

Smart Jar Using Node Esp8266

Project report

**Submitted in partial fulfillment of the requirements for
Smart Jar using Node MCU Esp8266**

**B Tech
Presidency University**

**Carried out at
Presidency University
Bengaluru**

**By
Dhanush E(20191CSE0126)
Dhanush BC(20191CSE0125)
Dilip Kumar M(20191CSE0132)
Devaraja(20191CSE0124)
Lakshmi
Narayana(20191CSE0159)**



Department of Computer Science & Engineering
Bengaluru



Certificate

This is to certify that the project entitled "**Smart Jar Using Node ESP8266** " has been successfully completed by Mr Dhanush E, Mr Dhanush BC, Mr Devaraja, Mr Dilip Kumar M, Lakshmi Narayana of Sixith semester B Tech at **Presidency University, Bengaluru**, as the Internet Of Things project in partial fulfilment for the award of B tech Degree course conducted by the Presidency University. The Project Report presented here is the bonafide work of the student.

Guide:

Prof. Riyaz

Head of the Department

Dr. Kalaisaran

Group Members:

ID	NAME
20191CSE0126	Dhanush E
20191CSE0125	Dhanush BC
20191CSE0132	Dilip Kumar M
20191CSE0124	Devaraja
20191CSE0159	Lakshmi Narayana

Acknowledgement

While performing our project, we had to take the help and guidelines of some respected persons who deserve our greatest gratitude. The completion of this project gave us immense pleasure.

We are highly indebted to Mr. Riyaz sir, for their guidance, constant supervision and for their support in completing our project.

We would like to express our gratitude to our parents for their kind co-operation and encouragement.

ABSTRACT

Nowadays, In our day to day life, people became more busy such that they are much more immersed in their work schedule and are unable to concentrate or take care of their household things. Most of the time people find it difficult to track their kitchen items, because nowadays it has become a room of least visit.

In a scenario, where people find hard time to prepare their own food, it is even more difficult for them to track their grocery items in their room. They even waste most of their time in ordering and tracking the remaining items in kitchen room.

In this project, We are going to make an IoT based Smart jar using Node MCU ESP8266 Module and ultrasonic sensor that tracks the exact percentage of the contents in the jar.

This helps the user to know the quantity of content in the jar and can access the information through the Internet from anywhere in the world.

Table of Contents

Acknowledgement

Abstract

1. Components Required	06
2.Features of Components Used	07
3.Pinout Diagram	10
4.Manual Connection of the Project	11
5.Code	12
6.Read Me	14
7.Conclusion	15

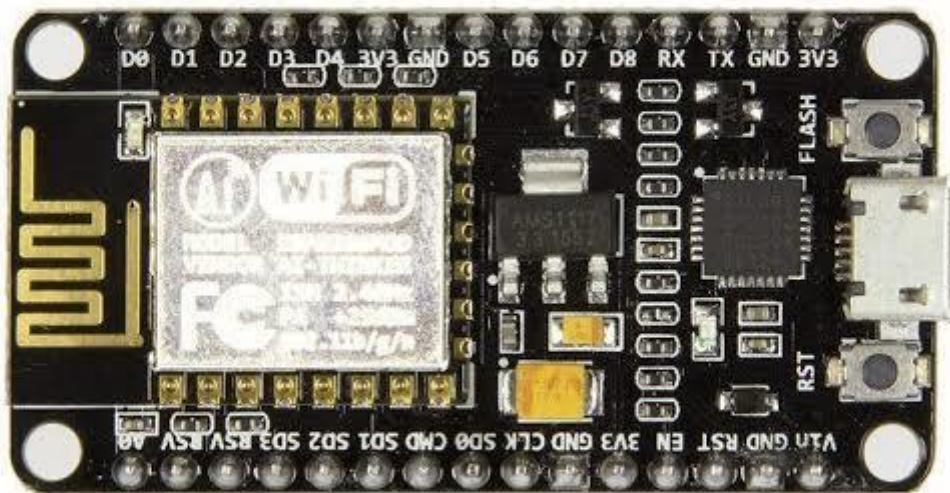
COMPONENTS USED

1. Node MCU Esp8266
2. Ultrasound sensor (HC-04)
3. Jumper Wires
4. USB Cable

FEATURE OF COMPONENTS USED

1. Node MCU ESP8266

The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip(SoC)calledthe ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer.



2. Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal.



3. Jumper Wires

A jump **wire** (also known as **jumper wire**, or **jumper**) is an electrical **wire**, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally **used** to interconnect the components of a breadboard or other prototype or test circuit, internally or with other



Male to Male Jumper
Wires



Female to Female Jumper
Wires



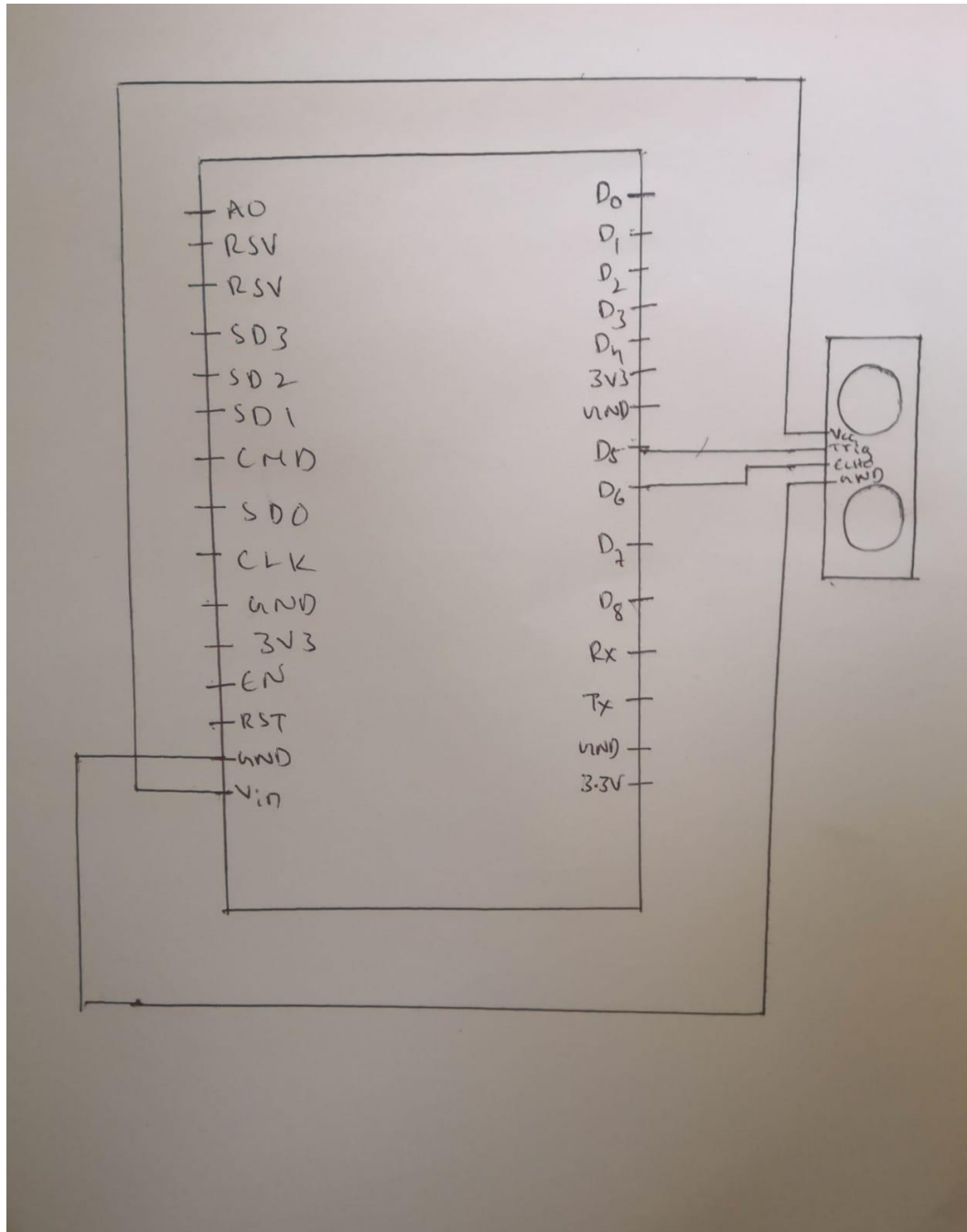
Male to Female Jumper
Wires

4. USB Cable

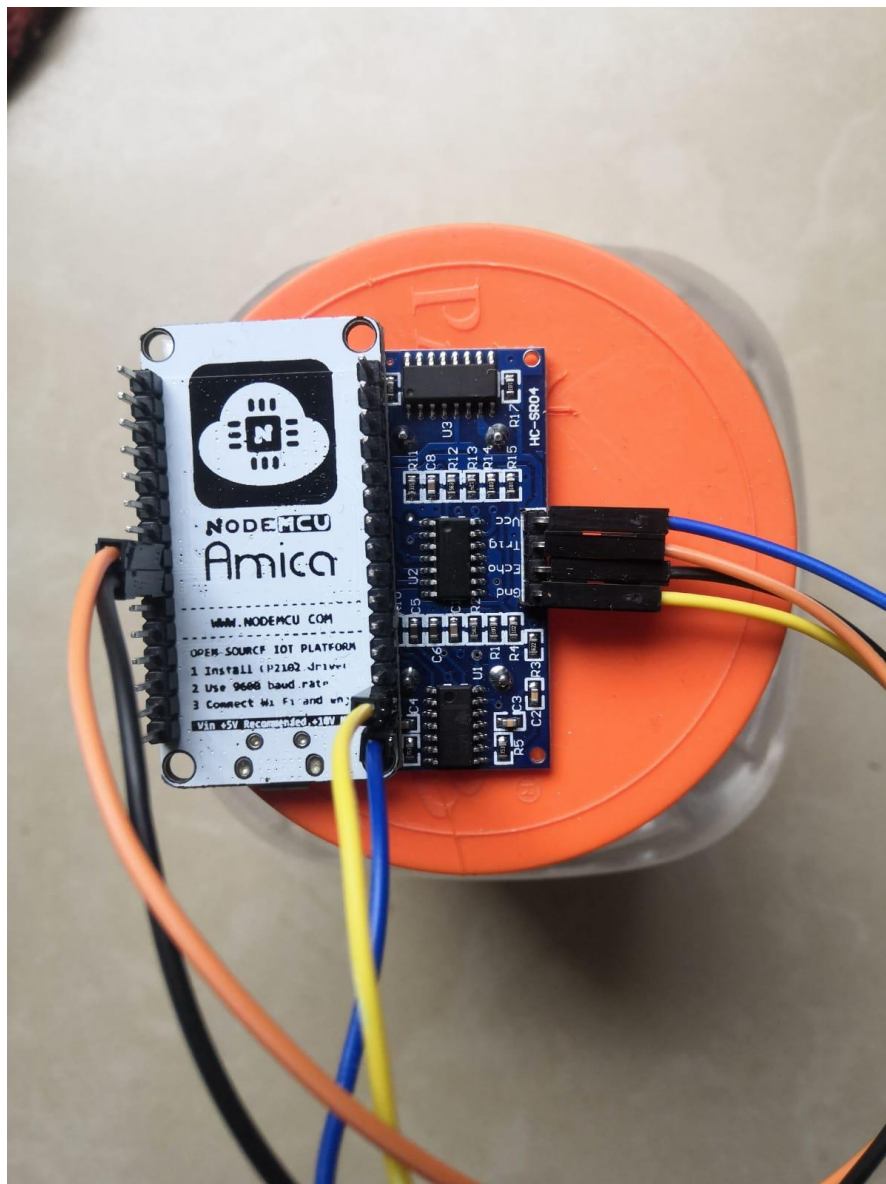
The term USB stands for "Universal Serial Bus". USB cable assemblies are some of the most popular cable types available, used mostly to connect computers to peripheral devices such as cameras, camcorders, printers, scanners, and more.



PINOUT DIAGRAMS



Manual Connection of the Project:



CODE

```
#include <ESP8266WiFi.h>
#define BLYNK_TEMPLATE_ID "Your blynk template id"
#define BLYNK_DEVICE_NAME "Smart Jar"
#define BLYNK_AUTH_TOKEN "Your blynk api key"
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid1[] = "wifi name";
char pass1[] = "wifi password";
BlynkTimer timer;
const int trigPin = D5;
const int echoPin = D6;
long duration;
int distance;
float level;
const char* ssid = "wifi name";
const char* password = "wifi password";
void send_event(const char *event);
const char *host = "maker.ifttt.com";
const char *privateKey = "Your API key of IFTTT";
WiFiServer server(80);
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  Blynk.begin(auth, ssid1, pass1);
  Serial.print("Connecting to Wifi Network");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("Successfully connected to WiFi.");
  Serial.println("IP address is : ");
  Serial.println(WiFi.localIP());
  server.begin();
  Serial.println("Server started");
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0340 / 2;
  Serial.println("Distance");
  Serial.println(distance);
}
```

```

level = ((12-distance)/12.0)*100;
Serial.println("level");
Serial.println(level);
delay(1000);
WiFiClient client = server.available();
Blynk.virtualWrite(V0, level);
Blynk.run();
timer.run();
if ( level <= 10) {
    send_event("jar_event");
}
}
void send_event(const char *event)
{
    Serial.print("Connecting to ");
    Serial.println(host);
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("Connection failed");
        return;
    }
    String url = "/trigger/";
    url += event;
    url += "/with/key/";
    url += privateKey;
    Serial.print("Requesting URL: ");
    Serial.println(url);
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" +
        "Connection: close\r\n\r\n");
    while(client.connected())
    {
        if(client.available())
        {
            String line = client.readStringUntil('\r');
            Serial.print(line);
        } else {
            delay(50);
        };
    }
    Serial.println();
    Serial.println("closing connection");
    client.stop();
}

```

README

- 1) First give the power supply to the Node MCU Esp8266 board
- 2) The ultrasound sensor is connected
- 3) Switch on the Wi-Fi
- 4) The node mcu connects to the Wi-Fi
- 5) The ultrasound wave sends the wave and the duration is noted
- 6) The duration noted is then converted into distance by multiplying with the speed
- 7) This distance is then converted to level
- 8) The data is displayed using Blynk app
- 9) If the quantity of the jar is less than ten percent sends a email notification

CONCLUSION

Through this project we came across various components which gave us more insight about the subject “Internet Of Things”. Our project was about Smart Jar Using Node MCU ESP 8266.

This objective of our project is to provide the improvement on Kitchen automation for the humans and save time for them and reduce food wastage. This is the main objective of our project.