SPRINT – 1

|  |  |
| --- | --- |
| **Date** | **15-11-2022** |
| **Team ID** | **PNT2022TMID36755** |
| **Project Name** | **Project - IOT based smart crop**  **protection for agriculture** |
| **Maximum Marks** | **20 Marks** |

import time

import sys

import ibmiotf.application # to install pip install ibmiotf

import ibmiotf.device

#Provide your IBM Watson Device Credentials

organization = "r5stg1" #replace the ORG ID

deviceType = "NODEMCU1"#replace the Device type wi

deviceId = "12345"#replace Device ID

authMethod = "token"

authToken = "12345678" #Replace the authtoken

def myCommandCallback(cmd): # function for Callback

print("Command received: %s" % cmd.data)

if cmd.data['command']=='ON':

print("MOTOR ON IS RECEIVED")

time.sleep(1)

print("MOTOR STARTED")

elif cmd.data['command']=='OFF':

print("MOTOR OFF IS RECEIVED")

time.sleep(1)

print("MOTOR STOPPED")

elif cmd.data['command']=='runfor30minutes':

print("MOTOR RUNS FOR 30 MINUTES")

print("MOTOR STARTED")

for i in range(1,31):

print("%d minutes to stop"%(30-i)) # use time.sleep(60) for delay of one minute in each iteration

print("MOTOR STOPPED")

if cmd.command == "setInterval":

if 'interval' not in cmd.data:

print("Error - command is missing required information: 'interval'")

else:

interval = cmd.data['interval']

elif cmd.command == "print":

if 'message' not in cmd.data:

print("Error - command is missing required information: 'message'")

else:

output=cmd.data['message']

print(output)

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:

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

Output:

