ASSIGNMENT 4

PYTHON PROGRAMMING

ASSIGNMENT DATE	17 OCTOBER 2022
STUDENT NAME	RAVICHANDRAN .T
STUDENT ROLL NUMBER	962719106024
MAXIMUM MARKS	2

Question:

Write code and connections in wowki for ultrasonic

sensor. Whenever distance is less than 100cms send

"alert" to IBM cloud and display in device recent events.

Program 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 "n6rl9n"//IBM Organition ID

#define DEVICE_TYPE "ElangoIoT"//Device type mentioned in ibm watson IOT

Platform

#define DEVICE_ID "06112002"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "98765432" //Token

String data3;

// 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
```

```cpp
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

const int trigpin = 5;

const int echopin = 18;

const int ledpin = 2;

long duration ;

float distance;

#define sound_speed 0.034

void setup()

{

Serial.begin(115200);

pinMode(trigpin, OUTPUT);

pinMode(echopin, OUTPUT);

pinMode(ledpin, OUTPUT);

wificonnect();

mqttconnect();

}

void loop()

{
```

```
digitalWrite(trigpin, LOW);

digitalWrite(trigpin, HIGH);

delayMicroseconds(10);

digitalWrite(trigpin, LOW);

duration= pulseIn(echopin,HIGH);

distance = duration * sound_speed /2;

if(distance<=100)

{

PublishData(distance);

delay(1000);

if (!client.loop())

{

mqttconnect();

}

digitalWrite(ledpin, HIGH);

Serial.println("Alert !!");

Serial.println(distance);

}

else

{

digitalWrite(ledpin, LOW);

}

delay(10); // this speeds up the simulation

}

// Retrieving to Cloud
```

```
void PublishData(float distance)

{

mqttconnect();// Function call for connecting to ibm


// creating the String in in form JSon to update the data to ibm cloud

String payload = "{\"Alert !! \": ";

payload += distance;

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())

{
```

```cpp
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: ");
```

```cpp
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=="lighton")

{

Serial.println(data3);

}

else

{

Serial.println(data3);

}

data3="";

}
```

```
53    if (...)
54    {
55      mqttconnect();
56    }
57    digitalWrite(ledpin, HIGH);
58    Serial.println("Alert !!");
59    Serial.println(distance);
60  }
61  else
62  {
63    digitalWrite(ledpin, LOW);
64  }
65  delay(1R); // this speeds up the simulation
66 }
67
68 // Retrieving to Cloud
69
70 void PublishData(float distance)
71 {
72   mqttconnect();// Function call for connecting to ibm
73   // creating the String in in form ibm to update the data to ibm cloud
74   String payload = "{\"Alert !!  \": ";
75   payload += distance;
76   payload += "}";
77   Serial.print("Sending payload : ");
78   Serial.println(payload);
79   if (client.publish(publishTopic, (char*) payload.c_str()))
80   {
```

```
78    if (client.publish(publishTopic, (char*) payload.c_str()))
79    {
80      Serial.println("Publish ok");// If it successfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publi
81    }
82    else
83    {
84      Serial.println("Publish failed");
85    }
86 }
87
88 void mqttconnect()
89 {
90   if (!client.connected())
91   {
92     Serial.print("Reconnecting client to ");
93     Serial.println(server);
94     while (!!!client.connect(clientId, authMethod, token))
95     {
96       Serial.print(".");
97       delay(500);
98     }
99     InitManagedDevice();
100    Serial.println();
101  }
102 }
103
104 void wificonnect() // Function definition for wificonnect
105 {
```

```
185   void WifiConnect() // function to obtain connection
186   {
187     Serial.println();
188     Serial.print("Connecting to ");
189     WiFi.begin("Wokwi-GUEST", "", 6);// Passing the wifi credentials to establish the connection
189     while (WiFi.status() != WL_CONNECTED)
110     {
111       delay(500);
112       Serial.print(".");
113     }
114     Serial.println("");
115     Serial.println("WiFi connected");
116     Serial.println("IP address: ");
117     Serial.println(WiFi.localIP());
118   }
119
120   void initManagedDevice()
121   {
122     if (client.subscribe(subscribetopic))
123     {
124       Serial.println((subscribetopic));
125       Serial.println("subscribe to cmd OK");
126     }
127     else
128     {
129       Serial.println("subscribe to cmd FAILED");
130     }
131   }
```

```
117     else
118     {
119       Serial.println("subscribe to cmd FAILED");
130     }
131   }
132
133   void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
134   {
135     Serial.print("callback invoked for topic: ");
136     Serial.println(subscribetopic);
137     for (int i = 0; i < payloadLength; i++)
138     {
139       //Serial.print((char)payload[i]);
140       data3 += (char)payload[i];
141     }
142     Serial.println("data: "+ data3);
143     if(data3=="lighton")
144     {
145       Serial.println(data3);
146     }
147     else
148     {
149       Serial.println(data3);
150     }
151     data3="";
152   }
```