SMART DUSTBIN FOR HOSPITAL WASTE MANAGEMENT

APPENDIX-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 Bin is = 80cm.

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;

perc = (diff/max)*100;

if (perc>=90)

{

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

SendMessage();

}
```

```
/*
else
{
 Serial.print("GarbageBin isFilled ");

 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()

{

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=\"+919791946938\"\r"); // Replace x with
mobile number delay(1000);

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

delay(100);

mySerial.println((char)26);// ASCII code of CTRL+Z
delay(1000);

}
```

PROGRAM FOR WASTE LEVEL SENSING

```
#define trigPin 12

#define echoPin 13

 void setup()

{

Serial.begin (9600);

pinMode(trigPin, OUTPUT);
```

```
pinMode(echoPin, INPUT);
}
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;
perc = (diff/max)*100;
if (perc>=90)
{
```

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

}
```

PROGRAM FOR GAS SENSOR DETECTION

```
#include <SoftwareSerial.h>

SoftwareSerial SIM900(7, 8); // gsm module connected here

String textForSMS;

int smokeS = A1; // smoke / gas  sensor connected with analog pin A1 of the arduino / mega.

int data = 0;

void setup() {

randomSeed(analogRead(0));

Serial.begin(9600);

SIM900.begin(9600); // for sim900D 19200. while enter 9600 for sim900A

pinMode(smokeS, INPUT);

}

void loop() {

data = analogRead(smokeS);

Serial.print("Smoke: ");

Serial.println(data);

 if ( data > 230) //
```

```
{
textForSMS =  "\nGas Or Smoke Detected";
sendSMS(textForSMS);
Serial.println(textForSMS);
Serial.println("message sent.");
delay(5000);
while(1)
{
}
//
 }
}
void sendSMS(String message)
{
SIM900.print("AT+CMGF=1\r");                    // AT command to send SMS message
delay(1000);
```

```
SIM900.println("AT + CMGS = \"+923171956677\"");  // recipient's mobile number, in international format

delay(1000);

SIM900.println(message);                    // message to send

delay(1000);

SIM900.println((char)26);                   // End AT command with a ^Z, ASCII code 26

delay(1000);

SIM900.println();

delay(100);                          // give module time to send SMS

 // SIM900power();                          // turn off module

}
```

GSM code to send Message to use when it is filled:

```
#include "SIM900.h"

#include “inetGSM.h”

#include “sms.h”

SMSGSM sms;

int  numdata;
```

```
int value;

boolean started =false;

char smsbuffer[160];

char n[20];

int PIN =A0;

int sensor =450;

int LED1=11;

void setup()

{

pinMode(LED1 , OUTPUT);

pinMode(PIN, INPUT);

value =analogRead(PIN);

serial.begin(9600);

Serial.println(value);

Serial.println("GSM Shield testing.");

if (value > sensor){

digitalWrite(LED1, HIGH);
```

```
}

if (gsm.begin(2400))

{

Serial.println("\nstatus=READY");

started=true;

}

else

serial.println (“\nstatus =IDLE”);

if  (value>sensor)

{

digitalWrite(LED1, HIGH);

sms.SendSMS("+917530001174", "RJ27 CB 3**9 Pollution Level
Exceeding , Attention Required "));

Serial.println("\nSMS sent OK");

}

};

Void loop()

{

if(started)
```

```
{

if(gsm.readSMS(smsbuffer, 160, n, 20))

{
Serial.println(n);

Serial.println(smsbuffer);

}

}

};
```

APPENDIX- 2

INTEGRATED CODE FOR SENSOR BASED SMARTDUSTBIN FOR HOSPITAL WASTE MANAEGENENT

```
#include "SIM900.h"

#include <SoftwareSerial.h> //

#include "inetGSM.h"

#include "sms.h"

SMSGSM sms;

int numdata;

int value;

boolean started=false;

char smbuffer[160];

char n[20];

int PIN =A0;

int sensor=450;
```

```
#include <Servo.h>

Servo myservo;

int pos = 0;

int calibrationTime = 30;      //the time when the sensor outputs a low impulse

long unsigned int lowIn;  //the amount of milliseconds the sensor
has to be low

long unsigned int pause = 5000;

boolean lockLow =true;

boolean takeLowTime;

int pirPin = 12;              //digital pin connected to the PIR's output

int pirPos = 13;             //connects to the PIR's 5V pin

void setup(){

pinMode(LED1, OUTPUT);

cvalue = analogRead (PIN);

serial. Begin(9600);

Serial.println(value);
```

```
Serial.println("GSM Shield testing.");

if (value > sensor){

digitalWrite(LED1, HIGH);


}

if (gsm.begin(2400)){

Serial.println("\nstatus=READY");

started=true;

}

else

Serial.println("\nstatus=IDLE");

if (value> sensor)  {

digitalWrite(LED1, HIGH);
```

```
(sms.SendSMS("+917530001174", "RJ27 CB 3**9 Pollution Level Exceeding , Attention Required "));

Serial.println(“\n SMS sent OK”);

}

Serial.begin(9600);

pinMode(pirPin, INPUT);

pinMode(pirPos,OUTPUT);

digitalWrite(pirPos,HIGH);

Serial.println("calibrating sensor ");

for(int i = 0; i <calibrationTime; i++)

{
 Serial.print(calibrationTime - i);

 Serial.print("-");

 delay(1000);


  }

Serial.println();
```

```
Serial.println("done");

//while making this Instructable, I had some issues with the
PIR's output

//going HIGH immediately after calibrating

//this waits until the PIR's output is low before
ending setup

while (digitalRead(pirPin) == HIGH) {

delay(500);

serial.print(“.”);

}

Serial.print("SENSOR ACTIVE");

}

void loop(){

if(started){

if(gsm.readSMS(smsbuffer, 160, n, 20))

Serial.println(n);

Serial.println(smsbuffer);
```

```
}

}

if(digitalRead(pirPin) == HIGH){

/*turns servo from 0 to 180 degrees and back

it does this by increasing the variable "pos" by 1 every 5 milliseconds until it hits 180and setting the servo's position in degrees to "pos" every 5 milliseconds it then does it in reverse to have it go back to learn more about this, google "for loops"to change the amount of degrees the servo turns, change the number 180 to the number of degrees you want it to turn

**/
for(pos = 0; pos < 180; pos += 1)

{

myservo.write(pos);

delay(5);

}

for(pos = 180; pos>=1; pos-=1)

{

myservo.write(pos);
```

```
delay(5);
}
if(lockLow){
lockLow = false;
Serial.println("---");
Serial.print(“motion detectd at”);
delay(50);
}
takeLowTime = true;
}
if(digitalRead(pirPin) == LOW){
if(takeLowTime){
lowIn = millis();
takeLowTime = false;
```

```
}

//if the sensor is low for more than the given pause, //we can assume the motion has stopped if(!lockLow && millis() - lowIn > pause){//makes sure this block of code is only executed again after //a new motion sequence has been detected

lockLow = true;

Serial.print("motion ended at "); //output

Serial.print((millis() - pause)/1000);


Serial.print((millis() - pause)/1000);

delay(50);

}

}

}
```

OUTPUT

ARDUINO PLATFORM

sketch_apr05c | Arduino 1.8.12

File Edit Sketch Tools Help

sketch_apr05c

```
#include <Servo.h>   //servo library

//-----Wiring---
#define trigPin 12
#define echoPin 11
#define servoPin 7
//----Wiring---

Servo servo;

long duration, dist, average;
int flag=0;   //button flag
long aver[3];   //array for average

void setup() {
    servo.attach(servoPin);
    pinMode(4,INPUT);  //button is connected to D3 pin
    pinMode(trigPin, OUTPUT);  //Assign trigger pin  as output
    pinMode(echoPin, INPUT);  //Assign echo pin for input
    servo.write(90);          //open cap on power on
    delay(1000);
    servo.write(3);
}

void measure() {  //here is for range sensor
digitalWrite(trigPin, LOW);
delayMicroseconds(5);
digitalWrite(trigPin, HIGH);
delayMicroseconds(15);
```

CHEMICAL WASTE

GENERAL WASTE

ALERT MESSAGE