A Theme-based Project Report

on

Smart Garbage Monitoring System

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

By

K AJAY REDDY

1602-20-733-002

G KRISHNA SAMEER

1602-20-733-013

Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Computer Science & Engineering

DECLARATION BY THE CANDIDATES

We, K Ajay Reddy, G Krishna Sameer bearing hall ticket number, 1602-20-733-002,1602-20-733-013, hereby declare that the project report entitled "**SMART GARBAGE MONITORING SYSTEM**" Department of Computer Science & Engineering, VCE, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Computer Science & Engineering.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

K Ajay Reddy, G Krishna Sameer

1602-20-733-002,1602-20-733-013

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Computer Science & Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**SMART GARBAGE MONITORING SYSTEM**" being submitted by K Ajay Reddy, G Krishna Sameer bearing 1602-20-733-002,1602-20-733-013, in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering is a record of bonafide work carried out by him/her under my guidance.

Internal Guide:	P. Narsaiah
Dr. P. Bhargavi	Assistant Professor
Professor, CSE.	Dept. of CSE.

ACKNOWLEDGEMENT

We express our sincere indebtedness towards our guide Dr.P.Bhargavi ma'am, Department of Computer Science and Engineering, for her invaluable guidance, suggestions and supervision throughout the project. Without her kind patronage and guidance, the project would not have taken shape. We would also like to express our gratitude and sincere regards for her kind approval of the project, time to time counselling and advice.

We would also like to thank our professor, Dr.P.Bhargavi ma'am and all other faculty members in the Department of Computer Science and Engineering, Vasavi College of Engineering, Hyderabad, for their kind guidance and encouragement time to time.

TABLE OF CONTENTS

PAGE NO.

LIST OF FIGURES

Page no

Abstract

The paper is based on the concept of Automation used in waste management system under the domain of Cleanliness and Hygiene. Dumping garbage onto the streets and in public areas is a common synopsis found in all developing countries and this mainly end up affecting the environment and creating several unhygienic conditions. In order to deal with these problems Smart dustbin is an ideology put forward which is a combination of hardware and software technologies i.e. connecting Wi-Fi system to the normal dustbin in order to provide free internet facilities to the user for a particular period of time. The technology awards the user for keeping the surrounding clean and thus work hand in hand for the proper waste management in a locality. Smart dustbin uses multiple technologies firstly the technology for measuring the amount of trash dumped secondly the movement of the waste and lastly sending necessary signals and connecting the user to the Wi-Fi system. Whenever the maximum level is reached, alert will be sent to the local authorities.

Introduction

IOT based garbage level monitoring system is an emerging technology that is utilized for monitoring waste fill level of public and industrial garbage bins. The fundamental purpose of a garbage level monitoring system is to help the municipal services to pick the trash at the right time before a garbage bin overflows and cause discomfort to general public.

IoT based garbage monitoring systems is not just for an efficient operation of municipal services but also for understanding general population's goods consumption and trend.

By understanding people's goods consumption and current trend a government can decide things like how much funds need to be allocated for the current or next financial year and depending on the consumption trend a government can invest in new technology to tackle pollution due to an emerging trend.

The traditional way of manually monitoring the wastes in waste bins is a cumbersome process and utilizes more human effort, time and cost which can easily be avoided with our present technologies.

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

Software/ Hardware Requirement Specifications

- **ARDUINO UNO**

  The Arduino UNO is a microcontroller board-based chip on the ATmega328P. It has sets of digital input/output pins (of which some can be used as ultrasonic sensor inputs), 6 analog inputs, 14 digital inputs - 4 of which are programmable with Arduino IDEusing USB cable. Arduino Uno can be powered using an external power supply which ranges from 7-20 volts. The USB port in the Arduino board is used to connect the board to the computer using the USB cable. The cable acts as a serial port and as the power supply to interface the board. Such dual functioning makes it unique to recommend and easy to use for beginners.

Fig. 1. Arduino UNO

- **Node MCU**

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266WiFi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language.

Fig. 2. Node MCU

- **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.

Fig. 3. Buzzer

- **Jump wires**

  Jumper wires are simply wires that have connector pins at each end,allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires.

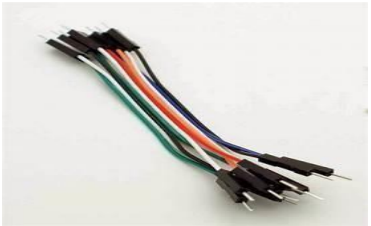


Fig. 4. Wires

- **Ultrasonic Sensor**
  The HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver.

Fig. 5. Ultrasonic Sensor

- **LCD**

  An electronic device that is used to display data and the message is known as LCD 16×2. As the name suggests, it includes 16 Columns & 2 Rows so it can display 32 characters (16×2=32) in total & every character will be made with 5×8 (40) Pixel Dots. So the total pixels within this LCD can be calculated as 32 x 40 otherwise 1280 pixels.

Fig. 6. LCD

System Architecture (Design)

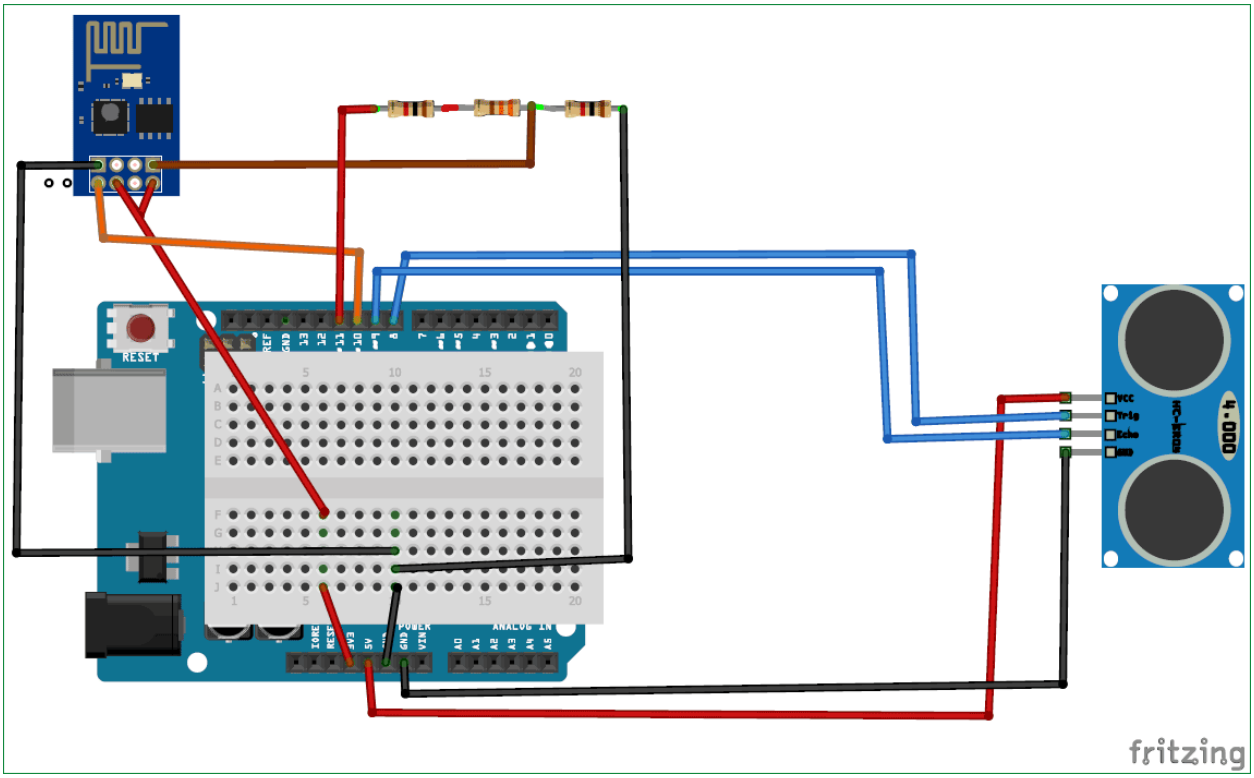


Fig. 7. Circuit Diagram

BLOCK DIAGRAM

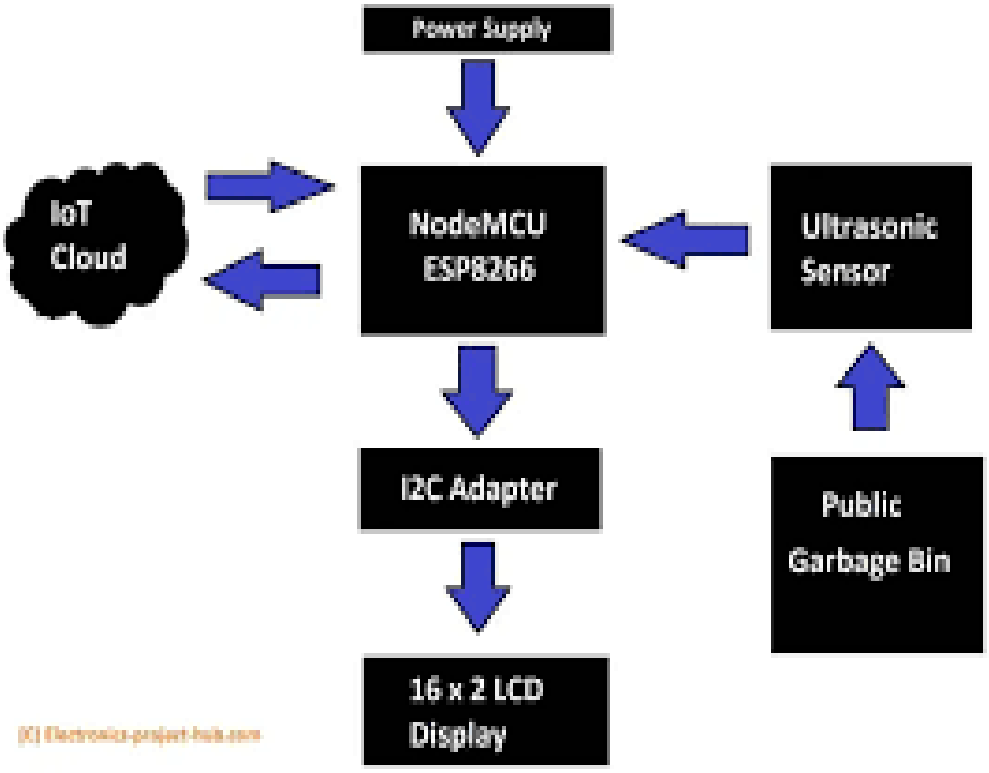


Fig. 8. Block Diagram

FLOWCHART

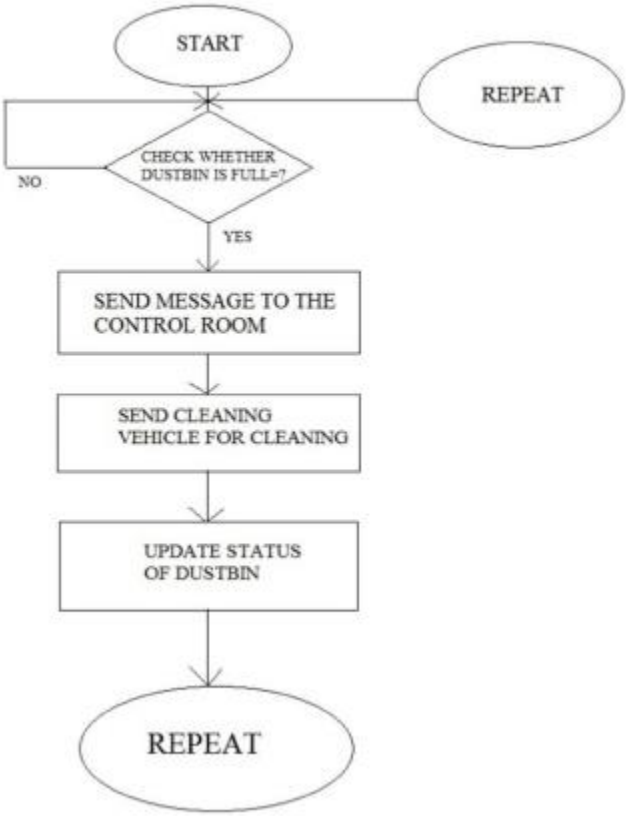


Fig. 9. Flowchart

Implementation Code.

```
#include <WiFiClientSecure.h>   // Include the HTTPS library
#include <ESP8266WiFi.h>        // Include the Wi-Fi library
#include <ESP8266WiFiMulti.h>   // Include the Wi-Fi-Multi library
#include "Arduino.h"
#include <EMailSender.h>
ESP8266WiFiMulti wifiMulti;     // Create an instance of the ESP8266WiFiMulti class, called 'wifiMulti'
uint8_t connection_state = 0;
uint16_t reconnect_interval = 10000;
WiFiClient  client;
String data1="";
String data2="cmd";
String data=" MESSAGE";
EMailSender emailSend("sivaprasadpilla7@gmail.com", "tvgcjlhvnpdetvvq");
void gmail()
{
  EMailSender::EMailMessage message;

    message.subject = "SMART_BIN_"+data1;
    message.message = "This_is_Mail_From_ESP8266:"+data1;

    EMailSender::Response resp = emailSend.send("saisathwik405@gmail.com", message);

    Serial.println("Sending status: ");
```

```
    Serial.println(resp.status);
    Serial.println(resp.code);
    Serial.println(resp.desc);
}
void upload()
{
const char* server4 = "api.thingspeak.com";
const char* _getLink4 =
"https://api.thingspeak.com/update?api_key=91SHW5GUCCRO6Y5M&field1=";
// Thingspeak.com
//const char* _getLink4 =
"https://api.thingspeak.com/update?api_key=6LYMPTR0BSXNNR5X&field1=";
// Thingspeak.com


 // Serial.println("data uploading");delay(1000);
  client.connect(server4,80);
 if (client.connect(server4,80))     // "184.106.153.149" or api.thingspeak.com
https://api.thingspeak.com/apps/thinghttp/send_request?api_key=CT9B331KB5PL
M1G5
  {
    String getStr4 = _getLink4;
    client.print("GET "+getStr4+data1+"\n");
    client.print("HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n\n\n");
  }
  client.stop();
```

```
}

void readdata()
{
  data1="";delay(1000);
const char* server4 = "api.thingspeak.com";
const char* _getLink4 = "
https://api.thingspeak.com/channels/562742/fields/1/last.txt"; // Thingspeak.com



  //Serial.println("data uploading");delay(1000);
  client.connect(server4,80);
 if (client.connect(server4,80))     // "184.106.153.149" or api.thingspeak.com
https://api.thingspeak.com/apps/thinghttp/send_request?api_key=CT9B331KB5PL
M1G5
  {
    String getStr4 = _getLink4;
    client.print("GET "+getStr4+"\n");
    client.print("HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n\n\n");
    client.available();
    data1=client.readString();delay(1000);
    //Serial.println(data1);delay(1000);

if(data1[0]=='*')
{
  if(data2==data1)
```

```
  {


  }
  else
  {
  Serial.println(data1);upload();
  }
  data2=data1;
}
/*


if((data1=="light1on")||(data1=="light1off")||(data1=="light2on")||(data1=="light2
off")||(data1=="fan1on")||(data1=="fan1off")||(data1=="fan2on")||(data1=="fan2off
"))
{
  Serial.print(data1);delay(1000);upload();
}


    if(data1[0]=='*')
    {
   Serial.println(data1);delay(10000);upload();
    }
    if((data1=="1")||(data1=="2")||(data1=="3")||(data1=="4")||(data1=="0"))
    {
   Serial.print(data1);delay(10000);
      }
      */
```

```
  }
  client.stop();
}



void setup()
{
  Serial.begin(9600);         // Start the Serial communication to send messages to the computer
  delay(10);
  //Serial.println('\n');

  wifiMulti.addAP("consciencetechnologies", "484conscience777");   // add Wi-Fi networks you want to connect to
  wifiMulti.addAP("sivaji", "sivaji.123");
  wifiMulti.addAP("ZTE-sUQdqa", "5hjgxyh9");
  wifiMulti.addAP("project", "project.123");
  wifiMulti.addAP("123456789", "123456789");

  //Serial.println("Connecting ...");
  int i = 0;
  while (wifiMulti.run() != WL_CONNECTED) { // Wait for the Wi-Fi to connect: scan for Wi-Fi networks, and connect to the strongest of the networks above
    delay(250);
    //Serial.print('.');
  }
  //Serial.println('\n');
```

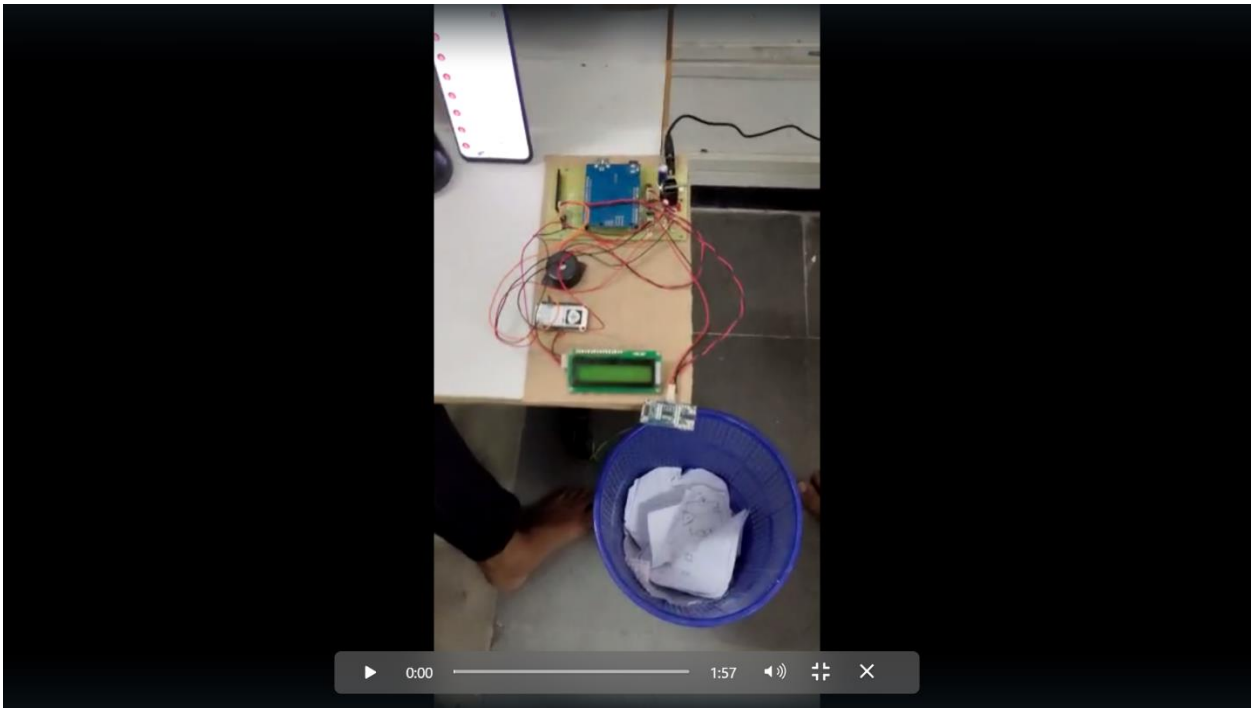
```
  //Serial.print("Connected to ");
  //Serial.println(WiFi.SSID());              // Tell us what network we're connected to
  //Serial.print("IP address:\t");
 Serial.println(WiFi.localIP());           // Send the IP address of the ESP8266 to the computer
  //Serial.println('\n');

  //readdata();
//gmail();
}


void loop()
{

while(1)
{
 // readdata();

 while(Serial.available())
 {
  data1=Serial.readString();delay(1000);
gmail();
//upload();
 }

}
```

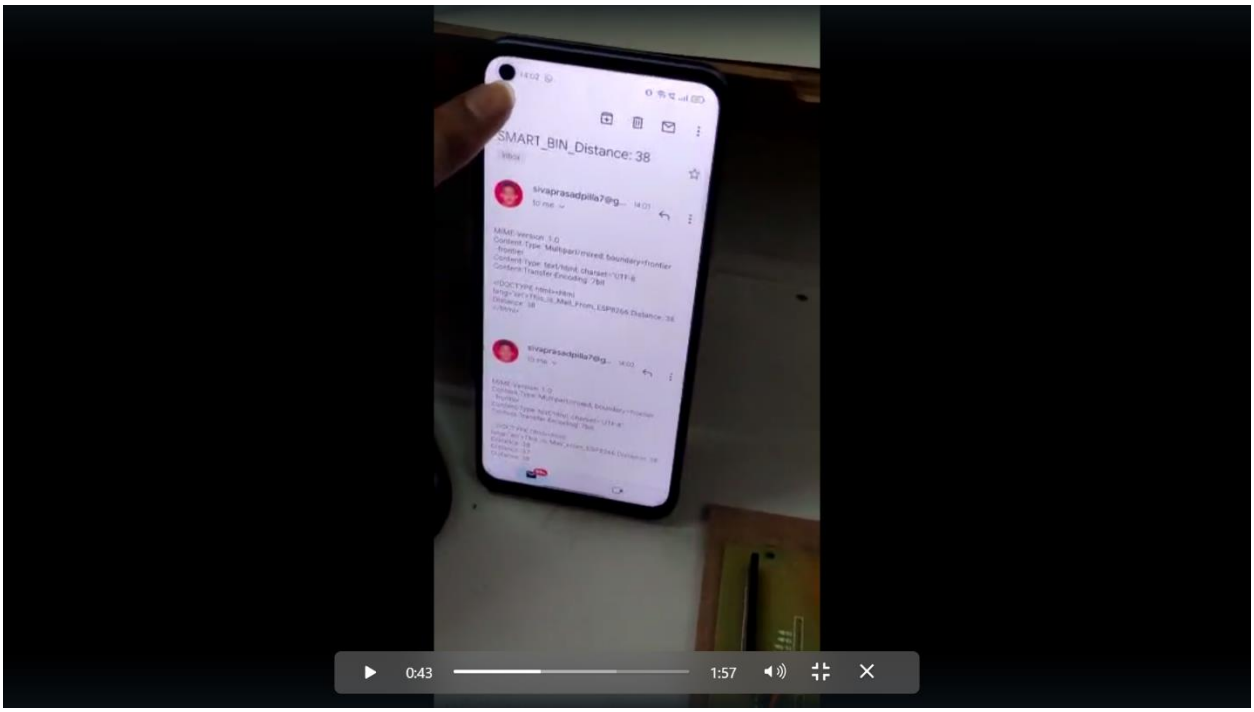
```
//YWROBOT
//Compatible with the Arduino IDE 1.0
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>


LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for a 16 chars and 2 line display
const int trigPin = 7;
const int echoPin = 6;
long duration;
int distance;
int buzzer=A0;
void setup()
{
  pinMode(buzzer, OUTPUT);digitalWrite(buzzer,LOW);
   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
  lcd.init();                      // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("Hello, world!");
  lcd.setCursor(2,1);
```

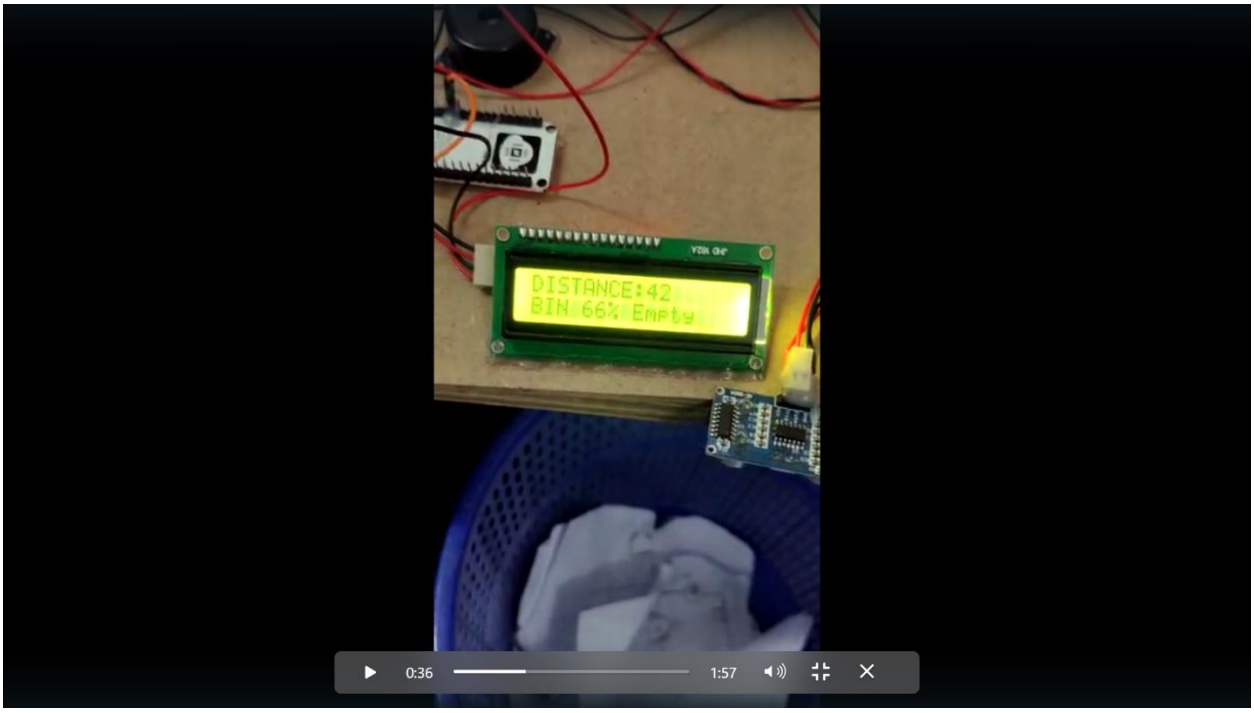
```
  lcd.print("Ywrobot Arduino!");

}


void loop()
{
   digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
  lcd.clear();lcd.print("DISTANCE:");lcd.print(distance);delay(1000);
  if(distance<=10)
  {
    lcd.setCursor(0,1);lcd.print("BIN 95% FULL");delay(1000);
   digitalWrite(buzzer,HIGH);delay(100);digitalWrite(buzzer,LOW);delay(100);
   Serial.println("BIN_FULL");delay(1000);
  }
```

```
 else if(distance>10 && distance<20)
  {
    lcd.setCursor(0,1);lcd.print("BIN 33% Empty");delay(1000);
  }
 else if(distance>20 && distance<50)
  {
    lcd.setCursor(0,1);lcd.print("BIN 66% Empty");delay(1000);
  }
 else if(distance>50)
  {
    lcd.setCursor(0,1);lcd.print("BIN Empty");delay(1000);
  }
}
```

Result







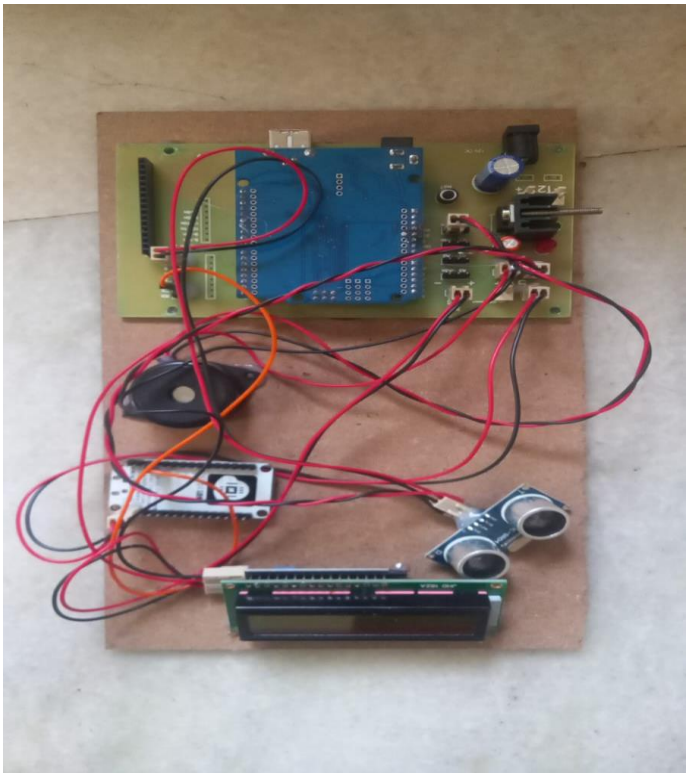


Fig. 10. Results

Testing:

For testing first connect the circuit to the power supply is given to the Arduino using computer and it can be done by using battery. In this way the whole testing circuit is built. Now we give input to the HC-SR04 by changing the level of solid garbage.. Change in garbage levels should be mailed. Summary of testing procedure:- 1) Connect the circuit according to the diagram 2) Give power to the system. 3) Vary garbage level for the ultrasonic sensor to give output. 4) Get the output on the LCD and email.

Conclusion:

We built an efficient garbage monitoring system which can be used to monitor the level of garbage in the dump. This data can be further used to plan garbage collection trips more efficiently, ultimately reducing overflowing bins and helping have better public sanitation.

Advantages:

- Very simple circuit.
- The HCSR04 sensor is very rugged.
- Helps monitor garbage levels.
- Uses very small amount of electricity.
- Ultimately helps in better planning of garbage pickups.
- Can help in reducing overflowing bins.
- Reduces trips to areas where the bins still have a lot of capacity.

Disadvantages:

- Cannot detect liquid waste.
- Only detects the top of the garbage level. It wouldn't realize if there is space left.

References:

[1] "Go Smart California," [Online]. Available: http://www.gosmartcalifornia.ca.gov/about/garbage/california.php. [Accessed 27 September 2018].

[2] R. Bhandari, "Electrification using smart garbage systems in Nepal," ELSEVIER, pp. 458-465, 2011.

[3] J. Bartlett, "Arise Energy Solutions," Arise Energy Solutions, LLC, [Online]. Available: http://ariseenergy.com/training-education/history-of-pv-smart-energy. [Accessed 28 September 2018].

[4] O. R. Otieno, "SMART GARBAGE MONITORING," University of Nairobi, 2009.

[5] "Energy.gov," [Online]. Available: https://www.energy.gov/energysaver/smart-dustbins. [Accessed 15 Nov 2018].

[6] "Economic Survey 2017/18," GoN, Ministry of Finance, June 2018. [Online]. Available: http://mof.gov.np/en/archive-documents/economic-survey-21.html?lang. [Accessed 13 10 2018].

[7] "Smart Garbage Monotoring Laboratory," [Online]. Available: http://solardat.uoregon.edu/dustBinProgram.html. [Accessed 20 Dec 2018].