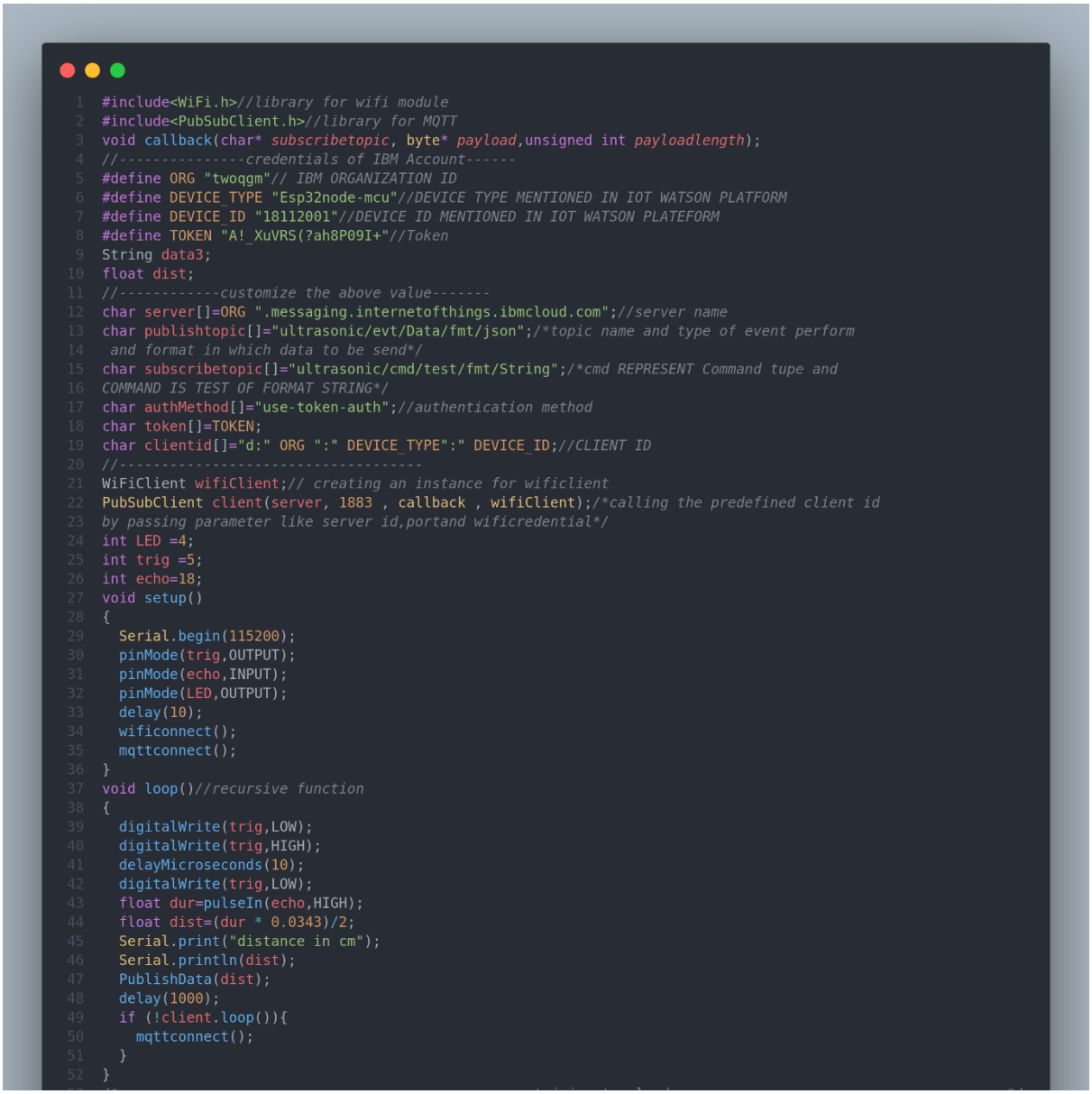
Personal assistant for senior citizen

STUDENT NAME: ANTONY RENOLD DICKSON K

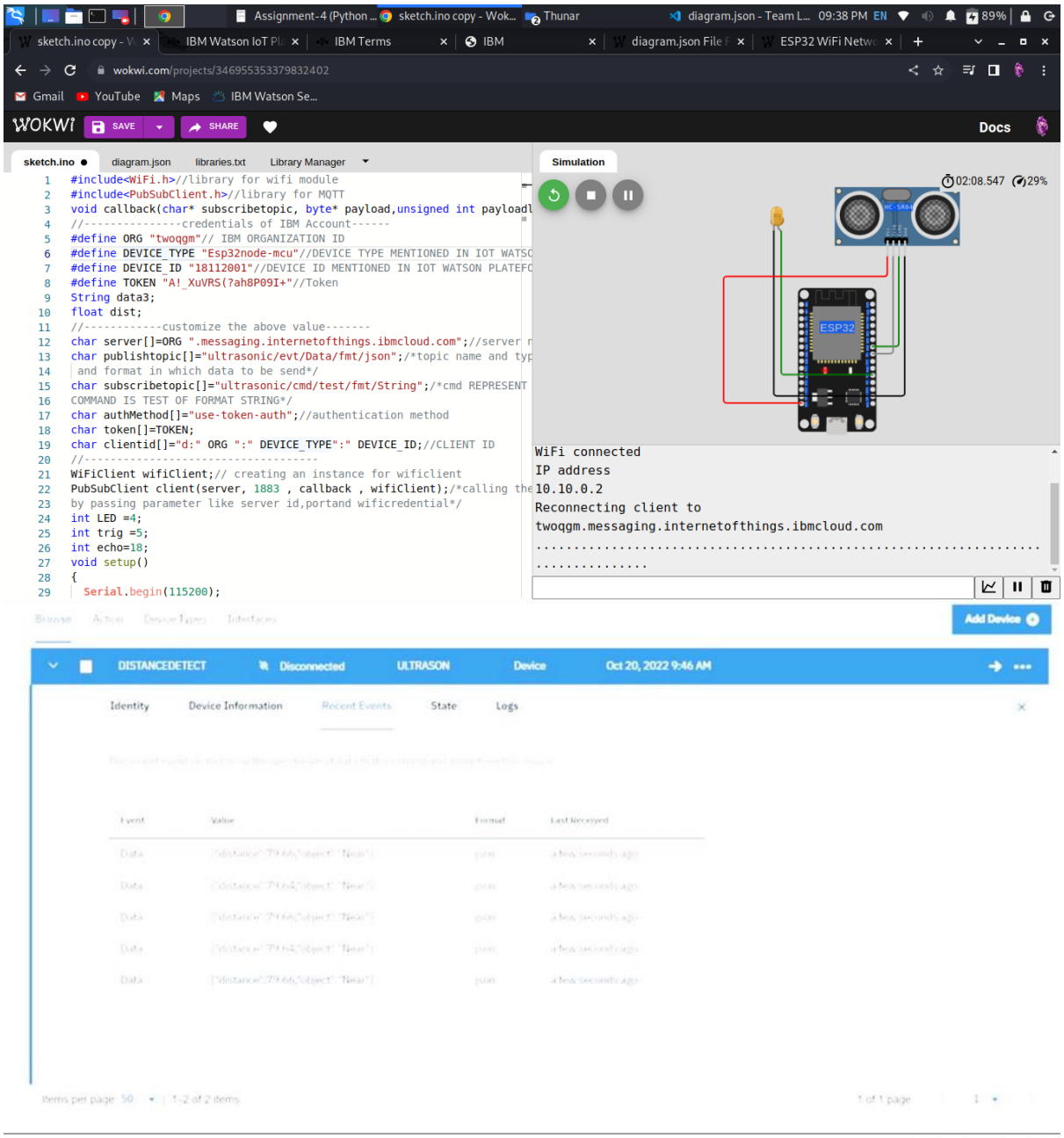
ROLL NO: 953019104003

TEAM ID: PNT2022TMID50622

URL: https://wokwi.com/projects/346955353379832402



```
#include<WiFi.h>//library for wifi module
#include<PubSubClient.h>//library for MQTT
void callback(char* subscribetopic, byte* payload,unsigned int payloadlength);
//---------------credentials of IBM Account------
#define ORG "twoqgm"// IBM ORGANIZATION ID
#define DEVICE_TYPE "Esp32node-mcu"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
#define DEVICE_ID "18112001"//DEVICE ID MENTIONED IN IOT WATSON PLATEFORM
#define TOKEN "A!_XuVRS(?ah8P09I+"//Token
String data3;
float dist;
//------------customize the above value-------
char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server name
char publishtopic[]="ultrasonic/evt/Data/fmt/json";/*topic name and type of event perform
 and format in which data to be send*/
char subscribetopic[]="ultrasonic/cmd/test/fmt/String";/*cmd REPRESENT Command tupe 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 an 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("distance in cm");
  Serial.println(dist);
  PublishData(dist);
  delay(1000);
  if (!client.loop()){
    mqttconnect();
  }
}
```


Assignment-4 (Python ...
sketch.ino copy - Wok...
Thunar
diagram.json - Team L...
09:38 PM
89%
sketch.ino copy - W
IBM Watson IoT Pl
IBM Terms
IBM
diagram.json File
ESP32 WiFi Netwo
wokwi.com/projects/346955353379832402
Gmail
YouTube
Maps
IBM Watson Se...
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include<WiFi.h>//library for wifi module
#include<PubSubClient.h>//library for MQTT
void callback(char* subscribetopic, byte* payload,unsigned int payloadl
//---------------credentials of IBM Account------
#define ORG "twoqgm"// IBM ORGANIZATION ID
#define DEVICE_TYPE "Esp32node-mcu"//DEVICE TYPE MENTIONED IN IOT WATSC
#define DEVICE_ID "18112001"//DEVICE ID MENTIONED IN IOT WATSON PLATEFC
#define TOKEN "A!_XuVRS(?ah8P09I+"//Token
String data3;
float dist;
//-----------customize the above value-------
char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server r
char publishtopic[]="ultrasonic/evt/Data/fmt/json";/*topic name and typ
and format in which data to be send*/
char subscribetopic[]="ultrasonic/cmd/test/fmt/String";/*cmd REPRESENT
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 an instance for wificlient
PubSubClient client(server, 1883 , callback , wifiClient);/*calling the
by passing parameter like server id,portand wificredential*/
int LED =4;
int trig =5;
int echo=18;
void setup()
{
Serial.begin(115200);
Simulation
02:08.547
29%
ESP32
WiFi connected
IP address
10.10.0.2
Reconnecting client to
twoqgm.messaging.internetofthings.ibmcloud.com
Add Device
DISTANCEDETECT
Disconnected
ULTRASON
Device
Oct 20, 2022 9:46 AM
Identity
Device Information
Recent Events
State
Logs

```
/*.....retriving to cloud.....*/
void PublishData(float dist){
  mqttconnect();//function call for connecting to ibm
  /*creating the string in form of JSON to update the data to ibm cloud*/
  String object;
  if(dist<100)
  {
    digitalWrite(LED,HIGH);
    Serial.println("no 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 its 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 defenition for wificonnects
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "",6);//PASSING THE WIFI CREDIDENTIALS TO ESTABLISH 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("dta: "+ data3);
  //if(data3=="Near")
  //{
  //Serial.println(data3);
  //digitalWrite(LED,HIGH);
  //}
  //else
  //{
  //Serial.println(data3);
  //digitalWrite(LED,LOW);
  //}
  data3="";
}
```

```
{
    "version": 1,
    "author": "VIGNESH M",
    "editor": "wokwi",
    "parts": [
      { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 5.67, "left": -820.91, "attrs": {} },
      { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -128.92, "left": -771.12, "attrs": {} },
      {
        "type": "wokwi-led",
        "id": "led1",
        "top": -111.38,
        "left": -868.58,
        "attrs": { "color": "orange" }
      }
    ],
    "connections": [
      [ "esp:TX0", "$serialMonitor:RX", "", [] ],
      [ "esp:RX0", "$serialMonitor:TX", "", [] ],
      [ "led1:A", "esp:D4", "green", [ "v0" ] ],
      [ "led1:C", "esp:GND.1", "black", [ "v0" ] ],
      [ "ultrasonic1:ECHO", "esp:D18", "green", [ "v0" ] ],
      [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v26.59", "h-222.45", "v199.17" ] ],
      [ "ultrasonic1:TRIG", "esp:D5", "gray", [ "v0" ] ],
      [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ]
    ]
  }
```