

PROJECT DEVELOPMENT PHASE

SPRINT-1

Team ID	PNT2022TMID48285
Project Name	Industrial Specific Fire Management System

PYTHON CODE:

```
import time
```

```
import sys
```

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import random
```

```
#Provide your IBM Watson Device Credentials
```

```
organization = "ge3f42"
```

```
deviceType = "Arduino"
```

```
deviceId = "1234"
```

```
authMethod = "token"
```

```
authToken = "FfR(Gr?Vsx?4c-*k45"
```

```
# Initialize GPIO
```

```
def myCommandCallback(cmd):
```

```
    print("Command received: %s" % cmd.data['command'])
```

```
status=cmd.data['command']  
if status=="lighton":  
    print ("led is on")  
elif status == "lightoff":  
    print ("led is off")  
else :  
    print ("please send proper command")
```

```
try:
```

```
    deviceOptions = {"org": organization, "type": deviceType, "id":  
deviceId, "auth-method": authMethod, "auth-token": authToken}  
    deviceCli = ibmiotf.device.Client(deviceOptions)  
    #.....
```

```
except Exception as e:
```

```
    print("Caught exception connecting : %s" % str(e))  
    sys.exit()
```

```
# Connect and send a datapoint "hello" with value "world" into the  
cloud as an event of type "greeting" 10 times
```

```
deviceCli.connect()
```

while True:

#Get Sensor Data from DHT11

temp=random.randint(90,110)

Humid=random.randint(60,100)

data = { 'temp' : temp, 'Humid': Humid }

#print data

def myOnPublishCallback():

**print ("Published Temperature = %s C" % temp, "Humidity =
%s %" % Humid, "to IBM Watson")**

**success = deviceCli.publishEvent("IoTSensor", "json", data,
qos=0, on_publish=myOnPublishCallback)**

if not success:

print("Not connected to IoT")

time.sleep(10)

deviceCli.commandCallback = myCommandCallback

Disconnect the device and application from the cloud

deviceCli.disconnect()

OUTPUT:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays a table of devices. The first device, ID '1234', is 'Connected' and is an 'Arduino' of class 'Device', added on 'Nov 15, 2022 1:25 PM'. Below the table, a 'Showing Raw Data' section indicates 'No Interfaces Available'. A table of raw data shows two entries: 'temp' with a value of '101' and 'Humid' with a value of '76'. Both are of type 'Number' and are 'IoTSensor' events received 'a few seconds ago'. A '0 Simulations running' message is visible at the bottom right of the data section.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Connected	Arduino	Device	Nov 15, 2022 1:25 PM	

Property	Value	Type	Event	Last Received
temp	101	Number	IoTSensor	a few seconds ago
Humid	76	Number	IoTSensor	a few seconds ago

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\haristark\Python\iot fire.py =====
2022-11-15 13:25:43,152 ibmiotf.device.Client INFO Connected successfully: d:ge3f42:Arduino:1234
Published Temperature = 94 C Humidity = 65 % to IBM Watson
Published Temperature = 110 C Humidity = 69 % to IBM Watson
Published Temperature = 101 C Humidity = 76 % to IBM Watson
Published Temperature = 109 C Humidity = 67 % to IBM Watson
```

RESULT: The python code is executed successfully.