NAAN MUDHALVAN – PROFESSIONAL READINESSFOR INNOVATION, EMPLOYMENT AND ENTERPRENEURSHIP

ASSIGNMENT – 3

STUDENT NAME	**SHOBANA S**
STUDENT ROLL NO	**NM2023TMID14436**

QUESTION:

Build wowki product, use ultrasonic sensor and detect thedistance from the object. Whenever distance is less than 100cms upload the value to the ibm cloud.in recent deviceevents upload the data from wokwi.

- Example: distance is 20 cms. Upload the 20 value to theibm cloud in recent event in the ibm iot platform device

- Submit the Assignment in PDF format in the Git repo.
- PDF should have wokwi share link, connections image, code, IBM cloud recent events image(Screenshot)

- Everyone in the team should submit the assignment as it is an individual task.

LINK:

https://wokwi.com/projects/364311694669799425

Code:

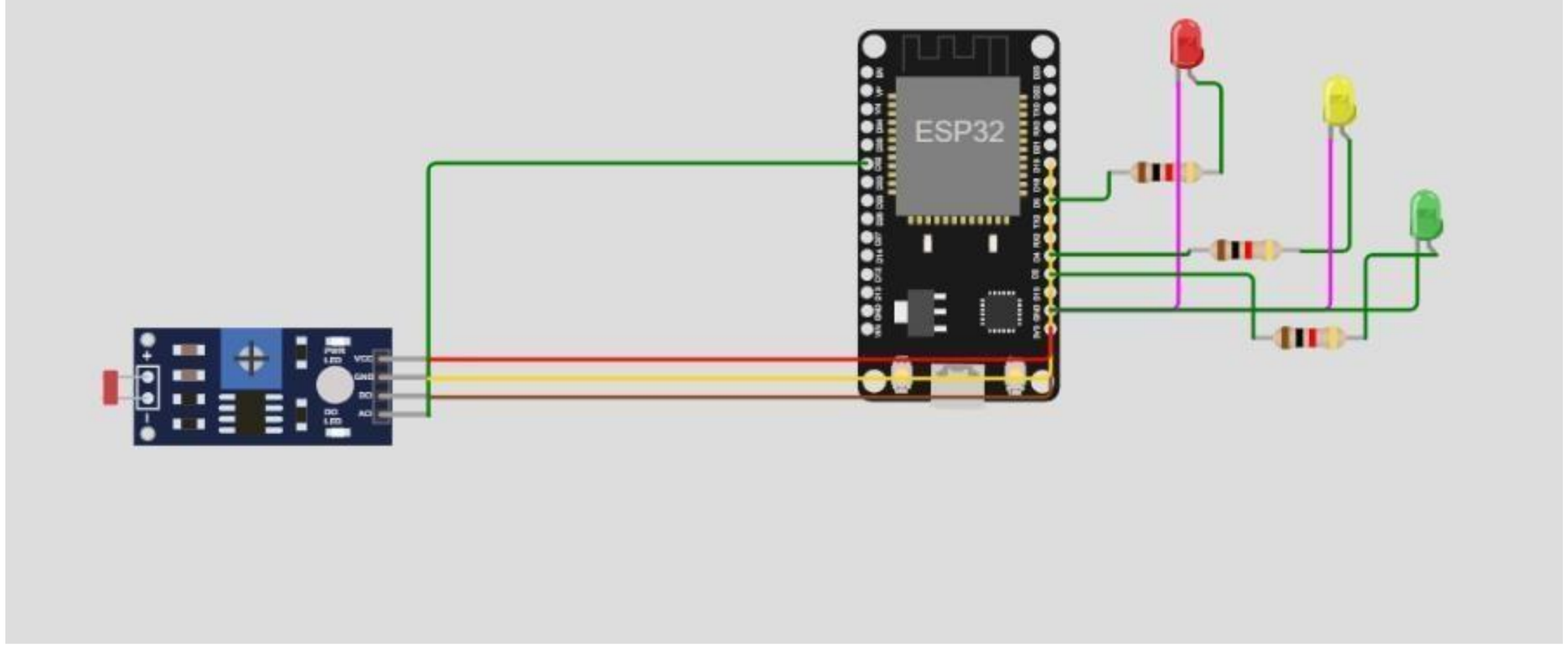
```
#include <WiFi.h
#include<PubSubClient.h

#define LED 5
#define LED2 4 #define LED3 2
int LDR = 32;
intLDRReading=0;
```

```
int threshold_val = 800;
int lEDBrightness = 0;intflag=0;
  void callback(char* subscribetopic, byte* payload, unsigned intpayloadLength);
//-------credentials of IBM Accounts------

#define ORG "stuloy"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform#define
DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform #define TOKEN
"12345678"                                    //Token
String data3; float h, t;
//-------- Customise the above values -------- char server[] = ORG ".messaging.internetofthings.ibmcloud. com";// Server
Name char publishTopic[] ="iot-2/evt/Data/fmt/json";// topic name and type of event perform and format inwhich
data to be send char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command type AND
COMMAND IS TEST OF FORMAT STRING char authMethod[]
= "use-token-auth";// authentication method char token[] = TOKEN; char clientId[] = "d:" ORG ":"
DEVICE_TYPE ":" DEVICE_ID;//client id


- - - - - - - - - - - - - - - - - - - - - - - - - -//        -
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefinedclient id by passing
parameter like server id,portand wificredential void setup()// configureing the ESP32
{
    Serial.begin(115200);

pinMode(LED,OUTPUT);
pinMode(LED2,OUTPUT);
pinMode(LED3,OUTPUT);
delay(10);
Serial.println();
wificonnect();
mqttconnect();
} void loop()// RecursiveFunction
{
    if (!client.loop()) {
mqttconnect();
    }
}


void mqttconnect()
{       if (!client.connected())
  {
          Serial.print("Reconnecting client to ");
  Serial.println(server);
          while (!!!client.connect(clientId, authMethod, token)) {Serial.print(".");
                                                  delay(500);
          }
            initManagedDevice();
            Serial.println();
      } }
  void wificonnect() //function defination for wificonnect
  {
     Serial.println();
```

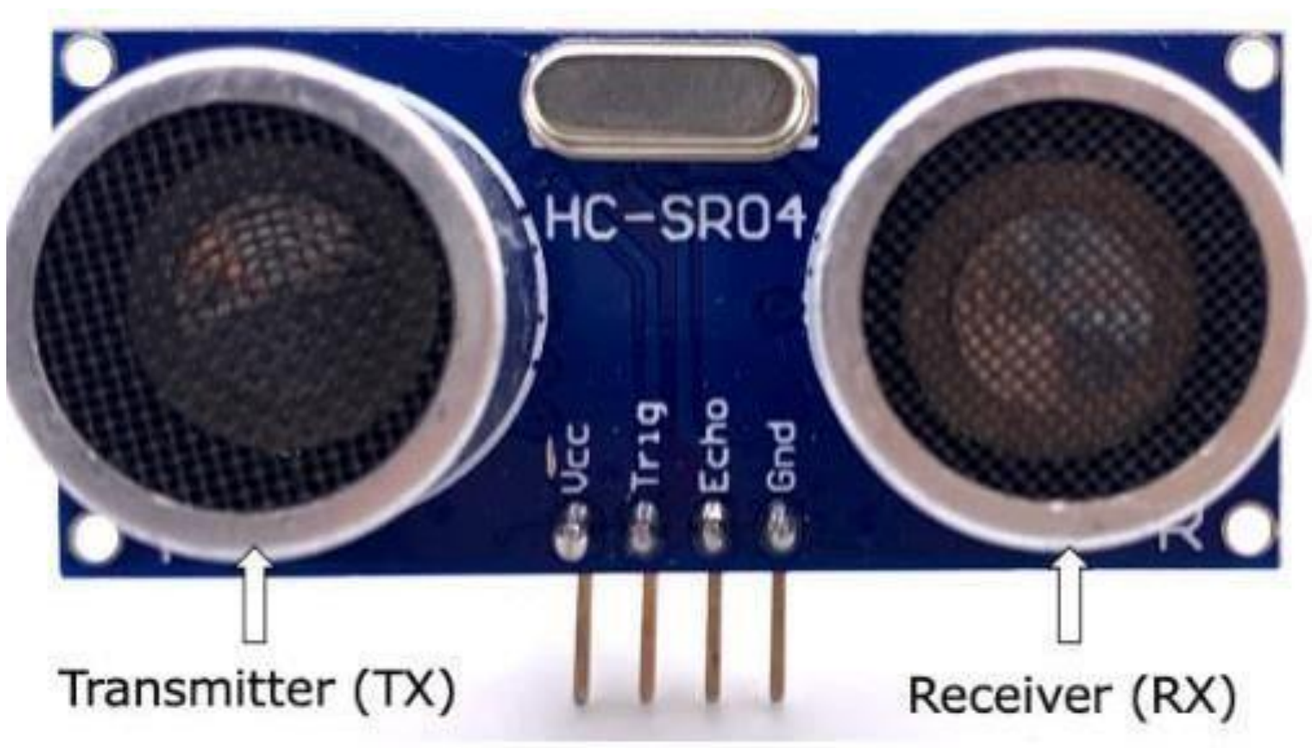
```
Serial.print("Connectingto ");

WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establishthe connection
while(WiFi.status() != WL_CONNECTED) {
delay(500);
    Serial.print(".");
}
Serial.println("");                          Serial.println("WiFi
connected");Serial.println("IP              address:              ");
   Serial.println(WiFi.localIP());
} void initManagedDevice(){
   if              (client.subscribe(subscribetopic))              {
Serial.println((subscribetopic));
       Serial.println("subscribe to cmd OK");
    } else {
       Serial.println("subscribe to cmd FAILED");
   }
} void callback(char* subscribetopic, byte* payload, unsigned intpayloadLength)
{


   Serial.print("callback      invoked      for      topic:      ");
   Serial.println(subscribetopic);
       for (int i = 0; i < payloadLength; i++)
{          //Serial.print((char)payload[i]);data3      +=
(char)payload[i];
   }

   Serial.println("data:  "+ data3);if(data3=="lighton1")
   {
Serial.println(data3); digitalWrite(LED,HIGH);
   }        else
if(data3=="lightoff1")
   {
Serial.println(data3);
digitalWrite(LED,LOW)
;

   }
   else if(data3=="lighton2")
   {
Serial.println(data3); digitalWrite(LED2,HIGH);
                                        }       else
if(data3=="lightoff2")
   {
Serial.println(data3); digitalWrite(LED2,LOW);
     }      else
if(data3=="lighton3")
   {
Serial.println(data3);
digitalWrite(LED3,HIGH);

   }        else
if(data3=="lightoff3")
   {
Serial.println(data3); digitalWrite(LED3,LOW);
```

```
    }
data3="";


}
```

CIRCUIT DIAGRAM:



IBM CLOUD RECENT EVENT IMAGE:



..