# Develop a python script

|  |  |
| --- | --- |
| Date | 17 September 2022 |
| Team ID | PNT2022TMID47453 |
| Project Name | Project - Industry-Specific Intelligent Fire  Management System |

## Python script for generating the random sensor values - Temperature, Flame Level and Gas Level to the IBM Watson IoT Platform.

## import time

## import sys

## import ibmiotf.application

## import ibmiotf.device

## import random

## #Provide your IBM Watson Device Credentials

## organization = "ilph7t"

## deviceType = "910019104702"

## deviceId = "910019104702"

## authMethod = "token"

## authToken = "-hht6G8AmqCPZN?Cgt"

## # Initialize GPIO

## def myCommandCallback(cmd):

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

## status=cmd.data['command']

## if status=="lighton":

## print ("fan is on")

## else :

## print ("fan is off")

## 

## #print(cmd)

## 

## 

## 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 device: %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

## 

## temperature=random.randint(0,100)

## flamesensor=random.randint(0,100)

## Gassensor=random.randint(0,100)

## 

## data = { 'temperature' : temperature , 'flame sensor': flamesensor ,'Gas sensor': Gassensor }

## #print data

## def myOnPublishCallback():

## print ("Published = temperature %s " % temperature , "flame sensor = %s %%" %flamesensor , "Gas sensor = %s %%" % Gassensor, "to IBM Watson")

## success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

## if not success:

## print("Not connected to IoTF")

## time.sleep(1)

## 

## deviceCli.commandCallback = myCommandCallback

## # Disconnect the device and application from the cloud

## deviceCli.disconnect()

# Output :

## 