

Smart Farmer - IoT Enabled Smart Farming Application

Batch No	PNT2022TMID21910
Student Roll number	142219205011

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 1 :

```
#include <WiFi.h>
#include <PubSubClient.h> void callback(char* subscribtopic, 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 subscribtopic[] =
"iot2/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/347141309997253202>

Output and Simulation :

The screenshot displays the Wokwi web interface. On the left, the 'sketch.ino' file is open, showing the Arduino code. The code includes headers for `<WiFi.h>` and `<PubSubClient.h>`, defines a callback function, and sets up an ESP32 device with specific credentials and a token. It also defines a trig pin and a sound speed constant. The `setup()` function initializes the serial port and the WiFi connection. The `loop()` function writes a low signal to the trig pin, delays for 2 microseconds, and then reads the distance from the HC-SR04 ultrasonic sensor. The distance is then printed to the serial monitor.

On the right, the 'Simulation' window shows a visual representation of the ESP32 board and the HC-SR04 sensor. The sensor's distance is displayed as 73cm. Below the simulation, the serial monitor shows the output of the code, including the distance reading and the alert message.

```

ALERT!!
Sending payload: {"Distance":72.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 72.96
ALERT!!
Sending payload: {"Distance":72.96,"ALERT!!":"Distance less than 100cms"}
Publish ok

```

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

sketchuno - Wokwi Arduino sim...sketchuno - Wokwi Arduino sim...IBMService Details - IBM CloudIBM Watson IoT Platform

92zbfc.internetofthings.ibmcloud.com/dashboard/devices/browse

GmailYouTubeMapsNew TabCONNECT Exams: L...NextStep- Tata Con...TCS LaunchpadK.L.N COLLEGE OF...Sci-Hub: common...Wokwi - Online Ard...Webmail Loginbatch9.aia - Google...

IBM Watson IoT Platform910619106069@smartinternz.comID: 92zbfc

Delete1 item selectedCancel

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

IdentityDevice InformationRecent EventsStateLogs

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

>2001DisconnectedraspberrypiDeviceOct

Items per page 501-2 of 2 items

0 Simulations running

Type here to search

10:06 PM01-11-2022