

# Smart Farmer-IOT Enabled Smart Farming Application

## Assignment -4

TEAM ID : PNT2022TMID29459

### Question-1:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud

### CODE :

```
#include <WiFi.h>
#include <PubSubClient.h> void callback(char* subscribetopic, byte* payload,
unsigned int payloadLength);
#define ORG "92zbfc"
#define DEVICE_TYPE "esp32"
#define DEVICE_ID "12345"
#define TOKEN "12345678" String data3; char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/Data/fmt/json"; char subscribetopic[]
= "iot-2/cmd/test/fmt/String"; char authMethod[] = "use-token-
```

```

auth"; char token[] = TOKEN; char clientId[] = "d:" ORG ":"
DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5; const int echoPin = 18; #define
SOUND_SPEED 0.034 long duration; float distance;
void setup() { Serial.begin(115200); pinMode(trigPin,
OUTPUT); pinMode(echoPin, INPUT); wificonnect();
mqttconnect();
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW); duration
= pulseIn(echoPin, HIGH); distance
= duration * SOUND_SPEED/2;
Serial.print("Distance (cm): ");
Serial.println(distance); if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000); if
(!client.loop()) {
mqttconnect();
} }
delay(1000)
; }
void PublishData(float dist) { mqttconnect();
String payload = "{\"Distance\": "; payload += dist; payload
+= ", \"ALERT!!\": \"\" \"Distance less than 100cms\"";
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
} 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()
{
  Serial.println();
  Serial.print("Connecting to "); WiFi.begin("Wokwi-GUEST", "", 6); 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++)
  {
  data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3); data3="";
}

```

Wokwi Link :

<https://wokwi.com/projects/345395196387656275>

Output and Simulation :

The screenshot displays the Wokwi online IDE interface. On the left, the 'sketch.ino' file is open, showing an Arduino sketch that includes libraries for WiFi, MQTT, and DHT11. The sketch defines a DHT sensor (DHT22) and a DHT pin (15). It creates a DHT instance and defines a callback function. The sketch also includes credentials for an IBM Watson IoT Platform account, such as the organization ID, device type, device ID, and token. The sketch is configured to send temperature and humidity data to the IoT platform.

On the right, the 'Simulation' window shows a visual representation of the ESP32 microcontroller and the DHT22 sensor. Below the simulation, the output console displays the following messages:

```
Humid:40.00
Sending payload: {"temp":24.00,"Humid":40.00}
Publish ok
temp:24.00
Humid:40.00
Sending payload: {"temp":24.00,"Humid":40.00}
Publish ok
```

The bottom status bar indicates a data transfer rate of 0.16 KB/s and 0.28 KB/s, along with the time 09:59 PM and date 02-11-2022.

Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

ID: 922bfc

Delete

1 item selected Cancel

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	esp32	Device	Nov 1, 2022 9:53 PM	

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":72.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":72.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":72.96,"ALERT!!":"Distance less than ...	json	a few seconds ago

> 2001 Disconnected raspberrypi Device Oct

Items per page 50 | 1-2 of 2 items

0 Simulations running