

HAZARDOUS AREA MONITORING FOR INDUSTRIAL PLANT POWERED BY IoT

Team ID: PNT2022TMID41272

Project Development Phase: Sprint 2

❖ Install the IDLE Python version 3.7.4 and install the required libraries

```
python01.py - C:\Users\WGNESH R\AppData\Local\Programs\Python\Python37\python01.py (3.7.4)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "6xryq0"
deviceType = "ESF32x"
deviceId = "19"
authMethod = "token"
authToken = "12345678"

# 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 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(-20,100)
    Humidity=random.randint(0,100)
    vibrate=random.randint(0,100)
    a="Warning!!!!..Temperature is high"
    data = { "temperature": temperature, "Humidity": Humidity, "vibrate": vibrate}

    #Publish Data
```

```
*Python 374 Shell*
File Edit Shell Debug Options Window Help

Warning!!!!..Temperature is high
Published temperature = 45 C Humidity = 93 % vibrate = 57 hz to IBM Watson
Published temperature = 88 C Humidity = 100 % vibrate = 0 hz to IBM Watson
Warning!!!!..Temperature is high
Published temperature = 21 C Humidity = 53 % vibrate = 5 hz to IBM Watson
Published temperature = 39 C Humidity = 20 % vibrate = 14 hz to IBM Watson
Published temperature = 94 C Humidity = 11 % vibrate = 55 hz to IBM Watson
Warning!!!!..Temperature is high
Published temperature = -10 C Humidity = 39 % vibrate = 34 hz to IBM Watson
Published temperature = 97 C Humidity = 51 % vibrate = 49 hz to IBM Watson
Warning!!!!..Temperature is high
Published temperature = 41 C Humidity = 17 % vibrate = 71 hz to IBM Watson
Published temperature = 40 C Humidity = 61 % vibrate = 27 hz to IBM Watson
Published temperature = -11 C Humidity = 82 % vibrate = 3 hz to IBM Watson
Published temperature = -8 C Humidity = 9 % vibrate = 80 hz to IBM Watson
Published temperature = 46 C Humidity = 45 % vibrate = 67 hz to IBM Watson
Published temperature = 23 C Humidity = 19 % vibrate = 87 hz to IBM Watson
Published temperature = 71 C Humidity = 55 % vibrate = 35 hz to IBM Watson
Published temperature = 34 C Humidity = 69 % vibrate = 87 hz to IBM Watson
Published temperature = -18 C Humidity = 70 % vibrate = 57 hz to IBM Watson
Published temperature = 6 C Humidity = 26 % vibrate = 9 hz to IBM Watson
Published temperature = 46 C Humidity = 24 % vibrate = 54 hz to IBM Watson
Published temperature = -6 C Humidity = 71 % vibrate = 67 hz to IBM Watson
Published temperature = 15 C Humidity = 46 % vibrate = 98 hz to IBM Watson
Published temperature = 14 C Humidity = 36 % vibrate = 11 hz to IBM Watson
Published temperature = 28 C Humidity = 79 % vibrate = 85 hz to IBM Watson
Published temperature = 48 C Humidity = 1 % vibrate = 87 hz to IBM Watson
Published temperature = 19 C Humidity = 32 % vibrate = 61 hz to IBM Watson
Published temperature = -16 C Humidity = 42 % vibrate = 74 hz to IBM Watson
Published temperature = 81 C Humidity = 82 % vibrate = 78 hz to IBM Watson
Warning!!!!..Temperature is high
Published temperature = 22 C Humidity = 23 % vibrate = 70 hz to IBM Watson
Published temperature = 32 C Humidity = 100 % vibrate = 85 hz to IBM Watson
Published temperature = 76 C Humidity = 12 % vibrate = 38 hz to IBM Watson
Published temperature = 37 C Humidity = 22 % vibrate = 61 hz to IBM Watson
Published temperature = 92 C Humidity = 89 % vibrate = 63 hz to IBM Watson
Warning!!!!..Temperature is high
Published temperature = 100 C Humidity = 47 % vibrate = 89 hz to IBM Watson
Command received: lighton
led is on
Warning!!!!..Temperature is high
Published temperature = 64 C Humidity = 6 % vibrate = 54 hz to IBM Watson
Published temperature = 64 C Humidity = 83 % vibrate = 61 hz to IBM Watson
Published temperature = 39 C Humidity = 92 % vibrate = 64 hz to IBM Watson
Published temperature = 80 C Humidity = 6 % vibrate = 87 hz to IBM Watson
Published temperature = -15 C Humidity = 2 % vibrate = 34 hz to IBM Watson
Published temperature = 54 C Humidity = 38 % vibrate = 31 hz to IBM Watson
```

PROGRAM :

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "6xryq0"

deviceType = "ESP32x"

deviceId = "19"

authMethod = "token"

authToken = "12345678"

# 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 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(-20,100)

    Humidity=random.randint(0,100)

    vibrate=random.randint(0,100)

    a="Warning!!!....Temperature is high"

    data = { 'temperature' : temperature, 'Humidity': Humidity, 'vibrate': vibrate}


    #print data

    def myOnPublishCallback():

        print ("Published temperature = %s C" % temperature, "Humidity = %s %" % Humidity,
        "vibrate = %s hz" % vibrate, "to IBM Watson")

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

    if not success:

        print("Not connected to IoT")

        time.sleep(1)

        if temperature>80:

            a="Warning!!!....Temperature is high"

            print("Warning!!!....Temperature is high")


    deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud

deviceCli.disconnect()

```