Smart Waste Management System

IoT (SS-2021)

at Hof University of Applied Sciences
Department of Computer Science
Software Engineering for Industrial Applications

Submitted to
Mr. Ivan Iakimenko
Prof. Dr. Jürgen Heym
Alfons-Goppel-Platz 1
95028 Hof
Germany

Submitted by
Manoj Prabakar
Satheeshkumar

Hof, July 24, 2021

Table of Contents

List of Abbreviations

API	Application Programming Interface
IDE	Integrated Development Environment
IFTTT	If this then that
IMU	Inertial Measurement Unit
IoT	Internet of Things
I2C	Inter Integrated Circuit
JSON	JavaScript Object Notation
mAh	milliampere-hour
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity

List of Figures

Chapter 1. Problem Statement

In developed cities, the metropolitan municipality team will monitor the dustbins around an area in a frequent period to check whether the dustbin is filled or not. Their time is wasted if the dustbin is not full which will affect their other works severely as well. Also, when the people who put the trash in the dustbin see that the garbage is full, they will have to wait for a certain period till the team arrives and clears out the trash. Till then they must keep their garbage in their house which can lead to attract more insects or mosquitoes pondering in their house. These incidents might surely anger both the people and the municipality team. It also affects the city's reputation which will be regarded as very less effective in hygiene maintenance. Our project-connect deals with M5StickC attached to the dustbin which will detect the distance between the garbage and the garbage lid, when its nearly filled it can then send a message to inform the team to clear the trash out that can help the situation.

Chapter 2. Project Specification

2.1. Problem Statement Summary

The garbage wastes are flooded in the streets which causes damage to the environment and the city's reputation. There are several scenarios in which the problem occurs.

- Each area / street has a common dustbin where the public disposes the garbage.
- Every individual household has home trash which gets overflown.
- Every organization has large number of commercial wastes which needs frequent disposals.

The waste management team will monitor 3 or 4 times per day to clear the bins out if garbage is full. If it is full and the trash is not cleared in the correct time, the garbage will have been overflown which is a pain for all, i.e., public, individual household, organization, and waste management team. As a result, the consequences of delay in garbage collection leaves the place in stink impacting city's reputation and affects public health. Because of the garbage overflow many insects and mosquitoes swirl around the place, this will further cause infections such as Malaria, Dengue, etc….. [1.] [2.]

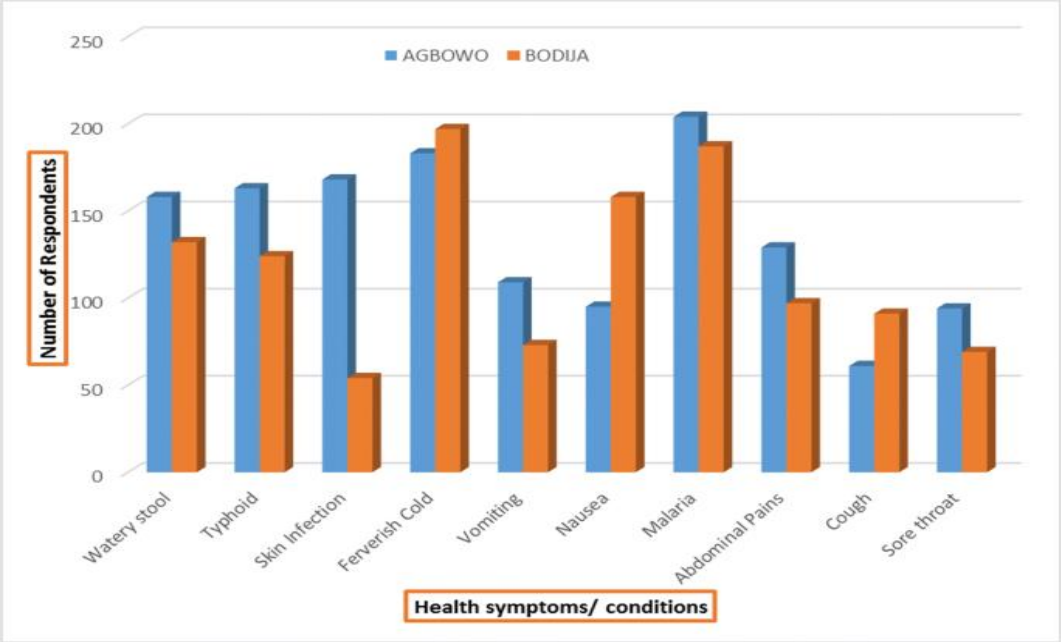

Figure 1. Impact of waste management

The above graph shows the impact the poor waste management has on public community health in the city of Ibadan – Nigeria [3.].

2.2. Sensing Requirements

There are several ways to check to see if the garbage bin is filled or not. We can track when someone opens the garbage lid and dumps into it using accelerometer sensor and send a notification to the team when bin is updated with another wastage. We can track if the garbage has any wastage filled by sensing the distance covered by the wastage inside the bin using ultrasonic sensor. Notifications should be sent as soon as 85% of the bin is filled which gives the time for team to come and clear out the bin before its fully filled. The limitation we think is the garbage bin should be closed properly to check the correct distance between the wastage and the lid.

2.3. Sensor Types

To measure the distance between the lid of the garbage bin and the waste filled inside the bin, we could use an ultrasonic sensor attached to the inside of lid which measures the distance by emitting the sound from the inside of lid to the targeted object (i.e., wastage). We can trigger the M5StickC to sense the distance only when garbage bin is opened or closed by using an accelerometer sensor which senses if the object is moved or not.

2.4. Sensor Implementation Examples

We have used sample code for advanced Arduino library for HC-SR04 ultrasonic sensor as a source of reference for our project implementation. The main function for the sensor is the getHitTime and distanceInCm functions which will return feedback about packets sensor sent out before timeout value [4.].

Chapter 3. Hardware Design

3.1. Hardware Summary

The hardware's we are going to use in our project are M5StickC, Ultrasonic Sensor, Accelerometer (In-Built) and grove cable. We are going to use the grove cable to connect the ultrasonic sensor to M5StickC. Essentially the grove port uses I2C Communication, and we can connect to 128 devices, suppose if we need more sensors in the future. The reference logic to implement using these sensors has been mentioned in the code example in the sensor requirements table. Whenever the trash bin opened and closed, we will read the data from Ultrasonic sensor to measure the distance to the trash and send it to the end user. The M5StickC consumes 80mAh its battery life is low. So, we can provide power via power bank with the high capacity. [5.] [6.]

3.2. Complete System Circuit Diagram

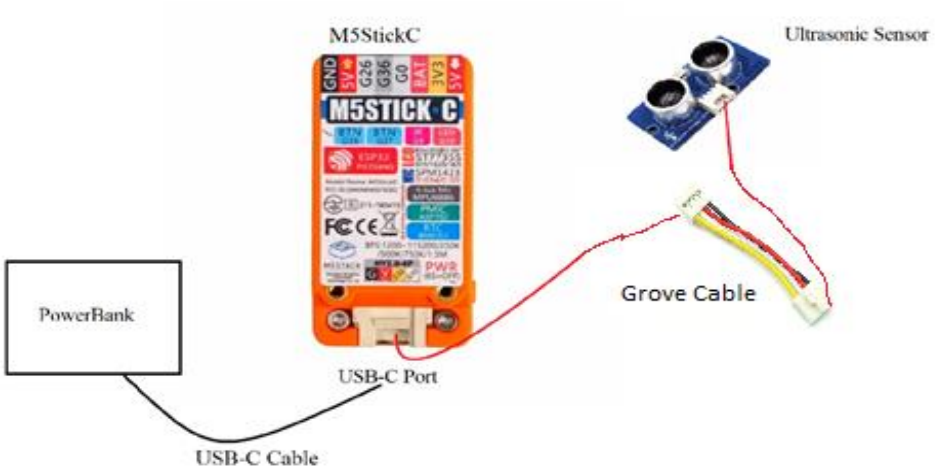

Figure 2. Circuit Diagram

- Power Bank is used for constant power supply to the M5StickC connected via USB-C port.
- Pins in Grove Port – GPI033, GPI032, 5V and GND [7.].
- M5StickC and Ultrasonic sensor has been connected using grove cable.
- Accelerometer is inbuilt with M5StickC.

3.3. Hardware Installation Diagram

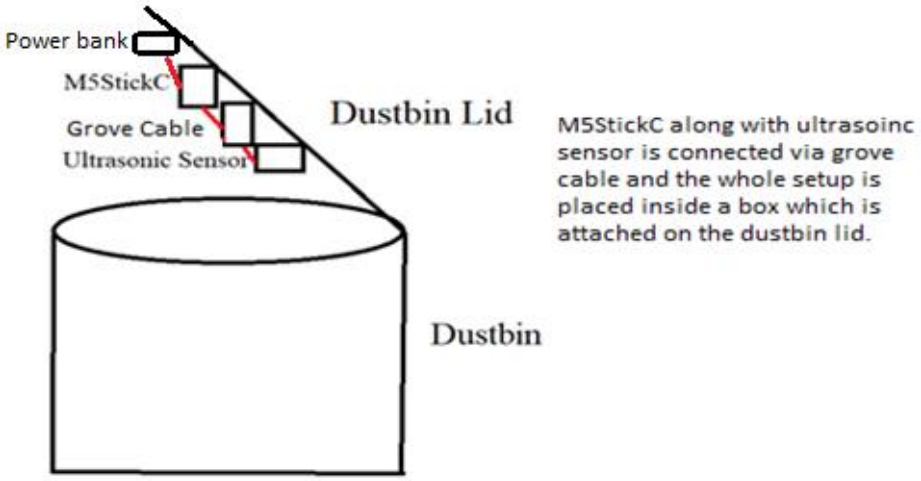

Figure 3. Hardware Installation Diagram

Chapter 4. Cloud Integration

4.1. Overall System Functionality

4.1.1. Cause-Effect Description

In most cities, the garbage is overflown which causes a serious negative impact on health and cleanliness issues. Our project-connect focuses on informing the garbage collector serviceman by phone or email notification whenever garbage is going to be full so that overflow can be avoided. People pour garbage in the trash can, so when the can is opened and closed, serviceman will be updated with the value (which is the distance from garbage to lid of bin) to understand how much space its remaining before it is filled.

4.1.2. Input-Data sources and Threshold to trigger

Input Data sources are ultrasonic sensor data value which measures the distance value from lid to garbage, accelerometer sensor value to sense the position of lid. The threshold to trigger the

reaction is opening and closing of garbage lid. Once the lid is closed, we will sense the ultrasonic sensor value and call the API to send distance value.

4.1.3. Possibility of errant reading

There is a possibility for errant readings which is the garbage lid being left open and not closed, in that case, we plan to mitigate them by setting a timer of 25 seconds when garbage lid is opened, if within that limit lid is not closed, then we will send notification.

4.2. Included Cloud Systems

4.2.1. Cloud software and protocol for data transfer

We are planning to use both Adafruit.io [8.] and IFTTT [9.] as cloud services. In Adafruit.io, we are using the dashboard for representing the trash filled (in percentage). When the filled percentage exceeds the limit of the trash (user-defined value), we will notify the service man using IFTTT cloud. IFTTT cloud software has various ways to notify our data in human readable format and offers different services like mail notification, exporting data to excel, notifying by phone message etc... We are using WIFI protocol for integrating with Adafruit.io cloud service. An API gateway is used behind the scenes for inter-communication between Adafruit.io and IFTTT.

4.2.2. Cloud services used for data-processing

We will send the data based on change of position of the garbage lid by reading the accelerometer value and ultrasonic distance value to calculate the volume of garbage bin. Adafruit.io and IFTTT requires us to have account. Services which we are using from Adafruit.io are feed and dashboard, and from IFTTT we are monitoring the feed on Adafruit.io and email notification. We will send the ultrasonic measurement value from M5StickC to Adafruit.io feed. IFTTT will trigger the email notification with the value once the adafruit service in the applet gets called.

4.2.3. Interaction b/w M5StickC and the End User

M5StickC —> Read accelerometer, ultrasonic values —> Send the values to Adafruit.io feed (using WIFI protocol) —> Call the adafruit service in IFTTT applet (via API gateway) —> Trigger email —> End User.

4.3. Interaction Diagram

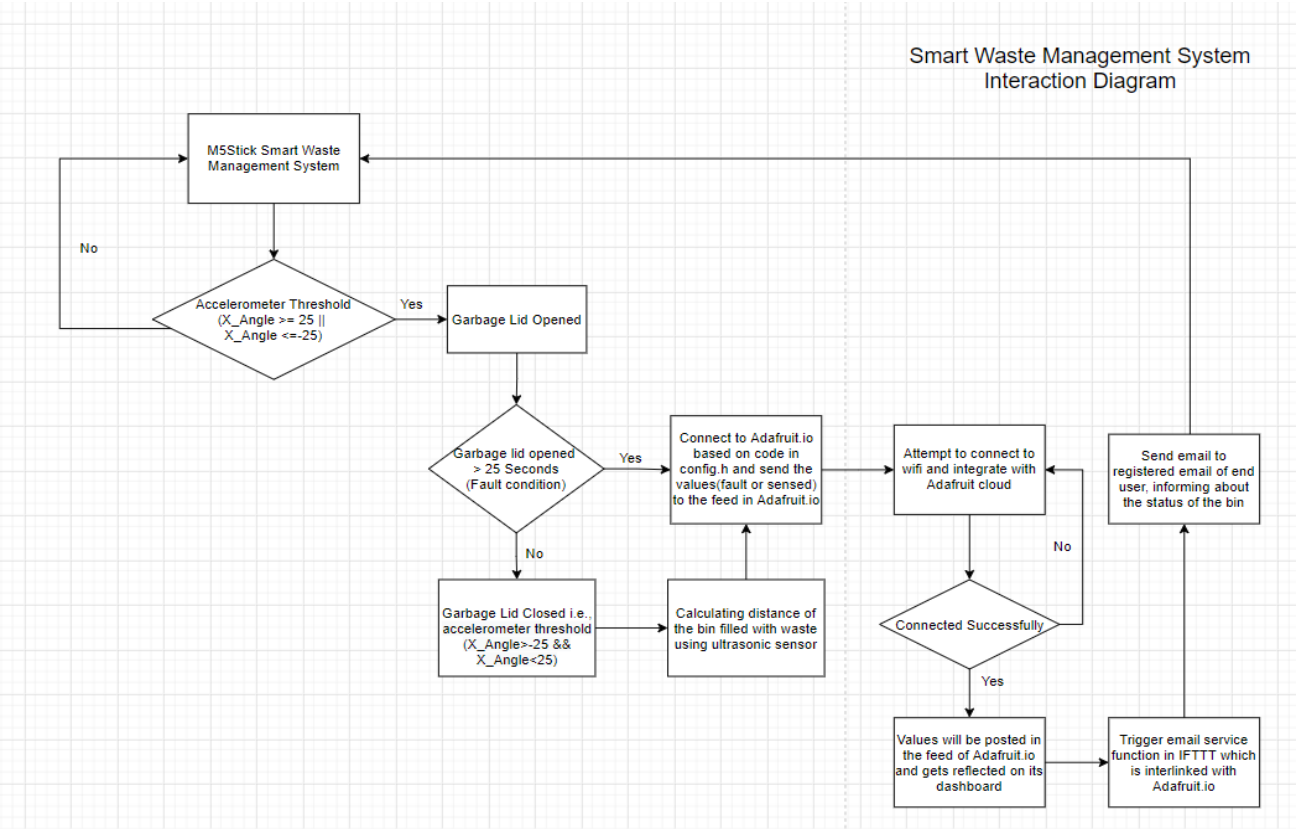

Figure 4. Interaction Diagram

4.4. Data Formatting

4.4.1. Transmitting data

To transmit the data to the IoT cloud service, we are using Adafruit.io. We have created the feed, which receives the values and displays on the dashboard in a user understandable format. Using API gateway, the adafruit service in IFTTT will be monitoring the values from the Adafruit.io feed and triggers email when the value exceeds the limit.

4.4.2. Sample Email format

Dear Team,

Kindly note that the volume of garbage is {{trash filled percentage}} at {{date and time}}

With Regards,
Group 6.

4.4.3. Sample Code

IFTTT_API

```
47 void accelermeterValueDetection(){
48   if (accY > 75 + initialaccY){
49     openedLid = true;
50     }
51
52   if( openedLid && ( 5 + initialaccY > accY < initialaccY - 5 ) ){
53     value1 = readUltrasonicvalue();
54     sendGarbageValue(value1);
55     openedLid = false;
56     }
57   else if( openedLid && timeOfOpenedLid > 60 ){
58     value1 = " The Garbage Lid is not closed";
59       sendGarbageValue(value1); // sending garabage value for Errant Condition ( The lid is not closed )
60     }
61 }
62 void loop() {
63     accelermeterValueDetection();
64     delay(500);
65 }
66
67 void sendGarbageValue(value1) {
68   // Send an HTTP GET request
69   // Check WiFi connection status
70   if (WiFi.status() == WL_CONNECTED) {
```

Figure 5. Sample Code

Chapter 5. Proof of Concept

5.1. Device Setup

The following are the steps for setting up the project.

Steps for setup:

- Place the M5StickC and ultrasonic sensor inside a box and mount the box on the lid of the dustbin by using double sided tape.
- Inside the box, make 2 holes for the ultrasonic sensor to detect the distance from the nearest waste inside the bin.
- Ultrasonic sensor is placed facing towards the garbage and M5StickC is placed in opposite direction to ultrasonic sensor.
- Connect the ultrasonic sensor to M5StickC using the grove port cable.
- Place the power bank inside the box to power the M5StickC and connect the power bank to the USB-C port of M5StickC.

Figure 6. Top View

Figure 7. Full Setup

5.2. Cloud Service

We are using the adafruit.io cloud to analyse the data received from M5StickC and IFTTT to send the mail notification to the service team.

Figure 8. Adafruit.io Dashboard

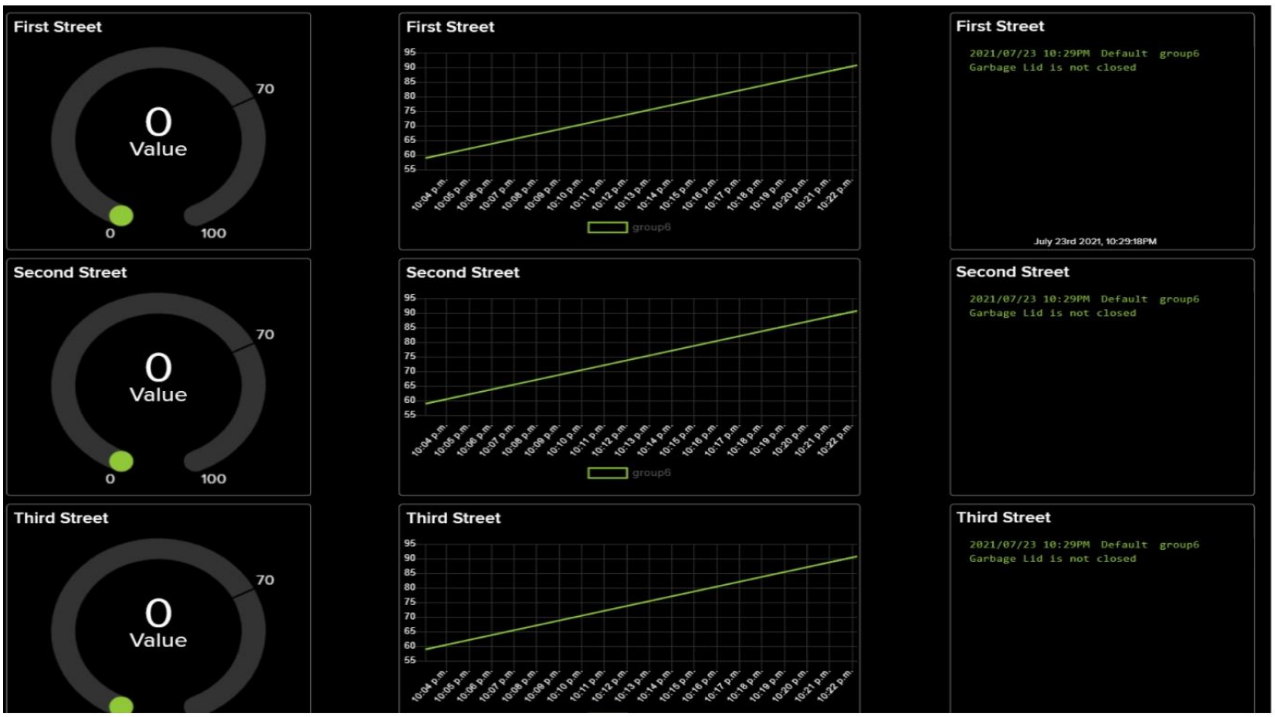

Figure 9. Commercial use case

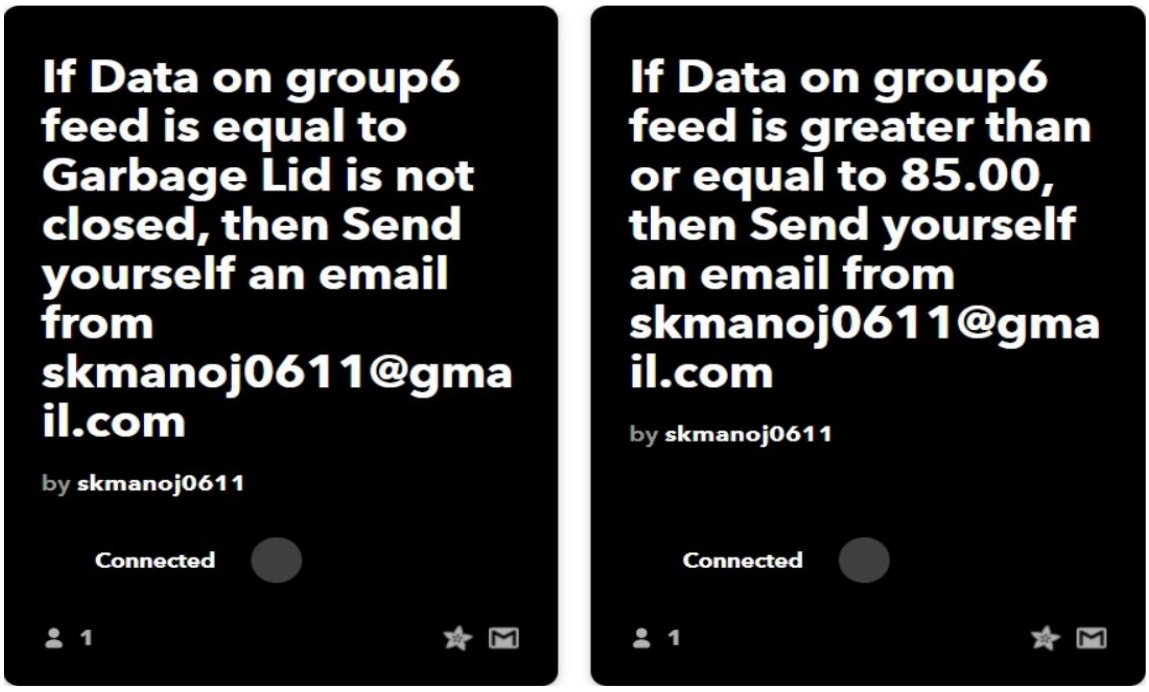

Figure 10. IFTTT applets

Chapter 6. Lessons Learned

First, we learned in detail about the features of M5StickC and its functionality. Though it is small device it is powerful in its service and performance. The other things we learned from working with M5StickC and Project Connect are as follows.

- There are inbuilt sensors within the M5StickC (Accelerometer, Gyroscope ...) which are quite useful for solving the direction-oriented problems (using IMU).
- Also, M5StickC can connect with Bluetooth and Wi-Fi technologies through which we can receive data as well as send data to cloud or any other smart devices.
- We also have hands-on exposure in transmitting data to API and receiving back the response through Wi-Fi and connecting the smart devices using Bluetooth.
- By working in Arduino IDE, we gathered information about the various libraries used for M5STickC.
- Various cloud platforms to analyse our data received from M5StickC such as Adafruit, Amazon Cloud, IFTTT etc...
- We have studied and practically applied about how to implement the ultrasonic sensor with M5Stick for measuring the distance.

6.1. Unique Challenges

- Selecting a cloud platform for data integration was the difficult problem we faced during our project connect considering the different number of use cases and the means to represent our data in a user understandable way. Initially we used IFTT to send our data through mail whenever it is updated, but it is not the optimized solution for delivering information to the end user. So, we have used Adafruit.io along with IFTTT. In Adafruit.io we represented our data using dashboards like graph, status indicator to the end-user.
- Ultrasonic integration with M5StickC:
  - Ultrasonic sensor with grove port or else with connecting wires.
  - Due to product availability of ultrasonic sensor with grove port along with grove cable, we have used that which makes our installation easier.
- Power Supply to M5StickC:
  - Initially we planned to charge the M5StickC by a charger once its power is drained out. This is not feasible because we need to remove M5StickC from our setup frequently.
  - So, we used power bank (10,000 mAh) to supply power constantly for M5StickC. This eliminates the need to remove M5StickC from the setup. M5StickC is 64 mAh, so the fully charged power bank will provide supply for 2 weeks approximately. During maintenance of garbage bin, the power bank will be charged using external supply.

Chapter 7. Future Improvements

If we had chance to redo this project again, we would have liked to use weight sensor as well along with ultrasonic sensor for determining the status of the garbage bin whether it is filled or not. We would want to improve our project in future by developing a web application and / or mobile application which displays our project connect information in a more ease of access way (without any user account). Also, we further wished to analyse the data obtained using machine learning algorithms for predicting the frequent time in which the garbage waste is filled.

Chapter 8. Bibliography

[1.] https://www.newindianexpress.com/cities/chennai/2019/dec/11/delay-in-garbage-collection-leaves-north-chennai-streets-in-stink-2074234.html

[2.] https://www.ecubelabs.com/overflowing-garbage-bins-5-impacts-on-health-and-environment-and-how-to-prevent/

[3.] https://link.springer.com/article/10.1007/s41050-018-0008-y

[4.] https://github.com/Spaguetron/ST_HW_HC_SR04

[5.] https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/

[6.] https://m5stack.github.io/UIFlow_doc/en/en/Hardwares/IMU.html

[7.] http://m5edu.com/Product/m5stick-c-micro-core/

[8.] https://io.adafruit.com/

[9.] https://ifttt.com/home

https://www.arduino.cc/reference/en/

Chapter 9. Appendix

Source Code: https://gitlab.hof-university.de/project-connect-group-6/project_connect_group_6

```
/* PROJECT CONNECT GROUP-6 SMART WASTE MANAGEMENT SYSTEM
* This is the code implementation of smart waste management system integrating with the adafruit io cloud
* and as well as ifttt for trigerring mail / message when the garbage is 85% filled.
*/

#include "Ultrasonic.h"
#include "config.h"
#include <M5StickC.h>

const float homeTrashHeight = 50;
const float commercialTrashHeight = 100;

float accX = 0.0F;
float accY = 0.0F;
float accZ = 0.0F;
double xAngle ;
long RangeInCentimeters;
float distanceValue;
float garbageFilledPercentage;
String garbageFilledStatus;
int delayCount = 0;
static bool openedLid = false;
// function to detect whether garbage lid is opened or not and triggering the data to adafruit accordingly.
void accelerometerTriggerDetection(double);
// function to read the ultrasonic sensor value.
float readUltrasonicvalue();



// set up the 'group6' feed
AdafruitIO_Feed *group6 = io.feed("group6");
// set up the ultrasonic sensor
Ultrasonic ultrasonic(33);

void setup() {

    M5.begin();
    Serial.begin(115200);
    M5.IMU.Init();
    M5.Lcd.setRotation(3);
    M5.Lcd.fillScreen(BLACK);
    M5.Lcd.setTextSize(1);
    M5.Lcd.setCursor(0, 10);
    M5.Lcd.println("Angle - X ");

    Serial.print("Connecting to Adafruit IO");
    // connect to io.adafruit.com
    io.connect();
    // wait for a connection
    while(io.status() < AIO_CONNECTED) {
      Serial.print(".");
      delay(500);
    }
    // we are connected
```

```
    Serial.println();
    Serial.println(io.statusText());
    pinMode(10, OUTPUT);
    digitalWrite(10, HIGH);
}

void loop() {

    io.run();
    M5.IMU.getAccelData(&accX,&accY,&accZ);
    xAngle = atan( accX / (sqrt(sq(accY) + sq(accZ)))); // got the angle calcuation from https://wizmoz.blog
    xAngle *= 180.00;
    xAngle /= 3.141592;
    M5.Lcd.setCursor(0, 20);
    M5.Lcd.printf(" %5.2f ",xAngle);
    // The accelerometer angle determines whether the lid is opened or not.
    accelerometerTriggerDetection(xAngle);
    delay(1000);
}

void accelerometerTriggerDetection(double xAngle){

    if(xAngle >= 25 || xAngle <= -25){
        Serial.print("Angle - ");Serial.println(xAngle);
        openedLid = true;
        digitalWrite(10, LOW); // Switching the LED on when the lid is opened.
```

```
        digitalWrite(10, LOW); // Switching the LED on when the lid is opened.
      }

    if(openedLid) {
      delay(5000);
      delayCount ++ ;
      // (i) After opening the lid and delay of 10 seconds , checking whether the lid is closed .
      // If lid is closed , then reading the ultrasonic value and converting to the percentage of garbage filled.
      // Then sending the filled % to adafruit feed to analyze the values and viewing in the dashboard.
      if(xAngle <= 25 && xAngle >= -25) {
        delayCount = 0;
        openedLid = false;
        digitalWrite(10, HIGH); // Switching the LED on when the lid is closed.
        Serial.println("Calculating garbage status after the dustbin is closed");
        distanceValue = homeTrashHeight - readUltrasonicvalue();
        garbageFilledPercentage = (distanceValue / homeTrashHeight ) * 100;
        if(garbageFilledPercentage < 0){
          return ;
          }
        garbageFilledStatus = String(garbageFilledPercentage);
        Serial.print(" % of Garbage Filled - "); Serial.println(garbageFilledStatus);
        group6->save(garbageFilledStatus);

      }
      //(ii) After opening the lid and delay of 25 secs ,if the lid is still not closed (Handling Error condition).
      // Then sending the error trigger value "Garbage Lid is not closed" to adafruit feed.
```

```
      else if((xAngle >= 25 || xAngle <= -25) && delayCount > 4) {
        delayCount = 0;
        // sending garabage value for Errant Condition ( The lid is not closed )
        garbageFilledStatus = "Garbage Lid is not closed" ;
        Serial.println(garbageFilledStatus);
        group6->save(garbageFilledStatus);
      }
    }
}

float readUltrasonicvalue(){
    RangeInCentimeters = ultrasonic.MeasureInCentimeters();
    Serial.print("Ultrasonic Distance -  "); Serial.print(RangeInCentimeters); Serial.println(" cm");
    M5.Lcd.setCursor(0, 40);  M5.Lcd.print(" Ultrasonic Sensor Value(in cm) - ");
    M5.Lcd.setCursor(0, 50);  M5.Lcd.printf(" %ld " ,RangeInCentimeters);
    return float(RangeInCentimeters);
}
```