Industry-specific intelligent fire management system

Sprint 1

Python code:

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


#Provide your IBM Watson Device Credentials
organization = "2piqlm"
deviceType = "Code"
deviceId = "123456"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO


def myCommandCallback1(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "sprinkleron":
        print("sprinkler is on")
    else:
        print("sprinkler is off")

    print(cmd)

def myCommandCallback2(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "fanon":
        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

        temp=random.randint(0,70)
        gas=random.randint(0,100)
        flame=random.randint(0,1)


        data = { 'temp' : temp, 'gas': gas, 'flame': flame }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Gas = %s %%" %
gas, "flame = %s %%" % flame,"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.commandCallback1 = myCommandCallback1
        deviceCli.commandCallback2 = myCommandCallback2

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```