**DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IMB IOT PLATFORM**

|  |  |
| --- | --- |
| **Team ID** | **PNT2022TMID24776** |
| **Project Name** | **Smart Farmer - IoT Enabled Smart Farming Application** |

**PYTHON CODE:**

import sys

import time

import ibmiotf.application

import ibmiotf.device

import random

#provide ibm watson device credential

organizaton ="rtf29"

deviceType ="Smart\_Farmers"

deviceId ="210419106036"

authMethod ="use-token-auth"

authToken ="Hari@2002"

#initialize GPIO

def mycommandcallback(cmd):

print("command recieved: %s" % cmd.data['command'])

status=cmd.data['command']

if status=="lighten":

print ("led is on")

else:

print ("led 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()

deviceCli.coonect()

while True:

#temp Sensor Data from DHt11

temp=random.randint(0,100)

Humid=random.randint(0,100)

data = {'temperature' : temp, 'humidity' : 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 iotf")

time.sleep(1)

deviceCli.commandCallback = mycommandCallback

# Disconnect the Device and application from the cloud

deviceCli.disconnect