SPRINT 2

Team ID : PNT2022TMID32082

Team leader : T.A.KAVIPRAKASH.

Team member 1 : B.CHIRADEEP.

Team member 2 : R.KUMARAVEL.

Team member 3 : S.POOVARASAN.

CODE:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt


void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "53soe3"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_ibm"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Sprint_2"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "XAeyOgS38sJ@0a*PM5"
String data3;
float dist;
//-------- 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 in which data to be send
```

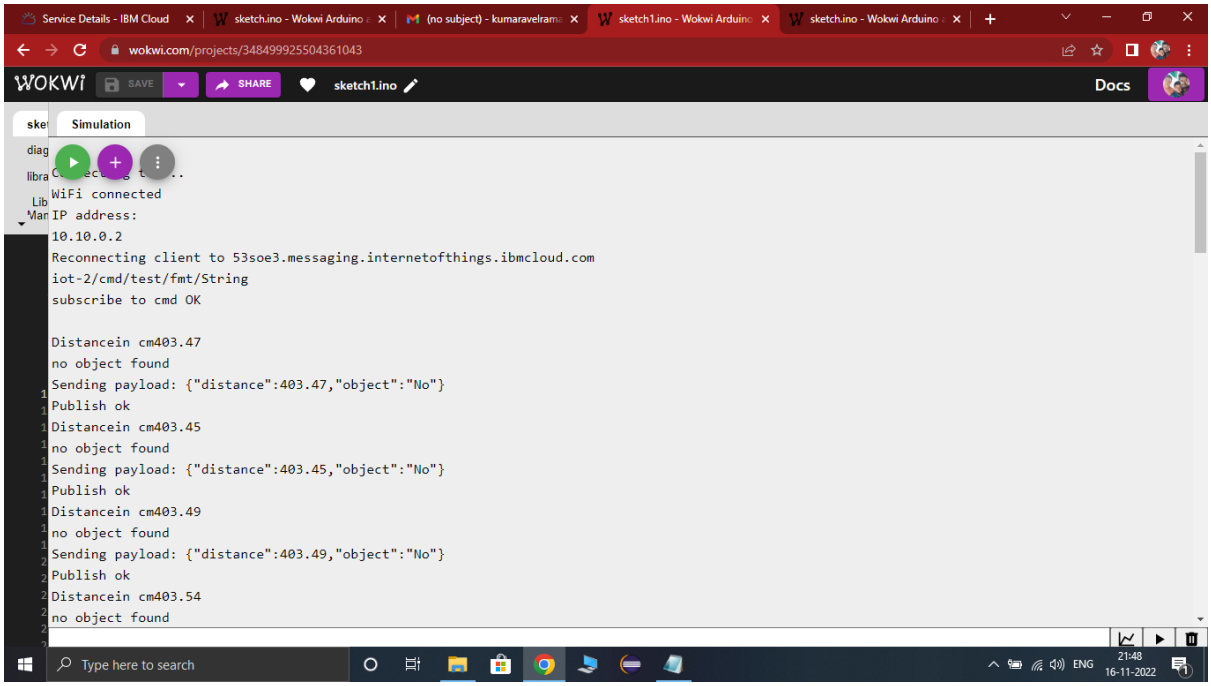
```
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

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 predefined client id by passing parameter like server id,portand wificredential


int LED = 4;

int trig = 5;

int echo = 18;

void setup()

{

Serial.begin(115200);

pinMode(trig,OUTPUT);

pinMode(echo,INPUT);

pinMode(LED, OUTPUT);

delay(10);

wificonnect();

mqttconnect();

}

void loop()// Recursive Function

{


 digitalWrite(trig,LOW);

  digitalWrite(trig,HIGH);

  delayMicroseconds(10);

  digitalWrite(trig,LOW);

  float dur = pulseIn(echo,HIGH);
```

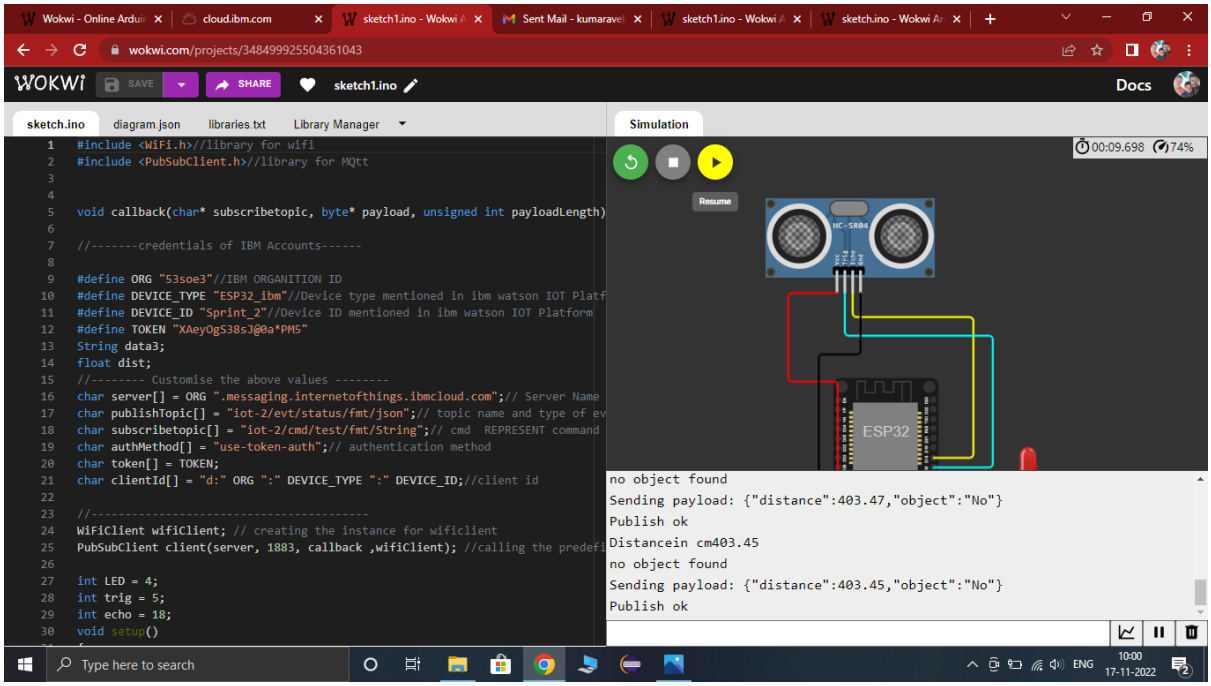
```
  float dist = (dur * 0.0343)/2;
  Serial.print ("Distancein cm");
  Serial.println(dist);



  PublishData(dist);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}

/*.....retrieving to Cloud.....*/

void PublishData(float dist) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String object;
  if (dist <100)
  {
    digitalWrite(LED,HIGH);
    Serial.println("object is near");
    object = "Near";
  }
  else
  {
    digitalWrite(LED,LOW);
    Serial.println("no object found");
    object = "No";
```

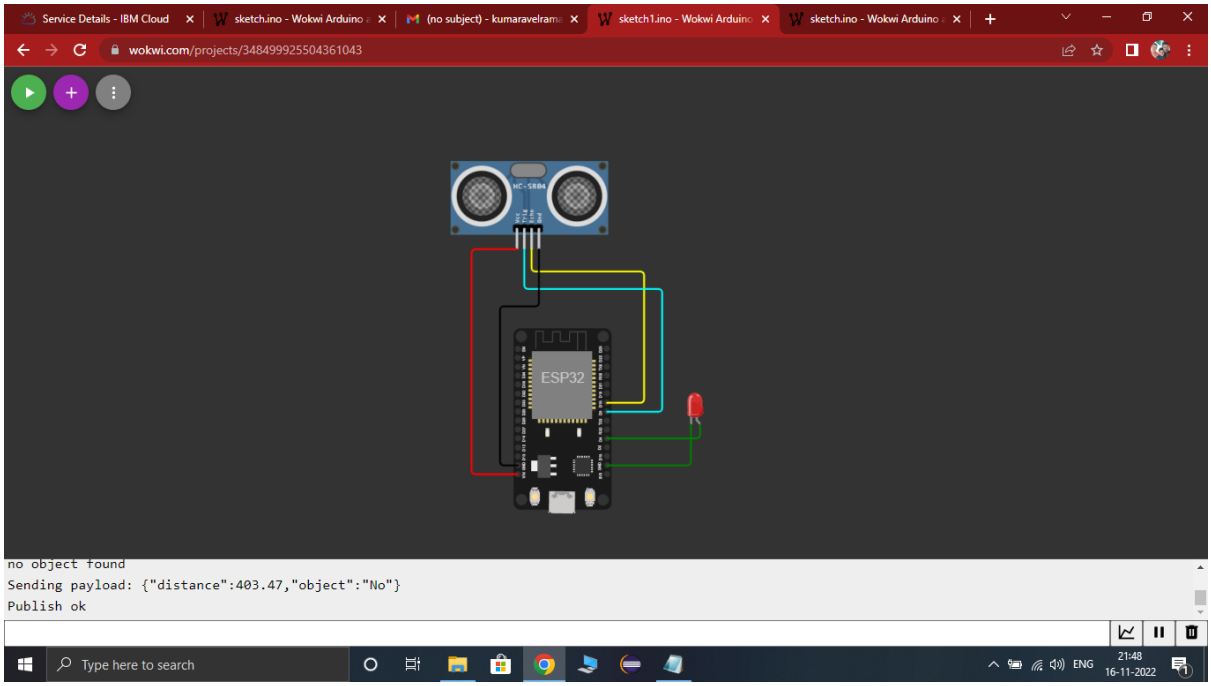
```
  }

  String payload = "{\"distance\":";
  payload += dist;
  payload += "," "\"object\":\"";
  payload += object;
  payload += "\"}";



  Serial.print("Sending payload: ");
  Serial.println(payload);




  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok
in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
  }


}
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("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the 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 int payloadLength)
```

```
{

  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=="Near")
//   {
// Serial.println(data3);
// digitalWrite(LED,HIGH);

//   }

//   else
//   {
// Serial.println(data3);
// digitalWrite(LED,LOW);

//   }
data3="";



}
```


wokwi.com/projects/348499925504361043
WOKWI
SAVE
SHARE
sketch1.ino
Docs
Simulation
WiFi connected
IP address:
10.10.0.2
Reconnecting client to 53soe3.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK
Distancein cm403.47
no object found
Sending payload: {"distance":403.47,"object":"No"}
Publish ok
Distancein cm403.45
no object found
Sending payload: {"distance":403.45,"object":"No"}
Publish ok
Distancein cm403.49
no object found
Sending payload: {"distance":403.49,"object":"No"}
Publish ok
Distancein cm403.54
no object found
Type here to search
ENG
21:48
16-11-2022


wokwi.com/projects/348499925504361043
WOKWI
SAVE
SHARE
sketch1.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
//-------credentials of IBM Accounts------
#define ORG "53soe3"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_ibm"//Device type mentioned in ibm watson IOT Platf
#define DEVICE_ID "Sprint_2"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "XAeyOgS38sJ@0a*PM5"
String data3;
float dist;
//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/status/fmt/json";// topic name and type of ev
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command
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 predefi
int LED = 4;
int trig = 5;
int echo = 18;
void setup()
Simulation
00:09.698
74%
Resume
HC-SR04
ESP32
no object found
Sending payload: {"distance":403.47,"object":"No"}
Publish ok
Distancein cm403.45
no object found
Sending payload: {"distance":403.45,"object":"No"}
Publish ok
Type here to search
ENG
10:00
17-11-2022



Device ID	Status	Device Type	Class ID
Sprint_2	Connected	ESP32_ibm	Device

Identity | Device Information | Recent Events | State | Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"distance":403.49,"object":"No"}	json	a few seconds ago
status	{"distance":403.54,"object":"No"}	json	a few seconds ago
status	{"distance":403.49,"object":"No"}	json	a few seconds ago
status	{"distance":403.49,"object":"No"}	json	a few seconds ago
status	{"distance":403.45,"object":"No"}	json	a few seconds ago

https://wokwi.com/projects/348499925504361043