

## PROJECT DEVELOPMENT PHASE

### PROJECT DEVELOPMENT - DELIVERY OF SPRINT-2

|              |  |
|--------------|--|
| Date         | 16 November 2022   |
| Team ID      | PNT2022TMID34385   |
| Project Name | Project - IoT Based Safety Gadget for Child Safety Monitoring and Notification |

PROGRAM:

PYTHON CODE:

```
import json

import wiotp.sdk.device

import time

myconfig = {
    "identity": {
        "orgId": "a7011a",
        "typeId": "IOT",
        "deviceId": "123"
    },
    "auth": {
        "token": "y-WFmI45YEKMF2ic2g"
    }
}

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

client.connect()
```

while True:

```
    name= "Smartbridge"
```

```
    #in area location
```

```
    #latitude=17.4225176
```

```
    #longitude=78.5458842
```

```
    #out area location
```

```
    latitude=17.4219272
```

```
    longitude=78.5488783
```

```
    myData={'name': name, 'lat': latitude,'lon': longitude}
```

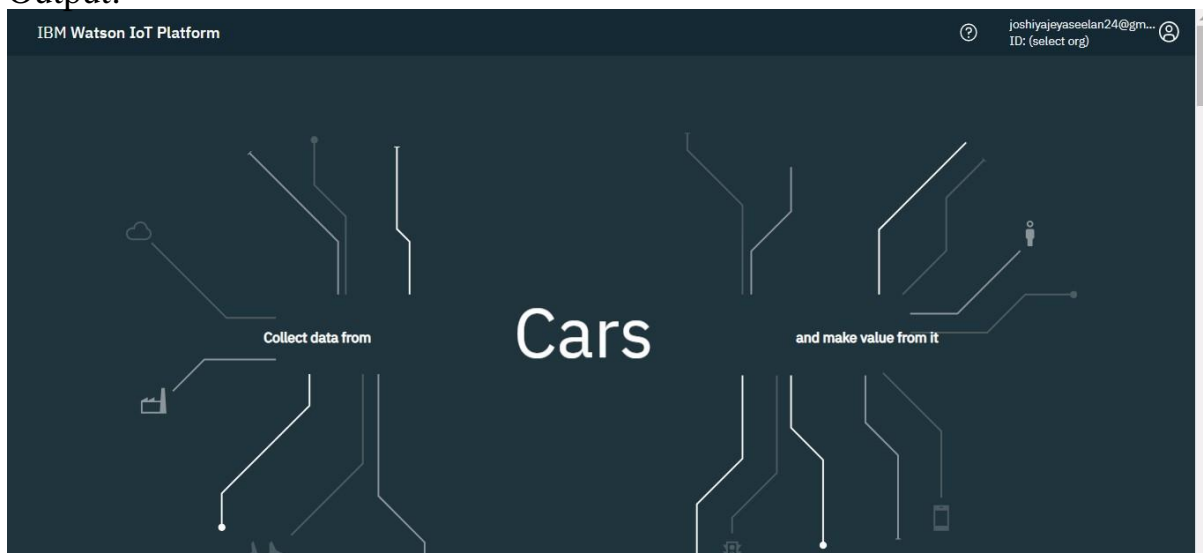
```
    client.publishEvent(eventId="status",msgFormat="json",data=myData,  
qos=0,onPublish=None)
```

```
    print("Data published to IBM IOT platform :",myData)
```

```
    time.sleep(5)
```

```
client.disconnect()
```

Output:



```
File Edit Format Run Options Window Help
import json
import winreg,osk.device
import time
myid = 1

def submitinfo():
    "config" = "logback"
    "platform" = "win7x64"
    "appidid" = "12345"
}
{
    "id": 1,
    "token": "90ab9c7e-080eb9a5c1"
}
client = winreg.osk.device.DevInfoClient(config=myConfig, logHandler=client.connect())

while True:
    name = "Smartbridge"
    pin area location
    latitude = 17.422576
    longitude = 78.545842
    rout area location
    latitude= 17.421972
    longitude= 78.548793
    register(name=name, lat=latitude, lon=longitude)
    client.publish(eventId='location', msgFormat='json', data=myData, qos=0, noPublish=True)
    Data published to IBM IoT platform:
    time.sleep(5)

client.disconnect()
```

| Identity   | Device Information                                | Recent Events | Status            | Logs |
|--|---|---------------|-------------------|------|
| The recent events listed show the live stream of data that is coming and going from this device. |   |               |                   |      |
| Event  | Value   | Format        | Last Received     |      |
| status   | ["name":"Smartbridge","lat":17.4219272,"lon":7... | json          | a few seconds ago |      |
| event_1  | ["name":"smartbridge","lat":17.4219272,"lon":7... | json          | a few seconds ago |      |
| event_1  | ["name":"smartbridge","lat":17.4219272,"lon":7... | json          | a few seconds ago |      |
| status   | ["name":"Smartbridge","lat":17.4219272,"lon":7... | json          | a few seconds ago |      |
| event_1  | ["name":"smartbridge","lat":17.4219272,"lon":7... | json          | a few seconds ago |      |

NOTIFICATION:

```

#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "a7011a"
#define DEVICE_TYPE "IOT"
#define DEVICE_ID "123"
#define TOKEN "y-WFmI45YEkmF2ic2g"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

```

```

const int trigpin=5;
const int echopin=18;
String command;
String data="";

```

```

long duration;
float dist;

```

```

void setup()
{
    Serial.begin(115200);
    pinMode(led, OUTPUT);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifiConnect();
    mqttConnect();
}

void loop() {
    bool isNearby = dist < 100;
    digitalWrite(led, isNearby);

    publishData();
    delay(500);

    if (!client.loop()) {
        mqttConnect();
    }
}

```

```

    }
}

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)) {
            Serial.print(".");
            delay(500);
        }
        initManagedDevice();
        Serial.println();
    }
}

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

void publishData()
{
    digitalWrite(trigpin, LOW);
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin, LOW);
    duration=pulseIn(echopin, HIGH);
    dist=duration*speed/2;
    if(dist<100){
        String payload = "{\"Alert Distance\":";
        payload += dist;
        payload += "}";

        Serial.print("\n");
    }
}

```

```

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

}

if(dist>100){
String payload = "{\"Distance\":\"";
payload += dist;
payload += "\"}";

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

}

}

```

