|  |
| --- |
|  |
| SMART WASTE MANAGEMENT |
| USING IBM WATSON SERVICES |
|  |
|  |
| **6/21/2019** |

|  |
| --- |
|  |

**COLLEGE:MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**BRANCH:ELECTRONICS AND COMMUNICATION ENGINEERING(ECE)**

**TEAM NAME:IOTIANS**

**TEAM MEMBERS:**

SMART WASTE MANAGEMENT USING IBM WATSON SERVICES

**ABSTRACT:**

The launch of the “Digital India Program” of the Government aims at transforming India into digital empowered society and knowledge economy and towards this IOT would play a very vital role. One of the key aspects of Smart City is Smart Garbage Management which involves monitoring Garbage in city towards removal appropriately and also disposing off.

Now-a-days we are experiencing the problem of garbage which gets accumulated in cities and societies. There is no proper supervision from Corporation towards monitoring the activity of Garbage collection. Mostly whenever we see any vacant plot, people throw the garbage and as such no one is there to manage which obviously increases day by day leading to health issues, unclean city and also creating problems during rainy season So, with the upcoming Internet of Things (IOT) Technology and digital India initiative, we are making a simple step towards developing an “IOT based Smart Garbage Management System” prototype which will help the people and the society and the country as whole in proper management of the garbage and time to time collection and deposition of the waste and garbage which will help in making a clean and smart city.

In additional, statistical analysis would be performed towards predicting the rate of bin getting filled up towards getting cleaned for proper management. So towards this, the management system is developed for easy handling, supervision and monitoring of garbage in the societies and there is no need for the person to keep track of the garbage bins filled and no need to inform the Corporation

**INTRODUCTION:**

Waste management is all the activities and actions required to manage waste from its inception to its final disposal [1]. This includes collection, transportation, treatment and disposal of waste together with monitoring and regulation. Waste collection methods vary widely among different countries and regions. Domestic waste collection services are often provided by local government authorities.

Curbside collection is the most common method of disposal in most countries, in which waste is collected at regular intervals by specialized trucks. Waste collected is then transported to an appropriate disposal area. Nowadays, cities with developing economies experience exhausted waste collection services, inadequately managed and uncontrolled dumpsites and the problems are worsening [2]. Waste collection method in such countries is an on-going challenge and many struggle due to weak institutions and rapid urbanization.

The garbage containers transmit signals to indicate that they are over 80% or 90% full and should be emptied. Via the mobile communications network, the signals are sent to a web based software application used by the waste management company. In the software, the capacity of the container is indicated, which is taken as a basis to plan the best route for waste collection garbage trucks travel only to those containers that actually need to be emptied. A robust ultrasonic sensor is installed in the garbage container and detects the fill level regardless of what has been deposited inside.

The whole system contains ULTRASONIC SENSOR, ARDUINO BOARD, GSM MODULE, BREAD BOARD, POWER SUPPLY (BATTERY). The sensor is fixed on to the bread board. the connection between the arduino board and sensor is made with the help of connecting wires. The working program is fed into the arduino board. The gsm module is also connected to the same arduino board with the help of wires. The power supply to the system is given with the help of a battery.

**HARDWARE COMPONENTS REQUIRED:**

* NODE MCU
* ULTRASONIC SENSOR
* DHT11 SENSOR
* SERVO MOTOR
* LEDS
* JUMP WIRES

**SOFTWARE COMPONENTS REQUIRED:**

* IBM CLOUD
* AURDINO IDE
* MIT APP INVENTER
* **NODE MCU:**

 The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. **NodeMCU** is an open source iot platform. It includes firmware which runs on the ESP8266 wi-Fi SoCcfrom 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. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1).

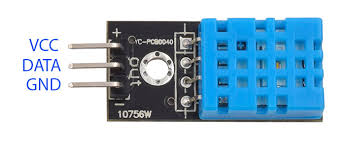
* **ULTRASONIC SENSOR:**

A special sonic transducer is used for the ultrasonic proximity sensors, which allows for alternate transmission and reception of sound waves. The sonic waves emitted by the transducer are reflected by an object and received back in the transducer. After having emitted the sound waves, the ultrasonic sensor will switch to receive mode. The time elapsed between emitting and receiving is proportional to the distance of the object from the sensor

*  **WORKING OF ULTRASONIC SENSOR**:

Ultrasonic sensors generate high-frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object

* **DHT11 SENSOR:**

**** The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly  simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

* **SERVO MOTOR:**

****

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc

* **BASIC SHIELD:**

This Basic Shield can be interface with 5V or 3.3V logic microcontroller boards like Arduino , AVR, PIC, 8051, ARM etc. Basic Shield is very popular shield for interfacing of electronics component with microcontroller like LED’s, variable resistor, Push Button, LDR, buzzer etc. All components are arranged in a proper manner so that you can use it with your microcontroller to learn basic programming of microcontroller in your projects. This shield helps you how to deal with basic electronics component in your project.

* **JUMP WIRES:**

 Jump wires (also called jumper wires) for solderless breadboarding can be obtained in ready-to-use jump wire sets or can be manually manufactured. The latter can become tedious work for larger circuits. Readyto-use jump wires come in different qualities, some even with tiny plugs attached to the wire ends. Jump wire material for ready-made or homemade wires should usually be 22 AWG (0.33 mm2 ) solid copper, tin-plated wire - assuming no tiny plugs are to be attached to the wire ends. The wire ends should be stripped 3 ⁄16 to 5 ⁄16 in (4.8 to 7.9 mm). Shorter stripped wires might result in bad contact with the board's spring clips (insulation being caught in the springs). Longer stripped wires increase the likelihood of short-circuits on the board. Needle-nose pliers and tweezers are helpful when inserting or removing wires, particularly on crowded boards.

**BLOCK DIAGRAM:**

**NODE MCU**

**IBM CLOUD**

**DHT 11 SENSOR**

**WEB APPLIACTION**

**MOBILE**

**ULTRASONIC SENSOR**

**WORKING :**

The garbage containers transmit signals to indicate that they are over 80% or 90% full and should be emptied. Via the mobile communications network, the signals are sent to a web based software application used by the waste management company. In the software, the capacity of the container is indicated, which is taken as a basis to plan the best route for waste collection garbage trucks travel only to those containers that actually need to be emptied.

A robust ultrasonic sensor is installed in the garbage container and detects the fill level regardless of what has been deposited inside. The whole system contains ULTRASONIC SENSOR, NODEMCU, SERVO MOTOR, DHT11,POWER SUPPLY. The sensor is fixed on to the node mcu. the connection between the NODE MCU board and sensor is made with the help of connecting wires. The working program is fed into the NODE MCU board.

The SERVO MOTOR is also connected to the same NODE MCU board with the help of wires. The power supply to the system is given with the help of a battery.

**ADVANNTAGES OF SMART WASTE MANAGEMENT SYSTEM:-**

* Less time and fuel consumption as the trucks go only to the filled containers.
* Decreased noise, traffic flow and air pollution as a result of less trucks on the roads.ϖ Our smart operating system enable two way communication between the dustbinϖ deployed in the city and service operator.
* Therefore the focus is only on collection of route based fill level of the containers. The sensors installed in the containers provide real time information on the fill level.ϖ This information helps determine when and where to prioritise collection.
* In this way both service providers and citizens benefit from an optimized systemϖ which results in major cost savings and less urban pollution. Reduces the infrastructure (trucks, containers), operating (fuel) and maintenanceϖ costs of the service by upto 30%. Applying this technology to the city optimises management, resources and costs, andϖ makes it a “SMART CITY”.
* Historical information on collections helps adapt the deployment of containers toϖ the actual needs of the city, therefore reducing the number of containers that clutter up the road and increasing public parking spaces.
* It keeps the surroundings clean and green, free from bad odour of wastes,ϖ emphasizes on healthy environment and keep cities more beautiful. Reducing manpower required to handle the garbage collection.ϖ

**Software program:-**

#include <Servo.h>

#include "DHT.h"

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#define DHTPIN D3 // what pin we're connected to

#define DHTTYPE DHT11 // define type of sensor DHT 11

#define ORG "ck3513"

#define DEVICE\_TYPE "embedded"

#define DEVICE\_ID "nodemcu"

#define TOKEN "test12345678"

const char\* ssid = "pulsation 4";

const char\* password = "pulsation@iiith";

String command;

const int trigpin= D8;

//const int trigpin2= D8;

const int echopin= D7;

//const int echopin2= D4;

const int led1= D1;

//const int led2= D6;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/evt/Data/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883,wifiClient);

DHT dht (DHTPIN, DHTTYPE);

Servo myservo;

int duration;

int distance;

void setup(){

pinMode(trigpin,OUTPUT);

// pinMode(trigpin2,OUTPUT);

pinMode(led1,OUTPUT);

// pinMode(led2,OUTPUT);

pinMode(echopin,INPUT);

//pinMode(echopin2,INPUT);

Serial.begin(9600);

myservo.attach(D2);

Serial.begin(115200);

Serial.println();

dht.begin();

Serial.print("Connecting to ");

Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.print("WiFi connected, IP address: ");

Serial.println(WiFi.localIP());

}

// #define DHTPIN D4 // what pin we're connected to

//#define DHTTYPE DHT11 // define type of sensor DHT 11

void loop(){

digitalWrite(trigpin,HIGH);

delay(1000);

digitalWrite(trigpin,LOW);

// digitalWrite(trigpin2,HIGH);

// digitalWrite(trigpin2,LOW);

duration=pulseIn(echopin,HIGH);

// duration=pulseIn(echopin2,HIGH);

distance = duration\*0.034/2;

Serial.println(distance);

float h = dht.readHumidity();

float t = dht.readTemperature();

Serial.print("humidity : ");

Serial.print(h);

Serial.print("%");

Serial.print(" temperature : ");

Serial.print(t);

Serial.println("\*C");

int pos;

if(distance<=3) // fill

{

Serial.println("dustbin is full");

digitalWrite(led1,LOW);

myservo.write(90); // tell servo to go to position in variable 'pos'

delay(10);

}// waits 15ms for the servo to reach the position

else{

Serial.println("dustbin is Empty");

digitalWrite(led1,LOW);

myservo.write(90); // tell servo to go to position in variable 'pos'

}

PublishData(t,h,distance);

if (!client.loop()) {

mqttConnect();

}

delay(100);

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() {

if (client.subscribe(topic)) {

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void PublishData(float temp, float humid,int distance){

if (!!!client.connected()) {

Serial.print("Reconnecting client to ");

Serial.println(server);

while (!!!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

Serial.println();

}

String payload = "{\"d\":{\"temperature\":";

payload += temp;

payload+="," "\"humidity\":";

payload += humid;

payload+="," "\"distance\":";

payload += distance;

payload += "}}";

Serial.print("Sending payload: ");

Serial.println(payload);

if (client.publish(topic, (char\*) payload.c\_str())) {

Serial.println("Publish ok");

} else {

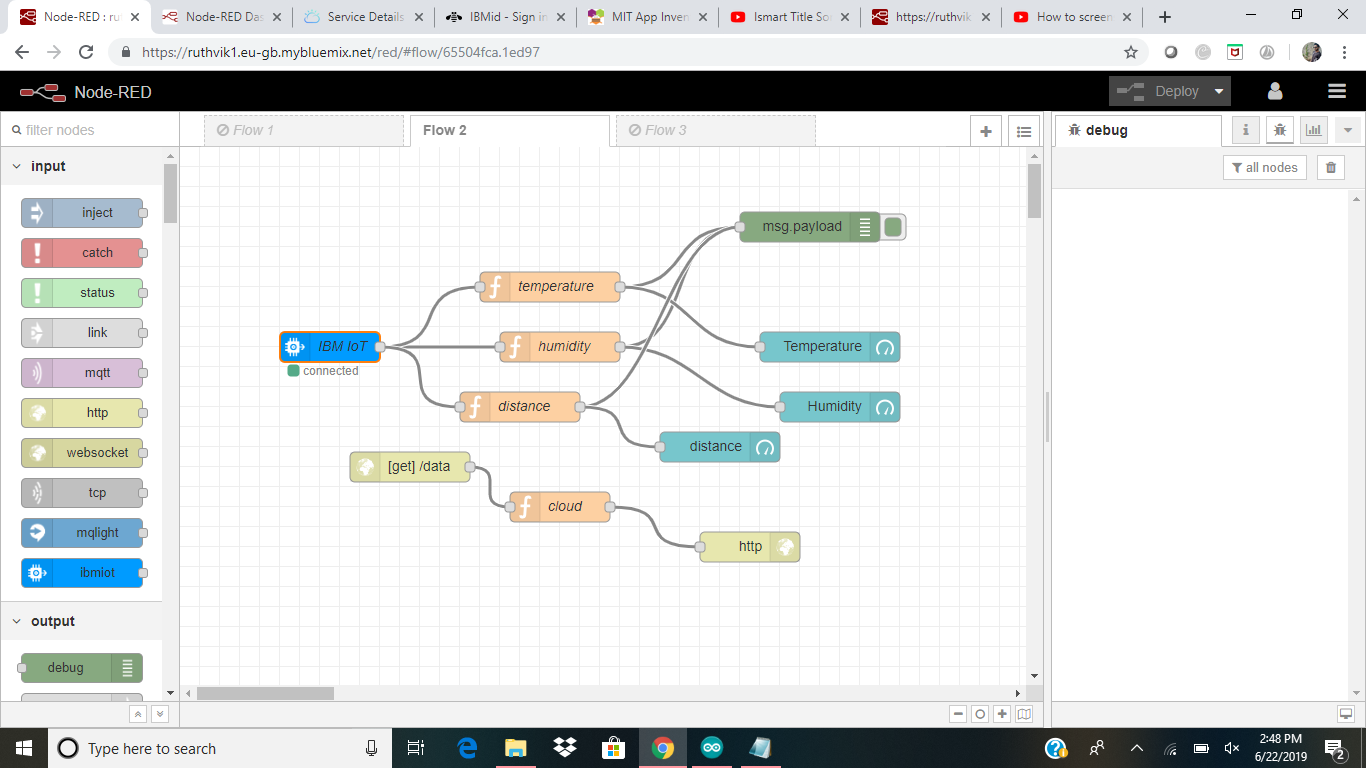
Serial.println("Publish failed");

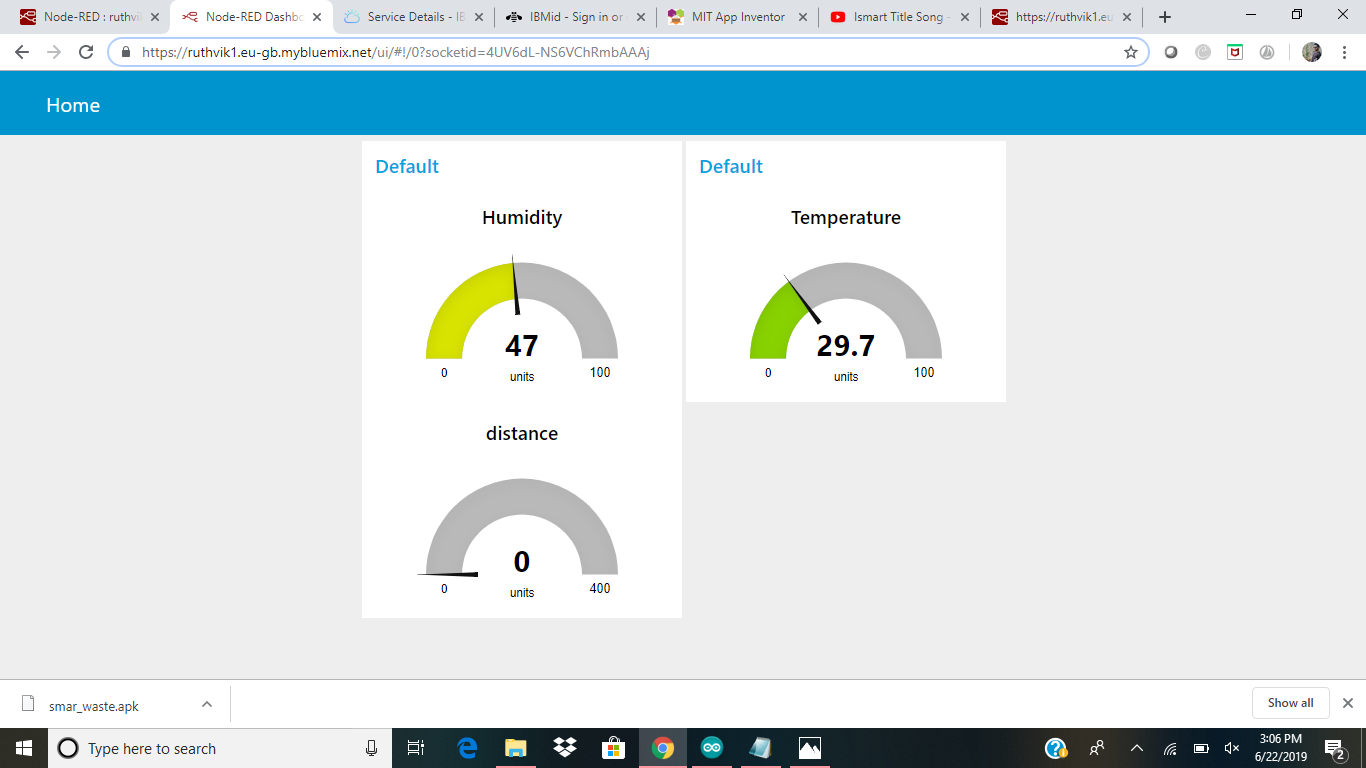
}

}

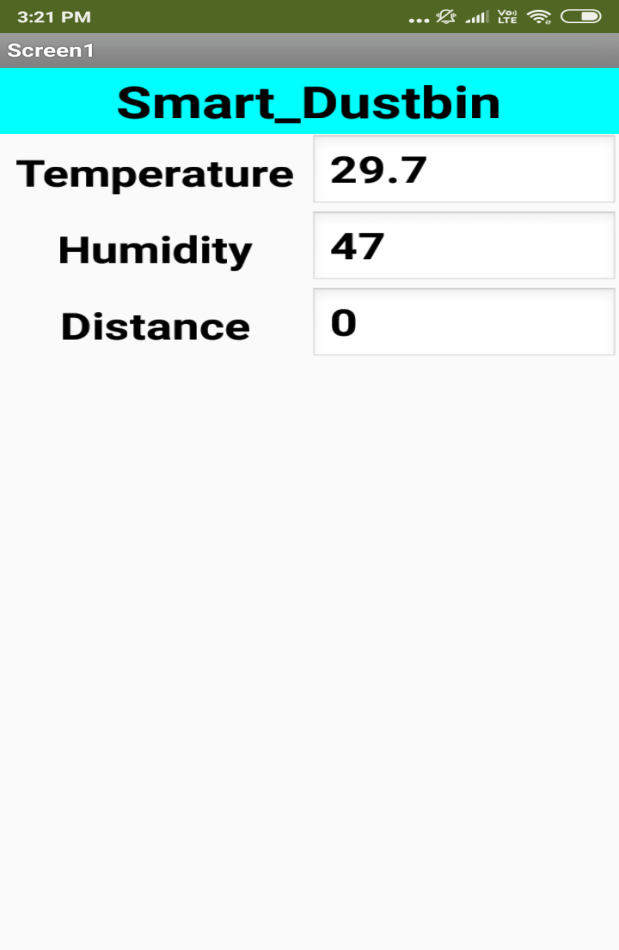
**NEED FOR IMPROVEMENT IN WASTE MANAGEMENT SYSTEM:-**

* By 2030, almost two-third of the world’s population will be living in cities. This factϖ requires the development of sustainable solutions for urban life, managing waste is a key issue for the health.
* Efficient and energy-saving waste management, reducing CO2,air pollution andϖ vehicle exhaust emissions—these are just a few examples for the demands of future cities
* . In views of that, the efficient use and responsible handling of resources become more important. Effectively managing waste is important in developed countries. Waste managementϖ may swallow upto 50% of a city’s budget, but only serve a small part of the population.
* Sometimes, upto 60%of waste is not being collected, it is often simply burned by theϖ roadside. It can pollute drinking water, it can spread disease to people living nearby.
* Even with great route optimization, the worker must still physically go to the dustbinϖ to check waste levels. Because of this, trucks often visit containers that do not need emptying, which wastes both time and fuel.
* Waste management prevents harm to human health and the environment byϖ reducing the volume and hazardous character of residential and industrial waste

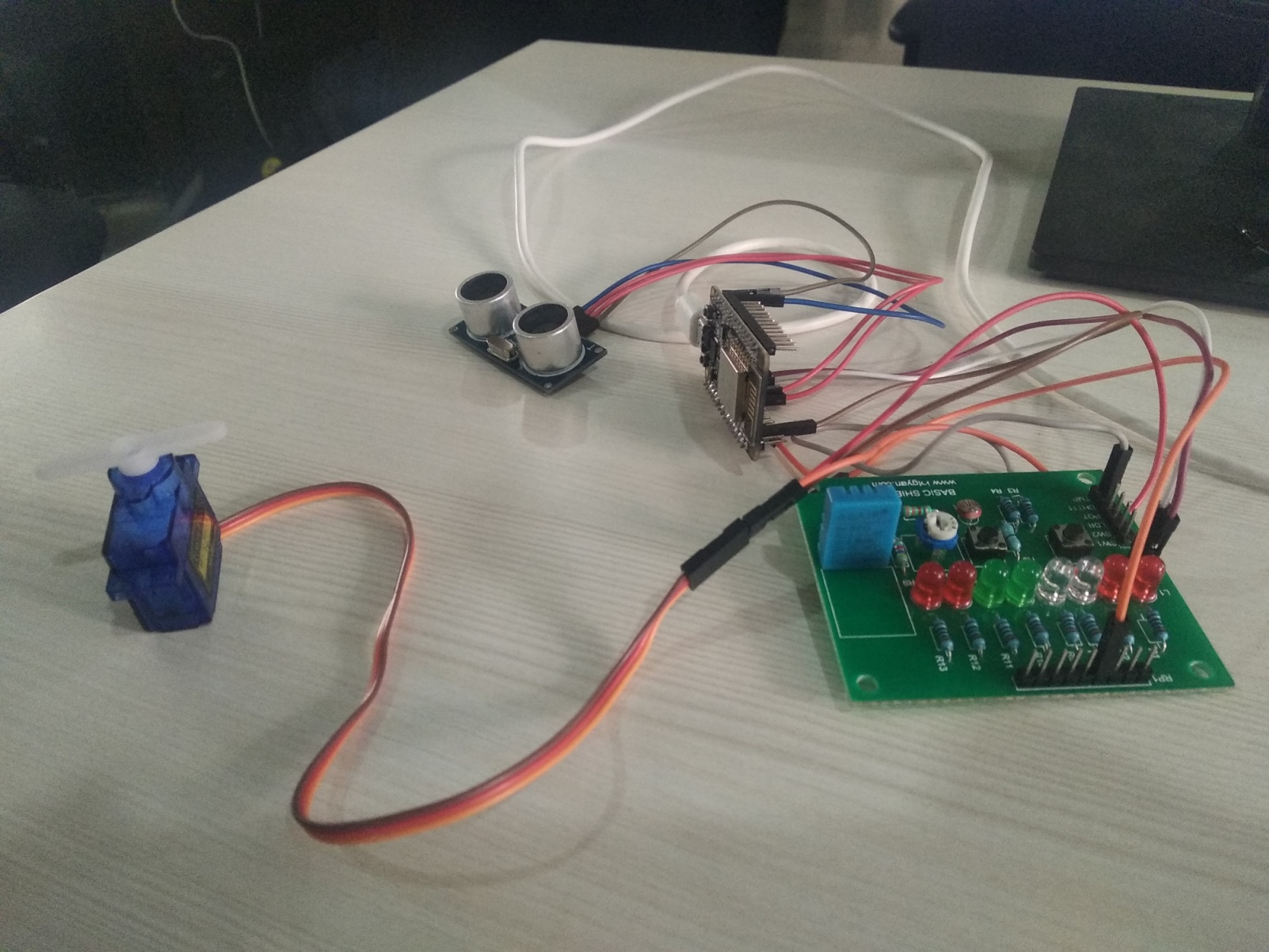




**OUTPUT ON SCREEN:-**



**HARDWARE SETUP:-**

****

**CONCLUSION :-**

* By using this method the collection of waste in the city becomes easier. It helps in reducing air pollution, traffic flow, man power, time and money.
* With the help of proper technology (GPS & SOFTWARE APPLICATIONS) we can guide the trucks in selecting the shortest path for garbage collection.
* This project can add an edge to the cities aiming to get smart and people-friendly