ABSTRACT

Urbanization centralized on a single city (Kathmandu) has increased pollution tremendously. Waste management has become a crucial issue to be considered although no proper system has yet been established. In this paper, smart bin is built on a microcontroller based platform Arduino Uno board which is interfaced with GSM modem and Ultrasonic sensor. Ultrasonic sensor is placed at the top of the dustbin which will measure the stature of the dustbin. The threshold stature is set as 10cm. Arduino will be programmed in such a way that when the dustbin is being filled, the remaining height from the threshold height will be displayed. Once the garbage reaches the threshold level ultrasonic sensor will trigger the GSM modem which will continuously alert the required authority until the garbage has been picked up. This results in a smarter waste management system and a cleaner and healthier Nepal. Further venture if we get sufficient aids is to implement image processing to distinguish between degradable and non-degradable waste so that people can be warned on site of their improper use of dustbins. A beeping sound will be played every time a non-degradable waste enters into a dustbin for degradable waste and vice-versa.

Keywords: Object Oriented Programming Language, Arduino Integrated Development Environment, Microcontroller.

Contents

List of Figures:

Chapter 1: Introduction

1.1 Background

A smarter city is a better city. Technological advancement has helped to diversify various aspects of the modern day life. The present scenario of our country when it comes to waste management and control of pollution looks really weak. If we continue to ignore this matter, Nepal will become the most polluted country of the world in no time. With this scenario in mind we studied on the use of Arduino and simple sensors to build a Smart Dustbin that looks like a simple initiative but is an impactful one.

The Smart Dustbin has an ultrasonic sensor to sense distance from the overflowing waste. All the feedbacks are given to the waste managers through a message in their cellular phones. Using this Dustbin, time and effort factors will be reduced to a minimal because all the energy will go into only the places where actual management is needed. This helps make the city smarter.

We plan to design the perfect solution to an uprising problem and work on making this city a better one when it comes to use of technology in all the aspects of life.

1.2 Problem Statement

To construct a smart bin capable of informing the municipality of the status of dustbins (filled or filling) throughout the city using Arduino Uno interfaced with Ultrasonic sensor and GSM Module.

1.3 Objectives

To replace conventional dustbins with a smarter one.

To keep the concerned authorities informed about filling dustbins. To address the issues of waste management.

To utilize technology to make the city smarter.

1.4 Motivation and Significance

1.4.1 Motivation

Our motivation comes from reading about the Smart garbage monitoring system using sensors with RFID over internet of things. The above project works by if someone is putting some garbage into the smart bins photo electric sensor will detect the clear representation objects and weight

sensor will be placed below the smart bin it will calculate the percentage of the garbage is present inside the bin and with help of IR sensor it will send the information to the authorities who are responsible for that particular area .So, the particular authorities can receive the messages until the bin is squashed and the each bin is given an particular ID it will in display in the screen of the respected officer and they can take immediate action. If the person coming nearby two meters the bin which contain RFID CARD READER with the help of RFID reader it will read all the information about that particular person and send a message to him what they dropped inside the bin and appreciation message for using the bin. Thus, we wanted to make something similar using minimal resources.

1.4.2 Significance

In today's date, technology has become a major asset in all aspects of life. Technology is rapidly growing into a major dependency. With the increase in population we have an increase in the garbage around urban areas. Here we propose a smart dustbin that operates automatically to help solve this issue using Arduino [6] and sensor based circuitry. Usual dustbins require to be opened by pressing foot against its lever and then throwing garbage. Also a person needs to keep track when it is full so that it can be emptied and does not overflow. Here we propose a smart dustbin that does all this by itself.

While developing the system, there shall be space for further modification. There shall be a proper documentation so that further enhancement becomes easy. As a whole the car is focused as a useful system that will come in handy to make lives more digital and automatic.

Chapter 2: Related Works

2.1 IOT water level sensor [2]

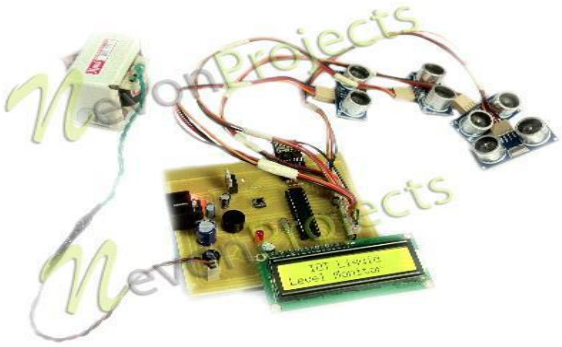

Figure 1: IOT Liquid Level Monitoring

This project IOT Liquid Level Monitoring system is a very innovative system which will inform the users about the level of liquid and will prevent it from overflowing. For this the system uses ultrasonic sensors placed over the containers to detect the liquid level and compare it with the container's depth. The LCD screen is used to display the status of the level of liquid in the containers. Whereas a web page is built to show the status to the user monitoring it.

2.2 Ultrasonic Range Finder [3]

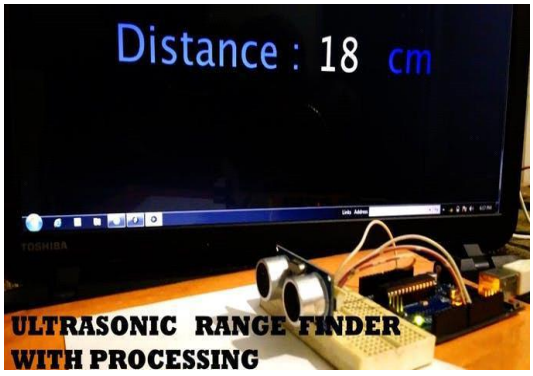

Figure 2: Ultrasonic Range Finder

This project uses ultrasonic sensor to measure the distance of obstacles. Ultrasonic sensor senses the obstacles and distance of obstacle are displayed on monitor using Arduino.

2.3 Parksol [2]

Figure 3: Parksol

The indicator PSL/USI is designed for mounting on the ceiling or a cable channel and show the availability of a parking space. It is recommended for covered parking lots. The indicator is connected to the ultrasound or magnetic ParkSol sensors. The system represents the number of free parking spaces on LED screens. The indicator has reflective optics, which ensures good visibility in the parking from long distance even in a sunny day. It has IP66 class. Indications options: red/green, red/blue, only green.

2.4 IOT Dustbin [5]

Figure 4: IOT Dustbin

This system consists of a sensor in order to detect human clap signal and on a clap of foot tap it opens automatically without anyone needing to press its lever. The dustbin opens automatically when it receives the signal and closes its hatch. Also the dustbin consists of a level sensing ultrasonic sensor that constantly measures the level of garbage in the bin and automatically detects if it is about to fill up.

Chapter 3: System Design

For communication purpose Bluetooth technology can also be used in the transmitter section. Bluetooth is a wireless networking standard that is aimed at remote control and sensor applications which is suitable for operation in harsh radio environments and in isolated locations. But, the main disadvantages of Bluetooth is short range, low complexity, and low data speed. Therefore, GSM is more advantages over Bluetooth for communication. Hence we use GSM modem. A GSM modem is a specialized type wireless modem that works with a GSM wireless network. It accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. A GSM modem can be an external device or a PC Card / PCMCIA Card. An external GSM modem is connected to a computer through a serial cable or a USB cable. When a GSM modem is connected to a computer, this allows the computer to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS message. GSM Modem sends and receives data through radio waves. It consists of a GSM/GPRS modem with standard communication interfaces like RS-232 (Serial Port), USB, so that it can be easily connected to the other devices. The power supply circuit is also built in the module that can be turn ON by using a suitable adaptor.

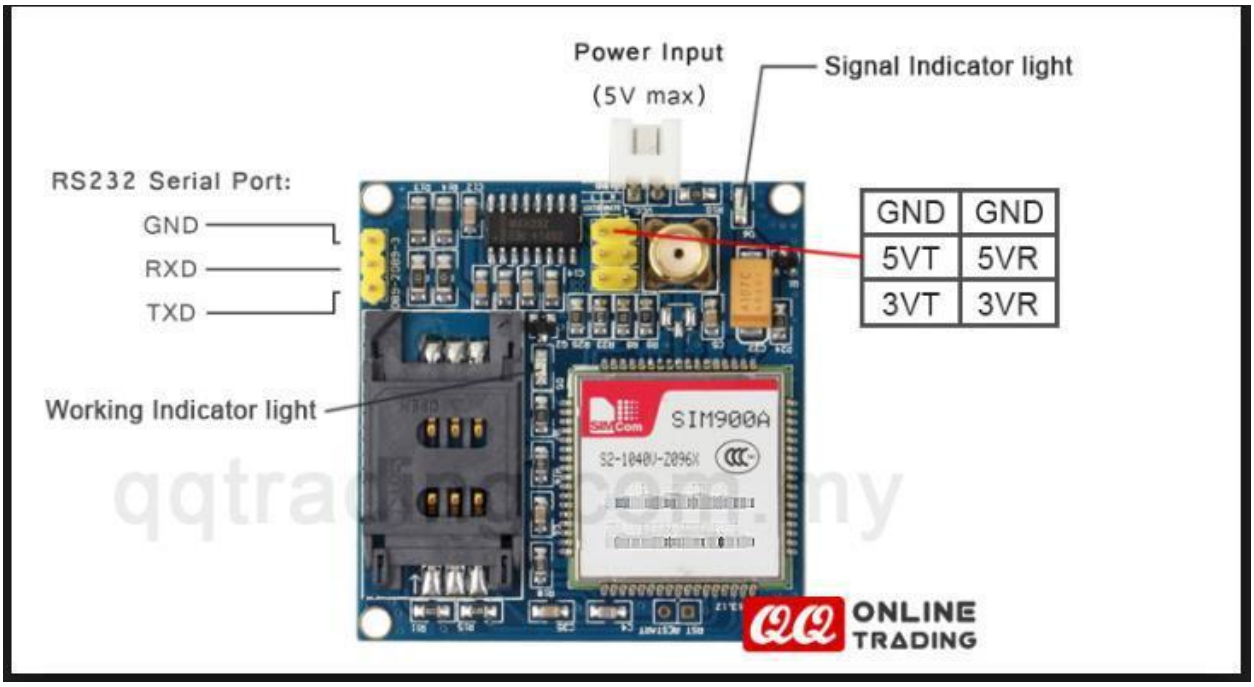

Figure 5: GSM Module pins

3.1 Block Diagram

The block diagram of transmitter section. Level detector consists of ultrasonic sensors which is used to detect the level of the garbage in the dustbin. The output of level detector is given to microcontroller. Ultrasonic sensor is used to indicate the different levels of the amount of the garbage collected in the dustbin which is placed in public area. When the dustbin is filled up to the highest level, this output is given to microcontroller to send the message to the concerned authority via GSM module as shown in below.

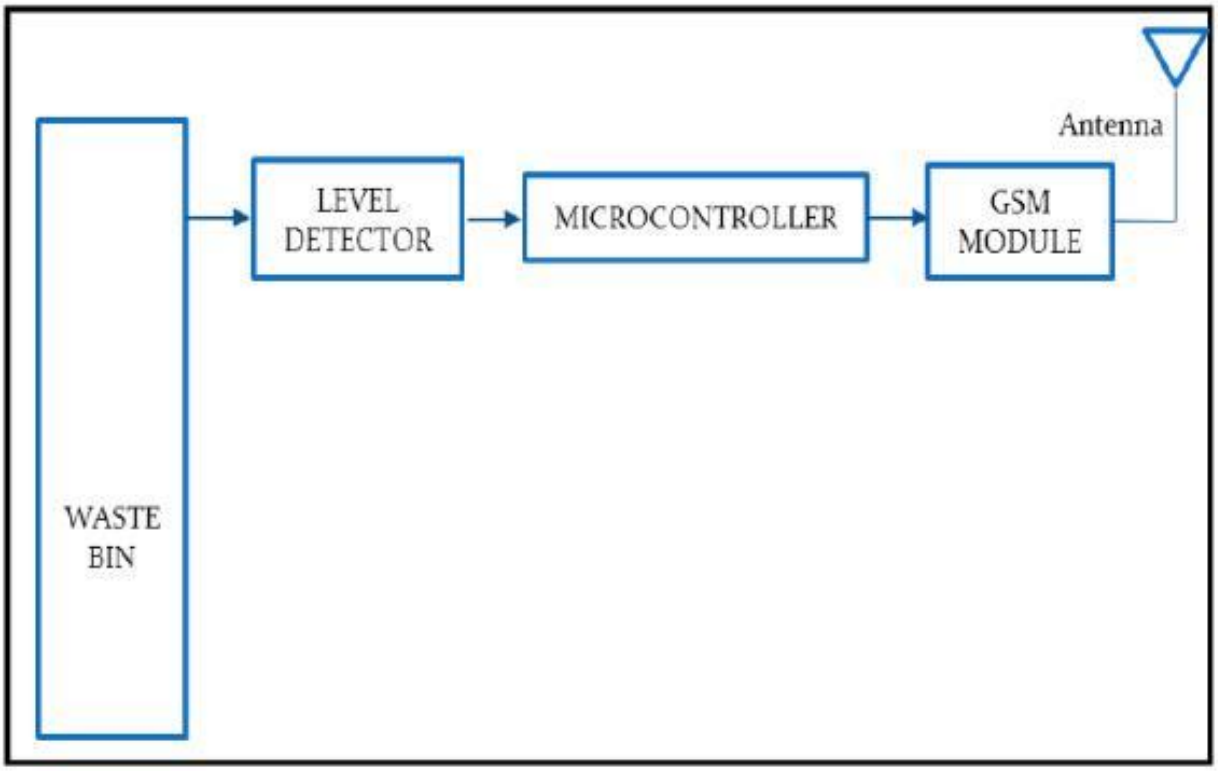

Figure 6: Transmission Part for dustbin

T

3.2 Flowchart

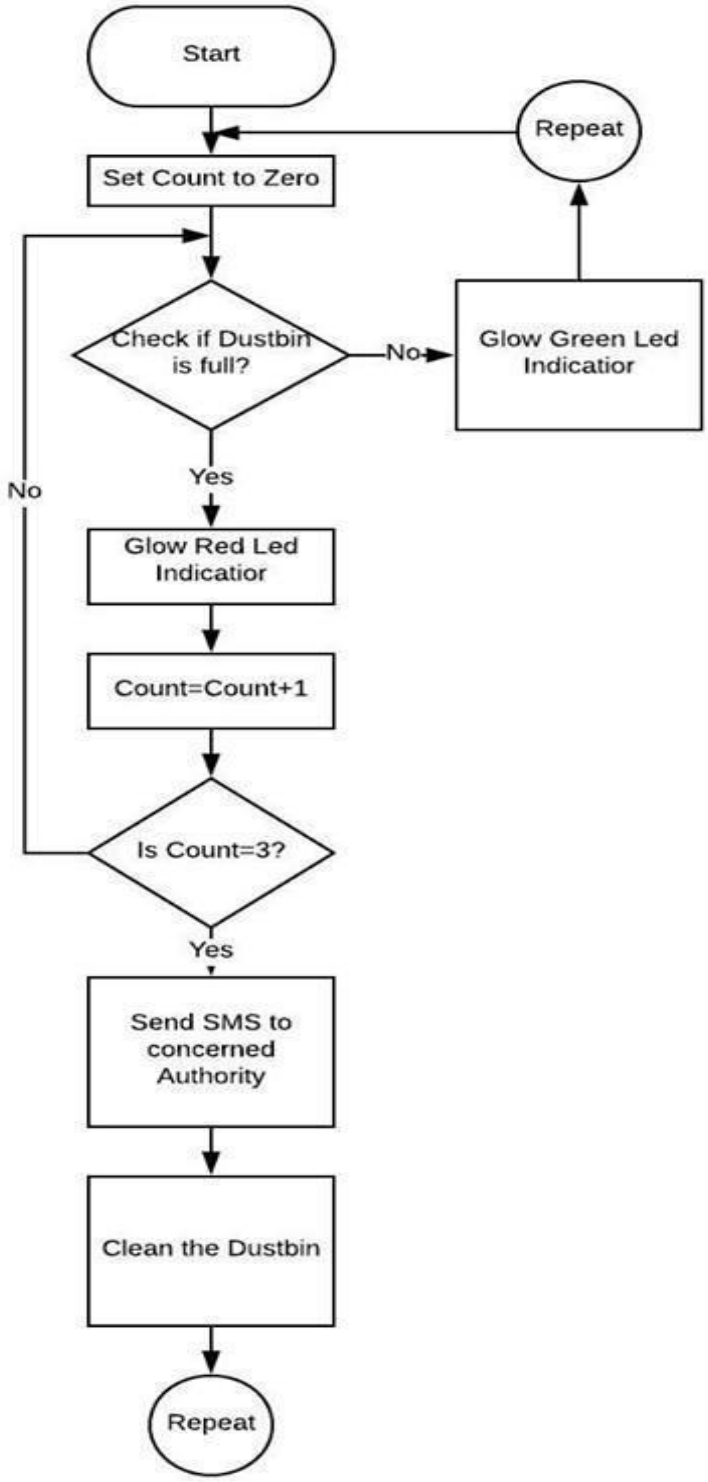

Figure 7: Flowchart for the system

3.3 Specifications and Requirements

3.3.1 Software Specifications

1. Programming Languages: Arduino
2. Operating System: Windows
3. IDE: Arduino IDE [6]

3.3.2. Hardware Specification

1. Arduino Uno [6]
2. Ultrasonic Sensor [6]
3. Breadboard [6]
4. GSM Module [6]

3.4 System Requirement Description

3.4.1 Software Description

3.4.1.1. Arduino IDE

The Arduino project provides the Arduino [6] 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.

Some functions:

1. **begin()**
   Sets the data rate in bits per second (baud) for serial data transmission. For communicating with the computer, use one of these rates: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200. You can, however, specify other rates - for example, to communicate over pins 0 and 1 with a component that requires a particular baud rate.
   Syntax: Serial.begin(speed)
2. **pinMode()**
   Configures the specified pin to behave either as an input or an output. Syntax: pinMode(pin, mode)
3. **digitalRead()**
   Reads the value from a specified digital pin, either HIGH or LOW. Syntax: digitalRead(pin)
4. **digitalWrite()**
   Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode(), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.
   Syntax: digitalWrite(pin, value)
5. **delay()**
   Pauses the program for the amount of time (in milliseconds) specified asparameter. (There are 1000 milliseconds in a second.)
   Syntax: delay(ms)
6. **pulseIn()**
   Reads a pulse (either HIGH or LOW) on a pin. For example, if value is HIGH, pulseIn() waits for the pin to go from LOW to HIGH, starts timing, then waits for the pin to go LOW and stops timing. Returns the length of the pulse in microseconds or gives up and returns 0 if no complete pulse was received within the timeout. The timing of this function has been determined empirically and will probably show errors in longer pulses. Works on pulses from 10 microseconds to 3 minutes in length.
   Syntax: pulseIn(pin, value)

3.4.1.2 Windows

Microsoft Windows [7] is a group of several graphical operating system families, all of which are developed, marketed, and sold by Microsoft. Each family caters to a certain sector of the computing industry. Active Windows families include Windows NT and Windows Embedded; these may encompass subfamilies, e.g. Windows Embedded Compact (Windows CE) or Windows Server. Defunct Windows families include Windows 9x, Windows Mobile and Windows Phone.

3.4.2 Hardware Description

3.4.2.1 Arduino Uno

Arduino[6] is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

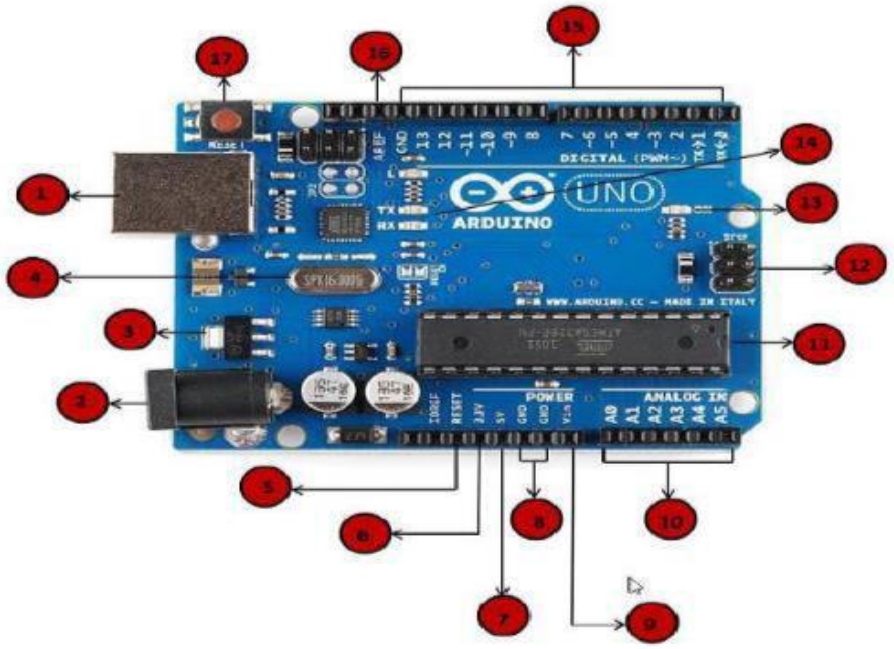

Figure 8: Arduino *Uno*

Description

1. **Power USB:**
   Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection.
2. **Power (Barrel Jack):**
   Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

3. **Voltage Regulator:**
   The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4. **Crystal Oscillator:**
   The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5. **(5 and 17) Arduino Reset:**
   You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6. **(6 – 9)**
   3.3V (6) − Supply 3.3 output volt
   5V (7) − Supply 5 output volt Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
   GND (8) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.
   Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply

10. **Analog pins:**
    The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11. **Main microcontroller:**
    Each Arduino board has its own microcontroller. You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12. **ICSP pin:**
    Mostly, ICSP is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13. **Power LED indicator:**
    This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14. **TX and RX LEDs:**
    On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins

responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15. **Digital I/O**:
    The Arduino UNO board has 14 digital I/O pins. (15) (Of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled can be used to generate PWM.

16. **AREF:**
    AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

    Microcontroller- ATmega2560
    Operating Voltage -5V
    Input Voltage (recommended) - 7-12V
    Input Voltage (limit) - 6-20V
    Digital I/O Pins 54 (of which 15 provide PWM
    output) Analog Input 16

3.4.2.2 Ultrasonic sensor (HCSR04)

The HC-SR04 ultrasonic sensor [6] uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1" to 13 feet. Its operation is not affected by sunlight or black material like sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

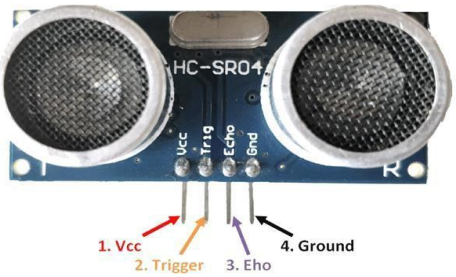

Figure 9: HC-SR04

Technical Specifications

Power Supply − +5V DC

Quiescent Current − <2mA

Working Current − 15mA

Effectual Angle − <15°

Ranging Distance − 2cm – 400 cm/1″ – 13ft

Resolution − 0.3 cm

Measuring Angle − 30 degree

3.4.2.3 Breadboard

Breadboards [6] are one of the most fundamental pieces when learning how to build circuits. Common use of breadboards is testing out new parts, such as Integrated circuits (ICs). When trying to figure out how a part works and constantly rewiring things, we want to have to solder connections each time.

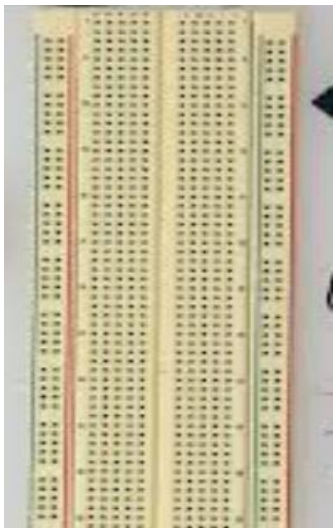

Figure 10: Breadboard

3.4.2.4 GSM Model

A GSM Module is basically a GSM Modem(like SIM 900) [6] connected to a PCB with different types of output taken from the board – say TTL Output (for Arduino, 8051 and other microcontrollers) and RS232 Output to interface directly with a PC (personal computer). The board will also have pins or provisions to attach mic and speaker, to take out +5V or other values of power and ground connections. These type of provisions vary with different modules.

Figure 11: GSM Module

A GSM network comprises of many functional units. These functions and interfaces are explained in this chapter. The GSM network can be broadly divided into:

1. The Mobile Station (MS)
2. The Base Station Subsystem (BSS)
3. The Network Switching Subsystem (NSS)
4. The Operation Support Subsystem (OSS)

Given below is a simple pictorial view of the GSM architecture

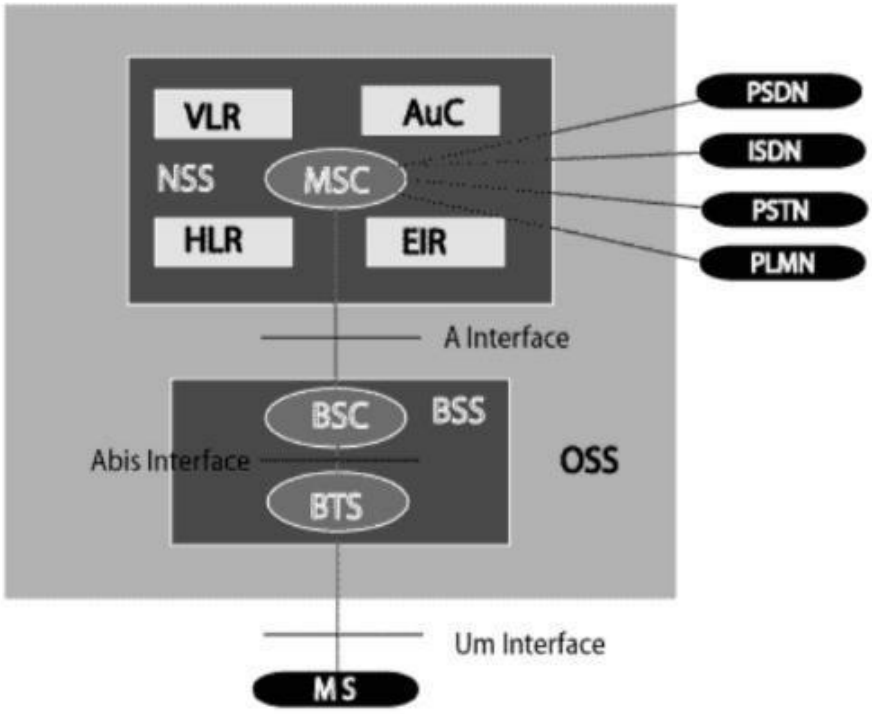

Figure 12: GSM Architecture

In a GSM network, the following areas are defined:

Cell: Cell is the basic service area; one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell.

Location Area: A group of cells form a Location Area (LA). This is the area that is paged when a subscriber gets an incoming call. Each LA is assigned a Location Area Identity (LAI). Each LA is served by one or more BSCs.

MSC/VLR Service Area: The area covered by one MSC is called the MSC/VLR service area.

PLMN: The area covered by one network operator is called the Public Land Mobile Network (PLMN). A PLMN can contain one or more MSCs.

Chapter 4: Implementation

The following steps have been performed to complete the project

1. Arduino is set up using a PC
2. Ultrasonic sensor is connected to the Arduino. Vcc is given 5V while the Trigpin and Echopin are connected to pin 11 and 12 respectively.
3. A breadboard is connected to establish connection between Arduino, ultrasonic sensor and GSM module.
4. The Vcc and GND of all the components are connected to the breadboard.
5. The TX and RX pins of the GSM module are connected to the number 5 and number 6 pin of Arduino.
6. Two LEDS are connected. The cathode is connected to pin 2 (red) and pin 4(green) while the anode is connected to the ground.
7. Necessary coding is done to run the total system as one.

4.1. Working Principle:

Figure 13: System Diagram

The technology behind the Arduino Smart Dustbin is pretty simple. There are mainly three blocks behind it: input, controller, and output. The input consists of an ultrasonic sensor that is capable of detecting obstacles in front of it at a range of up to about 10 feet. It is interfaced to the Arduino, which determines if the dustbin is full and sends the output in a SMS format via a Sim in the GSM module.

4.2 Interfacing the Arduino

A breadboard is used to connect the ultrasonic sensor to the Arduino using jumper wires. The GSM Module is directly connected to the Arduino. Two LEDs are used to note down the response of the system. The wiring diagram is shown below.

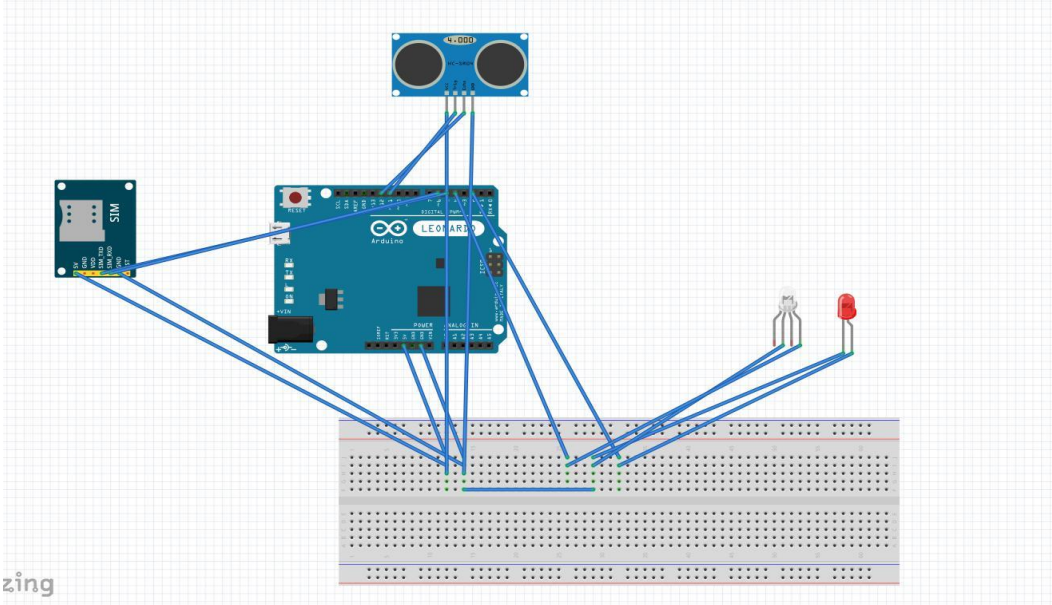

Figure14: Overall Combination with Arduino

Here are the connections for each part:

- Ultrasonic VCC to Arduino 5V.
- Ultrasonic GND to Arduino GND.
- Ultrasonic TRIG to Arduino pin 11.
- Ultrasonic ECHO to Arduino pin12.
- LED RED to Arduino pin 2.
- LED GREEN to Arduino pin 4.
- GSM Module TX to Arduino pin 5.
- GSM Module RX to Arduino pin 6.

SIM900 AT commands

set the SIM900 to text mode: AT+CMGF=1\r
send SMS to a number: AT+CMGS=PHONE_NUMBER (in international format)

The working principle of ultrasonic sensor is as follows:

High level signal is sent for 10us using Trigger.

The module sends eight 40 KHz signals automatically, and then detects whether pulse is received or not.
If the signal is received, then it is through high level. The time of high duration is the time gap between sending and receiving the signal.

Distance= (Time x Speed of Sound in Air (340 m/s))/2

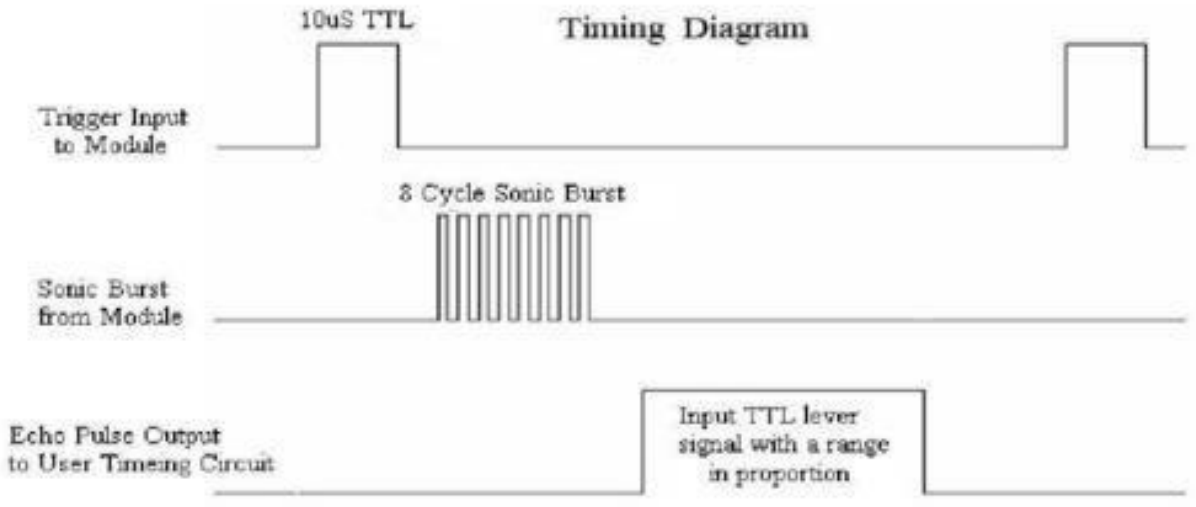

Figure15:Timing Diagram for ultrasonic sensor

4.3 Source Code

```
#include<SoftwareSerial.h> // Supports Serial Communication
SoftwareSerial GPRS(5, 6); // Set up communication on pin 5 and 6
int trigPin = 11; // trigger pin set to 11 for the ultrasonic sensor
int echoPin = 12; //ehco pin set to 11 for the ultrasonic sensor
int led = 2;
int ledg = 4;
int c = 0;
void setup() {
  pinMode(led, OUTPUT);
  pinMode(ledg, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  GPRS.begin(9600);
  Serial.begin(9600);
  delay(10000);
}
// main program
void loop() {
  long duration, distance;
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(1000);
  digitalWrite(trigPin, LOW);
  duration=pulseIn(echoPin, HIGH);
  distance =(duration/2)/29.1; // calculating the distance using the ultrasonic
  sensor Serial.print(distance); //printing distance on serial monitor
```

```
  Serial.println("CM");
  delay(1000);
  if((distance<=10))
  {
//setting counter to estimate time when the dustbin is full, this is done because sensor may be triggered //accidently some times
    c = c + 1;
    digitalWrite(ledg, LOW);
    delay(500);
    digitalWrite(led, HIGH);
    Serial.println("c =");
    Serial.print(c);
    delay(500);
    if((c==3)){
    delay(1000);
    sendSMS(); // calling sendSMS module
    delay(500);
    Serial.println("Sent SMS!");
    Serial.println( "dustbin is reached max level");
    delay(100);
    }
    delay(6000);

  }
  else if(distance>10)
  {
     c = 0;
```

```
      Serial.println("c =");
      Serial.print(c);
      digitalWrite(ledg, HIGH);
      delay(500);
      digitalWrite(led, LOW);
      delay(6000);
      }
}
void sendSMS()
{
GPRS.println("AT+CMGF=1");    // Set modem to text mode
delay(2000);
GPRS.write("AT+CMGS=");        // Start composing message
GPRS.write(0x22);              // hex equivalent of double-quote ""
GPRS.write("+9779813623309"); // the number to send the SMS to
GPRS.write(0x22);  GPRS.write(0x0D);              // hex equivalent of Carraige return
GPRS.write(0x0A);           // hex equivalent of newline
delay(2000);
GPRS.print("ID:=1 LOCATION:=KU DOCSE : THE ABOVE DUSTBIN REACHED ITS MAXIMUM CAPACITY PLEASE COME AND COLLECT TO KEEP CITY CLEAN"); // Send the text message to the GSM module
GPRS.write(0x0D);           // hex equivalent of Carraige return
GPRS.write(0x0A);           // hex equivalent of newline
delay(1000);
GPRS.write(26);         // equivalent of CTRL-Z
delay(5000);
}
```

4.4 Snapshots

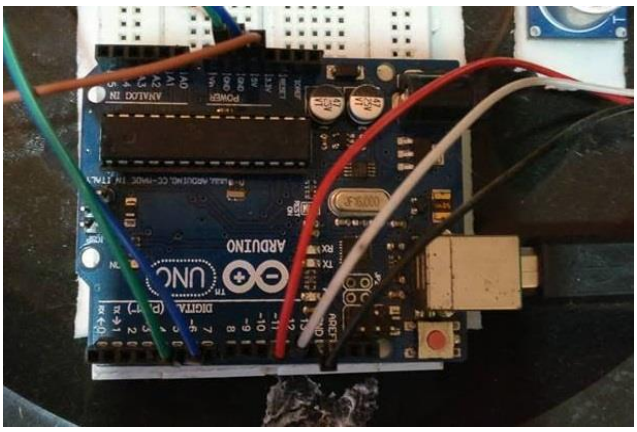

Figure16: Arduino Setup

Figure17: Ultrasonic sensor Used

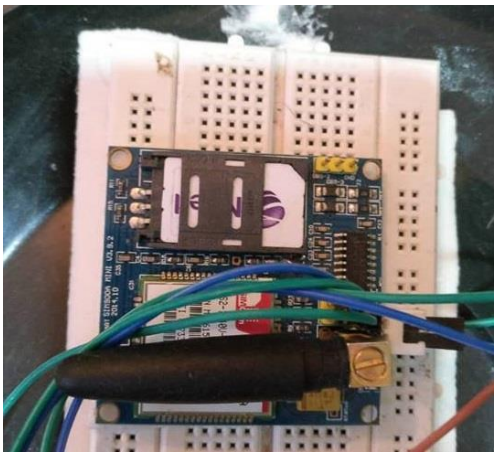

Figure18 GSM module 900a with Ncell Sim

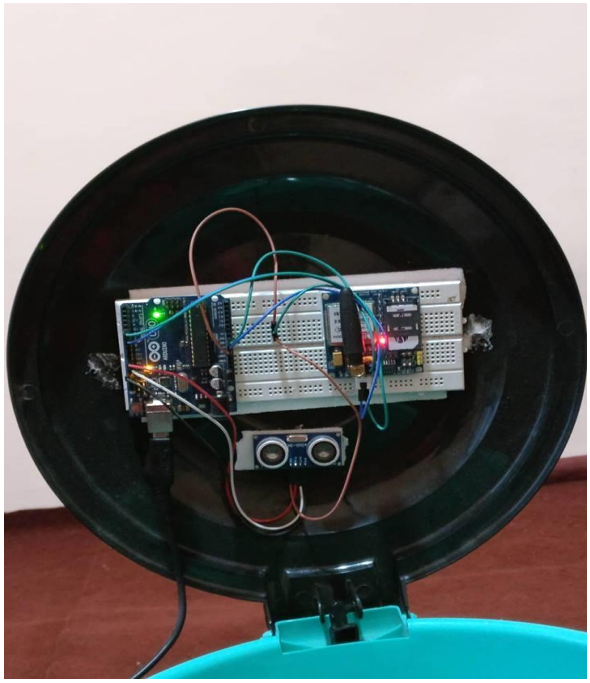

Figure19: Connection set up on the lid of the dustbin.

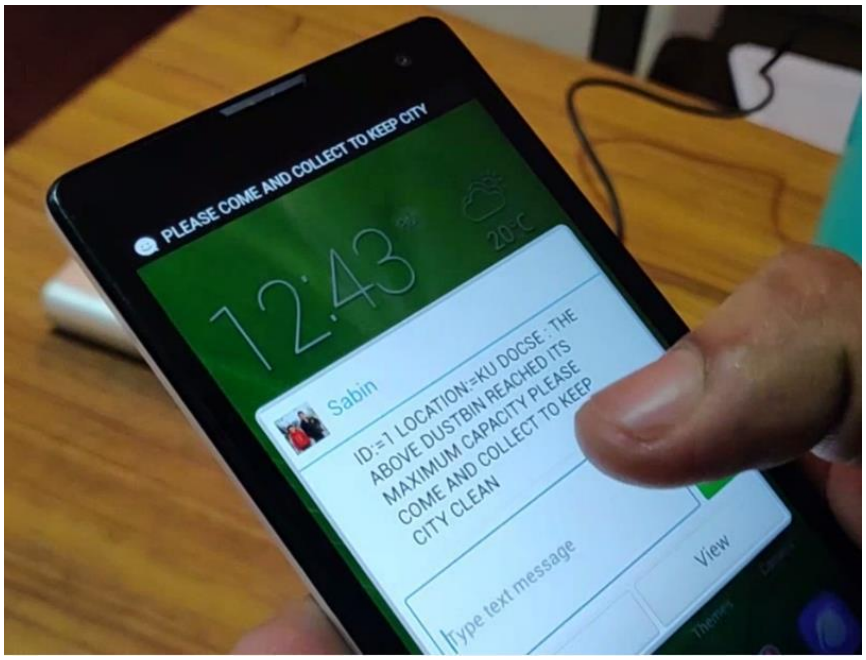

Figure20: SMS on mobile phone.

Chapter 5: Discussion on the Achievements

5.1 Challenges faced:

The major challenge we faced during the project is limited resources and insufficient sensors and hardware for the project. As a result, we couldn't build a more advanced smarter dustbin to make the car completely self-driving.

Likewise, since this was our first hardware project, the parts kept on malfunctioning at times which wasted a lot of time. The poor-quality connectors and wires also created a hindrance in getting the desired result from the program. Moreover, a major problem was finding the appropriate power source to supply sufficient power for the Breadboard, Arduino and DC motors without a PC connection. This hindered the further growth of the project.

5.2 Features:

1. **The dustbin detects waste and senses the distance the waste from it.**
   We have used an ultrasonic sensor to detect the obstacles. When an object is detected, a signal is sent to the Arduino and processed through it.
2. **Presence of Message sending mechanism**
   The GSM Module is used to send messages to the concerned authorities.
3. **Presence of LED indicators**
   LED indicators are present to show whether the dustbin is full or empty with the help of red and green LED lights.

Chapter 6: Cost Estimation

S.N.	Equipment	Cost per Unit	Quantity	Total Cost
1	Arduino	1500	1	1500
2	Ultrasonic Sensor	300	1	300
3	GSM Module	2000	1	800
4	USB Cable	150	1	150
5	Breadboard	120	1	120
6	Male to Male	65/packet	1	65
				Total: 2937

Table 1: Cost Estimation

Chapter 7: Conclusion and Recommendation

This research report presents the implementation of a smart dustbin that helps in proper waste management throughout the county because cell phones service has reached everywhere. Such small initiatives combine to make a better system and a better country as a whole. Moreover, this project can really take off with a substantial impact in the real world.

6.1 Limitations

Image processing and Object detection Mechanism could not be incorporated in our project due to unavailability of better sensor like IR sensor

System becomes inconsistent when network is not available.

No indication of various stages of waste filling in the management system. System needs to be frequently maintained.

6.2Future Enhancements

Our main enhancement that we are working to achieve is the distinguishing of degradable and non–degradable waste. For example:

Ram is a man that throws plastic in a Smart Bin. The Smart Bin will identify the plastic product and beep. A voice requests Ram to remove it. This will compel him to use the dustbin properly. Combine millions of Ram and we obtain a systematic society which ultimately leads to a cleaner Nepal.

The level of wastes in water bodies and the density of pollution can be calculated using the same algorithm and proper sensors.

Water level in tanks can be monitored to prevent overflow at homes.

Use of proper detection technique to detect the various stages of waste filling (i.e. warn when half-filled or indicate the current level of waste in every dustbin)

References

[1] Smart Garbage Monitoring System Using Arduino 101 Available at: https://www.hackster.io/Technovation/smart-garbage-monitoring-system-using-arduino-101-3b813c [Accessed 14 Apr. 2018].

[2] Smart Trash Bin 1.0 Available at: https://create.arduino.cc/projecthub/MaximaEnfield/smart-trash-bin-1-0-6c08a7

[Accessed 14Apr. 2018].

[3] Kim, S. and Cho, K. (2018). Usability and Design Guidelines of Smart Canes for Users with Visual Impairments. [Online] International Journal of Design. Available at: http://ijdesign.org/index.php/IJDesign/article/view/1209/559 [Accessed 15 Apr. 2018].

[4] Smart Dustbin Using Arduino Available at: https://www.robotsthenextspeciesonearth.com/p/blog-page_72.html [Accessed 9 Apr. 2018].

[5] Smart Dustbin using Arduino Uno Available at: http://www.nabnepal.org/status-of-blind [Accessed 15 Apr. 2018].

[6] About Arduino at:

https://www.arduino.cc/en/Guide/Introduction [Accessed: 12 Apr, 2018]

[7] About Embedded systems; https://www.wikipedia.org/embeddedsystems/Introduction

[Accessed: 2nd April, 2018]

Gantt chart:

Week	1	2	3	4	5	6	7	8	9	10	11	12
Documentation												
Testing												
Coding												
System design												
Requirement analysis												