SMART WASTE MANAGEMENT SYSTEM USING IoT

Submitted in partial fulfilment of the requirements
of the degree of

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

BY

KHAN SHAUZAB (14DET89)

ALIASGAR LIGHTWALA (14DET91)

NAIK NADEEM (14DET96)

KHAN SHAHNAWAZ (14DET87)

SUPERVISOR

PROF. SIRAJ PATHAN

UNIVERSITY OF MUMBAI

ANJUMAN I ISLAM KALSEKAR TECHNICAL CAMPUS

2016-2017

PROJECT REPORT APPROVAL FOR B.E.

This project report entitled "SMART WASTEMANAGEMENT SYSTEM USING IoT" is approved for the degree of ELECTRONICS AND TELECOMMUNICATION ENGINEERING.

EXAMINER

1)----------

2)----------

SUPERVISOR

1)----------

HOD

1)----------

DATE: 22nd ARIL 2017.

PLACE: PANVEL.

TABLE CONTENT

CONTENT **PAGE NOS**

ABSTRACT

In the present day scenario, many times we see that the garbage bins or Dust bin are placed at public places in the cities are overflowing due to increase in the waste every day. It creates unhygienic condition for the people and creates bad smell around the surroundings this leads in spreading some deadly diseases & human illness, to avoid such a situation we are planning to design “Smart Waste Management System using IoT”. In this proposed System there are multiple dustbins located throughout the city or the Campus,these dustbins are provided with low cost embedded device which helps in tracking the level of the garbage bins and an unique ID will be provided for every dustbin in the city so that it is easy to identify which garbage bin is full. When the level reaches the threshold limit, the device will transmit the level along with the unique ID provided. These details can be accessed by the concern authorities from their place with the help of Internet and an immediate action can be made to clean the dustbins.

CHAPTER 1

INTRODUCTION

Things that are connected to Internet and sometimes these devices can be controlled from the internet is commonly called as Internet of Things. The Internet of Things (IoT) is a concept in which surrounding objects are connected through wired and wireless networks without user intervention. In the field of IoT, the objects communicate and exchange information to provide advanced intelligent services for users. Owing to the recent advances in mobile devices equipped with various sensors and communication modules, together with communication network technologies such as Wi-Fi and LTE, the IoT has gained considerable academic interests. Owing to the characteristics and merits of IoT services, waste management has also become a significant issue in academia, industry, and government as major IoT application fields. An indiscriminate and illegal discharge of waste, an absence of waste disposal and management systems, and inefficient waste management policies have caused serious environmental problems and have incurred considerable costs for waste disposal. In our system, the Smart dust bins are connected to the internet to get the real time information of the smart dustbins. In the recent years, there was a rapid growth in population which leads to more waste disposal. So a proper waste management system is necessary to avoid spreading some deadly diseases. Managing the smart bins by monitoring the status of it and accordingly taking the decision. This waste is further picked up by the municipal corporations to finally dump it in dumping areas and landfills. But due to lack of resources, ineffective groundwork, some waste is not collected which poses serious health hazard to the surrounding environment. Proper cleaning intervals may provide a solution to this problem. But keeping a track of the status of the bin manually is a very difficult job. There are multiple dustbins are located throughout the city or the Campus In our system, the Smart dust bins are connected to the internet to get the real time information of the smart dustbins. These dustbins are interfaced with raspberry pi based system with ultrasonic sensors. Where the ultrasonic sensor detects the level of the dust in dustbin and sends the signals to raspberry pi the same signal are encoded and send to the application and it is received.

The data has been received, analyzed and processed in the database, which displays the status of the Garbage in the dustbin on the application of authoritized person mobile. The concerned authority get alert about dustbin is full and informs person whoever is responsible for collecting garbage from the particular areas. The garbage trucks collect the garbage from the completely full dustbin and dispose it.

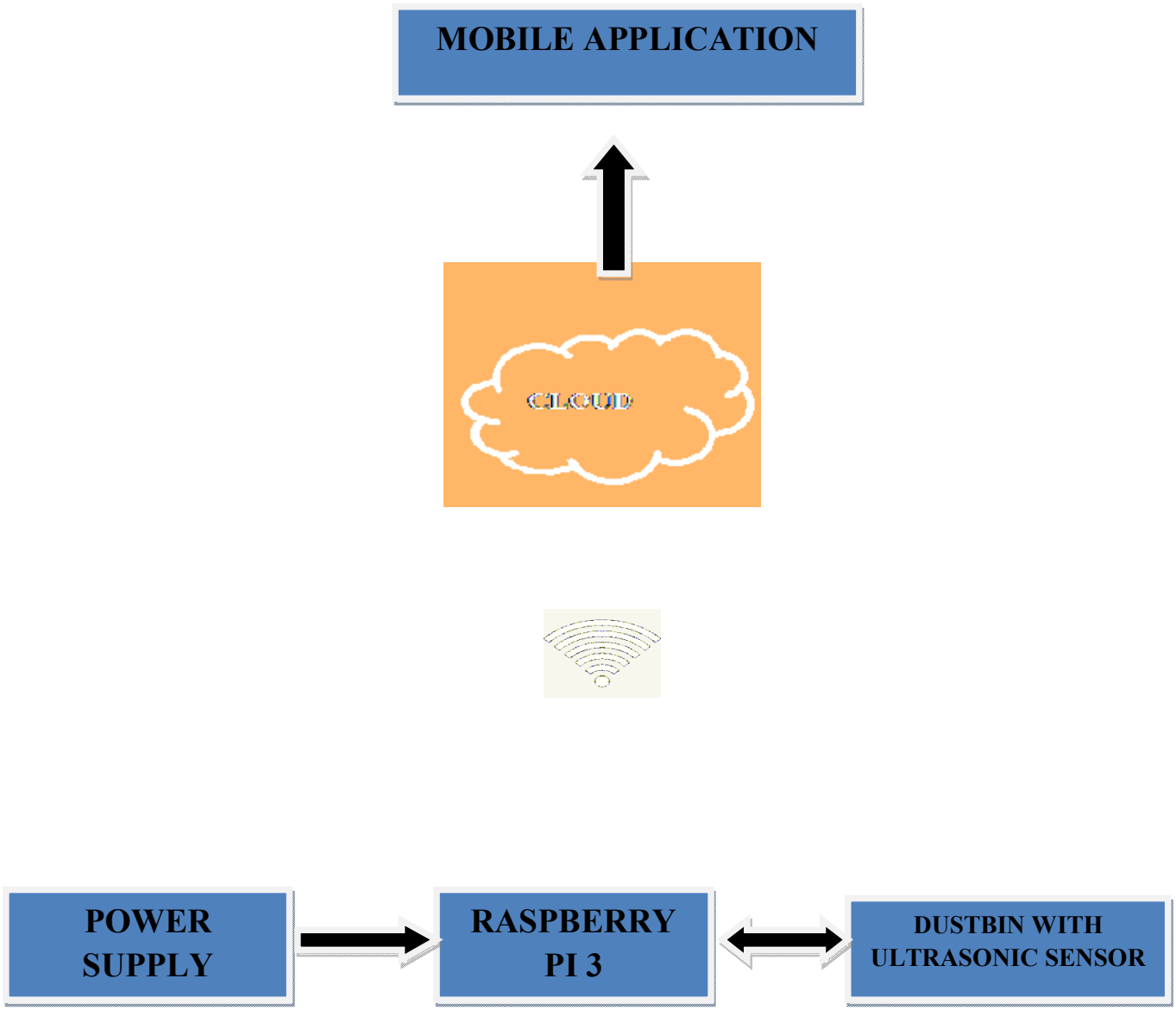
CHAPTER 2

LITERATURE SURVEY

Seven reports were reviewed in detail for the literature review, with the majority of these providing some evidence to support the theory that the introduction of waste collections is associated with a reduction in waste arisings. The following text should be reviewed with consideration given to the fact that these studies were not specifically designed to assess the impact of waste collections on at source food waste reduction. Therefore, evidence is taken from these reports to be used in different context from that in which it was collected. For example, a common theme across all of the reports was the fact that where a reduction in food waste arisings had been observed, there was limited data to confirm how much food waste had simultaneously been diverted from the residual waste stream to home composting and how much was a result of at source waste prevention behaviour. A number of the reports considered the diversion of waste to home composting, as a contributor to waste reduction, as this reduced the food waste arisings collected at the kerbside. Overall the reports demonstrate that while there is some evidence to support the theory that implementing a waste collection can lead to an overall reduction in collected waste, there is currently no significant evidence to demonstrate to what extent this is due to prevention at source as opposed to diversion to home composting. A number of the reports support the need for further research in this area.

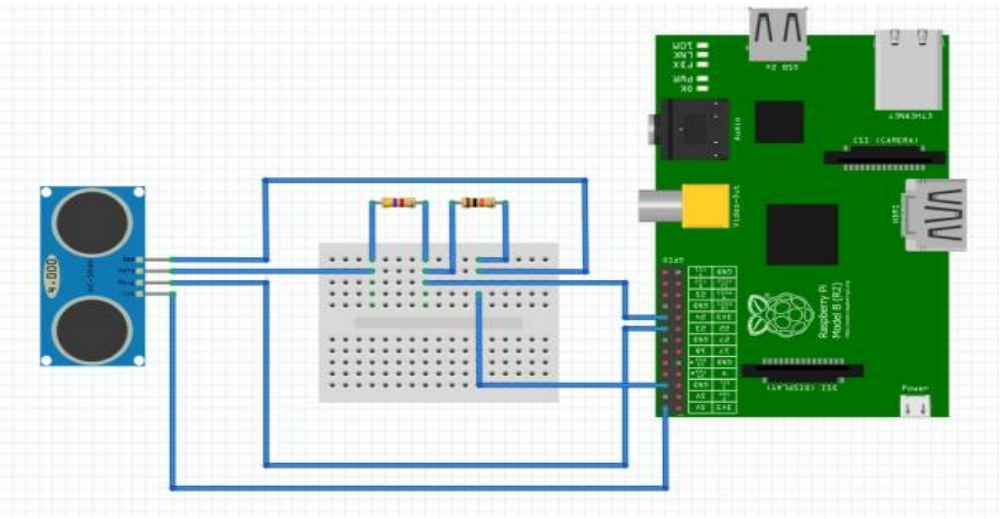
CHAPTER 3

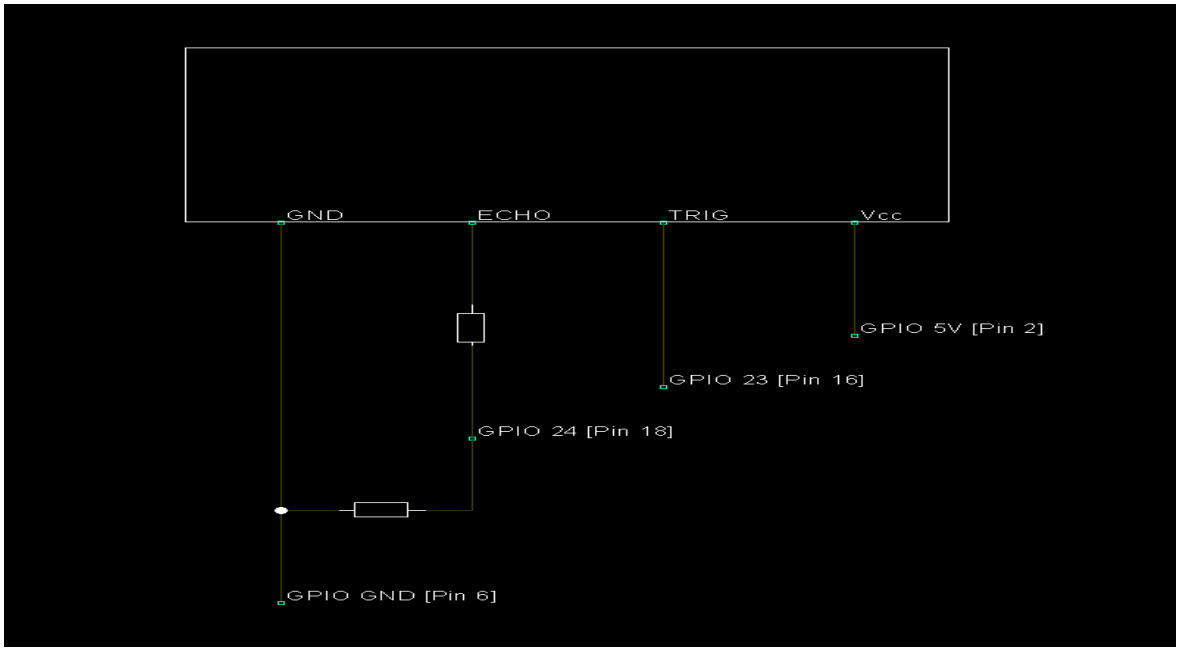
BLOCK DIAGRAM



CHAPTER 4

CIRCUIT DIAGRAM

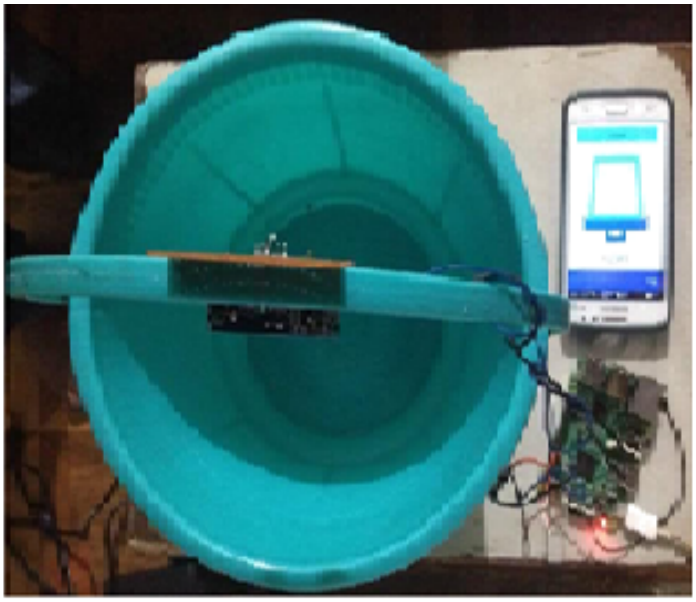




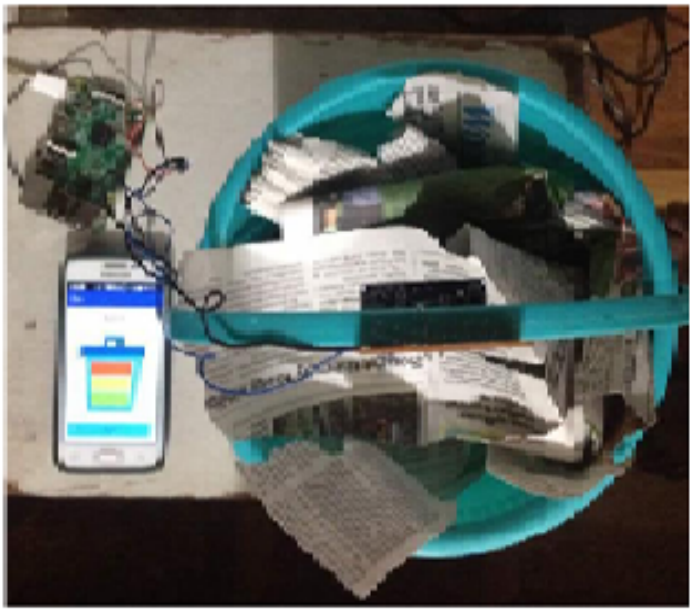
CHAPTER 5

WORKING

This project Smart Waste Management using IOTis a very innovative system which will help to keep the cities clean. This system monitors the garbage bins and informs about the level of garbage collected in the garbage bins via a mobile application. For this the system uses ultrasonic sensors placed over the bins to detect the garbage level and compare it with the garbage bins depth. The system makes use of Raspberry Pi for sending data . The system is powered by a 12V power supply. An application is built to show the status to the user monitoring it. The application gives a view of the garbage bins and highlights the garbage collected in colour in order to show the level of garbage collected.. Thus this system helps to keep the city clean by informing about the garbage levels of the bins by providing image of the bins via IOT application development platform. The authorized person recevies the indication of garbage dustbin is full through the application and then inform the concerned person who is responsible for the collection of garbage where the garbage bin is full in particular areas. The data get stored in the database created and than the data is retrieve In IoT applied to external and public environments, communication is important for service provisioning. In particular, since this type of IoT has a wide service domain, reliable communication is necessary for devices to communicate with each other. Therefore, the SGBs utilized in the proposed system communicate with each other based on a wireless mesh network , securing communication reliability. IoT devices in an external environment may need to move on occasion. With a battery-based power supply, the mobility of the proposed system is secured. In IoT with a wide service domain, data exchanges and services should be conducted seamlessly at any time and any location. User convenience has been enhanced with the advent of IoT.



EMPTY SMARTBIN



FULL FILLED SMARTBIN

CHAPTER 6

WHY IOT ?

The Internet of Things (IoTs) can be described as connecting everyday objects like smart-phones, Internet TVs, sensors and actuators to the Internet where the devices are intelligently linked together enabling new forms of communication between things and people, and between things themselves. Building IoTs has advanced significantly in the last couple of years since it has added a new dimension to the world of information and communication technologies. It is expected that the number of devices connected to the Internet will accumulate from 100.4 million in 2011 to 2.1 billion by the year 2021, growing at a rate of 36% per year. In the year 2011, 80% machine to machine (M2M) connections were made over mobile networks such as 2G and 3G and it is predicted that by 2021, this ratio will increase to 93% since the cost related with M2M over mobile networks are generally cheaper than fixed networks. Now anyone, from anytime and anywhere can have connectivity for anything and it is expected that these connections will extend and create an entirely advanced dynamic network of IoTs.

The development of the Internet of Things will revolutionize a number of sectors, from automation, transportation, energy, healthcare, financial services to nanotechnology. IoTs technology can also be applied to create a new concept and wide development space for smart homes to provide intelligence, comfort and to improve the quality of life. In general, the IoT promotes a heightened level of awareness about our world, and a platform from which to monitor the reactions to the changing conditions that said awareness exposes us to. And, like the advent of the Internet itself, the IoT enables myriad applications ranging from the micro to the macro, and from the trivial to the critical. Since we're focusing here on why the IoT is important, let's turn our attention to the "macro" and the "critical" first, and look at some provocative ideas that are already in development across the globe. Environmental monitoring applications of the IoT typically use sensors to assist in environmental protection by monitoring air or water quality atmospheric or soil conditions and can even include areas like monitoring the movements of wildlife and their habitats. Development of resource constrained devices connected to the Internet also means that other applications like earthquake or tsunami early-warning systems can also be used by

emergency services to provide more effective aid. IoT devices in this application typically span a large geographic area and can also be mobile. It has been argued that the standardization IoT brings to wireless sensing will revolutionize this area.

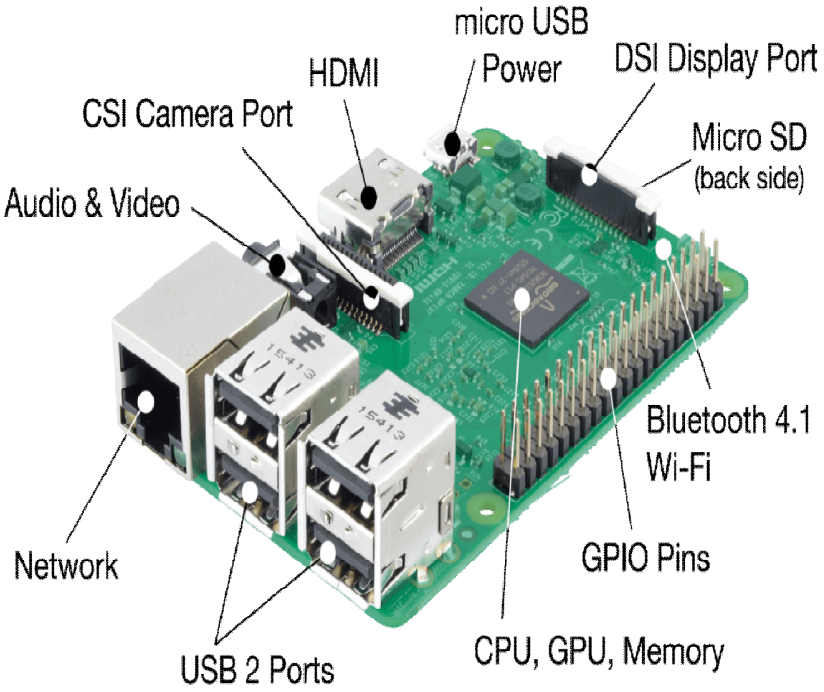
Monitoring controlling operations of urban and rural infrastructures like bridges, railway tracks, on- and offshore- wind-farms is a key application of the IoT. The IoT infrastructure can be used for monitoring any events or changes in structural conditions that can compromise safety and increase risk. It can also be used for scheduling repair and maintenance activities in an efficient manner, 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. 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 infrastructure related areas. Even areas such as waste management can benefit from automation and optimization that could be brought in by the IoT. Integration of sensing and actuation systems, connected to the Internet, is likely to optimize energy consumption as a whole. It is expected that IoT devices will be integrated into all forms of energy consuming devices (switches, power outlets, bulbs, televisions, etc.) and be able to communicate with the utility supply company in order to effectively balance power generation and energy usage. Such devices would also offer the opportunity for users to remotely control their devices, or centrally manage them via a cloud based interface, and enable advanced functions like scheduling (e.g., remotely powering on or off heating systems, controlling ovens, changing lighting conditions etc.).Besides home based energy management, the IoT is especially relevant to the Smart Grid since it provides systems to gather and act on energy and power-related information in an automated fashion with the goal to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity. Using advanced metering infrastructure (AMI) devices connected to the Internet backbone, electric utilities can not only collect data from end-user connections, but also manage other distribution automation devices like transformers and reclosers.

IoT device can be used to enable remote health monitoring and emergency notification systems These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands or advanced hearing aids. Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well.[Other consumer devices to encourage healthy living, such as, connected scales or wearable heart monitors, are also a possibility with the IoT. More and more end-to-end health monitoring IoT platforms are coming up for antenatal and chronic patients, helping one manage health vitals and recurring medication requirements.

CHAPTER 7

HARDWARE REQUIREMENTS

7.1. RASPBERRYPI-3



The Raspberry-Pi is series of credit card-sized single-boardcomputers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and developing countries.Several generations of Raspberry Pis have been released. The first generation Raspberry Pi 1 Model B was released in February 2012. It was followed by a simpler and inexpensive model Model A. In 2014 the foundation released a board with an improved design in Raspberry Pi 1 Model B+. The model laid the current "mainline" form-factor. Improved A+ and B+ models were released a year later. A cut down "compute" model was released in April

2014,and a raspberrypi zero with smaller size and limited input/output (I/O), general-purpose input/output (GPIO), abilities released in November 2015 for US$5. The Raspberry Pi 2 which added more RAM was released in February 2015. Raspberry Pi 3 Model B released in February 2016 is bundled with on-board WiFi and Bluetooth. As of 2016, Raspberry Pi 3 Model B is the newest mainline Raspberry Pi. These boards are priced between US $20-35.All models features Broadcom system on chip, which includes an ARM compatible central processing unit (CPU) and an on chip graphics processing unit (GPU, a VideoCore IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes. Most boards have between one and four USB slots, HDMIand composite video output, and a 3.5 mm phone jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I²C. The B-models have an 8P8C Ethernet port and the Pi 3 has on board Wi-Fi 802.11n and Bluetooth.The Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third party Ubuntu, Windows 10 IOT Core, RISC OS, and specialised media center distributions. It promotes Python and Scratch as the main programming language, with support for many other languages. The default firmware is closed source, while an unofficial open source is available.In February 2016, the Raspberry Pi Foundation announced that they had sold eight million devices, making it the best-selling UK personal computer, ahead of the Amstrad PCW. Sales reached ten million in September 2016. While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium IIof 1997–99. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capability of the Raspberry Pi are roughly equivalent to the performance of the Xbox of 2001.

The LINPACK single node compute benchmark results in a mean single precision performance of 0.065 GFLOPS and a mean double precision performance of 0.041 GFLOPS for one Raspberry Pi Model-B board. A cluster of 64 Raspberry Pi Model B computers, labeled "Iridis-pi", achieved a LINPACK HPL suite result of

1.14 GFLOPS (n=10240) at 216 watts for c. US$4000.Raspberry Pi 2 includes a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It is described as 4–6 times more powerful than its predecessor. The GPU is identical to the original. In parallelized benchmarks, the Raspberry Pi 2 could be up to 14 times faster than a Raspberry Pi 1 Model B+.The Raspberry Pi 3, with a quad-core Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.

7.2. ULTRASONIC SENSOR

In industrial applications, ultrasonic sensors are characterized by their reliability and outstanding versatility. Ultrasonic sensors can be used to solve even the most complex tasks involving object detection or level measurement with millimetre precision, because their measuring method works reliably .No other measuring method can be successfully put to use on such a wide scale and in so many different applications. The devices are extremely robust, making them suitable for even the toughest conditions. The sensor surface cleans itself through vibration, and that is not the only reason why the sensor is insensitive to dirt. The physical principle the propagation of sound works, with a few exceptions,practicallyanyenvironment.The measuring method employed by ultrasonic sensors has been viewed as an excessively complex technology, and only used as a solution for particularly difficult applications. Ultrasonic sensors have proven their reliability and endurance in virtually all industrialsectors.

These sectors include:

- Mechanical engineering/machine tool
- Food and beverage
- Woodworking and furniture
- Building materials
- Agriculture
- Construction
- Pulp and paper
- Material handling
- Level measurement

- Diffuse and Retroreflective Mode Sensors.

Ultrasonic sensors are most commonly used in the diffuse mode. A single ultrasonic transducer is used as both emitter and receiver and is typically contained in the same housing as the evaluation electronics.To reliably detect difficult objects, the majority of our diffuse mode sensors can be converted to retroreflective operation via software parameterization. Some ultrasonic sensors are supplied as retroreflective sensors from the outset.

- Thru-Beam Sensors.

Ultrasonic thru-beam sensors feature an extremely powerful acoustic beam. They offer a large detection range within compact housing dimensions. Unlike diffuse and retroreflective models, these sensors do not continuously switch between transmission and reception modes or wait for an echo signal to arrive. Thus, their response time is considerably faster, resulting in very high switching frequencies.

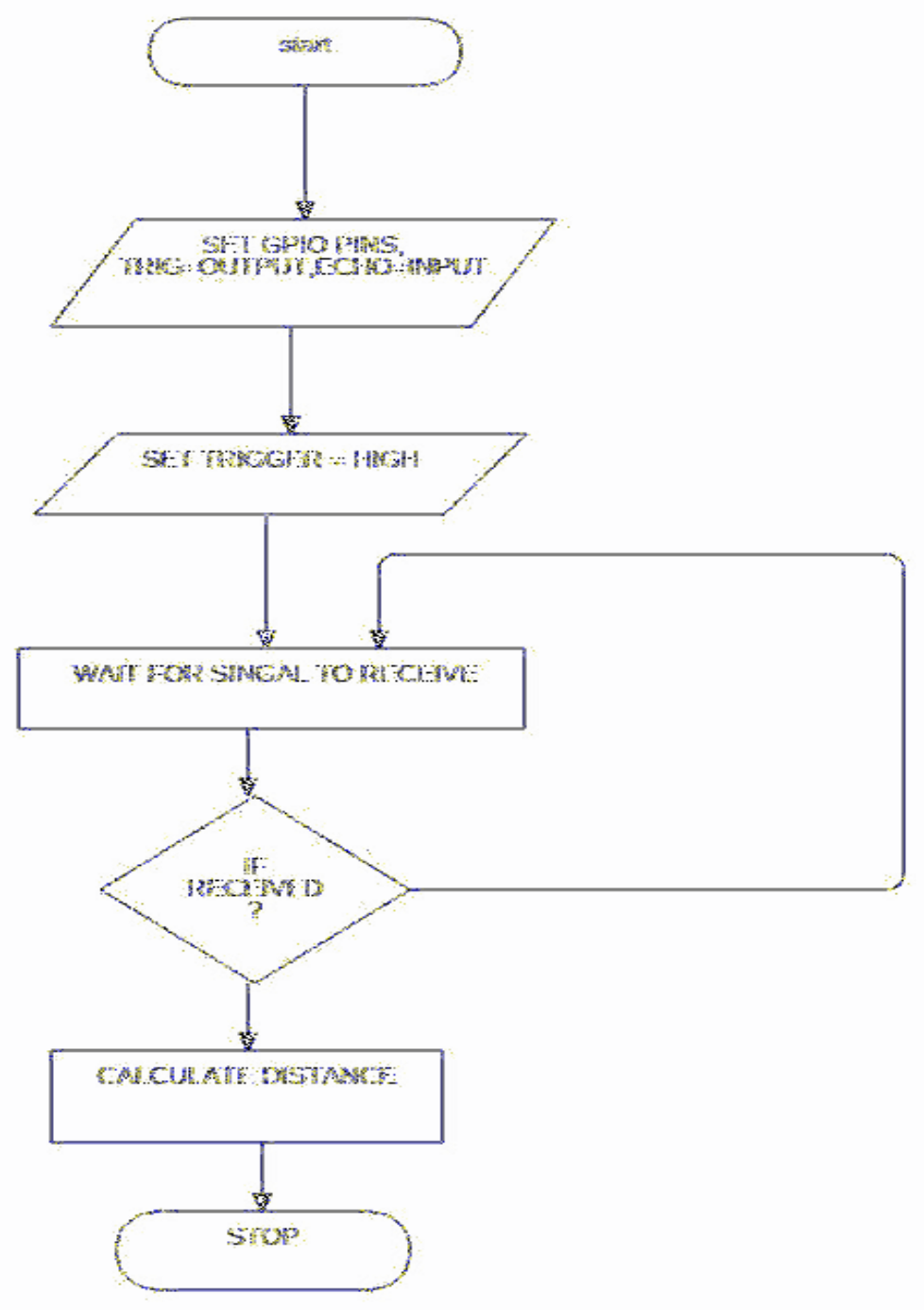
- Double Sheet Sensors.

Double sheet sensors are ultrasonic thru-beam sensors that have been optimized specifically for sheet feed applications, including double sheet detection, label counting, and splice detection.Double sheet and double material sensors distinguish between empty gaps, one, and two layers of material, including paper, metals, wooden boards, and glass panes.When performing label detection tasks, the sensors can distinguish between the label backing and the base material.

CHAPTER 8

CODING

8.1. INTERFACING RASPBERRY PI WITH ULTRASONIC SENSOR

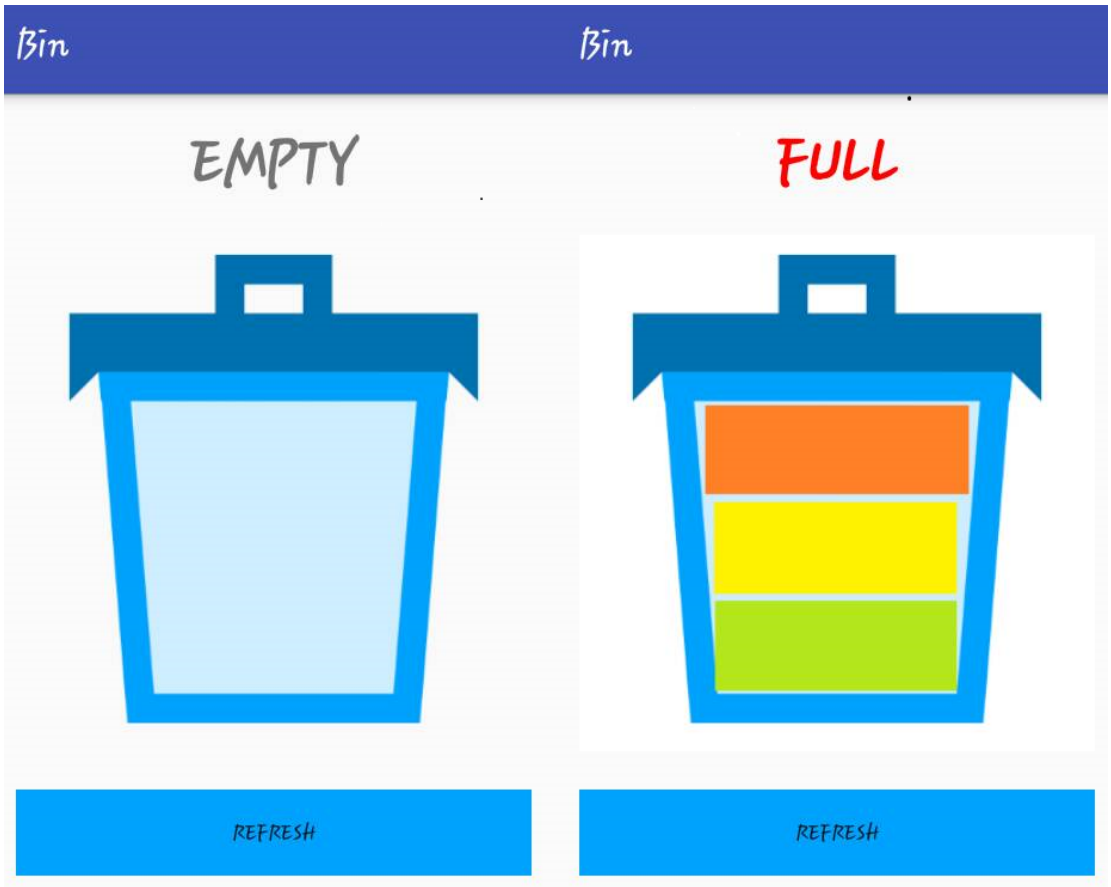


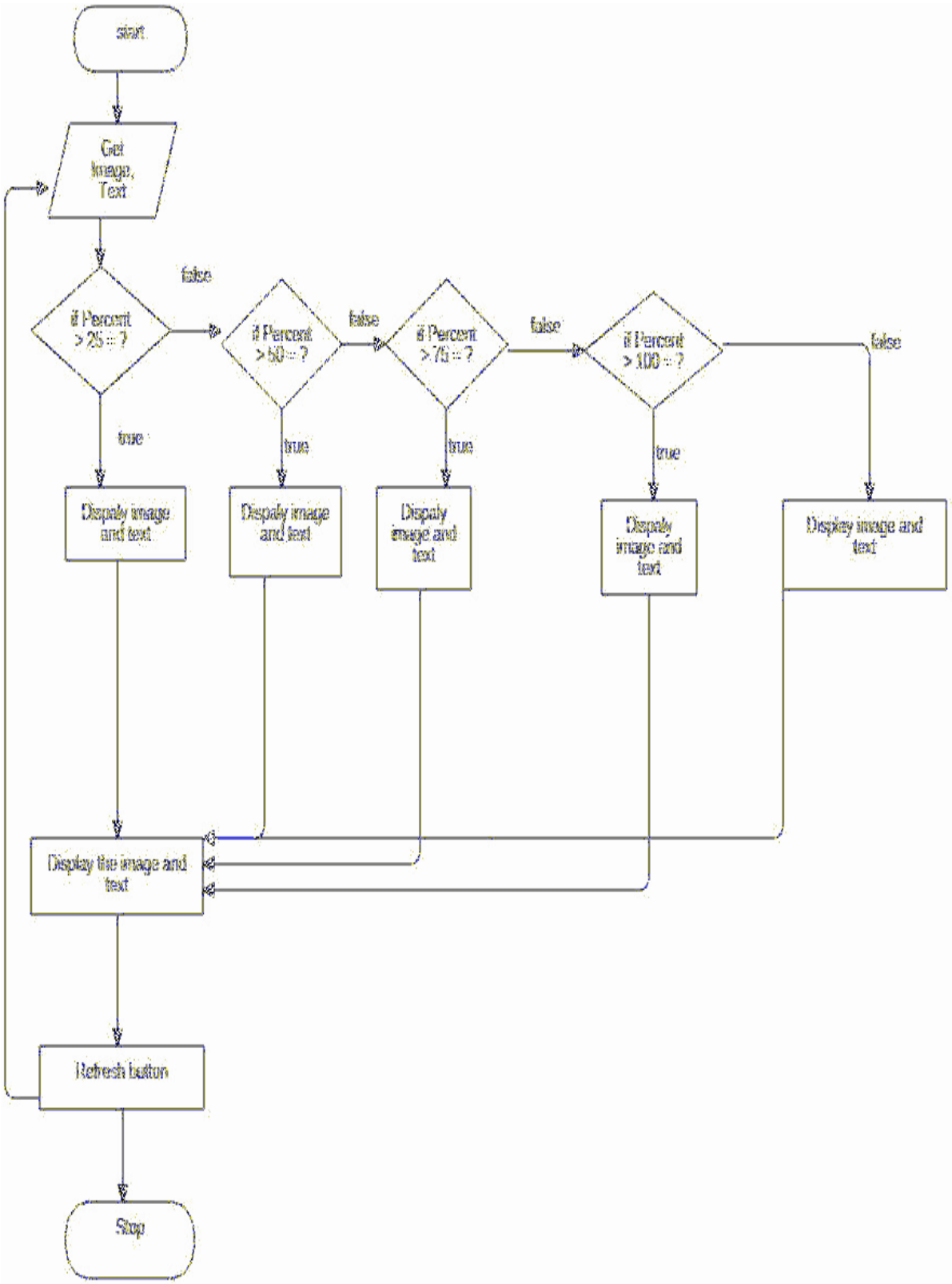
8.1.1 PYTHON PROGAMMING

```
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def distance():
# set Trigger to HIGH
GPIO.output(GPIO_TRIGGER, True)

# set Trigger after 0.01ms to LOW
time.sleep(0.00001)
GPIO.output(GPIO_TRIGGER, False)

StartTime = time.time()
StopTime = time.time()

# save StartTime
while GPIO.input(GPIO_ECHO) == 0:
StartTime = time.time()
# save time of arrival
while GPIO.input(GPIO_ECHO) == 1:
StopTime = time.time()
# time difference between start and arrival
TimeElapsed = StopTime – StartTime
# multiply with the sonic speed (34300 cm/s)
# and divide by 2, because there and back
distance = (TimeElapsed * 34300) / 2

return distance
```

```
if __name__ == '__main__':
try:
while True:
dist = distance()
print ("Measured Distance = %.1f cm" % dist)
percent = (100.0 - (dist * 100/40.0))
url =
"http://localhost:80/demoaddbin.php?bin_id=1&percent_filled="+str(percent)
x= urllib.urlopen(url)
print(x.read)
time.sleep(5)

# Reset by pressing CTRL + C
except KeyboardInterrupt:
print("Measurement stopped by User")
GPIO.cleanup()
```

8.2. SMART WASTE MANAGEMENT APPLICATION DEVELOPMENT




start
Get Image, Text
if Percent > 25 = ?
false
true
if Percent > 50 = ?
false
true
if Percent > 75 = ?
false
true
if Percent > 100 = ?
false
true
Dispaly image and text
Dispaly image and text
Dispaly image and text
Dispaly image and text
Display image and text
Display the image and text
Refresh button
Stop

8.2.1 PROGRAMMING FOR ACESSING DATABASE

```
package com.bin;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Handler;
import android.os.IBinder;
import android.support.annotation.Nullable;
import android.support.v4.app.NotificationCompat;
import android.util.Log;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
```

```
public class GetData extends Service {
String BASE_URL = "http://dustbin.000webhostapp.com/";
String POPMOVIES_BASE_URL = BASE_URL + "getresponsefrombin.php";
SharedPreferences preferences;
SharedPreferences.Editor editor;
@Override
 public void onCreate() {
Toast.makeText(this, "Service Called", Toast.LENGTH_SHORT).show();
 Log.d("Create:","called");
 //addNotification();
 super.onCreate();
 }
 @Override
 public int onStartCommand(Intent intent, int flags, int startId) {
 Log.d("onStartCommand:","called");
 preferences = getSharedPreferences("DustBin",MODE_PRIVATE);
final Handler handler = new Handler();
Runnable runnable = new Runnable() {
@Override
 public void run() {
 Log.d("handler","run();");
 new DustbinTask().execute();
handler.postDelayed(this, 5000);
}
};
//Start
handler.postDelayed(runnable, 1000);
 return START_STICKY;
 }
@Override
public void onDestroy() {
Log.d("Destroy:","called");
```

```
super.onDestroy();
}
@Nullable
@Override
public IBinder onBind(Intent intent) {
Log.d("Bind:","called");
return null;
}
 public class DustbinTask extends AsyncTask<Void, Void,Void>{
@Override
protected void onPreExecute() {
Log.d("onPreExecute","initiate");
try {
if (!new Network(GetData.this).isConnected()) {
Log.d("onPreExecute","No Internet Available!!");
cancel(true);
}
 }
catch (InterruptedException | IOException e) {
 e.printStackTrace();
}
 @Override
 protected Void doInBackground(Void... params) {
 HttpURLConnection urlConnection = null;
 BufferedReader reader = null;
 URL url;
 String MoviesJsonStr;
try {
url = new URL(POPMOVIES_BASE_URL);
 urlConnection = (HttpURLConnection) url.openConnection();
 urlConnection.setRequestMethod("GET");
urlConnection.connect();
InputStream inputStream = urlConnection.getInputStream();
```

```
StringBuilder buffer = new StringBuilder()
reader = new BufferedReader(new InputStreamReader(inputStream));
String line;
while ((line = reader.readLine()) != null) {
buffer.append(line).append("\n");
 }
MoviesJsonStr = buffer.toString();
 getMovieNames(MoviesJsonStr);
} catch (IOException | JSONException e1) {
e1.printStackTrace();
} finally {
if (urlConnection != null) {
 urlConnection.disconnect();
 }
 if (reader != null) {
 try {
reader.close();
} catch (final IOException ignored) {}
}
}
return null;
}
}
 private void getMovieNames(String MovieJsonStr) throws JSONException {
JSONObject MovieJson = new JSONObject(MovieJsonStr);
JSONArray movieLists = MovieJson.getJSONArray("bin_info");
for (int i = 0; i < movieLists.length(); i++) {
JSONObject jMovieDetails = movieLists.getJSONObject(i);
String name = jMovieDetails.getString("bin_id");
 int id = jMovieDetails.getInt("percent_filled");
 Log.d("DATA", name + " " + id);
 MainActivity.percent = id;
 if(id>=80){
```

```
addNotification(id);
}
}
//Log.v("Length: ", String.valueOf(movieLists.length()));
 }
//Show a notification
private void addNotification(int id) {
 int min, max;
 int percentage = preferences.getInt("last_percent",0);
 min = percentage - 5;
max = percentage + 5;
 if (min > id || id > max) {
Intent intent = new Intent(this, MainActivity.class
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
editor = preferences.edit();
editor.putInt("last_percent",id);
 editor.apply();
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0/*Request code*/,
intent, PendingIntent.FLAG_ONE_SHOT);
 //Set sound of notification
 Uri notificationSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
NotificationCompat.Builder notifiBuilder = new NotificationCompat.Builder(this)
.setSmallIcon(R.mipmap.ic_launcher
.setContentTitle(id + "% Dustbin Full")
 .setContentText("Please clear your trash")
 .setAutoCancel(true)
.setSound(notificationSound)
.setContentIntent(pendingIntent);
NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(999 /*ID of notification*/, notifiBuilder.build());
//stopSelf()
```

```
 } else {
        Log.d("addNotification","condition false");
      }
   }
}
```

8.2.2 MY ACTIVITY

```
package com.bin;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
```

```
import java.net.URL;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
 public static int percent = 0;
ImageView image;
TextView text;
SharedPreferences preferences;

 /* EditText ipET;*/

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
 //startAlert();
startService(new Intent(this, GetData.class));
sharedPreference();
image = (ImageView) findViewById(R.id.dustbinimage);
text = (TextView) findViewById(R.id.textpercent);
 /* ipET = (EditText) findViewById(R.id.ipAddr);*/
 if (percent < 25) {
text.setText("EMPTY");
image.setImageResource(R.drawable.trash_icon_empty);
} else if (percent < 50) {
\ text.setText(percent + "% FULL");
image.setImageResource(R.drawable.trash_icon_30);
} else if (percent < 75) {
text.setText(percent + "% FULL");
image.setImageResource(R.drawable.trash_icon_60);
```

```
} else if (percent < 100) {
 text.setText(percent + "% FULL");
 text.setTextColor(Color.BLUE);
 image.setImageResource(R.drawable.trash_icon_full);
} else {
text.setText("FULL");
text.setTextColor(Color.RED);
image.setImageResource(R.drawable.trash_icon_full);
 new GetPercentage(this).execute();
 }
 private void sharedPreference() {
preferences = getSharedPreferences("DustBin",MODE_PRIVATE);
SharedPreferences.Editor editor = preferences.edit();
editor.putInt("last_percent",0);
editor.apply();
 }
 public void startAlert() {
 Intent intent = new Intent(this, MyBroadcastReceiver.class);
 PendingIntent pendingIntent = PendingIntent.getBroadcast(
 this.getApplicationContext(), 234324243, intent, 0);
 AlarmManager alarmManager = (AlarmManager)
getSystemService(ALARM_SERVICE);
//alarmManager.set(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis() + (10000), pendingIntent);
alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis() + 1000, 1000, pendingIntent);
}
public void btnRefresh(View view) {
new GetPercentage(this).execute();
}
```

```
class GetPercentage extends AsyncTask<Void, Void, String>{
String URI = "http://dustbin.000webhostapp.com/getresponsefrombin.php";
ProgressDialog progressDialog;
 Context mContext;
GetPercentage(Context mContext) {
this.mContext = mContext;
 progressDialog = new ProgressDialog(mContext);
progressDialog.setTitle("Refreshing...");
 progressDialog.setMessage("Please wait...");
 }
@Override
protected void onPreExecute()
 Log.d("onPreExecute","initiate");
 try {
 if (!new Network(mContext).isConnected()) {
Log.d("onPreExecute","No Internet Available!!");
 cancel(true);
  return;
  }
  } catch (InterruptedException | IOException e) {
  e.printStackTrace();
   return;
  }
  progressDialog.show();
  }
 @Override
  protected String doInBackground(Void... params) {
   HttpURLConnection connection;
   BufferedWriter writer;
   InputStream is;
```

```
 String response = null;
try {
 URL url = new URL(URI);
connection = (HttpURLConnection) url.openConnection();
connection.setRequestMethod("GET");
connection.setDoOutput(true);
 connection.connect();
writer = new BufferedWriter(new
OutputStreamWriter(connection.getOutputStream()));
writer.flush();
writer.close();
 if(connection.getResponseCode() == HttpURLConnection.HTTP_OK){
is = connection.getInputStream();// is is inputstream
} else {
is = connection.getErrorStream();
}
BufferedReader reader = new BufferedReader(new InputStreamReader(
 is, "UTF-8"), 8);
StringBuilder sb = new StringBuilder();
String line = null;
while ((line = reader.readLine()) != null) {
sb.append(line).append("\n");
}
is.close();
response = sb.toString();
Log.d("JSON", response);
} catch (IOException e) {
progressDialog.dismiss()
e.printStackTrace();
 }
```

```
 return response ;
}
 @Override
 protected void onPostExecute(String s) {
progressDialog.dismiss();
try {
JSONObject respJsonObject = new JSONObject(s);
JSONObject binInfo =
respJsonObject.getJSONArray("bin_info").getJSONObject(0);
 percent = binInfo.getInt("percent_filled");
SharedPreferences.Editor editor = preferences.edit();
editor.putInt("last_percent",percent);
 editor.apply();
if (percent < 25) {
 text.setText("EMPTY");
 image.setImageResource(R.drawable.trash_icon_empty);
} else if (percent < 50) {
text.setText(String.format(Locale.US,"%d%% FULL", percent));
image.setImageResource(R.drawable.trash_icon_30)
} else if (percent < 75) {
 text.setText(String.format(Locale.US,"%d%% FULL", percent));
 image.setImageResource(R.drawable.trash_icon_60);
} else if (percent < 100) {
text.setText(String.format(Locale.US,"%d%% FULL", percent));
text.setTextColor(Color.BLUE);
image.setImageResource(R.drawable.trash_icon_full);
} else {
 text.setText("FULL");
text.setTextColor(Color.RED);
image.setImageResource(R.drawable.trash_icon_full);
```

```
}
} catch (JSONException e) {
e.printStackTrace();
}
}
}
}
```

8.2.3. PROGRAMMING FOR CONNECTING APPLICATION TO THE INTERNET

```
package com.bin;
import android.content.Context;
import android.net.ConnectivityManager;
import android.util.Log;

import java.io.IOException;

/**
 * Created by Sylvester on 03-Mar-17.
 */

class Network {
private Context mContext;
Network(Context mContext) {
this.mContext = mContext;
}
private boolean isNetworkAvailable() {
final ConnectivityManager connectivityManager = ((ConnectivityManager)
mContext.getSystemService(Context.CONNECTIVITY_SERVICE));
return connectivityManager.getActiveNetworkInfo() != null &&
connectivityManager.getActiveNetworkInfo().isConnected();
}
 boolean isConnected() throws InterruptedException, IOException
{
 if (isNetworkAvailable()) {
String command = "ping -c 1 google.com";
return (Runtime.getRuntime().exec (command).waitFor() == 0);
```

```
} else {
Log.d("isInternetAvailable", "No Network Connected");
return false;
}
}
}
```

CHAPTER 9

LIMITATIONS OF THE EXISTING SYSTEM

- Time consuming and less effective: trucks go and empty containers whether they are full or not.
- High costs.
- Unhygienic Environment and look of the city.
- Bad smell spreads and may cause illness to human beings.
- More traffic and Noise.

CHAPTER 10

ADVANTAGES OF THE PROPOSED SYSTEM

- Real time information on the fill level of the dustbin.
- Deployment of dustbin based on the actual needs.
- Cost Reduction and resource optimization.
- Improves Environment quality
- Fewer smells
- Cleaner cities
- Intelligent management of the server.
- Effective usage of dustbins.

CHAPTER 11

CONCLUSION

We have implemented real time waste management system by using smart dustbins to check the fill level of smart dustbins whether the dustbin are full or not. In this system the information of all smart dustbins can be accessed from Anywhere and anytime by the concern person and he/she can take a decision accordingly. 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.

CHAPTER 12

REFERENCES

[1] KanchanMahajan, "Waste Bin Monitoring System Using Integrated Technologies", International Journal of Innovative Research in Science, Engineering and Technology, Issue 3, Issue 7, July 2014.

[2] M. Al-Maaded, N. K. Madi, Ramazan Kahraman, A. Hodzic, N. G. Ozerkan , An Overview of Solid Waste Management and Plastic Recycling in Qatar, Springer Journal of Polymers and the Environment, March 2012, Volume 20, Issue 1, pp 186-194.

[3]Raghumani Singh, C. Dey, M. Solid waste management of Thou bal Municipality, Manipur- a case study Green Technology and Environmental Conservation (GTEC 2011), 2011 International Conference Chennai 21 – 24

[4] Vikrant Bhor, "Smart Garbage management System International Journal of Engineering Research & Technology (IJERT), Vol. 4 Issue 03, March-20152000.

[5] Narayan Sharma,, "Smart Bin Implemented for Smart City", International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015