

NAME:G.MANORANJITHAM

ROLL NO.;92211916054

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "nws9km"
#define DEVICE_TYPE "CROP"
#define DEVICE_ID "MANORANJITHAM"
#define TOKEN "manogopal"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

void publishData();

const int trigpin=5;
const int echopin=18;

String command;
String data="";

long duration;

int dist;

void setup()
{
```

```
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
wifiConnect();
mqttConnect();
}

void loop() {
publishData();
delay(500);
if (!client.loop()) {
mqttConnect();
}
}

void wifiConnect() {
Serial.print("Connecting to "); Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}

Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}

void mqttConnect() {
if (!client.connected()) {
Serial.print("Reconnecting MQTT client to "); Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(1000);
}
}
```

```
initManagedDevice();  
Serial.println();  
}  
}  
  
void initManagedDevice() {  
  if (client.subscribe(topic)) {  
    Serial.println(client.subscribe(topic));  
    Serial.println("subscribe to cmd OK");  
  } else {  
    Serial.println("subscribe to cmd FAILED");  
  }  
}  
  
void publishData()  
{  
  digitalWrite(trigpin,LOW);  
  digitalWrite(trigpin,HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigpin,LOW);  
  duration=pulseIn(echopin,HIGH);  
  dist=duration*speed/2;  
  if(dist<100){  
    DynamicJsonDocument doc(1024);  
    String payload;  
    doc["Distance Alert:"]=dist;  
    serializeJson(doc, payload);  
    delay(3000);  
    Serial.print("\n");  
    Serial.print("Sending payload: ");  
    Serial.println(payload);
```

```

if (client.publish(publishTopic, (char*) payload.c_str())) {

Serial.println("Publish OK");

} else {

Serial.println("Publish FAILED");

}

}

}

}

```

WokwiLink:

<https://wokwi.com/projects/347295666195661396>

Output:

The screenshot displays the Wokwi IDE interface. On the left, the sketch editor shows a C++ program for an ESP32. The code includes libraries for WiFi and PubSubClient, defines constants for the server, topic, and token, and sets up the HC-SR04 ultrasonic sensor. The main loop publishes the sensor's distance data to the IBM Watson IoT Platform. On the right, the simulation window shows the physical components: an ESP32 microcontroller and an HC-SR04 sensor. A dialog box indicates the sensor's distance is 62cm. The serial output window shows the successful transmission of a JSON payload: {"Distance Alert":61}.

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4 WiFiClient wifiClient;
5 #define ORG "nws9km"
6 #define DEVICE_TYPE "CROP"
7 #define DEVICE_ID "MANORANJITHAM"
8 #define TOKEN "manogopal"
9 #define speed 0.034
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/Data/fmt/json";
12 char topic[] = "iot-2/cmd/home/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16 PubSubClient client(server, 1883, wifiClient);
17 void publishData();
18 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 int dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(trigpin, OUTPUT);
28   pinMode(echopin, INPUT);
29   wifiConnect();

```

Simulation

00:10.808 98%

Editing Ultrasonic Distance Sensor

Distance: 62cm

Sending payload: {"Distance Alert":61}
Publish OK

Sending payload: {"Distance Alert":61}
Publish OK

26°C Cloudy 3:18 AM 11/3/2022

IBM Cloud:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Welcome to Project!', 'IBM', '(5) WhatsApp', 'sketchino - Wokwi A...', and two instances of 'IBM Watson IoT Platf...'. The browser address bar shows the URL 'nws9km.internetofthings.ibmcloud.com/dashboard/devices/browse'. The main header of the platform shows the user 'gopelmano58@gmail.com' with ID 'nws9km'. The left sidebar contains icons for various functions. The main content area is titled 'MANORANJITHAM' and shows the device is 'Disconnected'. The 'Recent Events' tab is selected, displaying a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. All events are of type 'Data' with a value of '{"Distance Alert":"61"}' in 'json' format, received 'a few seconds ago'. A status box at the bottom right indicates '1 Simulation running'. The Windows taskbar at the bottom shows the system time as 3:18 AM on 11/3/2022, with a temperature of 26°C and a cloudy weather forecast.

IBM Watson IoT Platform

gopelmano58@gmail.com
ID: nws9km

Browse Action Device Types Interfaces

Add Device +

MANORANJITHAM Disconnected CROP Device 3 Nov 2022 02:35

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
Data	{"Distance Alert":"61"}	json	a few seconds ago
Data	{"Distance Alert":"61"}	json	a few seconds ago
Data	{"Distance Alert":"61"}	json	a few seconds ago
Data	{"Distance Alert":"61"}	json	a few seconds ago
Data	{"Distance Alert":"61"}	json	a few seconds ago

1 Simulation running

26°C Cloudy 3:18 AM 11/3/2022