CHAPTER-1

INTRODUCTION

We are living in an age where tasks and systems are fusing together with the power of IOT to have a more efficient system of working and to execute jobs quickly! With all the power at our finger tips this is what we have come up with.

The Internet of Things (IoT) shall be able to incorporate transparently and seamlessly a large number of different systems, while providing data for millions of people to use and capitalize. Building a general architecture for the IoT is hence a very complex task, mainly because of the extremely large variety of devices, link layer technologies, and services that may be involved in such a system.

One of the main concerns with our environment has been solid waste management which impacts the health and environment of our society. The detection, monitoring and management of wastes is one of the primary problems of the present era. The traditional way of manually monitoring the wastes in waste bins is a cumbersome process and utilizes more human effort, time and cost which can easily be avoided with our present technologies.

This is our solution, a method in which waste management is automated. This is our IoT Garbage Monitoring system, an innovative way that will help to keep the cities clean and healthy.

1.1 PURPOSE OF STUDY

The main objective of our project involves applying IoT technology (electronics and applications) to the current urban waste management scenario and enables a two way communication between the infrastructures deployed in the city and the operators/administrators. A centralized system for real-time monitoring is our goal to achieve. In this way both the municipal and citizens benefit from an optimized system which results in major cost savings and less urban pollution.

1. Waste Level detection inside the garbage bins. Transmission of the information wirelessly to concerned officials.

2. System can be accessed anytime and from anywhere.

3. Real-time data transmission and access.

4. Avoids the overflows of garbage bins.

5. This project can only be used by municipal authorities or other private firms to tackle the current problem of urban waste collection.

6. This system has no individual use, but can be used by a city, state or a country.

7. Using this system, waste collection would become efficient and also reduction in transportation costs can be witnessed.

1.2 PROBLEM STATEMENT

Nowadays, there are tons of flats and apartments which have been built in the rapid urbanization area. This is due to high housing demands which have been drastically risen as a result of migration from villages to cities to find work.

In order to accommodate the growing population in the urban area, the government has also constructed more apartment complexes. There are several issues faced by the residents of the flats. One of them is disposal of solid waste. Unlike private houses, the residents of all the apartments use a common dustbin, which tends to fill up very quickly.

This overflowing of garbage is a sanitary issue which might cause diseases like cholera and dengue. Moreover it is a waste of fuel to travel around a complex or an area to find that some of the garbage are filled and some are not. Also, on rare days, problems might arise that there is so much garbage that the truck doesn't have enough capacity.

The idea struck us when we observed that the garbage truck use to go around the town to collect solid waste twice a day.

Although this system was thorough it was very inefficient. For example let's say street A is a busy street and we see that the garbage fills up really fast whereas maybe street B even after two days the bin isn't even half full. This example is something that actually happens thus it lead us to the "Eureka" moment!.

1.3 MOTIVATION

Bangalore city's current waste collection logistics is carried out by emptying containers according to predefined schedules and routes which are repeated at a set frequency. Such a System has major disadvantages:

A. Time consuming:
B. High costs
C. Greater traffic and congestion.
D. Unnecessary fuel consumption.
E. Increased noise and air pollution as a result of more trucks on the road .

All the above disadvantages are a result of lack of real time information resulting in unsuccessful collection of waste. The Bangalore Municipal itself finds this as a big problem and a big hurdle in between Bangalore Smart City initiative.

There is an urgent need to optimize the management of this service to reduce infrastructure, operating and maintenance costs, as well as reduce contamination directly associated with waste collection.

1.4 METHODOLOGY

This project consists of an Arduino board, two ultrasonic sensors, 6 LEDs consisting of 2 red, 2 yellow, 2 green LEDs, a GSM module.

The ultrasonic sensor in the lid of the garbage bin to measure the distance.

Then program an Arduino board to act according to the height detected by the ultrasonic sensor.

The 3 LEDs consisting of red, yellow and green color at our garbage bins. It will show the amount of garbage in the bin from outside. If the amount of garbage is low, the green LED will glow, if the garbage bin has a moderate amount of garbage that is around half filled, the yellow LED will glow and if the garbage bin is full, the red LED will glow showing that the garbage bin is completely filled and we need to empty the garbage bin. And if the red LED glows that is the garbage bin is full, our system sends a message to the garbage collector that the following garbage bin is completely filled and he needs to empty it soon.

The 2 Ultrasonic sensors at two different garbage bins to show the working of the product. Each of them will have separate LEDs to show the amount of garbage from outside.

CHAPTER-2

SYSTEM REQUIREMENTS AND LANGUAGE USED

2.1 HARDWARE AND SOFTWARE REQUIREMENTS

- **Hardware System Configuration:**

Processor - Intel Core i7

Speed - 2.70 GHz

RAM - 256 MB (min)

Hard Disk - 8 GB

- **Software System Configuration:**

Operating System - Windows 7

Programming Language - C++ language

Compiler - Arduino IDE

2.2 ABOUT THE LANGUAGE

C belongs to the structured, procedural paradigms of languages. It is proven, flexible and powerful and may be used for a variety of different applications. Although high level, C and assembly language share many of the same attributes.

Some of C's most important features include:

- Fixed number of keywords, including a set of control primitives, such as if, for, while, switch and do while.
- Multiple logical and mathematical operators, including bit manipulators
- Multiple assignments may be applied in a single statement.
- Function return values are not always required and may be ignored if unneeded.
- Typing is static. All data has type but may be implicitly converted.
- Basic form of modularity, as files may be separately compiled and linked.
- Control of function and object visibility to other files via extern and static attributes.

Data types in c refer to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.

The types in C can be classified as follows −

- **char:** The most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers.
- **int:** As the name suggests, an int variable is used to store an integer.
- **float:** It is used to store decimal numbers (numbers with floating point value) with single precision.
- **double:** It is used to store decimal numbers (numbers with floating point value) with double precision.

Different data types also have different ranges upto which they can store numbers. These ranges may vary from compiler to compiler. Below is list of ranges along with the memory requirement and format specifiers on 32 bit gcc compiler.

some of the functions used to program Arduino UNO are follows

- **Serial.available()** – checks for any data coming through serial port of arduino. The function returns the number of bytes available to read from serial buffer. If there is no data available, it returns a -1 (value less than zero).

- **Serial.read()** – Reads all the data available on serial buffer (or incoming serial data if put otherwise). It returns the first byte of incoming serial data.

- **mySerial.available()** – checks for any data coming from GSM module through the SoftwareSerial pins 9 and 10. It returns the number of bytes available to read from software serial port. It returns a -1 if no data is available to read.

- **mySerial.read()** – Reads the incoming data through software serial port.

- **Serial.write()** – Prints data to serial monitor of Arduino. So the function Serial.write(mySerial.read()) – prints the data collected from software serial port to serial monitor of Arduino.

- **SendMessage()** – is the function we created in our Arduino sketch to send an SMS. To send an SMS, we should set our GSM module to Text mode first. This is achieved by sending an AT Command "AT+CMGF=1" We send this command by writing this to SoftwareSerial port. To achieve this we use the mySerial.println() function.
  mySerial.println writes data to software serial port (the Tx pin of our Software Serial – that is pin 10) and this will be captured by GSM module (through its Rx pin). After setting the GSM module to Text mode, we should the the mobile number to which we shall send the SMS.
  This is achieved with AT command "AT+CMGS=\"+91xxxxxxxxxx\"\r" – where you may replace all x with the mobile number..
  In next step, we should send the actual content of SMS. The end of SMS content is identified with CTRL+Z symbol. The ASCII value of this CTRL+Z is 26. So we send a char(26) to GSM module using the line mySerial.println((char)26); Each and every AT command may be followed by 1 second delay. We must give some time for GSM

module to respond properly. Once these commands are send to GSM module, you shall receive an SMS in the set mobile number.

- **RecieveMessage()** – is the function to receive an SMS (a live SMS). The AT command to receive a live SMS is "AT+CNMI=2, 2, 0, 0, 0" – we just need to send this command to GSM module and apply a 1 second delay. Once you send this command, try sending an SMS to the SIM card number put inside GSM module. You will see the SMS you had sent displayed on your Arduino serial monitor.

2.3 HARDWARE COMPONENETS USED

2.3.1 Ultrasonic sensors

Ultrasonic sensors "are based on the measurement of the properties of acoustic waves with frequencies above the human audible range," often at roughly 40 kHz. They typically operate by generating a high-frequency pulse of sound, and then receiving and evaluating the properties of the echo pulse.

The main advantage of ultrasonic sensors is that measurements may be made without touching or otherwise impeding the target. In addition, depending on the distance measured, measurement is relatively quick (it takes roughly 6ms for sound to travel 1m). However, many factors such as temperature, angle, and material may affect measurements.

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.

Fig 2.3.1.1 ULTRASONIC SENSOR

2.3.2 GSM module

GSM/GPRS module is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile communication (GSM) is an architecture used for mobile communication in most of the countries. Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a SIM (Subscriber Identity Module) card just like mobile phones

to activate communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification.

A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.
2. Read, add, search phonebook entries of the SIM.
3. Make, Receive, or reject a voice call.

The MODEM needs AT commands, for interacting with processor or controller, which are communicated through serial communication.

These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM and GPRS cellular network.
A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.
A SIM card contains the following information:

- Subscriber telephone number (MSISDN)
- International subscriber number (IMSI, International Mobile Subscriber Identity)
- State of the SIM card
- Service code (operator)
- Authentication key
- PIN (*Personal Identification Code*)

- PUK (*Personal Unlock Code*)

Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, the following operations can be performed:

- Reading, writing and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

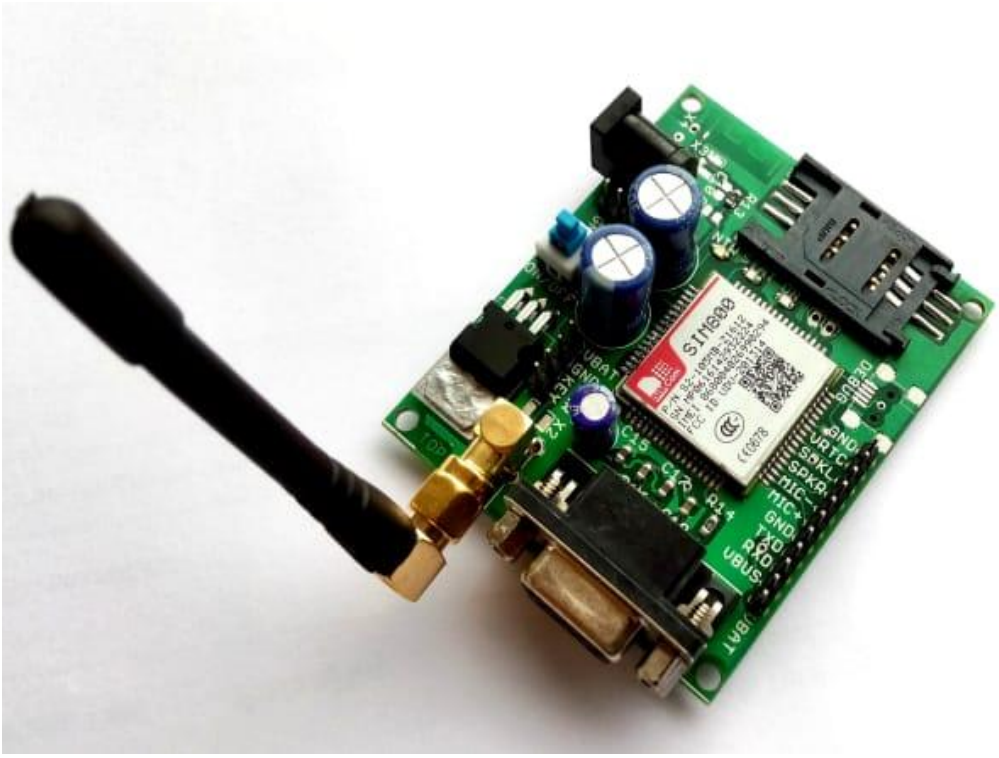


Fig2.3.2.1 GSM module SIM 800A

2.3.3 Arduino UNO

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 hardware and software was designed for artists, designers, hobbyists, hackers, newbie's, and anyone interested in creating interactive objects or environments.

Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a huge variety of Arduino-based projects.

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. Revision of the board has the following new features:

- Pin out: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin that is reserved for future purposes.

- Stronger RESET circuit.
- AT mega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform.

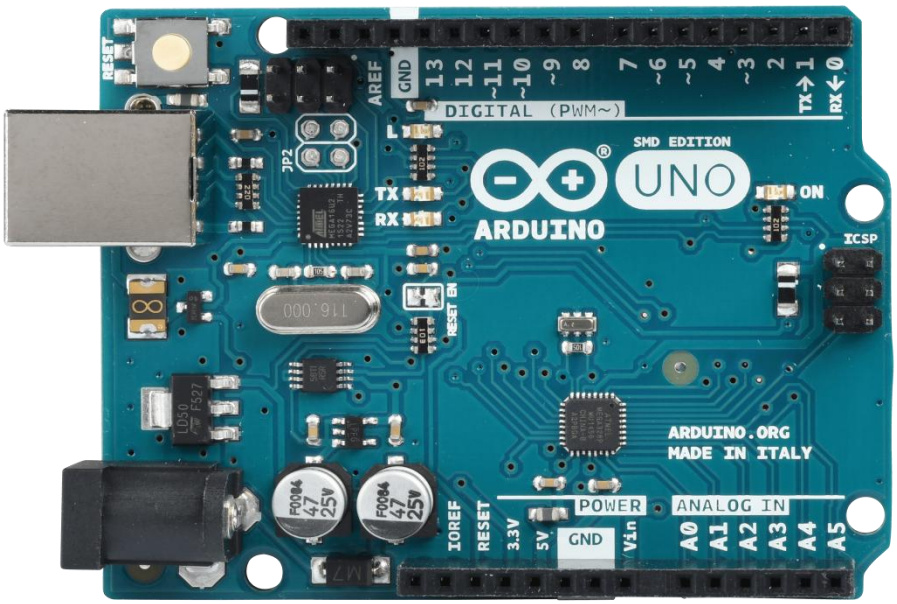


Fig 2.3.3.1 Arduino UNO

2.3.4 LED

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. LEDs are typically small (less than 1 mm2) and integrated optical components may be used to shape the radiation pattern.

Fig 2.3.4.1 LED diodes

2.3.5 JUMPER WIRES

A **jump wire** (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

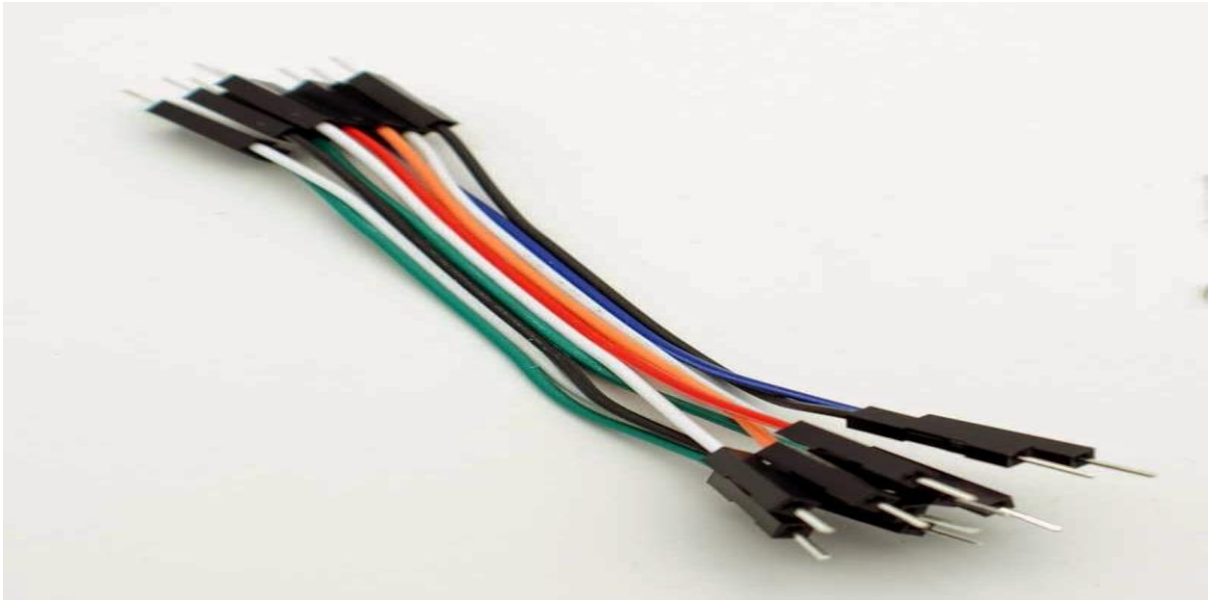


Fig 2.3.5.1 Male jumper wires

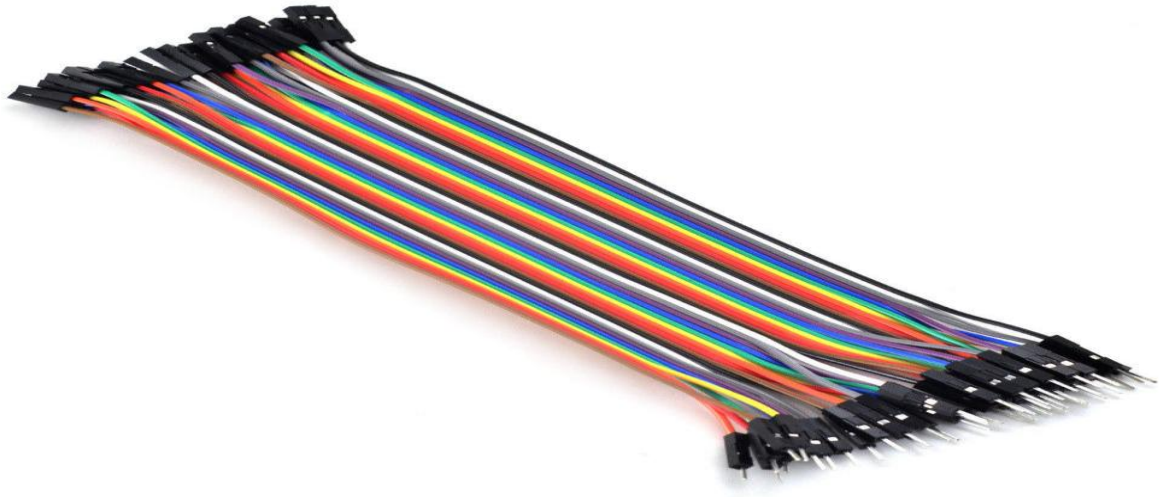


Fig 2.3.5.2 Male to female jumper wires.

2.3.6 Bread board

A **breadboard** is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread.

In the 1970s the **solder less breadboard** (a.k.a. **plug board**, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solder less breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solder less breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A strip board(Vero board) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

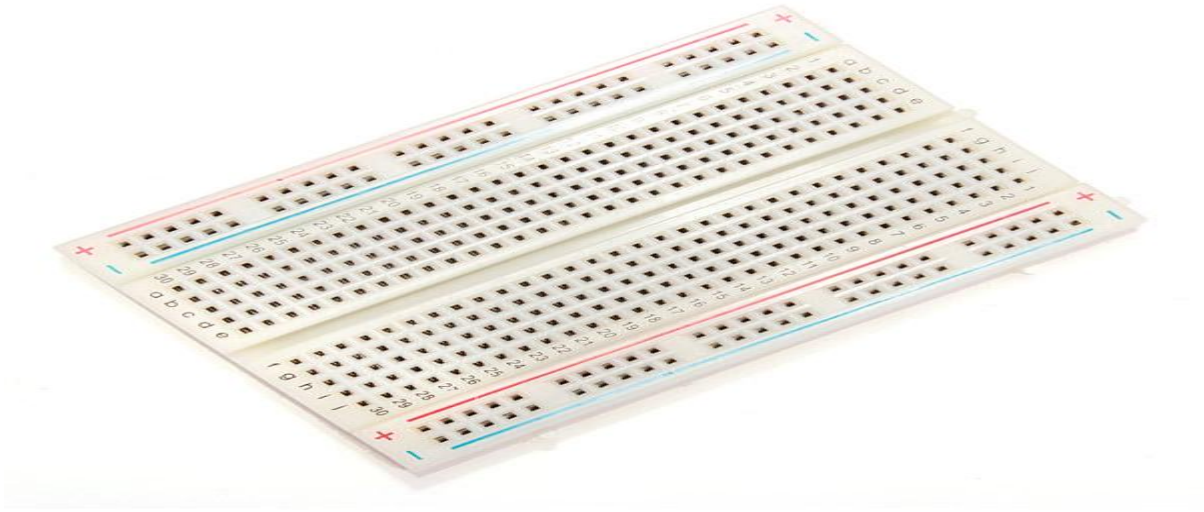
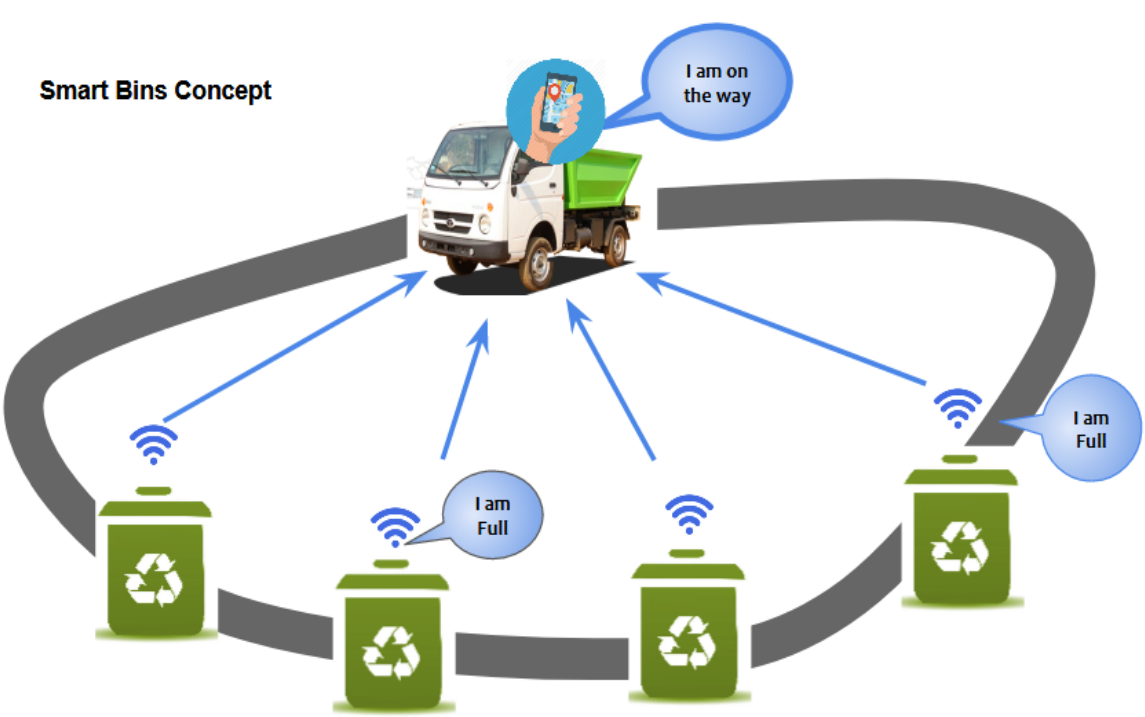


Fig 2.3.6.1 Bread Board

CHAPTER-3

SYSTEM DESIGN

3.1 ARCHITECTURE



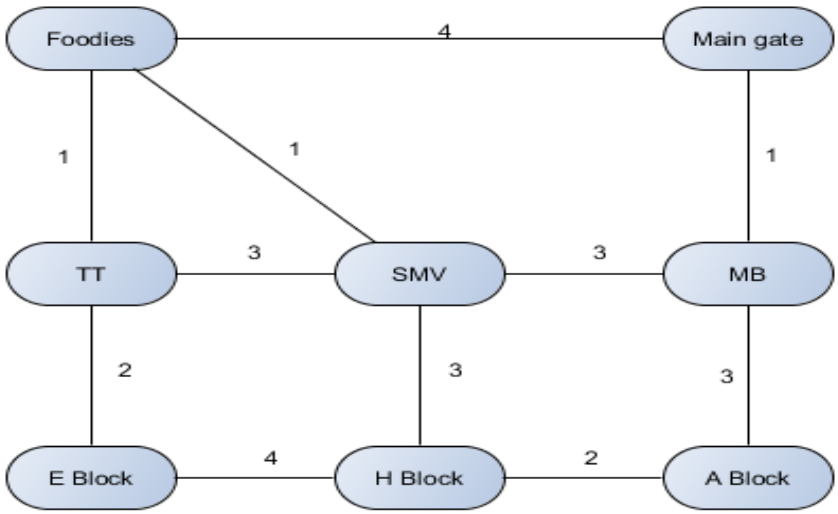
In this project, not only like all other garbage monitoring systems, it will warn the garbage collector when the bin is full but also it will give the shortest path to the location where the garbage bin is located. So that the garbage collector can easily find where the garbage bin is located. He will be given a shortest route consisting of different points in the locality to the destination that is where the garbage bin is full now.

For calculating the shortest path, we are using Dijkstra's algorithm. It is an algorithm to find the shortest path between the various vertices of the graph, it not only gives the total cost to reach that vertex but also gives the path which will be followed to get the minimum cost possible. First of all the distance between two vertices are set as infinite. Then the distance is calculated between the two nodes and it replaces the existing value of distance if it is shorter

than the existing distance. The same process is repeated for each and every node and hence the shortest path is calculated between the source and destination.

In our project first of all the weight between all the nodes are set as 99. Then we are changing the weight between the two nodes which are connected to each other as the weight of them. Then the source is set as the starting place of the garbage collector, here we have set it as "Main Gate". As no garbage bin is full in the start, the destination is set the same as the source. When in some garbage bin, the height of garbage exceeds a certain limit, the red LED glows and the destination is set as the location of the garbage bin which is full now. Then the Dijkstra's algorithm is run between the source and the destination and the shortest path is calculated. This shortest path is now sent to the garbage collector via SMS on his mobile. So now even if he is new in the city, he can rush to the place in no time as he has the shortest path.

This project doesn't needs any internet connection, so it is very beneficial as all the garbage collectors don't have a internet connection. They will get the path to the garbage bin as a text message on their phone directly. So the issue of internet connectivity is also solved by this project.



3.1.1 ABOUT DIJSKTRA's ALGORITHM

Dijkstra algorithm is also called single source shortest path algorithm. It is based on greedy technique. The algorithm maintains a list visited[ ] of vertices, whose shortest distance from the source is already known.

If visited[1], equals 1, then the shortest distance of vertex i is already known. Initially, visited[i] is marked as, for source vertex.

At each step, we mark visited[v] as 1. Vertex v is a vertex at shortest distance from the source vertex. At each step of the algorithm, shortest distance of each vertex is stored in an array distance[ ].

Dijkstra's Algorithm

1. Create cost matrix C[ ][ ] from adjacency matrix adj[ ][ ]. C[i][j] is the cost of going from vertex i to vertex j. If there is no edge between vertices i and j then C[i][j] is infinity.

2. Array visited[ ] is initialized to zero.

```
for(i=0;i<n;i++)
        visited[i]=0;
```

3. If the vertex 0 is the source vertex then visited[0] is marked as 1.

4. Create the distance matrix, by storing the cost of vertices from vertex no. 0 to n-1 from the source vertex 0.

```
for(i=1;i<n;i++)
        distance[i]=cost[0][i];
```

Initially, distance of source vertex is taken as 0. i.e. distance[0]=0;

5. for(i=1;i<n;i++)

– Choose a vertex w, such that distance[w] is minimum and visited[w] is 0. Mark visited[w] as 1.

– Recalculate the shortest distance of remaining vertices from the source.

– Only, the vertices not marked as 1 in array visited[ ] should be considered for recalculation of distance. i.e. for each vertex v

```
if(visited[v]==0)
        distance[v]=min(distance[v],
        distance[w]+cost[w][v])
```

Time Complexity

The program contains two nested loops each of which has a complexity of O(n). n is number of vertices. So the complexity of algorithm is $O(n^2)$.

Advantages:-

1) It is used in Google Maps

2) It is used in finding Shortest Path.

3) It is used in geographical Maps

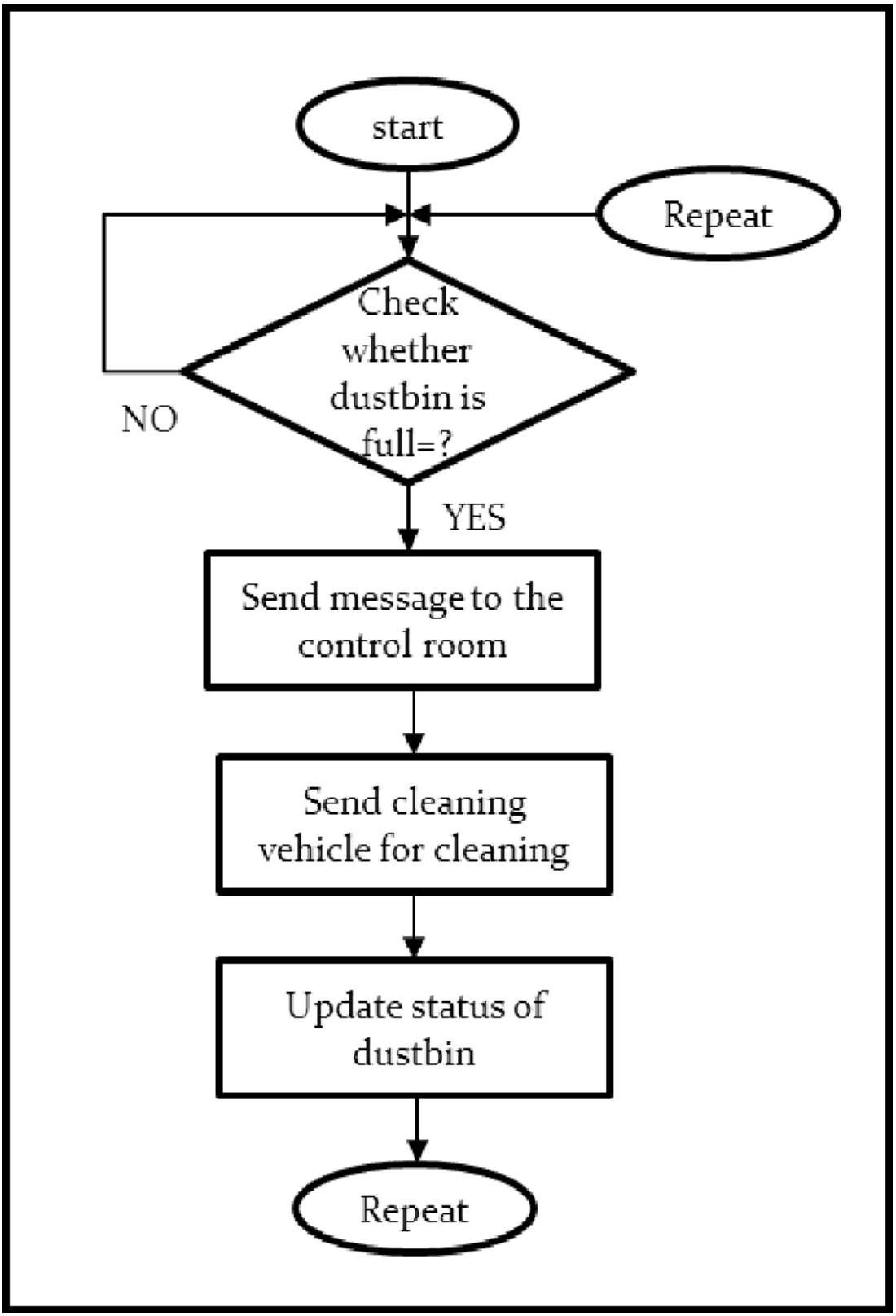
4) To find locations of Map which refers to vertices of graph.

5) Distance between the location refers to edges.

6) It is used in IP routing to find Open shortest Path First.

7) It is used in the telephone network.

3.2 FLOWCHART



3.3 ALGORITHM

step 1: start

step 2:check the status of the bin

if(binstatus==full):

print send the message to corparate

else:

continue

step 3:bin status

if(distance<=30):

print empty

elseif(distance>=30 && distance<=50):

print medium

else

print full

step 4: if(message==sent)

print shortest path to collect the bins

else

print error

step 5:stop

3.4 MODULES

- **Serial.available()** – checks for any data coming through serial port of arduino. The function returns the number of bytes available to read from serial buffer. If there is no data available, it returns a -1 (value less than zero).

- **Serial.read()** – Reads all the data available on serial buffer (or incoming serial data if put otherwise). It returns the first byte of incoming serial data.

- **mySerial.available()** – checks for any data coming from GSM module through the SoftwareSerial pins 9 and 10. It returns the number of bytes available to read from software serial port. It returns a -1 if no data is available to read.

- **mySerial.read()** – Reads the incoming data through software serial port.

- **Serial.write()** – Prints data to serial monitor of Arduino. So the function Serial.write(mySerial.read()) – prints the data collected from software serial port to serial monitor of Arduino.

- **SendMessage()** – is the function we created in our Arduino sketch to send an SMS. To send an SMS, we should set our GSM module to Text mode first. This is achieved by sending an AT Command "AT+CMGF=1" We send this command by writing this to SoftwareSerial port. To achieve this we use the mySerial.println() function. mySerial.println writes data to software serial port (the Tx pin of our Software Serial – that is pin 10) and this will be captured by GSM module (through its Rx pin). After setting the GSM module to Text mode, we should the the mobile number to which we shall send the SMS.

This is achieved with AT command “AT+CMGS=\”+91xxxxxxxxxx\”\r” – where you may replace all x with the mobile number..

In next step, we should send the actual content of SMS. The end of SMS content is identified with CTRL+Z symbol. The ASCII value of this CTRL+Z is 26. So we send a char(26) to GSM module using the line mySerial.println((char)26); Each and every AT command may be followed by 1 second delay. We must give some time for GSM module to respond properly. Once these commands are send to GSM module, you shall receive an SMS in the set mobile number.

- **RecieveMessage()** – is the function to receive an SMS (a live SMS). The AT command to receive a live SMS is “AT+CNMI=2, 2, 0, 0, 0” – we just need to send this command to GSM module and apply a 1 second delay. Once you send this command, try sending an SMS to the SIM card number put inside GSM module. You will see the SMS you had sent displayed on your Arduino serial monitor.

3.5 CODE AND IMPLEMENTATION

```
#include <SoftwareSerial.h>
#include <string.h>

SoftwareSerial mySerial(9, 10);
const int trigPin1 = 2;
const int echoPin1 = 4;5
const int trigPin2 = 3;
const int echoPin2 = 5;
int ledr1=13,ledy1=12,ledg1=11,ledr2=7,ledy2=6,ledg2=8;
int IN=99,N=9;

void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);
    pinMode(ledr1,OUTPUT);
    pinMode(ledy1,OUTPUT);
    pinMode(ledg1,OUTPUT);
    pinMode(ledr2,OUTPUT);
    pinMode(ledy2,OUTPUT);
    pinMode(ledg2,OUTPUT);
    delay(100);
}

void dijsktra(int cost[9][9],int source,int target)
{
    int dist[N],prev[N],selected[N]={0},i,m,min1,start,d,j,a,target1,x=0;
    char path[N],path1[50],path2[50];
    target1=target;
    for(i=1;i< N;i++)
    {
```

```
    dist[i] = IN;
    prev[i] = -1;
  }
 start = source;
selected[start]=1;
dist[start] = 0;
while(selected[target] ==0)
{
   min1 = IN;
   m = 0;
   for(i=1;i< N;i++)
   {
      d = dist[start] +cost[start][i];
      if(d< dist[i]&&selected[i]==0)
      {
         dist[i] = d;
         prev[i] = start;
      }
      if(min1>dist[i] && selected[i]==0)
      {
         min1 = dist[i];
         m = i;
      }
   }
   start = m;
   selected[start] = 1;
}
start = target;
j = 0;
while(start != -1)
{
```

```
    path[j++] = start+64;
    start = prev[start];
  }

  path[j]='\0';
  strrev(path);
  printf("\n");
  strcpy(path1,"main-gate");
  for(i=1;i<strlen(path);i++)
  {
    /*if (path[i]=='A')
      Serial.print(" main gate");*/
    if(path[i]=='B')
      strcat(path1,"--->MB");
    else if(path[i]=='C')
      strcat(path1,"--->SMV");
    else if(path[i]=='D')
      strcat(path1,"--->TT");
    else if(path[i]=='E')
      strcat(path1,"--->FOODIES");
    else if(path[i]=='F')
      strcat(path1,"--->E-BLOCK");
    else if(path[i]=='G')
      strcat(path1,"--->H-BLOCK");
    else if(path[i]=='H')
      strcat(path1,"--->A-BLOCK");
  }
  Serial.print(path1);
  if(target1>1 && x<2)
  {
   SendMessage(path1);
   x=x+1;
```

```
   }
}

void loop()
{
 int target=1,x1=0;
 long duration1, cm1;
 pinMode(trigPin1, OUTPUT);
 digitalWrite(trigPin1, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin1, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin1, LOW);
 pinMode(echoPin1, INPUT);

 duration1 = pulseIn(echoPin1, HIGH);
 cm1 = microsecondsToCentimeters(duration1);
 long duration2, cm2;
 pinMode(trigPin2, OUTPUT);
 digitalWrite(trigPin2, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin2, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin2, LOW);
 pinMode(echoPin2, INPUT);
 duration2 = pulseIn(echoPin2, HIGH);
 cm2 = microsecondsToCentimeters(duration2);
 delay(100);
 Serial.println("");

 if(cm1>50)
 {
```

```
  digitalWrite(ledr1,LOW);
  digitalWrite(ledy1,LOW);
  digitalWrite(ledg1,HIGH);
 }
 else if(cm1>10 && cm1<50)
 {
  digitalWrite(ledr1,LOW);
  digitalWrite(ledg1,LOW);
  digitalWrite(ledy1,HIGH);
 }
 else
 {
  digitalWrite(ledy1,LOW);
  digitalWrite(ledg1,LOW);
  digitalWrite(ledr1,HIGH);
  target=8;
   }
  if(cm2>50)
 {
  digitalWrite(ledr2,LOW);
  digitalWrite(ledy2,LOW);
  digitalWrite(ledg2,HIGH);
 }
 else if(cm2>10 && cm2<50)
 {
  digitalWrite(ledr2,LOW);
  digitalWrite(ledg2,LOW);
  digitalWrite(ledy2,HIGH);
 }
 else
 {
  digitalWrite(ledy2,LOW);
```

```
   digitalWrite(ledg2,LOW);
   digitalWrite(ledr2,HIGH);
   target=6;
  }

 int cost[9][9],i,j,w,ch,co;
   int source=1,x,y;
   for(i=1;i< N;i++)
       {for(j=1;j< N;j++)
           {cost[i][j] = IN;}}
   cost[1][2]=cost[2][1]=1;
   cost[1][5]=cost[5][1]=4;
   cost[2][3]=cost[3][2]=3;
   cost[2][8]=cost[8][2]=3;
   cost[3][4]=cost[4][3]=3;
   cost[3][5]=cost[5][3]=1;
   cost[3][7]=cost[7][3]=3;
   cost[4][5]=cost[5][4]=1;
   cost[4][6]=cost[6][4]=2;
   cost[6][7]=cost[7][6]=4;
   cost[7][8]=cost[8][7]=2;
   dijsktra(cost,source,target);

}
long microsecondsToCentimeters(long microseconds)
{
return microseconds/29/2;
}

 void SendMessage(char path[50])
{  mySerial.println("AT+CMGF=1\r");
  delay(1000);
```

```
  mySerial.println("AT+CMGS=\"+919939581716\"\r");
  delay(1000);
  mySerial.println(path);
  delay(100);
   mySerial.println((char)26);
  delay(1000);
}
```

3.5.1 IMPLEMENTATION

The figure 5 shows the circuit diagram of IR transmitter using the IC 555. Since the PCM carrier frequency of TSOP1738 is 38 KHz, so to transmit the accurate beam, IC 555 is used in „Astable Multivibrator" mode. This is achieved by using two resistors (R1 and R2) and the capacitor (C). The value of resisters and capacitors are calculated using the following equations.

Thigh = 0.693(R1+R2)C

(1) Tlow= 0.693R2 C

(2) T= Thigh + Tlow

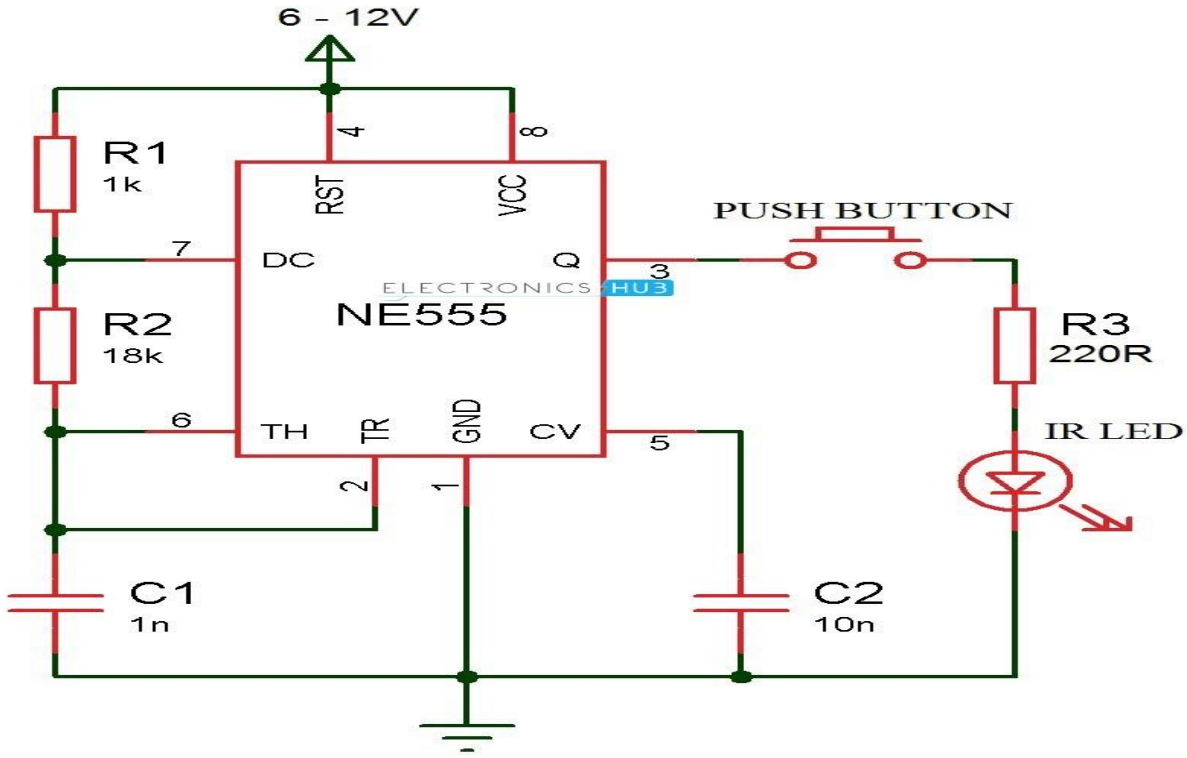
(3) F=1/T

(4) Where Thigh and Tlow are the time period for which the output of IC 555 is HIGH and LOW respectively; T is total time period of the output of IC 555 and „F" is the output frequency. Output of the circuit shown in figure 5 is IR beam and it is taken from IR LED. Furthermore this beam is used to detect the garbage level.

The IR sensor arrangement is act as level detector .The output of level detector is given to the microcontroller .

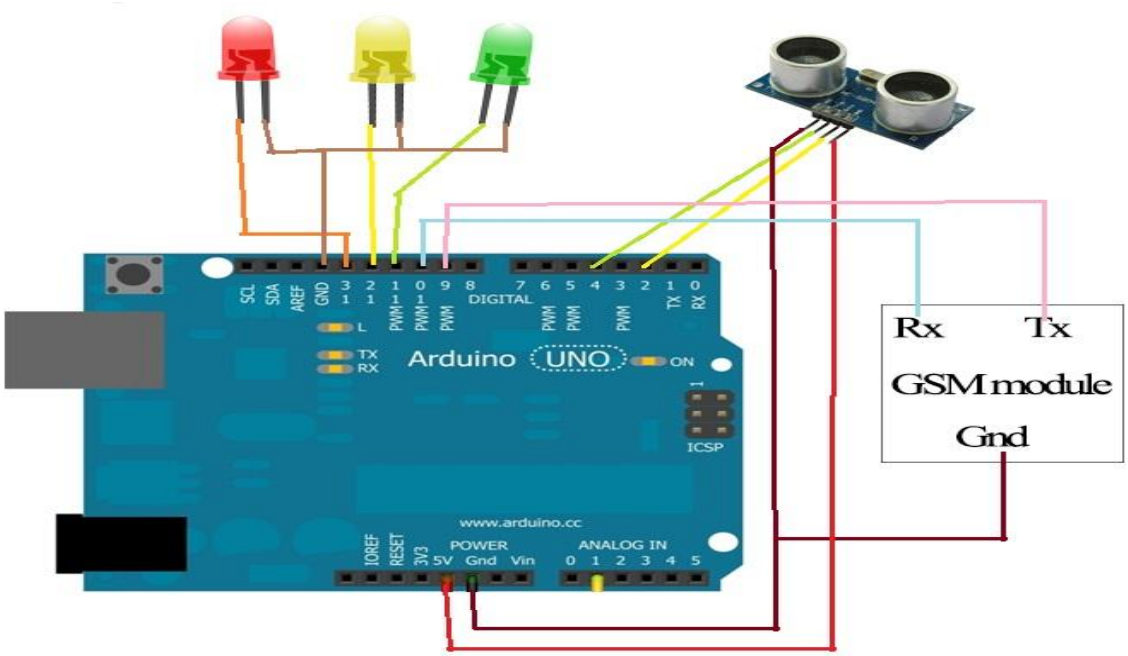
The AT commands are used to facilitate the messaging service through the GSM Module. This program is burned in the microcontroller with the help of Arduino software (IDE)

version1.6.1. These messages consist of information of garbage levels of respective dustbins. Depending on the information sent to control room, the authority informs the concern person of the respective area about garbage level. Then the concerned person makes sure that the garbage of that particular area is collected by sending the cleaning vehicles.



3.5.1.1 IR sensor transmitter

3.5.2 CIRCUIT DIAGRAM/ ARDUINO LAYOUT



CONNECTIONS FROM ULTRASONIC SENSORS TO ARDINUO UNO

- Echo pin of sensor1,sensor2 is connected to 4,5th pins of Arduino board .
- Trigger pin of sensor1,sensor2 is connected to 2,3rd pins of Arduino board.
- ground pins of both the sensors are connected to ground of Arduino Uno board.
- Vcc pins of sensors are connected to 5v pin of Arduino Uno board.

CONNECTIONS FROM GSM MODULE SIM800A TO ARDINUO UNO

- Tx pin of GSM module is connected to 9th pin of Arduino Uno board.
- Rx pin of GSM module is connected to 10th pin of Arduino Uno board.
- Vcc pin of GSM module is connected to 5v of Arduino board.
- ground pin is connected to ground pin of Arduino board.

CHAPTER-4

RESULT AND DISCUSSION

4.1 RESULT AND OUTPUT(Snapshots)

This implementation of Smart Garbage Collection System using IoT, assures the cleaning of dustbins soon when the garbage level reaches its maximum. If the dustbin is not cleaned in specific time, then the record is sent to the higher authority who can take appropriate action against the concerned official.

This system also helps to monitor the fake reports and hence can reduce the corruption in the overall management system. This reduces the total number of trips of garbage collection vehicle and hence reduces the overall expenditure associated with the garbage collection. It is ultimately helps to keep cleanliness in the society. This is quite a significant project in its originality and concept.

We are using Internet of Things theory which gives this project its charisma and uniqueness about the concept. The project aims at cleanliness of the areas where trash bins are located and the very basic management that it contains with it.

It aims at advanced management of the whole garbage collection system. We use ultrasonic sensors (details mentioned above) and its other hardware microcontrollers and processors such as Arduino for analyzing the garbage levels and sending information about it to administrators and then garbage trucks are being deployed by them.

1. Waste Level detection inside the garbage bins. Transmission of the information wirelessly to concerned officials

2. System can be accessed anytime and from anywhere.

3. Real-time data transmission and access.

4.Avoids the overflows of garbage bins.

5.This project can only be used by municipal authorities or other private firms to tackle the current problem of urban waste collection.

6. This system has no individual use, but can be used by a city, state or a country.

7. Using this system, waste collection would become efficient and also reduction in transportation costs can be witnessed.

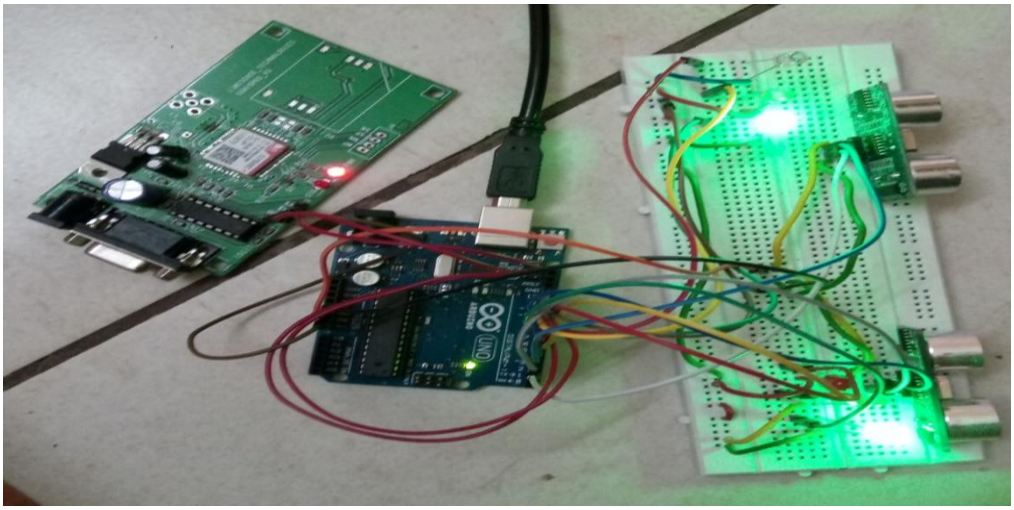


Fig 4.1.1 both the garbage bins are empty. Both showing green light.

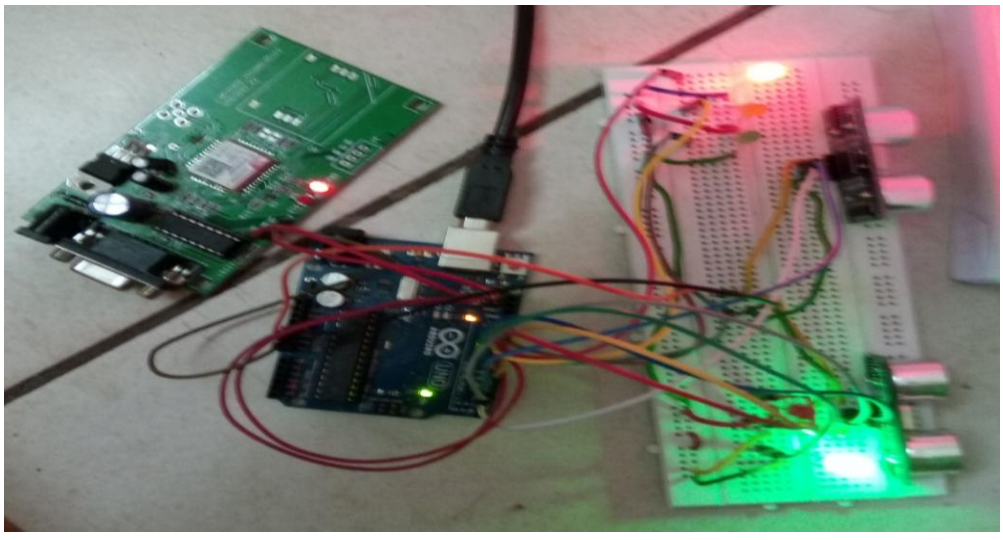


Fig4.1.2 When one of the garbage bins gets full:

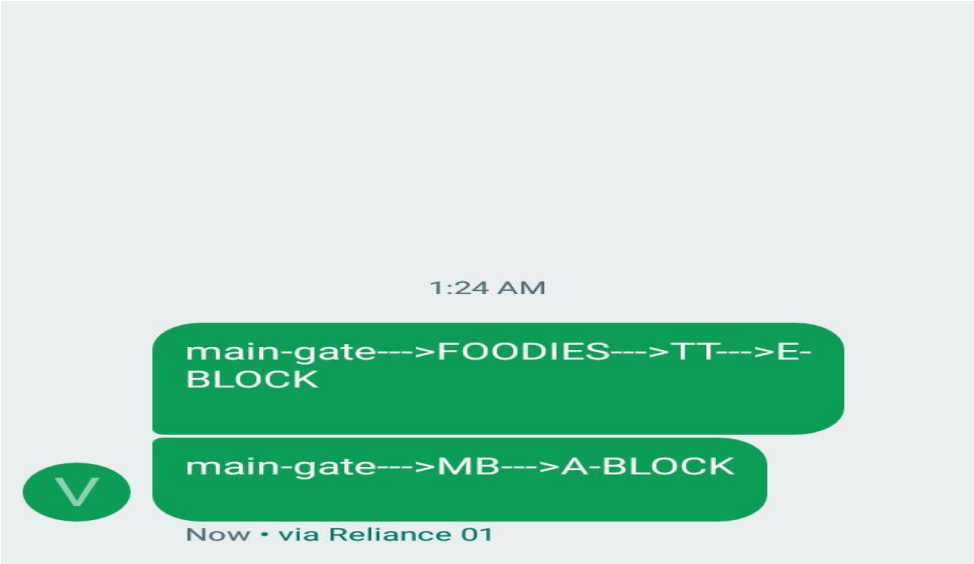


Fig 4.1.3 Messages reaching to respective person

CONCLUSION

By implementing this proposed system the cost reduction, resource optimization, effective usage of smart dustbins can be done. This system indirectly reducing traffic in the city . In major cities the garbage collection vehicle visit the area's everyday twice or thrice depends on the population of the particular area and sometimes these dustbins may not be full. Our System will inform the status of each and every dust bin in real time so that the concerned authority can send the garbage collection vehicle only when the dustbin is full. The scope for the future work is this system can be implemented with time stamp in which real-time clock shown to the concern person at what time dust bin is full and at what time the waste is collected from the smart dustbins.

With the help of this new proposed system we can able to avoid over flowing of garbage. This project work is the implementation of smart garbage management system using sensors, ARM7, GSM module, GPS and IOT. This system assures the cleaning of dustbins soon when the garbage level reaches its maximum. This reduces the total number of trips of garbage collection vehicle and hence reduces the overall expenditure associated with the garbage collection. This system also helps to monitor the fake reports and hence can reduce the corruption in the overall management system. This method finally helps in keeping the environment clean. Thus, the garbage collection is made more efficient.

REFRENCES

1. www.wenglor.com
2. https://www.arduino.cc/
3. http://ijireeice.com
4. Ikuo Ihara; Nagaoka University of Technology; Ultrasonic Sensing: Fundamentals and Its Applications to Nondestructive Evaluation.
5. Arduino, "Available at http://www.arduino.cc," 2010.
6. M. Batty, "Smart Cities, Big Data," Environment and Planning B: Planning and Design 2012, vol. 39, pp. 191–93.
7. Xu Li, Student Member, IEEE, Performance Evaluation of Vehicle-Based Mobile Sensor Networks for Traffic Monitoring.
8. Yusuf Abdullahi Badamasi, The Working Principle Of An Arduino, Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on 29 Sept.-1 Oct. 2014.
9. Pedro Reis , Rui Pitarma, Celistino Goncalves, Intelligent System for Valorizing Solid Urban Waste, Filipe Caetano Faculty of Engineering UBI University of Beira Interior Covilha, Portugal, 2015.
10. Adnan Aijaz, Member, IEEE; Cognitive Machine-to Machine Communications for Internet-of-Things: A Protocol Stack Perspective.
11. https://in.search.yahoo.com/yhs/search;_ylt=AwrwXxQvE8NcrksAehPnHgx.;_ylc=X1MDMjExNDcyMzU1OQRfcgMyBGZyA3locy1wdHktcHR5X2V4dGVuc2lvbgRncHJpZANFOEpmQ2ZyRVNPaWJrenlNdE5xUllBBG5fcnNsdAMwBG5fc3VnZwMwBG9yaWdpbgNpbi5zZWFyY2gueWFob28uY29tBHBvcwMwBHBxc3RyAwRwcXN0cmwDMARxc3RybAMzMwRxdWVyeQNBRFZBTlRBR0VTJTIwZGlqa3N0cmElMjBhbGdvcml0aG0EdF9zdG1wAzE1NTYyODg1MjA-?p=ADVANTAGES+dijkstra+algorithm&fr2=sb-top&hspart=pty&hsimp=yhs-pty_extension&param1=20190126&param2=91a447e2-c281-4b48-ab53-c59a7dab4084&param3=tv_~IN~appfocus1&param4=d-lp0-dsf_tv--bb8~Chrome~about+dijkstra+algorithm~1E278C573C4F8C1856CE7C16A9051479&vm=r .

12. https://r.search.yahoo.com/_ylt=awrxhwalgmncr20aexnnhgx.;_ylu=x3odmtbyympmzja4bgnvbg8dc2czbhbvcwmzbhz0awqdbhnlywnzcg/rv=2/re=1556318502/ro=10/ru=http%3a%2f%2fwww.engpaper.com%2fiot2016.htm/rk=2/rs=06fqeudun1gxbbiokynwe9qd828-

APPENDIX

NUMBER OF HARDWARE COMPONENTS USED

S.NO	HARDWARE COMPONENT	NO.OF USED
1	Arduino Uno	1
2	GSM module	1
3	Ultrasonic sensors	2
4	Female to female jumper wire	7
5	Female to male jumper wires	11
6	Male to male jumper wires	6
7	Bread board	1