ASSIGNMENT TASK - 4

Assignment date	**29 October 2022**
Project name	**Iot Based Smart Crop Protection System for Agriculture**
Team ID	**PNT2022TMID01702**
Maximum mark	**2 Marks**

QUESTION 1:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100cms send "alert" to ibm cloud and display in device recent events.

CODE:

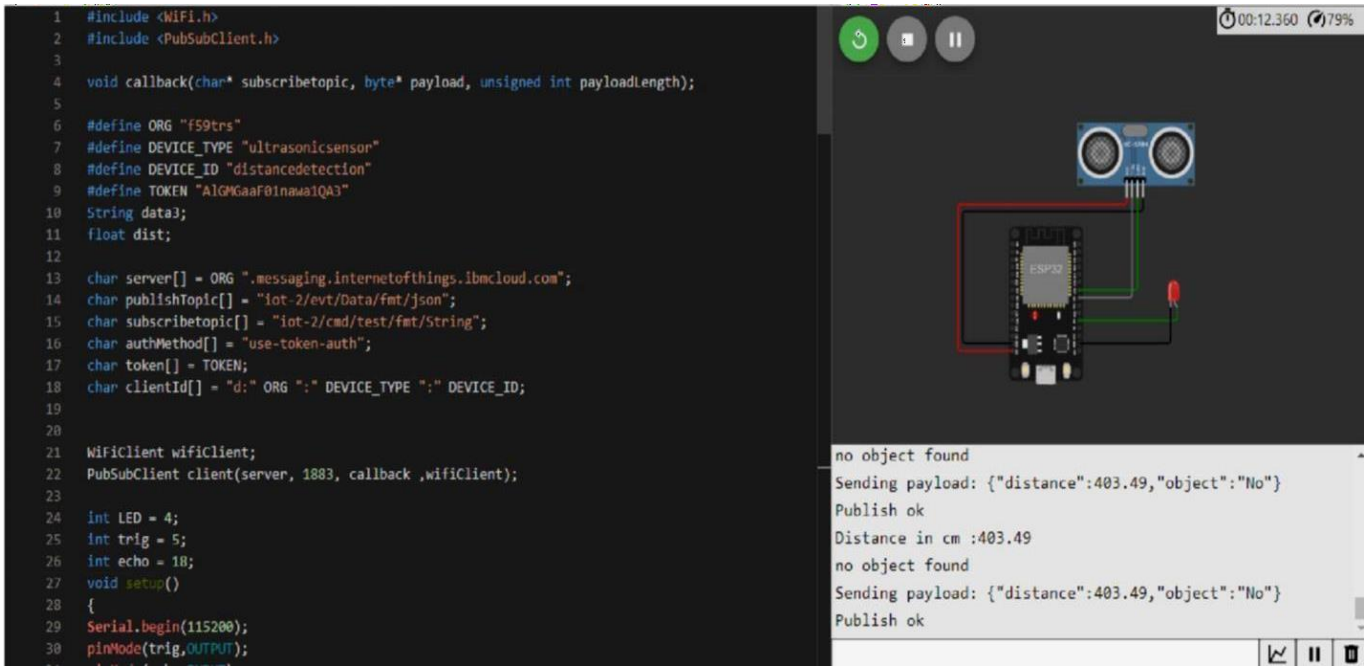
```
#include <WiFi.h>
#include <PubSubClient.h>
 void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
#define ORG "akptwo"
#define DEVICE_TYPE "ESP32_Controller"
#define DEVICE_ID "BME280_Sensor"
#define TOKEN "pySeb&4Lc@4tEHID(n"
String data3; float dist;
 char server[] = 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);
 int LED = 4;
int trig = 5;
int echo = 18;
void setup()
{
Serial.begin(115200);
pinMode(trig,OUTPUT);
```

```
pinMode(echo,INPUT);
pinMode(LED, OUTPUT);
delay(10);
wificonnect();
mqttconnect(); } void
loop()
{   digitalWrite(trig,LOW);
digitalWrite(trig,HIGH);
delayMicroseconds(10);
digitalWrite(trig,LOW);   float
dur = pulseIn(echo,HIGH);   float
dist = (dur * 0.0343)/2;
  Serial.print ("Distance in cm :");
  Serial.println(dist);

  PublishData(dist);
delay(1000);   if
(!client.loop()) {
mqttconnect();
  }
}
void PublishData(float dist) {
mqttconnect();   String
object;   if (dist <100)   {
    digitalWrite(LED,HIGH);
Serial.println("object is near");     object
= "Near";
  }
else   {
digitalW
rite(LED
,LOW);
Serial.p
rintln("
no
object
found");
object =
"No";
  }

  String payload = "{\"distance\":";
payload += dist;   payload += ","
"\"object\":\"";   payload +=
object;   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];
  } data3="";
  }
```

OUTPUT:

When object is nearer to Ultrasonic sensor



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

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

#define ORG "f59trs"
#define DEVICE_TYPE "ultrasonicsensor"
#define DEVICE_ID "distancedetection"
#define TOKEN "AlGMGaaF01naWa1QA3"
String data3;
float dist;

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);

int LED = 4;
int trig = 5;
int echo = 18;
void setup()
{
Serial.begin(115200);
pinMode(trig,OUTPUT);
```

```
no object found
Sending payload: {"distance":403.49,"object":"No"}
Publish ok
Distance in cm :403.49
no object found
Sending payload: {"distance":403.49,"object":"No"}
Publish ok
```

Data sent to the ibm cloud when the object is near