**Project Requirement and Specification**

**On**

**Smart Dustbin- Garbage Management**

**(CSE IV Semester Mini project)**

**2021-2022**

****

Submitted to: Submitted by:

**Department of Computer                Dhruv Singhal**

**application for the**

**degree of BTECH**

**Roll. No.: 2014388**

**Sem-4th**

Sec- ‘H’

Btech (CSE)

**----------------------------Contents----------------------------**

* **About project**
* **Equipments requirement**
* **Code used in making project**
* **Block Diagram**
* **Reference**

1. **About Project**

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.

**It’s Features-**

❖ The smart, sensor based dustbin will judge the level of waste in it and send the message directly to the municipal corporation.

❖ It can sense all the type of waste material either it is in the form of solid or liquid.

❖ According to the filled level of the dustbin, the vehicles from the municipal corporation will choose the shortest path with the help of the “TRANSPORTATION SOFTWARE”, which will save their time.

❖ It emphasizes on “DIGITAL INDIA”.

❖ The system is simple. If there is any problem with any equipment in the future, that part is easily replaceable with new one without any difficulty and delay.

1. **Equipment’s required-**
2. GARBAGE CONTAINER
3. ULTRASONIC SENSOR( to sense level by sound wave)
4. ARDUINO BOARD - helps to operate sensor and gsm
5. GSM MODULE
6. BREAD BOARD
7. JUMP WIRES
8. **Code used in making of project**
   1. **PROGRAM FOR WASTE LEVEL SENSING**

#define trigPin 12 //sensor trigpin connected to arduino pin 12

#define echoPin 13 // sensor echopin connected to arduino pin 13

void setup()

{

Serial.begin (9600); // to exchange messages at a rate of 9600 bits per second (speed of

//data sharing from arduino)

pinMode(trigPin, OUTPUT); //sets trigpin as output

pinMode(echoPin, INPUT); // sets echopin as input

}

void loop()

{

long duration;// variable for the duration of sound wave travel

long distance; //variable for distance measurement

int max = 80; // Let consider as Height of the Garbage Bin is = 80 cm.

float diff, perc;

digitalWrite(trigPin, LOW);

delayMicroseconds(2); //sets the trigpin low for 2 seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10); //sets the trigpin high for 10 seconds

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); //read the echopin,return soundwave travel time

distance = (duration/2) / 29.1;

diff = max - distance; // 'diff' variable tells u that, how much the Garbage Bin is Left

to fill.

perc = (diff/max)\*100;

// 'perc' variable tells u that, how much percentage the

Garbage Bin is filled.

if (perc>=90)

{

Serial.println("Garbage Bin is FULL."); //display the message on monitor

// When the Garbage Bin

is filled more than 90%, then this Error Message will Displayed.

}

else

{

Serial.print("Garbage Bin is Filled ");

Serial.print(perc);

Serial.println(" %.");

// These 3 Lines are print, that how

much the Garbage Bin is Filled...Ex. "Garbage Bin is Filled 70%.".

}

/\*

if (distance >= 400 || distance <= 2)

{

Serial.println("Out of range");

}

else

{

Serial.print(distance);

Serial.println(" cm");

}

\*/

delay(500); // to stop program for half second

}

* 1. **PROGRAM FOR MESSAGE SENDING**

SoftwareSerial mySerial(9, 10);

#define trigPin 12

#define echoPin 13

void setup()

{

mySerial.begin(9600); // Setting the baud rate of GSM Module

Serial.begin (9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

delay(100);

}

void loop()

{

long duration, distance;

int max = 80; // Let consider as Height of the Garbage Bin is = 80 cm.

float diff, perc;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1;

diff = max - distance; // 'diff' variable tells u that, how much the Garbage Bin is Left to fill.

perc = (diff/max)\*100;

// 'perc' variable tells u that, how much percentage the

Garbage Bin is filled.

if (perc>=90)

{

//Serial.println("Garbage Bin is FULL.");

// When the Garbage Bin is filled more

than 90%, then this Error Message will Displayed.

// Call the Function of Send SMS.

SendMessage();

// Send Message Function Call.

}

/\*

else

{

Serial.print("Garbage Bin is Filled ");

Serial.print(perc);

Serial.print(" %.");

// These 3 Lines are print, that how much the Garbage

Bin is Filled...Ex. "Garbage Bin is Filled 70%.".

}

\*/

/\*

if (distance >= 400 || distance <= 2)

{

Serial.println("Out of range");

}

else

{

Serial.print(distance);

Serial.println(" cm");

}\*/

delay(500);

}

void SendMessage() //to send message

{

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+918792574742\"\r"); // Replace x with mobile number

delay(1000);

mySerial.println("Garbage Bin is Full."); // The SMS that we want to send

delay(100);

mySerial.println((char)26); // ASCII code of CTRL+Z

delay(1000);

}

* 1. **PROGRAM FOR SHORT ROUTE OPTIMIZATION**

#include<stdio.h>

#include<stdlib.h //memory allocation

#define infinity 999

int nd,n=26,v,a[50];

float dist[50];

float

cost[26][26]={999,3.55,4.35,5.85,7.3,7.5,9.45,7.45,6.05,3.45,4.65,6.35,8.8,8.95,10.1,9.55,9.1

5,8.1,6.85,3.3,2.2,3.4,3.7,5.65,6.35,5.1,3.55,999,8.35,2.3,3.7,4.25,5.95,3.75,2.55,2.1,3.33,3.

55,5.1,5.75,6.9,6.75,6.35,5.3,4.05,4.8,3.7,2.7,2.35,4.3,5,4.4,4.35,0.85,999,1.5,2.95,3.4,5.1,2.

9,1.7,2.9,4.1,2.7,4.25,4.9,6.05,5.9,5.5,4.45,3.2,5.6,4.5,3.5,3.15,5.1,5.8,5.45,5.85,3.2,1.5,999,

1.45,1.9,3.6,1.6,2.8,4.4,5.4,3.7,2.95,3.6,4.75,6.9,6.5,5.45,4.2,7.1,6,5,4.65,6.6,7.3,6.95,7.3,3.

7,2.95,1.45,999,3.35,5.05,3.05,4.25,5.85,6.85,5.15,4.4,5.05,5.2,8.33,7.95,6.9,5.65,8.55,7.45,

6.45,6.1,8.05,8.75,8.4,7.5,4.25,3.4,1.9,3.35,999,1.7,3.5,4.7,6.3,7.3,5.6,3.75,3.8,4.95,6.35,6.7

5,7.35,6.1,9,7.9,6.9,6.55,8.5,9.2,8.85,9.45,5.95,5.1,3.6,5.05,1.7,999,3.4,4.6,8,7,5.3,2.05,2.7,

2.85,5.25,5.65,9.05,7.8,10.7,9.6,8.6,8.25,10.2,10.9,10.55,7.45,3.75,2.9,1.6,3.05,3.5,3.4,999,

1.2,5,3.8,2.1,1.35,2,3.15,4.55,4.9,3.85,2.6,8.5,7.4,6.4,6.05,7.2,7.9,8.35,6.05,2.55,1.7,2.8,4.2

5,4.7,4.6,1.2,999,3.8,2.6,0.9,2.55,3.2,4.35,4.1,3.7,2.65,1.4,7.3,6.2,5.2,4.35,6,6.7,7.15,3.45,2.

1,2.9,4.4,5.85,6.3,8,5,3.8,999,1.2,2.9,6.15,6.8,7.95,6.1,5.7,4.65,3.4,4.7,3.6,2.6,2.25,2.2,2.9,4

.55,4.65,3.3,4.1,5.4,6.85,7.3,7,3.8,2.6,1.2,999,1.7,4.95,4.3,5.45,4.9,4.5,3.45,2.2,5.9,4.8,3.8,3

.45,3.4,4.1,5.75,6.35,3.55,2.7,3.7,5.15,5.6,5.3,2.1,0.9,2.9,1.7,999,3.25,2.6,3.75,3.2,2.8,1.75,

0.5,7.6,6.5,5.5,5.15,5.1,5.8,7.45,8.8,5.1,4.25,2.95,4.4,3.75,2.05,1.35,2.55,6.15,4.95,3.25,999

,0.65,1.8,3.2,3.6,4.65,3.75,9.85,8.75,7.75,6.9,8.35,9.05,9.7,8.95,5.75,4.9,3.6,5.05,3.8,2.7,2,3.2,6.8,4.3,2.6,0.65,999,1.15,2.55,2.95,4,3.1,10.2,9.1,8.1,7.75,7.7,8.4,10.5,10.1,6.9,6.05,4.75,

6.2,4.95,3.85,3.15,4.35,7.95,5.45,3.75,1.8,1.15,999,1.4,1.8,2.85,4.1,11.35,10.25,9.25,8.9,8.8

5,9.26,11.2,9.55,6.75,5.9,6.9,8.35,6.35,5.25,4.5,4.1,6.1,4.9,3.2,3.2,2.55,1.4,999,0.4,1.45,2.7,

10.8,9.7,8.7,8.35,8.3,9,10.65,9.15,6.35,5.5,6.5,7.95,6.75,5.65,4.9,3.7,5.7,4.5,2.8,3.6,2.95,1.8

,0.4,999,1.05,2.3,10.4,9.3,8.3,7.75,7.9,8.6,10.25,8.1,5.3,4.45,5.45,6.9,7.35,9.05,8.85,2.65,4.

65,3.45,1.75,4.65,4,2.85,1.45,1.05,999,1.25,9.35,8.25,7.25,6.9,6.85,7.55,9.2,6.85,4.05,3.2,4.

2,5.65,6.1,7.8,2.6,1.4,3.4,2.2,0.5,3.75,3.1,4.1,2.7,2.3,1.25,999,8.1,7,6,5.65,5.6,6.3,7.95,3.3,4

.8,5.6,7.1,8.55,9,10.7,8.5,7.3,4.7,5.9,7.6,9.85,10.2,11.35,10.8,10.4,9.35,8.1,999,1.1,2.3,2.8,4

.8,5.5,4,2.2,3.7,4.5,6,7.45,7.9,9.6,7.4,6.2,3.6,4.8,6.5,8.75,9.1,10.25,9.7,9.3,8.25,7,1.1,999,1.

2,1.7,3.7,4.4,2.9,3.4,2.7,3.5,5,6.45,6.9,8.6,6.4,5.2,2.6,3.8,5.5,7.75,8.1,9.25,8.7,8.3,7.25,6,2.3

,1.2,999,0.5,2.5,3.2,1.7,3.7,2.35,3.15,4.65,6.1,6.55,8.25,6.05,4.35,2.25,3.45,5.15,6.9,7.75,8.

9,8.35,7.95,6.9,6.65,2.8,1.7,0.5,999,2,2.7,2.3,5.65,4.3,5.1,6.6,8.05,8.5,10.2,7.2,6,2.2,3.4,5.1,

8.35,7.7,8.85,8.3,7.9,6.85,5.6,4.8,3.7,2.5,2,999,0.7,3.2,6.35,5,5.8,7.3,8.75,9.2,10.9,7.9,6.7,2.

9,4.1,5.8,9.05,8.4,9.26,9,8.6,7.55,6.3,5.5,4.4,3.2,2.7,0.7,999,2.5,5.1,4.4,5.45,6.95,8.4,8.85,1

0.55,8.35,7.15,4.55,5.75,7.45,9.7,10.05,11.2,10.65,10.25,9.2,7.45,4,2.9,1.7,2.3,3.2,2.5,999};

void dij()

{

int i,u,count,w,flag[26];

float min;

for(i=0;i<n;i++)

{

flag[i]=0;

dist[i]=cost[v][i];

}

count=2;

while(count<=n)

{

min=99;

for(w=0;w<n;w++)

if(dist[w]<min && !flag[w])

{

min=dist[w];

u=w;

}

flag[u]=1;

count++;

for(w=0;w<n;w++)

if((dist[u]+cost[u][w]<dist[w]) && !flag[w])

dist[w]=dist[u]+cost[u][w];

}

for(i=0;i<n;i++)

if(i==v)

dist[i]=999;}

int sort()

{

int i,j,flag,temp;

for(i=0;i<nd;i++)

for(j=0;j<nd-i-1;j++)

{

if(dist[a[j+1]]<dist[a[j]])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

flag=a[0];

for(i=0;i<nd;i++)

a[i]=a[i+1];

nd--;

printf("%d ",flag);

return flag;

}

void main()

{

int count,i,j;

printf("\n Enter the source vertex: \n");

scanf("%d",&v);

printf("\n Enter the number of active nodes: \n");

scanf("%d",&count);

printf("\n Enter the active nodes \n");

for(i=0;i<count;i++)

{

scanf("%d",&a[i]);

}

nd=count;

for(i=0;i<count-1;i++)

{

dij(n,v);

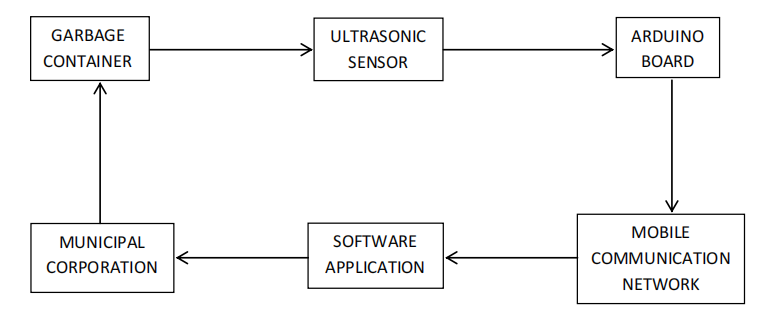
v=sort();

}

printf("%d",a[0]);

}

**4.Block Diagram-**



1. **References-**
2. <https://iopscience.iop.org/article/10.1088/1757-899X/263/4/042027/pdf>
3. <https://www.instructables.com/Smart-Garbage-Monitoring-System-Using-Internet-of-/>
4. <https://www.electronicshub.org/smart-dustbin-using-arduino/>
5. *United Nations Environmental Programme (2013). "Guidelines for National Waste*

*Management Strategies Moving from Challenges to Opportunities."*