

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribtopic,byte* payload, unsigned int payloadLength);
4 #define ORG "tca72r"
5 #define DEVICE_TYPE "raspberrypi"
6 #define DEVICE_ID "12345"
7 #define TOKEN "12345678"
8 String data3;
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/Data/fmt/json";
12 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
16
17 WiFiClient wifiClient;
18 PubSubClient client(server,1883,callback,wifiClient);
19
20 #define ECHO_PIN 12
21 #define TRIG_PIN 13
22 #define led 14
23
24 void setup() {
25   // put your setup code here, to run once:
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(TRIG_PIN, OUTPUT);
29   pinMode(ECHO_PIN, INPUT);
30   wifiConnect();
31   mqttConnect();
32 }
33 float readDistanceCM() {
34   digitalWrite(TRIG_PIN, LOW);
35   delayMicroseconds(2);

```

Simulation

00:06.132 87%

Restart the simulation

```

publish ok
Measured distance: 83.00
Sending payload:{"ALERT":83.00}
publish ok
Measured distance: 42.00
Sending payload:{"ALERT":42.00}
publish ok

```

WOKWI

sketch.ino diagram.json libraries.txt Library Manager

```

31 mqttconnect();
32 }
33 float readDistanceCM() {
34   digitalWrite(TRIG_PIN, LOW);
35   delayMicroseconds(2);
36   digitalWrite(TRIG_PIN, HIGH);
37   delayMicroseconds(10);
38   digitalWrite(TRIG_PIN, LOW);
39   int duration=duration(1,200);
40   //Serial.println(duration);
41   //duration = pulseIn(ECHO_PIN, HIGH);
42   return duration ;
43   //Serial.println(duration);
44 }
45 }
46
47 void loop() {
48   float distance = readDistanceCM();
49   //Serial.println(distance);
50
51   bool isNearby = distance < 100;
52   digitalWrite(led, isNearby);
53
54   Serial.print("Measured distance: ");
55   Serial.println(distance);
56   if(distance<100){
57     PublishData2(distance);
58   }
59   }else{
60     PublishData1(distance);
61   }
62 }
63 //PublishData(distance);
64 delay(1000);

```

Simulation

00:13.147 94%

publish ok
Measured distance: 90.00
Sending payload:{"ALERT":90.00}
publish ok
Measured distance: 191.00
Sending payload:{"distance":191.00}
publish ok

WOKWI

sketch.ino diagram.json libraries.txt Library Manager

```

63 //PublishData(distance);
64 delay(1000);
65 if(!client.loop()){
66   mqttconnect();
67 }
68
69 //delay(2000);
70 }
71 void PublishData1(float dist){
72   mqttconnect();
73   String payload= "{"distance\":";
74   payload += dist;
75   payload+="}";
76
77   Serial.print("Sending payload:");
78   Serial.println(payload);
79
80   if(client.publish(publishTopic,(char*)payload.c_str())){
81     Serial.println("publish ok");
82   } else{
83     Serial.println("publish failed");
84   }
85 }
86 void PublishData2(float dist){
87   mqttconnect();
88   String payload= "{"ALERT\":";
89   payload += dist;
90   payload+="}";
91
92   Serial.print("Sending payload:");
93   Serial.println(payload);
94
95   if(client.publish(publishTopic,(char*)payload.c_str())){
96     Serial.println("publish ok");

```

Simulation

00:20.978 99%

publish ok
Measured distance: 123.00
Sending payload:{"distance":123.00}
publish ok
Measured distance: 68.00
Sending payload:{"ALERT":68.00}
publish ok

WOKWI

sketch.ino

```

96 Serial.println("publish ok");
97 } else{
98   Serial.println("publish failed");
99 }
100 }
101 void mqttconnect(){
102   if(!client.connected()){
103     Serial.print("Reconnecting to");
104     Serial.println(server);
105     while(!client.connect(clientID, authMethod, token)){
106       Serial.print(".");
107       delay(500);
108     }
109     initManagedDevice();
110     Serial.println();
111   }
112 }
113 }
114 void wificonnect(){
115   Serial.println();
116   Serial.print("Connecting to");
117   WiFi.begin("Wokwi-GUEST","",6);
118   while(WiFi.status()!=WL_CONNECTED){
119     delay(500);
120     Serial.print(".");
121   }
122   Serial.println("");
123   Serial.println("WIFI CONNECTED");
124   Serial.println("IP address:");
125   Serial.println(WiFi.localIP());
126 }
127 }
128 }
129 void initManagedDevice(){
130   if(client.subscribe(subscribeTopic)){

```

Simulation

00:27.692 100%

publish ok
Measured distance: 154.00
Sending payload:{"distance":154.00}
publish ok
Measured distance: 143.00
Sending payload:{"distance":143.00}
publish ok

WOKWI

sketch.ino

```

129 void initManagedDevice(){
130   if(client.subscribe(subscribeTopic)){
131     Serial.println(subscribeTopic);
132     Serial.println("subscribe to cmd ok");
133   }else{
134     Serial.println("subscribe to cmd failed");
135   }
136 }
137 }
138 void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
139   Serial.print("callback invoked for topic:");
140   Serial.println(subscribeTopic);
141   for(int i=0; i<payloadLength; i++){
142     data3 += (char)payload[i];
143   }
144   Serial.println("data:"+ data3);
145   if(data3=="lighton"){
146     Serial.println(data3);
147     digitalWrite(led,HIGH);
148   }else{
149     Serial.println(data3);
150     digitalWrite(led,LOW);
151   }
152   data3="";
153 }

```

Simulation

00:32.857 101%

publish ok
Measured distance: 13.00
Sending payload:{"ALERT":13.00}
publish ok
Measured distance: 116.00
Sending payload:{"distance":116.00}
publish ok

Inbox (35) - bas...Welcome to Pro...IBMService Details - IBM Watson IoTIBM Watson IoTsketch.imo - Woll

tca72r.internetofthings.ibmcloud.com/dashboard/devices/browse

GmailYouTubeMaps

IBM Watson IoT Platform

BrowseActionDevice TypesInterfaces

Device IDStatusDevice Type

12345Disconnectedraspberrypi

IdentityDevice InformationRecent EventsState

The recent events listed show the live stream of data that is coming and going

Event	Value
event_1	{"randomNumber":10}
Data	{"ALERT":59}
event_1	{"randomNumber":27}
Data	{"distance":106}
Data	{"distance":155}

Device Type: raspberrypi

Events1New event type

Event type nameevent_1Send

Schedule20Every Minute

PayloadSpecify the event payload in the editor window or by uploading a CSV file

1{"randomNumber": random(0, 100)}

2}

3

Upload a CSV file

What functions can I apply?

CancelSave

Scanned with CamScanner