

SPRINT - 1

Date	28 October 2022
Team ID	PNT2022TMID21246
Project Name	Project – Smart Farmer- IoT based Smart Farming Application

OBJECTIVE:

Connecting Sensors with Arduino.

PYTHON CODE:

```
import time
import random
#import ibmiotf.application
import ibmiotf.device
import sys
config={
    "org":"nq4lh2",
    "type" : "abcd",
    "id":"123",
    "auth-method":"token",
    "auth-token":"123456789"
}
client= ibmiotf.device.Client (config)
client.connect()

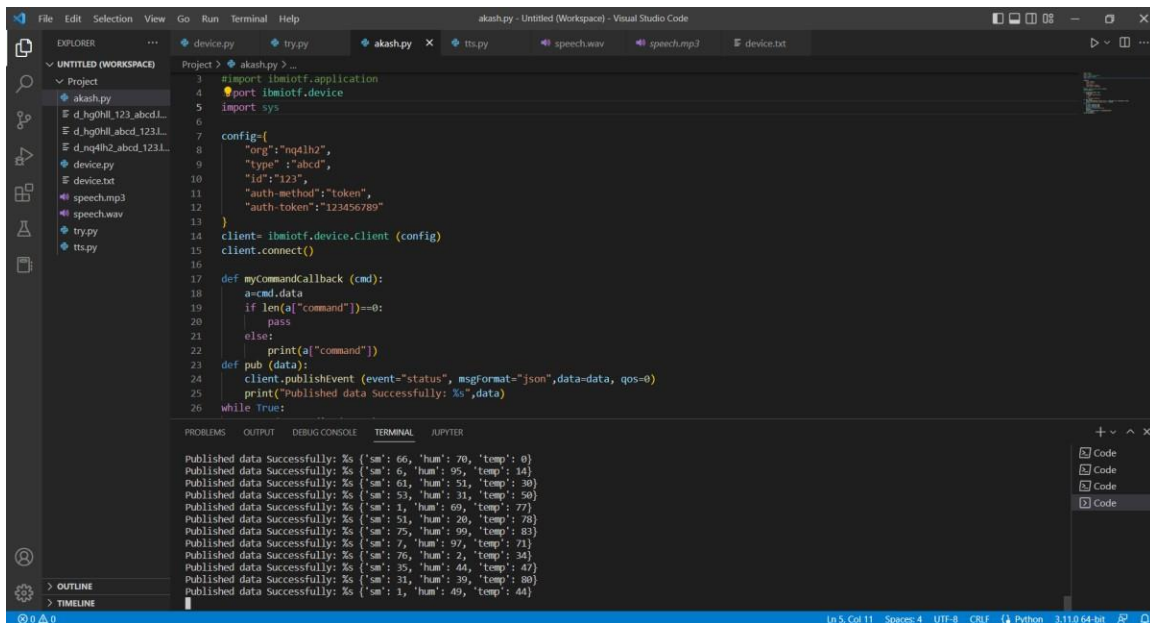
def myCommandCallback (cmd):
    a=cmd.data
    if len(a["command"])==0:
        pass
    else:
```

```

        print(a["command"])
def pub (data):
    client.publishEvent(event="status",
msgFormat="json",data=data, qos=0)
    print("Published data Successfully: %s",data)
while True:
    s=random.randint(0,100)
    h=random.randint(0,100)
    t=random.randint(0,100)
    data={"sm":s,"hum":h,"temp":t}
    pub(data)
    client.commandCallback = myCommandCallback
client.disconnect()

```

OUTPUT:



The screenshot shows a Visual Studio Code window with a Python script named `akash.py` and its terminal output. The script uses the `ibmiotf` library to connect to an MQTT broker and publish data. The terminal output shows the script running successfully, publishing data at regular intervals.

```

3 #import ibmiotf.application
4 #port ibmiotf.device
5 import sys
6
7 config={
8     "org":"mqlh2",
9     "type": "abcd",
10    "id":"123",
11    "auth-method":"token",
12    "auth-token":"123456789"
13}
14 client= ibmiotf.device.Client (config)
15 client.connect()
16
17 def myCommandCallback (cmd):
18     a=cmd.data
19     if len(a["command"])==0:
20         pass
21     else:
22         print(a["command"])
23
24 def pub (data):
25     client.publishEvent (event="status", msgformat="json",data=data, qos=0)
26     print("Published data Successfully: %s",data)
27 while True:

```

Terminal Output:

```

Published data Successfully: %s {'sm': 66, 'hum': 70, 'temp': 0}
Published data Successfully: %s {'sm': 6, 'hum': 95, 'temp': 14}
Published data Successfully: %s {'sm': 61, 'hum': 51, 'temp': 30}
Published data Successfully: %s {'sm': 53, 'hum': 31, 'temp': 50}
Published data Successfully: %s {'sm': 1, 'hum': 69, 'temp': 77}
Published data Successfully: %s {'sm': 51, 'hum': 20, 'temp': 70}
Published data Successfully: %s {'sm': 75, 'hum': 99, 'temp': 83}
Published data Successfully: %s {'sm': 7, 'hum': 97, 'temp': 71}
Published data Successfully: %s {'sm': 76, 'hum': 2, 'temp': 34}
Published data Successfully: %s {'sm': 35, 'hum': 44, 'temp': 47}
Published data Successfully: %s {'sm': 31, 'hum': 39, 'temp': 80}
Published data Successfully: %s {'sm': 1, 'hum': 49, 'temp': 44}

```