IOT BASED GARBAGE MONITORING SYSTEM

A project submitted to the Bharathidasan University
in partial fulfillment of the requirements
for the award of the Degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted by

SASIKUMAR A
Register Number: 185113154

Under the guidance of
Mrs. M.RACHEAL BETTY SUGUMARI,M.Sc.IT.,M.Phil.,
Assistant Professor

UG DEPARTMENT OF COMPUTER APPLICATIONS
BISHOP HEBER COLLEGE (AUTONOMOUS)
(Nationally Reaccredited at the 'A' Grade by NAAC with the CGPA of 3.58 out of 4)
(Recognized by UGC as "College of Excellence")
(Affiliated to Bharathidasan University)

TIRUCHIRAPPALLI-620 017

APRIL – 2021

DECLARATION

I hereby declare that the project work presented is originally done by me under the guidance of **Mrs. M.RACHEAL BETTY SUGUMARI,M.Sc.IT.,M.Phil.,, Assistant Professor, Department of Computer Applications, Bishop Heber College (Autonomous), Tiruchirappalli-620 017** and has not been included in any other thesis/project submitted for any other degree.

Name of the Candidate	**:**	**SASIKUMAR A**
Register Number	**:**	**185113154**
Batch	**:**	**2018-2021**

Signature of the Candidate

Mrs. M.RACHEAL BETTY SUGUMARI,M.Sc.IT.,M.Phil.,,

Assistant Professor,

Department of Computer Applications,

Bishop Heber College (Autonomous),

Tiruchirappalli – 620017

Date:

CERTIFICATE

This is to certify that the project work entitled **"GARBAGE MONITORING SYSTEM"** is a bonafide record work done by **SASIKUMAR A, Register Number: 185113154** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF COMPUTER APPLICATIONS** during the period **2018 - 2021.**

Place: Trichy **Signature of the Guide**

UG DEPARTMENT OF COMPUTER APPLICATIONS

BISHOP HEBER COLLEGE (AUTONOMOUS),

(Nationally Reaccredited at the 'A' Grade by NAAC with the CGPA of 3.58 out of 4)
(Recognized by UGC as "College of Excellence")
(Affiliated to Bharathidasan University)

TIRUCHIRAPPALLI - 620017

Date:

Course Title: Project **Course Code: U18CA6PJ**

CERTIFICATE

The Viva-Voce examination for the candidate **SASIKUMAR A (185113154)** was held on _____.

Signature of the Guide **Signature of the HOD**

Examiners:

1.

2.

CONTENT

S.NO	CHAPTERS	PAGE NO
1	**PROJECT DESCRIPTION**	
	1.1 Introduction	**1**
	1.2 Existing System	**2**
	1.3 Proposed System	**2**
	1.3.1 Hardware specifications	**3**
	1.3.2 Software specifications	**3**
2	**LOGIAL DEVELEOPMENT**	
	2.1 Flow Diagram	**4**
	2.2 Architectural Diagram	**5**
3	**LITERATURE SURVEY**	**6**
4	**PROGRAM DESIGN**	
	4.1 Modules	**8**
	4.2 Module Description	**9**
5	**TESTING**	
	5.1 System Testing	**26**
	5.1.1 Test Approach	**26**
6	**CONCLUSION**	**31**
7	**REFERENCES**	**33**
8	**APPENDIX**	
	8.1 Source code	**34**
	8.2 Output Screens	**42**

ACKNOWLEDGEMENT

I take this opening to thank the people who have assisted me in this project work. I would like to express my first and foremost gratitude to this “Almighty”, for the merciful shower of grace and blessing in the entire endeavor to bring the project successful. Then my parents, who were pillars and effort along of my each and every activity.

I wish to express my sincere thanks to **Dr. D. PAUL DAYABARAN, M.Sc., M.Phil., PGDCSA, Ph.D.,** Principal, Bishop Heber College, Tiruchirappalli who gave me the opportunity to do the course in this college.

I would like to express my sincere thanks and profound sense of gratitude to **Dr. J. JAMES MANOHARAN, MCA., M.Phil., Ph.D., PGDCA.,** Associate professor and Head, UG Department of Computer Applications, Bishop Heber College, Tiruchirappalli.

I express my immense gratitude and respects to my guide **Mrs. J. OLIVIYA CAPITOLA M.C.A., M. Phil.,** Associate professor, who gave valuable guidance and excellent suggestions in improving the quality of the project.

I would like to thank all the staff members of the UG Department of Computer Applications, who put their effort to shape me. I will be failing in my duty if I don’t thank my friends who stood with me in each and every activity as a great wall by extending their excellent support.

Finally, I would like to thank everybody who was important to the successful realization of my project, as well as expressing my apology that I could not mention personally one by one.

SASIKUMAR.A

ABSTRACT

The Project entitled as **"GARBAGE MONITORING SYSTEM"** In the present scenario, one of the main problems that our country is facing is that heaps and bounds of waste is getting segregated day by day. We see daily that our garbage bins are overloaded with trash which creates many problems and leads to unhygienic conditions. The waste from the bins and its foul smell often given invitation to a number of communicable diseases like Dengue and many other related diseases. The detection monitoring and efficient management of waste thus becomes the primary concerns of the present era. The traditional way of monitoring the waste in waste bins is a complex, difficult and tedious process and takes more human effort and time and cost which is not compatible with the present technologies in any way.

"**The project IOT Garbage monitoring system** hence comes to the picture which is very unique system to keep the mounding amount of waste under check and tackle the growing garbage in a more sustainable and time saving manner. This is an advanced method in which waste management has been made automated. This system monitors the garbage bins by measuring the depth of bins using ultrasonic sensors and sending that information to the microcontroller which in turn processes that information and displays the level of waste collected in the bins using a Mobile application. This webpage also sends all the information to garbage collection vehicles driver will go to that bin only in which the level of garbage has cleared the threshold level".

PROJECT DESCRIPTION

1. PROJECT DESCRIPTION

1.1 INTRODUCTION

Pollution is the spread of contaminants into an environment that causes instability, disorder, harm or discomfort to the environment. Solid waste management is one of the major environmental problems of India. Solid waste management is the collection, transport, disposal, managing and monitoring of waste material. Garbage may consist of the municipal solid waste construction waste, commercial Garbage may consist of the municipal solid waste construction waste, commercial waste ,industrial waste etc… left over the city. This project is useful for creating "Smart City" and it is based on "Internet of Things". For healthy lifestyle cleanliness is needed and it begins with the use of rash bins. This project will help to eradicate or minimize the solid waste disposal problem. In present scenario, many times we see the garbage bins gets overloaded due to increase in solid waste everyday.

It creates unhygienic environment and bad smell in the society and because of this many disease get spread in the society to avoid this situation we are designing "Garbage monitoring system using Internet of Things" In this proposed system the multiple trash bins are located throughout the city, these trash bins are embedded with low cost embedded device. When the dustbin gets half filled that is when the threshold value become 50% then the corporation will get notification and when the garbage level will reach the threshold value 80% then the notification will get half filled. The proposed system is cost effective because it will notify twice to the organization and they will get time to optimize the cost of transportation.

1.2 EXISTING SYSTEM

There is no existing framework to monitor the dustbin system on an entire day. Existing system does not have provision to send messages and alert municipal when the bins are filled. Hence the bins over-fill and contaminate the whole area. Existing system contains an ultrasonic sensor which gets damaged during rainy reason. It also does not cover in all directions rather than in a straight line. So in the proposed system we use the sensor inside the dustbin to measure garbage level.

1.3 PROPOSED SYSTEM

Our proposed work in a smart garbage monitoring system where the lid will be opened only when a person operates when needed and does not open and close automatically. So it reduces mosquitoes growing on rainwater thus reducing disease growth. The thermal sensor is used to detect humans or any living things interrupting the dustbin. Thus it helps us to choose bin which is filled rather than to search for all of them..

1.3.1 HARDWARE REQUIREMENTS

- Processor : Intel processor 3.0 GHz
- RAM : 4GB
- Hard disk : 500 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Mouse : Logitech mouse
- Monitor : 15 inch color monitor

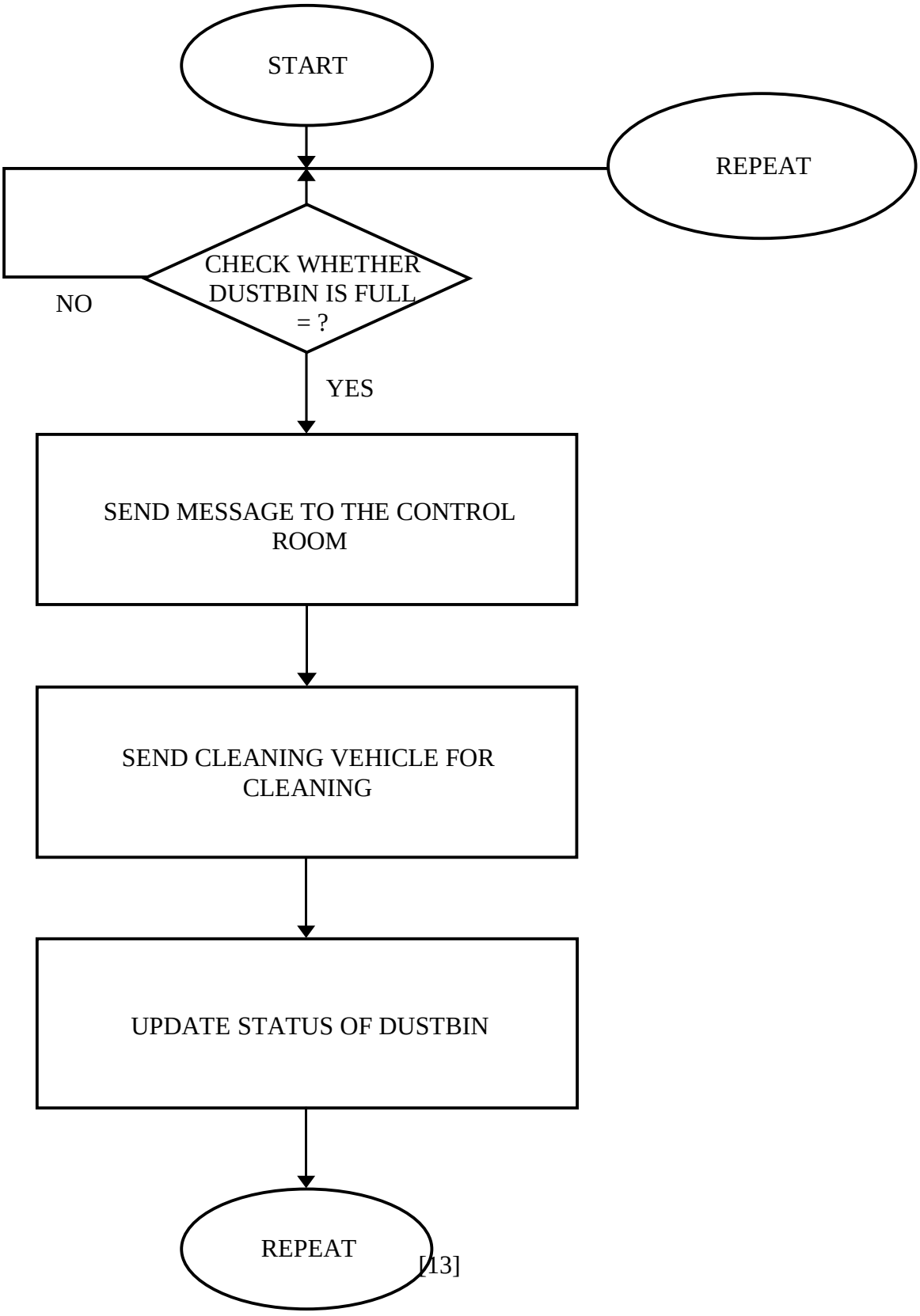
1.3.2 SOFTWARE REQUIREMENTS

- Operating System : Windows OS
- System type : 32-bit or 64-bit Operating System
- DLL : Depends upon the title

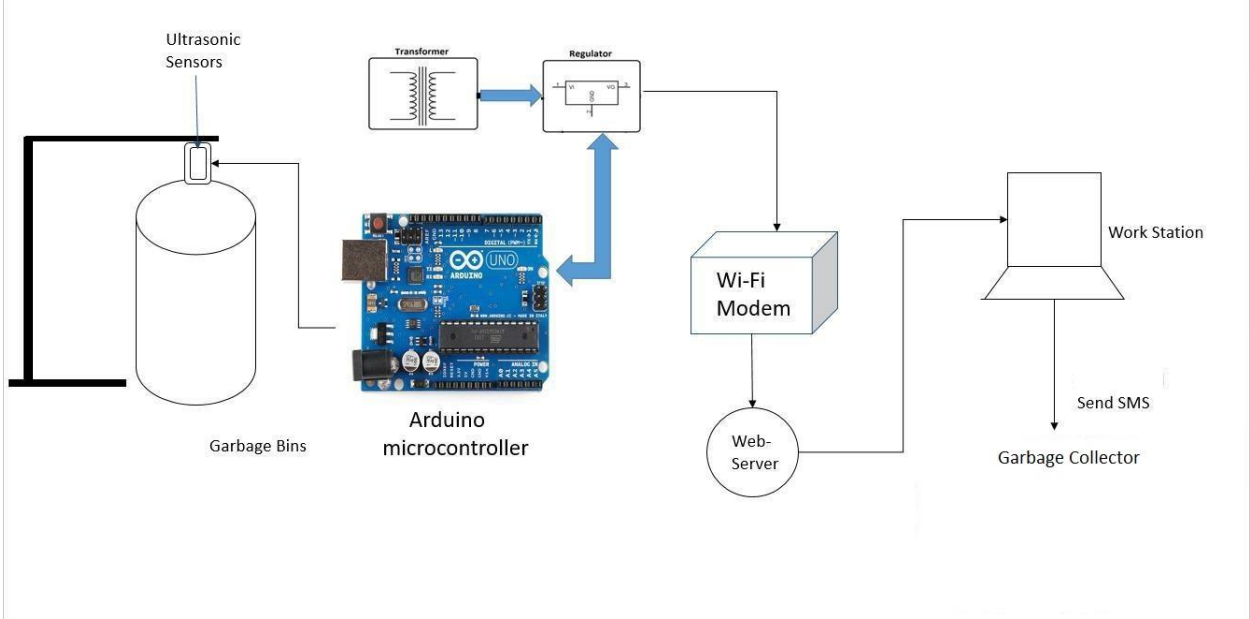
LOGICAL DEVELOPEMENT

2. LOGICAL DEVELOPMENT

2.1 DATA FLOW DIAGRAM



2.2 ARCHETECTURAL DIAGRAM



LITERATURE SURVEY

3.LITERATURE SURVEY

This is not an original idea, for the implementation of smart dustbins ; the idea has existed for many years, after the IoT field finding its grip in our lives. This is, however an original plan for designing a smart garbage bin with Ultrasonic sensor , metal detector, microcontroller (16F877A), xbee-rx (an XBee reactive extensions API) and MAX232 for transmission of data to the server. The ultrasonic sensor will be used to detect the level of garbage in dustbin. Metal detector will be used to detect the opening and closing of lid We have set two threshold values 50% and 80%.when the garbage level reach 50% and 80% the notification will get display on the web page Accordingly information is processed that is controller checks if the threshold level is exceeded or not, and this information will transfer to the server by using Zigbee . By using garbage monitoring system the information of all smart dustbins can be accessed from anywhere and anytime by the concern person and he/she can take the decision accordingly.

Some of the following garbage type packaging waste, Agricultural waste, Inorganic waste, Liquid waste etc. In solid waste bin monitoring system garbage bin set the public place then Camera set for garbage bin location. The camera captured image for garbage bin. Radio Frequency Identification (RFID), GPS and GIS send image for work station. The RFID reader and camera are mounted in the truck, when truck come closer to the bin RFID reader communicated RFID tag. The System are use controlling Hut. This Controlling Hut are SMS Technology. The GPS and GPRS mapping server to analyzing data of various location. The Control station compiled all the information and stored in the system database. The bin status and waste truck was monitored.

In waste bin monitoring system using zig bee and Global mobile communication system(GSM).The sensors are place in the common garbage bins placed at the public place when the garbage reaches the level of the sensors. Then that indicated will give in indication to the driver by ARM7 they sending SMS using GSM technology. The technology use by Zig bee, Global mobile system (GSM), ARM 7 Controller. The range of communication of the zig bee is almost 50 meter. They use for range GSM Module, analyzing the image we get an idea about level of garbage. The zig bee and GSM system would be able to monitor the solid waste collection process. This technique overcome some disadvantages which are use of minimum route, low cost, fuel use, clean environment. [2] The waste management is built around several element. Waste item, the waste item and the domestic bin to end in the collecting vehicles Use the waste identification for sorting process. Base on RFID technology new trash bag is added in a collective container. The technology use Radio Frequency Identification (RFID), Smart vehicular and Trash Bag.

PROGRAM DESIGN

4. PROGRAM DESIGN

Program design is the process that an organization uses to develop a program. It is most often an iterative process involving research, consultation, initial design, testing and redesign.

4.1 MODULES

- ESP8266 BOARD
- ARDRUINO IDE
- ULTRA SONIC SENSOR
- BLYNK APP
- LED LIGHTS
- POWER SOURCE
- WIFI MODEM
- WORK STATION (PC OR MOBILE)
- ADAFRUIT MQTT
- IFTTT(If This Than That)

4.2 MODULES DESCRIPTION

4.2.1 ARDUINO IDE

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package. The Arduino is a microcontroller board based on the ATmega8. It has 14 digital – input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC to- DC adapter or battery to get started .The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2HWB line to ground, making it easier to put into DFU mode.

4.2.2 HC-SR04 ULTRASONIC SENSOR

HC-SR04 is an ultrasonic sensor which is used for measuring the distance between the top of the lid to the top of the garbage.

PIN NUMBER	PIN NAME	DESCRIPTION
1	VCC	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	GND	This pin is connected to the Ground of the system.

4.2.2.1 ULTRASONIC SENSOR WORKING

The HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module. Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

Figure 4.2.2 : HCSR04 ULTRASONIC SENSOR

4.2.3 NODE MCU

Node MCU is an open source IoT platform. It includes firmware which runs on the ESP8266Wi- Fi SoC from Express if Systems, and hardware which is based on the ESP-12 module. The term "Node MCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the Lua project, and built on the Express-if Non- OS SDK for ESP8266. It uses many open source projects, such as Lua- cjson, and spiffs.

Node MCU provides a way to connect different sensors to their controllers wirelessly via wi-fi. Since, it is an improved version of the ESP8266 it has better and easier programming, with better voltage stability and more reliability

Figure 4.2.3: Node MCU

Micro controllers have small internal memory which is not enough to save sensors generated data for long time, either you have to use some external memory device or can save the data on some cloud using internet.

Node MCU pin out is having labels D0 to D8 and RX-TX but when programming it using Arduino IDE we observe that its labels are not matching with IO connections. Lets see actual connections of Node MCU with ESP8266 i.e. ESP-12.

Node MCU is an open source IoT platform. It includes firmware which runs on the ESP8266Wi -Fi SoC from Express if Systems, and hardware which is based on the ESP-12 module. The term "**Node MCU**" by default refers to the firmware rather than the dev kits.

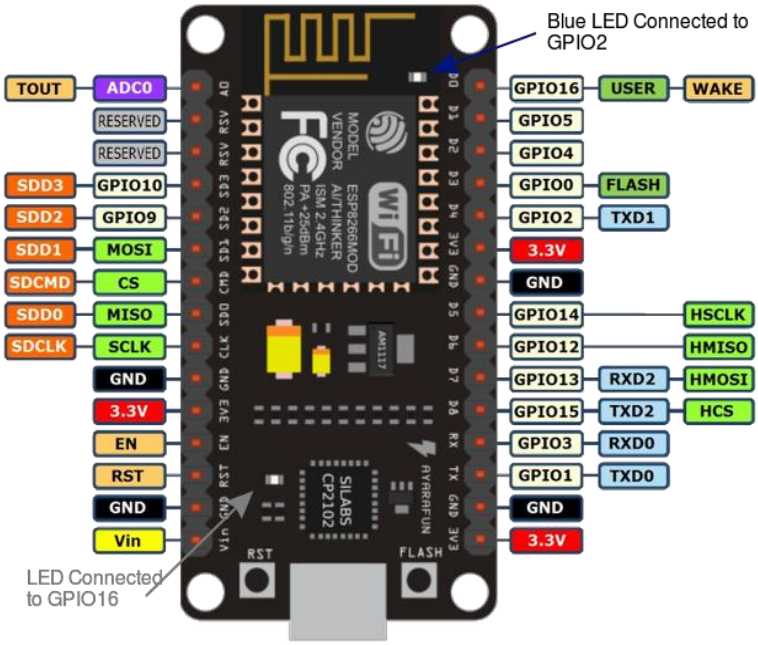


Figure:4.3.2 Node MCU with Parts

Node MCU has weird pin mappings. Pin numbers written on the board itself do not correspond to ESP8266 GPIO pin numbers. We have constants defined to make using this board easier: Node MCU Circuit Diagram

4.2.3.1 NODE MCU CIRCUIT CONNECTION

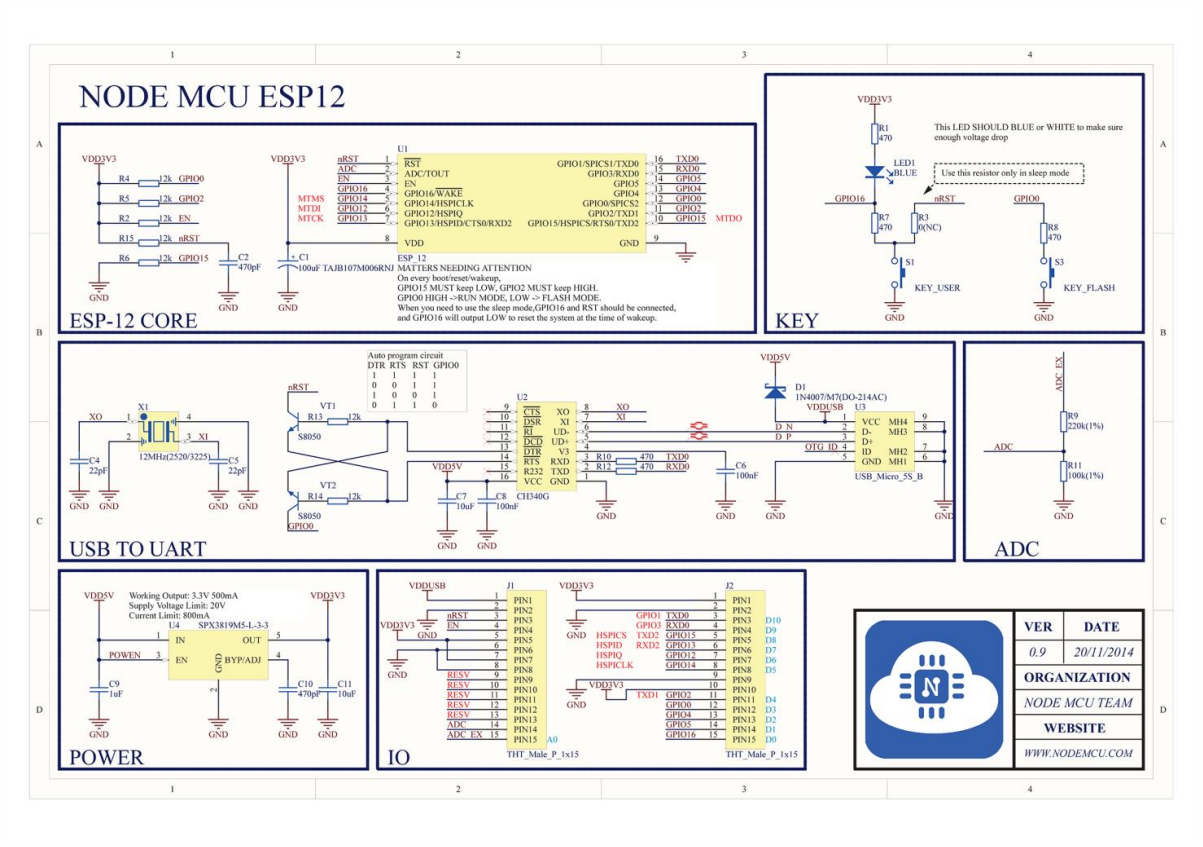


Figure:4.2.3.1 Node MCU Circuit Connection

Circuit diagram of Node MCU makes clearer about connections of Node MCU with ESP8266 GPIO. Never use GPIO0 (D3) as input pin, this pin is flash button. If it is low at power on state this will put ESP in programming mode

How to start with Node MCU?

- Node MCU Development board is featured with WIFI capability, analogue pin, digital pins and serial communication protocols.

- To get start with using Node MCU for IoT applications first we need to know about how to write/download Node MCU firmware in Node MCU Development Boards. And before that where this Node MCU firmware will get as per our requirement.

- There is online Node MCU custom builds available using which we can easily get our custom Node MCU firmware as per our requirement.

- To know more about how to build custom Node MCU firmware online and download it refer Getting started with Node MCU

Here is another way of developing Node MCU with a well-known IDE i.e. Arduino IDE. We can also develop applications on Node MCU using Arduino development environment. This makes easy for Arduino developers than learning new language and IDE for Node MCU.

Node MCU is an open-source firmware IoT platform. Node MCU is like an Arduino board. It contains 14 digital and 6 analog pins. It contains the wi-fi module. ESP-12 is used for hardware-based modulo. The firmware uses the Lua scripting language. It contains 17 GPIO pins. 6 pins used for flash.

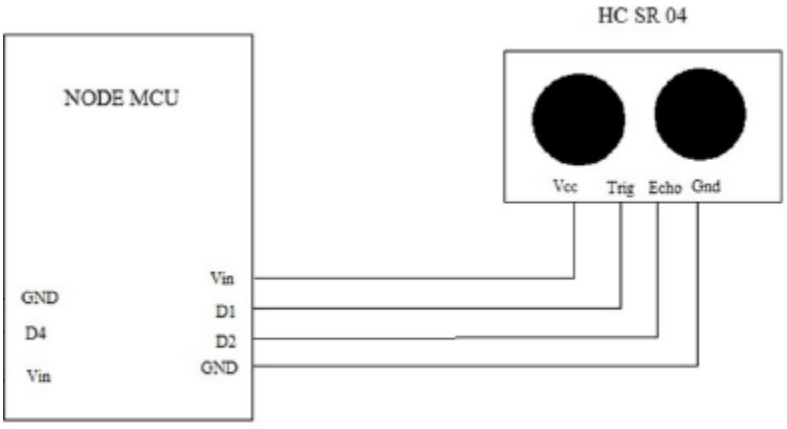


Figure 4.2.3: Node MCU with Ultrasonic Sensor

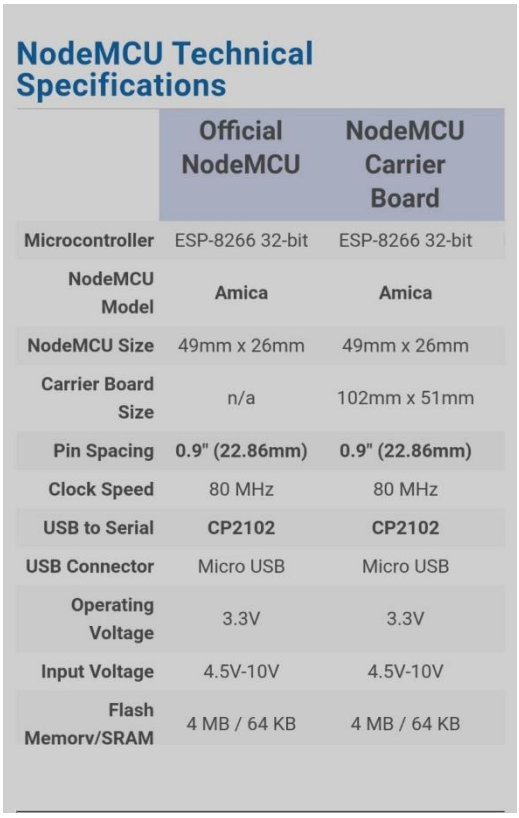
4.2.3.2 POWER SUPPLY

Power Supply Digital Power Supply ESP8266EX has two digital pins for power supply, pin11 and pin17. For digital power supply, there is no need to add additional filter capacitors. The operating voltage range of digital power supply pins is 1.8 V ~ 3.3 V.

4.2.3.3 POWER-ON SEQUENCE

Power-on Sequence and Power Reset Power-on Sequence ESP8266EX uses 3.3 V as the system power supply. It is not allowed that Pin7 CH_EN be powered on prior to that the 3.3 V system power supply is powered on. Pin32 serves as a RST pin which can be dangled when it is not used. The reset pin is held low level when the chip is enabled. In order to avoid reset caused by external interference, the lead is generally required to be short, and no external pull-up resistor is necessary. Pin7 CH_EN can also be used as a reset pin. When the voltage for CH_EN pin is low, the chipset will be powered off.

4.2.3.4 NODE MCU TECHNICAL SPECIFICATION



NodeMCU Technical Specifications

	Official NodeMCU	NodeMCU Carrier Board
Microcontroller	ESP-8266 32-bit	ESP-8266 32-bit
NodeMCU Model	Amica	Amica
NodeMCU Size	49mm x 26mm	49mm x 26mm
Carrier Board Size	n/a	102mm x 51mm
Pin Spacing	0.9" (22.86mm)	0.9" (22.86mm)
Clock Speed	80 MHz	80 MHz
USB to Serial	CP2102	CP2102
USB Connector	Micro USB	Micro USB
Operating Voltage	3.3V	3.3V
Input Voltage	4.5V-10V	4.5V-10V
Flash Memory/SRAM	4 MB / 64 KB	4 MB / 64 KB

Figure:4.2.3.4 Node MCU Technical Specifications

4.2.4 BLYNK APP

Blynk is an Internet of Things (IOT) server, Blynk is also a platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where a graphic interface can be built for the project by simply dragging and dropping widgets. It's really simple to set everything up and it will start tinkering in less than 5 minutes. Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of the choice. Whether Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get online and ready for the Internet of Things.

Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet .It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets .Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you online 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 do many other cool things

There are three major components in the platform:

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

4.2.5 ADAFRUIT MQTT

MQTT, or message queue telemetry transport, is a protocol for device communication that Adafruit IO supports. Using a MQTT library or client you can publish and subscribe to a feed to send and receive feed data.

If you aren't familiar with MQTT check out this introduction from the Hive MQ blog. All of the subsequent posts in the MQTT essentials series are great and worth reading too. To use the MQTT API that Adafruit IO exposes you'll need a MQTT client library. For Python, Node.js, and Arduino you can use Adafruit's IO client libraries as they include support for MQTT (see the client libraries section). For other languages or platforms look for a MQTT library that ideally supports the MQTT 3.1.1 protocol.

4.2.5.1 CONNECTION DETAILS

You will want to use the following details to connect a MQTT client to Adafruit IO:
Host: io.adafruit.com
Port: 1883 or 8883 (for SSL encrypted connection)
Username: your Adafruit account username (see the accounts.adafruit.com page here to find yours) **Password:** your Adafruit IO key (click the AIO Key button on a dashboard to find the key) We strongly recommend using SSL if your MQTT client allows it. If the MQTT library requires that you set a client ID then use a unique value like a random GUID. Most MQTT libraries handle setting the client ID to a random value automatically though.

4.2.5.2 TOPICS

Adafruit IO's MQTT API exposes feed data using special topics. You can publish a new value for a feed to its topic, or you can subscribe to a feed's topic to be notified when the feed has a new value. Any one of the following topic forms is valid for a feed:

(username)/feeds/(feed name or key)

(username)/f/(feed name or key)

Where (username) is your Adafruit IO username (the same as specified when connecting to the MQTT server) and (feed name or key) is the feed's name or key. The smaller '/f/' path is provided as a convenience for small embedded clients that need to save memory. Check out our guide to Feed Naming for the full details. For example if your username is mosfet and you're accessing a feed called Photocell One (which has a Key of photocell-one) you can use any of these paths:

mosfet/feeds/Photocell One
mosfet/f/Photocell One
mosfet/feeds/photocell-one
mosfet/f/photocell-one

To append a new value to a feed perform a MQTT publish against the feed path and provide the new feed value as the payload of the request. To be notified of a change in a feed perform a MQTT subscribe against the feed path. When a new value is added to the feed then the Adafruit IO MQTT server will send a notification with the new feed value in the payload. You can also subscribe to the parent 'feeds' path to be notified when any owned feed changes using MQTT's # wildcard character. For example the mosfet user could subscribe to either:

mosfet/feeds/#
mosfet/f/#

Once subscribed to the path above any change to a feed owned by mosfet will be sent to the MQTT client. The topic will specify the feed that was updated, and the payload will have the new value.

Be aware the MQTT server sends feed updates on all possible paths for a specific feed. For example, subscribing to mosfet/f/# and publishing to mosfet/f/photocell-one would produce messages from: mosfet/f/photocell-one, mosfet/f/photocell-one/json, and mosfet/f/photocell-

one/csv; each referring to the same updated value. To reduce noise, make sure to grab the specific topic of the feed / format you're interested in and change your subscription to that.

PLEASE NOTE: As we adjust which identifiers we use for Feeds internally, the feed updates you receive when using a wildcard will include but may not be limited to the ones shown above. If you'd like to avoid the formatted feeds ("/json" and "/csv" topics) but still see all the feeds you're publishing to, you can use MQTT's + wildcard in place of #. In this case, subscribing to mosfet/f/+ would produce output on mosfet/f/photocell-one, but not mosfet/f/photocell-one/json

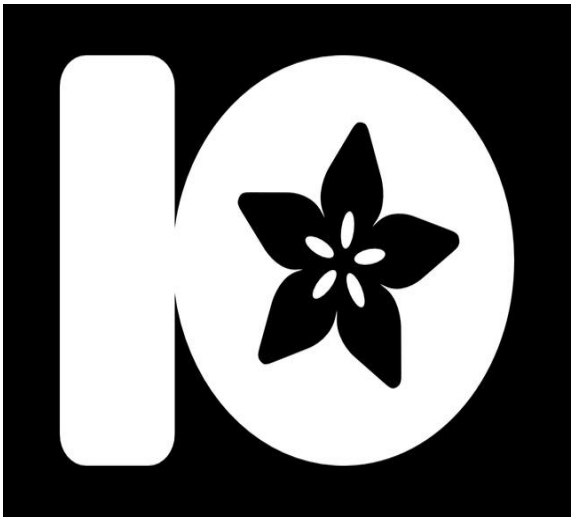


Figure 4.2.5 ADAFRUIT MQTT LOGO

4.2.6 IFTTT

IFTTT derives its name from the programming conditional statement “if this, then that.” What the company provides is a software platform that connects apps, devices and services from different developers in order to trigger one or more automations involving those apps, devices and services.

Here are just three if this, then that automations you can run with IFTTT:

- If you make a call on your Android phone, then a log of that call is added to a Google spreadsheet.
- If you add a new task to your Amazon Alexa to-dos, then it will be added to your iOS Reminders app.
- If the International Space Station passes over your house, then you'll get a smartphone notification about it. (Yes, this is an actual IFTTT applet.)

The automations are accomplished via applets — which are sort of like macros that connect multiple apps to run automated tasks. You can turn on or off an applet using IFTTT's website or mobile apps (and/or the mobile apps' IFTTT widgets). You can also create your own applets or make variations of existing ones via IFTTT's user-friendly, straightforward interface. IFTTT is simple to use. You download the mobile app (for Android here or for Apple's iOS here) create a free account and you're up and running with automations in minutes.

There are a bewildering array of applets available, so IFTTT helpfully provides automation recommendations for new users to try. Its Collections groups together applets for different platforms – such as iOS, Android and voice assistants – and showcases everything from applets for news and weather services to home automation.

An increasingly popular way to use IFTTT is in conjunction with Amazon's Alexa voice assistant. Much of these applets center around internet-of-things use cases such as controlling smart home devices with voice commands directed at Echo and Echo Dot speakers. This could entail telling Alexa to make a cup of coffee with WeMo's connected coffee maker or changing the color of Hue smart lights each time Alexa plays a new song.

IFTTT helps you connect all of your different apps and devices. When you sign up for a free account, you can enable your apps and devices to work together to do specific things they couldn't do otherwise. For example, you can back up your Instagram photos to Dropbox, have your lights turn on when you enter your home, or automatically remind a Slack channel about a meeting.

IFTTT is both a website and a mobile app. The free service launched in 2010 with the following slogan: "Put the Internet to work for you". It's changed a lot in recent years, however. Currently, with IFTTT, you can connect all your "services" together so that tasks are automatically completed. There are numerous ways you can connect all your services - and the resulting combinations are called "Applets".

Applets essentially automate your daily workflow, whether it's managing smart home devices or apps and websites. So, for instance, if you own the Philips Hue smart lighting system, you could use IFTTT to automatically turn on a light every time you're tagged in a Facebook photo. In another example, you could use IFTTT to automatically email readers when they comment on your WordPress blog.

4.2.6.1 APPLETS

Applets are specific things that can happen when you connect services. For instance, you could use an Applet to sync Amazon Alexa to-dos with your Google Calendar, or you could use one that lets you create events in your iPhone Calendar via Google Assistant. There are tonnes of Applets to choose from; IFTTT says there are 11 million users running more than a billion Applets a month.

To find already-created Applets on IFTTT, go to the IFTTT search page.
To manage all your IFTTT Applets, go here.
Note: Applets were once called "recipes."

How do you set up an Applet?
Here's an example Applet for Twitter and Instagram:

4.2.6.1 LOGIN TO IFTTT'S WEBSITE

- Go to your username and click services.

- You may see some auto-generated Applets based on your account information, such as your time zone and email address.
- Search for an Applet or a service you'd like to find a relevant Applet for, such as Instagram.
- Select one you want to use. For instance, there's an Applet that lets you tweet your Instagram as native photos on Twitter.
- Click the "Turn on" button to turn on the Applet.
- IFTTT will ask for permission to access both your Instagram and Twitter. Click OK and then authorize access to each service.
- Once done, you'll be brought back to the turned-on Applet on IFTTT.
- To view your Applets, select My Applets from the top of your IFTTT dashboard.

4.2.6.3 HOW DO YOU CREATE AN APPLET?

- Login to IFTTT's website.
- Go to your username and click New Applet.
- Select the +This in the "If This Then That" logo toward the center-top of the page. This will allow you to look for the "trigger" that will tell your Applet when to run.
- You can then search for and select a service. In this example, we'll use Twitter. You'll need to connect to it and authorize access to your account.
- You'll then see a choice of triggers. Pick one. In this example, we'll use "New tweet from search". Maybe there's a tech conference happening and you want to collect all the tweets used with that conference's hashtag (like, #CES2018).
- Now, you need to select the +That in the "If This Then That" logo toward the center-top of the page. This is where you'll specify what you want to happen next.
- Choose an action service. For instance, you can select the weekly email digest option. This will bundle up all those hash tagged tweets.
- Review your Applet and click Finish when done.
- It'll now be live. To view your Applet, select My Applets from the top of your IFTTT dashboard.

4.2.6.4 WHAT ARE IFTTT WIDGETS?

Widgets are shortcuts that allow you to run certain Applets with the touch of a button on your iOS or Android device. In order to use them, you need to have the IFTTT app on your phone. After you turn on a widget Applet, you can add it as a icon in your Today screen on iOS, or your home screen on iOS or Android. Step-by-step instructions on how do that are available from IFTTT's Help center.

Figure: 4.2.6 IFTTT LOGO

TESTING

5. TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

5.1 SYSTEM TESTING

5.1.1 TEST APPROACH

We will test the project in two stages: software and hardware. The software part is to be tested via the Arduino IDE, whereas the hardware part has to be tested physically. It is necessary to check whether the system is working properly or not. To check whether the readings are accurate, we will check the distance pointed out by the sensor by a meter tape.

After uploading the code, open the Serial Monitor and it will show you an IP address as shown below.

Type this IP address in your browser, it will show you the output as shown below. You will have to refresh the page again if you want to see again that the trash can is empty or not.

So this how this Garbage Monitoring System works, this project can be further enhanced by adding few more features in it like we can set one more message when the Trash Can is half filled or we can trigger a Email/SMS to alert the user when Trash Basket is full.

There can be three results in this project. They are

1) When the garbage is **LOW , GREEN COLOR LED** will be ON

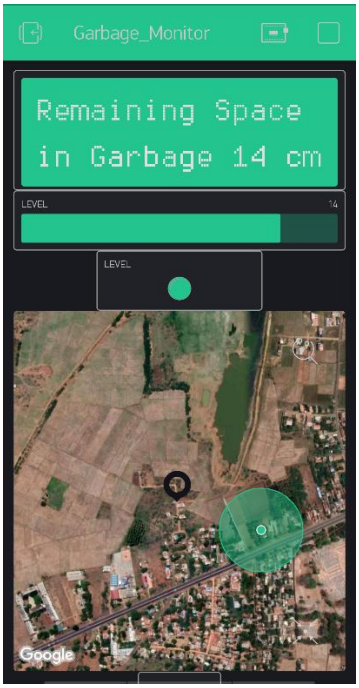


Figure:5.1.1 Low Level

2) When the garbage is **HALF, YELLOW LED** will be ON.

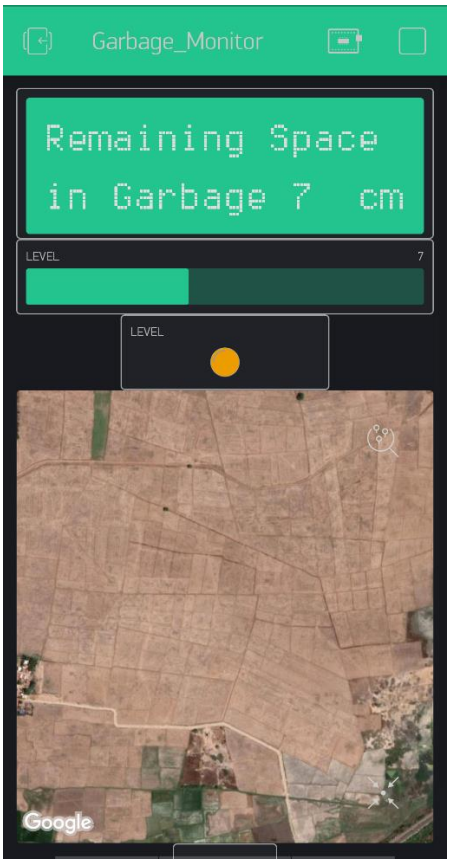
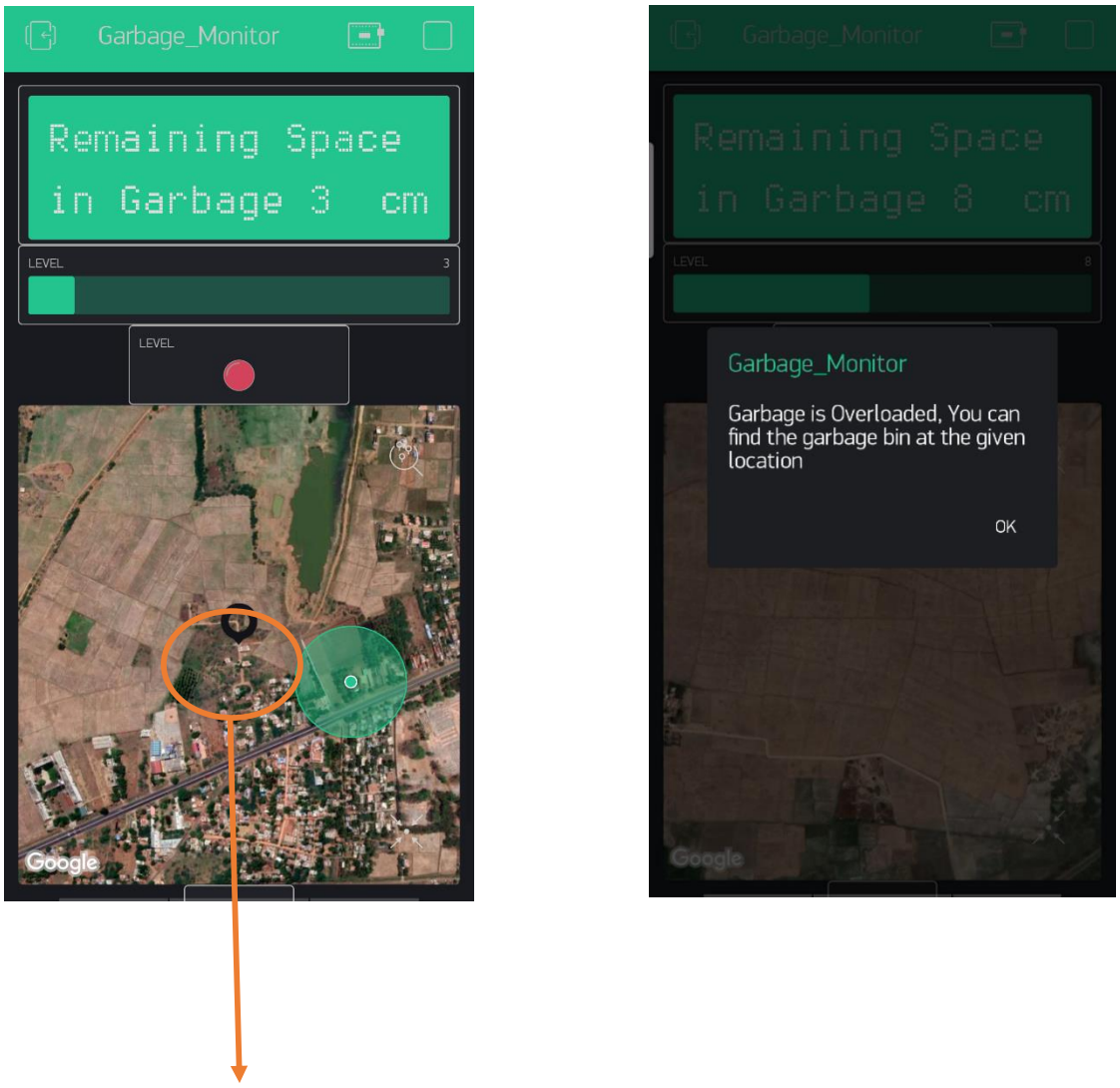


Figure:5.1.2 Half Level

3) When the garbage is **FULL, RED LED** will be ON.



GARBAGE LOCATION

Figure:5.1.3 High Level

5.1.1.1 NOTIFICATIONS:

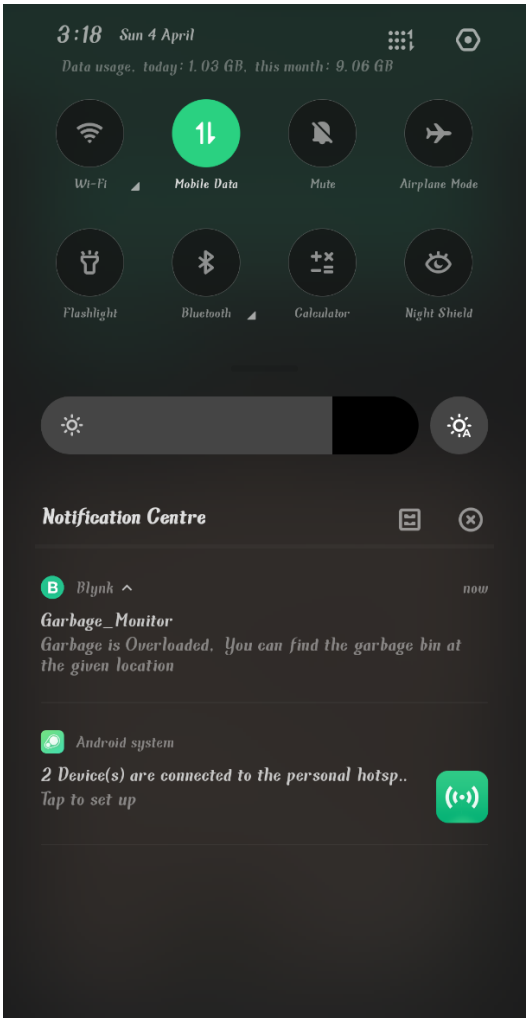


Figure:5.1.4 Alert Notification

CONCLUSION

6. CONCLUSION

We built an efficient garbage monitoring system which can be used to monitor the level of garbage in the dump. This data can be further used to plan garbage collection trips more efficiently, ultimately reducing overflowing bins and helping have better public sanitation. In this project, an integrated system of Wi-Fi modem, IoT, GSM, 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 analyzed 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 to collection truck. 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.

6.1 ADVANTAGES:

- Very simple circuit.
- The HCSR04 sensor is very rugged.
- Helps monitor garbage levels.
- Uses very small amount of electricity.
- Ultimately helps in better planning of garbage pickups.
- Can help in reducing overflowing bins.
- Reduces trips to areas where the bins still have a lot of capacity.

6.2 FUTURE ENHANCEMENT

The main objective is to maintain the level of cleanliness in the city and form an environment which is better for living .by using this system we can constantly check the level of the garbage in the dustbins which are placed in various parts of the city. If a particular dustbin has reached the maximum level then the employees can be informed and they can immediately take certain actions to empty it as soon as possible. The employees can be check the status of these bins anytime on their mobile phones. This can prove to be a very useful system if used properly.

6.3 APPLICATIONS

This project can also be used in the **" SMART CITY".** This project is also helpful in the government project of **"SWACHH BHARAT ABHIYAN".**

REFERENCES

7.REFERENCES

7.1 BOOK REFERENCE

1. Navghane S S, Killedar M S and Rohokale D V 2016 IoT Based Smart Garbage and waste collection, International Journal of Advanced Research in Electronics And Communication.
2. Monika K A, Rao N, Prapulla S B and Shobha G 2016 Smart Dustbin-An EfficientGarbage Monitoring System International Journal of Engineering Science andComputing 6 7113-16.
3. Medvedev A, Fedchenkov P, Zaslavsky A, Anagnostopoulos T and KhoruzhnikovS,2015 Waste management as an IoT-enabled service in smart cities In Conference on Smart Spaces Springer International Publishing 104-15.

7.2 WEBSITE REFERENCE

1. www.buildofy.in/smart_home_designs
2. https://create.arduino.cc/projecthub/Technovation/smart-garbage-monitoring-systemusing-arduino-101-3b813c
3. https://github.com/sourabhdeshmukh/Smart-Dustbin
4. http://invent.module143.com/temperature-and-humidity-using-nodemcu-blynk/
5. Prakash, Prabhu,"IoT based management system for smart cities", International Journal of Innovative Research in Science, Engineering and Technology, vol 4, Issue 2, February 2016.
6. Kanchan Mahajan, Prof J. S. Chitode,"Waste Bin Monitoring system Using Integrated Technologies", International Journal of Innovative Research in Science,Engineering and Technology(An ISO 3297:2007 Certified Organization) Vol 3, Issue 7, July 2014.
7. Prof.R.M.Sahu, Akshay Godse, Pramod Shinde, Reshma Shinde," Garbage and Street Light Monitoring System using Internet of Things", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, Vol 4, Issue 4, 4 April 2016.

APPENDIX

8.APPENDIX

8.1 SOURCE CODE

//Including Necessary Header Files

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <BlynkSimpleEsp8266.h>
```

//Declarations And Defenitions

```
#define WIFI_SSID "KP WIFI"
#define WIFI_PASS "keerthi2000"
```

//Authenticating details of Adafruit ID

```
#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "Keerthi_Priyan"
#define MQTT_PASS "aio_DZtJ26dstgmN2Dx84dK1L5XPSM4y"

#define BLYNK_PRINT Serial
#define TRIGGERPIN 2  //Setting Trigger pin to D3 Pin
#define ECHOPIN    0  //Setting Trigger pin to D4 Pin
#define BLYNK_GREEN     "#23C48E"
#define BLYNK_YELLOW    "#ED9D00"
#define BLYNK_RED       "#D3435C"
```

//LED Lights on Garbage

```
int ledGreen = 14;
int ledYellow = 12;
int ledRed = 13;

int f=0; //Flag value to give notification to the user
int f2=0;//Flag value to turn on or off the Garbage Monitor device and App

float lat = 10.775925;//Lattitude And Longtitude Of Garbage's Location
float lon = 78.630097;

char auth[] = "z_amJVAuCvC11G7Ylg65zoO0wEqYlxvZ"; //Blynk App's Authentication Code To Connect Over Wifi-Network
```

// My WiFi credentials

```
char ssid[] = "KP WIFI";
char pass[] = "keerthi2000";
```

//Declaration Of Mobiles App's LCD,LED,MAP Pins

```
WidgetLCD lcd(V1);
WidgetLED led1(V2);
WidgetMap myMap(V3);
```

//Setting up MQTT

```
WiFiClient client;
```

```
Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT, MQTT_NAME, MQTT_PASS);

Adafruit_MQTT_Subscribe onoff = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME "/f/GM_ONOFF");
Adafruit_MQTT_Publish LightsStatus = Adafruit_MQTT_Publish(&mqtt, MQTT_NAME "/f/LightsStatus");


void setup()
{
  Serial.begin(9600);
  pinMode(TRIGGERPIN, OUTPUT);  //Setting up Ultra Sonic Sensor's Input Output
  pinMode(ECHOPIN, INPUT);
   pinMode(LED_BUILTIN, OUTPUT);
   pinMode(ledGreen, OUTPUT);
   pinMode(ledYellow, OUTPUT);
   pinMode(ledRed, OUTPUT);
  //Subscribe to the onoff topic
  mqtt.subscribe(&onoff);

}
```

//Funtion deefenition to connect Adafruit server to Garbage Monitor

```
void MQTT_connect()
{

  // Stop if already connected
  if (mqtt.connected() && mqtt.ping())
  {
```

```
  //    mqtt.disconnect();
      return;
  }

  int8_t ret;

  mqtt.disconnect();

  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) // connect will return 0 for connected
  {
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);  // wait 5 seconds
    retries--;
    if (retries == 0)
    {
      ESP.reset();
    }
  }
  Serial.println("MQTT Connected!");
}
```

//Function defenition to give outputs to the Garbage Monitor's App

```
void blynk()
{
  lcd.clear();
```

```
 lcd.print(0, 0, "Remaining Space in Garbage"); // use: (position X: 0-15, position Y: 0-1, "Message you want to print")
```

// Working code of Ultra Sonic Sensor to Calculate Distance

```
 long duration, distance;
 digitalWrite(TRIGGERPIN, LOW);
 delayMicroseconds(3);
 digitalWrite(TRIGGERPIN, HIGH);
 delayMicroseconds(12);
 digitalWrite(TRIGGERPIN, LOW);
 duration = pulseIn(ECHOPIN, HIGH);
 distance = (duration/2) / 29.1;   //Formula to find distance
 Serial.print(distance);
 Serial.println("Cm");
 lcd.print(27,0, distance);
 lcd.print(30,0,"cm");
 Blynk.virtualWrite(V4,distance);  //To show Level Of Empty Space in Garbage via App


 if((distance>1) &&(distance <=5)) //To Check Garbage's Space through Various if Statements
 {
  if(f==1)
  {
    Blynk.notify("Garbage is Overloaded, You can find the garbage bin at the given location");
```

////To send Mobile Notifaction And Location of Garbage

```
    f=0;
  }
```

```
  // Turn LED on, so colors are visible
  led1.setColor(BLYNK_RED);
  Serial.println("LED on V2: red");
  led1.on();
 digitalWrite(ledRed, HIGH);
 digitalWrite(ledYellow, LOW);
 digitalWrite(ledGreen, LOW);
 myMap.location(0, lat, lon, "Garbage is here");
}


else if(distance >5 && distance <=10)
{
  f=1;
  // Turn LED on, so colors are visible
  led1.setColor(BLYNK_YELLOW);
  Serial.println("LED on V2: Yellow");
  led1.on();
  digitalWrite(ledRed, LOW);
 digitalWrite(ledYellow, HIGH);
 digitalWrite(ledGreen, LOW);
}

else if(distance >10)
{
  f=1;
  led1.setColor(BLYNK_GREEN);
  Serial.println("LED on V2: Green");
  led1.on();
  digitalWrite(ledRed, LOW);
 digitalWrite(ledYellow, LOW);
```

```
   digitalWrite(ledGreen, HIGH);
  }
   Adafruit_MQTT_Subscribe * subscription;
  while ((subscription = mqtt.readSubscription(1000)))
  {
    //If we're in here, a subscription updated...
    if (subscription == &onoff)
    {
      //Print the new value to the serial monitor
      Serial.print("APP's Status: ");
      Serial.println((char*) onoff.lastread);
      Serial.println("Turning OFF Garbage Monitor App");
      //If the new value is  "ON", turn the light on.
      //Otherwise, turn it off.
      if (!strcmp((char*) onoff.lastread, "OFF"))
        {
          f2=0;
        }
      delay(1500);
    }
  }
}

void loop()
{
  //Connect/Reconnect to MQTT
  MQTT_connect();

  //Read from our subscription queue until we run out, or
  //wait up to 5 seconds for subscription to update
  Adafruit_MQTT_Subscribe * subscription;
```

```
  while ((subscription = mqtt.readSubscription(5000)))
  {
    //If we're in here, a subscription updated...
    if (subscription == &onoff)
    {
      //Print the new value to the serial monitor
      Serial.print("APP's Status: ");
      Serial.println((char*) onoff.lastread);
      Serial.println("Turning ON Garbage Monitor App");
      //If the new value is  "ON", turn the light on.
      //Otherwise, turn it off.
      if (!strcmp((char*) onoff.lastread, "ON"))
      {
       f2=1;
        //active low logic
        Blynk.begin(auth, ssid, pass);
        Blynk.run();
        while(f2==1)
        {
          blynk();
        }
      }
      else
      {
        Serial.println("ERROR");
      }
    }
  }
}
```

8.2 OUTPUT SCREENS

Figure:8.2.1 Blynk App Output Screen

Figure:8.2.2 Garbage Monitoring Computer Output

GARBAGE MOBILE OUTPUT

Figure:8.2.3 NOTIFICATION

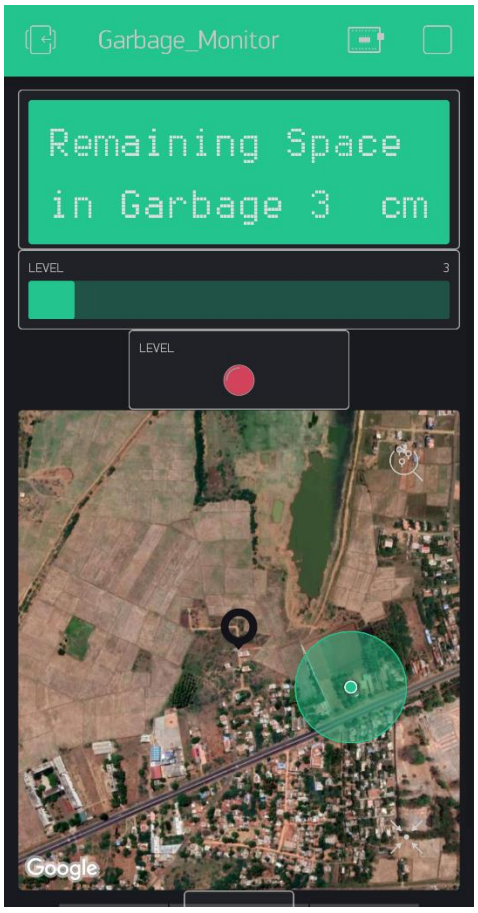


Figure:8.2.4 HIGH LEVEL

NOTIFICATIONS

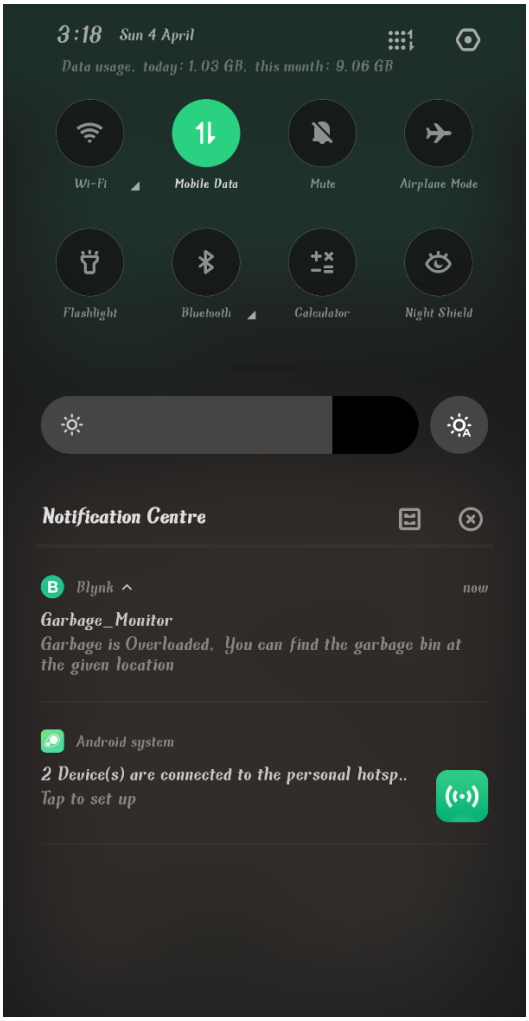


Figure:8.2.5 ALERT NOTIFICATION

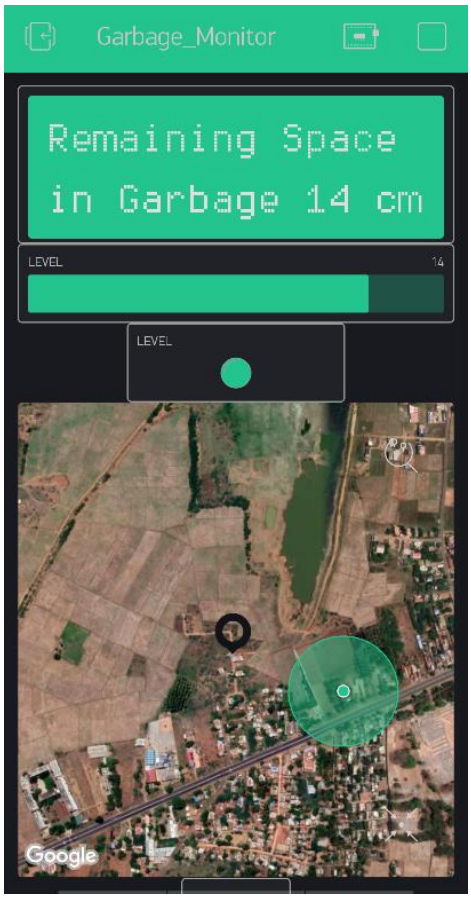


Figure:8.2.6 LOW LEVEL

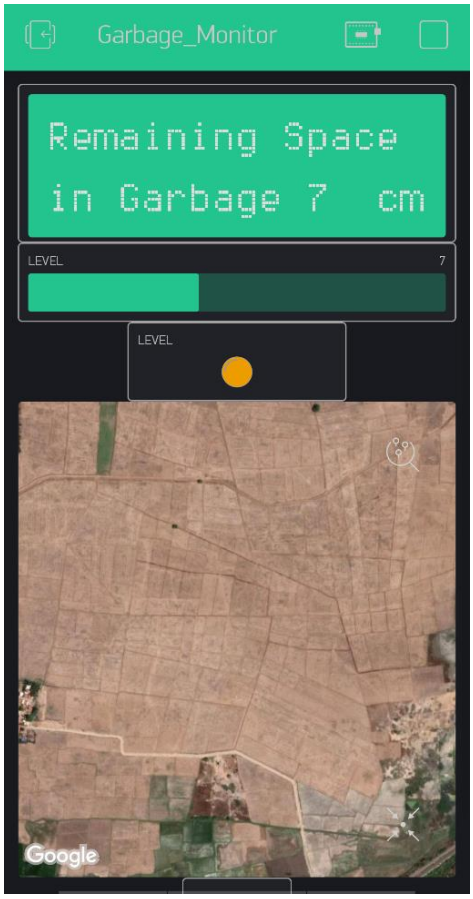


Figure:8.2.7 HALF LEVEL

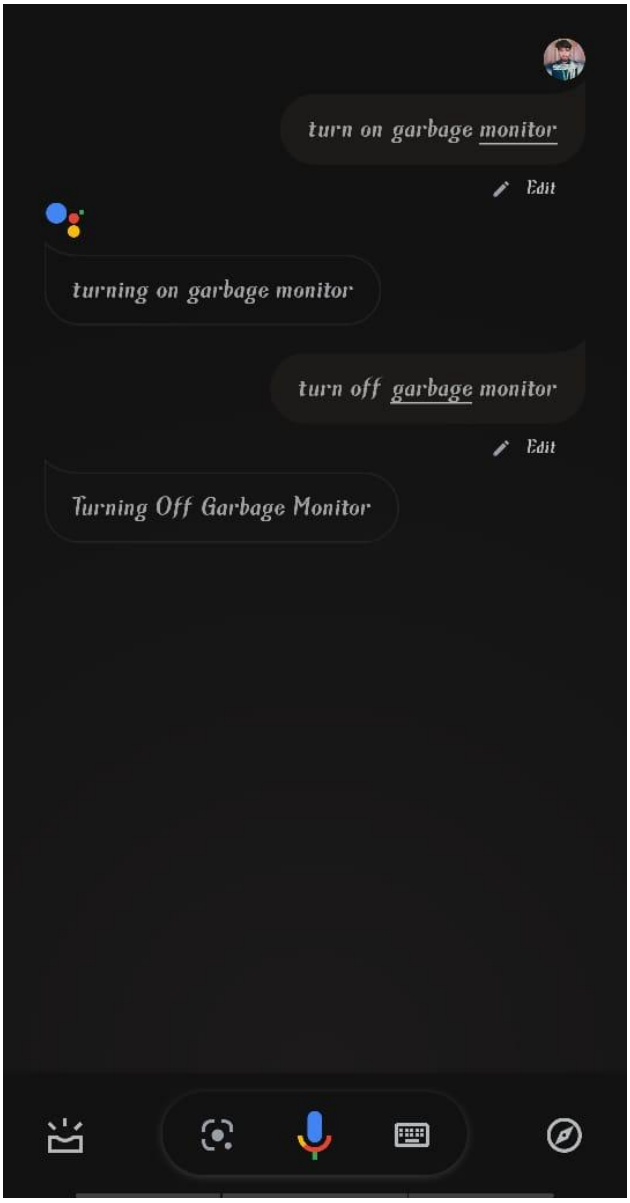


Figure:8.2.8 GOOGLE ASSISTANT VOICE OVER