PROJECT DEVELOPMENT PHASE

SPRINT DELIVERY -4

|  |  |
| --- | --- |
| Date | 19 November 2022 |
| Team ID | PNT2022TMID27179 |
| Project ID | Smart farmer- IoT based smart farming application |

import time import

sys

import ibmiotf.application

import ibmiotf. device

import random

#Provide your IBM Watson

Device Credentials organization = "qxplm8"

device Type = "NodeMcu"

deviceId = "12345"

authMethod = "auth-token"

authToken = ")22UmCJOV1XDNEs) BR"

# Initialize GPIO def myCommandCallback(cmd):

Print ("Command received: %s" % cmd.data['command']status=cmd.data['command']

if status=="motor on":

print ("motor is on")

elif status == "motor off":

print ("motor is off")

else: print ("please send proper command")

try:

device Options = {"org": organization, "type": device Type, "id": deviceId,

"auth-method": authMethod, "auth-token": authToken}

DeviceCli = ibmiot. DeviceClient (déceptions)

#.............................................. 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()

deviceCli.connect()

while True:

#Get Sensor Data from DHT11

Temp=random. Randint (90,110)

Humid=random.randint(60,100)

Mois=random. Randint (20,120)

data = {'temp’: temp, 'Humid': Humid, ‘Mois’: Mois}

#Print data def myOnPublishCallback ():

print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,

“Moisture =%s deg c”% Mois “to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data,

qos=0, onPublish=myOnPublishCallback)

if not success:

print ("Not connected to IoTF")

time. Sleep (10)

deviceCli.commandCallback = myCommandCallback # Disconnect the device and

application from the

cloud deviceCli.disconnect()



