Smart Dustbin

Sensor Lab Mini Project Report submitted in partial fulfillment.

of the requirement for the degree of

T. E. (Information Technology)

Submitted By

20101A0009 – Sakshi Talele
20101A0014 – Akanksha Amdekar
20101A0021 – Satyam Mishra
20101A0023 – Rohan Kadu

Under the Guidance of

Prof. Kanchan Dhuri

Department of Information Technology

Vidyalankar Institute of Technology

Wadala(E), Mumbai 400 037

University of Mumbai

2022-23

CERTIFICATE OF APPROVAL

For

Mini Project Report

On

Sensor Lab

This is to Certify that

Sakshi Talele
Akanksha Amdekar
Satyam Mishra
Rohan Kadu

Have successfully carried out Mini Project entitled

"**Smart Dustbin**"

In partial fulfillment of degree course in

Information Technology

As laid down by University of Mumbai during the academic year 2022-23

Under the Guidance of
Prof. Kanchan Dhuri

Signature of Guide

Head of Department

Examiner 1

Examiner 2

ACKNOWLEGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. We express our profound gratitude we give to our **Prof. Kanchan Dhuri** Ma'am, our respectable project guide, for her gigantic support and guidance. Without her counseling our project would not have seen the light of the day.

We extend our sincere thanks to **Dr. Vipul Dalal**, Head of the Department of Information Technology for offering valuable advice at every stage of this undertaking. We would like to thank all the staff members who willingly helped us. We are grateful to VIDYALANKAR INSTITUTE OF TECHNOLOGY for giving us this opportunity.

The days we have spent in the institute will always be remembered and also be reckoned as guiding in our career.

1. **Sakshi Talele**
2. **Akanksha Amdekar**
3. **Satyam Mishra**
4. **Rohan Kadu**

Table of Contents

List of Figure

Abstract

This project deals with the problem of Dustbin management in smart cities, where the garbage collection system is not optimized. This project enables the organizations to meet their needs of smart garbage management system. As in our country tonnes of garbage is generated daily it is difficult to look after each place and it is major challenge in the urban towns to manage waste. This system will help in keeping the environment hygiene and clean with the help of reduction in garbage bin overflow.

So here we have come with the solution by designing a monitoring system which will take into consideration the garbage level of the bin using ultrasonic sensors. This system allows the user to know the percentage fill level of each garbage bin and if the dustbin is filled more than 80% then it will act accordingly in a locality or city at all time, to give a cost effective and time saving route to the concerned authorities. Also, it will display the Air Quality Index (AQI) inside the bin.

In this project we have used one 'MQ-2 Gas Sensor' to detect methane gas released from decomposing or rotten waste & two ' HC-SR04 Ultrasonic Sensor' to determine the level of garbage in the garbage bin. Once the garbage bin is 80% filled, a notification will be sent to the relevant people telling them to empty the garbage bins. A visual real time display of the bins can also be seen.

Introduction

Though the world is in a stage of up gradation, there is yet another problem that has to be dealt with. Garbage! Pictures of garbage bins being overfull and the garbage being spilled out from the bins can be seen all around. This leads to various diseases as large number of insects and mosquitoes breed on it. A big challenge in the urban cities is solid waste management. Hence, smart dustbin is a system which can eradicate this problem or at least reduce it to the minimum level.

Our present Prime Minister of India, Shri Narendra Modi ji has introduced the concept of implementing 100 smart cities in India. “Swachh Bharat Abhiyaan” was initiated to ensure a clean environment. Majority of viruses and bacterial infections develop in polluted environment. Safeguarding the environment using technology sources is needed at present. Majority of the public environment seems to be polluted with the waste material. So, modernization of the restaurants is needed by imparting the smart technology. Amounts of waste are largely determined by two factors: first, the population in any given area, and second, its consumption patterns.

Aim & Objectives

The aim of the project is to develop a Smart Bin Monitoring System which will monitor the level of trash and the Air Quality Level in the trash bin and according to the level of trash then it will be displayed on the app made on the MIT Inventor App.

The core objectives are:

- To build a clean environment and to empower the " SWACHH BHARATH" Mission.
- Provide a proper waste management system to avoid spreading of some deadly diseases.
- Gather system requirements
- Evaluate and study the platform required for the system
- Program Node mcu.
- Interface board with Ultrasonic Sensor, Gas Sensor.
- Program using MIT App Inventor
- Evaluate and test the system
- Maintain system
- Send notification
- Firebase Database

Problem Definition

Limitations of Existing System:

1) Time Consuming.

2) No alert when the bin gets full.

3) There is no record for working of garbage collectors.

4) Fuel consumption is more.

Instead of investing large amount of money and time on traditional methods of Garbage collection methods, to avoid the all above limitations, we present a “Smart Bin Monitoring System” which is automated, efficient, optimized and which will save our resources.

Proposed System

4.1 Block Diagram

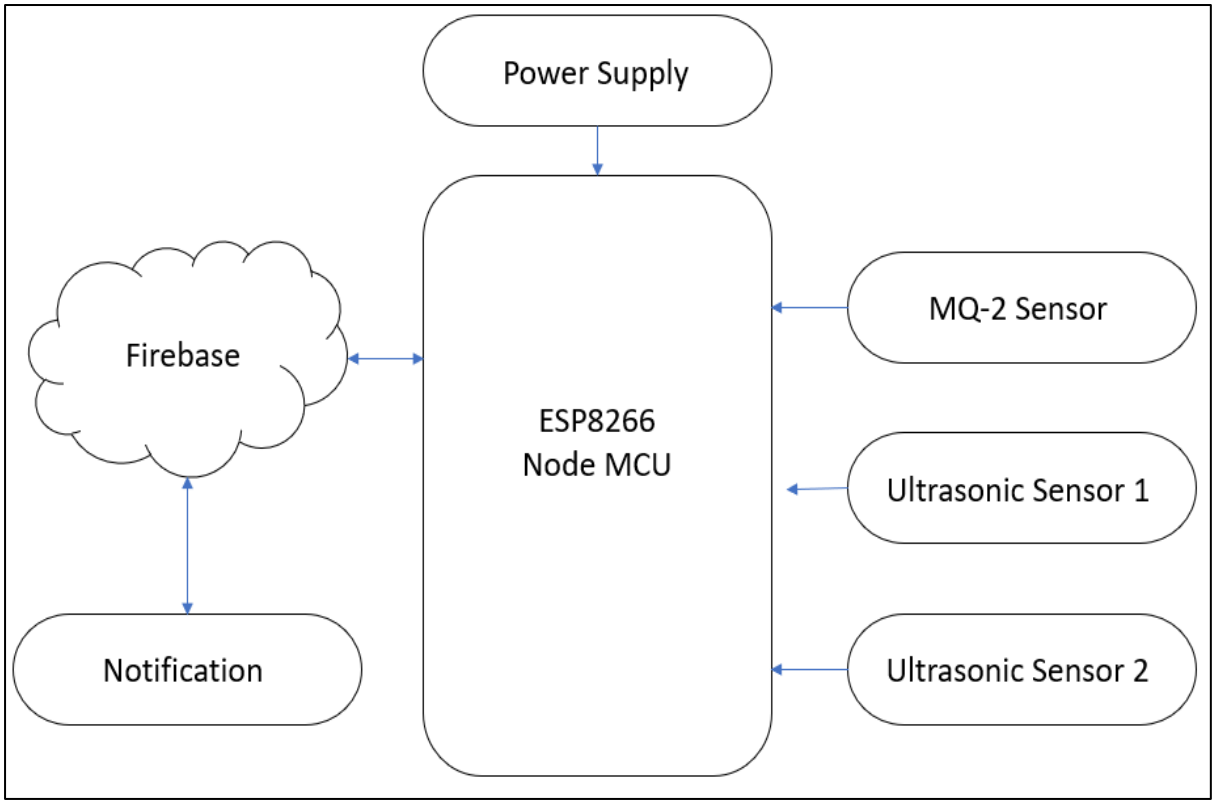


Fig 4.1: Block Diagram

4.2 Flow Chart

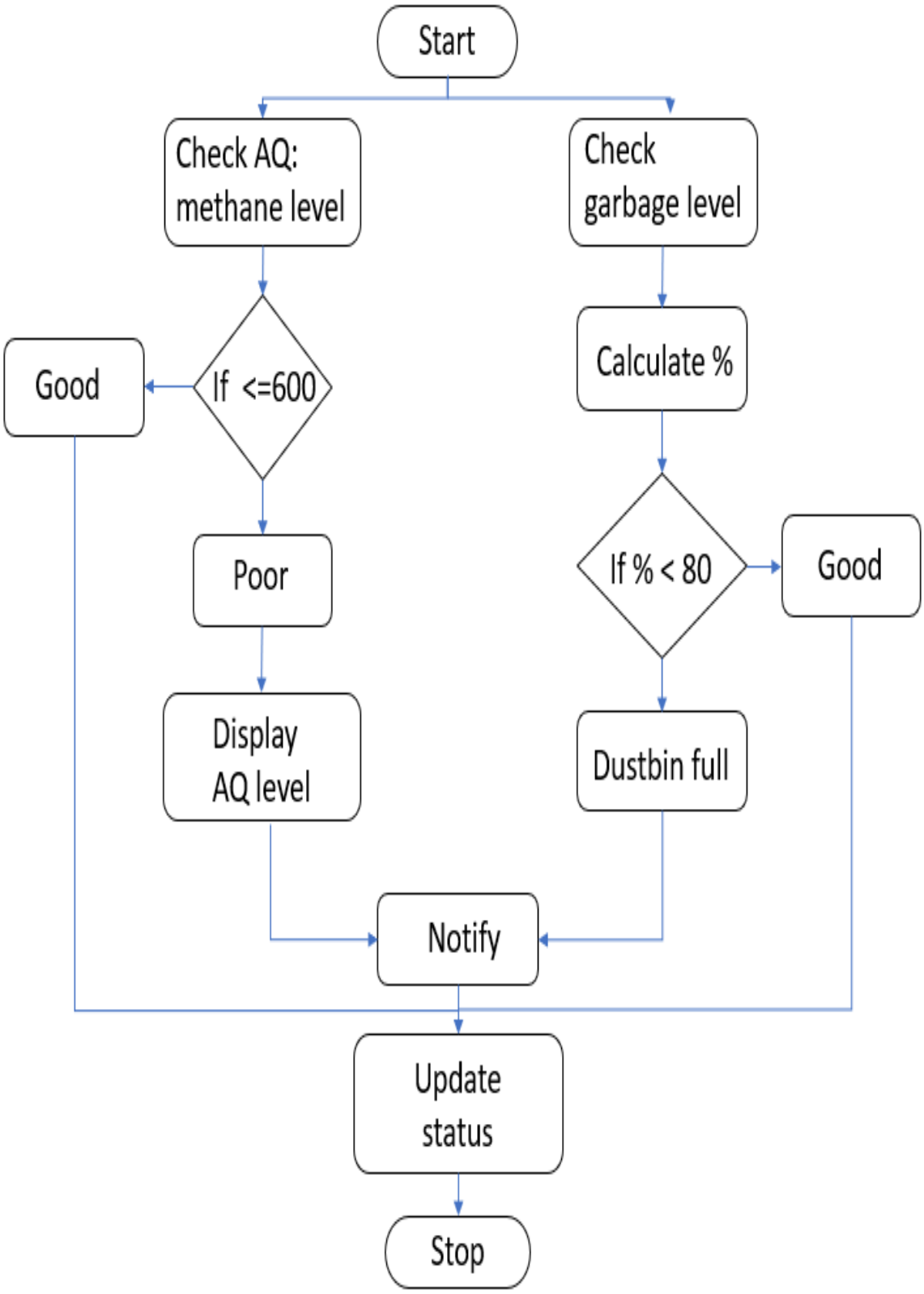


Fig 4.2: Flow chart

Components

5.1 Hardware

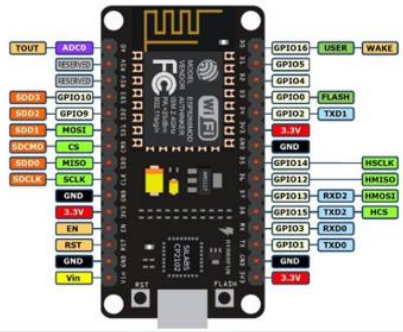
5.1.1 Node MCU:

NodeMCU is an open-source LUA based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board. Since NodeMCU is an open-source platform, its hardware design is open for edit/modify/build.

NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer to the ESP8266 WiFi Module.

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects



NodeMCU ESP8266 Pinout

Fig 5.1: Nodemcu

5.1.2 Ultrasonic Sensor

This sensor is a high-performance ultrasonic range finder. It is compact and measures an amazingly wide range from 2cm to 4m. This sensor can be connected directly to the digital I/O lines of your microcontroller and distance can be measured in time required for travelling of sound signal using simple formula as below. The module works on 5V DC input and also gives an output signal directly for detection of any obstacle up to 4M. As soon as the signals are transmitted the “Echo” pin goes to high level and remains in high level until the same sound waves are received by the receiver. If the received sound waves are same as what the same sensor transmitted then the Echo pin goes to low level. If no object is detected within 5M after 30ms the Echo signal will automatically go to low level. In our project, we used 2 Ultrasonic sensors; one for automatic opening and one for garbage level indication.

Fig 5.2: Ultrasonic Sensor

5.1.3 MQ-2 Gas Sensor

The MQ2 sensor is one of the most widely used in the MQ sensor series. It is a MOS (Metal Oxide Semiconductor) sensor. The MQ2 gas sensor operates on 3.3V to 5V DC and consumes approximately800mW.Itcandetect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide concentrations ranging from 200 to 10000 ppm. Note that the MQ2 gas sensor detects multiple gases, but cannot identify them!Therefore, it is best suited for measuring changes in a known gas density rather than detecting which one is changing.

Fig 5.3: MQ-2 Sensor

5.1.4 Breadboard

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holeson the top of the board. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

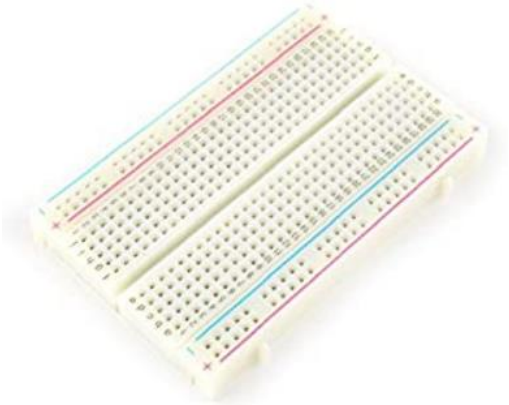


Fig 5.4: Breadboard

5.1.5 Jumper wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering.

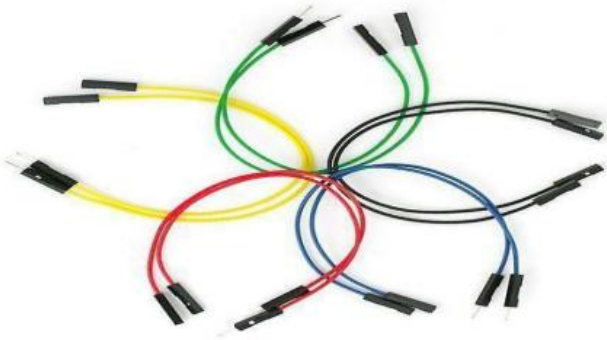


Fig 5.5: Jumper wires

5.2 Software

5.2.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

Fig 5.6: Arduino IDE

5.2.2 Firebase

It is a mobile application development platform from Google with powerful features for developing, handling, and enhancing applications. Firebase is a backend platform for building web and mobile applications.

Fig 5.7: Firebase

Code

HTML code:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dustbin Monitoring System</title>
    <script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-database.js"></script>
    <script src="https://smtpjs.com/v3/smtp.js"></script>
    <script>

        const firebaseConfig = {
            apiKey: "AIzaSyCAjgz4E_MoB95TUrIFx0cwIklH5AC5d1c",
            authDomain: "sensorlab-32dfe.firebaseapp.com",
            databaseURL: "https://sensorlab-32dfe-default-rtdb.firebaseio.com",
            projectId: "sensorlab-32dfe",
            storageBucket: "sensorlab-32dfe.appspot.com",
            messagingSenderId: "491529729745",
            appId: "1:491529729745:web:f6814c539237adfb8a37f0",
            measurementId: "G-546K27CR4F"
        };
        firebase.initializeApp(firebaseConfig);

        // Declearing All The Constant
        // var sensor1Value,sensor2Value,aqisensor1Value,aqisensor2Value;

        var dbRef = firebase.database().ref().child("garbageLevels");
        dbRef.on("value", function (snapshot) {
            var sensor1Value = snapshot.child("sensor1").val();
            var sensor2Value = snapshot.child("sensor2").val();
            document.getElementById("sensor1-value").innerHTML = sensor1Value;
            document.getElementById("sensor2-value").innerHTML = sensor2Value;
            // document.getElementById("trash1").style.height = (4*(sensor1Value)) + "%";
            // document.getElementById("trash2").style.height = (4*(sensor2Value)) + "%";


            if(sensor1Value <= 25  ){

                document.getElementById("trash1").style.height = (sensor1Value) + "%";
                document.getElementById("trash1").style.backgroundColor = "green";

            }
            else if(sensor1Value <= 75 && sensor1Value >25  ){

                document.getElementById("trash1").style.height = (sensor1Value) + "%";
                document.getElementById("trash1").style.backgroundColor = "greenyellow";

            }
```

```
    else{

       document.getElementById("trash1").style.height = (sensor1Value) + "%";
       document.getElementById("trash1").style.backgroundColor = "red";
    }

    if(sensor2Value <= 25  ){

       document.getElementById("trash2").style.height = (sensor2Value) + "%";
       document.getElementById("trash2").style.backgroundColor = "green";

    }
    else if(sensor2Value <= 75 && sensor2Value >25  ){

       document.getElementById("trash2").style.height = (sensor2Value) + "%";
       document.getElementById("trash2").style.backgroundColor = "greenyellow";

    }
    else{
       document.getElementById("trash2").style.height = (sensor2Value) + "%";
       document.getElementById("trash2").style.backgroundColor = "red";
    }




    document.getElementById("trash2").style.height = (sensor2Value) + "%";
    if(sensor1Value >= 80 || sensor2Value >=400) {
    Email.send({
       Host: "smtp.elasticemail.com",
       Port:2525,
       Username: "sm6003830@gmail.com",
       Password: "3A012D29061764FF57AA2881EB368794B8E4",
       To: 'mishrasatyam2643@gmail.com',
       From: "sm6003830@gmail.com",
       Subject: "Dustbin is full!",
       Body: "<html><h1>Please empty the dustbin. </h1></html>"
    })
}
});
var dbRef = firebase.database().ref().child("aqi");
dbRef.on("value", function (snapshot) {
    var aqisensor1Value = snapshot.child("aqisensor1").val();
    var aqisensor2Value = snapshot.child("aqisensor2").val();
    document.getElementById("aqi-sensor1-value").innerHTML = aqisensor1Value;
    document.getElementById("aqi-sensor2-value").innerHTML = aqisensor2Value;
    if (sensor2Value >= 80 || aqisensor2Value >=400){
    Email.send({
       Host: "smtp.elasticemail.com",
       Port:2525,
       Username: "sm6003830@gmail.com",
       Password: "3A012D29061764FF57AA2881EB368794B8E4",
       To: 'mishrasatyam2643@gmail.com',
```

```
            From: "sm6003830@gmail.com",
            Subject: "Test email",
            Body: "<html><h1>Please Empty Your Dustbin..... </h1></html>"
        })
    }

    });

</script>

<style>
    * {
        margin: 0;
        padding: 0;
    }

    .main {
        display: flex;
        flex-direction: column;
    }

    .level {
        display: flex;
    }

    .dis {
        display: flex;
        gap: 4rem;
        margin: 2rem auto;
        border-radius: 2rem;
        box-shadow: 0px 0px 8px 0px rgba(0 0 0 /25%);
        padding: 1rem 5rem;
    }

    html {
        font-family: Cairo;
        display: block;
        margin: 0px auto;
        text-align: center;
        color: #333333;
        background-color: #fbfbfb;
        overflow: hidden;
    }

    body {
        margin: 0%;
        padding: 0%;
    }

    .info {
        position: absolute;
        display: flex;
        flex-direction: column;
        padding: 1rem;
        background: white;
```

```
    border-radius: 1rem;
    top: 5rem;
    box-shadow: 2px 2px 2px 0px rgba(0 0 0 /25%);
}

.col {
    display: flex;
    gap: 2rem;
    margin-bottom: 1rem;
}

.red {
    background-color: rgb(213 0 35);
    width: 2rem;
    height: 2rem
}

.yellow {
    background-color: rgb(213, 213, 0);
    width: 2rem;
    height: 2rem
}

.green {
    background-color: rgb(1.5 213 0);
    width: 2rem;
    height: 2rem
}

h1 {
    width: 100%;
    font-size: 2rem;
    background-color: rebeccapurple;
    padding: 1rem;
    color: white;
}

h2 {
    font-family: Poppins;
    font-size: 1.4rem;
    margin-bottom: 1.5rem;
}

.container {
    display: flex;
    margin: auto;
    justify-content: center;
    gap: 5rem;
    top: 2rem;
    position: relative;
}

.level {
    display: flex;
    justify-content: center;
```

```
}

.level input {
   position: relative;
   border: 0rem;
   top: 50%;
   transform: translateY(-50%);
}

.lid {
   position: relative;
   width: 20rem;
   height: 3rem;
   background: rgb(50, 38, 38);
   border-radius: .4rem;
   outline: #000000 solid 1px;
   z-index: 1;
}

.jarr {
   position: relative;
   width: 20rem;
   /* background-color: rgb(255, 255, 255); */
   background-color: rgb(226 226 226);
   height: 415px;
   /* border-left: 2px solid black;
   border-right: 2px solid black;
   border-bottom: 2px solid black;  */
   clip-path: polygon(0% 0%, 100% 0%, 90% 100%, 10% 100%);
}

#trash1 {
   position: absolute;
   width: 100%;
   /* background-color:greenyellow; */
   height: 0;
   bottom: 0rem;
   transition: height 0.5s;
}

.jarl {
   position: relative;
   width: 20rem;
   /* background-color: rgb(255, 255, 255); */
   background-color: rgb(226 226 226);
   height: 415px;
   /* border-radius: 0rem 0rem 5rem 5rem solid black; */
   clip-path: polygon(0% 0%, 100% 0%, 90% 100%, 10% 100%);

}

#trash2 {
   position: absolute;
   width: 100%;
   /* background-color:greenyellow; */
```

```
            height: 0;
            bottom: 0rem;
            transition: height 0.5s;
        }


        .top {
            width: 20rem;
            height: 1rem;
            background: #2f2f2f;
            border-radius: .4rem .4rem 0rem 0rem;
        }

        .handle {
            width: 5rem;
            height: .8rem;
            background: #ffffff;
            border: 2px solid #333333;
            border-radius: .4rem .4rem 0rem 0rem;
        }

    </style>
</head>

<body>

    <div class="main">
        <h1>Dustbin Monitoring System</h1>
        <!-- <div class="dis">
            <div class="level">
                <input type="color" id="color-picker" value="#3cda2a" disabled>
                <h2>level between 0% to 25%</h2>
            </div>
            <div class="level">
                <input type="color" id="color-picker" value="#da9e2d" disabled>
                <h2>level between 25% to 75%</h2>
            </div>
            <div class="level">
                <input type="color" id="color-picker" value="#c00000" disabled>
                <h2>level between 75% to 100%</h2>
            </div>
        </div> -->
        <div class="container">
            <div class="conright">
                <h2>First Dustbin</h2>
                <div class="jarr">
                    <div id="trash1"></div>
                    <!-- <div class="handle"></div> -->
                    <div class="top"></div>

                </div>
                <br>
                <h3>GarbageLevel: <span id="sensor1-value"></span> </h3>
                <br>
                <h3>Air Quality Index (AQI): <span id="aqi-sensor1-value"></span> PPM</h3>
```

```
        </div>
        <div class="conleft">

          <h2>Second Dustbin</h2>

          <div class="jarl">
            <div id="trash2"></div>
            <!-- <div class="handle"></div> -->
            <div class="top"></div>

          </div>
          <br>
          <h3>GarbageLevel: <span id="sensor2-value"></span> </h3>
          <br>
          <h3>Air Quality Index (AQI): <span id="aqi-sensor2-value"></span> PPM</h3>
        </div>
      </div>

    </div>

</body>

</html>
```

Node MCU Code:

```
#include <FirebaseESP8266.h>
#include <ESP8266WiFi.h>

//  Wi-Fi network credentials
const char* ssid = "Sakshi";
const char* password = "shreya123";

// Firebase project's URL and secret
#define FIREBASE_HOST "https://sensorlab-32dfe-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "AIzaSyCAjgz4E_MoB95TUrIFx0cwIklH5AC5d1c"

// Initialize Firebase
FirebaseData firebaseData;
FirebaseData firebaseData1;

// ultrasonic sensor pins
#define trigPin D7
#define echoPin D8
#define trigPin1 D5
#define echoPin1 D6

// Define the maximum distance to detect garbage level (in centimeters)
#define maxDistance 50

void setup() {
  // Start serial communication
  Serial.begin(9600);
```

```
  // Connect to Wi-Fi network
  WiFi.begin(ssid, password);
  Serial.println("Connecting to Wi-Fi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to Wi-Fi...");
  }
  Serial.println("Connected to Wi-Fi!");

  // Initialize the ultrasonic sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Initialize Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Serial.println("Firebase initialized!");
}

void loop() {
  // Get the distance from the ultrasonic sensor
   float duration, distance,percentage,duration1, distance1,percentage1;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;
  percentage = (3*(100-(((96-distance)/96)*100)));
  Firebase.setString(firebaseData, "/garbageLevels/sensor1", String(percentage).c_str());
  if (firebaseData.dataAvailable()) {
      Serial.println("Garbage level of First sensor uploaded to Firebase!");
    } else {
      Serial.println("Error uploading garbage level of First Sensor to Firebase:");
      Serial.println(firebaseData.errorReason());
    }
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = (duration1 / 2) / 29.1;
  percentage1 = (3*(100 - (((96-distance)/96)*100)));

  Firebase.setString(firebaseData1, "/garbageLevels/sensor2", String(percentage1).c_str());
    if (firebaseData1.dataAvailable()) {
      Serial.println("Garbage level of second sensor uploaded to Firebase!");
    } else {
      Serial.println("Error uploading garbage level of Second Sensor to Firebase:");
      Serial.println(firebaseData1.errorReason());
    }
}
```

Implementation

7.1 Circuit Diagram

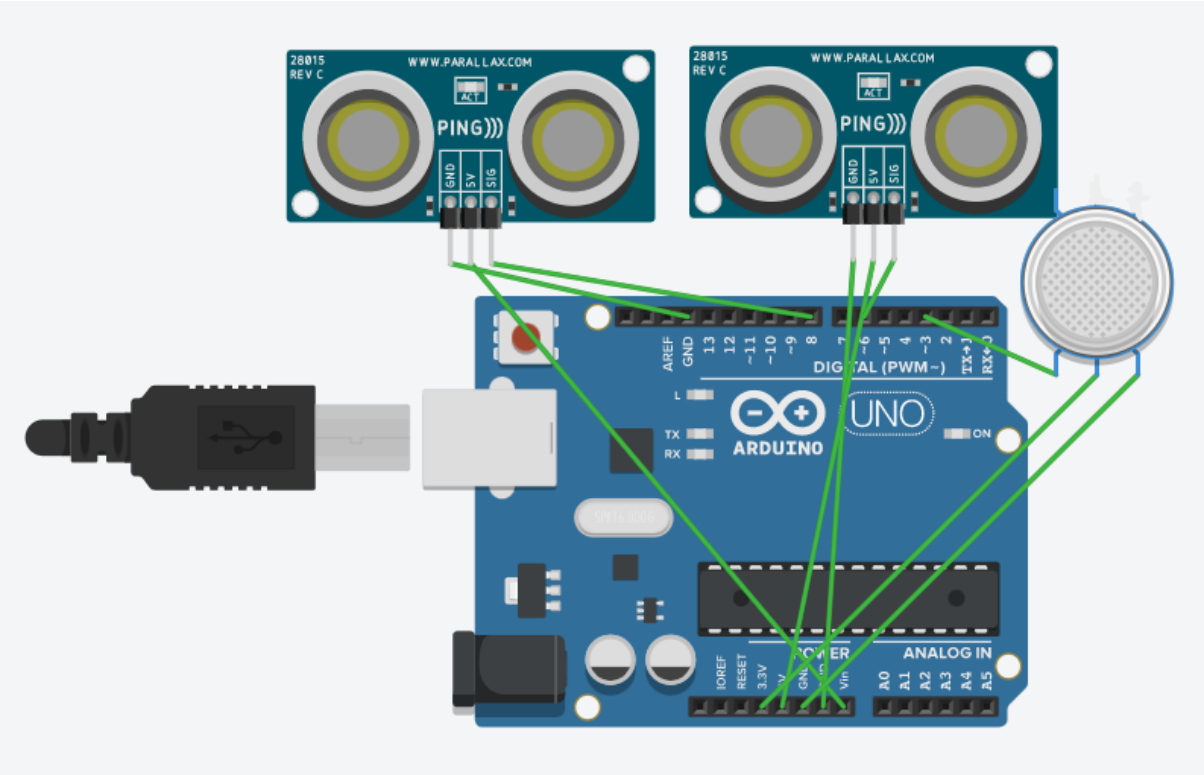


Fig 7.1: Circuit Diagram

Connections

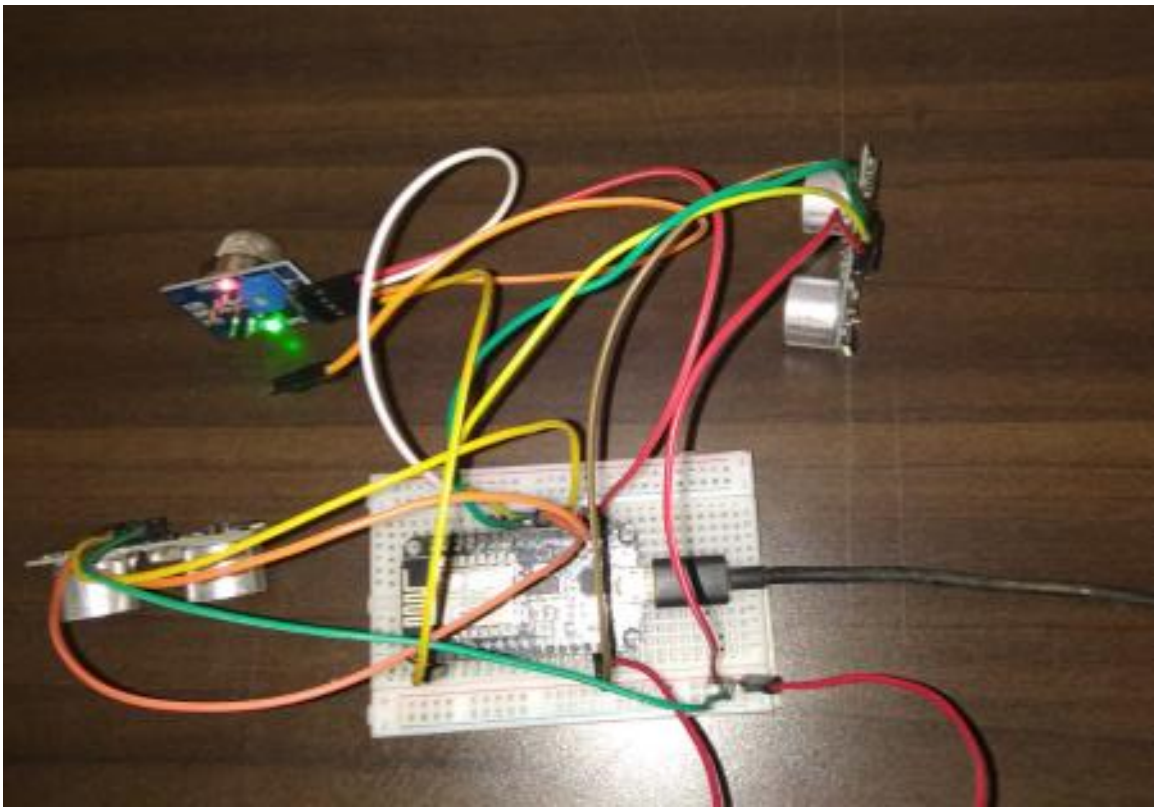


Fig 7.2: Connections

7.2 Working

The ultrasonic sensor at the bottom of lid will indicate the status of Bin. The ultrasonic sensors read the distance, and then calculate the percentage of how full the dustbin is. If the trash percentage is greater than 80, The trash bin in the visual display will turn “RED” and if the trash percentage is greater than 25, It will turn “YELLOW” and if Trash percentage is lesser than or equal 25 it will indicate “GREEN” displaying the Trash percentage.

If the dustbin fills upto 80% or more of its capacity, a notification is sent to the registered email id of the authorities.

To calculate the AQI inside the bin, we are using the MQ-2 Gas Sensor which helps in calculating the AQI of the bin. If the AQI inside the bin is less than 370 (AQI < 400), then the AQI is considered as “Good”. If the value of AQI is between 400 and 600 (400 < AQI < 600), then the AQI is considered as “Moderate”. If the value of AQI is greater than 600 (AQI > 600) then the AQI is considered as “Poor”. If the AQI is “Poor” then an alert is sent on the mobile phone alerting to empty the bin.

Results

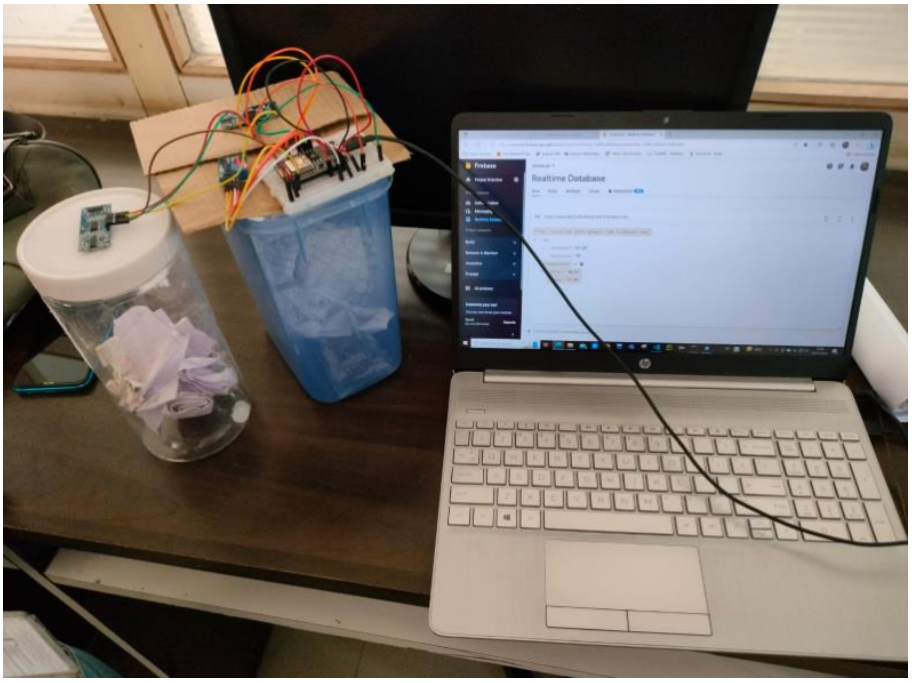


Fig 8.1: Setup

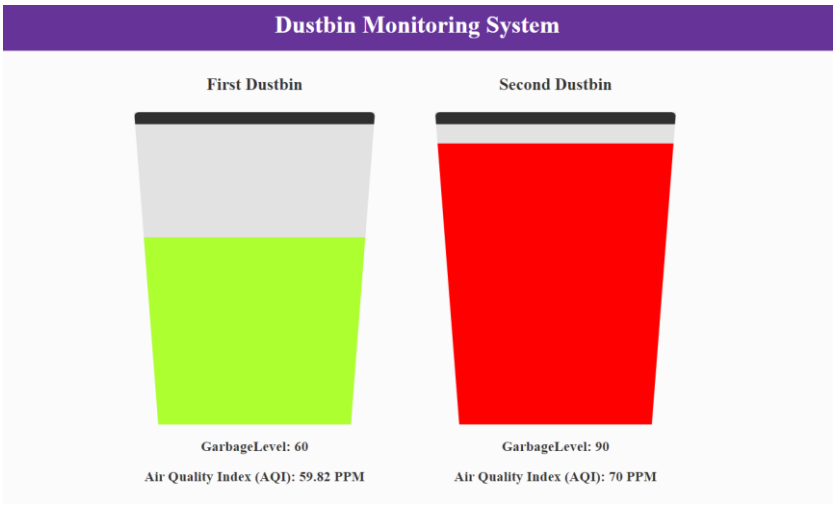


Fig 8.2: Visualization-1

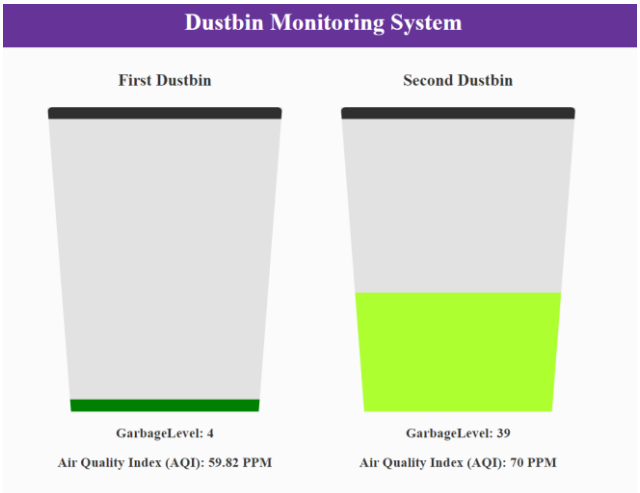


Fig 8.3: Visualization-2

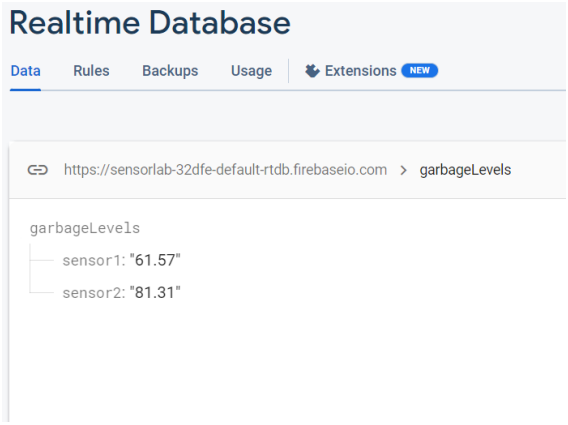


Fig 8.4: Realtime Database

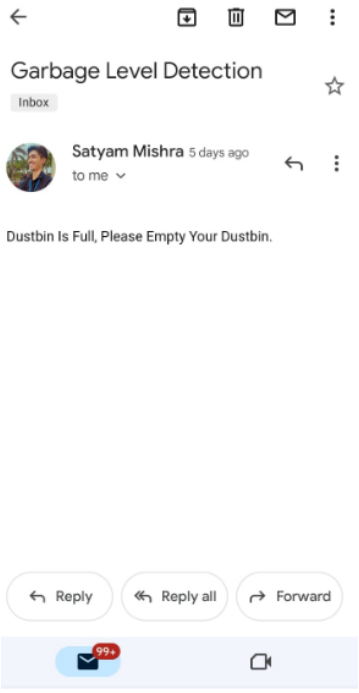


Fig 8.5: Email Notification

8.1 Discussion

During the whole period of the project, we gained a lot of knowledge on the Bread board and programming in Arduino IDE. If we talk about the achievements out of the project when starting to do the project it was to interface it with the ultrasonic sensor, We came to know a lot about the sensor that they only work at their given input voltage some sensor require to calibrate before they can be used and transmit that data via any wireless technology.

Conclusion & Future Scope

We have made a successful attempt in resolving the issues faced by the citizens and the workers by making a working model which is described in this project report. We have successfully created a garbage monitoring system which is able to accurately detect the level of the garbage and it also detects the smell emitted by the gas and display it on the application which on the mobile and send the alerts accordingly.

As to the testing and result analysis, the system able to be as an observing system where it will send reminder and alert notifications to the cleaning department when the ultrasonic sensor calculates the level of the rubbish in the dustbin.

Proposed model in this project can be implemented by the Garbage Management authorities as this is efficient, optimized, and automated and which will save resources and meet their objectives. It can be a small contribution towards “Swachha Bharat Abhiyan” & “Smart City”.

The above method is just a stepping stone for implantation of IOT. There can be many enhancements done for this prototype which can be a revolutionary change in maintaining our environment clean and healthy.

The few enhancements can be done are:

- The implementation of more collective bins placed side by side where it automatically detects the type.

- These dustbins can be placed with a GPS tracker where the dustbins in a particular locality can be located easily and the waste can be emptied.

References

[1] https://www.researchgate.net/publication/339301914

[2] https://www.instructables.com/id/Smart-Garbage-Monitoring-System-Using-Internetof-/

[3] https://youtu.be/dEzw_PYTwFw

[4] https://components101.com/ultrasonic-sensor-working-pinout-datasheet

[5] https://youtu.be/9yrP1CZN3Ds

Reviewed Papers:

1. Telugu Maddileti, Harish Kurakula, 'IOT BASED SMART DUSTBIN'- International Journal Of Scientific & Technology Research Volume 9, Issue 02, February 2020.

2. M.R. Sundarakumar, S. Arun kumar, S. Naveen, Kamal Kumar Ray, 'SMART DUSTBINGARBAGE MONITORING SYSTEM BY EFFICIENT ARDUINO BASED'- Elsevier's SSRN eLibrary – Journal of Information Systems & eBusiness Network -ISSN: 1556-5068 2017.

3. Ms. Kalyani S.Ghuge, Mrs. Sangita M.Jaybhaye, Mrs.Bharati.P.Vasgi, 'SMART DUSTBIN USING NODEMCU' - Journal of Applied Science and Computations(JASC) ISSN NO: 1076-5131 Volume VI, Issue VI, JUNE/2019.

4. Jamil Abedalrahim Jamil Alsayaydeh, Adam Wong Yoon Khang, Win Adiyansyah Indra, Vadym Shkarupylo and Jayananthinii Jayasundar, "DEVELOPMENT OF SMART DUSTBIN BY USING APPS"- ARPN Journal of Engineering and Applied Sciences VOL. 14, NO. 21 - November 2019, ISSN 1819-6608.

GitHub Link: