

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);
#include <SoftwareSerial.h>
SoftwareSerial esp8266(2,3);
#define trigPin1 6          // us1
#define echoPin1 7
#define buzzer 4

long duration, distance, sensor1;
float level=0;
#define DEBUG true
void setup()
{
  Serial.begin(9600);
  esp8266.begin(115200); // your esp's baud rate might be different
  lcd.begin(16, 2); //initializing LCD
  lcd.setCursor(0,0);
  lcd.print("Smart dustbin");
  lcd.setCursor(0,1);
  lcd.print("using iot");
  sendData("AT+RST\r\n",2000,DEBUG); // reset module
  sendData("AT+CWSAP=\"garbage level\", \"12345678\",5,3\r\n",2000,DEBUG);
  sendData("AT+CWMODE=2\r\n",1000,DEBUG); // configure as access point hotspot
  sendData("AT+CIFSR\r\n",1000,DEBUG); // get ip address
  sendData("AT+CIPMUX=1\r\n",1000,DEBUG); // configure for multiple connections
  sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // turn on server on port 80

  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(buzzer, OUTPUT);
}
void loop()
{
  ultrasensor(trigPin1, echoPin1);
  sensor1 = distance;
  level=((19-sensor1)*10)/1.80;
  level=level;
  Serial.print("sensor1=");
  Serial.println(level);
  delay(50);

  Serial.print("level=");
  Serial.println(level);
  lcd.clear();

  lcd.setCursor(3,0);
  lcd.print(" level ");
  lcd.setCursor(3,1);
  lcd.print(level);
  lcd.setCursor(9,1);
  lcd.print("% ");
  delay(50);
  if (level >70)
  {
    digitalWrite(buzzer,HIGH);
  }
  else
  {

```

```

    digitalWrite(buzzer, LOW);
}
counter++;
if((counter>200)&&(counter<400)){digitalWrite(led, HIGH);}
if(counter>400){counter==0;digitalWrite(led, LOW);}
    wifi();
}

void ultrasensor(int trigPin,int echoPin)
{
    digitalWrite(trigPin, LOW); // Added this line
    delayMicroseconds(2); // Added this line
    digitalWrite(trigPin, HIGH);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;
}

void wifi()
{
    if(esp8266.available()) // check if the esp is sending a message
    {
        if(esp8266.find("+IPD,"))
        {
            delay(500);
            // subtract 48 because the read() function returns
            //the ASCII decimal value and 0 (the first decimal number) starts at 48
            String webpage ;
            webpage = "HTTP/1.1 200 OK\nContent-Type: text/html\nConnection: close\nRefresh:
5\n\n<!DOCTYPE HTML>\n<html>\n\n";
            String cipSend = "AT+CIPSEND=";
            cipSend += connectionId;
            cipSend += ",";
            cipSend +=webpage.length();
            cipSend += "\r\n";
            sendData(cipSend,1000,DEBUG);
            sendData(webpage,1000,DEBUG);

            webpage ="<h1>=====dustbin monitoring system=====";
            webpage += "<h1> Current level(in %) = ";
            webpage += String(level);
            webpage += "<h1>";
            if (level >70)
            {
                webpage += "<h1> Dustbin is full ";
            }
            if (counter>200)
            {
                webpage += "<h1> Dustbin Alert ";
            }
            webpage += "<h1>";
            cipSend = "AT+CIPSEND=";
            cipSend += connectionId;
            cipSend += ",";
            cipSend +=webpage.length();
            cipSend += "\r\n";
            sendData(cipSend,1000,DEBUG);
            sendData(webpage,1000,DEBUG);

```

```

    String closeCommand = "AT+CIPCLOSE=";
    closeCommand+=connectionId; // append connection id
    closeCommand+="\r\n";
    sendData(closeCommand, 3000, DEBUG);
}
}
}

String sendData(String command, const int timeout, boolean debug)
{
    String response = "";
    esp8266.print(command); // send the read character to the esp8266
    long int time = millis();
    while( (time+timeout) > millis())
    {
        while(esp8266.available())
        {
            // The esp has data so display its output to the serial window
            char c = esp8266.read(); // read the next character.
            response+=c;
        }
    }
    if(debug)
    {
        Serial.print(response);
    }
    return response;
}

```