

IOT Based Smart Crop Protection System for Agriculture.

Team ID - PNT2022TMID2282

MAIN CODE 2:

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

#Provide your IBM Watson Device Credentials
organization = "7zn1xs"
deviceType = "cropprotection"
deviceId = "cropprotectionssystemid"
authMethod = "token"
authToken = "ejrRfZRYwhhZCz!mUR"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="alarmon":
        print ("alarm is on")
    elif status == "alarmoff":
        print ("alarm 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

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

    data = { 'Temperature' : temp, 'Humidity': Humid,'moisture':moist,}
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %% "
% Humid, "moisture = %s %% " %moist, "to IBM Watson")

```

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