Subject : Microprocessor

Class : TI-V/M-1

Lecturer : Albert Suwandhi S.T., M.TI.

: 1. Sarah Veronica (1513121536)

: 2. Fransiskus Ndruru(1109121135)

: 3. Agus Leonardo Gulo(1513121532)

Group Name : C Virus

Duties of Each Member :

1. Sarah Veronica :

    - Coding Developer and Output Designer

2. Fransiskus Ndruru :

    - Coding Developer

3. Agus Leonardo Gulo :

    - Equipment Provider

SMART GARBAGE MONITORING SYSTEM USING INTERNET OF THINGS (IOT)

BACKGROUND

We are going to make an **IOT Garbage Monitoring System** which will tell us that whether the trash can is empty or full through the webserver and you can know the status of your 'Trash Can' from anywhere in the world over the Internet. It will be very useful and can be installed in the Trash Cans at public places as well as at home.

In this IOT Project, an **Ultrasonic Sensor** is used for detecting whether the trash can is filled with garbage or not. Here Ultrasonic Sensor is installed at the top of Trash Can and will measure the distance of garbage from the top of Trash can and we can set a threshold value according to the size of trash can. If the distance will be less than this threshold value, means that the Trash can is full of garbage and we will print the message "Basket is Full" on the webpage and if the distance will be more than this threshold value, then we will print the message "Basket is Empty". Here we have set the Threshold value of 5cm in the Program code. We will use **ESP8266 Wi-Fi module** for connecting the Arduino to the Blynk.

Components Required:

Hardware :

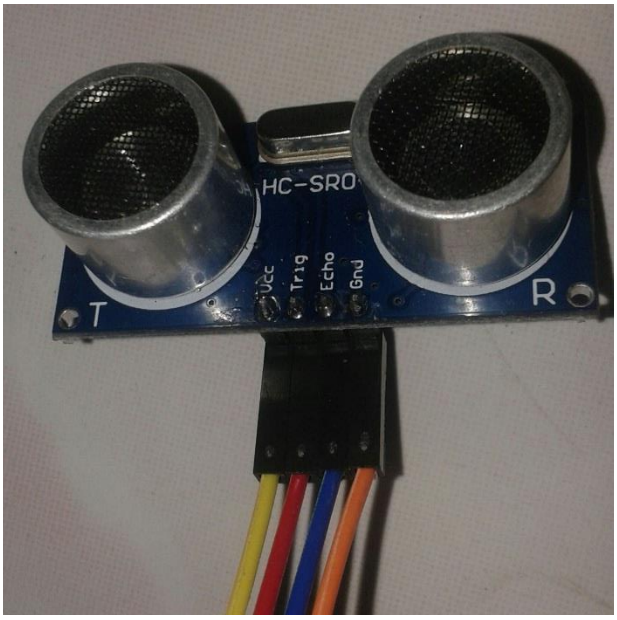
- Arduino Uno
- ESP-12-E Wi-Fi module
- HC-SR04 Ultrasonic sensor
- Breadboard
- Jumper Wires

Software :

- Arduino IDE
- Blynk

HC-SR04 Ultrasonic Sensor:

The Ultrasonic Sensor is used to measure the distance with high accuracy and stable readings. It can measure distance from 2cm to 400cm or from 1 inch to 13 feet. It emits an ultrasound wave at the frequency of 40KHz in the air and if the object will come in its way then it will bounce back to the sensor. By using that time which it takes to strike the object and comes back, you can calculate the distance.



The ultrasonic sensor has four pins. Two are VCC and GND which will be connected to the 5V and the GND of the Arduino while the other two pins are Trig and Echo pins which will be connected to any digital pins of the Arduino. The trig pin will send the signal and the Echo pin will be used to receive the signal. To generate an ultrasound signal, you will have to make the Trig pin high for about 10us which will send a 8 cycle sonic burst at the speed of sound and after striking the object, it will be received by the Echo pin.

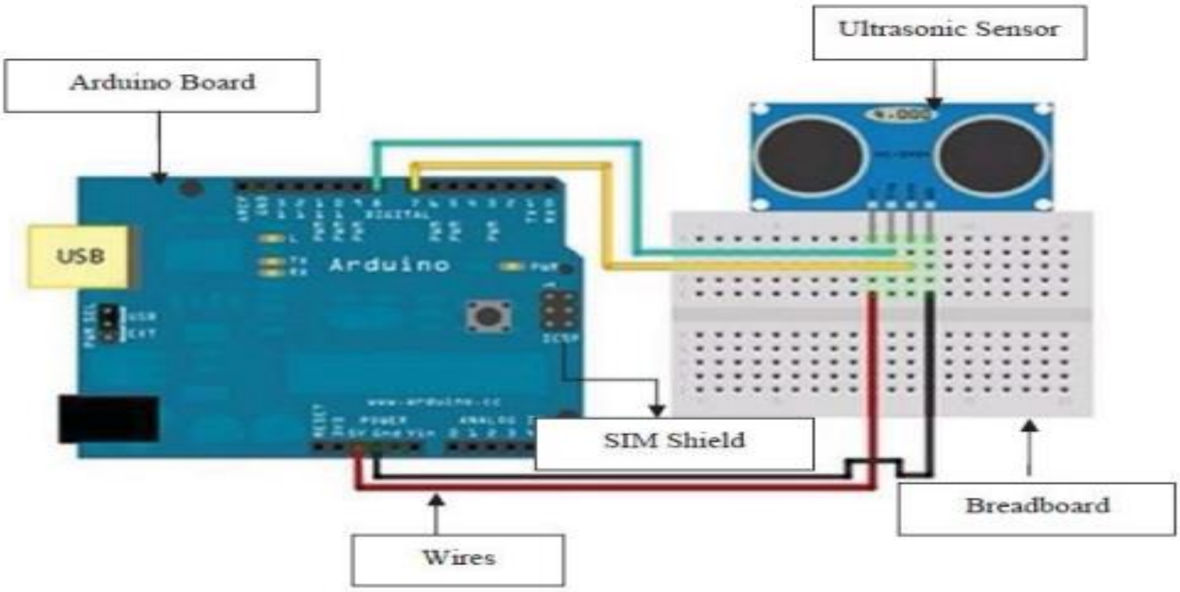
ESP-12-E Wi-Fi Module:

ESP-12-E is a Wi-Fi module which will give your projects access to Wi-Fi or internet. It is a very cheap device but it will make your projects very powerful. It can communicate with any microcontroller and make the projects wireless. It is in the list of most leading devices in the IOT platform. It runs on 3.3V and if you will give it 5V then it will get damageThe TX and RX pins will be responsible for the communication of ESP-12-E with the Arduino. The RX pin works on 3.3V so you will have to make a voltage divider for it as it used for implementation.

Circuit Diagram and Explanation:

First of all we will connect the ESP8266 with the Arduino. ESP8266 runs on 3.3V and if you will give it 5V from the Arduino then it won't work properly and it may get damage. Connect the VCC to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it. Three 1k resistors connected in series will do the work for us. Connect the RX to the pin 11 of the Arduino through the resistors as shown in the figure below and also the TX of the Arduino to the pin 10 of the Arduino.

Now it's time to connect the HC-SR04 ultrasonic sensor with the Arduino. Connections of the ultrasonic sensor with the Arduino are very simple. Connect the VCC and the ground of the ultrasonic sensor to the 5V and the ground of the Arduino. Then connect the TRIG and ECHO pin of ultrasonic sensor to the pin 8 and 9 of the Arduino respectively.

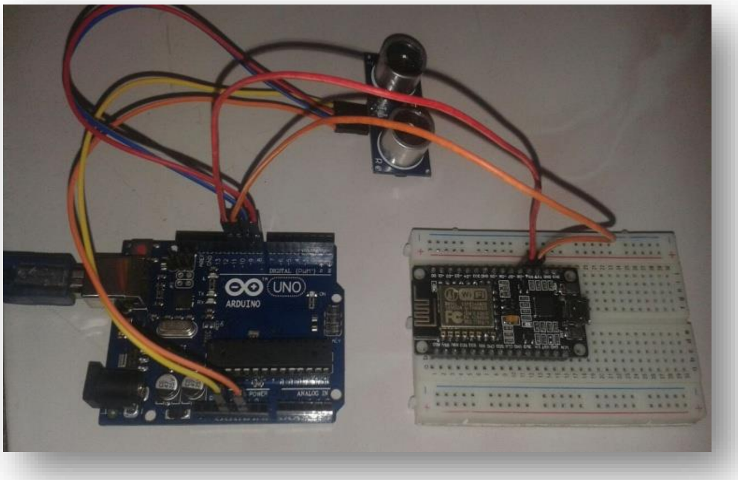


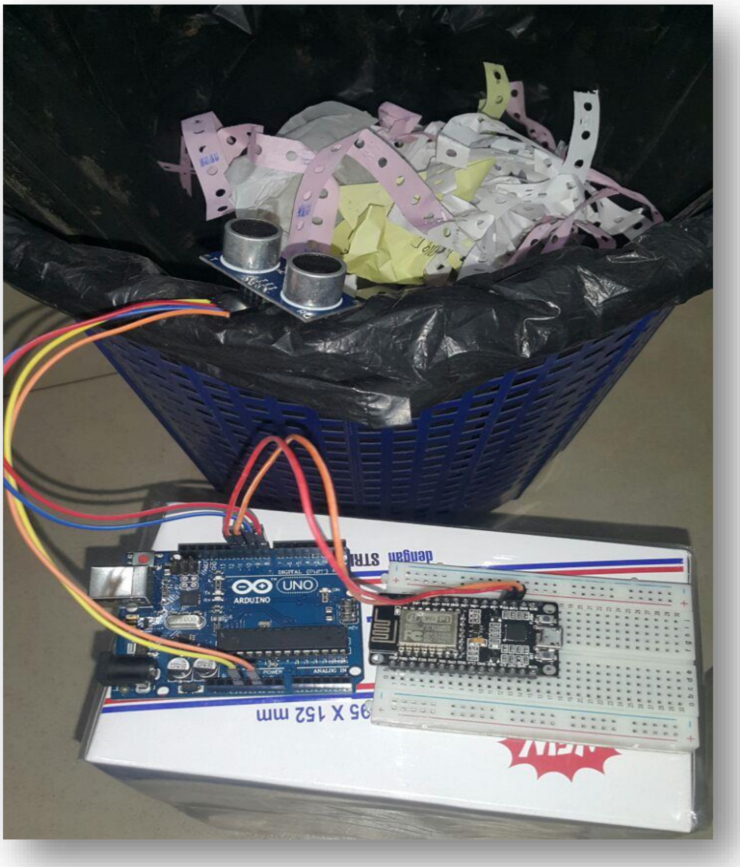
Source Code :

```
#include <SoftwareSerial.h>                    // including the library for the software serial

#define BLYNK_PRINT Serial

#define DEBUG true

SoftwareSerial esp8266(10,11);                /* This will make the pin 10 of arduino as RX pin and pin
11 of arduino as the TX pin Which means that you have to connect the TX from the esp8266 to the pin
10 of arduino and the Rx from the esp to the pin 11 of the arduino*/


 // You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

charauth[] = "d5a0b16e3dd741629971f30851f98f01";

charssid[] = "saaa";                    // Your WiFi credentials.

char pass[] = "pszz0607";         // Set password to "" for open networks.

constinttrigPin = 8;               // Making the arduino's pin 8 as the trig pin of ultrasonic sensor

constintechoPin = 9;          // Making the arduino's pin 9 as the echo pin of the ultrasonic sensor

// defining two variable for measuring the distance

long duration;

int distance;

void setup()

{

Serial.begin(9600);                    // Setting the baudrate at 9600

esp8266.begin(9600);               // Set the baudrate according to you esp'sbaudrate.
youresp'sbaudrate might be different from mine

pinMode(trigPin, OUTPUT);   // Setting the trigPin as Output pin

pinMode(echoPin, INPUT);    // Setting the echoPin as Input pin


sendData("AT+RST\r\n",2000,DEBUG);                      // command to reset the module

sendData("AT+CWMODE=2\r\n",1000,DEBUG);       // This will configure the mode as access point

sendData("AT+CIFSR\r\n",1000,DEBUG);               // This command will get the ip address

sendData("AT+CIPMUX=1\r\n",1000,DEBUG);     // This will configure the esp for multiple
connections

sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG);     // This command will turn on the server on
port 80
```

```
}
void loop()
{
digitalWrite(trigPin, LOW);        // Making the trigpin as low
delayMicroseconds(2);              // delay of 2us
digitalWrite(trigPin, HIGH);    // making the trigpin high for 10us to send the signal
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);      // reading the echopin which will tell us that how much time
the signal takes to come back
distance= duration*0.034/2;              // Calculating the distance and storing in the distance variable

if(esp8266.available())                  // This command will that check if the esp is sending a message
  {
if(esp8266.find("+IPD,"))
    {
delay(1000);
intconnectionId = esp8266.read()-48;/* We are subtracting 48 from the output because the read()
function returns the ASCII decimal value and the first decimal number which is 0 starts at 48*/
     String webpage = "<h1>IOT Garbage Monitoring System</h1>";
webpage += "<p><h2>";
if (distance<5)
       {
webpage+= " Trash can is Full";
        }
else{
webpage+= " Trash can is Empty";
          }
webpage += "</h2></p></body>";
     String 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;
closeCommand+="\r\n";
sendData(closeCommand,3000,DEBUG);
    }
  }
}

String sendData(String command, constint timeout, boolean debug)
{
    String response = "";
esp8266.print(command);
longint time = millis();
while( (time+timeout) >millis())
    {
while(esp8266.available())
      {
char c = esp8266.read();
response+=c;
      }
    }
if(debug)
    {
Serial.print(response);
    }
    }
```

Photo :





Conclution :

By implementing this project we can able to monitor the level of garbage in the dust bins placed at public places,according to that we can collect garbage of particular which will avoid overflow conditions and helps to reduce pollution as well as different hazards of health. This system wil reduce the wastage of fuel by reducing number of trips of garbage collection vehicle. Hence intelligent garbage monitoring system will makes the garbage collection more efficient.