

SMART GARBAGE SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in

Information Technology

By

UPENDHER RAO POLISHETTY(37120054)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND

TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH 2021



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **UPENDHER RAO POLISHETTY(37120054)**, who carried out the project entitled “**SMART GRABAGE SYSTEM**” under our supervision from DEC 2020 to JAN 2020.

Internal Guide

Dr. A.SIVASANGARI, M.E,Ph.D.,

Head of the Department

Dr.R SUBHASHINI M.E,Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I UPENDHER RAO POLISHETTY(37120054), here by declare that the Project Report entitled “**SMART GRABAGE SYSTEM**” done by me under the guidance of **Dr. A. SIVASANGARI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology .

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** or their encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T SASIKALA M.E Ph .D**, Dean, School Of Computing and **Dr .R SUBHASHINI M.E Ph. D** ,Head of the Department, Dept .of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. SIVASANGARI** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Information Technology who were helpful in many ways for the completion of the project.

ABSTRACT

In uncertain times like natural disaster satellite imagery plays a key role in navigating the places which affected the most and also to monitor the fore coming situation, to get better insights from the images, it need be convincible that the quality of the image meets the quality that is required. Presently, we have wide range of approaches to restore the good quality image from distorted image using different Deep learning techniques. To solve the problem with degraded images, we have different methods like Full resolution convolutional neural network, Deep resolution convolution neural network and Very deep resolution neural network and PSNR (Peak signal to noise ratio). Among all PSNR provide better colors. But PSNR absences in providing better quality of textures. To overcome this problem, we have proposed a model to restore the quality of an image using SRCNN (Super Resolution Convolution Neural Network). Super resolution is one of the image restoration technique where we will split each layer and then will perform end-to-end mapping in each layer to obtain a high-quality image from a diminished one. With this proposed system we will restore the better texture for an image with input as degraded image and yields the similar scope image as output with High tenacity.

CONTENTS

ACKNOWLEDGEMENT	iv
ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER No.	TITLE	Page no.
1	INTRODUCTION	1
	1.1 LITERATURE REVIEW	1
	1.2 DISADVANTAGES OF EXISTING SYSTEM	2
2	INTRODUCTION OF ARDUINO UNO	3
	2.1 ARDUINO UNO	3
	2.1.1 HARDWARE	4
	2.1.2 SOFTWARE DEVELOPMENT	5
	2.1.3 APPLICATIONS	6
	2.1.4 TECHNICAL SPECIFICATIONS	6
	2.1.5 PROGRAMMING	7
	2.1.6 POWER	7
	2.1.7 MEMORY	8
	2.1.8 INPUT AND OUTPUT	8
	2.1.9 COMMUNICATION	9
	2.1.10 AUTOMATIC(SIFTWARE) RESET	10
3	FUNCTIONAL MODULES AND SOFTWARE MODULES DESCRIPTION	11
	3.1 ULTRASONIC SENSOR	11
	3.2 BUZZER	12
	3.3 POWER SUPPLY	12
	3.4 GSM	13
	3.4.1 ARCHITECTURE OF GSM NETWORK	14
	3.5 LCD	15
	3.6 SOFTWARE MODULE DESCRIPTION	16
	3.6.1 ARDUINO INSTALLATION	16
4	PROJECT IMPLEMENATATION	20
	4.1 BLOCK DIAGRAM	20

	4.2 DESIGN AND IMPLEMENTATION	20
	4.3 SNAPSHOT OF THE PROJECT	23
	4.4 RESULT	24
	4.5 ADVANTAGES	24
5	CONCLUSION	25
	REFERENCES	26

LIST OF FIGURES

FIGURE NO.	TITLE OF THE FIGURE	PAGE NO.
2.1	Arduino UNO	4
3.1	Ultrasonic sensor	12
3.2	Buzzer	13
3.3	Components of power supply	13
3.4	GSM Module	14
3.5	General Architecture of a GSM network	16
3.6	LCD display	17
3.7	Downloading the Arduino UNO software	18
3.8	Unzipping the downloaded file	18
3.9	Opening on existing project example	19
3.10	Creating new project	20
4.1	Block diagram of the project	21
4.3	Experimental implementation of the project	23

CHAPTER 1

INTRODUCTION

This project GSM based Garbage Monitoring system is 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 SMS. 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 arduino uno board, LCD screen, GSM modem for sending data. The system is powered by a 12V transformer. The LCD screen is used to display the status of the level of garbage collected in the bins. Whereas GSM is built to show the status to the user, monitoring it with SMS. The SMS consists of text related to all garbage bins. The LCD screen shows the status of the garbage level. The system puts on LCD screen continuously monitoring of garbage with arduino board. Thus this system helps to keep the city clean by informing about the garbage levels of the bins by providing SMS to the respective person.

1.1 LITERATURE REVIEW:

The Garbage monitoring has to be effectively and efficiently implemented the variety of proposal we put and some of them already implemented, But it cannot be effect on the system performance so we done a project among this problem in a different method in low cost using embedded module GSM. The smart garbage monitoring system in cities using GSM proposed a method. which monitors the garbage bins and informs about the level of garbage collected in the garbage bins via a web page, in which system uses ultrasonic sensors placed over the bins to detect the garbage level and compare it with the garbage bins depth. The proposed system uses Arduino family microcontroller, LCD screen, Wi-Fi modem for sending data and a buzzer, GSM, Ultrasonic Sensor.

This proposed system assures the cleaning of dust bin soon when the

garbage level reaches its maximum threshold. The embedded C technique is enhancing the system to achieve the desired result. Thus this system helps to keep the city clean by informing about the garbage levels of the bins by providing SMS to the respective person.

1.2 DISADVANTAGES OF EXISTING SYSTEM

1. Time consuming and less effective.
2. High costs.
3. Unhygienic environment and look of the city.
4. Bad smell spreads and may cause illness to human beings.
5. More traffic and noise.

CHAPTER 2

INTRODUCTION OF ARDUINO UNO

2.1 ARDUINO UNO

Arduino is a computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (*shields*) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

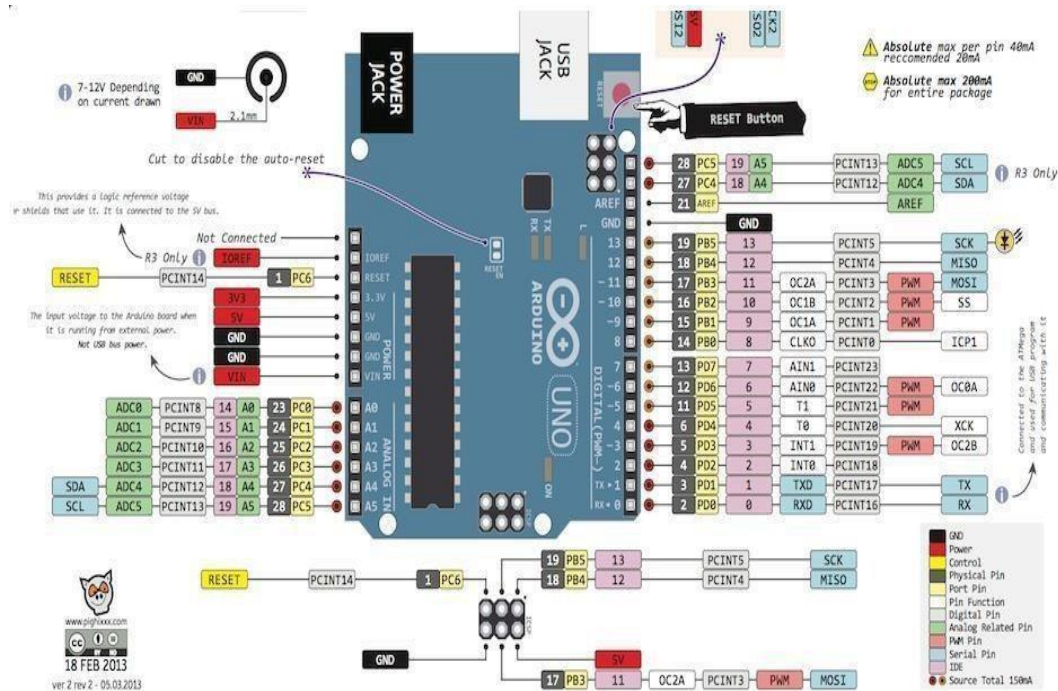


Fig 2.1: Arduino UNO

2.1.1 HARDWARE:

Arduino is open-source hardware. The source code for the IDE is released under the GNU General Public License, version 2. Nevertheless, an official Bill of Materials of Arduino boards has never been released by Arduino staff.

An Arduino board consists of an Atmel 8-, 16- or 32-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560), but other makers' microcontrollers have been used since 2015. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lilypad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a bootloader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the optiboot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor-transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using to USB serial adapter chip such as FTDI FT232

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Arduino Uno provides 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots.

2.1.2 SOFTWARE DEVELOPMENT:

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages *Processing* and *Wiring*. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a *sketch*. Sketches are saved on the development computer as text files with the file extension *.ino*. Arduino Software (IDE) pre-1.0 saved sketches with the extension *.pde*.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an

executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

2.1.3 APPLICATIONS:

- Ardupilot, drone software and hardware
- Gameduino, an Arduino shield to create retro 2D video games
- ArduinoPhone, a do-it-yourself cellphone
- Water quality testing platform
- Automatic titration system based on Arduino and stepper motor
- DC motor control using Arduino and H-Bridge
- Advanced RISC architecture

2.1.4 TECHNICAL SPECIFICATIONS:

MICROCONTROLLER	ATmega328P
OPERATING VOLTAGE	5V
INPUT VOLTAGE(RECOMMENDED)	7-12 V
INPUT VOLTAGE (LIMITS)	6-20V
DIGITAL I/O PINS	14(OF WHICH 6 PROVIDE PWM OUTPUT)
PWM DIGITAL I/O PINS	6
ANALOG INPUT PINS	6
DC CURRENT PER I/O PIN	20 MA
DC CURRENT FOR 3.3V PIN	50 MA
FLASH MEMORY	32 KB(ATmega328p) of which 0.5KB is used for bootloader
SRAM	2 KB (ATmega328p)
EEOROM	1KB (ATmega328p)
CLOCK SPEED	16 MHz

2.1.5 PROGRAMMING:

The Arduino Uno can be programmed with the (Arduino Software (IDE)). Select "Arduino Uno" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino/Genuino Uno comes pre-programmed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resealing the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The Arduino/Genuino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

2.1.6 POWER:

The Arduino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter or battery.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board.

The power pins are as follows:

- **Vin:** The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.
- **IOREF:** This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

2.1.7 MEMORY:

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

2.1.8 INPUT AND OUTPUT:

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **LED:** 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **TWI:** A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

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

function. There are a couple of other pins on the board:

- **AREF:** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

2.1.9 COMMUNICATION:

Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is

required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

2.1.10 AUTOMATIC (SOFTWARE) RESET:

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the Amega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 Nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labelled "RESET-EN".

CHAPTER 3

FUNCTIONAL MODULES & SOFTWARE MODULES DESCRIPTION

3.1 ULTRASONIC SENSOR:

Ultrasonic transducers and ultrasonic sensors are devices that generate or sense ultrasound energy. They can be divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

The sensor has two main parts:

A transducer that creates an ultrasonic sound and another listens to its echo. To use this sensor to measure distance, the robot's brain must measure the amount of time it takes for the ultrasonic sound to travel.

Sound travels at approximately 340 meters per second. This corresponds to about 29.412us (microseconds) per centimeter. To measure the distance the sound has travelled we use the formula:

$$\text{Distance} = (\text{Time} \times \text{Speed of Sound}) / 2.$$

The "2" is in the formula because the sound has to travel back and forth. First the sound travels away from the sensor, then it bounces off of a surface and returns back.

The easy way to read the distance as centimeters is use the formula:

$$\text{Centimeters} = ((\text{Microseconds} / 2) / 29).$$

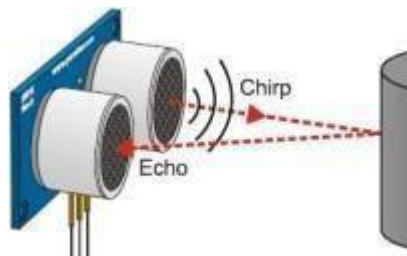


Fig 3.1: Ultrasonic sensor

3.2 BUZZER:

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

- This buzzer is a loudspeaker that produces the sound when electricity is passed through it.



Fig3.2: Buzzer

3.3 Power Supply:

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage. The block diagram of regulated power supply is shown in the figure 3.2

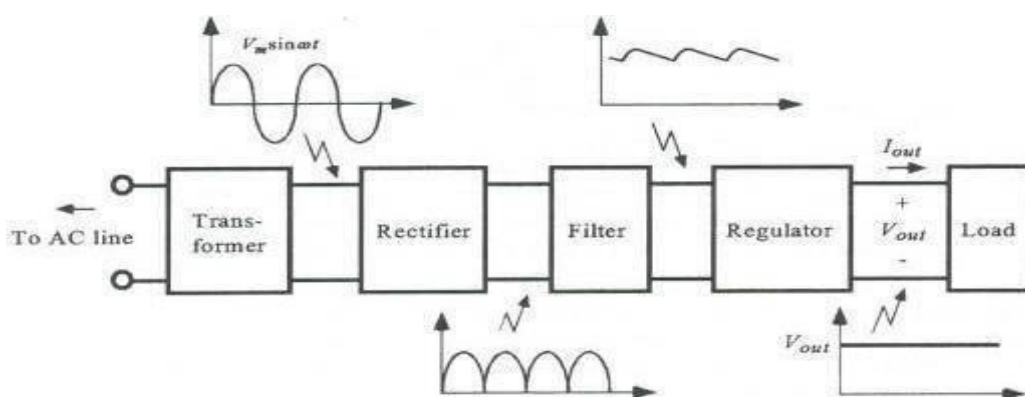


Fig3.3: Components of power supply

3.4 GSM (Global System for Mobile communication):



Fig 3.4: GSM Module

The Global System for Mobile communication, usually called GSM, Telecommunications Standards Institute (ETSI) to describe protocols for second generation (2G) digital cellular networks used by mobile phones. The GSM standard was developed as a replacement for first generation (1G) analog cellular networks, and originally described a digital, circuit switched network optimized for full duplex voice telephony. This was expanded over time to include data communications, first by circuit switched transport, then packet data transport via GPRS (General Packet Radio Services) and EDGE (Enhanced Data rates for GSM Evolution or EGPRS). Further improvements were made when the 3GPP developed third generation (3G) UMTS standards followed by fourth generation (4G) LTE Advanced standards. "GSM" is a trademark owned by the GSM Association.

GSM is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity.

The ubiquity of the GSM standard makes international roaming very common between mobile phone operators, enabling subscribers to use their phones in many parts of the world.

GSM differs significantly from its predecessors in that both signaling and speech channels are Digital call quality, which means that it is considered a second generation (2G) mobile phone system. This fact has also meant that data communication was built into the system from the 3rd Generation Partnership Project (3GPP).

GSM is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of time division multiple access (Time Division Multiple Access) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1800 MHz frequency band.

GSM together with other technologies is part of an evolution of wireless mobile telecommunication that includes High-Speed Circuit-Switched Data (High-Speed Circuit-Switched Data), General Packet Radio System (General Packet Radio Services), Enhanced Data GSM Environment (Enhanced Data GSM Environment), and Universal Mobile Telecommunications Service (Universal Mobile Telecommunications System).

3.4.1 ARCHITECTURE OF GSM NETWORK:

A GSM network is composed of several functional entities, whose functions and interfaces are specified. Figure 1 shows the layout of a generic GSM network. The GSM network can be divided into three broad parts. The Mobile Station is carried by the subscriber. The Base Station Subsystem controls the radio link with the Mobile Station. The Network Subsystem, the main part of which is the Mobile services Switching Center (MSC), performs the switching of calls between the mobile users, and between mobile and fixed network users. The MSC also handles the mobility management operations. The Mobile Station and the Base Station Subsystem communicate across the Um interface, also known as the air interface or radio link. The Base Station Subsystem communicates with the Mobile services Switching Center across the A interface.

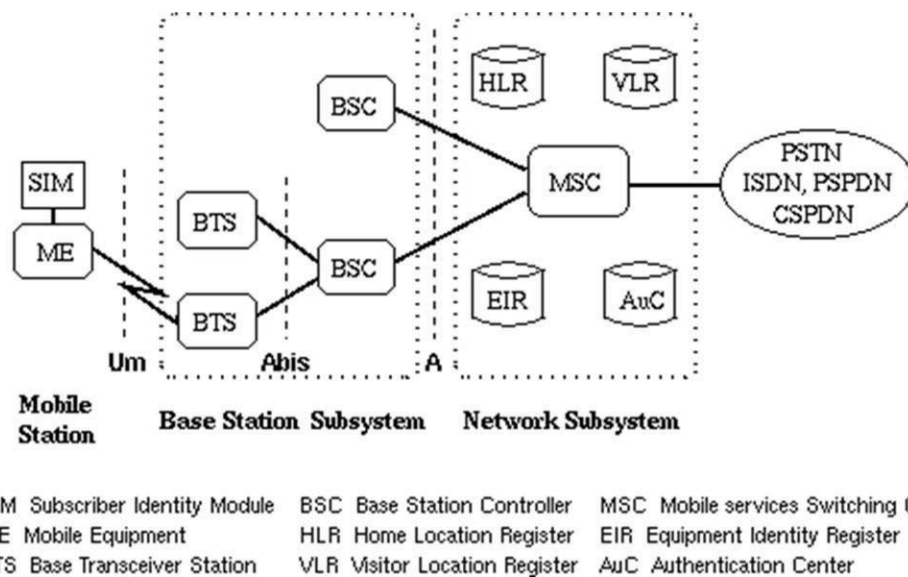


Fig 3.5: General Architecture of a GSM network

3.5 LCD:

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

1. The declining prices of LCDs.
2. The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.
3. Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.
4. Ease of programming for characters and graphics.

These components are “specialized” for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.



Fig 3.6: LCD display

A model as shown in figure 6.3 here is for its low price and great possibilities most frequently used in practice. It is based on the HD44780 microcontroller (*Hitachi*) and can display messages in two lines with 16 characters each. It displays all the alphabets, Greek letters, punctuation marks, mathematical symbols etc. In addition, it is possible to display symbols that user makes up on its own. Automatic shifting message on display (shift left and right), appearance of the pointer, backlight etc. are considered as useful characteristics.

3.6 SOFTWARE MODULES DESCRIPTION:

3.6.1 ARDUNIO INSTALLATION:

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board. In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1: First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, ArduinoDuemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible With your operating system (Windows, IOS, or Linux).After your file download is complete, unzip the file.

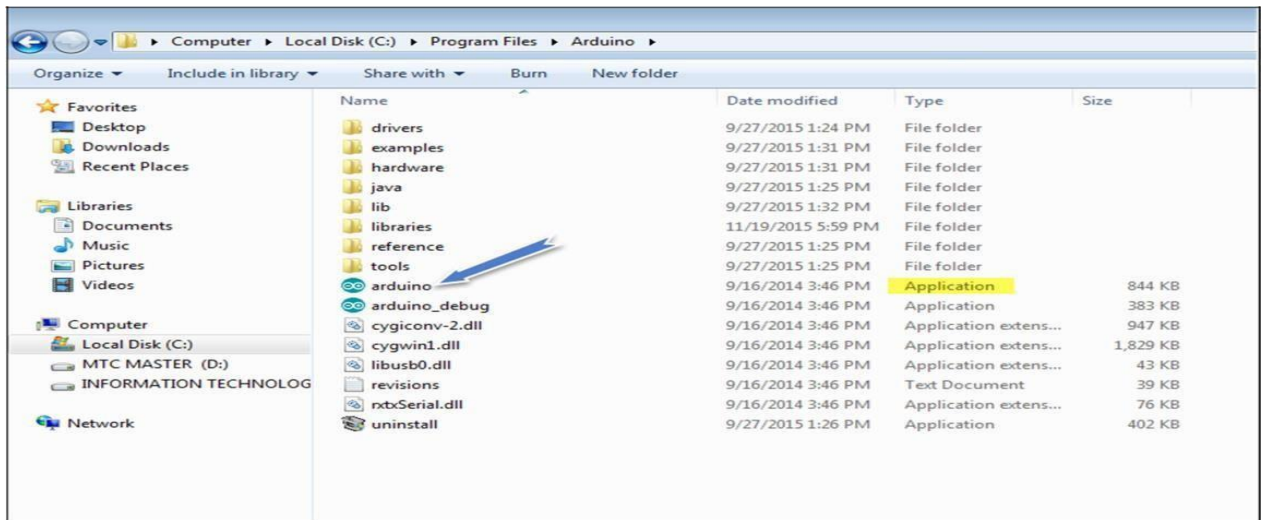


Fig 3.7: Downloading the Arduino UNO software



Fig 3.8: Unzipping the downloaded file

Step 3: Power up your board. The Arduino Uno, Mega, Duemilanove and ArduinoNano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an ArduinoDiecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4: Launch Arduino IDE.After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5: Open your first project.

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example.
- To create a new project, select File □ New
- To open an existing project example, select File -> Example -> Basics -> Blink.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

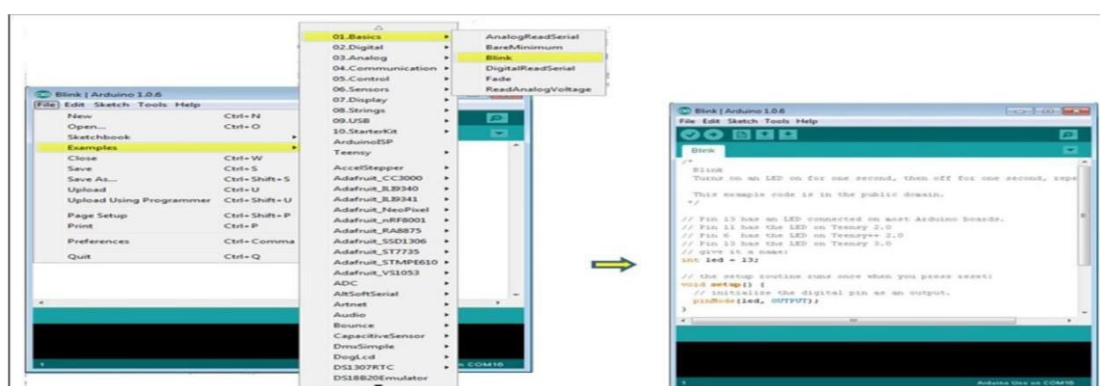


Fig3.9: Opening on existing project example

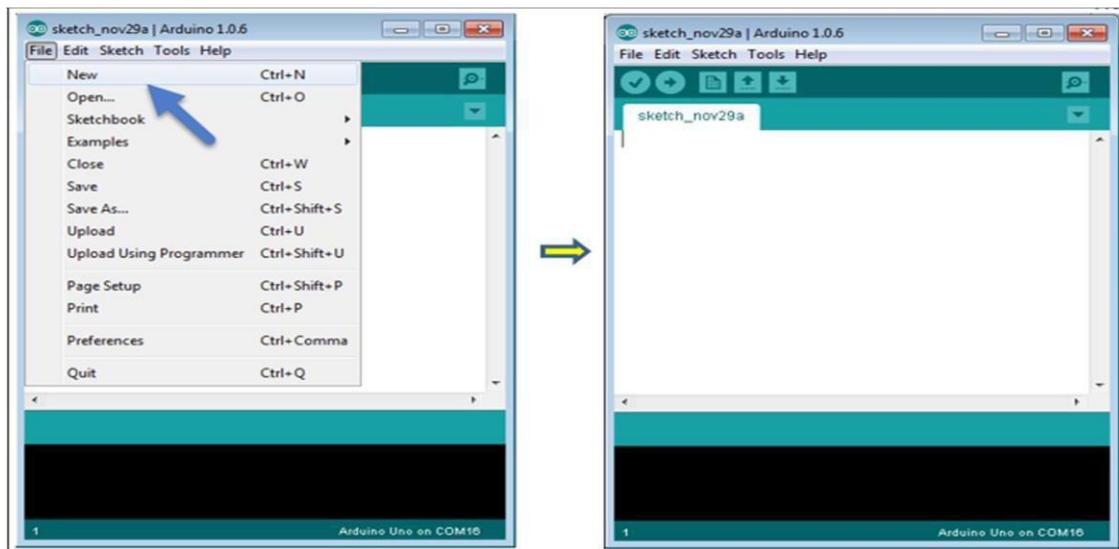


Fig3.10: Creating new project

Step 6: Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools -> Board and select your board

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 BLOCK DIAGRAM:

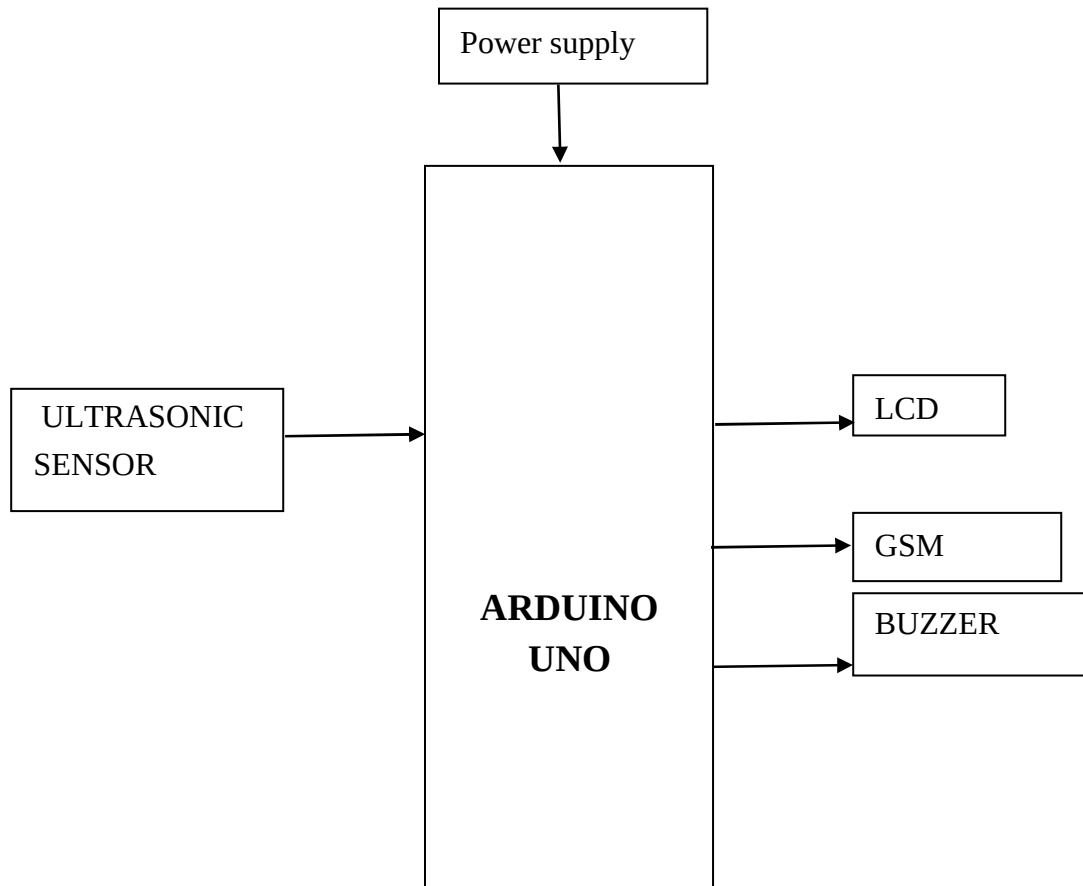


Fig4.1: Block diagram of the project

4.2 DESIGN AND IMPLEMENTATION:

Here arduino board, GSM modem operates with 12V DC, LCD display, sensor circuit operates with DC 5V supply and this supply is provided by regulator of LM7805. 12V step down transformer with rectifier and filter is used to give power supply (or adapter can be use depends upon our requirement). Every Arduino board needs a way to be connected to a power source.

The Arduino UNO can be powered from USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled (1) and the barrel jack is labeled (2). The USB connection is also load code onto your Arduino board PINS (5V, 3.3V, GND, ANALOG, DIGITAL, PWM, AREF) The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a bread board / PCBs and some wire. They usually have black plastic „headers“ that allow you to just plug a wire right into the board.

The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions. GND (3): Short for „Ground“. There are several GND pins on the Arduino, any of which can be used to ground your circuit. 5V (4) & 3.3V (5): As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts. Analog (6): The area of pins under the „Analog In“ label (A0 through A5 on the UNO) is Analog In pins. These pins can read the signal from an analog sensor (like a light sensor) and convert it into a digital value that we can read. Digital (7): Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED). PWM (8): You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO).

These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). AREF (9): Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins. RESET BUTTON the Arduino has a reset button (10). Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times.

POWER LED INDICATOR Just beneath and to the right of the word “UNO” on your circuit board, there's a tiny LED next to the word „ON“ (11). This

LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit! Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non- contact measurement function, the ranging accuracy can reach to 3mm.

The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work: (1) Using IO trigger for at least 10us high level signal, (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back. (3) IF the signal back, through high level , time of high output IO dura LCD stands for liquid crystal display. They come in many sizes 8x1 , 8x2 , 10x2 , 16x1 , 16x2 , 16x4 , 20x2 , 20x4 ,24x2 , 30x2 , 32x2 , 40x2 etc. Many multinational companies like Philips Hitachi Panasonic make their own special kind of LCD's to be used in their products. All the LCD's performs the same functions (display characters numbers special characters ASCII characters etc).

Heir programming is also same and they all have same 14 pins (0-13) or 16 pins (0 to 15). A GSM modem is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone. GSM Modem comes in interfaces like USB, and Serial. GSM Modem is however the main difference is that GSM Modem is wireless, while dial-up modem is wired (telephone previously). GSM is used here to interface with microcontroller and Microcontroller command to the GSM modem with AT (abbreviation of Attention) command set implemented in our program.

4.3 SNAP SHOTS OF THE PROJECT:

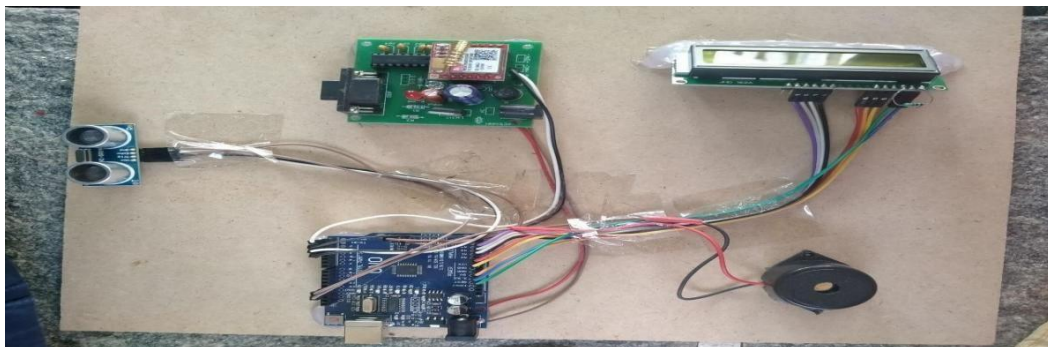


Fig 4.3 Experimental implementation of the project

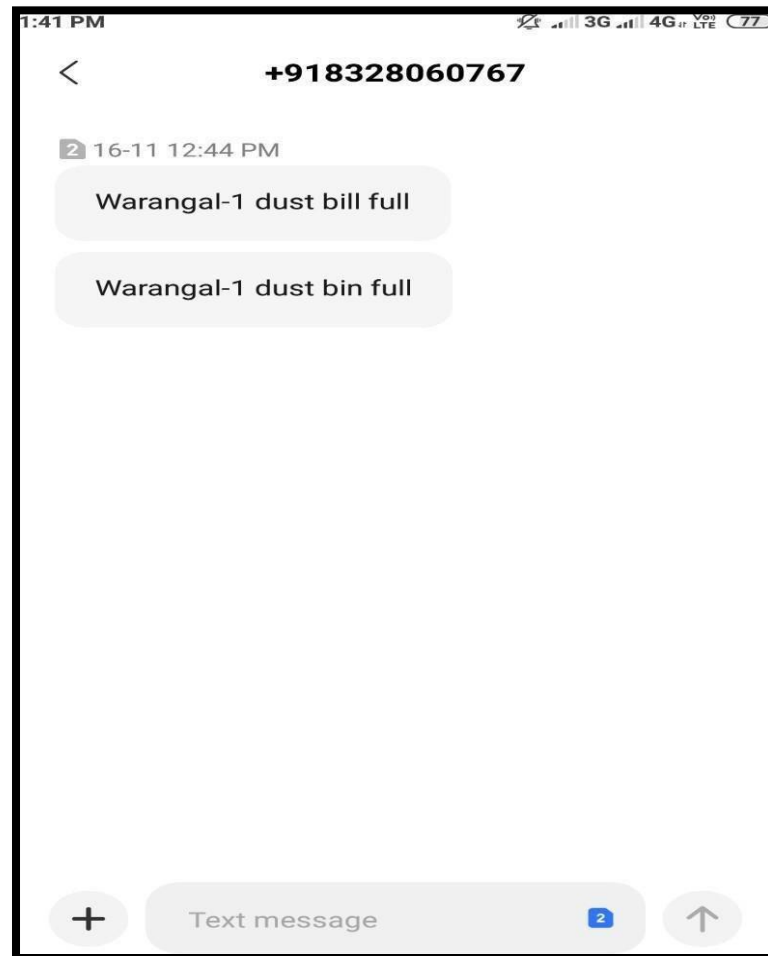


Fig 4.2: Message received from the GSM Module

4.4 RESULT:

The daily generated waste from urban cities must be collected by the municipality of the city but due to some constraints this waste is not being collected on time to avoid such attenuation we develop this project named as “GSM based Garbage wastage monitoring system”. The system detects garbage to dustbin & send message through GSM module. The message is received to the no. which we have given in the program. It gets details of the dustbin status from the SMS. It sends SMS in the form of percentage. Thus this system helps

to keep the city clean by informing about the garbage levels of the bins by providing SMS to the respective person.

4.5 ADVANTAGES:

1. Low design time and real time information on the fill level of the dustbin.
2. Avoids the overflows of garbage bins.
3. This system is applicable for both the indoor and outdoor environment.
4. Setting the destination is very easy.
5. It is dynamic system. It improves environment quality and reduces human effort.
6. Less space and better performance.
7. Low power consumption.
8. Monitoring and collection can be done in on time.
9. System can be accessed anytime and from anywhere.
10. Using this system, waste collection would become efficient and also reduction in transportation costs can be witnessed.

CHAPTER 5

CONCLUSION

In past few years, the growth of cities is rapidly going high. And in coming few years the cities would become developed and smart one. But, the smart city is incomplete without a smart garbage management system. So, we have designed a system for proper management of garbage. In this project we have studied and implemented concept of Smart City with the help GSM. This system assures the cleaning of dustbin soon when the garbage level reaches its maximum threshold. The Embedded C technique is enhancing the system to achieve the desired result. We have successfully implemented and tested the proposed system. From this paper, we believe that people get encouraged to build some other systems using different techniques and help nation to become diseases free. Also we believe, encouragement from the side of government can transform the prototype into a product.

REFERENCES

- [1] Flora A. "Towards a clean environment: A proposal on sustainable and integrated solid waste management system for universiti Kebangsaan Malaysia". Report from Alam Flora. 2009.
- [2] Anagnostopoulos,T., Zaslavsky,A., Kolomvatsos, K., Medvedev,A., Amirian,P., Morley,J., & Hadjiefthymiades,S. "Challenges and Opportunities of Waste Management in IoT-enabled Smart Cities: A Survey", IEEE Trans. Sustain. Comput.. , vol.2, no. 3, Apr. 2017,pp. 275-289.
- [3] S. Longhi et al., "Solid Waste Management Architecture Using Wireless Sensor Network Technology" In Proc. of 5th IEEE International Conference on New Technologies, Mobility and Security (NTMS), May. 2012, pp. 1-5.
- [4] Bhor,V., Morajkar,P., Gurav,M., Pandya,D., & Deshpande,A, "Smart Garbage Management System", International Journal of Engineering Research & Technology, vol. 4, no. 3, March. 2015, pp. 1117-1120.
- [5] Al Mamun M.A., Hannan M.A., and Hussain A., "Real time solid waste bin monitoring system framework using wireless sensor network",In proc. of the IEEE International Conference on Electronics, Information and Communications (ICEIC 2014), SEP. 2014, pp. 01-02.
- [6] M. Faccio, A. Persona, and G. Zanin, "Waste collection multi objective model with real time traceability data," Waste Management, vol. 31, no. 12, DEC. 2011,pp. 2391-2405..
- [7] Vikrant Bhor, Pankaj Morajkar, Amol Deshpande, "Smart Garbage Management System", International Journal of Engineering Research & Technology (IJERT), vol. 4, no. 3, MAR. 2015, pp. 1117-1120.
- [8] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?", In proc. of I st International Workshop on the Security of the Internet of Things (SecIoT10), DEC. 2010.
- [9] Narayan Sharma, Nirman Singha, Tanmoy Dutta, "Smart Bin Implementation for Smart Cities", International Journal of Scientific & Engineering Research, vol. 6, no. 9, SEP. 2015,pp. 787-791.

- [10] Prashant A. Shinde, Y. B. Mane, Pandurang H. Tarange, "Real time vehicle monitoring and tracking system based on embedded Linux board and android application", In proc. of IEEE International Conference on Circuit Power and Computing Technologies (ICCPCT), JULY. 2015, pp.01-07.
- [11] Sharif A, Li J, Khalil M, Kumar R, Sharif MI, Sharif A. Internet of Things - Smart Traffic Management System for Smart Cities Using Big Data Analytics. 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). Chengdu. 2017: 281-284.
- [12] Munadi R, Rakhman A, Perdana D. Smart Garage Implementation and Design Using Whatsapp Communication Media. TELKOMNIKA Telecommunication Computing Electronics and Control. 2018; 16(3): 1107-1113.
- [13] Javaid S, Sufian A, Pervaiz S, Tanveer M. Smart traffic management system using Internet of Things. 20th International Conference on Advanced Communication Technology (ICACT). Chuncheon-si Gangwon-do. 2018; 393- 398.
- [14] Yang J, Hao S, Zhang X, Guo B, Liu Y, Dong B, Liu Y. Family Health Monitoring System Based on the Four Sessions Internet of Things. TELKOMNIKA Telecommunication Computing Electronics and Control. 2015; 13 (1): 314-320.