

SPRINT DELIVERY – 1

Date	5 November 2022
Team ID	PNT2022TMID22555
Project Name	Smart Waste Management System for Metropolitan Cities

Functional Requirement – simulation creation(connect sensor arduino with python code).

User story : USN – 1

STEP 1: Type the given Python Code in Compiler.

PYTHON CODE :

```
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
```

```
# watson device details
```

```
organization = "no0d4e"
devicType = "Ultrasonic-sensor"
deviceId = "kt"
authMethod = "token"
authToken = "12345678"
```

```
#generate random values for randomo variables (temperature&humidity)
```

```
def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)
```

```
try:
    deviceOptions={"org": organization, "type": devicType,"id": deviceId,"auth-
method":authMethod,"auth-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("caught exception connecting device %s" %str(e))
    sys.exit()
```

```
#connect and send a datapoint "temp" with value integer value into the cloud as a type of event
for every 10 seconds
deviceCli.connect()
```

```
while True:
```

```
    current_bin_level = random.randint(10,70)
    current_bin_weight = random.randint(5,15)
```

```
    data_to_publish= {
        'CURRENT_BIN_LEVEL': current_bin_level,
        'CURRENT_BIN_WEIGHT':current_bin_weight
    }
```

```
    def myOnPublishCallback():
        print ("DATA PUBLISHED TO IBM WATSON")
```

```
    success = deviceCli.publishEvent("Ultrasonic_sensor", "json", data_to_publish, qos=0,
on_publish=myOnPublishCallback)
```

```
    if not success:
        print("not connected to ibmiot")
        time.sleep(30)
```

```
    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

STEP 2- Now Compile the code in Python Compiler.

[illegible]

STEP 3- Type the given Wokwi Code in Compiler.

WOKWI CODE-

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

#define ORG " no0d4e"
#define DEVICE_TYPE "Ultrasonic-sensor"
#define DEVICE_ID "kt"
#define TOKEN "12345678"

#define speed 0.034
#define led 14
String data3;
float h,t;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/554517/fmt/json"; char topic[] =
"iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";

char token[] = TOKEN;

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

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifClient);

const int trigpin=5;

const int echopin=18;

String command;
String      data="";

long duration;

foat dist;

void setup()
{
```

```
Serial.begin(115200);

pinMode(led, OUTPUT);

pinMode(trigpin, OUTPUT);

pinMode(echopin, INPUT);

wifConnect();

mqttConnect();

}

void loop() {

bool isNearby = dist < 100;

    digitalWrite(led, isNearby);

publishData();

    delay(500);

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

void wifConnect() {

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()); }
```

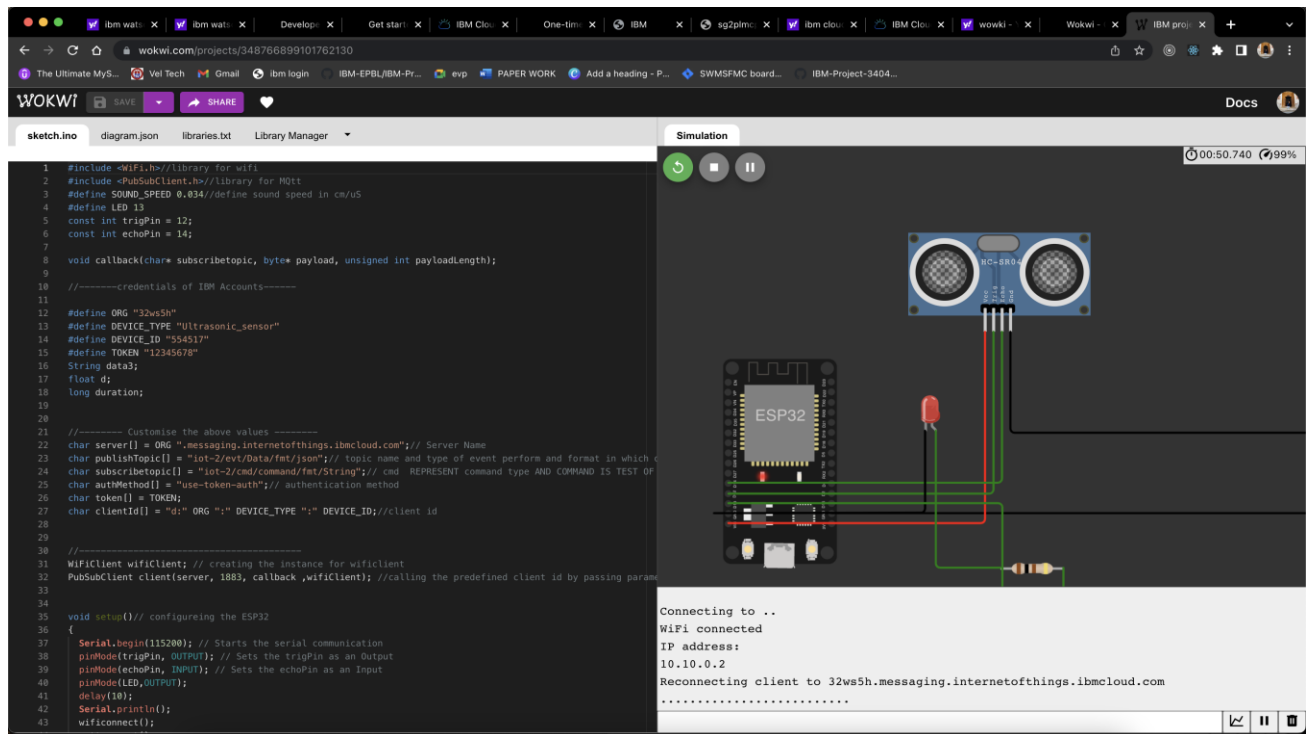
```
void mqttConnect() {  
    if (!client.connected()) {  
    }  
}
```

```
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(binlevel>95){  
        String payload = "{\"Alert\":\"";  
                                payload += binlevel; payload += "\"";  
    }  
}  
}
```

STEP 4- Now Compile the code in Wokwi Compiler and Simulate it for further.

OUTPUT :



STEP 5- The below link is regarding Code & Output.

<https://wokwi.com/projects/348766899101762130>