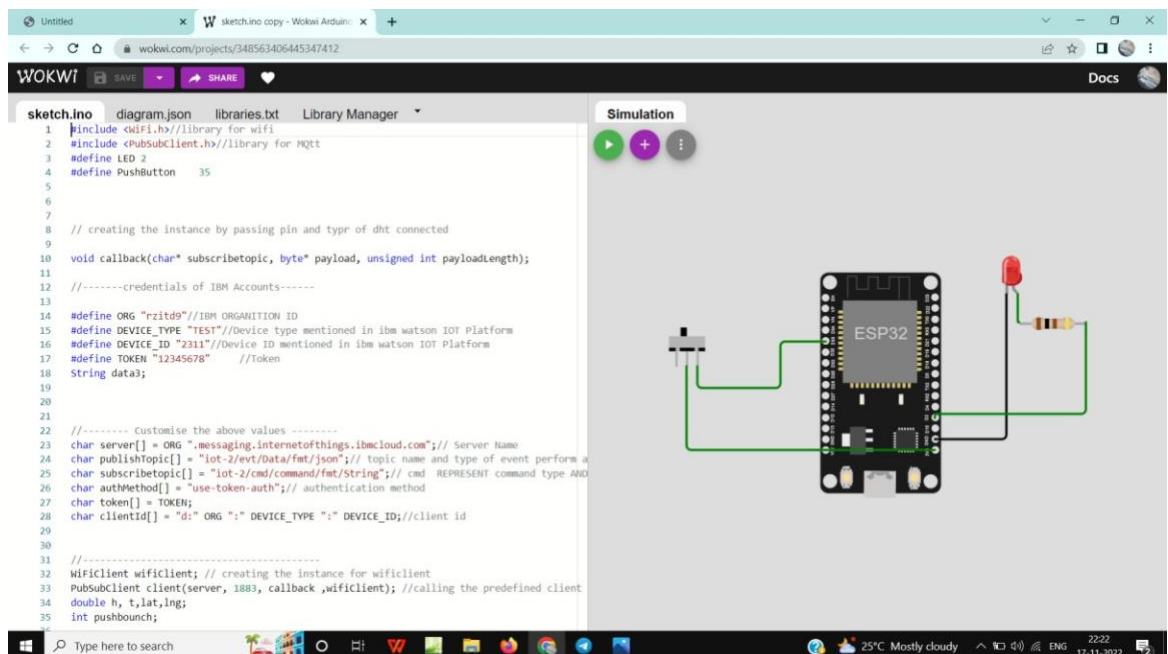


## SPRINT 1

Project Topic : IoT based Safety Gadget for child Safety Monitoring and Notification

TeamID : PNT2022TMID18967

- We Connect the ESP32 with a LED and a PUSHBUTTON. and a Resister.
- And Writing Code for that.
- From this Simulation We can get Message in Our IBM Cloud Device on Recent Tabs.

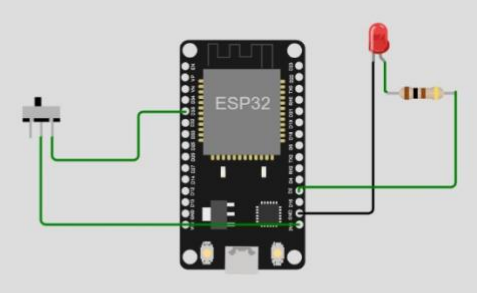


WOKWI

sketch.ino diagram.json libraries.txt Library Manager

```
37 void setup()// configuring the ESP32
38 {
39   Serial.begin(115200);
40   pinMode(LED,OUTPUT);
41   pinMode(PushButton, INPUT);
42   delay(10);
43   Serial.println();
44   wifiConnect();
45   mqttConnect();
46 }
47
48 void loop()// Recursive Function
49 {
50
51   h = 11.12756;
52   t = 7.777666;
53   lat = 9.10232;
54   lng = 10.10232;
55   pushbounch = digitalRead(PushButton);
56
57   PublishData(t, h);
58   delay(1000);
59   if (!client.loop()) {
60     mqttConnect();
61   }
62 }
63
64
65
66 /*.....retrieving to Cloud.....
67
68 void PublishData(double LAT, double LONG) {
69   mqttConnect();//function call for connecting to ibm
70   /*
71   creating the String in in form JSON to update the data to ibm cloud
```

Simulation

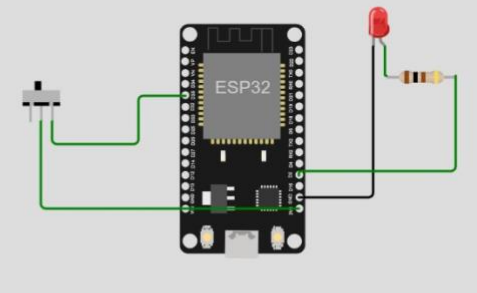


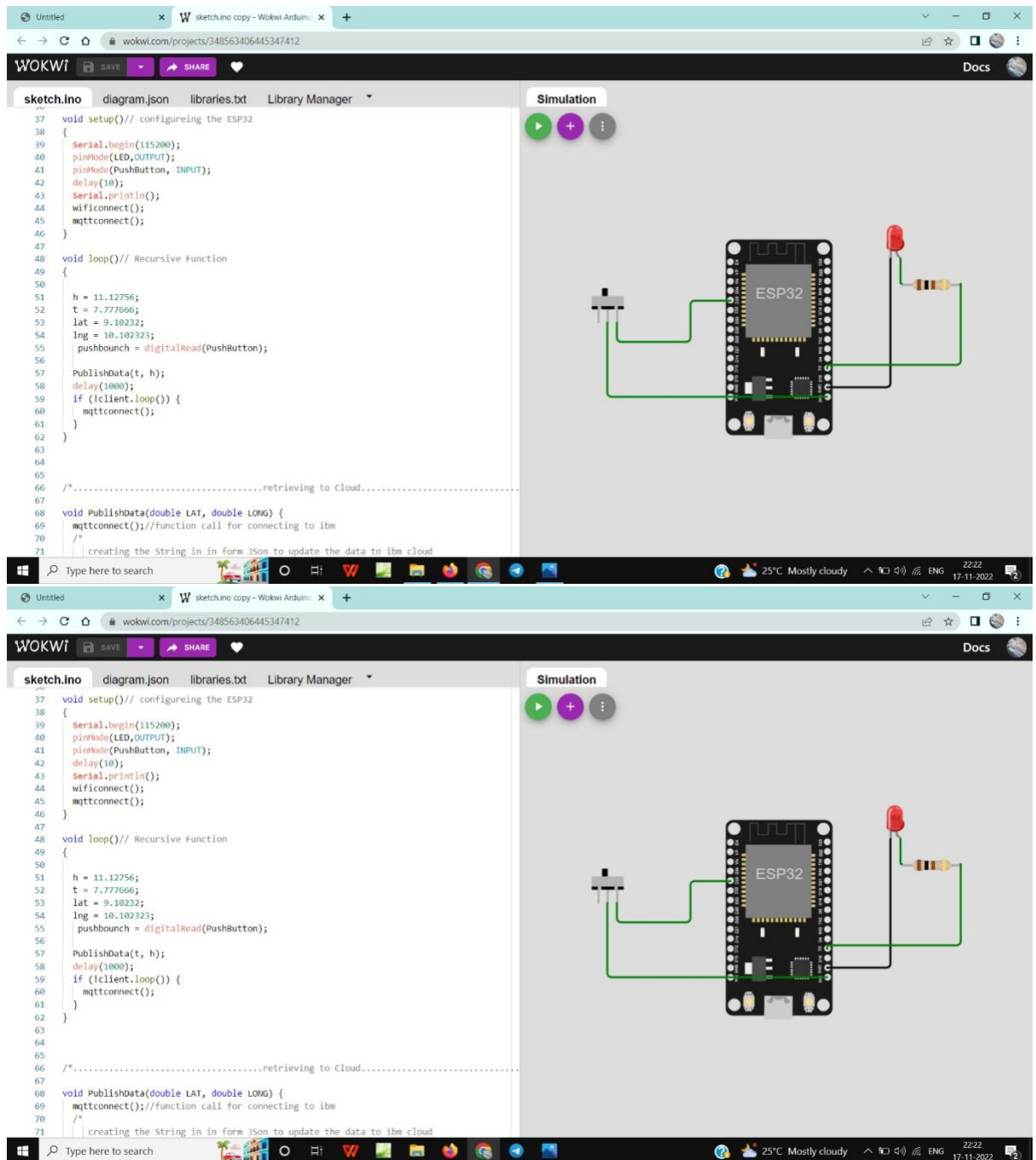
WOKWI

sketch.ino diagram.json libraries.txt Library Manager

```
37 void setup()// configuring the ESP32
38 {
39   Serial.begin(115200);
40   pinMode(LED,OUTPUT);
41   pinMode(PushButton, INPUT);
42   delay(10);
43   Serial.println();
44   wifiConnect();
45   mqttConnect();
46 }
47
48 void loop()// Recursive Function
49 {
50
51   h = 11.12756;
52   t = 7.777666;
53   lat = 9.10232;
54   lng = 10.10232;
55   pushbounch = digitalRead(PushButton);
56
57   PublishData(t, h);
58   delay(1000);
59   if (!client.loop()) {
60     mqttConnect();
61   }
62 }
63
64
65
66 /*.....retrieving to Cloud.....
67
68 void PublishData(double LAT, double LONG) {
69   mqttConnect();//function call for connecting to ibm
70   /*
71   creating the String in in form JSON to update the data to ibm cloud
```

Simulation





After Simulation the Device is getting Connected to the IBM Cloud . And Whenever we ON,

- the Push Button the LED gets ON ,
- the Alert Message will received by the Cloud on the Recent Events.

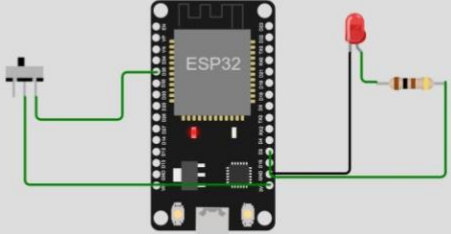
Wokwi

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define LED 2
4 #define PushButton 35
5
6
7
8 // creating the instance by passing pin and type of dht connected
9
10 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
11
12 //-----credentials of IBM Accounts-----
13
14 #define ORG "rzitd9"//IBM ORGANIZATION ID
15 #define DEVICE_TYPE "TEST"//Device type mentioned in ibm watson IOT Platform
16 #define DEVICE_ID "2311"//Device ID mentioned in ibm watson IOT Platform
17 #define TOKEN "12345678" //Token
18 String data3;
19
20
21
22 //----- Customise the above values -----
23 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
24 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event performed
25 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type
26 char authMethod[] = "use-token-auth"; // authentication method
27 char token[] = TOKEN;
28 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
29
30
31 //-----
32 WiFiClient wificlient; // creating the instance for wificlient
33 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
34 double h, t, lat, lng;
35 int pushbounc;
36
```

Simulation

00:02.997 104%



Connecting to ....  
Wifi connected  
IP address:  
10.10.0.2  
Reconnecting client to rzitd9.messaging.internetofthings.ibmcloud.com

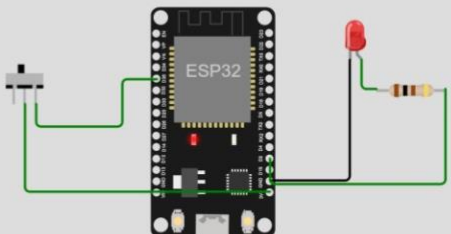
Wokwi

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define LED 2
4 #define PushButton 35
5
6
7
8 // creating the instance by passing pin and type of dht connected
9
10 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
11
12 //-----credentials of IBM Accounts-----
13
14 #define ORG "rzitd9"//IBM ORGANIZATION ID
15 #define DEVICE_TYPE "TEST"//Device type mentioned in ibm watson IOT Platform
16 #define DEVICE_ID "2311"//Device ID mentioned in ibm watson IOT Platform
17 #define TOKEN "12345678" //Token
18 String data3;
19
20
21
22 //----- Customise the above values -----
23 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
24 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event performed
25 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type
26 char authMethod[] = "use-token-auth"; // authentication method
27 char token[] = TOKEN;
28 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
29
30
31 //-----
32 WiFiClient wificlient; // creating the instance for wificlient
33 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
34 double h, t, lat, lng;
35 int pushbounc;
36
```

Simulation

00:02.997 104%



Connecting to ....  
Wifi connected  
IP address:  
10.10.0.2  
Reconnecting client to rzitd9.messaging.internetofthings.ibmcloud.com

Wokwi Arduino IDE interface showing a sketch for an ESP32 connected to a push button and an LED. The sketch configures the ESP32 to send data to an IBM Watson IoT Platform.

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define LED 2
4 #define PushButton 35
5
6
7
8 // creating the instance by passing pin and type of dht connected
9
10 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
11
12 //-----credentials of IBM Accounts-----
13
14 #define ORG "rzitd9"//IBM ORGANIZATION ID
15 #define DEVICE_TYPE "TEST"//Device type mentioned in ibm watson IOT Platform
16 #define DEVICE_ID "2311"//Device ID mentioned in ibm watson IOT Platform
17 #define TOKEN "12345678" //Token
18 String data3;
19
20
21
22 //----- Customise the above values -----
23 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
24 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event performed
25 char subscribetopic[] = "iot-2/cmd/command/fmt/string"; // cmd REPRESENT command type
26 char authMethod[] = "use-token-auth"; // authentication method
27 char token[] = TOKEN;
28 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
29
30
31 //-----
32 WiFiClient wificlient; // creating the instance for wificlient
33 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
34 double h, t, lat, lng;
35 int pushbounc;
36
```

Simulation status: 00:10.290 95%

Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok  
Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok  
Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok

Wokwi Arduino IDE interface showing the same sketch as above, but with a different clientId in the code.

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define LED 2
4 #define PushButton 35
5
6
7
8 // creating the instance by passing pin and type of dht connected
9
10 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
11
12 //-----credentials of IBM Accounts-----
13
14 #define ORG "rzitd9"//IBM ORGANIZATION ID
15 #define DEVICE_TYPE "TEST"//Device type mentioned in ibm watson IOT Platform
16 #define DEVICE_ID "2311"//Device ID mentioned in ibm watson IOT Platform
17 #define TOKEN "12345678" //Token
18 String data3;
19
20
21
22 //----- Customise the above values -----
23 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
24 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event performed
25 char subscribetopic[] = "iot-2/cmd/command/fmt/string"; // cmd REPRESENT command type
26 char authMethod[] = "use-token-auth"; // authentication method
27 char token[] = TOKEN;
28 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
29
30
31 //-----
32 WiFiClient wificlient; // creating the instance for wificlient
33 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
34 double h, t, lat, lng;
35 int pushbounc;
36
```

Simulation status: 00:10.290 95%

Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok  
Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok  
Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok



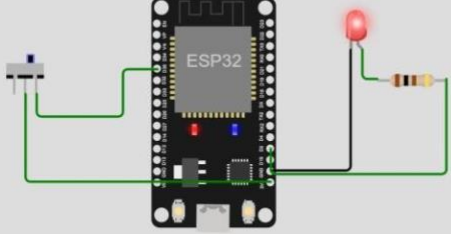
WOKWI

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define LED 2
4 #define PushButton 35
5
6
7
8 // creating the instance by passing pin and type of dht connected
9
10 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
11
12 //-----credentials of IBM Accounts-----
13
14 #define ORG "r4tds" //IBM ORGANIZATION ID
15 #define DEVICE_TYPE "TEST" //device type mentioned in ibm watson IOT Platform
16 #define DEVICE_ID "2311" //Device ID mentioned in ibm watson IOT Platform
17 #define TOKEN "12345678" //Token
18 String data3;
19
20
21
22 //----- Customise the above values -----
23 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
24 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event performed
25 char subscribtopic[] = "iot-2/cmd/command/fmt/string"; // cmd REPRESENT command type
26 char authMethod[] = "use-token-auth"; // authentication method
27 char token[] = TOKEN;
28 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
29
30
31 //-----
32 WiFiClient wifiClient; // creating the instance for wifiClient
33 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client
34 double h, t, lat, lng;
35 int pushbounc;
36
```

Simulation

00:12.105 101%



Publish ok  
Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok  
Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok  
Sending payload: {"Latitude":7.78,"Longitude":11.13}  
Publish ok

Type here to search

25°C Mostly cloudy

22:24  
17-11-2022

IBM Watson IoT Platform

7rp2ov.internetofthings.ibmcloud.com/dashboard/devices/browse

keerthanaarhanari@gmail.com ID: 7rp2ov

Browse Action Device Types Interfaces

Add Device

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Disconnected	NodeMCU	Device	14 Nov 2022 8:55 PM	
12345	Connected	TestDevice	Device	16 Nov 2022 6:00 PM	

Identity Device Information Recent Events State Logs

Device ID: 12345  
Device Type: TestDevice  
Date Added: 16 Nov 2022 6:00 PM  
Added By: keerthanaarhanari@gmail.com  
Connection Status: Connected  
Connection Time: 18 Nov 2022 10:36 AM  
Client Address: 223.228.146.162 SecureToken

```
2022-11-18 10:36:11,635 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:7rp2ov:TestDevice:12345
Data published to IBM Iot Platform: {'name': 'Smartbridge', 'lat': 17.4225176, 'lon': 78.5458842}
Data published to IBM Iot Platform: {'name': 'Smartbridge', 'lat': 17.4225176, 'lon': 78.5458842}
Data published to IBM Iot Platform: {'name': 'Smartbridge', 'lat': 17.4225176, 'lon': 78.5458842}
```

2022-11-18 10:36:11,635 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:7rp2ov:TestDevice:12345
Data published to IBM Iot Platform: {'name': 'Smartbridge', 'lat': 17.4225176, 'lon': 78.5458842}
Data published to IBM Iot Platform: {'name': 'Smartbridge', 'lat': 17.4225176, 'lon': 78.5458842}
Data published to IBM Iot Platform: {'name': 'Smartbridge', 'lat': 17.4225176, 'lon': 78.5458842}
Data published to IBM Iot Platform: {'name': 'Smartbridge', 'lat': 17.4225176, 'lon': 78.5458842}