|  |  |
| --- | --- |
| Team ID | PNT2022TMID46309 |
| Project Name | IoT Based Smart Crop Protection System forAgriculture |

SOURCE CODE

import time import sys

import ibmiotf.application import ibmiotf.device import random

#Provide your IBM Watson Device Credentials organization = "z22obn"

deviceType = "IBM" deviceId = "IBMID1" authMethod = "token" authToken = "TOKENIBM" # Initialize GPIO

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 temp=random.randint(0,100) Humid=random.randint(0,100) soilmoisture = random.randint(0,100)

#Assume

if temp>=50 and Humid>=50 or soilmoisture>=60 : motion = 1

print(" ") print("Motion detected..!")

else :

motion = 0

data = { 'temp' : temp ,'Humid': Humid , 'soilmoisture' : soilmoisture , 'Motion' : motion } #print data

def myOnPublishCallback():

print ("Published to IBM Watson...!")

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

%s %%" % soilmoisture )

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF") time.sleep(10)

def myCommandCallback(command): print("Command received: %s" % command.data) command=command.data['command'] print(command)

if(command=='sprinkler has been switched on'): print('sprinkleron')

elif(command=='sprinkler has been switched off'): print('sprinkleroff')

elif(command=='motor has been switched on'): print('motoron')

elif(command=='motor has been switched off'): print('motoroff')

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myCommandCallback)

if not success:

print("Command not received")

deviceCli.commandCallback = myCommandCallback

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