IOT ENABLED SMART DUSTBIN

A project report submitted to the

BHARATHIDASAN UNIVERSITY,

TIRUCHIRAPPALLI –23

in partial fulfilment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Submitted by

RAMACHANDRAN M
(Register Number – 21MCA69)

Sponsored by Rashtriya Uchchatar Shiksha Abhiyan (RUSA)

Under the Supervision and Guidance of

Dr. P. SUMATHY

Assistant Professor

SCHOOL OF COMPUTER SCIENCE, ENGINEERING & APPLICATIONS
BHARATHIDASAN UNIVERSITY
TIRUCHIRAPPALLI – 620 023
TAMILNADU
MAY-2023

SCHOOL OF COMPUTER SCIENCE, ENGINEERING & APPLICATIONS
BHARATHIDASAN UNIVERSITY
TIRUCHIRAPPALLI –23

CERTIFICATE

This is to certify that the major project work entitled "**IOT ENABLED SMART DUSTBIN**" is a bonafide work done by **Mr. RAMACHANDRAN M, Reg. No: 21MCA69** in partial fulfilment of the requirement for the award of the degree of "**MASTER OF COMPUTER APPLICATIONS**" under the guidance **of Dr. P. SUMATHY, Assistant Professor, School of Computer Science, Engineering & Applications, Bharathidasan University, Tiruchirappalli–23** during the academic year 2022 – 2023.

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

The Project evaluation and viva-voice examination held on_____ at the School of Computer Science, Engineering and Applications, Bharathidasan University, Tiruchirappalli – 23.

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

First and foremost, I would like to thank **God Almighty** for the abundant shower of grace blessings on me throughout my entire project.

I wish to express my deep sense of gratitude to **Dr. G. Gopinath Ganapathi, MCA., Ph.D., Professor and Head, School of Computer Science, Engineering and Applications, Bharathidasan University, Tiruchirappalli-23** for giving me this opportunity to do this project.

I would like to express my genuine regard and respect to my project guide **Dr. P. Sumathy, Assistant Professor, School of Computer Science, Engineering and Applications, Bharathidasan University, Tiruchirappalli-23** for the encouragement, guidance, inspiration, and constructive suggestions that help me in carrying out this project successfully.

It is my duty to express my thanks to all **Teaching** and **Non-Teaching Staff members** of Department of Computer Science & Applications, Bharathidasan University, Tiruchirappalli-23 those who offered me help directly and indirectly by their suggestions.

Finally, I would like to thank my beloved family members, friends those who extended support towards the successful completion of my project.

(RAMACHANDRAN M)

21MCA69

IOT ENABLED SMART DUSTBIN

ABSTRACT

Today, one of the challenges of most cities and towns are confronting is the decline in condition of cleanness of the environment regarding the garbage management. This occurs due to the mismanagement of the garbage collection. This mismanagement creates the spread of garbage in community which in turn create sun healthy condition in the immediate area. It also stimulates several serious diseases amongst the people in close proximity and degrades the beauty of the area. To avoid mismanagement of the garbage and to improve the cleanness of the society, Garbage monitoring system is designed. In the proposed system, the level of the garbage is detected with the help of ultrasonic sensor and sent to the authorized agency for garbage collection through buzzer system. The garbage in a dustbin is all overflow of the dustbin and also spelled out of the dustbin. Many people are throwing garbage on that dustbin which are already full or overflow of the dustbin. Due to this unclean of garbage bins bad smell is created and also toxic and unhygienic garages which are bad for the environment is produced. This creates a very bad look of the city which is a way to support to the air pollution and to some harmful diseases which are easily spreadable. In the proposed system the ultrasonic sensor, ardino uno, LCD monitor, GSM AND GPS for location detection, and alert buzzer are used for developing IOT based smart dustbin system.

TABLE OF CONTENTS

S.NO	Titles	Page No.
1	**1. Introduction** 1.1 Internet of Things 1.2 Garbage Level Indicator 1.3 Project Description 1.4 Existing System 1.5 Proposed System	1-11
2	**2. Environment Specification** 2.1 Hardware Specification 2.2 Software Specification	12
3	**3. Program Design** 3.1 Modules 3.2 Detailed Description 3.3 System Architecture 3.4 Use Case Diagram 3.5 Feasibility Study 3.6 Data Flow Diagram	13-38
4	**4.Conclusion and Future Enhancements** 4.1 Conclusion of the Project 4.2 Future Enhancement	39-40
5	**5.References** 5.1 Book Reference	41
6	**6.Appendix** 6.1 Source Code 6.2 Screenshots	42-66

CHAPTER 1

INTRODUCTION

1.1 INTERNET OF THINGS

The current practice of waste disposal is unplanned and out of control due to lack of labor efforts of people working on the ground level emptying the garbage bins whenever they are full. The phenomenon of getting bins full is not fully dependent on a time pattern, instead it sometimes becomes abruptly full or sometimes requires more than normal time to become full. The detection, monitoring and management of waste is one of the key problems of the current era. The traditional way of manually examining the wastes in waste bins is an inconvenient process and utilize more labor work, time and cost which can easily be avoided with our present technologies. The idea struck us when we noticed that the garbage truck used to go around the town to collect solid waste twice a day. Monitoring and controlling operations of sustainable urban and rural infrastructures like bridges, railway tracks and on- and offshore wind farms is a key application of the IoT. The IoT infrastructure can be used for monitoring any events or changes in structural conditions that can compromise safety and increase risk. The IoT can benefit the construction industry by cost-saving, time reduction, better quality workday, paperless workflow and increase in productivity. It can help in taking faster decisions and saving money in Real-Time Data Analytics. It can also be used for scheduling repair and maintenance activities efficiently, by coordinating tasks between different service providers and users of these facilities. IoT devices can also be used to control critical infrastructure like bridges to provide access to ships. The usage of IoT devices for monitoring and operating infrastructure is likely to improve incident management and emergency response coordination, and quality of service, up-times and reduce costs of operation in all inf restructure-related areas. Even areas such as waste management can benefit from automation and optimization that could be brought in by the IoT.

One of a famous technology in this world is an Internet of Things (IOT). Network connecting objects is referred as IOT. The feature of this technology is to communicate and exchange the data among themselves. IOT are activate with the devices like sensor, motor and some UNO board. Trash bin is used for storage the waste management in the world. In regular activities, a normal dustbin utilized for throwing the waste and dustbin is filled to empty the waste inside a dustbin. This is the basic use of a normal dustbin where no components are used,

no coding is performed, and everything is done by hand manually. If the dustbin is full, people start throwing the waste around the dustbin, it causes smell and various diseases. To avoid this kind of problem, we used for IOT and some technology to keep the dustbin or garbage bin and environment very clean.

1.2 GARBAGE LEVEL INDICATOR

IOT based Garbage Monitoring System will inform us whether the trash can is empty or full via IOT Gecko web development platform and you can know the status of your "Trash can". This proposed system uses ultrasonic sensors placed over the garbage bins to detect the garbage level and compare it with the garbage bin depth. This system makes use of AVR family microcontroller, LCD screen, Wi-Fi Modem and a buzzer. It is powered by 12V transformer. LCD screen is used to show the status of the level of garbage collected in the bins. A particular limit is set i.e., 5 units in the program code and if the garbage level exceeds it, system puts on the buzzer. Also, the web page gives a visual view of the garbage bins and highlights the garbage collected in green color in order to show the proportion of garbage collected. In this project, we will try to build an Internet of Things (IOT) based system will automatically notify and keep such garbage clean in a proper manner. Each and every person in the world disposes the waste in the dustbin and it full, they empty the waste inside the bin. This is the basic use of a normal dustbin where no components are used, no coding is performed, and everything is manual. The maintenance of the bin is also not proper where the lid in the overflowing of the waste from the bin. The second method is use of dustbin with different segregations like green and blue bins which is placed together or the dustbin where only recyclable waste should be disposed. The third method uses Arduino, servomotor, GSM module, ultrasonic sensor for doing the same result and it is not cost efficient. Ultrasonic sensor is present inside the dustbin where the height of the waste inside the dustbin is measured and it send a mail when the dustbin is above 70 percent. Only sending the notification is the existing method.

1.3 PROJECT DESCRIPTION

IoT based sensor system is applied to detect the volume of trash. The GPS (Global Positioning system) system is used to identify the location of these smart bins. This location information is communicated to the waste management department through GSM (Global

System for Mobile Communications) on smartphones. Using the Google Maps the location of the dustbin can be found. With progress in human technology, we have seen a substantial progress in the amount of waste generated. Recycling is the only way to manage this huge amount of waste. But recycling requires garbage to be segregated. Without segregation garbage cannot be recycled because different type of garbage requires different recycling processes. Also, it is important to educate users and instruct them every time they come near the dustbin about instructions about throwing the trash. For this purpose, we design a garbage disposal system that uses multiple dustbins with a voice-based system that speaks to the user each time he she stands before the dustbin.

The system makes use of a camera to detect presence if any person in front of the dustbin. If a person is detected, the system issues voice instructions to the user about throwing right garbage in the right bin. In case the dustbin is full it instructs the user to find another dustbin to throw garbage in. To develop this system, we make use of a raspberry Pi controller. The controller is interfaced with a camera and a voice speaker for detection and communication. The controller gets dustbin level input using ultrasonic level sensors each having LED indicators interfaced to it. The level sensors are used to constantly feed the raspberry pi with bin levels. The raspberry pi is also interfaced with a Wi-Fi module to transmit the level data over the internet. The Level sensor panels are made to be easily mounted over any dustbin. This allows the system to be easily screwed over any dustbin for instant installation.

The data is transmitted over IOT-to-IOT gecko platform which displays the bin level data over internet. This indication can be used to alert the authorities that the garbage bins need to be emptied. Thus, the system automates garbage segregation and level monitoring to help counter the garbage crisis using IOT. IoT based garbage level monitoring system is an emerging technology that is utilized for monitoring waste fill level of public and industrial garbage bins. The fundamental purpose of a garbage level monitoring system is to help the municipal services to pick the trash at the right time before a garbage bin overflows and cause discomfort to general public. An advanced garbage monitoring system can not only measure garbage level, it can also detect toxic chemical substances, flammable gases and even radioactive materials using

advanced sensors and alert the authorities immediately via internet before any disastrous incident occurs.

1.4 EXISTING SYSTEM

1.4.1 SMART DUSTBIN

As people are getting smarter, so are the things. While the thought comes up for Smart cities. There is a requirement for Smart waste management. It is a common sight to witness garbage spilled out in and around the dustbins. The area around an improperly maintained dust bins can house disease spreading insects like mosquitoes, flies, bees and driver ants. The environment around a dustbin is also conducive for increasing the pollution level in air. Air pollution due to a dustbin can produce bacteria and virus which can produce life threatening diseases in human beings. The idea of Smart Dustbin is for the Smart buildings, Colleges, Hospitals and Bus stands. The Smart Dustbin thus thought is an improvement of normal dustbin by elevating it to be smart using sensors and logics. For Smart dustbin operation we are using ultrasonic sensor for detecting distance and object and another sensor servomotor is used for opening and closing the dustbin top and we are also using PIR sensor which is used for calculating the level of dustbin and also, we are using led which glows showing the level of dustbin up to which it is filled.

1.4.2 SMART DUSTBIN WITH IOT NOTIFICATION

Today, one of the challenges of most cities and towns are confronting is the decline of cleanliness of the environment regarding the garbage management. This occurs due to the mismanagement of the garbage collection. This mismanagement creates the spread of garbage in community which in turn creates unhealthy condition in the immediate area. It also stimulates several serious diseases amongst the people in close proximity and degrades the beauty of the area. To avoid mismanagement of the garbage and to improve the cleanliness of the society ,smart dustbin with IOT notifications is designed .In the proposed system the level of the garbage is detected with the help of ultrasonic sensor and sent to the authorized agency for garbage collection through GSM system.IR sensor is used to detect the motion of the people coming to the garbage bin with trash while the bin is at full status and block adding of any more garbage to

the bin through informing them by buzzer .The GSM and the peripheral sensors used are interfaced through the Arduino microcontroller .Depending on the received messages through the GSM at control room it is displayed on LCD and the authorized person inform the drivers to collect the garbage on time. This will capably help to monitor the garbage collection to make the environment smart, clean and safe.

1.4.3 A REVIEW OF IOT BASED INTELLIGENT BINS & SMART WASTE MANAGEMENT SYSTEMS

With the evolution of IoT, came the enrichments of human lifestyle. However, one issue that still needs improvement is the waste management and handling of garbage collection. The total generation of waste in India is about 150 million tons every day and hence waste management in India has become a very challenging task. This paper focuses on the IoT based system that can support the existing workforce to manage waste generated in the sectors of a smart city. In this paper, we have the critical analysis of existing literature which is relevant to intelligent dustbins and the mechanisms associated with IoT. Though, the literature consists of a lot many research contributions, but, here, we have analyzed around twenty research papers. We have analyzed all the research works on common basis and represented them in tabular form so that comparison between the different methods can be easily done. The emphasis is on the platform used by the concerned authors, the software and programming language used and the various performance evaluation parameters like number of sensors, type of storage, and the other parameters like fire detection, waste prediction, waste classification, route scheduling, approx. cost, etc. Hence, the merits and demerits of each of the existing approach can be determined. Finally, the findings are summarized related to the studied and analyzed research papers.

1.4.4 IOT-BASED SMART BINS

In this paper, we describe the formatting guidelines for IJCA Journal Submission Nowadays; waste management has become a major problem in life cycle. It includes the collection, transport, treatment and disposal of waste, together with monitoring and regulation of the waste management process. The significant cause of waste management is brisk growth in the rate of urbanization and thus there is a need of proper planning. To avoid all such harmful scenarios and

maintain public cleanliness and health, we intend to propose a solution for this problem “Smart Bin” which will focus on resolving this problem. This process will alarm and notify the authorized person through a software when the garbage bin is about to fill. This system pivots around the overflowing bins and keeping the areas clean.

1.4.5 SMART WASTE MANAGEMENT USING IOT POWERED DUSTBIN

The 21st century is the era of technological development. Estimation by Cisco says that more than 50 billion devices will be connected to the internet in the 21st Century. For the betterment of India internet of things play most important role. Devices are connected used for the purpose of safety and quality of life, city is surrounded by vehicle and infrastructure. System integrators, network operators and technology provide these are used in the working of government to give the smart solution. On standards-based communications platform it is difficult to generate the solution. Hence, we proposed solution for a smart waste collection management based on providing, IoT prototype with sensors. This will make things to become “SMART” and influence the lives of humans. Data can be read, collect and transmit large amount of data over the Internet. As the technological advancement increases in urbanization, industrialization and population governments across the global will need to device sustainable development plans. In recent years the and government are investing huge sums of money towards establishing smarter cities because notion of Smart City has been trending across the global. A Smart Waste Management System makes a smart city complete. In this project we are going to use the sensors and other components to make the smart dustbin which help to make clean and green India also help to reduce the diseases occur due to waste.

1.4.6 SMART DUSTBIN USING ARDUINO

The main objective of the project is to design a smart dustbin which will help in keeping our environment clean and also eco friendly. We are inspired from Swaach Bharat Mission. Nowadays technologies are getting smarter day-by-day so, as to clean the environment we are designing a smart dustbin by using Arduino. This smart dustbin management system is built on the microcontroller-based system having ultrasonic sensors on the dustbin. If dustbin is not maintained than these can cause an unhealthy environment and can cause pollute that affect our health. In this proposed technology we have designed a smart dustbin using ARDUINO UNO,

along with ultrasonic sensor, servo motor, and battery jumper wire. After all hardware and software connection, now Smart Dustbin program will be run. Dustbin lid will when someone comes near at some range than wait for user to put garbage and close it. It's properly running or not. For social it will help toward health and hygiene, for business for we try to make it affordable to many as many possible. So that normal people to rich people can take benefit from it. The rate increasing population in our country has increasing rapidly and also, we have increase in garbage which have increased environmental issue. Dustbin is a container which collects garbage's or stores items which recyclable or non-recyclable, decompose and non-decompose. They are usually used in homes, office etc., but in case they are full no one is there to clean it and the garbage are spilled out. The surrounding of a dustbin is also conducive for increasing the pollution level. Air pollution due to a dustbin can produce bacteria and virus which can produce life harmful diseases for human. Therefore, we have designed a smart dustbin using ARDUINO UNO, ultrasonic sensor which will sense the item to be thrown in the dustbin and open the lid with the help of the motor. It is an IOT based project that will bring a new and smart way of cleanliness. It is a decent gadget to make your home clean, due to practically all offspring of home consistently make it grimy and spread litter to a great extent by electronics, rappers and various other things. Since the smart dustbin is additionally intriguing and children make fun with it so it will help to maintain cleanliness in home. It will be applied for various type of waste. Dustbin will open its lid when someone/object is near at some range then it will wait for given time period than it will close automatically. Here lid will close when you don't want to use and it will only open when it required.

1.4.7 SMART DUSTBIN WITH AUTOMATIC OPEN/CLOSE COVER

In this recent world, urbanization has increased tremendously. At the same phase, there is increasing amount of in waste production. Waste management has been a crucial issue to be considered. This report is a different way to achieve this good cause. In this report, smart bin is built on a microcontroller-based platform Arduino - Uno board, which is interfaced with Ultrasonic sensor. It will stop overflowing of dustbins along roadsides and localities as smart Dustbins are managed in real time. Once these smart bins are implemented on a large scale by replacing the traditional bins, the waste can be quickly managed to its efficient level as it avoids unnecessary lumping of wastes on roadside. Foul smell from these rotten wastes that remain

untreated for a long time, due to negligence of authorities and carelessness of public may lead to long term problems. Breeding of insects and mosquitoes can create nuisance around promoting unclean environment. This may even cause dreadful diseases. The goal of this project is to keep our environment clean. It also aims at creating a clean as well as green environment. The smart bins are used as ultrasonic sensors which detect the garbage. The container is divided into three levels of garbage being collected in it. Every time the garbage crosses a level the sensors receive the data of comes garbage to the bin. This data is further goes to the servo motor threw the Arduino Uno circuit board. Placing the ultrasonic sensors at the top of the bin, like on the cover of the bin. The comparison is done with help of microcontroller. After analyzing the image an idea about level of garbage in the can and from the load cell sensor, weight of garbage can be known. Accordingly, information is processed that is controller checks if the threshold level is exceeded or not. This is convenient to use but economically not reliable. Instead of using plenty of bins in an unordered fashion around the city, minimal number of smart bins can be used. Using only one sensor at the surface level instead of three not only makes it affordable but also achieves the same result

1.4.8 GARBAGE MONITORING SYSTEM USING ARDUINO

In last few years there is a rapid growth in urban development plans, the concept of smart cities. While the thought comes up for Smart cities there is a requirement for Smart waste management. The idea of Garbage monitoring system is for the Smart buildings, Colleges, Hospitals and Bus stands. The Garbage monitoring system thought is an improvement of normal dustbin by elevating it to be smart using sensors. Garbage monitoring system is a new idea of implementation which makes a normal dustbin smart using ultrasonic sensors for garbage level detection, display and sending message to the concern department person updating the status of the bin using GSM modem. Garbage! In our daily life, we see the pictures of garbage bins being overfull and all the garbage smells out. This leads to the number of diseases as large number of insects and mosquitoes breed on it. A big face up to the smart cities is solid waste management, not only in India almost all the countries in the world. This project gives the most efficient ways to keep our environment clean and green. The upcoming large number of smart cities, large numbers of responsibilities is also required to be fulfilled. The most important need of a smart way of life begins with cleanliness and cleanliness begins with

smart dustbin. A people will get its waste dispatch properly only if the dustbins are placed well and collected well. The main problem in the current waste management system in most of the cities is the damaging status of dustbins. So, by using the new technology we send the information to the concern persons and display boards are arranged in the concern offices. The progress of waste across the entire city can be tracked and thus can be monitored by a single system efficiently and concretely. This system can prove to be a revolution for the whole waste management system of future smart cities.

1.4.9 AUTOMATED GARBAGE MONITORING SYSTEM USING ARDUINO

Today, one of the challenges of most cities and towns are confronting is the decline in condition of cleanness of the environment regarding the garbage management. This occurs due to the mismanagement of the garbage collection. This mismanagement creates the spread of garbage in community which in turn creates unhealthy condition in the immediate area. It also stimulates several serious diseases amongst the people in close proximity and degrades the beauty of the area. To avoid mismanagement of the garbage and to improve the cleanness of the society, Garbage monitoring system is designed. In the proposed system, the level of the garbage is detected with the help of ultrasonic sensor and sent to the authorized agency for garbage collection through GSM system. PIR sensor is used to detect the motion of the people coming to the garbage bin with trash while the bin is at full status and block adding of any more garbage to the bin through informing them by speaker. The GSM and the peripheral sensors used are interfaced through the Arduino microcontroller. A GUI is also developed to monitor the desired information related to the garbage bins for different selected locations. Depending on the received messages through the GSM at control room it is displayed on LCD and the authorized person inform the drivers to collect the garbage on time. This will capably help to monitor the garbage collection to make the environment smart, clean and safe.

1.5 PROPOSED SYSTEM

The Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, the other receives them. The speed of sound is approximately 341 meters (1100 feet) per second in air. The ultrasonic sensor uses this information along with the time difference between the sending and receiving the sound pulse to determine the distance to an object. It uses the following mathematical equation: Distance = (Time x Speed of Sound)/2 Time = the time been when an ultrasonic wave is transmitted and when it is received. SMART DUSTBIN is an ARDUINO based project. Here we are using Arduino for code execution, for sensing we used ultrasonic sensor which will open lid and wait for few moments. It will bring drastic changes in tern of cleanliness with the help of technology. Everything is getting with smart technology for the betterment of human being. So this help in maintaining the environment clean with the help of technology. It is a sensor based dustbin so it would be easy to access/use for any age group. Our aim is also to make it cost effective so that many numbers of people can get the benefit from this. And it should be usable to anyone and helpful for them. Garbage level detection is the done by ultrasonic sensors (HC-SR04).The ultrasonic sensors is placed on top of the dustbin facing the bottom. The sensors continuously emits the sonic waves, when the sonic waves hit the object and reflect back, the echo in the sensors senses the waves and calculates the distance of the object. Arduino Mega 2560 is used for controlling whole the process detecting garbage in different places and depending on the program first display in LCD to reminding the garbage level in the bin even though the garbage is not take out from the bin then the particular bin information is sent to higher officials through GSM.

In ‘smart dustbin’ system, the level of garbage in the dustbins is detected with the help of Sensor systems, and communicated to the authorized control room through buzzer system. Microcontroller is used to interface the sensor system with alert system. In this system, the Ultrasonic sensor is used for garbage level detection by using ultrasonic sound waves and also LCD display for the indication of the dustbin alert. The GSM and the GPS system is implemented in this proposed system for the location detection of the dustbin and the alert will be send to the system automatically the stages of the system smart dustbin. Buzzer module is

used for communication purpose, to send message to the higher officials when the dustbin is not cleaned. Alert alarm is used to display the location of the dustbin is full at the control room. Arduino board is used to interface the sensor, alert module. The ultrasonic sensor is act as level detector. The output of level detector is connected to the microcontroller. Depending on the microcontroller program in first level the dustbin filled information is alerted.

1.5.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Wide detecting scope.
- Stable and long lifetime.
- Fast response and high sensitivity.
- A reduction in the number of waste collections needed by up to 80%, resulting in less manpower, emissions, fuel use and traffic congestion.
- A reduction in the number of waste bins needed.

CHAPTER 2

ENVIRONMENTAL SPECIFICATION

2.1 HARDWARE SPECIFICATION

- Microcontroller : Arduino
- Sensor : Ultrasonic sensor
- Connectivity : Battery
- Power Supply : Connecting wires
- Location system : GPS
- SMS connection : GSM
- ALERT : BUZZER

2.2 SOFTWARE SPECIFICATION

- Software : Arduino
- Language : C++

CHAPTER 3

PROGRAM DESIGN

3.1 MODULES

- **Arduino** UNO
- Ultrasonic Sensor
- GSM Module
- GPS Module
- LCD display
- buzzer alert
- Adapter

3.2. DETAILED DESCRIPTION

3.2.1 ULTRASONIC SENSOR

Ultrasonic sensors work by transmitting a pulse of sound, much like sonar detectors, outside the range of human hearing. This pulse travels away from the range finder in a conical shape at the speed of sound (340 m/s). The sound reflects off an object and back to the range finder. The sensor interprets this as an echo and calculates the time interval bet the in sending the signal and receiving the echo. This interval is then computed by a controller to determine the distance of the object.

DISTENCE = TROUND TRIP* V SOUND /2

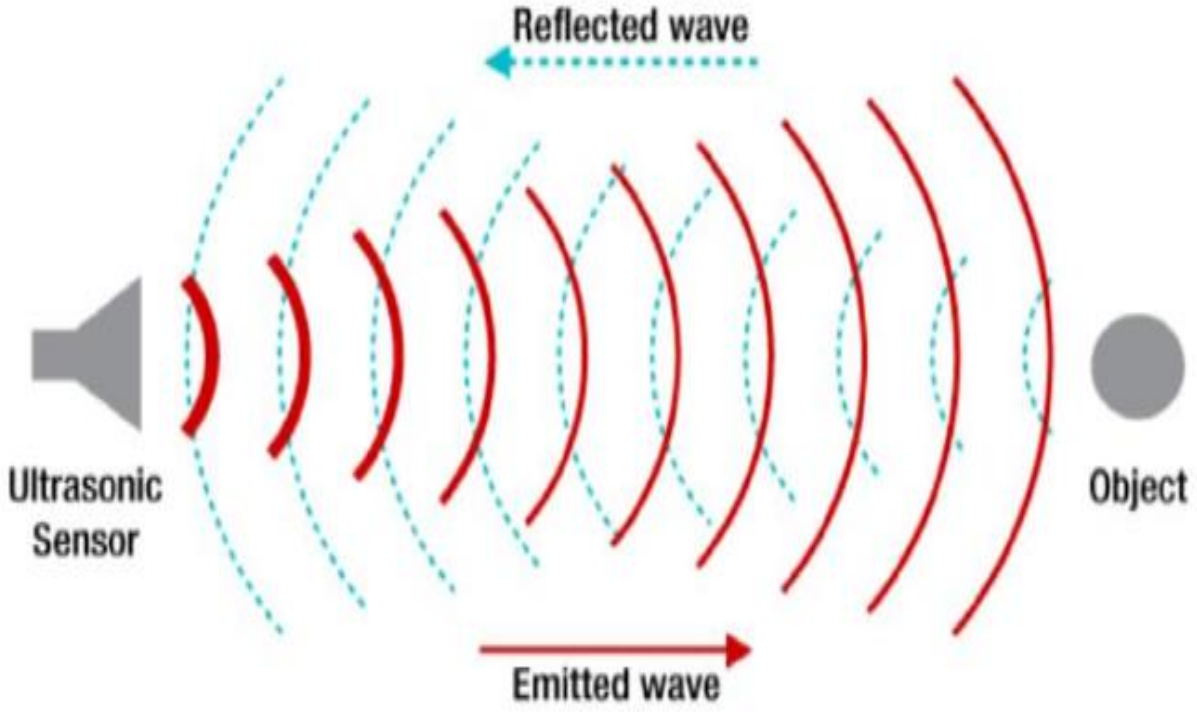


Fig 3.1 ULTRASONIC SENSOR

The ultrasonic sensor is a piezoelectric transducer, which is able to convert an electrical signal into mechanical vibrations, and mechanical vibrations into an electrical signal. Therefore, in a mono static approach, the ultrasonic sensor is a transceiver which operates as both a speaker and microphone at a single frequency.

The only requirement for ultrasonic sensing is that the target material is a solid or liquid. This enables contactless detection of :

- Metal
- Plastic
- Glass
- Wood
- Rocks

 Sand

 Oil

 Water

 Other hard, non-sound absorbent materials.

These materials are able to reflect sound back towards the sensor through the air. Certain objects can be more difficult to detect, like angled surfaces that direct the 4 echo away from the sensor, or permeable targets like sponge, foam, and soft clothing. These absorb more reflected ultrasonic energy.

TYPES OF ULTRASONIC SENSOR

- Ultrasonic diffuse proximity sensors.
- Ultrasonic retro-reflective sensors.
- Ultrasonic through-beam sensors

Ultrasonic sensors are also called as transceivers but more generally they are called as transducers since it works on a principle similar to radar or sonar. Its evaluate attributes of a target by interpreting the choose from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the it. Finally, it also calculates the time interval bet the in sending the signal and receiving the echo to determine the distance to the object

APPLICATIONS OF ULTRASONIC SENSOR

It uses to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.

• It used to measure the distance within a wide range of 2cm to 400cm.

• Used to map the objects surrounding the sensor by rotating it.

• Depth of certain places like THE is, pits etc. can be measured since the waves can penetrate through water.

ULTRASONIC SENSOR SPECIFICATIONS

- Length: 4.5 cm (1 ¾ in)
- Width: 2.0 cm (¾ in)
- Height: 1.4 cm (½ in)

- Typical price: Around $4
- Supply voltage: 5V
- Operating voltages: 3V or 5V (trigger), 5V all other I/O ports
- Working current: 15mA
- Operating range: 2 cm to 400 cm (1 in – 13 ft)
- Claimed precision: 0.3cm, more realistically: 1cm
- Measuring angle: 15 degrees
- Quiescent Current : : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1″ - 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
-

Description	Parallax PING	Generic HC-SR04
Range of Measurements	2 cm – 300 cm	2 cm – 400 cm
Resolution (Raging Accuracy)	N/A	3 mm
Ultrasonic Range	40 kHz	40 kHz
Supply Voltage	+5VDC	+5VDC
Operating Temperature	0º – 70 º C	0º – 60 º C
Size	22 mm H x 46 mm W x 16 mm D	20 mm H x 45 mm W x 15 mm D
Weight	9 grams	
Interface (PINs)	GRD : Ground 5V : +5VDC SIG : Signal (I/O pin)	GRD : Ground VCC: +5VDC Trig : Trigger (INPUT) Echo: Echo (OUTPUT)
Applications	▪ Security systems ▪ Interactive animated exhibits ▪ Parking assistant systems ▪ Robotic navigation	▪ Distance measurement. ▪ Distance Ranging. ▪ Robotics for mapping. ▪ Colored Line sensing ▪ Object/obstacle detection
Effectual Angle	Not Specified	< 15º
Measuring Angle	Not Specified	30 º
Input Trigger Pulse	2 µs (min), 5 µs typical	10 µs
Echo Holdoff	750 µs	Not Specified
Burst Frequency	200 µs @ 40 kHz	Not Specified
Echo Return Pulse (min)	115 µs	Not Specified
Echo Return Pulse (Max)	18.5 ms	Not Specified
Delay before next measurement	200 µs	Not Specified

3.2.2 ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power pin 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 pin 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.

The Arduino project started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller, at a cost that was a considerable expense for many students. In 2003, Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas, who are known for work on the Processing language. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing, and library functions to easily program the microcontroller.[10] In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forked the project and renamed it Arduino. Early arduino boards used the FTDI USB-to-UART serial chip and an ATmega168.[10] The Uno differed from all preceding boards by featuring the ATmega328P microcontroller and an ATmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

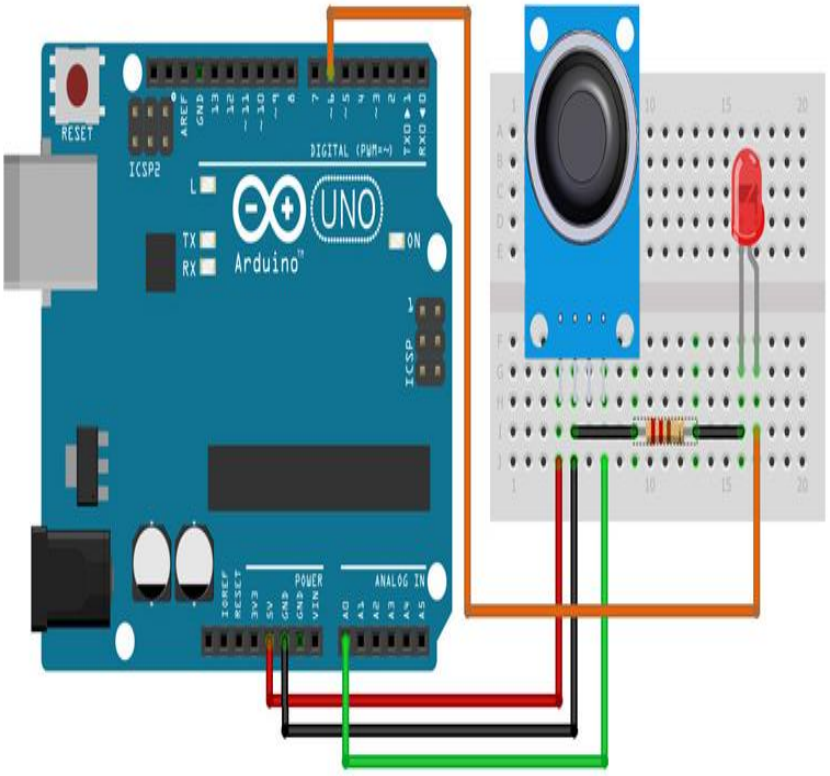


Fig 3.2 ARDUINO UNO

Power pin

The Arduino Uno can be power pined via the USB connection or with an external power pin supply. The power pin source is selected automatically. External (non-USB) power pin can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power pin jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POTHE R connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, hoTHEver, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pin pins are as follows:

 VIN: The input voltage to the Arduino board when it's using an external power pin source (as opposed to 5 volts from the USB connection or other regulated power pin source). You can

supply voltage through this pin, or, if supplying voltage via the power pin jack, access it through this pin.

 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power pin either from the DC power pin jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. THE don't advise it.

 3V3. A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

 GND. Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

 Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

 External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

 PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.

 SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

 LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference() function. Additionally, some pins have specialized functionality:

 TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

 AREF - Reference voltage for the analog inputs. Used with analog Reference().

 Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. See also the mapping betTHEen Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical. Communication The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U firmware uses the standard USB COM drivers, and no external driver is needed. Otherer, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details,

see the reference and tutorials. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

 On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

 On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the loTHEring of DTR can be THE ll-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following halfsecond or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board

receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

3.2.3 GPS

The Global Positioning System consists of 24 satellites, that circle the globe once every 12 hours, to provide worldwide position, time and velocity information. GPS makes it possible to precisely identify locations on the earth by measuring distance from the satellites. Developed by the Department of Defense in 1973, GPS was originally designed to assist soldiers and military vehicles, planes, and ships in accurately determining their locations world- wide. Today, the uses of GPS have extended to include both the commercial and scientific worlds.

Fig 3.3 GSM IOT Kit

The exact location of a moving object is detected by at least three of these GPS satellites. The information is transmitted to a server via a wireless radio standard: The satellite broadcasts its position and time using coded radio signals. it is used to locate the exact location of the dustbin.

3.2.4 GSM

Extended coverage GSM IoT (EC-GSM-IoT) is a standard-based Low Power Wide Area technology. It is based on eGPRS and designed as a high capacity, long range, low energy and low complexity cellular system for IoT communications. A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. The modem (modulator-demodulator) is a critical part here.

Fig 3.4 GSM IOT Connection

GSM stands for Global System for Mobile Communication. It is a digital cellular technology used for transmitting mobile voice and data services. Important facts about the GSM are given below −

- The concept of GSM emerged from a cell-based mobile radio system at Bell Laboratories in the early 1970s.
- GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard.
- GSM is the most widely accepted standard in telecommunications and it is implemented globally.
- GSM is a circuit-switched system that divides each 200 kHz channel into eight 25 kHz time-slots. GSM operates on the mobile communication bands 900 MHz and 1800 MHz

in most parts of the world. In the US, GSM operates in the bands 850 MHz and 1900 MHz.

- GSM owns a market share of more than 70 percent of the world's digital cellular subscribers.
- GSM makes use of narrowband Time Division Multiple Access (TDMA) technique for transmitting signals.
- GSM was developed using digital technology. It has an ability to carry 64 kbps to 120 Mbps of data rates.
- Presently GSM supports more than one billion mobile subscribers in more than 210 countries throughout the world.
- GSM provides basic to advanced voice and data services including roaming service. Roaming is the ability to use your GSM phone number in another GSM network.

GSM digitizes and compresses data, then sends it down through a channel with two other streams of user data, each in its own timeslot.

There are three primary types of touch switches: capacitive, resistance and piezo, each of which use a different mechanism. Capacitive touch switches work by measuring capacitance — just like their capacitive touchscreen counterparts.

The ultrasonic sensor connection work will be done with the identification of the distant of the object. The distant verification will be done with the blind stick verification. The sensor has 4 pins. ***VCC*** and ***GND*** go to ***5V*** and ***GND*** pins on the Arduino, and the ***Trig*** and ***Echo*** go to any digital Arduino pin. Using the ***Trig*** pin we send the ultrasound wave from the transmitter, and with the ***Echo*** pin we listen for the reflected signal.

If we receive a reflected pulse, the Echo pin will go down sooner than those 38ms. According to the amount of time the Echo pin was HIGH, we can determine the distance the sound wave traveled, thus the distance from the sensor to the object.

For that purpose we are using the following basic formula for calculating distance:

Distance = Speed x Time

That actually know both the speed and the time values. The time is the amount of time the Echo pin was HIGH, and the speed is the speed of sound which is 340m/s. There's one additional step we need to do, and that's divide the end result by 2. that's because we are measuring the duration the sound wave needs to travel to the object and bounce back.

Let's say the Echo pin was HIGH for 2ms. If we want the get the distance result in cm, we can convert the speed of sound value from 340m/s to 34cm/ms.

***Distance = (Speed x Time) / 2* = (34cm/ms x 1.5ms) / 2 = 25.5cm.**

So, if the Echo pin was HIGH for 2ms (which we measure using the ***pulseIn()*** function), the distance from the sensor to the object is 34cm.

The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board.

3.2.5 LCD DISPLAY

The LCD (Liquid Crystal Display) is a type of display that uses the liquid crystals for its operation. Here, we will accept the serial input from the computer and upload the sketch to the Arduino. The characters will be displayed on the LCD. The LiquidCrystal library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

Fig 3.5 LCD DISPLAY

The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins:

A register select (RS) pin that controls where in the LCD's memory you're writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next. A Read/Write (R/W) pin that selects reading mode or writing mode

An Enable pin that enables writing to the registers

8 data pins (D0 -D7). The states of these pins (high or low) are the bits that you're writing to a register when you write, or the values you're reading when you read.

There's also a display contrast pin (Vo), power supply pins (+5V and GND) and LED Backlight (Bklt+ and BKlt-) pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. The LiquidCrystal Library simplifies this for you so you don't need to know the low-level instructions.

The Hitachi-compatible LCDs can be controlled in two modes: 4-bit or 8-bit. The 4-bit mode requires seven I/O pins from the Arduino, while the 8-bit mode requires 11 pins. For displaying text on the screen, you can do most everything in 4-bit mode, so example shows how to control a 16x2 LCD in 4-bit mode.

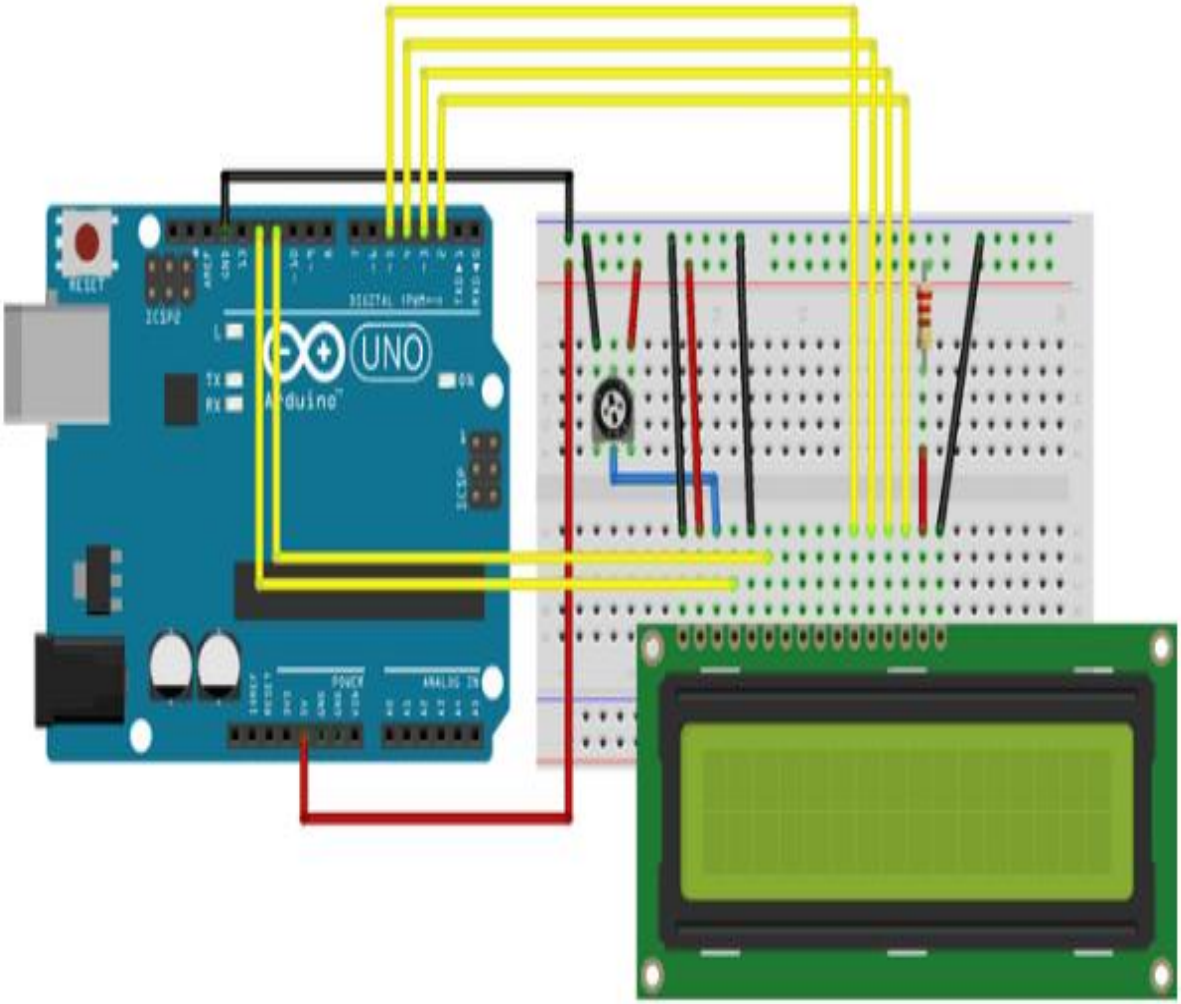


Fig 3.6 LCD circuit connection

3.2.6 BUZZER ALERT

The GSM module has an on-board micro controller acting as watchdog for the system, should any problem with the network connectivity be detected the unit will be automatically re-set and reconnected to the network. The on-board LEDs indicate power on (green), SMS message sent (yellow) and SMS message failed (red). In case of any garbage fill the concern of the user can done a extraction of data with the complete verification system with the buzzer alert.

3.2.7 POWER ADAPTER AND CONNECTING CABLES

Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 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. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

Fig 3.7 POWER ADAPTER AND CONNECTING CABLES

After wiring and attaching all the devices and setting up to the Smart Dustbin, now observe all the important setup whether they are well connected or something missed. After connection set up now next step is to submit/upload code in Arduino and supply power to the circuit. When system is powered ON, Arduino keeps monitoring for any things that come near the sensor at give range. When Ultrasonic sensor detect any object for example like hand or others, here Arduino calculates its distance and if it less than a certain predefines value than servo motor get activate first and with the support of the extended arm of the lid. Lid will open for a given time than it will automatically close.

3.3 SYSTEM ARCHITECTURE

System architecture can be defined as a structure composed of components, and rules characterizing the interaction of these components. System architecture diagram is a starting point of product description visualization – a product structure overview. Three types of system architectures are identified, integrated, distributed and mixed, (partly integrated and partly distributed). It is shown that the type of interfaces defines the type of architecture. Integrated systems have more interfaces, which furthermore are vaguely defined. An application architecture diagram provides a high-level graphical view of the application architecture, and helps you identify applications, sub-applications, components, databases, services, etc, and their interactions. See Application and Business Services (EAM). System - A packaged application.

If you restricted yourself to four layers, they may be defined as:

1) Algorithm,

2) Programming language/compiler,

3) processor/memory,

4) I/O. Other abstraction definitions may contain three layers:

1) Application,

2) System software,

3) Hardware

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.

- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software.

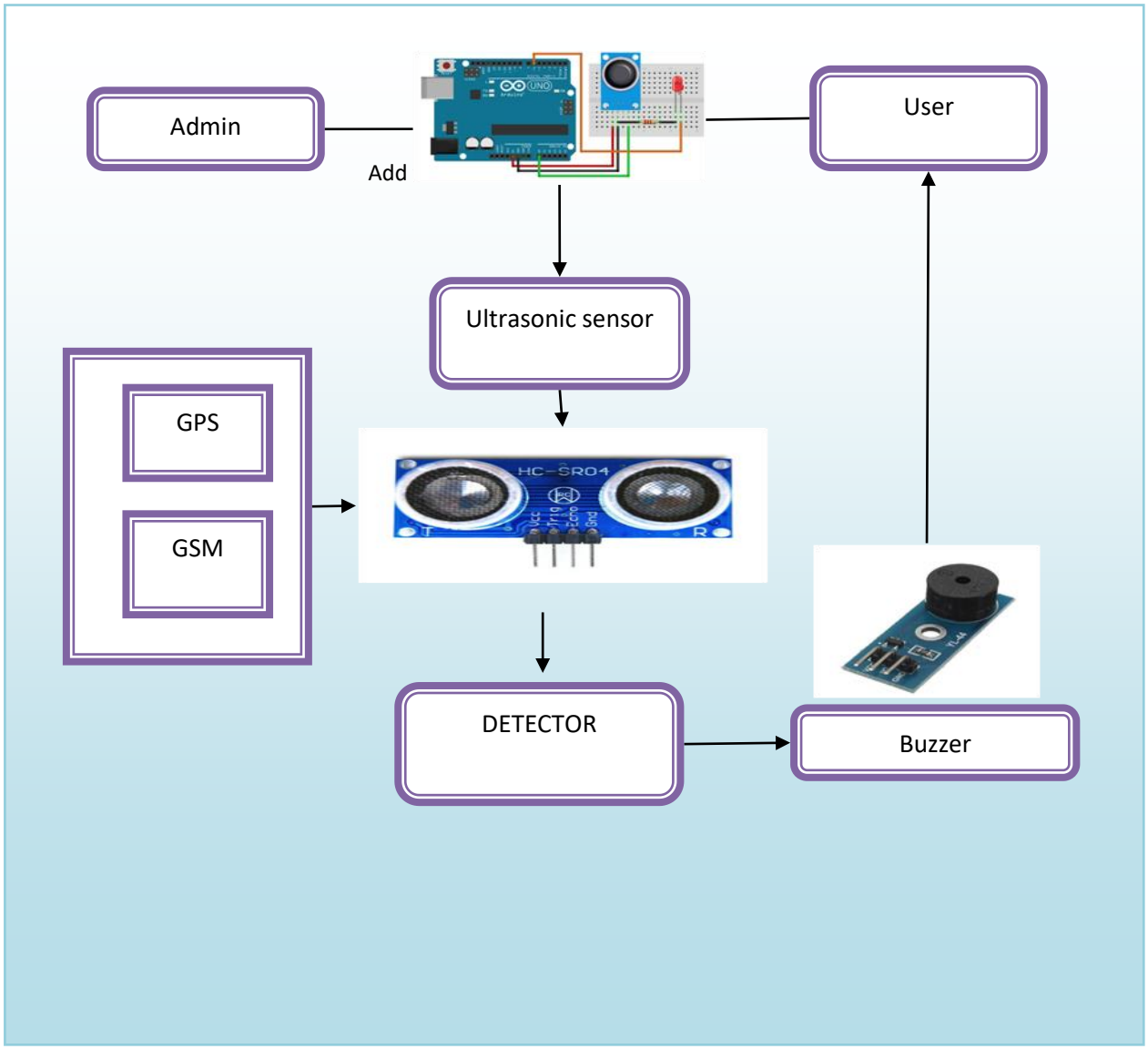


Fig 3.8 System Architecture

3.4USE CASE DIAGRAM

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. The actor can be a human or other external system. An actor in the Unified Modeling Language (UML) "specifies a role played by a user or any other system that interacts with the subject." "An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject." UML Use Case Include. Use case include is a directed relationship between two use cases which is used to show that behavior of the included use case (the addition) is inserted into the behavior of the including (the base) use case.

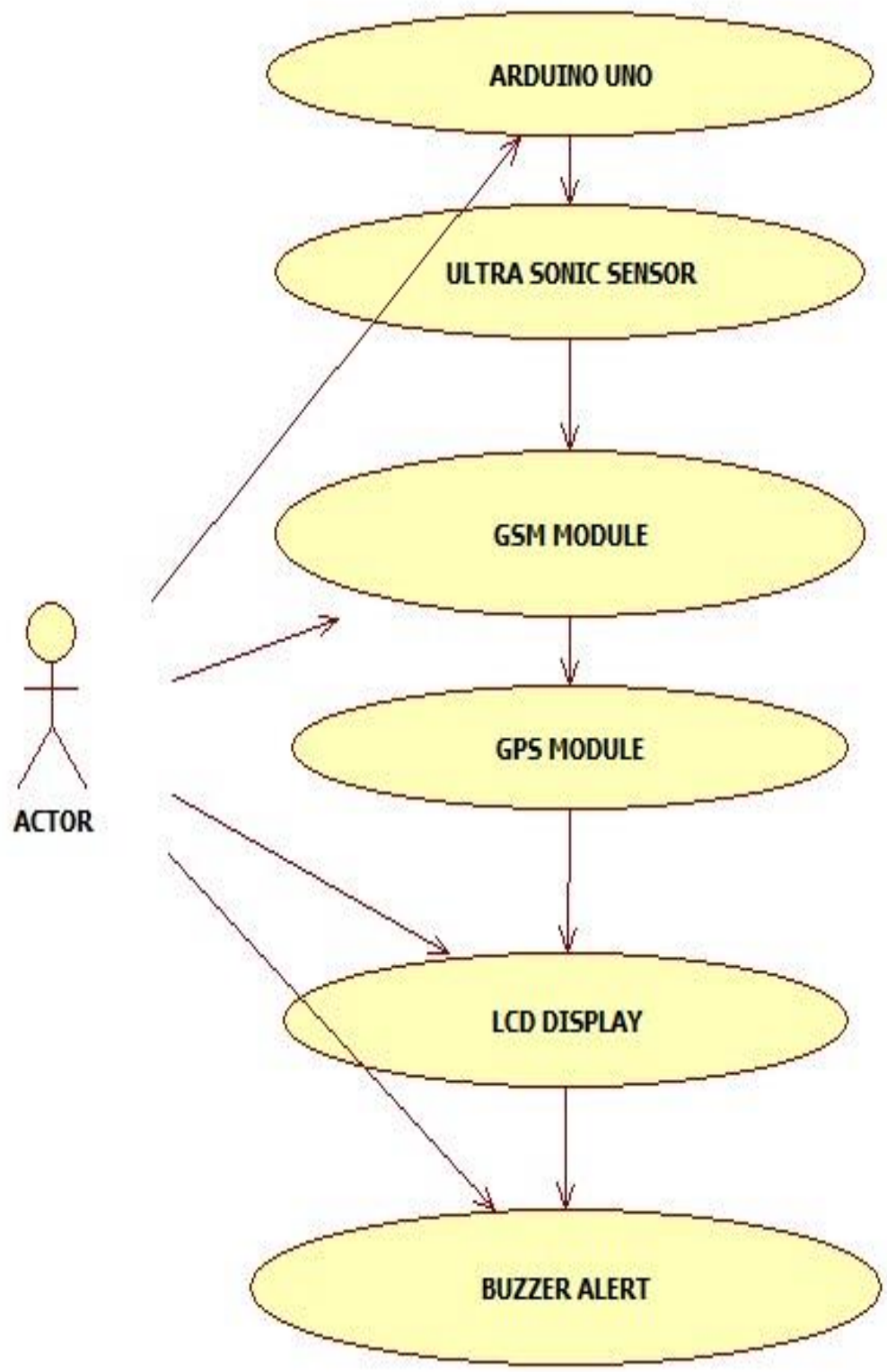


Fig 3.9 Use case diagram

3.5 FEASIBILITY STUDY

In this feasibility study, were expected to tackle a real world problem. In a nutshell, we aimed at providing a technical solution to a real world problem which would help society in some form or the other. The purpose of this project would be to identify ways and means to make the lives of blind people much easier. To be specific, this project will help blind people identify obstacles and make their next movement according to presence or absence of obstacle. At the same time, we realize there may be circumstances wherein the blind person may be bewildered about his next movement. In such a situation, he can press the e-SOS (Save Our Souls) button on his stick which will enable the process of live streaming system. (Here, live streaming indicates the nearby environment of the blind person) The streaming signal with location is sent to his family member's Android phone via an Android application. An e-SOS is Morse coded distress signal. In simple words, SOS is distress call/ emergency signals which the blind person sends to his family member.

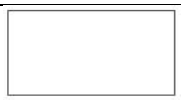
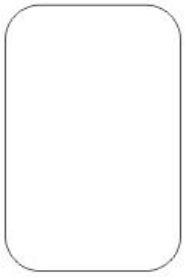
There are three types of Internet of Things applications that have the lowest usage percentage compared to other applications. The least-used application is the use of robotics in construction industry. this happens because of the complexity of project and due to that the repetition of work is less. The another reason is that, if robotics is used then the threat of job security to labors and engineers may occur. Second less used application is Primavera for project management purpose. This happens because of lack of knowledge about the software and the alternative software Microsoft Project is easy to use, so the Primavera is not adopted by most of the firms.The third application is the use of sensor technology to manage the waste. Waste management is the main thing to take care but unfortunately there is less awareness and adaptive steps are taken to look into it. However, the findings show that the use of Internet of Things applications is still under-utilized in the construction industry. This is because there are only 14 types of applications that are higher in usage rates than those that do not use them, and 22 applications are less used.The evaluation of the respondents' knowledge of the Internet of Things application found that the majority of respondents had known each type of Internet of Things application. This proves that the construction industry players are aware of the presence of this application although its usage is still less because of lack of skills to use, lack of interest in change, lack of budget to purchase and install new technologies. To increase the use of

Internet of things application in construction industry, implementation and encouragement should be provided to each players in construction industry.

3.6 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols:

Symbol	**Description**
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

LEVEL 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

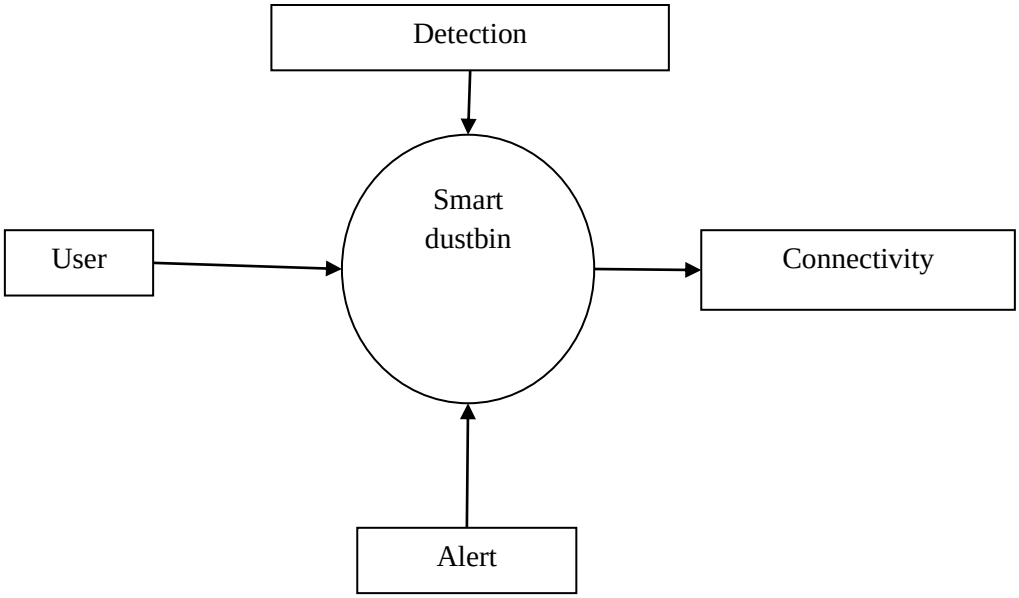


Fig 3.10 level 0 DFD

LEVEL 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes. A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into sub processes that can then be analyzed and improved on a more intimate level.

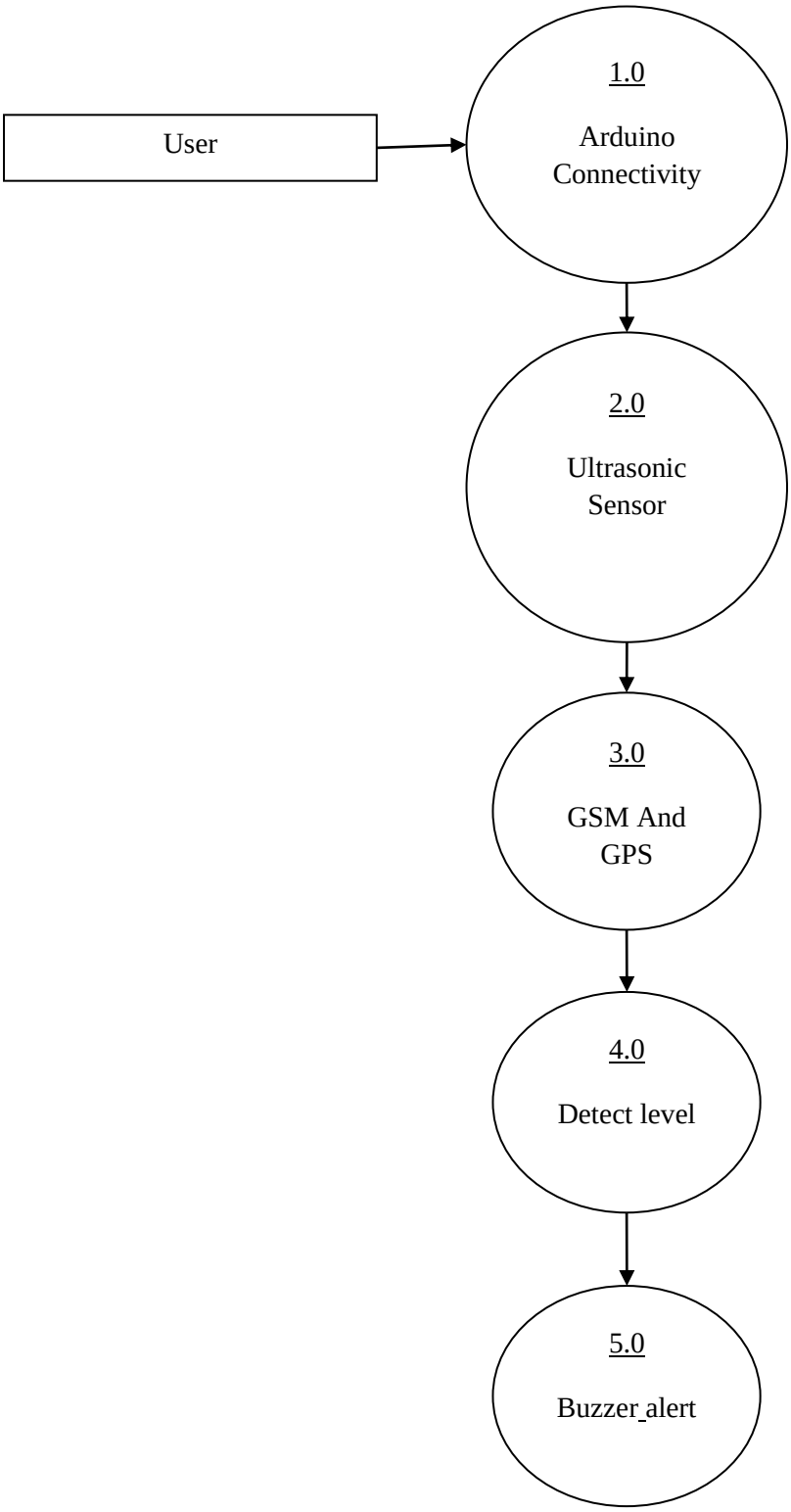


Fig 3.11 Level 1 DFD

CHAPTER 4

CONCLUSION AND FUTURE ENHANCEMENTS

4.1 CONCLUSION OF THE PROJECT

With the second largest population in the world , India is a developing country which has large amount of waste produced every day. There is production of 62 million tons of municipal solid waste (MSW) each year in urban India.70% of which is collected and 20% get treated. Because of nature of different types of waste it is very difficult to discard the waste. Segregation plays a very important role by reducing the waste by reusing. Treating the waste also becomes easier if segregation is done at the base level. The fig shows the increase in the waste generated from the year 2016, to 2050 across various parts of the world. Hence this indicates the increase in requirement for efficient processing of the waste to maintain ecological balance. The model developed in this paper is efficient and durable since it require less power for its operation and no human supervision. The model can also detect when the bin is full asking the authorities to come and collect .This efficiently reduces the man power, wastage of time and fuel required by collecting van. This model fits perfectly as a replacement to older bins and works well with the idea of smart city. With the future scope the bins can be made solar powered with better segregation techniques like digital image processing and the waste collected in the bins can be made to increase the storage capacity.

Now that all the devices have been wired, attached, and set up on the Smart Dustbin, it is important to carefully monitor and assess the entire setup. Either they have good connections or something is being overlooked. The subsequent action to establish a connection is to upload or submit the code to the Arduino board and provide it with a power supply. the loop. The Arduino continuously watches for any objects within a predetermined range when it is switched on. When An ultrasonic sensor is capable of detecting different objects, such as a hand or other objects. An Arduino device is used to calculate the distance of the object, and if it's below a certain threshold, the sensor will register it. The ultrasonic sensor is triggered initially based on a predefined value, and it is assisted by the extended arm of the lid. The lid will remain open for a specific duration. It will close on its own.

4.2 FUTURE ENHANCEMENT

The waste materials can be segregated into biodegradable, non-bio degradable and metals by using more sensors. The discarded things can be processed to extract or recover materials in an effective way and resources or convert them to energy as usable heat, electricity, fuels. The large-scale introduction of automatic waste management in villages, platforms, hospitals, industries, etc. Real time monitoring and controlling of waste management by using IoT. A prediction system by the given data to predict the variation in the amount of waste and to adjust the timing of management. This method described above is to move towards IOT implantation. All smart dustbin methods based on IOT are very helpful for cleaning the waste. An Ultrasonic sensor utilize the maximum peaks of rubbish on a dustbin. Many devices may be used in a variety of systems.

5.REFERENCES

5.1 BOOK REFERENCE:

[1] BelalChowdhury, Morshed U Chowdhury "RFID- based Real-time Smart Waste Management System", in 2007 Australasian Telecommunication Networks and Applications Conference December 2nd – 5th 2007, Christchurch, New Zealand.

[2] SauroLonghi, DavideMarzioni†, EmanueleAlidor, Gianluca Di Bu` o, Mario Prist, Massimo Grisostomi and MatteoPirro "Solid Waste Management Architecture using Wireless Sensor Network technology" in Universit`aPolitecnicadelle Marche, Dipartimento di Ingegneriadell'Informazione Via BrecceBianche. snc, 60131 Ancona, Italy 2012.

[3] ShubhamThakker, R.Narayanamoorthi, "Smart and Wireless Waste Management" in IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ,2015.

[4] Andrei Borozdukhin, Olga Dolinina and VitalyPechenkin, "Approach to the Garbage Collection in the Smart Clean City Project" in, Yuri Gagarin State Technical University of Saratov, Saratov, Russia 2016.

[5] Thompson A.F, Afolayan A.H, Ibidunmoye E.O, "Application of Geographic Information System to Smart Garbage Monitoring and Clearance System using Internet of Things 189 Solid Waste Management" in Pan African International Conference on Information Science, Computing and Telecommunications (2013).

REFERENCE:

1. https://www.irejournals.com/formatedpaper/1701045.pdf
2. https://www.rcciit.org/students_projects/projects/aeie/2018/GR11.pdf
3. https://iopscience.iop.org/article/10.1088/1757-899X/263/4/042027/pdf

6.APPENDIX

6.1 SOURCE CODE

```
#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

SoftwareSerial mySerial(10,11);

int msg_count=0;

int msg_duration=5;

int minuit=60;

int msg_check=msg_duration*minuit;

char msg;

char call;

int sms=0,sms1=0;

//////////////////////////////////////////////////

long distance=0;

////////////////////////////////////////////////////////lcd

LiquidCrystal_I2C lcd(0x27,20,4);

////////////////////////////////////////////////////////location

String latitude = "10.8225",longitude = "78.6865";
```

```
int RXPin = 3;
int TXPin = 2;
int GPSBaud = 9600;
TinyGPSPlus gps;
SoftwareSerial gpsSerial(RXPin, TXPin);
////////////////////////////////////////////////////////////////
const int pingPin = 7; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6; // Echo Pin of Ultrasonic Sensor
////////////////////////////////////////////////////////////////
void setup()
{
  Serial.begin(9600);
mySerial.begin(9600);
  gpsSerial.begin(GPSBaud);
  ////////////////////////////////////////////////////
  lcd.init();                    // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(1,0);
```

```
  lcd.print("loading...");
  lcd.setCursor(1,1);
  lcd.print("Dustpin");
  ////////////////////////////////////


// start communication with the SIM900A in 9600
//SendMessage();
}


void loop()
{
  lcd.init();                    // initialize the lcd
  lcd.backlight();
  lcd.setCursor(1,0);
  lcd.print("");
  lcd.setCursor(1,1);
  lcd.print("");


   distance=ultrasonic_sensor();
  Serial.println(distance);
```

```
  // Serial.print("distance");


  get_location();
//  delay(200);
//display_message_lcd("level","Full");
  if (distance<10)
  {
distance=ultrasonic_sensor();
  Serial.println(distance);
  // Serial.print("distance");


  display_message_lcd("Dustpin","Full");
  //
  Serial.println(msg_count);
  if(msg_count<2)
  {
     SendMessage("Dustpin Full : "+latitude+"-"+longitude);
    // Sendcall();
  }
  else if(msg_count==50)
```

```
distance=ultrasonic_sensor();
  Serial.println(distance);
  // Serial.print("distance");

  {
    msg_count=0;
Sendcall();
    msg_count=0;
  }
  else if(msg_count>50)
  {

   msg_count=0;
  }

  msg_count=msg_count+1;
  }
  else if((distance>=10)and(distance<=20))
  {
  display_message_lcd("Dustpin","Half");
```

```
  }
  else if(distance>20)
  {
    display_message_lcd("Dustpin","Low");
  }
}
 long microsecondsToInches(long microseconds) {
   return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
   return microseconds / 29 / 2;
}
void display_message_lcd(String message,String message1)
{
  lcd.setCursor(1,0);
  lcd.print(message);
  lcd.setCursor(1,1);
  lcd.print(message1);
```

```
}
long ultrasonic_sensor()
{
long duration, inches, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
//  Serial.print(inches);
//  Serial.print("in, ");
//  Serial.print(cm);
//  Serial.print("cm");
//  Serial.println();
//  delay(100);
```

```
  return cm;
}
void get_location()
{


          latitude = String(gps.location.lat(),6);
         longitude = String(gps.location.lng(),6);


        long l1 = (gps.location.lat());
         long l2 = (gps.location.lng());
    //     Serial.println(l1);
      //   if(l1==0)
        // {
          // Serial.println("reading");
//            float data1 = random(0, 10); // generate the integers
//            float data2 = random(0, 100); // generate the numbers after the decimal
point
//            float data = data1 + data2 / 100;
//            Serial.println(data);
//            long l11=10.82253;
```

```
//            long l22=78.6865;
//             Serial.println(l11);
//            l1=l11+(long)random(1.150, 5.90)+0.12;
//            l2=l22+(long)random(1.45, 5.50)+0.12;
//            latitude=String(l1);
//            longitude=String(l2);
//            Serial.println(latitude);
        // }
        // else
         //{
           //Serial.println("loading");
        // }


           lcd.setCursor(1,0);
  lcd.print(message);
  lcd.setCursor(1,1);
  lcd.print(message1);


        if (gpsSerial.available() > 0)
```

```
    if (gps.encode(gpsSerial.read()))
      displayInfo();


  // If 5000 milliseconds pass and there are no characters coming in
  // over the software serial port, show a "No GPS detected" error
//  if (millis() > 5000 && gps.charsProcessed() < 10)
//  {
//    Serial.println("No GPS detected");
// //   while(true);
//  }
}
void SendMessage(String message)
{
  Serial.println(message);
  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);  // Delay of 1000 milli seconds or 1 second
  mySerial.println("AT+CMGS=\"+917339333830\"\r"); // Replace x with mobile
number
  delay(1000);
  mySerial.println(message);// The SMS text you want to send
```

```
  delay(100);
  mySerial.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
}
void Sendcall()
{
  mySerial.println("ATD+ +918270976658;"); //  change ZZ with country code and
xxxxxxxxxxx with phone number to dial
//updateSerial();
delay(25000); // wait for 20 seconds...
mySerial.println("ATH"); //hang up
//updateSerial();
}
void displayInfo()
{
  if (gps.location.isValid())
  {
    Serial.print("Latitude: ");
    Serial.println(gps.location.lat(), 6);
    Serial.print("Longitude: ");
```

```
  Serial.println(gps.location.lng(), 6);
  Serial.print("Altitude: ");
  Serial.println(gps.altitude.meters());
}
else
{
  Serial.println("Location: Not Available");
}

Serial.print("Date: ");
if (gps.date.isValid())
{
  Serial.print(gps.date.month());
  Serial.print("/");
  Serial.print(gps.date.day());
  Serial.print("/");
  Serial.println(gps.date.year());
}
else
{
```

```
  Serial.println("Not Available");
}

Serial.print("Time: ");
if (gps.time.isValid())
{
  if (gps.time.hour() < 10) Serial.print(F("0"));
  Serial.print(gps.time.hour());
  Serial.print(":");
  if (gps.time.minute() < 10) Serial.print(F("0"));
  Serial.print(gps.time.minute());
  Serial.print(":");
  if (gps.time.second() < 10) Serial.print(F("0"));
  Serial.print(gps.time.second());
  Serial.print(".");
  if (gps.time.centisecond() < 10) Serial.print(F("0"));
  Serial.println(gps.time.centisecond());
}
else
{
```

```
    Serial.println("Not Available");
  }


  Serial.println();
  Serial.println();
  delay(1000);
}
```

Lcd display coding

```
/*
  LiquidCrystal Library - Hello World

 Demonstrates the use a 16x2 LCD display.  The LiquidCrystal
 library works with all LCD displays that are compatible with the
 Hitachi HD44780 driver. There are many of them out there, and you
 can usually tell them by the 16-pin interface.

 This sketch prints "Hello World!" to the LCD
 and shows the time.
```

The circuit:

* LCD RS pin to digital pin 12

* LCD Enable pin to digital pin 11

* LCD D4 pin to digital pin 5

* LCD D5 pin to digital pin 4

* LCD D6 pin to digital pin 3

* LCD D7 pin to digital pin 2

* LCD R/W pin to ground

* LCD VSS pin to ground

* LCD VCC pin to 5V

* 10K resistor:

* ends to +5V and ground

* wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008

by David A. Mellis

library modified 5 Jul 2009

by Limor Fried (http://www.ladyada.net)

example added 9 Jul 2009

by Tom Igoe

modified 22 Nov 2010

by Tom Igoe

modified 7 Nov 2016

by Arturo Guadalupi

This example code is in the public domain.

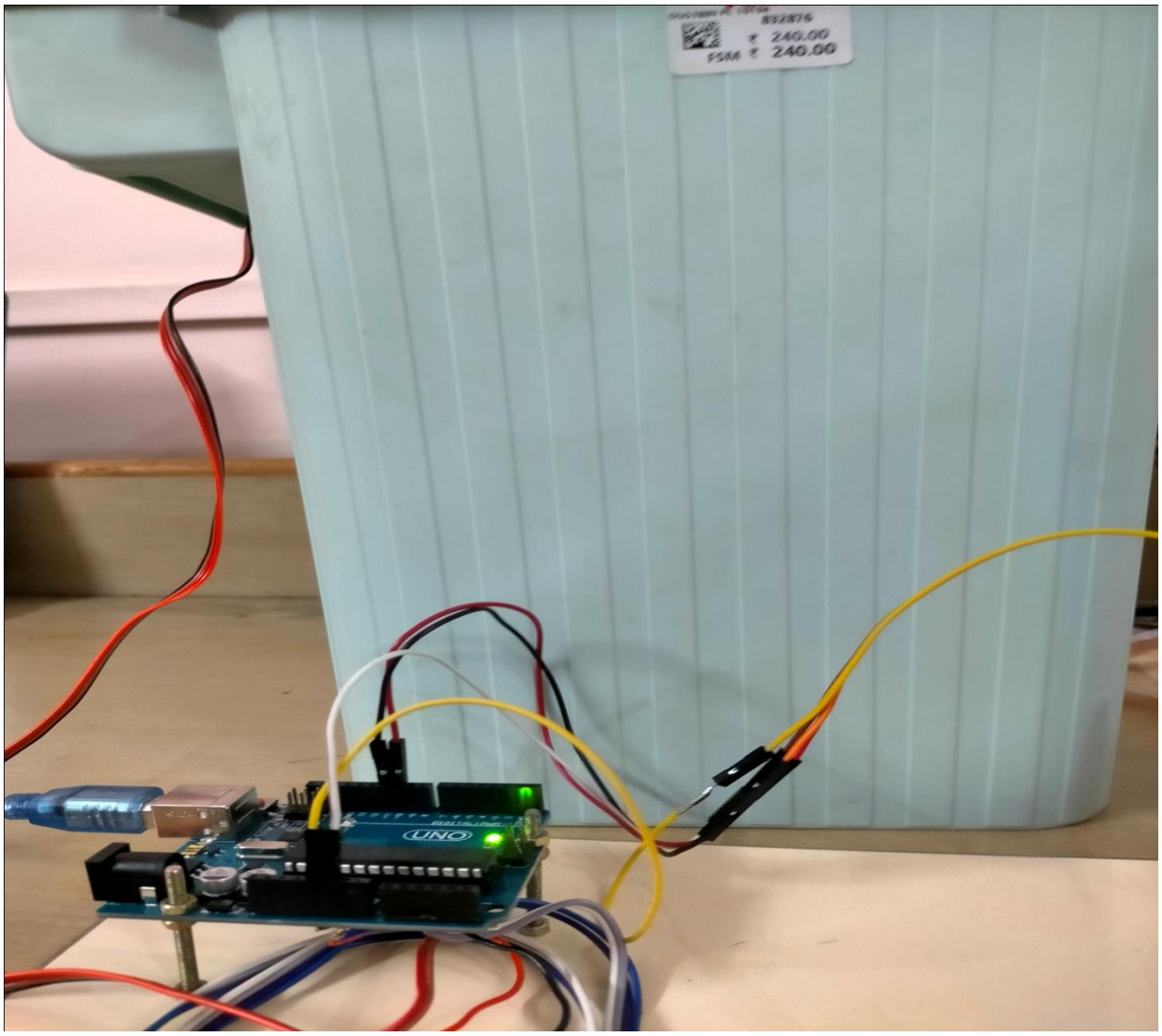
https://docs.arduino.cc/learn/electronics/lcd-displays

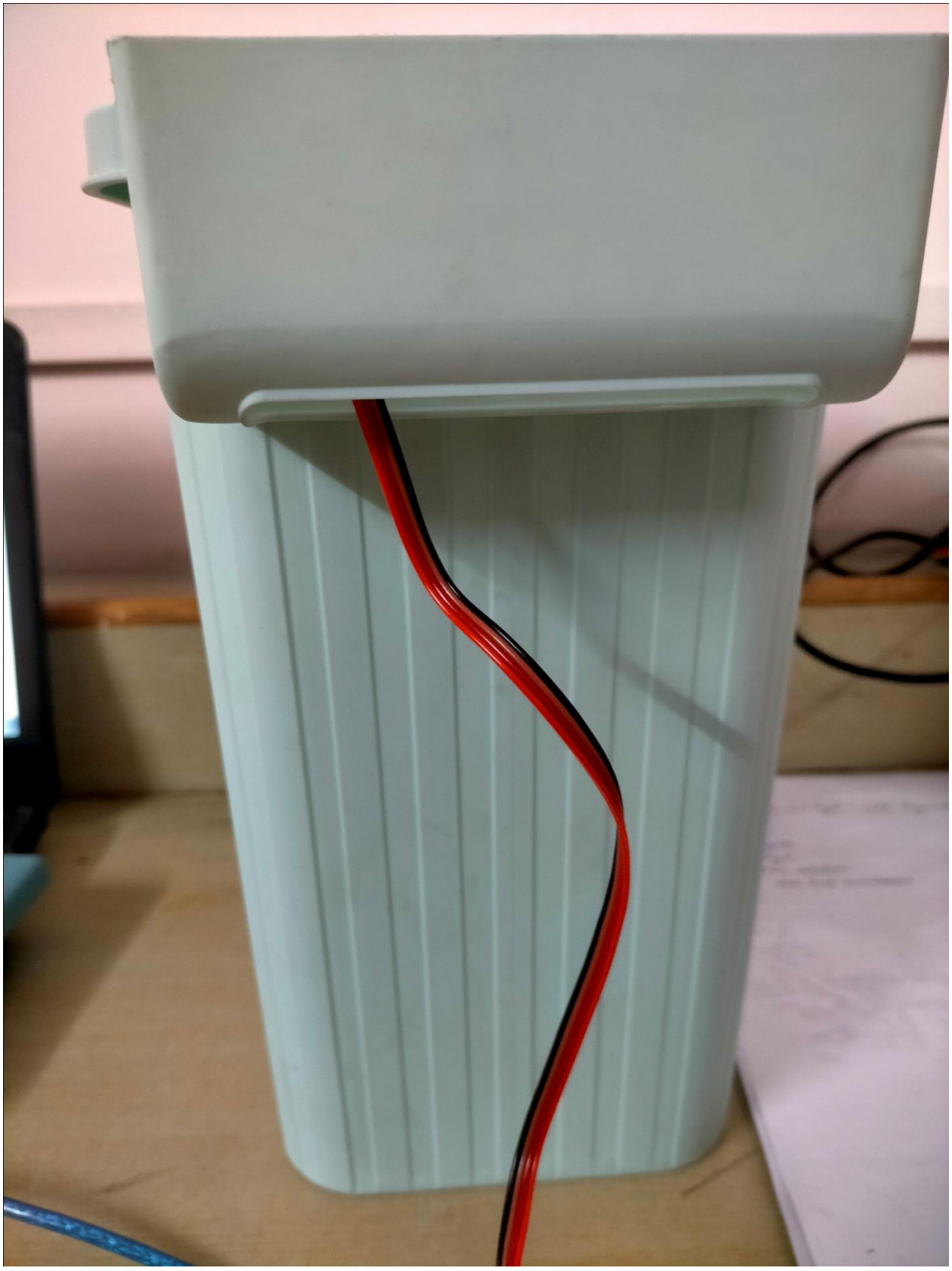
*/

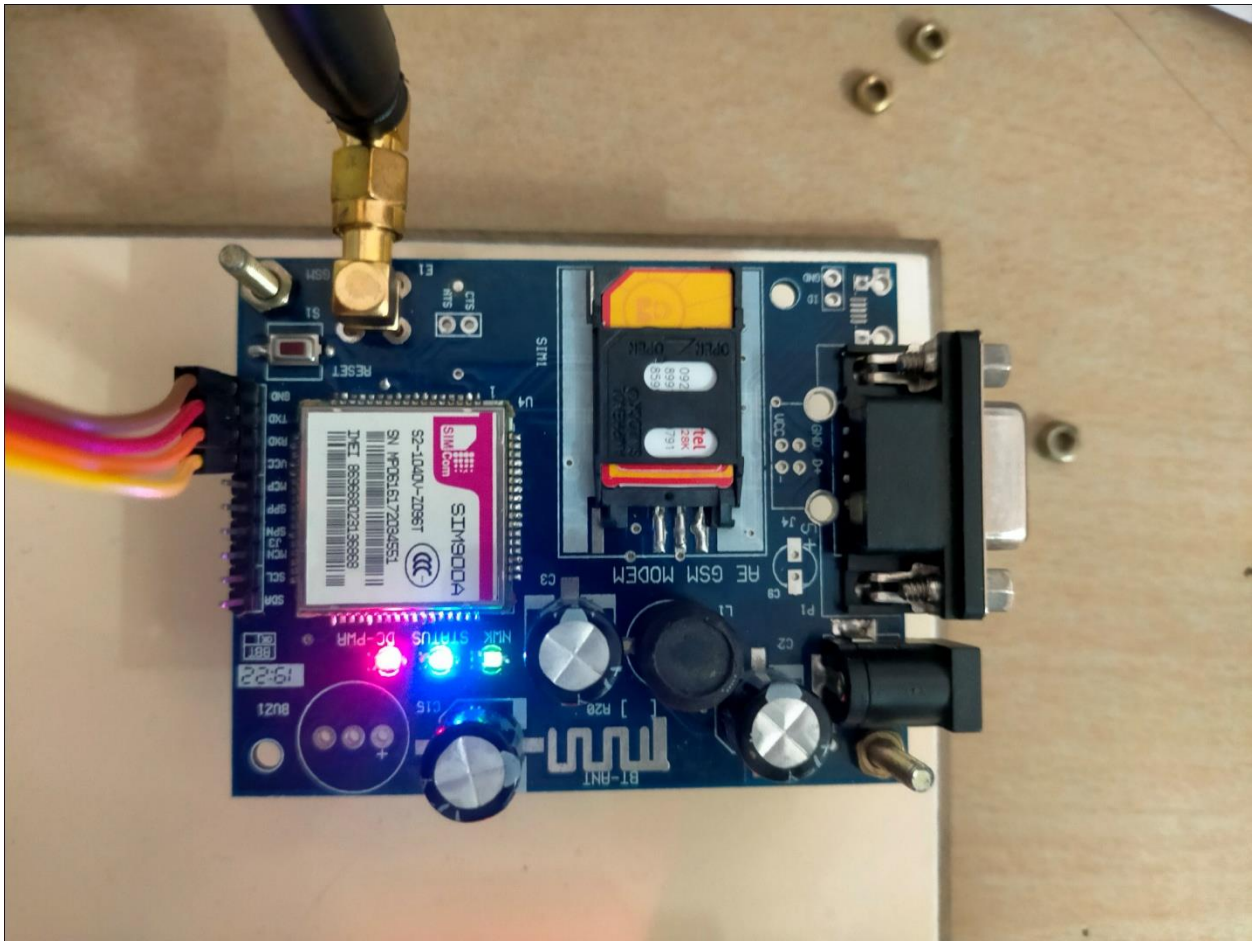
```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

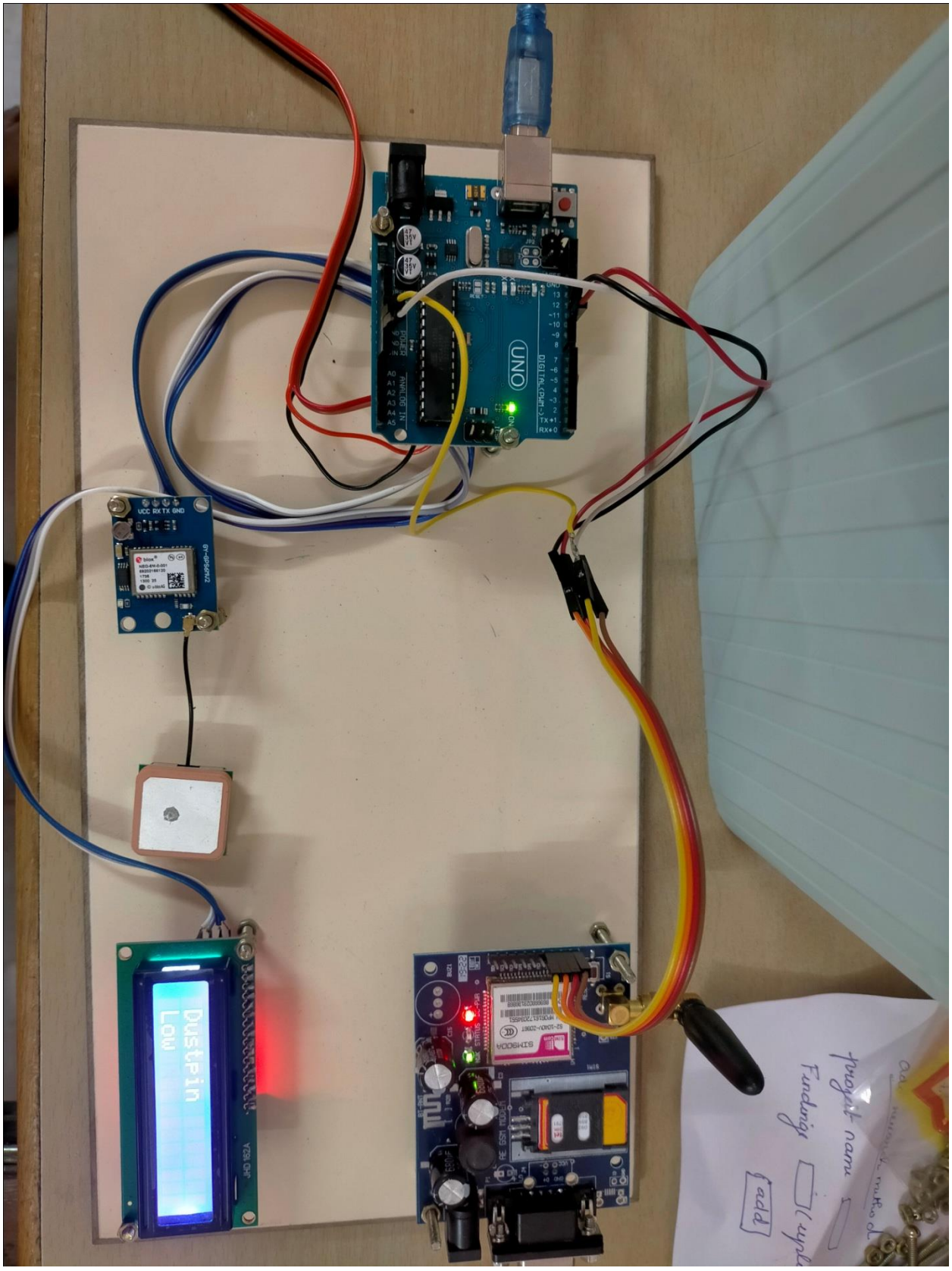
```
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

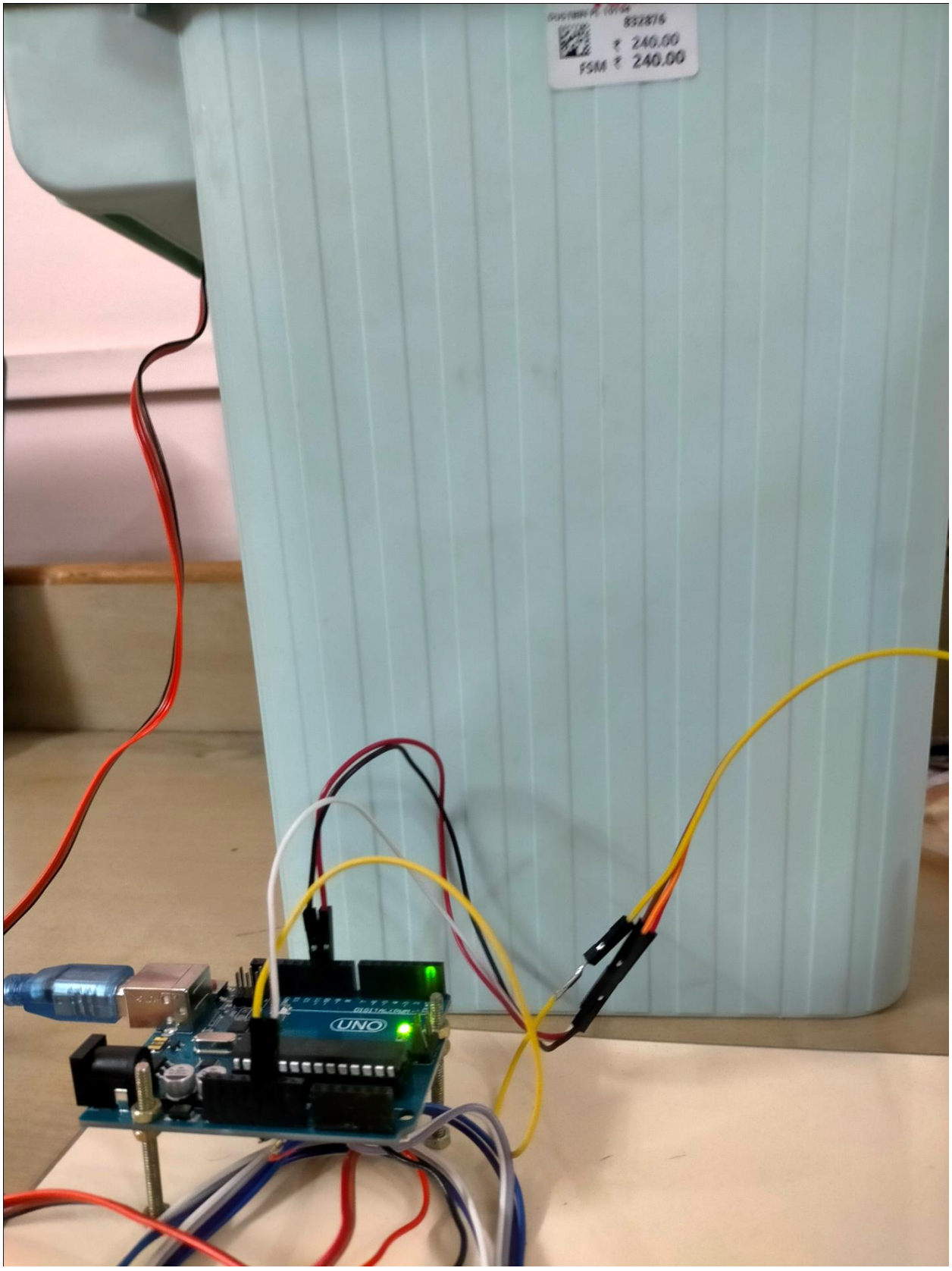
6.2 SCREENSHOTS



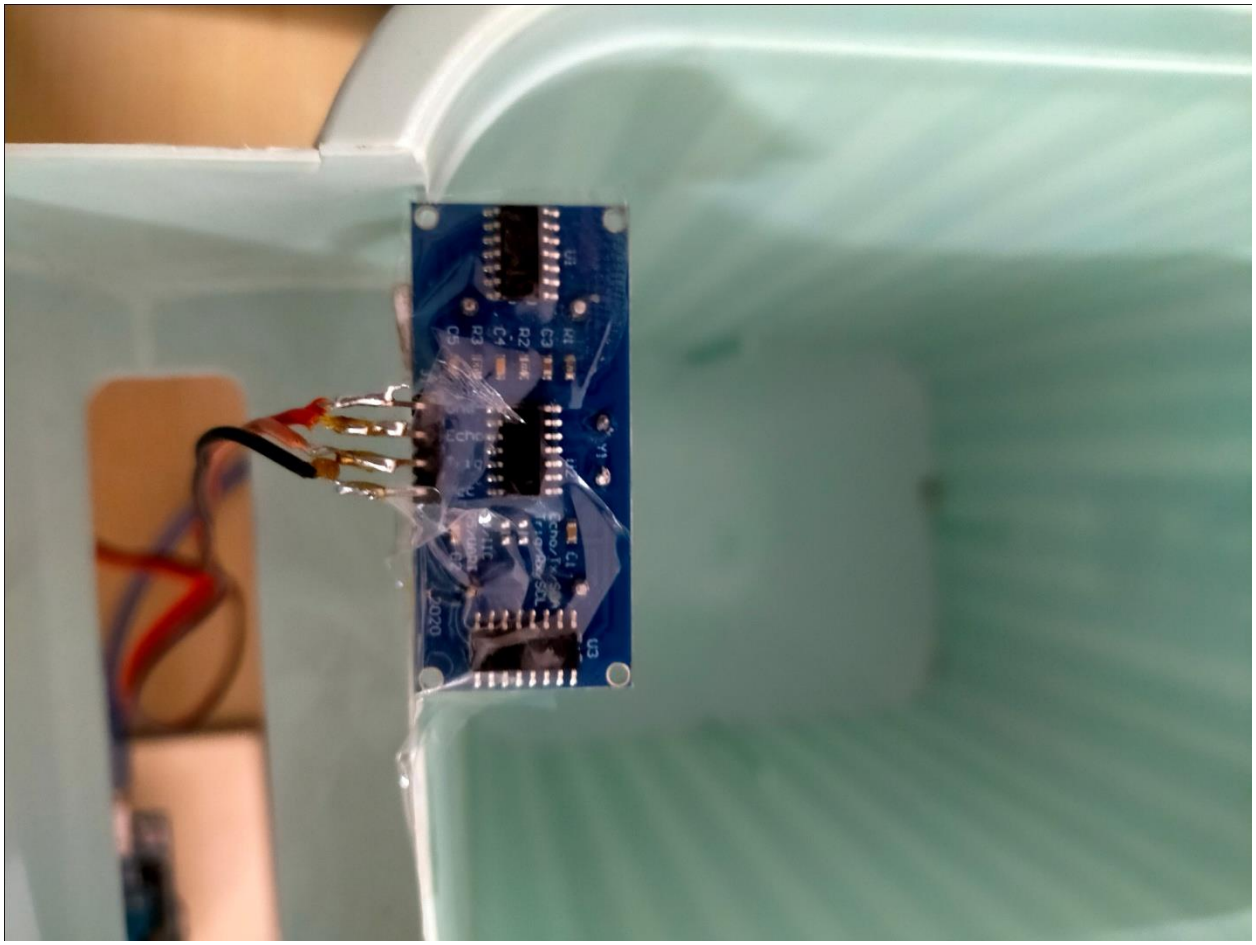


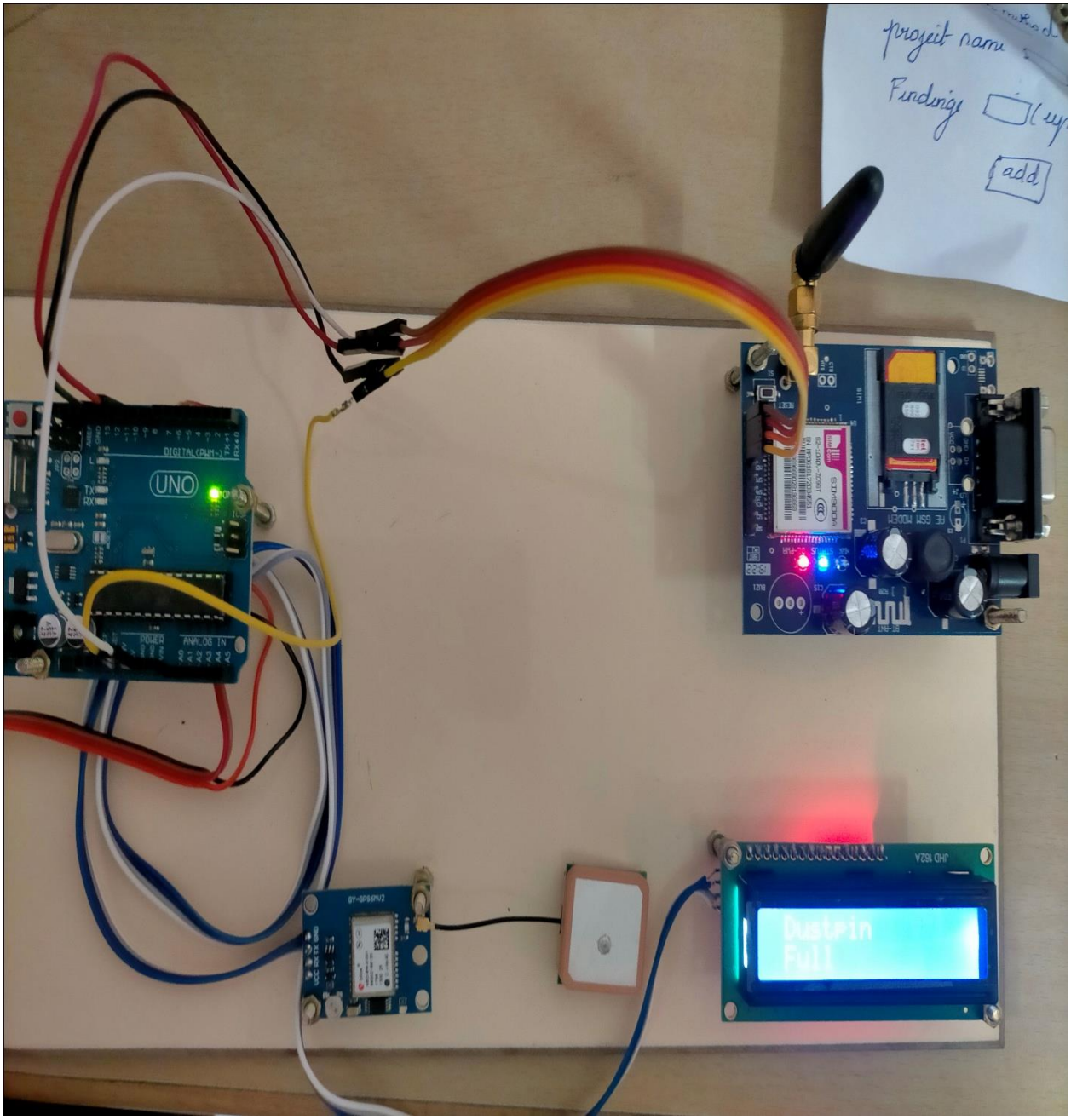

SIM900A
AE GSM MODEM
RESET
STATUS
DC-PWR
BT-ANT


UNO
VCC RX TX GND
DustPin
Low


832876
₹ 240.00
FSM ₹ 240.00
UNO


JHD 162A
DustPin
Half




projeit nam
Findings
add
UNO
DIGITAL (PWM~)
POWER
ANALOG IN
SIMCOM
GY-GPS6MV2
Dustpin
Full