

SPRINT - 2

DATE	30 OCTOBER 2022
TEAM ID	PNT2022TMID25296
PROJECT NAME	SMART WASTE MANAGEMENT FOR METROPOLITANCITIES-IOT

CODE :

```
#include <WiFi.h>                // library
for wifi #include <PubSubClient.h>
                                // library

for MQTT#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

//----- credentials of IBM Accounts -----

#define ORG "wjmfdn"            // IBM organisation id
#define DEVICE_TYPE "abcd"      // Device type mentioned in ibm watson
iot platform#define DEVICE_ID "1234"          // Device ID
mentioned in ibm watson iot platform #define TOKEN "12345678" //
Token

//----- customise above values -----

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name

char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform
and format inwhich data to be send

char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is
test format ofstrings

char authMethod[] = "use-token-auth"; //

authentication methodchar token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id

//-----

WiFiClient wifiClient; // creating instance

for wificlientPubSubClient client(server, 1883, wifiClient);

#define ECHO_PIN 12

#define

TRIG_PI

N 13
```

```
float
dist;
void setup()
{
    Serial.begin(115200);
    pinMode(LED_BUILTIN,
    OUTPUT);
    pinMode(TRIG_PIN,
    OUTPUT);
    pinMode(ECHO_PIN,
    INPUT);
    //pir pin

    pinMode(34, INPUT);
    //ledpins
    pinMode(2
    3,
    OUTPUT);
    pinMode(2
    , OUTPUT);
    pinMode(4
    , OUTPUT);
    pinMode(1
    5,
    OUTPUT);
    lcd.init();
    lcd.backlig
    ht();
    lcd.setCurs
    or(1, 0);
    lcd.print("")
    );
    wifiConnec
    t());
```

```

    mqttConne
    ct();
}
float readcmCM()
{
    digitalWrite(TRIG
    _PIN, LOW);
    delayMicrosecon
    ds(2);
    digitalWrite(TRIG
    _PIN, HIGH);
    delayMicrosecon
    ds(10);
    digitalWrite(TRIG
    _PIN, LOW);
    int duration =
    pulseIn(ECHO_PIN, HIGH);
    return duration * 0.034 / 2;
}
void loop()
{
    lc
    d.
    cl
    ea
    r()
    ;
    p
    u
    bli
    sh
    D
    at
    a(

```

```

);
de
la
y(
50
0)
;
if (!client.loop())
{
    mqttConnect();          // function call to connect to IBM
}
}
/* -----retrieving to cloud----- */
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))
    {
        Seri
        al.p
        rint
        ("."
        );
        del
        ay(
        500
        );
    }
    initManagedD
    evice();
    Serial.println()
    ;
}
}

void initManagedDevice()
{
    if (client.subscribe(topic))
    {
        Serial.println("IBM subscribe to cmd OK");
    }
    else
    {
        Serial.println("subscribe to cmd FAILED");
    }
}

void publishData()

```

```

{
    float cm = readcmCM();
    if(digitalRead(34))                //pir motion detection
    {
        Serial.println("Motion
        Detected");
        Serial.println("Lid
        Opened");
        digitalWrite(15, HIGH);
        if(digitalRead(34)==
        true)
        {
            if(cm <= 60)                //Bin level detection
            {
                digitalWrite(2, HIGH);
                Serial.println("High Alert!!!,Trash bin is about
                to be full");Serial.println("Lid Closed");
                lcd.print("Full!
                Don't use");
                delay(2000);
                lcd.clear();
                digitalWrite
                e(4, LOW);
                digitalWrite
                e(23,
                LOW);
            }
            else if(cm > 60 && cm < 120)
            {
                digitalWrite(4, HIGH);
                Serial.println("Warning!!,Trash is about to cross 50%
                of bin level");digitalWrite(2, LOW);
                digitalWrite(23, LOW);
            }
        }
    }
}

```

```
}  
else if(cm > 120)  
{  
    digitalWrite(23,  
    HIGH);  
    Serial.println("Bin is  
    available");  
    digitalWrite(2,LOW)  
    ; digitalWrite(4,  
    LOW);  
}  
    delay(10000);  
    Serial.println("  
    Lid Closed");  
  
}  
else  
{  
    Serial.println("No motion  
    detected");  
    digitalWrite(2, LOW);  
    digitalWrite(15, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(23, LOW);  
}  
}  
else  
{  
    digitalWrite(15, LOW);  
}  
if(cm <= 60)  
{  
    digitalWrite(21,HIGH);
```

```

String payload =
"{\"High_Alert\":\"";
payload += cm;
payload += " }";
Serial.print("\n");
Serial.print("Sending
payload: ");
Serial.println(payload);

d);

if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud
successfully,prints publish okelse prints publish failed
{
Serial.println("Publish OK");
}
}
else if(cm <= 120)
{
digitalWrite(22,HIGH);
String payload =
"{\"Warning\":\"";
payload += cm ;
payload
+= " }";
Serial.print
nt("\n");

Serial.print("Sending
payload: ");
Serial.println(payload);
d);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}

```



```

}
else
{
  Serial.println("Publish FAILED");
}
}

```

```

else
{
  Serial.println();
}

```

```
float inches = (cm / 2.54);
```

/

```
/print on lcd lcd.setCursor(0,0);
```

```
lcd.prin
```

```
t("Inch
```

```
es");
```

```
lcd.set
```

```
Cursor(
```

```
4,0);
```

```
lcd.set
```

```
Cursor(
```

```
12,0);
```

```
lcd.prin
```

```
t("cm")
```

```
;
```

```
lcd.set
```

```
Cursor(
```

```
1,1);
```

```
lcd.prin
```

```
t(inche
```

```
s, 1);
```

```
lcd.set
```

```
Cursor(
```

```
11,1);
```

lcd.prin

t(cm,

1);

lcd.set

Cursor(

14,1);

delay(1

000);

lcd.clea

r());

}

CIRCUIT :

The screenshot displays the Wokwi IDE interface. On the left, the Arduino IDE editor shows a C++ sketch for an ESP32. The sketch includes libraries for WiFi, MQTT, and a liquid crystal display (LCD). It defines credentials for IBM Watson IoT and sets up a PubSubClient for MQTT communication. The setup function initializes the serial port, LED, buzzer, and LCD. The main loop prints the received message to the serial monitor and the LCD, and triggers the buzzer.

```
1 #include <WiFi.h> // library for wifi
2 #include <PubSubClient.h> // library for MQTT
3 #include <LiquidCrystal_I2C.h>
4 LiquidCrystal_I2C lcd(0x27, 20, 4);
5
6 //----- credentials of IBM Accounts -----
7
8 #define ORG "cbseji" // IBM organisation id
9 #define DEVICE_TYPE "abcd" // Device type mentioned in ibm watson iot p
10 #define DEVICE_ID "1234" // Device ID mentioned in ibm watson iot platform
11 #define TOKEN "12345678" // Token
12
13 //----- customise above values -----
14
15 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // serve
16 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic
17 char topic[] = "iot-2/cmd/led/fmt/String"; // cmd R
18 char authMethod[] = "use-token-auth"; // auth
19 char token[] = TOKEN;
20 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client
21
22 //-----
23
24 WiFiClient wifiClient;
25 PubSubClient client(server, 1883, wifiClient); // creatin
26
27 #define ECHO_PIN 12
28 #define TRIG_PIN 13
29 float dist;
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(LED_BUILTIN, OUTPUT);
35   pinMode(TRIG_PIN, OUTPUT);
36   pinMode(ECHO_PIN, INPUT);
```

On the right, the simulation window shows a 3D model of the circuit. An ESP32 microcontroller is connected to an LCD display and a buzzer. The LCD is connected to the ESP32's I2C pins (SCL to pin 22, SDA to pin 21). The buzzer is connected to the ESP32's GPIO pins (VCC to pin 5, GND to pin 4, and TRIG to pin 13). The ESP32 is also connected to a USB-C port for power and data.

