|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assignment-4**   |  |  | | --- | --- | | AssignmentDate | 31October2022 | | StudentName | TIMPLE RANI M | | StudentRollNumber | 422019106004 | | MaximumMarks | 2Marks |   **Question-1:**  Writecodeandconnectionsinwokwiforultrasonicsensor.Wheneverdistanceislessthan100cmsend"alert"toIBMcloudanddisplayindevicerecentevents.  WOWKILINK:  <https://wokwi.com/projects/347112319943180884>  **Solution:**  #include<WiFi.h>  #include<WiFiClient.h>  #include<PubSubClient.h>  constinttrigPin=5;  constintechoPin=18;  //definesoundspeedincm/uS  #defineSOUNDSPEED0.034  #defineCMTOINCH0.393701  longduration;  floatdistanceCm;  floatdistanceInch; | |
| voidcallback(char\*subscribetopic,byte\*payload,unsigned  //-------credentialsofIBMAccounts------  #defineORG"b4flhz"//IBMORGANITIONID  #defineDEVICETYPE"Assignment-4"//Devicetypementionedin#defineDEVICEID"KalaivananID"//DeviceIDmentionedinibm  #defineTOKEN"w?k\*p3hAGb&V&+jutF"//Token  Stringdata3; | intpayloadLength);  ibmwatsonIOTPlatform  watsonIOTPlatform |
| //--------Customisetheabovevalues--------  charserver[]=ORG".messaging.internetofthings.ibmcloud.com";//ServerName  charpublishTopic[]="iot-2/evt/Data/fmt/json";//topicnameandtypeofeventperformandformatinwhichdatatobesend  charsubscribetopic[]="iot-2/cmd/test/fmt/String";//cmdREPRESENTcommandtypeAND  COMMANDISTESTOFFORMATSTRING  charauthMethod[]="use-token-auth";//authenticationmethod  chartoken[]=TOKEN;  charclientId[]="d:"ORG":"DEVICETYPE":"DEVICEID;//clientid  WiFiClientwifiClient;//creatingtheinstanceforwificlient | |

|  |
| --- |
| PubSubClientclient(server,1883,callback,wifiClient);  voidsetup(){  **Serial**.begin(115200);//Startstheserialcommunication  pinMode(trigPin,OUTPUT);//SetsthetrigPinasanOutput  pinMode(echoPin,INPUT);//SetstheechoPinasanInput  **Serial**.println();  wificonnect();  mqttconnect();  }  voidloop(){  //ClearsthetrigPin  digitalWrite(trigPin,LOW);  delayMicroseconds(2);  //SetsthetrigPinonHIGHstatefor10microseconds  digitalWrite(trigPin,HIGH);  delayMicroseconds(10);  digitalWrite(trigPin,LOW);  //ReadstheechoPin,returnsthesoundwavetraveltimeinmicrosecondsduration=pulseIn(echoPin,HIGH);  //Calculatethedistance  distanceCm=duration\*SOUNDSPEED/2;  //Converttoinches  distanceInch=distanceCm\*CMTOINCH;  //PrintsthedistanceintheSerialMonitor  **Serial**.print("Distance(cm):");  **Serial**.println(distanceCm);  **Serial**.print("Distance(inch):");  **Serial**.println(distanceInch);  PublishData(distanceCm);  delay(1000);  if(!client.loop()){  mqttconnect();  }  }  voidPublishData(floatCm){  mqttconnect();//functioncallforconnectingtoibm  /\*  creatingtheStringininformJSontoupdatethedatatoibmcloud  \*/  Stringpayload="{\"Distance(cm)\":";  payload+=Cm;  payload+="}"; |

|  |
| --- |
| **Serial**.print("Sendingpayload:");  **Serial**.println(payload);  if(client.publish(publishTopic,(char\*)payload.cstr())){  **Serial**.println("Publishok");//ifitsucessfullyuploaddataonthecloudthenitwillprintpublishokinSerialmonitororelseitwillprintpublishfailed  }else{  **Serial**.println("Publishfailed");  }  }  voidmqttconnect(){  if(!client.connected()){  **Serial**.print("Reconnectingclientto");  **Serial**.println(server);  while(!!!client.connect(clientId,authMethod,token)){  **Serial**.print(".");  delay(500);  }  initManagedDevice();  **Serial**.println();  }  }  voidwificonnect()//functiondefinationforwificonnect  {  **Serial**.println();  **Serial**.print("Connectingto");  WiFi.begin("Wokwi-GUEST","",6);//passingthewificredentialstoestablishtheconnection  while(WiFi.status()!=WLCONNECTED){  delay(500);  **Serial**.print(".");  }  **Serial**.println("");  **Serial**.println("WiFiconnected");  **Serial**.println("IPaddress:");  **Serial**.println(WiFi.localIP());  }  voidinitManagedDevice(){  if(client.subscribe(subscribetopic)){  **Serial**.println((subscribetopic));  **Serial**.println("subscribetocmdOK");  }else  {  **Serial**.println("subscribetocmdFAILED");  }  }  voidcallback(char\*subscribetopic,byte\*payload,unsignedintpayloadLength){ |

|  |
| --- |
| **Serial**.print("callbackinvokedfortopic:");  **Serial**.println(subscribetopic);  for(inti=0;i<payloadLength;i++){  //Serial.print((char)payload[i]);  data3+=(char)payload[i];  }  }  **Output:** |

|  |
| --- |
|  |