Project Report on

Smart Garbage Monitoring System using Internet of Things (IOT)”

Submitted by

BRADULA SHETTY	B3-42
SHIVANI SHETTY	B3-44
SOMESH SHINDE	B3-45
JYOTI SOME	B3-55

Submitted to:

Mrs. Vidya Kubde

Subject In-charge/ Practical Incharge

(Sensor Lab)

Department of IT Engineering

Datta Meghe College of Engineering,

Airoli, Navi Mumbai – 400708

[2021 - 2022]

<u>ABSTRACT</u>:

This project IOT Based Garbage Monitoring System is a very smart system which will help to keep our village and cities. We see that in our cities public dustbins are overloaded and it create unhygienic conditions for people and that place leaving a bad smell. To avoid all these things we are going to implement a project Smart Garbage Monitoring System By Using IOT. These dustbins are interfaced with NodeMCU system having ultrasonic sensor along with central system showing the Current status of garbage as well as humidity & temperature on blynk app. To increase the cleanness in the country government started the various project. This project is helpful for government project of "SWACHH BHARAT ABHIYAN".

INTRODUCTION:

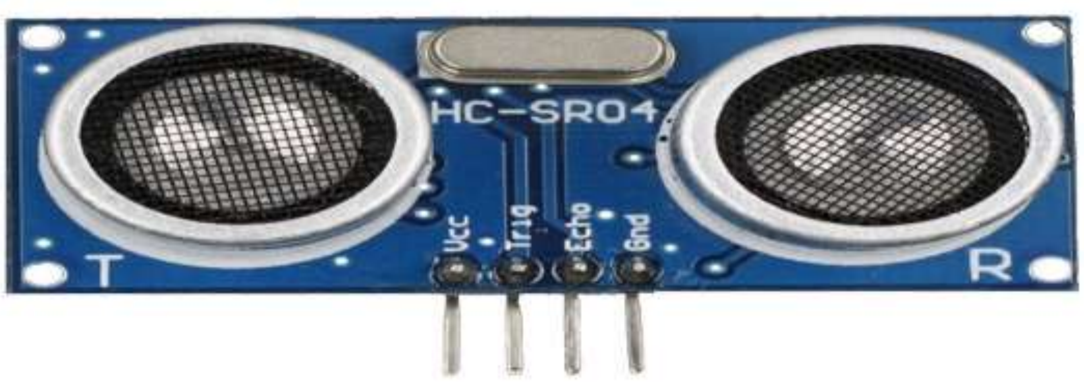
We are living in an age where tasks and systems are fusing together with the power of IOT to have a more efficient system of working and to execute jobs quickly! With all the power at our finger tips this is what we have come up with.The Internet of Things (IoT) shall be able to incorporate transparently and seamlessly a large number of different systems, while providing data for millions of people to use and capitalize. Building a general architecture for the IoT is hence a very complex task, mainly because of the extremely large variety of devices, link layer technologies, and services that may be involved in such a system. One of the main concerns with our environment has been solid waste management which impacts the health and environment of our society. The detection, monitoring and management of wastes is one of the primary problems of the present era. The traditional way of manually monitoring the wastes in waste bins is a cumbersome process and utilizes more human effort, time and cost which can easily be avoided with our present technologies. This is our solution, a method in which waste management is automated. This is our IoT Garbage Monitoring system, an innovative way that will help to keep the cities clean and healthy.

COMPONENTS:

NodeMCU

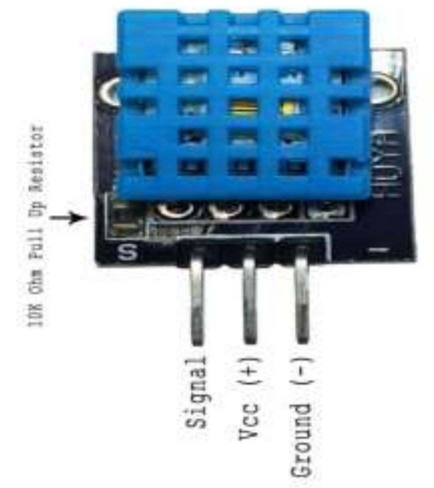
Nodemcu – ESP8266

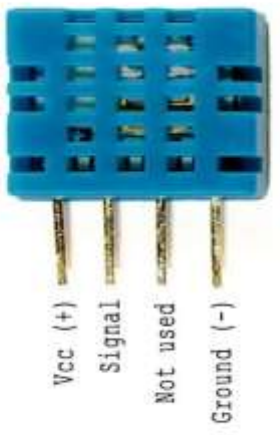
HC-SR04 ULTRASONIC SENSOR



HC-SR04 (Ultrasonic sensor)

ULTRASONIC SENSOR WORKING :





DHT11 (Temperature & Humidity Sensor)

LITERATURE SURVEY:

RFID technology is used for collection of data regarding garbage container. RFID tag detected within the frequency range and when any tag comes to the range of RFID reader, it automatically reads data from RFID reader, then filters collected data and arranges it into specific formatted SMS. After that, the data is sent to central server sends the information to the web server as well as authorized person's mobile phone. This Paper proposed a method as follows. The level of garbage in the bin is detected by using the ultrasonic sensor and communicates to control room using GSM system. Four IR sensors are used to detect the level of the garbage bin. When the bin is full

The output of the fourth IR is active low and this output is given to microcontroller to send a message to control room through GSM In this paper ZigBee, GSM and ARM7 controller is used to monitor the garbage bin level. When garbage bin is full, this message of garbage level is sentto ARM7 controller. Then ARM7 will send the SMS through GSM to authority as to which bin is overflowing and requires cleaning up.

The paper proposed method as. Ultrasonic sensors are used to sense the level of bin and load cell are used as a secondary sensor. If the level sensors are failing then load cell can be used as a reference. When the bin is full GSM send the message to the server room. This message contains the coordinate of the bin which is provided by GPS module. The microcontroller receives the input from GSM and performs signal processing. Microcontroller communicates to GSM by using UART.

In the paper the system is designed in such a way that it avoids overflow of the bin by sending an alert. It uses Arduino Uno R3 as a microcontroller for reading data from sensors.This technology mainly uses RFID reader which is interfaced with a microcontroller for the verification process.When RFID tag interrupts the RFID reader, the sensor will check the status of the bin and send it to the web server.

Optimal Route:

To solving optimization problem is to Information about filling garbage containers handled automatically by the system during the development of an optimal route. A GSM module is used to communicate with server room. When the garbage containers about to overflow, with the help of GSM module, an alert will be sent. The GPS module will help to identify the location of the garbage container. The alert signal will also contain the coordinates of garbage container which will be provided by GPS module. In this study, garbage collection routes optimization using Geographical Information System (GIS-Arc View) was studied. Many routes were selected in the different area of the city and the present routes were optimized to reduce the length of the routes and consequently the time is taken to complete the collection. We are using Bing

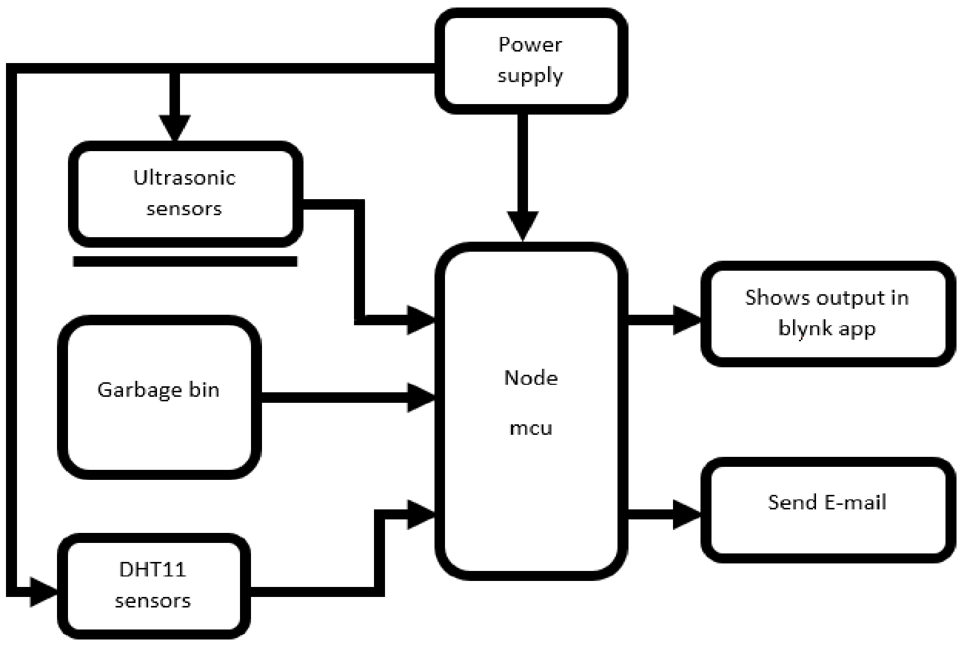
Maps API to show the geographical location of the garbage container. Currently, the GPS techniques to automatically detect the location of the garbage container have not been used. The Admin can also get the shortest path comprising of all the filled garbage container by clicking on the 'Show Path' button and he can redirect the garbage collection vans along that path. For finding the shortest path Bing Maps Routing Engine has been used. GSM & GPS. As we go through the introduction in next section we will discuss the proposed system in sub section

PROPOSED SOLUTION:

This System monitors the garbage bin and informs the level of garbage bins collection how many garbage in the garbage bin. The system uses ultrasonic sensor placed over the bins to detect the garbage level and compare it with the garbage depth. If garbage level is 90% or less than 90% then it's ok. But if garbage level is above 90% their Arduino gives information above bin level to server ESP8266 01 module. A Server is used to store data and shows of all dustbins level on the blynk app. App shows temperature and humidity of garbage. If garbage level is more than 90% app will send mail to appropriate email-id of Waste Management system of Municipal Corporation.
NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS.
NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, EspressifSystems[6] began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a TensilicaXtensa LX106 core,[citation needed] widely used in IoT applications (see related projects). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.
In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the NodeMCU included more than 40 different modules. Due to resource constraints users need to select the modules relevant for their project and build a firmware tailored to their needs.
Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likesover the Internet. It's a digital dashboard where you can build a graphic interface for yourproject by simply dragging and dropping widgets.Blynk is not tied to some specific board orshield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Piis linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get youonline and ready for the Internet Of Your Things. Blynk was
designed for the Internet of Things.It can control hardware remotely, it can display sensor data, it can store data, visualize it and domany other cool things.

There are three major components in the platform:
Blynk App - allows to you create amazing interfaces for your projects using variouswidgets we provide.
Blynk Server - responsible for all the communications between the smart phone andhardware. You can use our Blynk Cloud or run your private Blynk server locally. It'sopen-source, could easily handle thousands of devices and can even be launched on aRaspberry Pi.
Blynk Libraries - for all the popular hardware platforms - enable communication withthe server and process all the incoming and out coming commands.

BLOCK DIAGRAM:



WORKING:

Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likesover the Internet. It’s a digital dashboard where you can build a graphic interface for yourproject by simply dragging and dropping widgets.Blynk is not tied to some specific board orshield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Piis linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get youonline and ready for the Internet Of Your Things. Blynk was designed for the Internet of Things.It can control hardware remotely, it can display sensor data, it can store data, visualize it and domany other cool things.

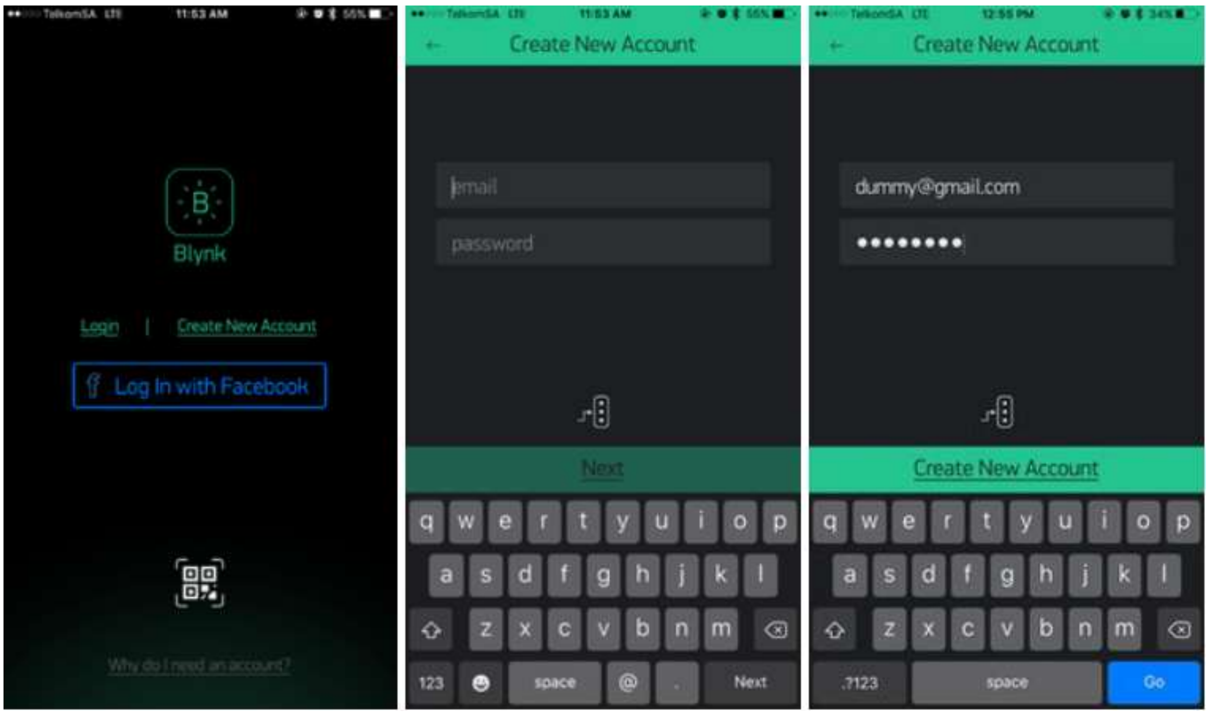
There are three major components in the platform:

Blynk App - allows to you create amazing interfaces for your projects using variouswidgets we provide.
Blynk Server - responsible for all the communications between the smart phone andhardware. You can use our Blynk Cloud or run your private Blynk server locally. It’sopen-source, could easily handle thousands of devices and can even be launched on aRaspberry Pi.
Blynk Libraries - for all the popular hardware platforms - enable communication withthe server and process all the incoming and out coming commands.

Creating a Blynk Account

Select “Create New Account”

Complete the email field and set a secure password in the password field (as shown below):

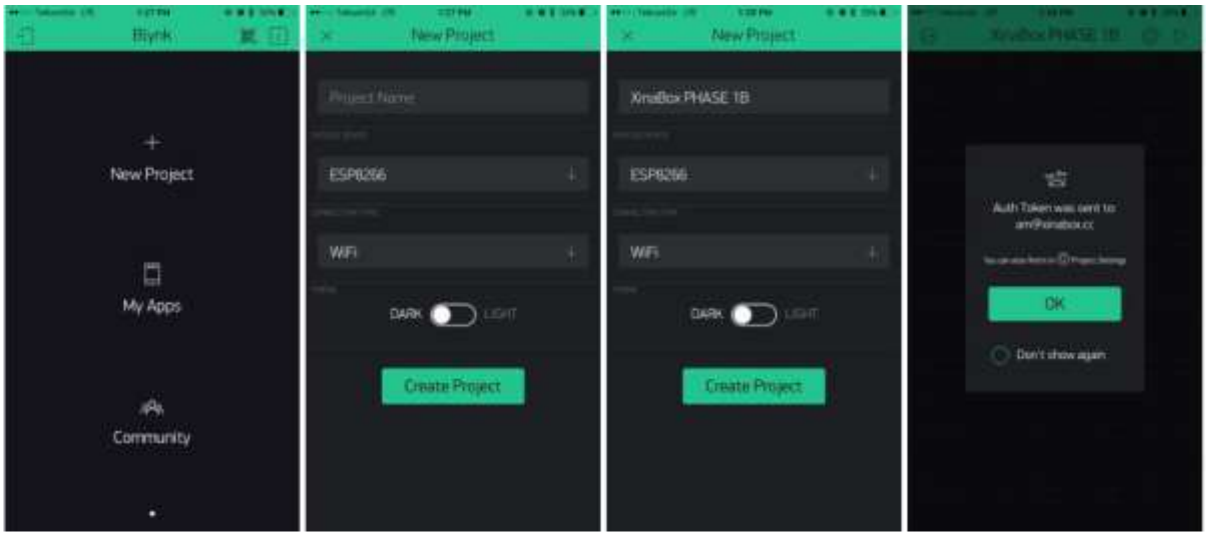


F:Blynk app sign up

Provisioning for Blynk in the xFlasher
Preparing for the Captive Portal Blynk Setup

After creating an account, select "New Project"
Complete the Project Name field
Select “Create Project”

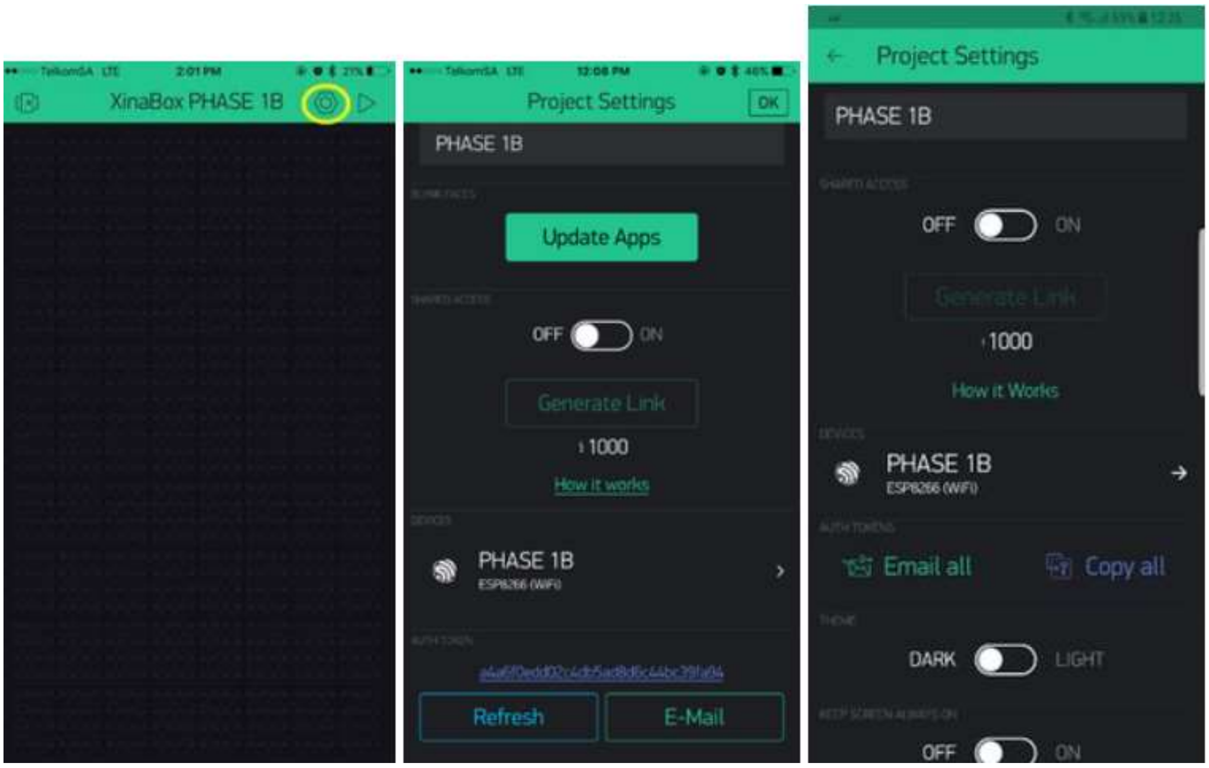
1. A pop-up notification of an authentication token for the X in a box captive portal sent to your e-mail address should appear (as shown below):



Blynk app create new project

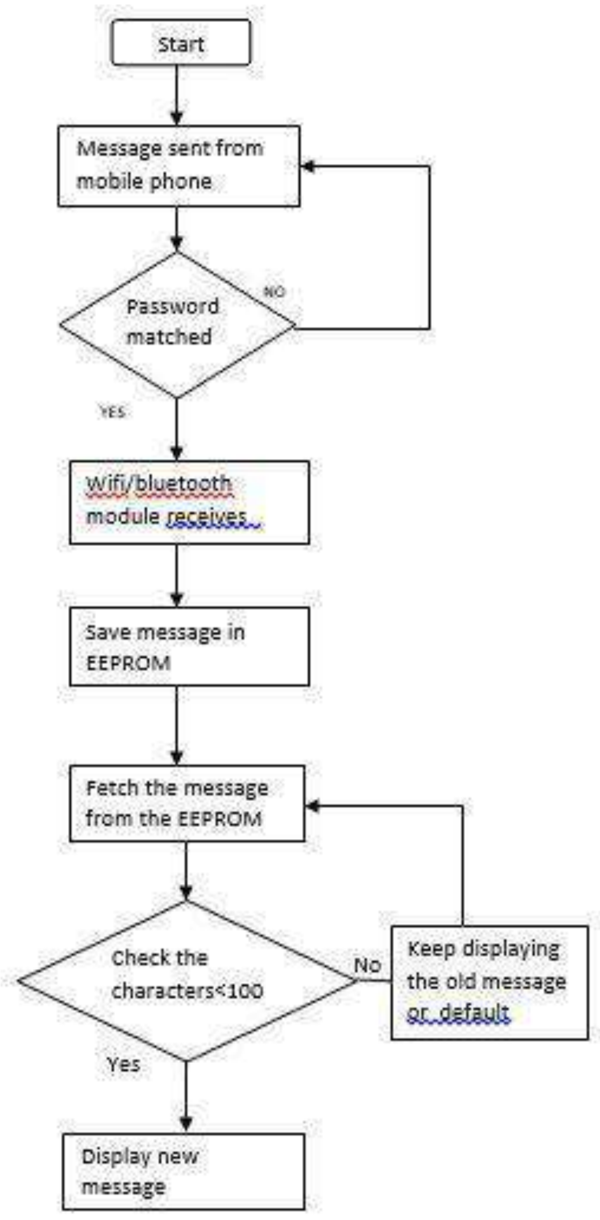
Copying the Blynk Authentication Token

- To reach the Auth Token select the icon in your new project window (marked with a yellow ring below left)
- The copy function between the Blynk IOS app and Android app differs slightly
- **IOS** (below center) requires (below center) one to select and hold the token code (in blue) until it "copies to clipboard"
- **Andriod** (below right) requires one to simply select the "**Copy all**" option

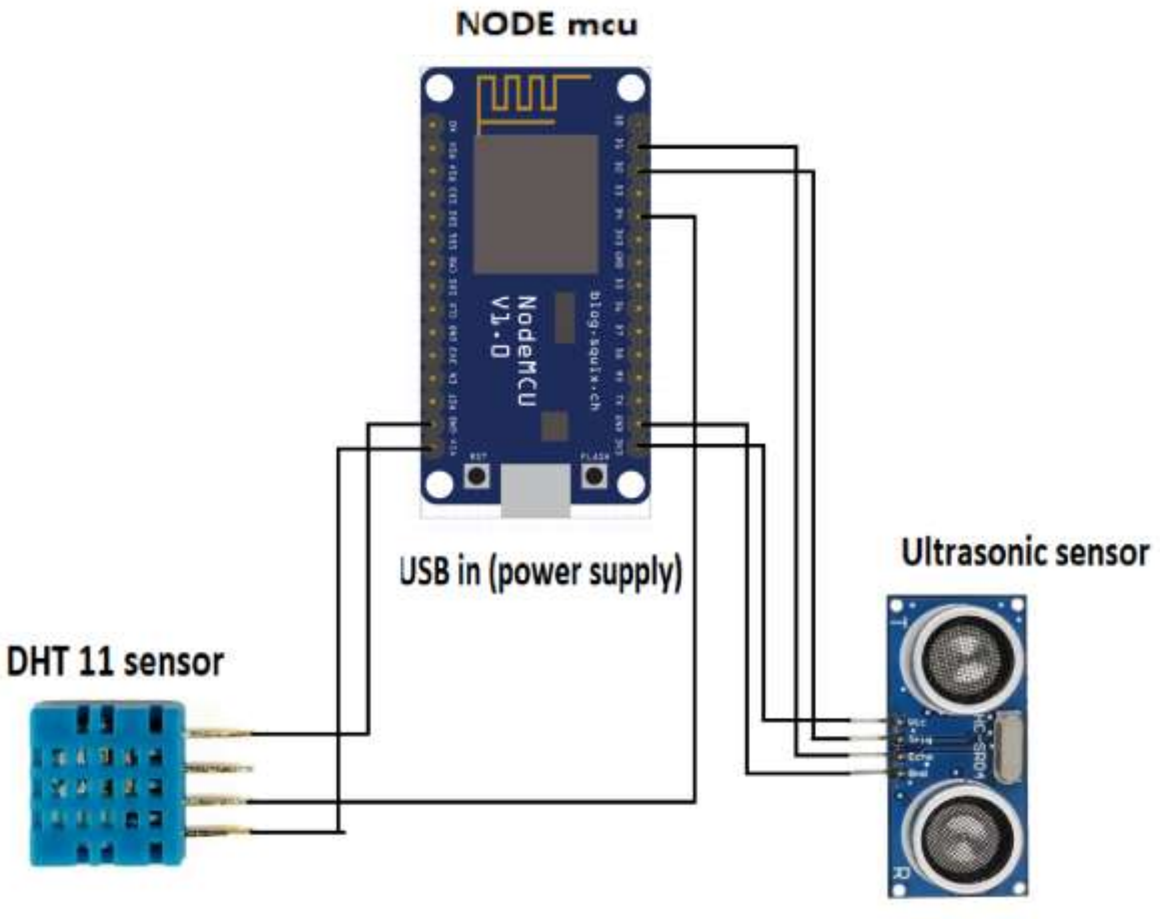


Blynk app add components

FLOW CHART:


Start
Message sent from mobile phone
Password matched
NO
YES
Wifi/bluetooth module receives
Save message in EEPROM
Fetch the message from the EEPROM
Check the characters<100
No
Keep displaying the old message or default
Yes
Display new message

CIRCUIT DIAGRAM:



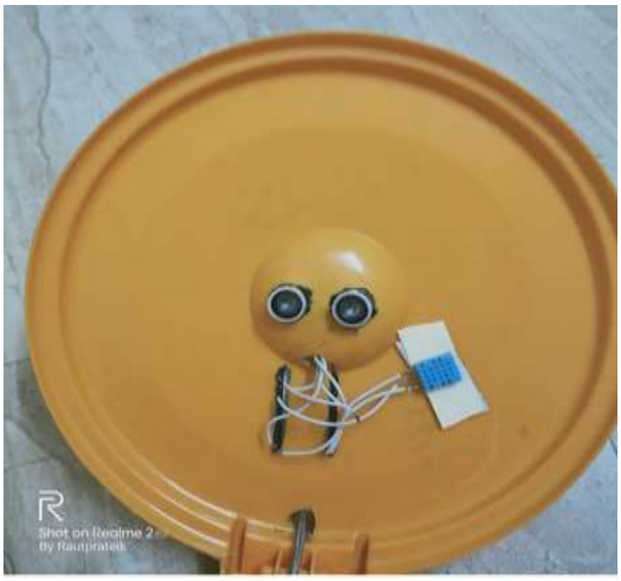
CODE:

```
/*
*****************************************
14CORE ULTRASONIC DISTANCE SENSOR CODE TEST
*****************************************
*/
#define TRIGGER 4
#define ECHO    5
// NodeMCU Pin D1 > TRIGGER | Pin D2 > ECHO
#define BLYNK_PRINT Serial    // Comment this out to disable prints and save
space
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SimpleTimer.h>
#include <DHT.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).


charauth[] = "3054e76b900b4bb68141020213e856ee";
// Your WiFi credentials.
// Set password to "" for open networks.
charssid[] = "POP";
char pass[] = "pop12345";
#define DHTPIN 2          // Digital pin 4
// Uncomment whatever type you're using!
#define DHTTYPE DHT11     // DHT 11
//#define DHTTYPE DHT22   // DHT 22, AM2302, AM2321
//#define DHTTYPE DHT21   // DHT 21, AM2301
DHT dht(DHTPIN, DHTTYPE);
SimpleTimer timer;
// This function sends Arduino's up time every second to Virtual Pin (5).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
voidsendSensor()
{
float h = dht.readHumidity();
float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
if (isnan(h) || isnan(t)) {
Serial.println("Failed to read from DHT sensor!");
return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
Blynk.virtualWrite(V6, h);  //V5 is for Humidity
Blynk.virtualWrite(V7, t);  //V6 is for Temperature
}
void setup() {
Serial.begin (9600);
Blynk.begin(auth, ssid, pass);
pinMode(TRIGGER, OUTPUT);
pinMode(ECHO, INPUT);
pinMode(BUILTIN_LED, OUTPUT);
dht.begin();
  // Setup a function to be called every second


timer.setInterval(1000L, sendSensor);
}
void loop() {
long duration, distance;
digitalWrite(TRIGGER, LOW);
delayMicroseconds(2);
digitalWrite(TRIGGER, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER, LOW);
duration = pulseIn(ECHO, HIGH);
distance = (duration/2) / 29.1;
if (distance <= 20) {
Blynk.virtualWrite(V0, 255);
}
else {
Blynk.virtualWrite(V0, 0);
  }
if (distance <= 15) {
```

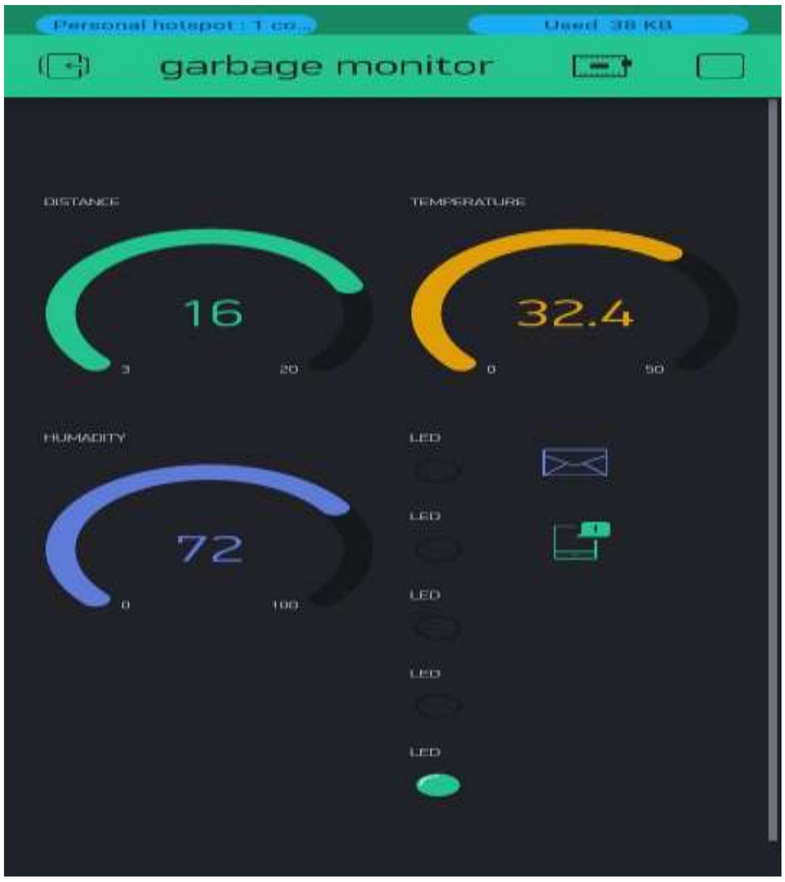
```
Blynk.virtualWrite(V1, 255);
}
else {
Blynk.virtualWrite(V1, 0);
  }
if (distance <= 10) {
Blynk.virtualWrite(V2, 255);
}
else {
Blynk.virtualWrite(V2, 0);
  }
if (distance <= 6) {
Blynk.virtualWrite(V3, 255);
}
else {
Blynk.virtualWrite(V3, 0);
  }
if (distance <= 3) {


Blynk.virtualWrite(V4, 255);
Blynk.email("pratikraut6296@gmail.com", "ESP8266 Alert", "Dust bin ID:001
is full. Please clean it as soon as possible..");
Blynk.notify("ESP8266 Alert - Dust bin is full");
}
else {
Blynk.virtualWrite(V4, 0);
  }
Serial.print(distance);
Serial.println("Centimeter:");
Blynk.virtualWrite(V5, distance);
delay(200);
Blynk.run();
timer.run(); // Initiates SimpleTimer
```

OUTPUT:

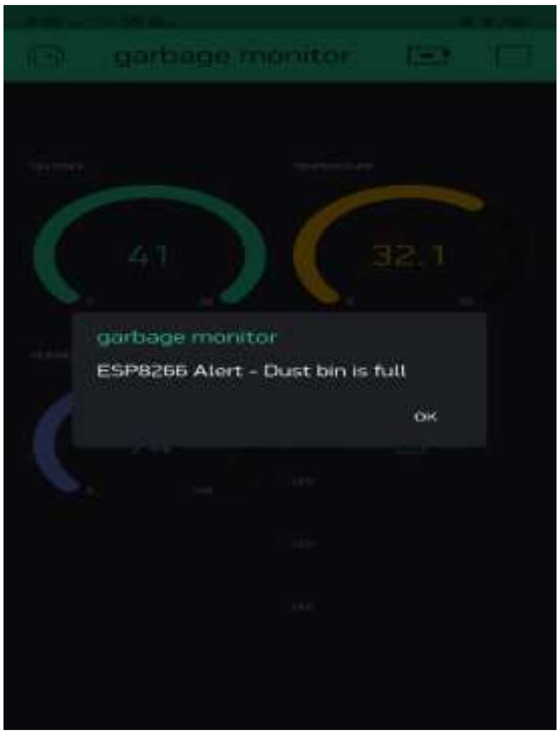
Dust bin



Dust bin with sensors



Blynk app output



garbage monitor

ESP8266 Alert - Dust bin is full

FUTURE SCOPE:

1. This project can also be used in the” SMART CITY”.
2. This project is also helpful in the government project of “SWACHH BHARAT ABHIYAN”.
3. Domestic
4. Hotels
5. Homes
6. Malls
7. Railway Station
8. Bus Stop
9. Gardens
10. Colleges

CONCLUSION:

In this project, an integrated system of Wi-Fi modem, IoT, Ultrasonic Sensor is introduced for efficient and economic garbage collection. The developed system provides improved database for garbage collection time and waste amount at each location. We analysed the solutions currently available for the implementation of IoT. By implementing this project we will avoid over flowing of garbage from the container in residential area which is previously either loaded manually or with the help of loaders in traditional trucks. It can automatically monitor the garbage level & send the information toMunicipal Corporation. The technologies which are used in the proposed system are good enough to ensure the practical and perfect for solid garbage collection process monitoring and management for green environment.

REFERENCES:

1. https://www.nodemcu.com/index_en.html#fr_54745c8bd775ef4b99000011
2. https://www.arduino.cc/
3. https://www.viralsciencecreativity.com/tutorials
4. https://blynk.io/
5. https://github.com/
6. https://en.wikipedia.org/