CHAPTER 1

INTRODUCTION

1.1 GENERAL

Technology has reformed almost every part of human beings. But it seems to lag behind in enhancement of waste disposal system. The nation lags behind in waste recycling and management system. Hence this prototype is aimed at modifying one significant part of waste management that can improve the efficiency of waste management.

Usually, the Waste Trucks will scavenge streets after garbage bins and will collect the waste. Sometimes a road is visited after the Bin overflows with garbage and a complaint is raised to the corporation. This has a lot of disadvantages as wastes are not properly managed, the bins overflow with scraps and litter the surroundings; traffic ensures as the truck blocks the road at random times.

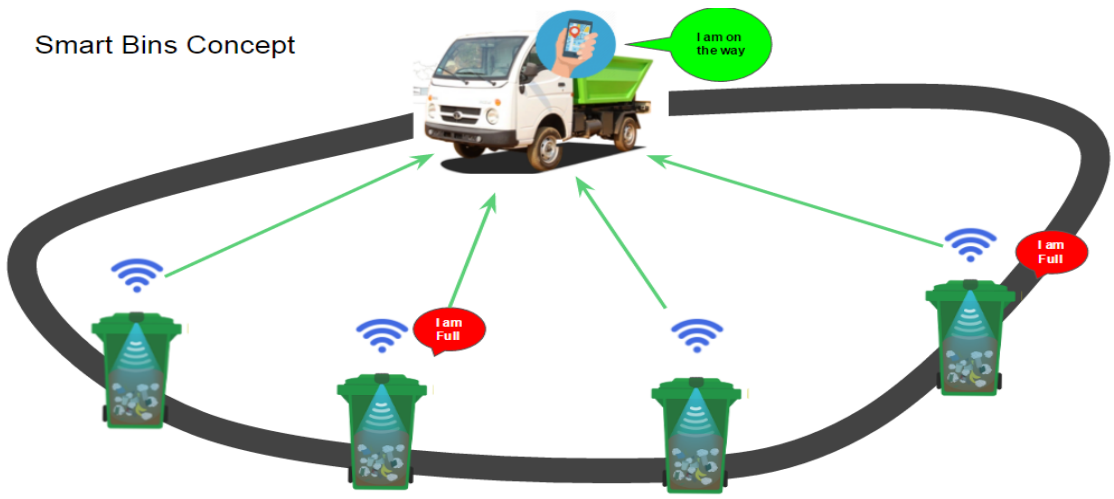

Fig 1.1 The base visualization

To combat this issue, we propose an idea featuring it with a level indicating sensor integrated to the internet using Wi-Fi Module. The whole setup is connected to a central microcontroller. Likewise, all the Bins in an area are given the similar configuration, and these are interconnected wirelessly. Each area is distinguished and is given a separate new identification in the 'Coordination Controller.' This is a web portal designed precisely where all the data of the bins are accumulated online in this portal. So by a pre-fed algorithm, the Coordination Controller will be able to determine which area needs immediate attention. So when an area needs immediate attention, the nearby truck is notified immediately. The algorithm prioritizes areas on a basis like Schools; Hospitals are given high priority, while the less critical area is given least priority. It will be able to analyze which area requires regular attention based on the history and will provide it with a high priority. So if a particular area has a maximum waste generation, based on a pattern that area will be offered primary priority.

1.2 SCOPE OF THE PROJECT

a. To be implemented at places mainly schools and hospital, park, malls, apartments so that these areas can be kept clean.
b. This method is Simple, Cheap and easy to monitor and control.
c. Will stop the spread of infectious disease which has been one of a major thread in our nation.
d. Proper waste disposal is critical due to the fact that certain types of wastes can be hazardous and can contaminate the environment if not handled properly. These types of waste also have the potential to cause disease or get into water supplies.
e. According to recent articles and survey our nation has been the poor in keeping a proper and balance waste management system.

1.3 DOMAIN OVERVIEW

Embedded systems is a computer based system which binds both hardware and software functions in a whole used to perform a user specified task. Embedded systems is used as a navigation tools like global positioning system (GPS), Mobile phones, Sensors, Telecom applications for communication between various devices performing a pre-defined task. We use real time operating system RTOS as per performance and functional requirements. We bridge together Embedded applications with Internet of things (IOT) which is system of interrelated computing devices used to transfer data over network without requiring human to human or human to computer interaction.

CHAPTER 2

LITERATURE SURVEY

2.1 GENERAL

A Detailed research analyze on the existing methodologies that is being implemented in our country will set a rise to a lot of constraints. Usually, the Waste Vans will scavenge streets after trash bins and will collect the waste. Sometimes a road is visited after the Bin overflows with garbage and a complaint is raised to the corporation. This has a lot of disadvantages as wastes are not properly managed, the bins overflow with scraps and litter the surroundings; traffic ensures as the truck blocks the road at random times. Our government has been continuously introducing a lot of schemes to develop the current waste management system which is being followed. A recent survey has recorded that India has the poorest waste clearance methodology. But with the technologies and machinery available this disastrous situation can be elucidated. Several demerits which have stemmed up can be solved and by the implementation of this technology, it will result in a huge impact on the development of the nation's reputation. This will surely will help our country set high standards. Countries like Denmark, Sweden, and Iceland have achieved perfection to an extent such that it has begun importing waste from other countries to keep their recycle plants running.

2.2 EXISTING MODELS AROUND THE WORLD

From research reference stated, Pune Municipal Corporation is administrating the city and serving the citizens by using new technology. Using the new technologies like GPS Tracking System, Ultra High Frequency Radio Frequency Identification, IOT Sensors along with mobile

and web application to improve and smoothen ground level mechanisms for waste collection and efficient processing and re-cycling of waste.

From an another paper four young men from Nashik came up with a unique idea in a bid to further their efforts to keep the country litter-free maybe they could entice people into disposing waste in waste-bins if they could promise them rewards and money. This has recorded in “The Hindu” Newspaper stating people has initiated to adapt to such technologies even more when such devolvement’s is being implement.

Fig 2.1 The Component Visualization

Thus we aimed at modifying one significant part of waste management that can prove to improve the efficiency of waste management. The waste bins is featured with a level indicating sensor integrated to the internet using Wi-Fi Module. The whole setup is connected to a central microcontroller. Likewise, all the Bins in an area are given the similar

configuration, and these are interconnected wirelessly. Each area is distinguished and is given a separate new identification in the ‘Coordination Controller.’ This is a web portal designed precisely where all the data of the bins are accumulated online in this portal. So by a pre-fed algorithm, the Coordination Controller will be able to determine which area needs immediate attention. The trucks will be fitted with a GPS sensor, and an App-enabled PDA will be handed to the driver. So when an area needs immediate attention, the nearby truck is notified immediately.

CHAPTER 3

SYSTEM ANALYSIS

3.1 GENERAL

The system that is being followed in our country is very outdated and ancient. Normally, the Government allocated waste trucks is send to places across the urban and some sub-urban areas. At the point of visit to each waste bins are been recycled. In some cases these trash bin might also overflow which eventually leaving garbage to spill out onto the pathways or roads. Several countries are now started different ways to keep the environment clean. But there are a lot of demerits and a proper method is not found yet.

The existing system has a poor processing method which leads to environment hazards. Even the monitoring of the waste is not within the control. This will surely cause to a big mess in the near future.

3.1.1 Constraints

a. Causes spread of infectious disease which has been one of a major thread in our nation.
b. Proper waste disposal is critical due to the fact that certain types of wastes can be hazardous and can contaminate the environment if not handled properly.
c. These types of waste also have the potential to cause disease or get into water supplies.
d. According to recent articles and survey our nation has been the poor in keeping a proper and balance waste management system.
e. Overflowing of dustbins along roadsides and localities.

3.2 PROPOSED SYSTEM

We propose an idea where we enhance the waste bins by featuring it with a level indicating sensor integrated to the internet using Wi-Fi Module. The whole setup is connected to a central microcontroller. Likewise, all the Bins in an area are given the similar configuration, and these are interconnected wirelessly. Each area is distinguished and is given a separate new identification in the 'Coordination Controller.' This is a web portal designed precisely where all the data of the bins are accumulated online in this portal. So by a pre-fed algorithm, the Coordination Controller will be able to determine which area needs immediate attention. The trucks will be fitted with a GPS sensor, and an App-enabled PDA will be handed to the driver. So when an area needs immediate attention, the nearby truck is notified immediately. The algorithm prioritizes areas on a basis like Schools; Hospitals are given high priority, while the less critical area is given least priority. It will be able to analyze which area requires regular attention based on the history and will provide it with a high priority. So if a particular area has a maximum waste generation, based on a pattern that area will be offered primary priority. So when the level of a bin reaches a maximum, it'll be immediately reported through the Coordination Controller to the nearby truck and that particular container is emptied. This method is Simple, Cheap and easy to implement. This will benefit over the older process in many ways. The wastes are managed efficiently, regular collection ensures. Littering and overflow of waste is prevented. Schools and Hospital areas can be kept clean. A Microcontroller, Wi-Fi Module and the level sensor will be able to change the way Waste is managed.

3.2.1 Modification

a. The system is automated and involves internet of things (IOT) were all the system output, sensor processing, feedback is send automatically to the assigned person personal mobile and can be monitored from a central processing system anywhere across the whole world.
b. When load inside the waste bin reaches to a specified marginal value it automatically sends a default alert notification to the assigned controller or truck driver to an App through Wi-Fi connectivity.
c. Proximity sensor enabled outside the trash bins at appropriate positions, so if any trash is kept outside it automatically triggers a buzzer sound indicating the trash needs to be thrown inside the bin.

3.2.2 Advantages

a. It will stop overflowing of dustbins along roadsides and localities.
b. The filling and cleaning time of smart bin will also be reduced thus making empty and clean dustbins available to common people.
c. It also aims at creating a clean as well as green environment.
d. By using the route algorithm it will smartly find the shortest route thus it will reduce the number of vehicles used for garbage collection.
e. Send optimized routes directly to drivers.
f. It will reduce fuel Consumption.
g. Less amount of fuel consumed by vehicles thus can save a large amount of money as well.

CHAPTER 4

REQUIREMENT SPECIFICATIONS

4.1 GENERAL

The requirement specification is a technical specification of requirements for the software products. It is the first step in the requirement analysis process. It lists the requirements of a particular software system including functional, performance and security requirements. It lays out functional and nonfunctional requirements. The requirements also provide usage scenario from a user, an operational and an administrative perspective. The purpose is to provide the detailed overview of the project, its parameters, goals, the user interface, and hardware and software requirements.

4.2 HARDWARE AND SOFTWARE SPECIFICATIONS

The hardware and software specifications are mentioned as below.

4.2.1 Hardware Specifications

a. Raspberry Pi 3 Microcontroller
b. Global Positioning System (GPS)
c. Ultra Sound Sensor
d. Proximity Sensor
e. Buzzer Indicator
f. TFT LED Screen
g. Battery Power Bank

4.2.1.1 Hardware Description

a. Ultra Sound Sensor

Model	: HC SR04
Operating Voltage	: 5V DC
Operating Frequency	: 40 Hz
Operating Current	: 15 mA

Ultrasound sensor is used as a level indicating sensor. It his placed at appropriate positions inside the trash bin such that the total amount of waste load that is being stored in the bin can be monitored and other actions can be taken. The maximum range in which sound waves can be transmitted is 4m and the minimum range in which sound waves can be transmitted is 2cm.

b. Proximity Sensor

Model	: Silicon IR Proximity Sensor
Operating Voltage	: 5V DC
Operating Current	: 300 mA

The use of proximity sensor is to detect any object like waste baggage when thrown or kept just outside the waste bin. This sensor is placed at appropriate position like outer front side of the trash bin. Its maximum range of detect is approximately 2m. The detection range can be calibrated as per the distance from bin to ground with proper calculations.

c. Buzzer

Operating Voltage : 5V DC

The piezo buzzer produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle. These buzzers is used as an alert to the user of an event corresponding to a switching action, counter signal or sensor input. It is used as an alarming circuits. When Proximity sensor is triggered or Ultrasound Sound Sensor reaches its user defined marginal level (Varies according to the storage capacity of the bins being used) the buzzer is triggered.

d. Global Positioning System

Model : Neo 6M U-Blox

Operating Voltage : 5V DC

Global Positioning System is used for tracking the location of each bins located in various places. The Global Positioning System is connected up with satellites, ground stations, and receivers. Once the receiver calculates its distance from four or more satellites, it knows exactly where you are. It is interfaced with the central microcontroller when ultrasound level indicating sensor level reaches its marginal level it automatically enables the GPS placed inside the waste bin. GPS locks the exact position of that particular bin is been overloaded.

e. Tft Led Screen

TFT LED Screen is interfaced with the controller. The messages that is needed to be notify can be displayed. It has an operating voltage of 5V. Important messages, notification, photos, can be displayed.

f. Battery Power Bank

Capacity	: 2800mAH
Operating Voltage	: 5V DC

Recharge power bank can be used to provide supply to central microcontroller. The micro controller used will separate and supply the required amount of power to each hardware components. This battery power pack is rechargeable and can get charged and used again and again.

g. Central Microcontroller

Model	: Raspberry Pi 3 Model B
Operating Voltage	: 5V DC

a. Broadcom BCM2837 64bit ARMv7 Quad Core Processor powered Single Board Computer running at 1.2GHz
b. 1 GB RAM
c. BCM43143 Wi-Fi on board
d. Bluetooth Low Energy (BLE) on board
e. 40pin extended GPIO
f. 4 x USB 2 ports
g. 4 pole Stereo output and Composite video port
h. Full size HDMI
i. CSI camera port for connecting the Raspberry Pi camera
j. DSI display port for connecting the Raspberry Pi touch screen display
k. Micro SD port for loading your operating system and storing data
l. Upgraded switched Micro USB power source (now supports up to 2.4 Amps)
m. Expected to have the same form factor has the Pi 2 Model B, however the LEDs will change position

4.2.2 Software Specifications

Microcontroller Software	:	Linux
Programming Language	:	Python 2.7 and Python 3
IDE for Software Development	:	Leafpad
Compiler for Python	:	Python IDLE
Operating System	:	Linux Distro – Raspbian Stretch
Mobile Application	:	Pushetta
Mobile Operating System	:	Android
Software for Android Application Interfacing for Python	:	Pushetta API packages
Tools Used	:	Python 2.7 and 3

4.2.2.1 Software Description

Linux

Linux is packaged in a form known as a Linux distribution (or distro for short) for both desktop and server use. The defining component of a Linux distribution is the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds.

Linux was originally developed for personal computers based on the Intel x86 architecture, but has since been ported to more platformsthan any other operating system. Because of the dominance of the Linux kernel-

based Android OS on smartphones, Linux has the largest installed base of all general-purpose operating systems. Linux is also the leading operating system on servers and other big iron systems such as mainframe computers, and the only OS used on TOP500 supercomputers (since November 2017, having before gradually eliminated all competitors). It is used by around 2.3% of desktop computers. The Chromebook, also runs the Linux kernel-based Chrome OS

Python

Python is an interpreted high level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is source software and has a community-based development model, as do nearly all of its variant implementations. C, Python is managed by the non-profit Python Software Foundation.

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such

as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

a. Graphical user interfaces
b. Web frameworks
c. Multimedia
d. Databases
e. Networking
f. Test frameworks
g. Automation
h. Web scraping
i. Documentation

j. System administration

k. Scientific computing

l. Text processing

m. Image processing

Leafpad

Leafpad is an open source text editor for Linux, BSD, and Maemo. Created with the focus of being a lightweight text editor with minimal dependencies, it is designed to be simple and easy-to-compile. Leafpad is the default text editor for LXDE Desktop environment, including Lubuntu, and also Xubuntu.

Released under the terms of the GNU General Public License, Leafpad is free software

Leafpad is a simple GTK+ text editor that emphasizes simplicity. As development focuses on keeping weight down to a minimum, only the most essential features are implemented in the editor. Leafpad is simple to use, is easily compiled, requires few libraries, and starts up quickly.

Leafpad's features include a code set option, auto code set detection, an unlimited Undo/Redo feature, and drag and drop capabilities.

Leafpad has a small footprint compared to editors such as gedit or Kate.

Currently Leafpad has the following features:

a. Code set option (Some OpenI18N registered)
b. Auto code set detection (UTF-8 and some code sets)

c. Unlimited Undo/Redo
d. Auto/Multi-line Indent
e. Display line numbers
f. Drag and Drop
g. Printing

Python Idle

IDLE (short for integrated development environment or integrated development and learning environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit (wrapper functions for Tcl/Tk).

IDLE is intended to be a simple IDE and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

According to the included README, its main features are:

Multi-window text editor with syntax highlighting, auto completion, smart indent and other.

Python shell with syntax highlighting.

Integrated debugger with stepping, persistent breakpoints, and call stack visibility.

IDLE has been criticized for various usability issues, including losing focus, lack of copying to clipboard feature, lack of line numbering options,

and general user interface design; it has been called a "disposable" IDE, because users frequently move on to a more advanced IDE as they gain experience.

Linux Distro – Raspbian

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

The initial build of over 35,000 Raspbian packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012. However, Raspbian is still under active development with an emphasis on improving the stability and performance of as many Debian packages as possible.

Raspbian is an unofficial port of Debian Wheezy armhf with compilation settings adjusted to produce optimized "hard float" code that will run on the Raspberry Pi. This provides significantly faster performance for applications that make heavy use of floating point arithmetic operations. All other applications will also gain some performance through the use of advanced instructions of the ARMv6 CPU in Raspberry Pi.

Although Raspbian is primarily the efforts of Mike Thompson (MP Thompson) and Peter Green (plug wash), it has also benefited greatly from the enthusiastic support of Raspberry Pi community members who wish to get the maximum performance from their device.

Pushetta

Pushetta is a system made to push real time notifications to many different devices (mobile phones, browsers, smart TV ...).

It address a simple problem: get real time information without using outdated methods like Emails.

Pushetta is made to make it simple send broadcast communications to groups of subscribers. It works in a really simple way: as publisher we can create a thematic group, every user that subscribes this group will receive a notification every time you push a message.
It can be compared with SMS with many advantages:

a. It's free of charge
b. No needs of phone numbers or other personal data
c. Sender gets accurate statistics about subscribers and delivered messages
d. Can be extended to support devices other than phones (i.e. Smart TV, web browsers, ...)

Push notification are powerful tools but today this power is unexpressed, Pushetta is thought to make them available for everyone. Main field for which is thought (although not limited to) is IoT.

First step to begin with Pushetta is to make a Channel. A Channel represents a topic it will be subscribed by users interested in receiving information's about this topic. It can be better explained with a simple use case: a weather forecast company creates a Channel named "Meteo Torino", every day the company sends a notification with day forecast. A user that live in Torino can subscribes to this Channel using one of Apps and every day a notification will inform him about forecast in Torino.
Pushetta authentication is token based, a User registered on Pushetta get a token he/she have to use with API calls. You can get your token from API itself of reading it in Dashboard. You can also send notifications from web using Channel detail page on this site.

To use Pushetta with Python We need to install pushetta lib, using following command

```
pip install pushetta
```

When the lib is installed using it is as simple as the following sample

```
from pushetta import Pushetta

API_KEY="00112233445566778899aabbccddeeff00112233"

CHANNEL_NAME="MyChannel"

p=Pushetta (API_KEY)

p.pushMessage (CHANNEL_NAME, "Hello World")
```

CHAPTER 5

SYSTEM DESIGN

5.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

5.2 ARCHITECTURE DIAGRAM

An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

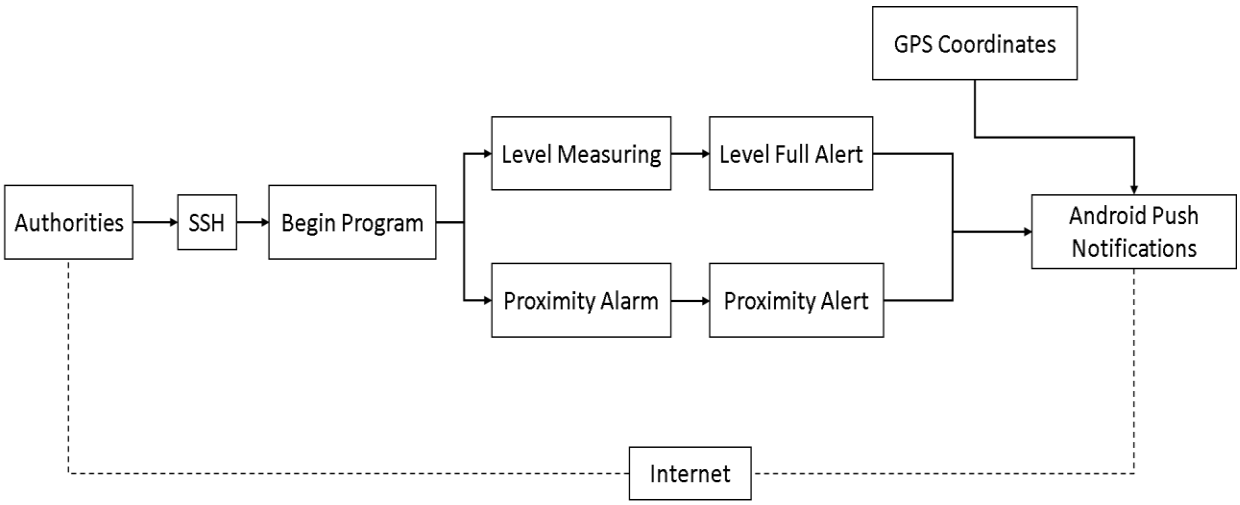

Fig 5.1 Architecture Diagram

5.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modelling its process aspects.

It concerns things like where the data will come from and go to as well as where it will be stored. The graphical depiction identifies source of data and how it interacts with other data sources to reach a common output. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

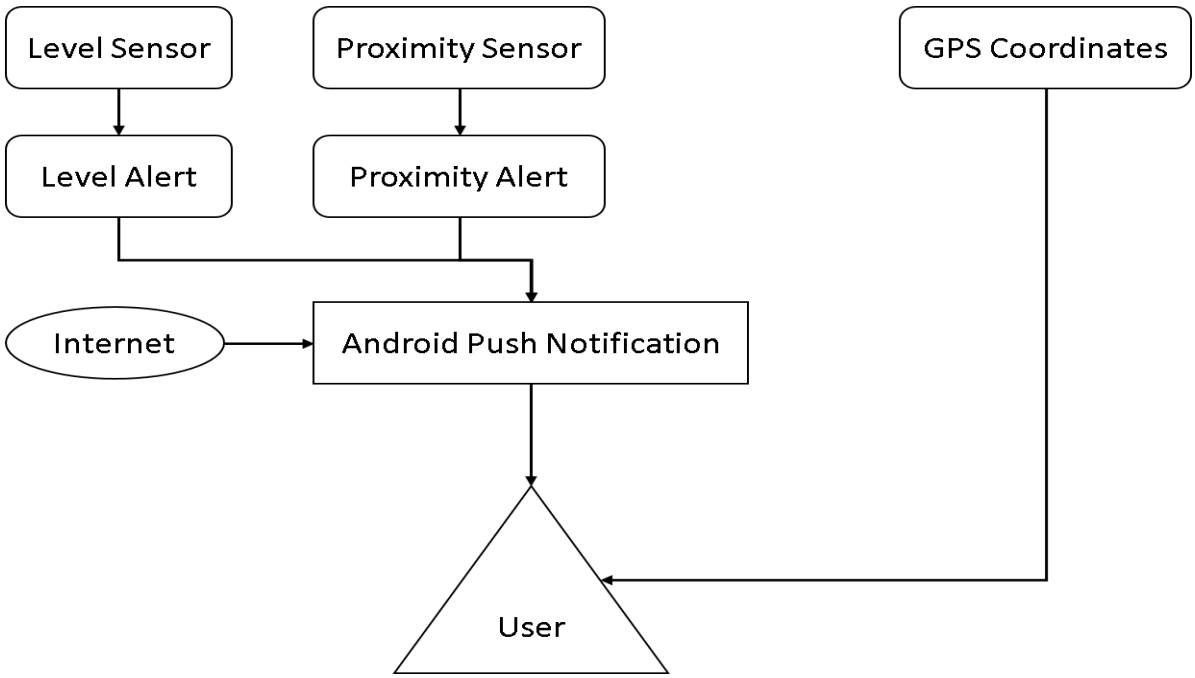

Fig 5.2 Data Flow Diagram

5.4 SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in

the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

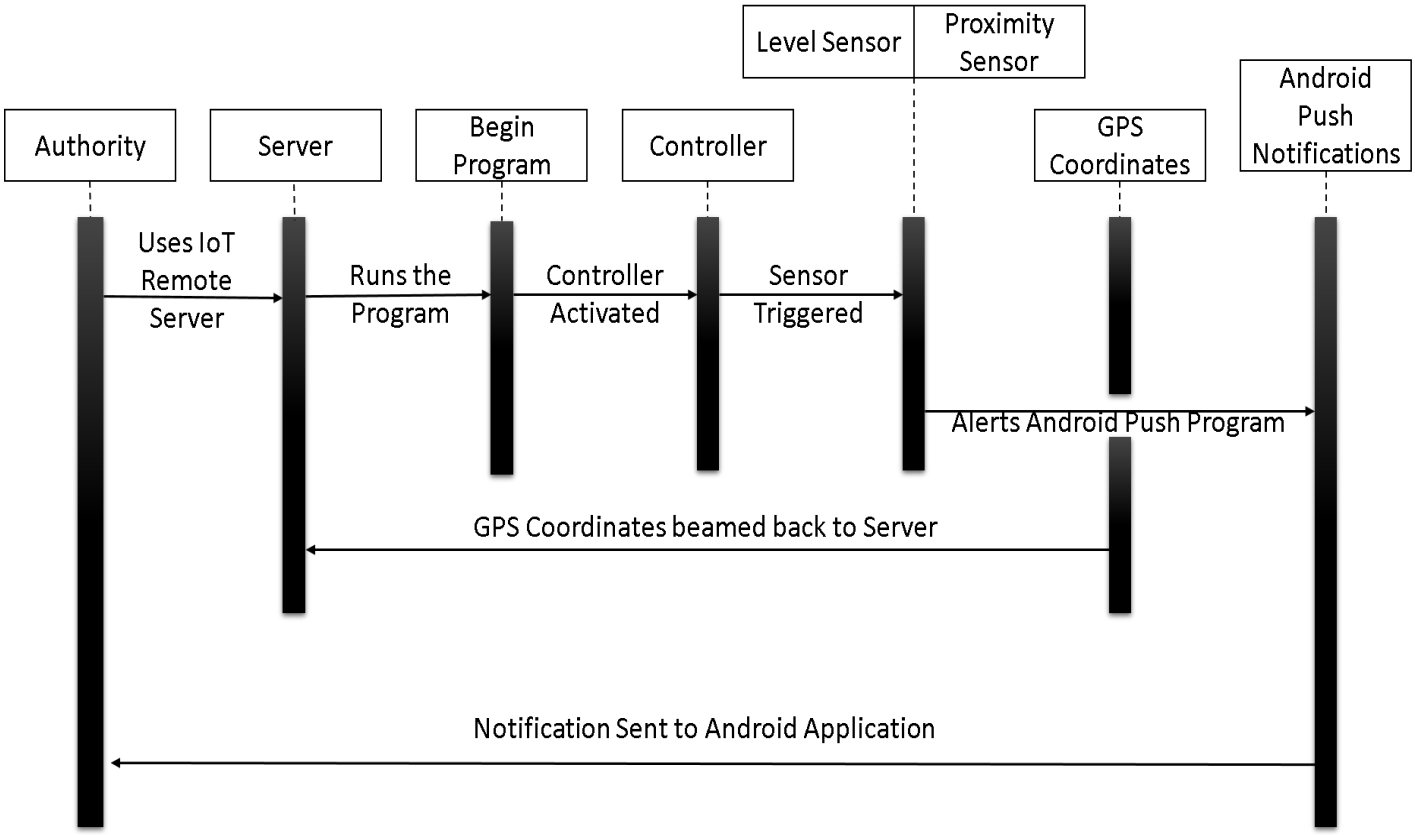

Fig 5.3 Sequence Diagram

5.5 COMPONENT DIAGRAM

Component-based development and object-oriented development go Hand-in-hand, and it is generally recognized that object technology is the Preferred foundation from which to build components.

I typically use UML 2 component diagrams as an architecture-level artefact, either to model the business software architecture, the technical software architecture, or more often than not both of these architectural aspects. Physical architecture issues, in particular hardware issues, are better addressed via UML deployment diagrams or network diagrams. In fact I'll often iterate back and forth between these diagrams.

Component diagrams are particularly useful with larger teams.

Your initial architectural modelling efforts during cycle 0 should focus on identifying the initial architectural landscape for your system. UML component diagrams are great for doing this as they enable you to model the high-level software components, and more importantly the interfaces to those components. Once the interfaces are defined, and agreed to by your team, it makes it much easier to organize the development effort between sub teams. You will discover the need to evolve the interfaces to reflect new requirements or changes to your design as your project progresses, changes that need to be negotiated between the sub teams and then implemented appropriately

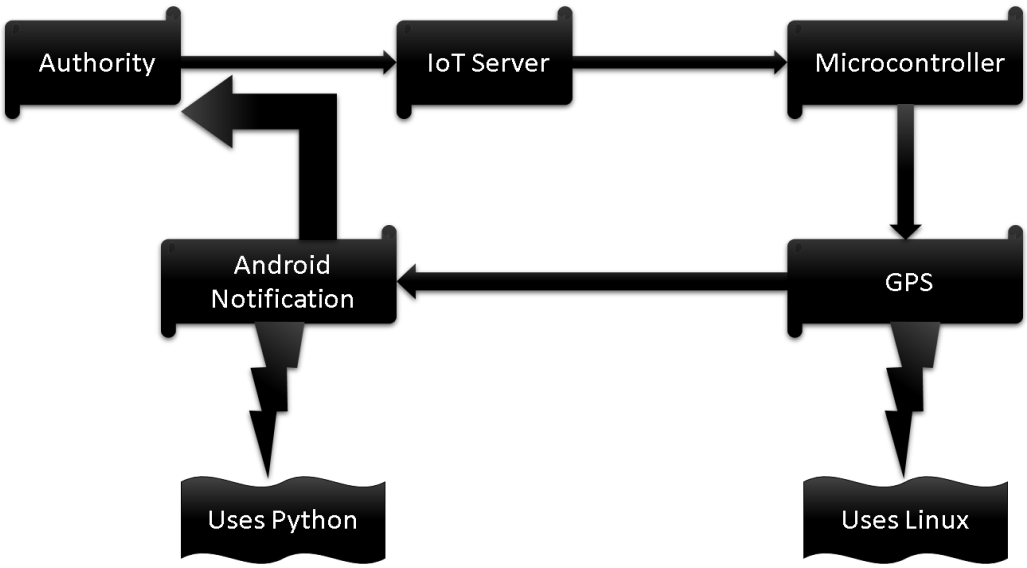

Fig 5.4 Component Diagram

CHAPTER 6

SYSTEM DESCRIPTION

6.1 GENERAL

A software application in general is implemented after navigating the complete life cycle method of a project. Various life cycle processes such as requirement analysis, design phase, verification, testing and finally followed by the implementation phase result in a successful project management. System implementation is an important stage of theoretical design is turned into practical system. Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user and so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly. The final stage is to document the entire system which provides components and the operating procedures of the system.

6.2 LIST OF MODULES

6.2.1 Sensors Interconnections With Raspberry Pi 3:

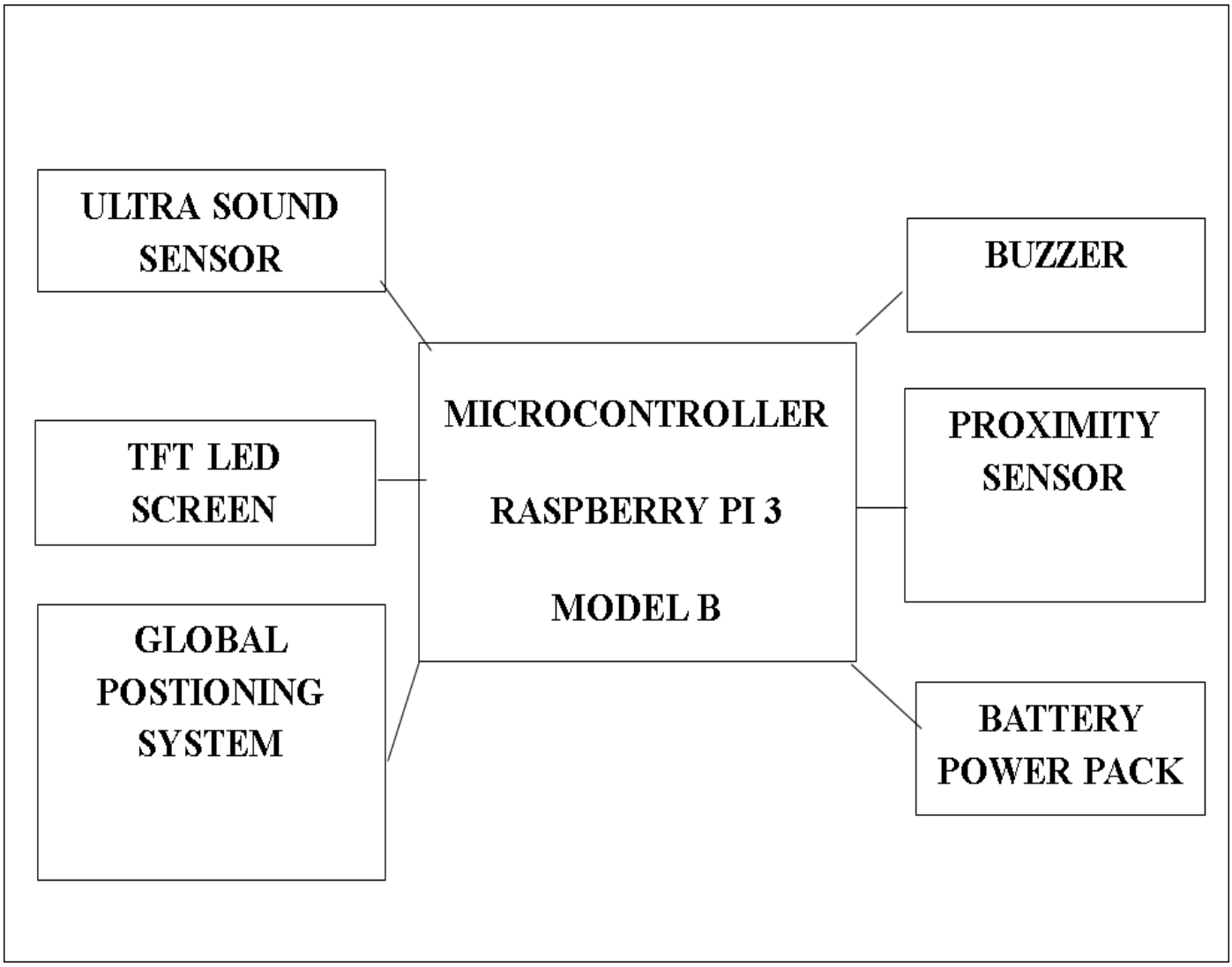

Fig 6.1 Interconnections with RPI

The above flow chart describes how our prototype is being interfaced. All the modules, sensors, input, output devices that is been used are

interconnected to our central micro-controller. With the various outputs from the modules and other sensing devices, our microcontroller is set with an in-build program which instructs the other device and the system functions as a whole simultaneously.

6.2.2 Microcontroller To App Communication:

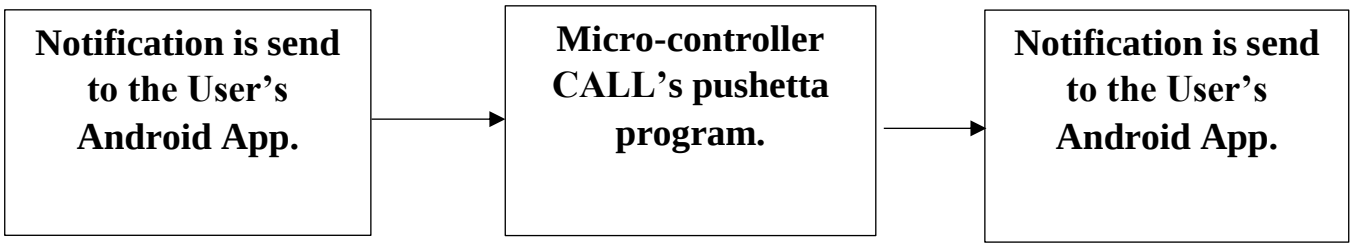

Fig 6.2 Microcontroller to App

6.3 MODULE DESCRIPTION

a. Sensors Interconnections With Raspberry Pi 3

1. Interfacing Ultra sound sensor with raspberry pi 3.
2. Interfacing Global Positioning System with raspberry pi 3.
3. Interfacing Proximity sensor with raspberry pi 3.
4. Interfacing Buzzer with raspberry pi 3.
5. Interfacing TFT LED Screen with raspberry pi 3.

b. Microcontroller To App Communication

1. Micro-controller to Android App communication Internet of Things IoT.
2. IOT ENABLED SERVER

6.4 MODULE IMPLEMENTATION

6.4.1 Interfacing Ultra Sound Sensor with Raspberry Pi 3.

Ultrasound sensor is used as a level indicating sensor. It his placed at appropriate positions inside the trash bin such that the total amount of waste load that is being stored in the bin can be monitored and other actions can be taken. The maximum range in which sound waves can be transmitted is 4m and the minimum range in which sound waves can be transmitted is 2cm.

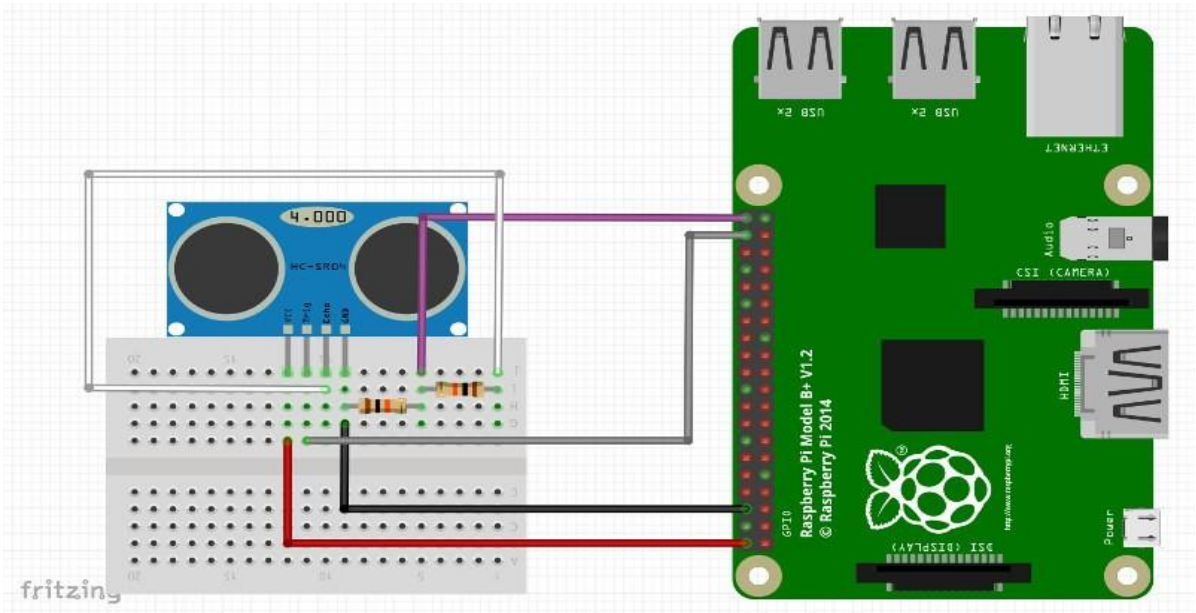

Fig 6.3 Interfacing Ultra Sound Sensor

Ultrasonic sensors emit short, high-frequency sound pulses at regular intervals. If they strike an object, then they are reflected back as echo signals to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo.

HC SR-04 is the model of ultrasound sensor is been used. It is operated with 5V DC.

It has Vcc, Grd, Echo, and Trigger pins.

c. Vcc pins of GPS is connected to any one of the GPIO pins of raspberry pi 3. In the execution terminal we can assign that GPIO pins to supply 5V or else we can directing connect to the Vcc Pin of Raspberry pi 3 which is 2 pin.
d. Grd pins of GPS is connected to any one of the GPIO pins of raspberry pi 3.

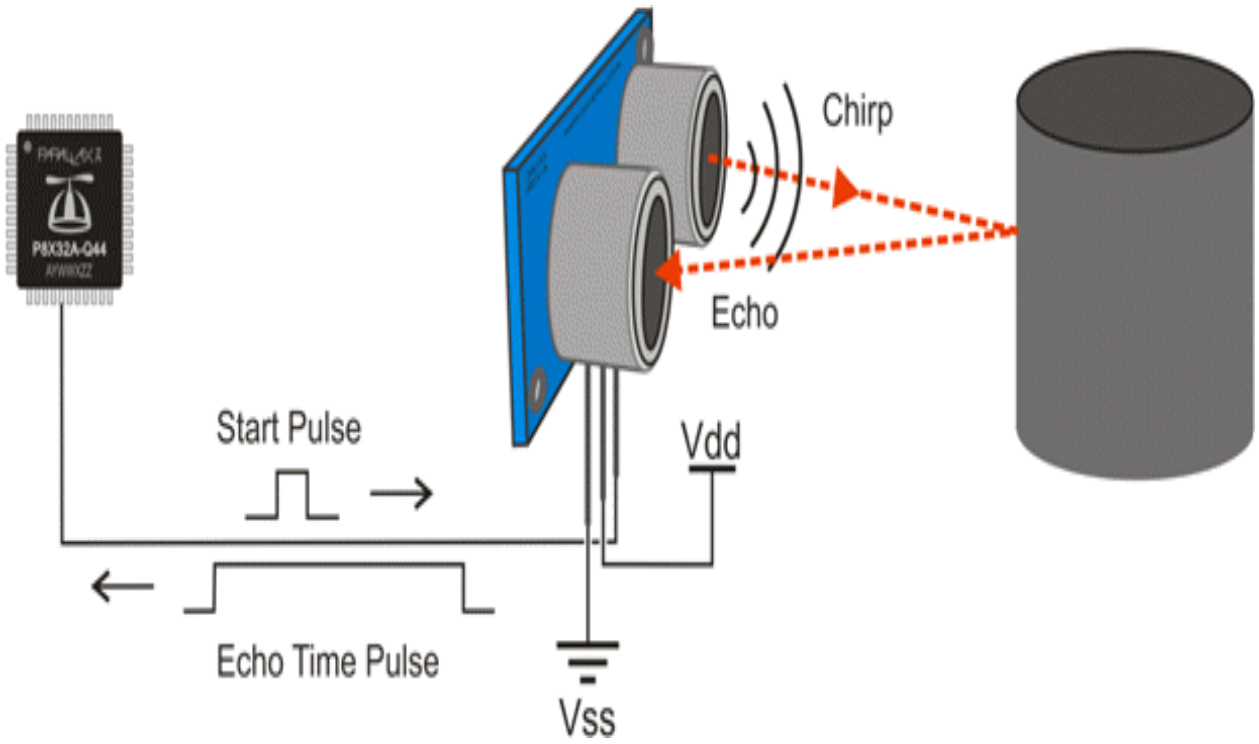

Fig 6.4 Interfacing Ultra Sound Sensor

e. In the execution terminal we can assign that GPIO pins to be ground or earthed or else we can directing connect to the Gnd Pin of Raspberry pi 3 which is 39 pin.
f. Echo and trigger is also connected to any of the GPIO pins of raspberry pi 3.

6.4.2 Interfacing Global Positioning System with Raspberry Pi 3

Global Positioning System is used for tracking the location of each bins located in various places. The Global Positioning System is connected up with satellites, ground stations, and receivers. Once the receiver calculates its distance from four or more satellites, it knows exactly where you are. It is interfaced with the central microcontroller when ultrasound level indicating sensor level reaches its marginal level it automatically enables the GPS placed inside the waste bin. GPS locks the exact position of that particular bin is been overloaded. GPS module used here is Neo 6M U-Blox. It operates at 5V DC. GPS has TX, RX, Vcc, and Grd.

g. Vcc pins of GPS is connected to any one of the GPIO pins of raspberry pi 3. In the execution terminal we can assign that GPIO pins to supply 5V or else we can directing connect to the Vcc Pin of Raspberry pi 3 which is 2 pin.
h. Grd pins of GPS is connected to any one of the GPIO pins of raspberry pi 3. In the execution terminal we can assign that GPIO pins to be ground or earthed or else we can directing connect to the Gnd Pin of Raspberry pi 3 which is 39 pin.

i. TX of GPS is connected to the RX of Raspberry pi 3 and RX of GPS is connected to the TX of Raspberry pi 3. This is done so that it can be able to send and receive data.

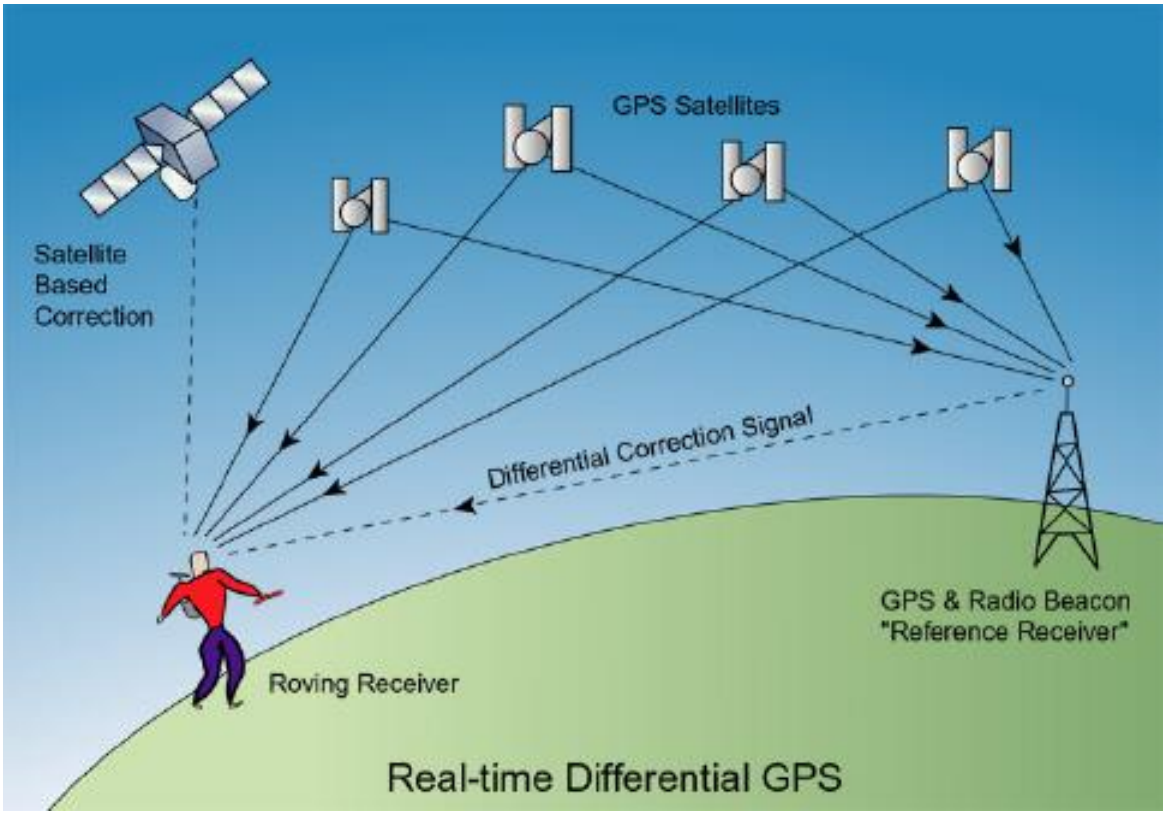

Fig 6.5: Interfacing Global Positioning System with Raspberry Pi 3

6.4.3 Interfacing Proximity Sensor With Raspberry Pi 3

The use of proximity sensor is to detect any object like waste baggage when thrown or kept just outside the waste bin. This sensor is placed at appropriate position like outer front side of the trash bin. Its maximum range of detect is approximately 2m. The detection range can be calibrated as per the distance from bin to ground with proper calculations. Proximity sensor operates at 5V DC supply

Model : Silicon IR Proximity

Operating Voltage	: 5V DC
Operating Current	: 300 mA

j. Vcc pins of GPS is connected to any one of the GPIO pins of raspberry pi 3. In the execution terminal we can assign that GPIO pins to supply 5V or else we can directing connect to the Vcc Pin of Raspberry pi 3 which is 2 pin.

k. Grd pins of GPS is connected to any one of the GPIO pins of raspberry pi 3. In the execution terminal we can assign that GPIO pins to be ground or earthed or else we can directing connect to the Gnd Pin of Raspberry pi 3 which is 39 pin.

l. TX of GPS is connected to the RX of Raspberry pi 3 and RX of GPS is connected to the TX of Raspberry pi 3. This is done so that it can be able to send and receive data.

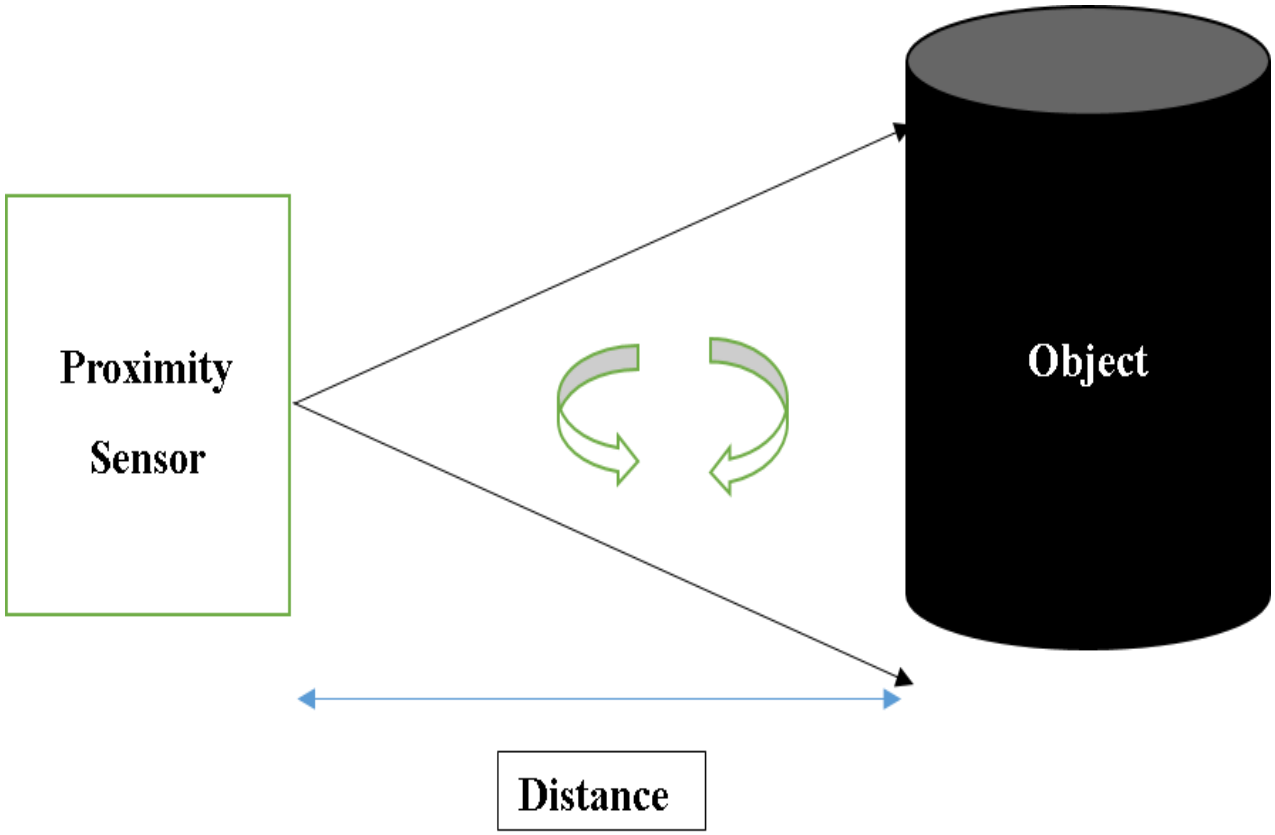

Fig 6.6 Interfacing Proximity Sensor With Raspberry Pi 3

6.4.4 Interfacing Buzzer With Raspberry Pi 3

Operating Voltage : 5V DC

The piezo buzzer produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle. These buzzers is used as an alert to the user of an event corresponding to a switching action, counter signal or sensor input. It is used as an alarming circuits. When Proximity sensor is triggered or Ultrasound Sound Sensor reaches its user defined marginal level (Varies according to the storage capacity of the bins being used) the buzzer is triggered.

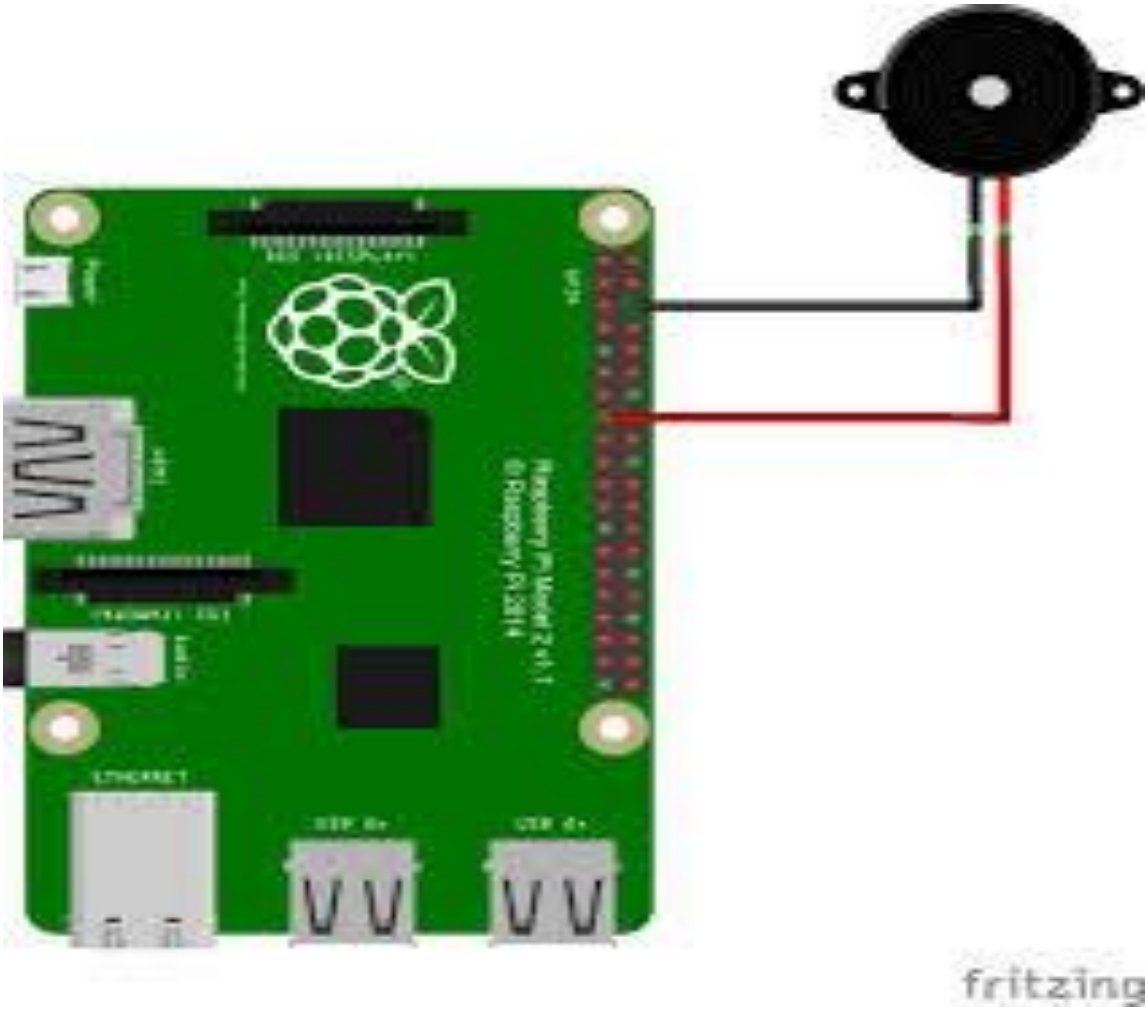

Fig 6.7 Interfacing Buzzer With Raspberry Pi 3

m. Vcc pins of GPS is connected to any one of the GPIO pins of raspberry pi 3. In the execution terminal we can assign that GPIO pins to supply 5V or else we can directing connect to the Vcc Pin of Raspberry pi 3 which is 2 pin.

n. Grd pins of GPS is connected to any one of the GPIO pins of raspberry pi 3. In the execution terminal we can assign that GPIO pins to be ground or earthed or else we can directing connect to the Gnd Pin of Raspberry pi 3 which is 39 pin.

6.4.5 Interfacing Tft Led Screen With Raspberry Pi 3

TFT LED Screen is interfaced with the controller. The messages that is needed to be notify can be displayed. It has an operating voltage of 5V. Important messages, notification, photos, can be displayed.

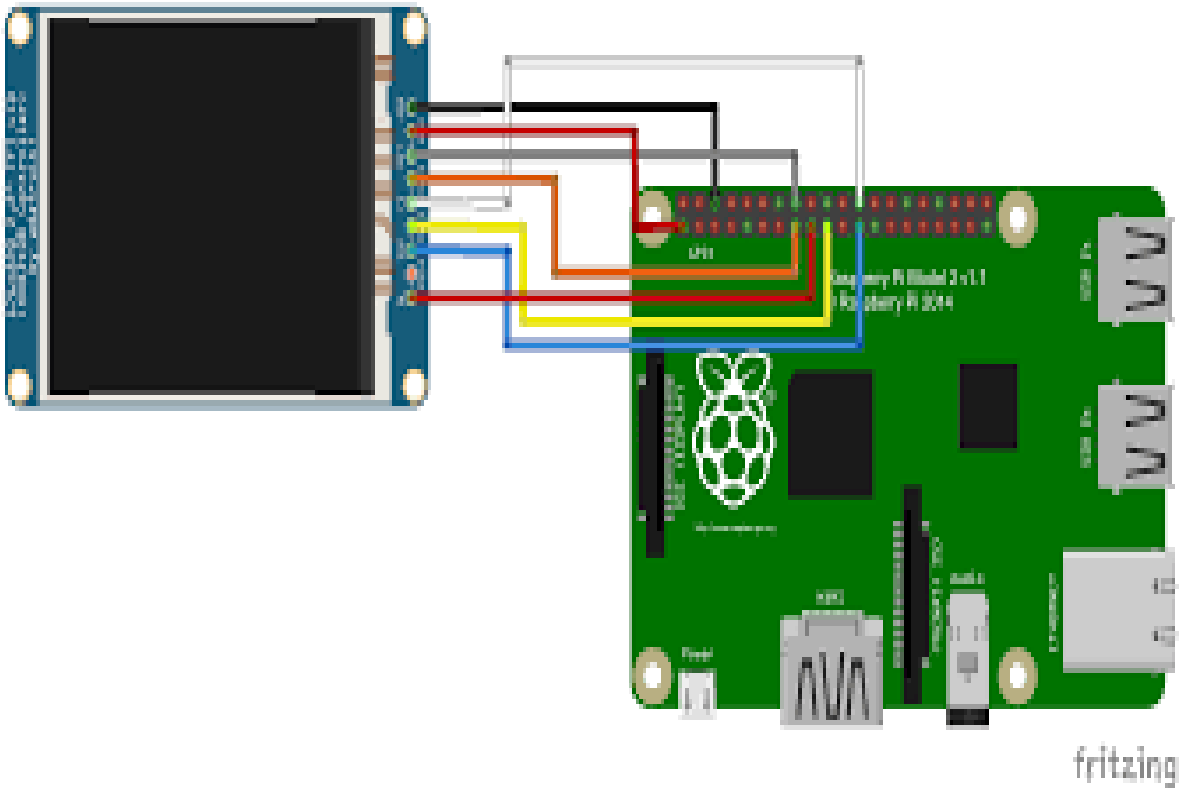

Fig 6.8 Interfacing Tft Led Screen With Raspberry Pi 3

6.5 MICROCONTROLLER TO APP COMMUNICATION

6.5.1 Micro-Controller To Android App Communication Internet Of Things Iot.

Pushetta is a system made to push real time notifications to many different devices (mobile phones, browsers, smart TV ...).
It address a simple problem: get real time information without using outdated methods like Emails.

Pushetta is made to make it simple send broadcast communications to groups of subscribers. It works in a really simple way: as publisher we can create a thematic group, every user that subscribes this group will receive a notification every time you push a message.

Steps to send notifications from our raspberry pi 3 during alert automatically to our android phone:

a. We can assign a unique channel name (the code for this is attached below) with this unique channel name we can subscribe with our mobile android phone.
b. An app named Pushetta is available in google play store.
c. Once the download is over the app, go to subscribe enter the unique channel address assigned within the program.
d. Once it is done you will get a pop up screen indicating your channel is been subscribed.
e. So during the run time of our prototype when the bin reaches its marginal value that is also set within the program a default alert message is sent to your

f. android phone. Every time an alert arises this notification is sent automatically to the user.

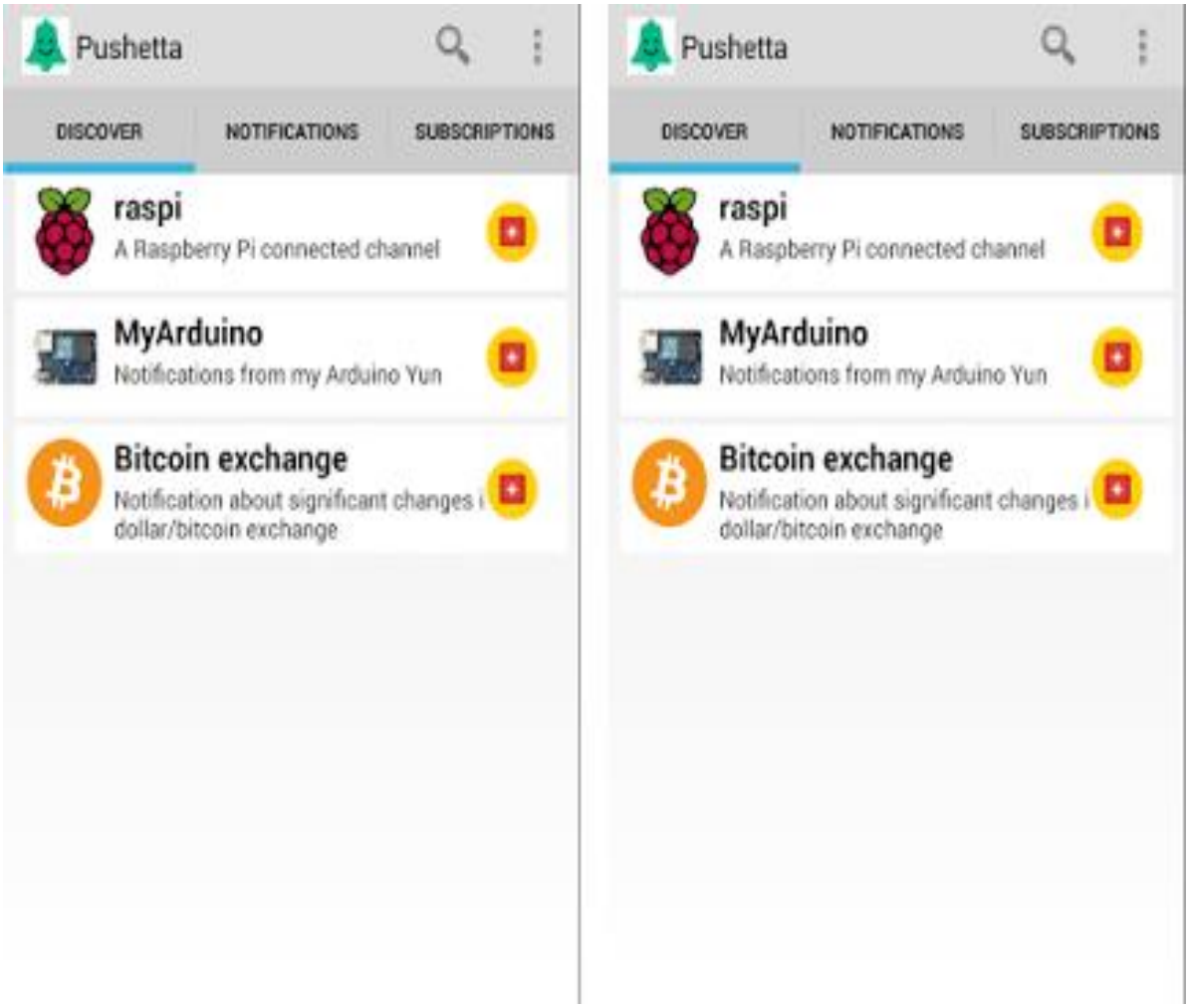

Fig 6.9 Pushetta App

6.5.2 Iot Enabled Server

A Cloud server will be maintained where all the data from each of the microcontrollers are fed wirelessly through a Wi-Fi module interfaced to the Arduino. The data from the level sensor, proximity sensor, GPS, will be uploaded and data will be throttled back in real time. Each of the devices should be constantly connected to the internet over the working duration. A simple Linux based cloud network will be used as Virtual Private Server

which will allow it to be exploited for the length of IoT applications. This will enable the devices to connect to the network and thus will allow manipulation wirelessly. Operating via TCP/IP clients, the network connection will be made easy and router port forwarding is necessary. Code complexity is greatly reduced and smaller devices with lesser power will be able to even browse effortlessly.

CHAPTER 7

SYSTEM TESTING

7.1 GENERAL

Testing is the process of detecting errors. Testing plays a critical role in assuring quality and ensuring the reliability of software. The results of testing are used later on during maintenance.

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

7.2 TESTING OBJECTIVES

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time.

a. Testing is a process of executing a program with the intent of finding an error.
b. A good test case is one that has a high probability of finding error, if it exists.
c. The tests are inadequate to detect possibly present errors.
d. The software conforms to the quality and reliable standards.

7.3 TYPES OF TESTING

System testing is stage of implementation which is aimed at ensuring that the system works accurately and efficient before live operation commences. Testing is vital the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

7.3.1 Unit Testing

In the lines of strategy, all the individual functions and modules were put to the test independently. By following this strategy all the errors in coding were identified and corrected. This method was applied in combination with the White and Black Box testing Techniques to find the errors in each module.

7.3.2 Integration Testing

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets and conducted.

The objective is to take unit tested modules and combine them test it as a whole. Thus, in the integration-testing step all the errors uncovered are corrected for the next testing steps.

7.3.3 Validation Testing

The outputs that come out of the system are as a result of the inputs that go into the system. In order to get the expected inputs that go into the system should be correct and proper. So this testing is done to check if the inputs are correct and they are validated before it goes into the system for processing.

7.3.4 Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required. This is done in regard to the following point:

1. Input screen design
2. Output screen design

An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system procedures operate to system specifications and that the integrity of important data is maintained. Performance of an acceptance test is actually the user's show. User motivation is very important for the successful performance of the system. After that a comprehensive report is prepared.

7.3.5 Performance Testing

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

a. Load testing - It is the simplest form of testing conducted to understand the behavior of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are also monitored.

b. Stress testing - It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.

c. Soak testing - Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.

d. Spike testing - Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the workload.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 GENERAL

Using this methodology by integrating IoT with waste management, the system will be greatly optimized and the system will be highly efficient. In most of the countries, waste management is a serious issue as most of the deadly diseases are caused due to poor management of waste. Countries like Austria, Netherlands, and Denmark have mastered waste management to a greater level. So all the nations in the world must be implementing better waste management systems to ensure a better and healthy scenario for its citizens.

8.2 CONCLUSION

India's present system of waste recycling and waste management is one of the world most cumbersome and low-rated method. With a mega population of more than 1.3 billion people, the waste created will be in exponentially large amounts. So when waste is being created in such large amounts, the system must be equally effective in managing it. For example Countries like Norway, Denmark, Netherlands have a system so effective that they have run out of waste, to run their recycling machineries that they have now begun to import waste in bulks from neighboring countries. India must pull up its waste management system before it goes out of control.

Our method is one of the methods of contributing to this cause. Our method will help prevent leakage, spillage and littering of wastes on the streets and public areas, and will help in collecting the waste effectively for recycle use.

8.3 FUTURE SCOPE AND DEVELOPMENTS

Internet of Things is a rapidly developing technology. It is the inter-networking of physical devices, vehicles, buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data.

The main aim of this project is to reduce human resources and efforts along with the enhancement of a smart city vision. We have often seen garbage spilling over from dustbins on to streets and this was an issue that required immediate attention. The proverb "Cleanliness is next to god and clean city is next to heaven" inspired us to conceptualized the project.

Smart dustbin helps us to reduce the pollution. Many times garbage dustbin is overflow and many animals like dog or rat enters inside or near the dustbin. This creates a bad scene. Also some birds are also trying to take out garbage from dustbin. This project can avoid such situations. And the message can be sent directly to the cleaning vehicle instead of the contractor's office.

Swatch Bharat Abhiyan (English: Clean India Mission and abbreviated as SBA or SBM for "Swatch Bharat Mission") is a national campaign by the Government of India, covering 4,041 statutory cities and towns, to clean the streets, roads and infrastructure of the country. In our system, the Smart dustbins are connected to the internet to get the real time information of the smart dustbins.

REFERENCES

[1] Urlagunta, Nagaraju. (2017). “Smart Dustbin For Economic Growth.”

[2] F. Folianto, Y. S. Low and W. L. Yeow, "Smartbin: Smart waste management system, (2015) “IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore,

[3] B. Chowdhury and M. U. Chowdhury, "RFID-based real-time smart waste management system, (2007) Australasian Telecommunication Networks and Applications Conference, Christchurch, 2007,

[4] A. S. Wijaya, Z. Zainuddin and M. Niswar, "Design a smart waste bin for smart waste management, (2017) “5th International Conference on Instrumentation, Control, and Automation (ICA), Yogyakarta, 2017,

[5] Jinila, Bevish. (2017).” Smart Waste Management.”

[6] Ravichandran, Dr. (2016). “Intelligent Dustbin to Detect and Identify Materials. 9. 191-194”.

APPENDIX I

SIMULATION CODING

Atmega320 Embedded C++

Mcufriend Library
Program For Tft Display Message From Sdcard

```
// MCUFRIEND UNO shields have microSD on pins 10, 11, 12, 13

// The official <SD.h> library only works on the hardware SPI pins

// e.g. 11, 12, 13 on a Uno  (or STM32 Nucleo)

//

// copy all your BMP files to the root directory on the microSD with your PC

// (or another directory)


#include <SPI.h>          // f.k. for Arduino-1.5.2

//#define USE_SDFAT

#include <SD.h>           // Use the official SD library on hardware pins


#include <Adafruit_GFX.h>   // Hardware-specific library

#include <MCUFRIEND_kbv.h>
```

```
MCUFRIEND_kbv tft;


#if defined(ESP32)
#define SD_CS     5
#else
#define SD_CS     10
#endif
#define NAMEMATCH ""        // "" matches any name
//#define NAMEMATCH "tiger"   // *tiger*.bmp
#define PALETTEDEPTH   0     // do not support Palette modes
//#define PALETTEDEPTH   8     // support 256-colour Palette

char namebuf[32] = "/";   //BMP files in root directory
//char namebuf[32] = "/bitmaps/";  //BMP directory e.g. files in
/bitmaps/*.bmp

File root;
int pathlen;
```

```
void setup()
{
   uint16_t ID;
   Serial.begin(9600);
   Serial.print("Show BMP files on TFT with ID:0x");
   ID = tft.readID();
   Serial.println(ID, HEX);
   if (ID == 0x0D3D3) ID = 0x9481;
   tft.begin(ID);
   tft.fillScreen(0x001F);
   tft.setTextColor(0xFFFF, 0x0000);
   bool good = SD.begin(SD_CS);
   if (!good) {
      Serial.print(F("cannot start SD"));
      while (1);
   }
   root = SD.open(namebuf);
```

```
    pathlen = strlen(namebuf);
}


void loop()
{
    char *nm = namebuf + pathlen;
    File f = root.openNextFile();
    uint8_t ret;
    uint32_t start;
    if (f != NULL) {
#ifdef USE_SDFAT
        f.getName(nm, 32 - pathlen);
#else
        strcpy(nm, (char *)f.name());
#endif
        f.close();
        strlwr(nm);
```

```
    if (strstr(nm, ".bmp") != NULL && strstr(nm, NAMEMATCH) != NULL) {
        Serial.print(namebuf);
        Serial.print(F(" - "));
        tft.fillScreen(0);
        start = millis();
        ret = showBMP(namebuf, 5, 5);
        switch (ret) {
           case 0:
              Serial.print(millis() - start);
              Serial.println(F("ms"));
              delay(5000);
              break;
           case 1:
              Serial.println(F("bad position"));
              break;
           case 2:
              Serial.println(F("bad BMP ID"));
```

```
                break;
            case 3:
                Serial.println(F("wrong number of planes"));
                break;
            case 4:
                Serial.println(F("unsupported BMP format"));
                break;
            case 5:
                Serial.println(F("unsupported palette"));
                break;
            default:
                Serial.println(F("unknown"));
                break;
          }
       }
   }
   else root.rewindDirectory();
}
```

```
#define BMPIMAGEOFFSET 54

#define BUFFPIXEL      20

uint16_t read16(File& f) {
    uint16_t result;         // read little-endian
    f.read((uint8_t*)&result, sizeof(result));
    return result;
}

uint32_t read32(File& f) {
    uint32_t result;
    f.read((uint8_t*)&result, sizeof(result));
    return result;
}

uint8_t showBMP(char *nm, int x, int y)
```

```
{
  File bmpFile;
  int bmpWidth, bmpHeight;    // W+H in pixels
  uint8_t bmpDepth;           // Bit depth (currently must be 24, 16, 8, 4, 1)
  uint32_t bmpImageoffset;    // Start of image data in file
  uint32_t rowSize;           // Not always = bmpWidth; may have padding
  uint8_t sdbuffer[3 * BUFFPIXEL];    // pixel in buffer (R+G+B per pixel)
  uint16_t lcdbuffer[(1 << PALETTEDEPTH) + BUFFPIXEL], *palette =
NULL;
  uint8_t bitmask, bitshift;
  boolean flip = true;        // BMP is stored bottom-to-top
  int w, h, row, col, lcdbufsiz = (1 << PALETTEDEPTH) + BUFFPIXEL,
buffidx;
  uint32_t pos;               // seek position
  boolean is565 = false;      //


  uint16_t bmpID;
  uint16_t n;                 // blocks read
  uint8_t ret;
```

```
if ((x >= tft.width()) || (y >= tft.height()))
   return 1;              // off screen


bmpFile = SD.open(nm);      // Parse BMP header
bmpID = read16(bmpFile);    // BMP signature
(void) read32(bmpFile);     // Read & ignore file size
(void) read32(bmpFile);     // Read & ignore creator bytes
bmpImageoffset = read32(bmpFile);       // Start of image data
(void) read32(bmpFile);     // Read & ignore DIB header size
bmpWidth = read32(bmpFile);
bmpHeight = read32(bmpFile);
n = read16(bmpFile);        // # planes -- must be '1'
bmpDepth = read16(bmpFile); // bits per pixel
pos = read32(bmpFile);      // format
if (bmpID != 0x4D42) ret = 2; // bad ID
else if (n != 1) ret = 3;   // too many planes
```

```
  else if (pos != 0 && pos != 3) ret = 4; // format: 0 = uncompressed, 3 =
565

  else if (bmpDepth < 16 && bmpDepth > PALETTEDEPTH) ret = 5; //
palette

  else {

    bool first = true;

    is565 = (pos == 3);               // ?already in 16-bit format

    // BMP rows are padded (if needed) to 4-byte boundary

    rowSize = (bmpWidth * bmpDepth / 8 + 3) & ~3;

    if (bmpHeight < 0) {              // If negative, image is in top-down order.

      bmpHeight = -bmpHeight;

      flip = false;

    }



    w = bmpWidth;

    h = bmpHeight;

    if ((x + w) >= tft.width())       // Crop area to be loaded

      w = tft.width() - x;

    if ((y + h) >= tft.height())      //
```

```
        h = tft.height() - y;


      if (bmpDepth <= PALETTEDEPTH) {   // these modes have separate
palette
        bmpFile.seek(BMPIMAGEOFFSET); //palette is always @ 54
        bitmask = 0xFF;
        if (bmpDepth < 8)
          bitmask >>= bmpDepth;
        bitshift = 8 - bmpDepth;
        n = 1 << bmpDepth;
        lcdbufsiz -= n;
        palette = lcdbuffer + lcdbufsiz;
        for (col = 0; col < n; col++) {
          pos = read32(bmpFile);    //map palette to 5-6-5
          palette[col] = ((pos & 0x0000F8) >> 3) | ((pos & 0x00FC00) >> 5)
| ((pos & 0xF80000) >> 8);
        }
      }
```

```
// Set TFT address window to clipped image bounds
tft.setAddrWindow(x, y, x + w - 1, y + h - 1);
for (row = 0; row < h; row++) { // For each scanline...
   // Seek to start of scan line.  It might seem labor-
   // intensive to be doing this on every line, but this
   // method covers a lot of gritty details like cropping
   // and scanline padding.  Also, the seek only takes
   // place if the file position actually needs to change
   // (avoids a lot of cluster math in SD library).
   uint8_t r, g, b, *sdptr;
   int lcdidx, lcdleft;
   if (flip)   // Bitmap is stored bottom-to-top order (normal BMP)
      pos = bmpImageoffset + (bmpHeight - 1 - row) * rowSize;
   else        // Bitmap is stored top-to-bottom
      pos = bmpImageoffset + row * rowSize;
   if (bmpFile.position() != pos) { // Need seek?
      bmpFile.seek(pos);
      buffidx = sizeof(sdbuffer); // Force buffer reload
```

```
        }


        for (col = 0; col < w; ) {  //pixels in row
          lcdleft = w - col;
          if (lcdleft > lcdbufsiz) lcdleft = lcdbufsiz;
          for (lcdidx = 0; lcdidx < lcdleft; lcdidx++) { // buffer at a time
            uint16_t color;
            // Time to read more pixel data?
            if (buffidx >= sizeof(sdbuffer)) { // Indeed
              bmpFile.read(sdbuffer, sizeof(sdbuffer));
              buffidx = 0; // Set index to beginning
              r = 0;
            }
            switch (bmpDepth) {          // Convert pixel from BMP to TFT
format
              case 24:
                b = sdbuffer[buffidx++];
                g = sdbuffer[buffidx++];
```

```
    r = sdbuffer[buffidx++];
    color = tft.color565(r, g, b);
    break;
case 16:
    b = sdbuffer[buffidx++];
    r = sdbuffer[buffidx++];
    if (is565)
        color = (r << 8) | (b);
    else
        color = (r << 9) | ((b & 0xE0) << 1) | (b & 0x1F);
    break;
case 1:
case 4:
case 8:
    if (r == 0)
        b = sdbuffer[buffidx++], r = 8;
    color = palette[(b >> bitshift) & bitmask];
    r -= bmpDepth;
```

```
                    b <<= bmpDepth;
                    break;
                }
                lcdbuffer[lcdidx] = color;

            }
            tft.pushColors(lcdbuffer, lcdidx, first);
            first = false;
            col += lcdidx;
          }          // end cols
        }            // end rows
        tft.setAddrWindow(0, 0, tft.width() - 1, tft.height() - 1); //restore full
screen
        ret = 0;       // good render
    }
    bmpFile.close();
    return (ret);
}
```

PROGRAM FOR HC-SR04 ON RASPBERRY PI
PYTHON 3 IDLE

```
#Libraries
import RPi.GPIO as GPIO
import time

#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BCM)

#set GPIO Pins
GPIO_TRIGGER = 18
GPIO_ECHO = 24

#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)
            time.sleep(1)

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

APPENDIX II

Sensors

HC SR-04

Fig: HC SR-04

Proximity Sensor

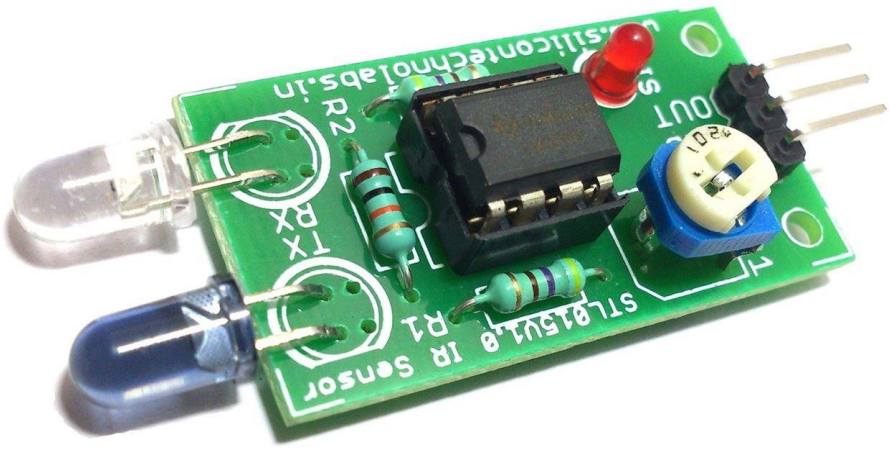

Fig: Proximity Sensor

Ublox Neo 6M GPS and Antennae

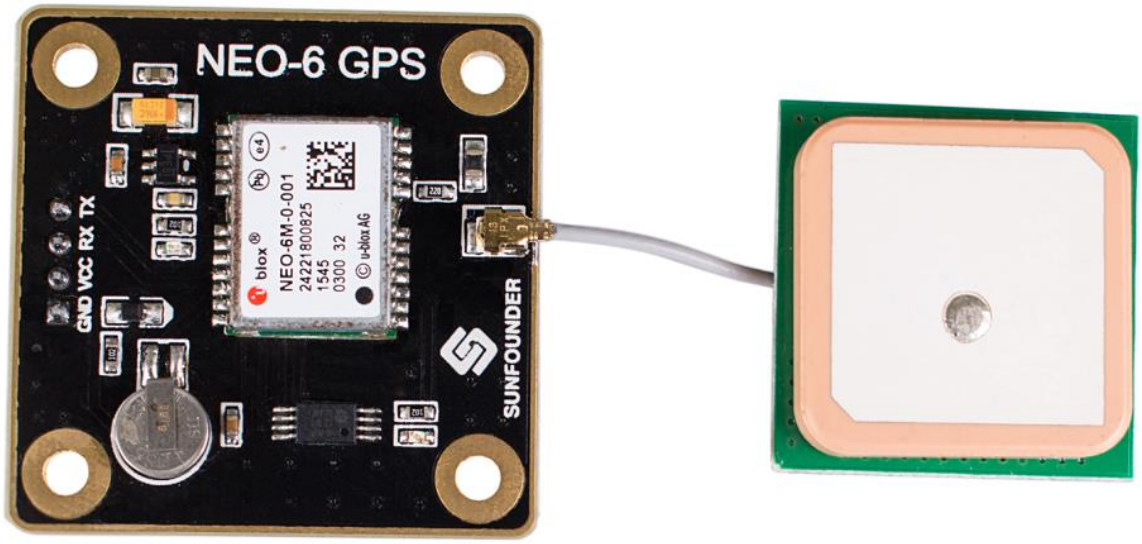

Fig: Ublox Neo 6m

Raspberry Pi 3

Fig: Raspberry Pi 3

Samsung Battery Pack

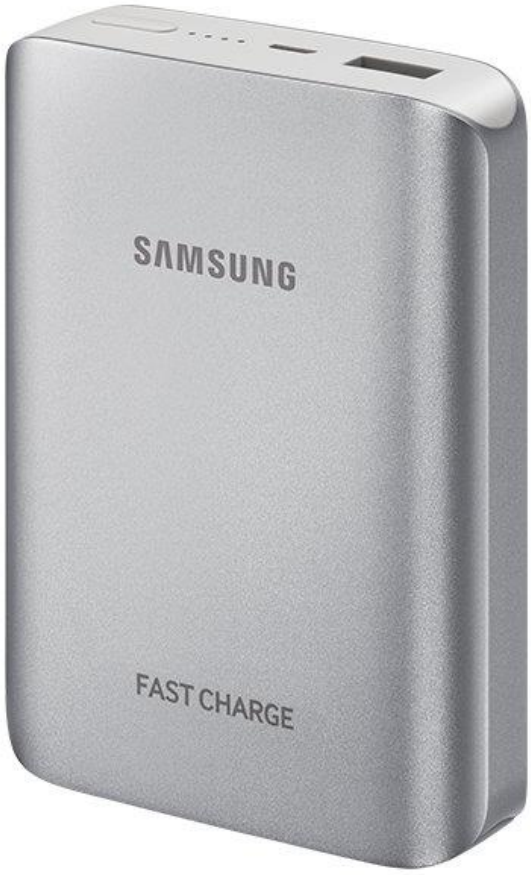

Fig: Battery Pack

Working Images

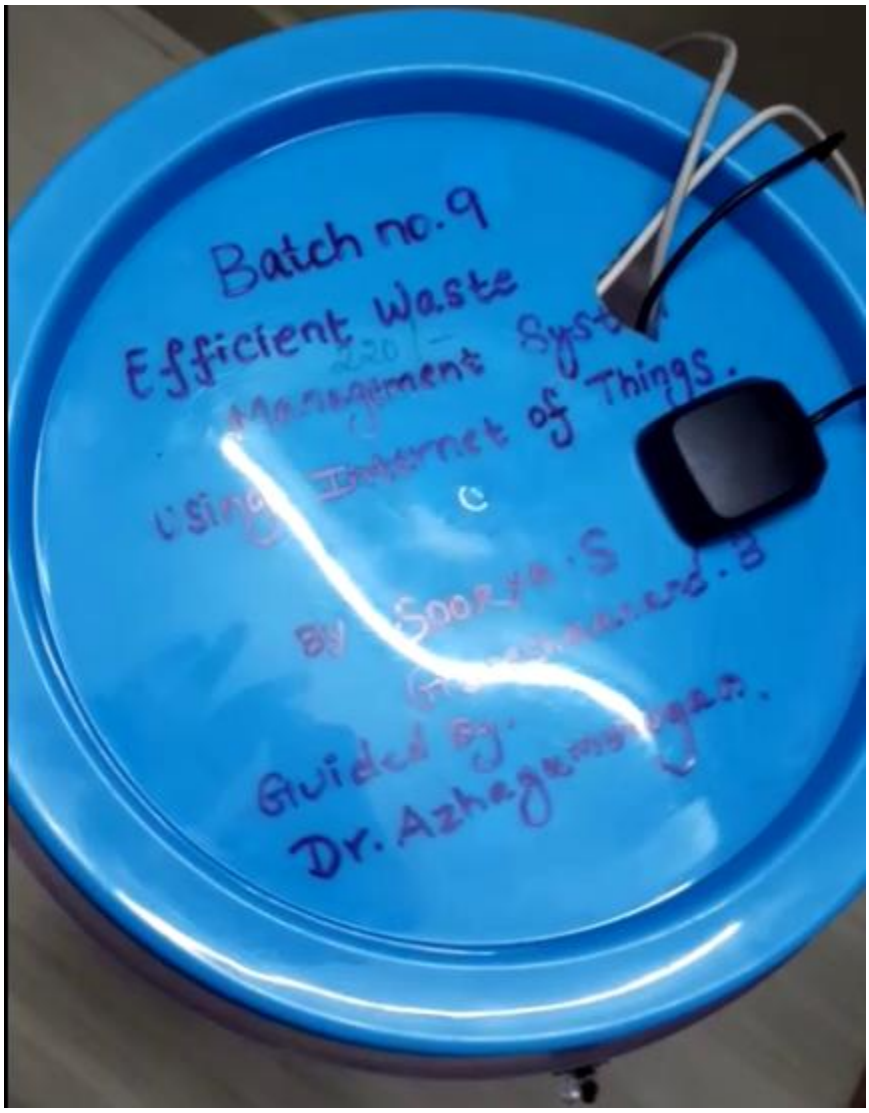

Fig: Completed Model Top View

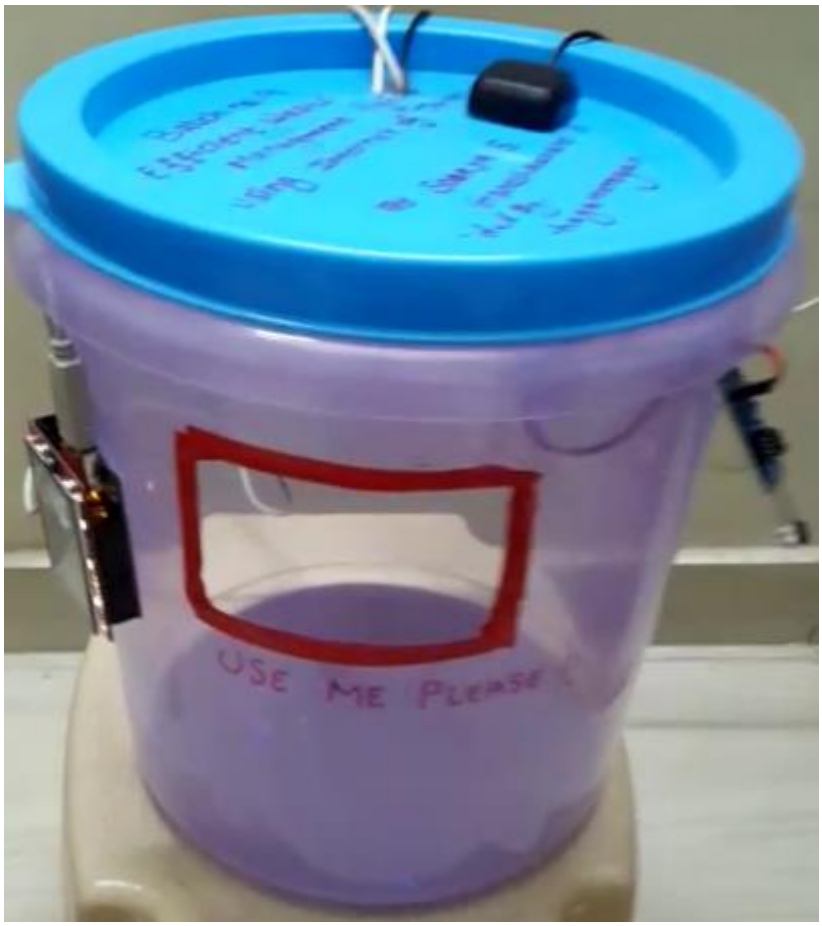

Fig: Completed Model Horizontal View

Fig: Completed Model Internal View

Fig: Screen Message

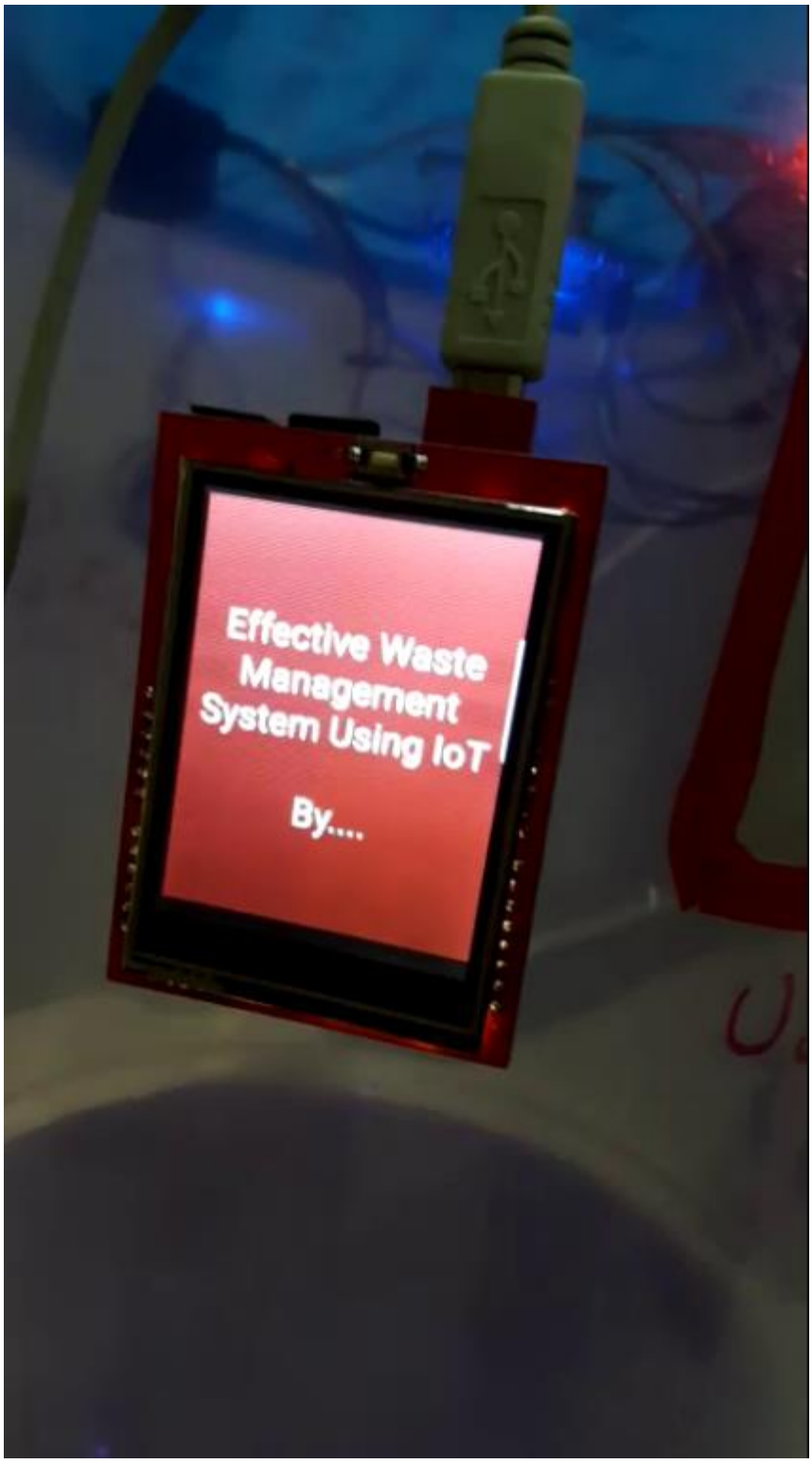

Fig: Screen Message

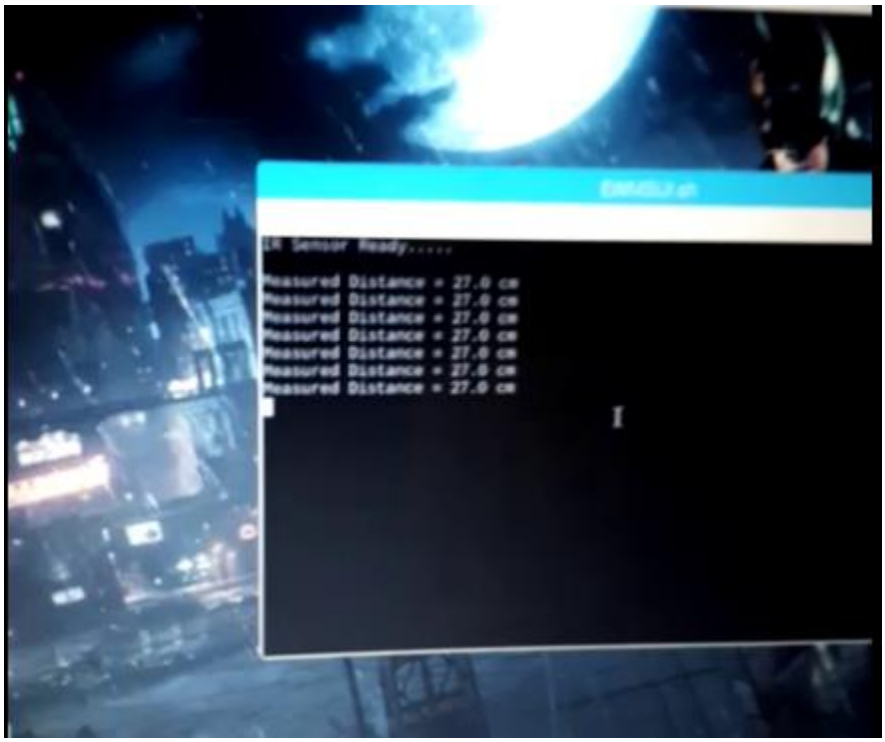

Fig: Working Terminal

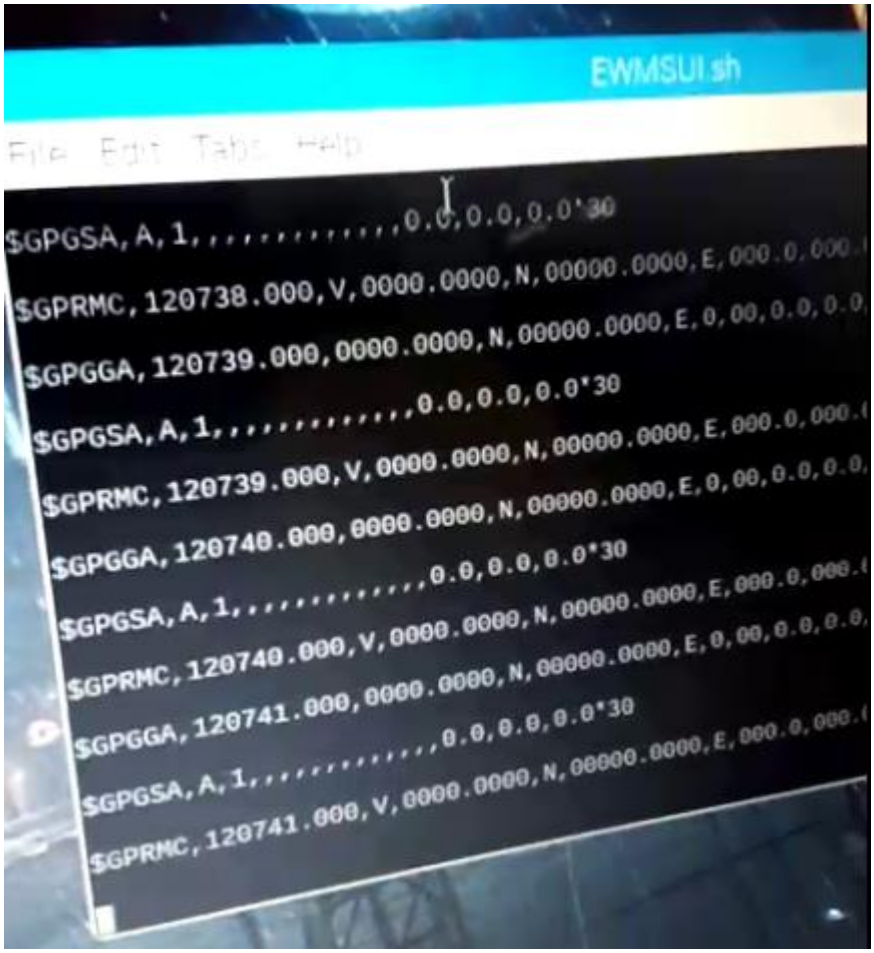

Fig: Working GPS

APPENDIX III

We have presented this project in First International Conference on POWER, ENERGY, CONTROL AND TRANSMISSION SYSTEMS-2018(22.02.18-23.02.18) held at Sri Sai Ram Engineering College, Chennai. We have attached our Certificate of Merit below.

Sri
SAI RAM ENGINEERING COLLEGE
NBA Accredited & ISO 9001 : 2008 Certified Institution
Sai Leo Nagar, West Tambaram, Chennai - 44. Tamil Nadu, India. www.sairam.edu.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

IEEE Technically sponsored

IEEE PES Power & Energy Society SSEC Student Branch Chapter

First International Conference on
POWER, ENERGY, CONTROL AND TRANSMISSION SYSTEMS - 2018
(Harnessing Power and Energy for an affordable Electrification of India)
22.02.2018 - 23.02.2018

Certificate of Merit

This is to certify that ~~Dr.~~ / ~~Mr.~~ /~~Ms.~~ SOORYA . S
of Sri Sairam Engineering College
has ~~participated~~ / presented a paper titled IoT Enabled Coherent
Debritus Management in the
International Conference on Power, Energy, Control and Transmission Systems - 2018
organized by the Department of Electrical and Electronics Engineering, Sri Sai Ram Engineering College, Tamil Nadu, India, during 22nd & 23rd February 2018.

Dr. R. Azhagumurugan, Conference Co-chair
Prof. AL. Kumarappan, Conference Chair
Dr. C.V. Jayakumar, Principal
Sai Prakash LeoMuthu, Chief Executive Officer

Technical Sponsors
IEEE | IEEE young professionals | IEEE MADRAS SECTION | IEEE young professionals Power & Energy Society
Sairam www.sairamgroup.in

Sri

SAI RAM ENGINEERING COLLEGE

NBA Accredited & ISO 9001 : 2008 Certified Institution

Sai Leo Nagar, West Tambaram, Chennai - 44. Tamil Nadu, India. **www.sairam.edu.in**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

IEEE Technically sponsored

IEEE PES Power & Energy Society SSEC Student Branch Chapter

First International Conference on
POWER, ENERGY, CONTROL AND TRANSMISSION SYSTEMS - 2018

(Harnessing Power and Energy for an affordable Electrification of India)

22.02.2018 - 23.02.2018

Certificate of Merit

This is to certify that ~~Dr.~~ / Mr. /~~Ms.~~ GANESH AANAND. B

of Sri Sairam Engineering college

has ~~participated~~ / **presented** a paper titled IoT enabled Coherent Detritus Management in the

International Conference on Power, Energy, Control and Transmission Systems - 2018 organized by the Department of Electrical and Electronics Engineering, Sri Sai Ram Engineering College, Tamil Nadu, India, during 22nd & 23rd February 2018.

Dr. R. Azhagumurugan	Prof. AL. Kumarappan	Dr. C.V. Jayakumar	Sai Prakash LeoMuthu
Conference Co-chair	Conference Chair	Principal	Chief Executive Officer

Technical Sponsors

IEEE | IEEE young professionals | IEEE MADRAS SECTION | IEEE young professionals Power & Energy Society

Sairam INSTITUTIONS www.sairamgroup.in