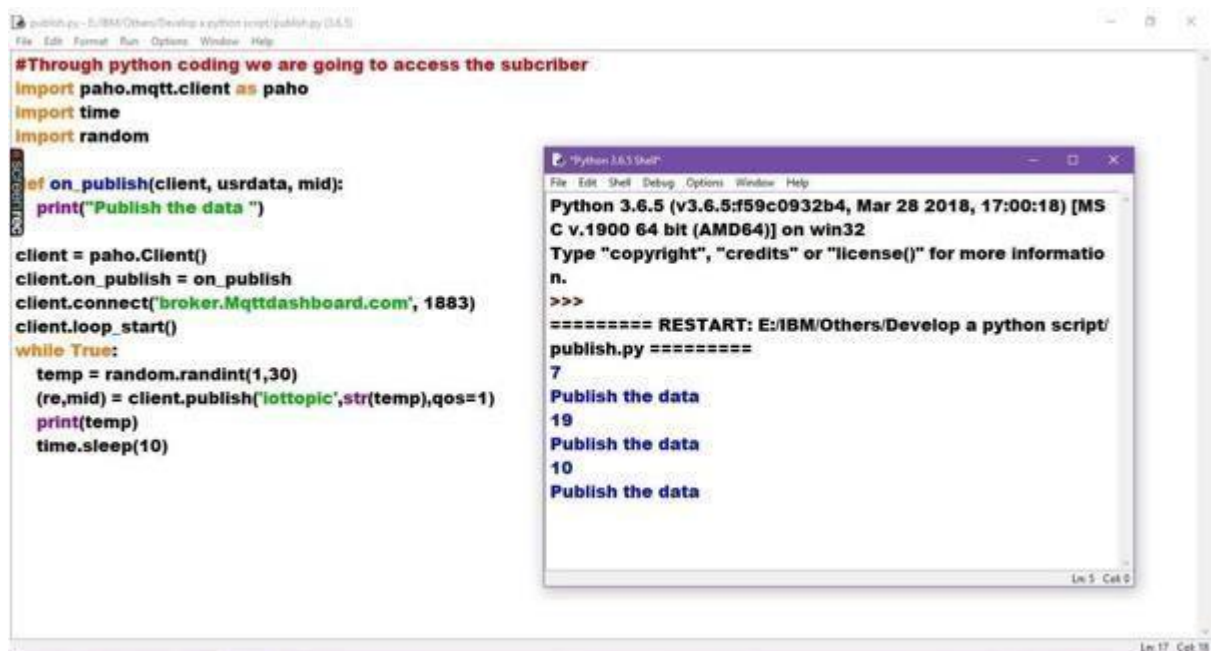


Develop the Python Script

(Publish data to IBM cloud)

Date	08 November 2022
Team ID	PNT2022TMID48564
Project Name	Industry-specific intelligent fire management system

Industry-specific intelligent fire management system



The screenshot shows a Python script in a text editor and its execution output in a shell window. The script, named `publish.py`, imports `paho.mqtt.client` as `paho`, `time`, and `random`. It defines a function `on_publish` that prints "Publish the data". The script creates a `paho.Client` instance, sets `on_publish` as the callback, connects to `broker.mqttdashboard.com` on port 1883, and starts the loop. It then enters a `while True` loop where it generates a random integer between 1 and 30, publishes it to the `lottopic` with QoS=1, prints the value, and sleeps for 10 seconds.

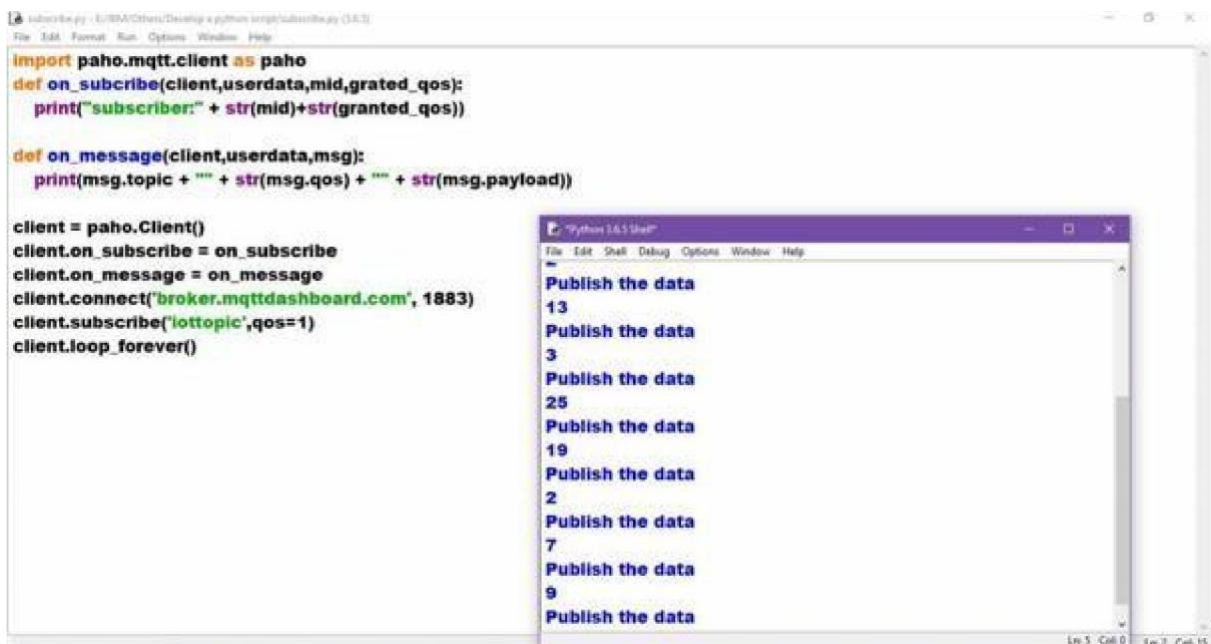
```
#Through python coding we are going to access the subscriber
import paho.mqtt.client as paho
import time
import random

def on_publish(client, userdata, mid):
    print("Publish the data ")

client = paho.Client()
client.on_publish = on_publish
client.connect('broker.mqttdashboard.com', 1883)
client.loop_start()
while True:
    temp = random.randint(1,30)
    (re,mid) = client.publish('lottopic',str(temp),qos=1)
    print(temp)
    time.sleep(10)
```

The shell window output shows the Python version (3.6.5), the restart command, and the repeated output of "Publish the data" with values 7, 19, 10, and 7.

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MS
C v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more informati
n.
>>>
===== RESTART: E:/IBM/Others/Develop a python script/
publish.py =====
7
Publish the data
19
Publish the data
10
Publish the data
```



The screenshot shows a Python script in a text editor and its execution output in a shell window. The script, named `subscribe.py`, imports `paho.mqtt.client` as `paho`. It defines two functions: `on_subscribe` which prints the subscriber ID and granted QoS, and `on_message` which prints the message topic, QoS, and payload. The script creates a `paho.Client` instance, sets the callbacks, connects to `broker.mqttdashboard.com` on port 1883, subscribes to the `lottopic` with QoS=1, and starts the loop forever.

```
import paho.mqtt.client as paho
def on_subscribe(client,userdata,mid,grated_qos):
    print("subscriber:" + str(mid)+str(granted_qos))

def on_message(client,userdata,msg):
    print(msg.topic + "" + str(msg.qos) + "" + str(msg.payload))

client = paho.Client()
client.on_subscribe = on_subscribe
client.on_message = on_message
client.connect('broker.mqttdashboard.com', 1883)
client.subscribe('lottopic',qos=1)
client.loop_forever()
```

The shell window output shows the repeated output of "Publish the data" with various values: 13, 3, 25, 19, 2, 7, 9, and 7.

```
Publish the data
13
Publish the data
3
Publish the data
25
Publish the data
19
Publish the data
2
Publish the data
7
Publish the data
9
Publish the data
```

MSN IndiaIBMIBM-EPBLIBM-EPBLReset yourPonni NarTraining CIBM-ProjeMy IBMService DIBM VMy IBM

pg685h.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platformhh0109477@gmail.comID: (select org)

BrowseActionDevice TypesInterfaces

Add Device

Device ID	Status	Device Type	Class ID	Date Added	
abcd	Disconnected	123	Device	Nov 3, 2022 12:13 PM	

IdentityDevice InformationRecent EventsStateLogs

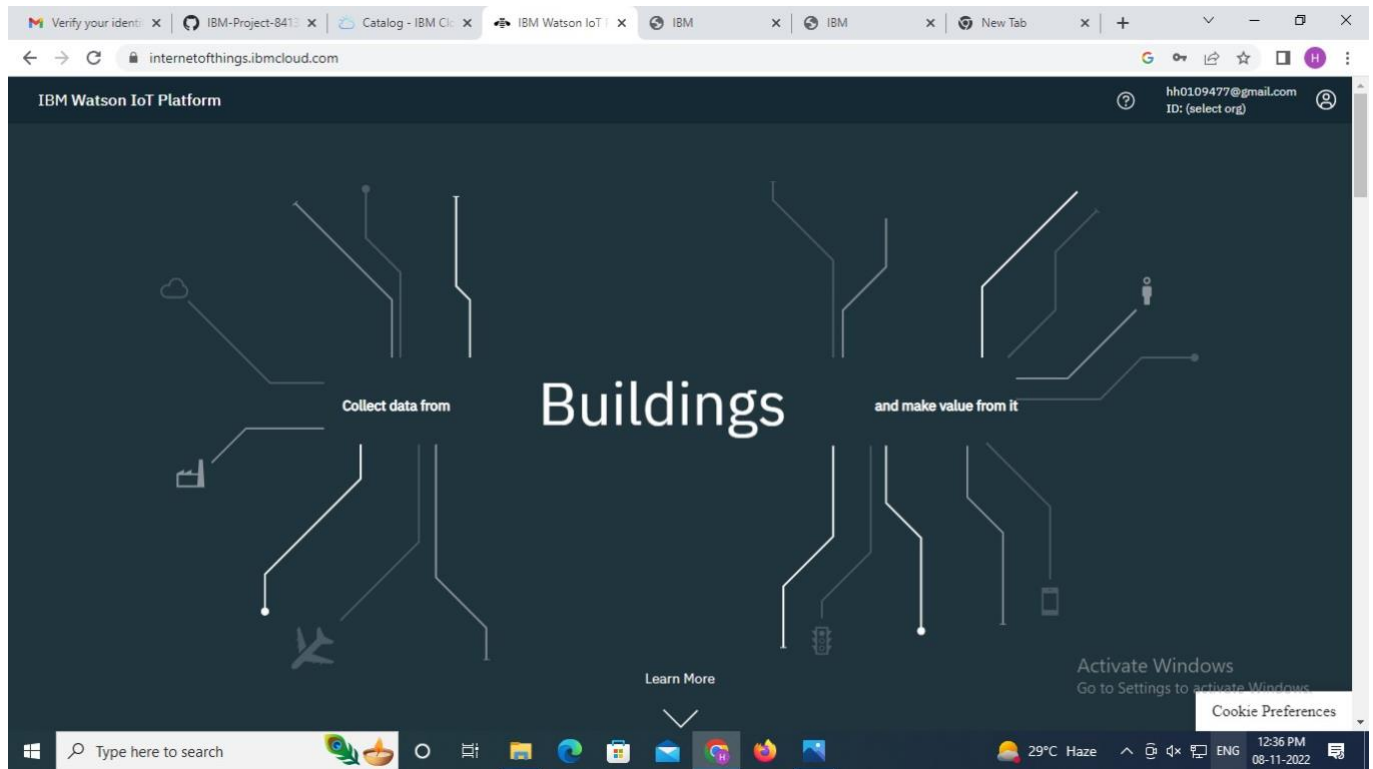
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"randomNumber":74}	json	a few seconds ago
event_1	{"randomNumber":47}	json	a few seconds ago
event_1	{"randomNumber":45}	json	a minute ago
event_1	{"randomNumber":19}	json	a minute ago
event_1	{"randomNumber":79}	json	a minute ago

1 Simulation running

Type here to search

IBM ...Scre...Node...What...node...IBM C...ENG00:1004-11-2022



Program :

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random

myConfig = {"identity":
{
    "orgId": "hj5fmy",
    "typeId": "NodeMCU",
    "deviceId": "12345" },
    "auth": { "token": "12345678" }
}

def myCommandCallback(cmd):    print("Message received from IBM IoT
Platform: %s" % cmd.data['command'])    m=cmd.data['command']

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

while True:
    temp=random.randint(-20,125)    hum=random.randint(0,100)
```

```
myData={'temperature':temp, 'humidity':hum}
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback    time.sleep(2)
client.disconnect()
```