. This chapter also shows how to create a model to predict IAQ and compare different models' performances.

6.1 Different database techniques

Table 1. Different database techniques run time



Method	Latency
mongodb, python (get data from database)	0.10228089094161988
Mysql, python (get data from database)	0.1297243356704712
Mongodb, Pi, python (write data into database)	1.1352250576019287
Mysql, Pi, python (write data into database)	0.15151119232177734

We compute the MySQL and Mongo dB's latency time. Their latency times are shown in this table, **it** shows that MySQL always shows better performance than Mongo dB under the same situation. So **finally,** we decided to use MySQL as our database.

6.2 Introduction to the IAQ value

As we mentioned before, air quality within the garbage can is also one of the criteria we wanted to use to determine whether a trash can needs to be dumped, and we choose the IAQ value (Indoor Air Quality) to be our measurement. The IAQ value is the air quality within and around buildings and structures, especially as it relates to the health and comfort of building occupants. The following table shows the values of IAQ and its corresponding levels of health concern.

Table 2. IAQ value range and the corresponding levels of health concern

Indoor Air Quality Values	Levels of Health Concern
0-50	Good
51-100	Moderate
101-150	Unhealthy for Sensitive Groups
151-200	Unhealthy
201-300	Very Unhealthy
301-500	Hazardous

6.3 Data preprocessing

Though we have already decided using the IAQ value, one of the challenges we faced is that we didn't know the exact formula to calculate the IAQ value, and we only know that the formula contains 4 factors, which are 'pressure', 'gas concentration', 'temperature', and 'humidity'. After searched on the internet, we found several datasets that contains those 4 factors and their corresponding IAQ value, then we download those datasets, and began further exploring the connection between these 4 factors and the IAQ value. The total volume of data our downloaded dataset has is around 122,499, after we cleaned the dataset (removing outliers, filling missing values), there's a total of 118875 data left, the table below is part of the final dataset.

Table 3. Part of the Final Dataset

Pressure	Gas concentration	Temperature	Humidity	IAQ value
97150.438	690577.563	31.015	15.621	36.388
97145.359	852564.063	30.645	15.186	83.604
97167.508	1071716.375	30.365	15.09	239.756
97183.156	1140651.875	30.505	14.705	149.547

Then, we did data normalization to the rest of the dataset, which is to make each feature values to has mean of 0 and standard deviation of 1, we hope in this way it could help our ML models have a better performance.

6.4 Using ML to calculate the IAQ value

After processing the dataset, we split it into training set and testing set at the proportion of 8:2, then we applied a total number of 11 different ML models, we turned the parameters of all the 10 models to make sure that each model is at their best performance level, the final score (here we use R square) of each model is listed in the following table.

Table 4. Part of the Final Dataset

Model	Training Set Score	Testing Set Score
Decision Tree Regressor	0.999999958	0.997265515
Extra Tree Regressor	0.999999945	0.9965937
Bagging Regressor	0.999687705	0.998507088
Random Forest Regressor	0.999790253	0.998703317
Gradient Boosting Regressor	0.978557943	0.977686245
AdaBoost Regressor	0.817860294	0.811680796
Linear Regression	0.70157146	0.695984314
SVR	0.670375303	0.654638055
MLP Regressor	0.495161199	0.494499264

According to the above table, we could say that the top 4 models are all perform very well, their R square are all above 0.977 which is extremely high and proves that these 5 models are all very powerful in predicting the IAQ value. But we also need to take latency into account, one thing we should mention is that since our gas sensor needs some time to be stabilized after it is turned on, the running time of each model when the gas sensor is just turned on will be different from the time the gas sensor is stabilized, we measured those models' running time both in the 2 situations, and the final result are showed in Table 4.

Table 5. Latency of each model

Model	First Using Time	Stable Time
MLP Regressor	1.545044899	0.019520044
Linear Regression	1.634927273	0.01952004
AdaBoost Regressor	1.780713797	0.197161436
Gradient Boosting Regressor	2.001030207	0.128690958
Decision Tree Regressor	2.405306816	0.259356499
Extra Tree Regressor	2.41068244	0.256337166
SVR	2.41068244	0.145899057
Bagging Regressor	9.203768015	4.586039305
Random Forest Regressor	Model is too big, cannot be put into Raspberry Pi	

After comprehensive comparison, we chose Gradient Boosting Regressor as our final model, because it's score is very high (0.997) and it's running time is also short (0.12s when the gas sensor is stabilized) compared to other models. Therefore, every time our gas sensor detected the values of the four factors ('pressure', 'gas concentration', 'temperature', and 'humidity') within the garbage can, these values will be normalized first according to the normalize rule of what we did before, and then be put into the Gradient Boosting Regressor and help calculate the

IAQ value, then if the value is greater than 150, our reminder system will inform the users that they need to dump this garbage can.

7 Lessons learned

In this project, we learned how to apply real-time systems to our daily life items, we learned how to use IOT, and we understood the factors that may affect real-time systems. We understand the importance of using real-time systems and reducing latency in real-world project development. We learned different ways to reduce latency time and efficiency.

In addition to real-time systems, we also learned how to design and complete projects rationally. In the process of completing a project, we need to understand the knowledge of IOT connections and the overall system architecture so that we can better optimize the delays and better fulfill the requirements needed.

In future project production, we learned to be conscious of the real-time latency of the system and not to focus on the implementation without caring about the efficiency of the implementation.

8 Conclusion and future work

We have designed and implemented a garbage dumping reminder system which has good efficient. Our work represents an implementation on developing the IOT and real-time system on a small item. Empirical results showed that our reminder system is highly efficient on reporting the garbage status in a short time. It has very little latency which is fully meet the user's needs and experience. Moreover, we use the AWS server to run database and website server which is very stable and safe. It can also handle large volumes of data and work.

For the future work, our system can not make multi-thread work since we only do experiments on one trash can. In the real world’s satisfaction, we need monitor many trash cans at the same time. In this case, parallel work should be taken into consideration since if we do all the work in sequence way, we cannot decrease latency and monitor all the trash cans in real time. We can also gather more dataset to make our model more. Furthermore, we can make our system dormant under normal conditions and make it only work when an abnormal situation occurs to save energy and decrease losses.

REFERENCES

[1] B. Chowdhury and M. U. Chowdhury, “RFID-based Real-time Smart Waste Management System,” in Australasian Telecommunication Networks and Applications Conference, 2007, no. December, pp. 175–180.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, 2014.

[3] F. Matten, “From smart devices to smart everyday objects,” Proc. Smart Objects Conf., no. April, pp. 15–16, 2003.

[4] https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme680/

[5] https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial