SMART PLANT COMMUNICATOR SYSTEM WITH IBM CLOUD

Abstract :

The most important factors for the quality and productivity of plant growth are temperature, humidity, light and the soil moisture levels. Continuous monitoring of these environmental variables gives information to grow better how each factor affects growth and how to manage maximal growth of plants. Climate control and monitoring of the plant is one of important aspect. The aspects we are presenting resembles the concept of precision agriculture. The main motivation of the project is for the user to monitor the plants and also could manipulate the resources provided to the plants depending on the climate of the plant's location. This could help user not only to give the resources to the plants everyday without much manual effort also helps the constant and healthy growth of a plant. This project is designed as a smart plant communicator system with IBM cloud based on IOT. In this project we used different modules such as IOT, Node red, Temperature, soil moisture levels, Humidity

Table of contents:

INTRODUCTION:

Overview:

We live in a world where everything can be controlled and operated automatically, but there are still a few important sectors in our country where automation has not been adopted or not been put to a full-fledged use, perhaps because of several reasons one such reason is cost. Plant monitoring form an important part of the agriculture and horticulture sectors in our country as they can be used to grow plants under controlled climatic conditions for optimum produce. Automating a plant monitoring and controlling of the climatic parameters which directly or indirectly govern the plant growth and hence their produce.

Purpose:

The automated system will reduce the need of manpower, hence reducing the error. for a largescale area, it is quite impossible for a farmer to monitor the efficiency of the system by implementing this technology, the farmers can easily monitor the system using their smart phone. Also due to busy life these days we are not able to keep proper care of plants such as watering plant, to check whether plant is getting sufficient sunlight etc. To easy this we are making an IOT based automation system in which user can monitor plant parameters such as temperature, humidity, moisture and can also water them.

Literature survey:

Existing problem:

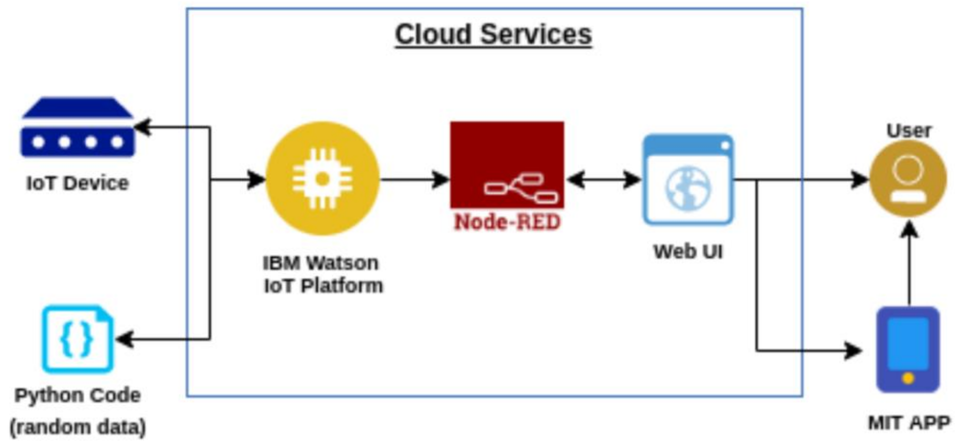
We have studied many previous works done in this field by different researchers. Use of this technology plays important role in increasing the production as well as in reducing the manpower efforts. IoT Based Plant Monitoring System shows, In India about 35% of land was under reliably irrigated. And the 2/3rd part of land is depending on monsoon for the water. Farmers are facing problems related to watering system that how much water has to supply and at what time? Sometimes overwatering causes the damage to the plant and as well as waste of water. Hence for avoid such damage we need to maintain approximate water level in soil.

Proposed solution:

So that we tried to make a system which will monitor the health conditions of the plants. Our systems will check different environmental parameters and with the connection of an IOT platform we are going to have the information of the conditions of our plants and then we can take effective steps for the welfare of the plants. In this system the heart of our system will be worked as a cloud platform. If any change in the temperature, humidity and soil moisture levels occurs, a message will be sent to the user smart phone.

Theoretical Analysis :

Block Diagram



Software Requirements :

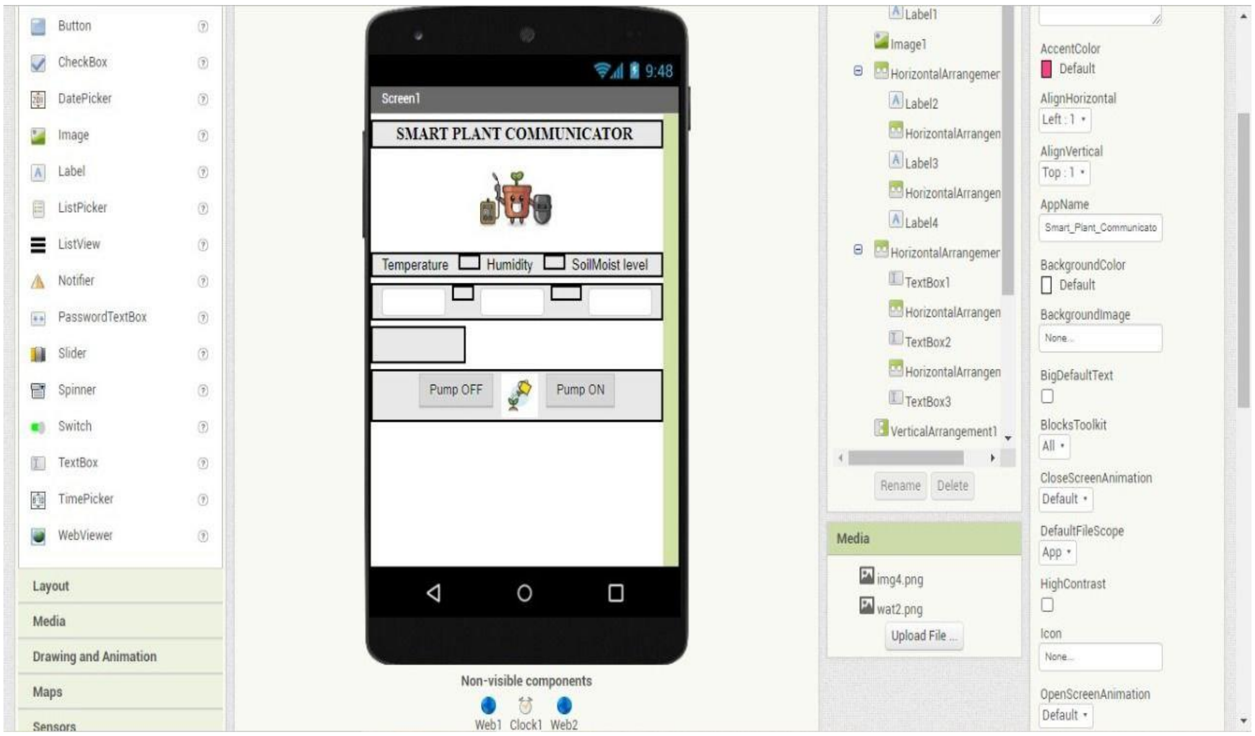
The project is done using software components.

1. Language Used:
    1.1.1. Python
2. Software Components:
    2.1.1. Python IDLE 3.9.7
    2.1.2. IBM cloud
3. IBM Watson
4. Node red Application
5. MIT App Inventor

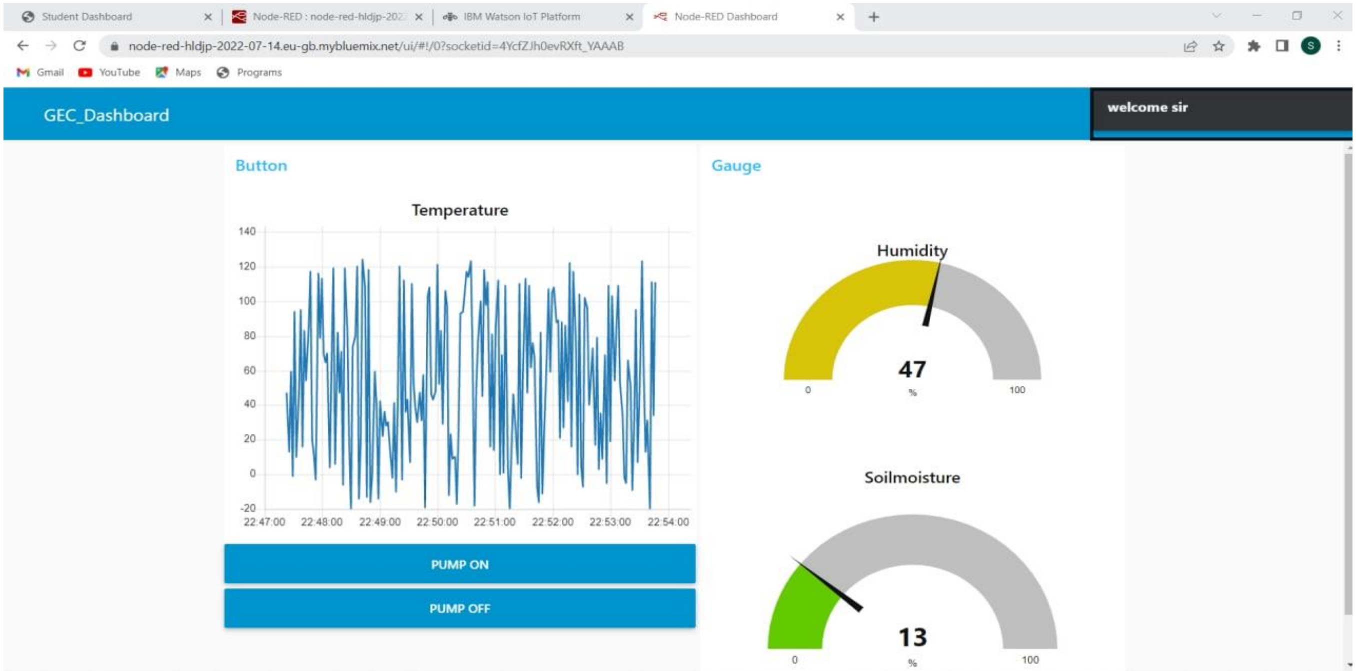
Experimental Investigations :

This system helps in monitoring and controlling the climatic conditions that are good for the development of a specific plant. By utilizing this system, plants development can be improved alongside augmented yield, regardless of the climate conditions. The propose system is cost effective but we could not implement our propose system for this Corona pandemic. In this IoT based greenhouse monitoring system empower ranchers to keep steady over their harvest conditions. This guarantees water system and treatment exercises are comparable to the real needs of developed plants for boosted yields. For instance, readings on soil volumetric water content show whether yields are submerged pressure. Similarly, estimations of soil saltiness give valuable experiences on preparation prerequisites. In view of this information, sprinkler and splashing system can be naturally gone on to address ongoing harvest requests while limiting manual mediation. This system realizes the elements of showing constant information about nursery condition factors, information inquiry and setting the warning value. Likewise in this framework forecast is never really out the necessary arrangement if any issue happens in any gadget associated with the framework. The home apparatuses can be controlled through Smartphone .Smart phone utilizing Wi-Fi. Here raspberry pi utilized as worker system and Wi-Fi as communication protocol.

Results :



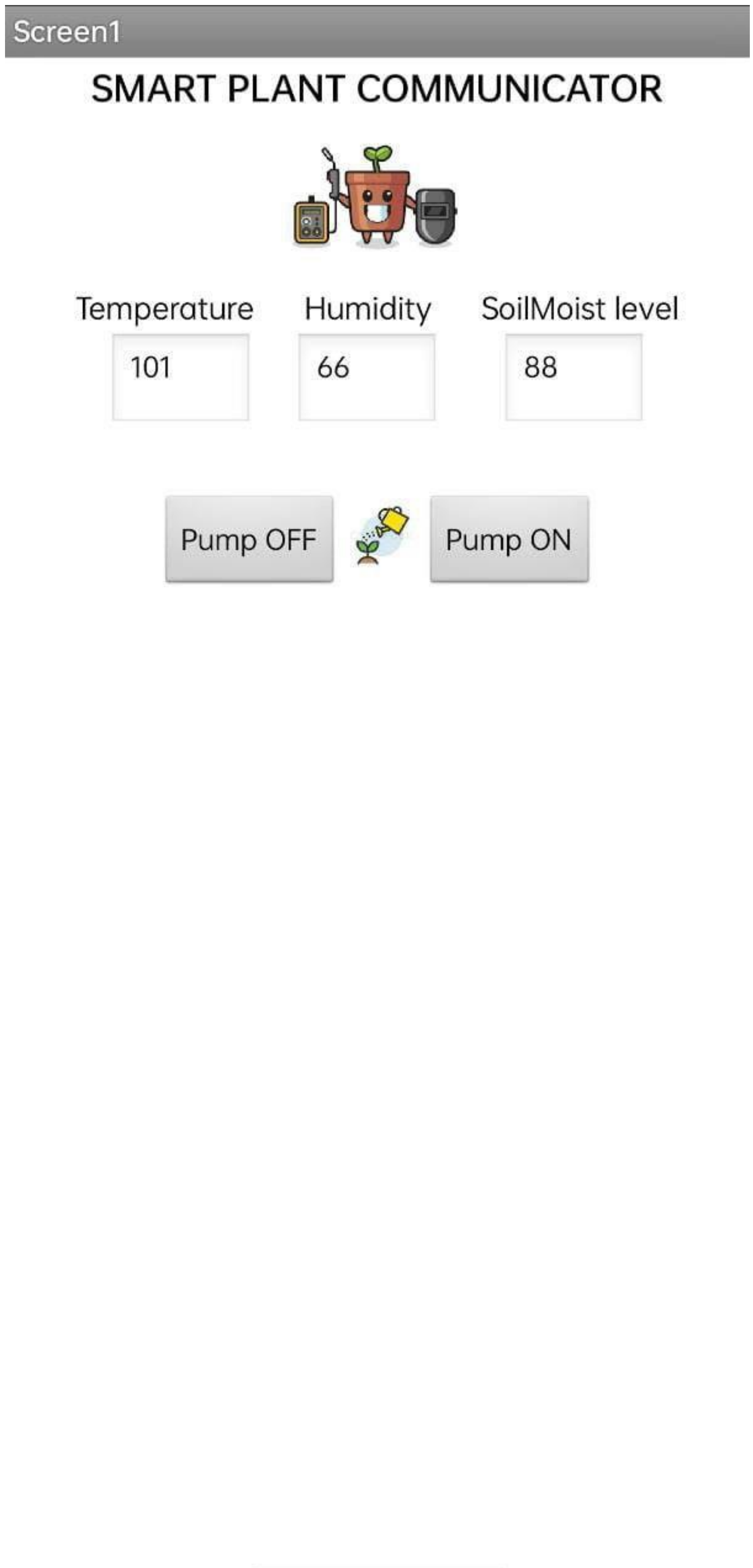
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Sensors
9:48
Screen1
SMART PLANT COMMUNICATOR
Temperature
Humidity
SoilMoist level
Pump OFF
Pump ON
Non-visible components
Web1
Clock1
Web2
Label1
Image1
HorizontalArrangemer
Label2
HorizontalArrangen
Label3
HorizontalArrangen
Label4
HorizontalArrangemer
TextBox1
HorizontalArrangen
TextBox2
HorizontalArrangen
TextBox3
VerticalArrangement1
Rename
Delete
Media
img4.png
wat2.png
Upload File ...
AccentColor
Default
AlignHorizontal
Left : 1
AlignVertical
Top : 1
AppName
Smart_Plant_Communicato
BackgroundColor
Default
BackgroundImage
None...
BigDefaultText
BlocksToolkit
All
CloseScreenAnimation
Default
DefaultFileScope
App
HighContrast
Icon
None...
OpenScreenAnimation
Default



IDLE Shell 3.9.7

File Edit Shell Debug Options Window Help

```
Published data Successfully: %s {'temperature': 9, 'humidity': 89, 'soilmoisture': 86, 'guest': 5}
Published data Successfully: %s {'temperature': 58, 'humidity': 51, 'soilmoisture': 47, 'guest': 10}
Published data Successfully: %s {'temperature': 111, 'humidity': 17, 'soilmoisture': 84, 'guest': 3}
Published data Successfully: %s {'temperature': -6, 'humidity': 43, 'soilmoisture': 96, 'guest': 10}
Published data Successfully: %s {'temperature': 6, 'humidity': 66, 'soilmoisture': 78, 'guest': 7}
Published data Successfully: %s {'temperature': 84, 'humidity': 17, 'soilmoisture': 18, 'guest': 9}
Published data Successfully: %s {'temperature': -11, 'humidity': 76, 'soilmoisture': 34, 'guest': 0}
Published data Successfully: %s {'temperature': 0, 'humidity': 75, 'soilmoisture': 36, 'guest': 6}
Published data Successfully: %s {'temperature': 61, 'humidity': 57, 'soilmoisture': 45, 'guest': 1}
Published data Successfully: %s {'temperature': -7, 'humidity': 25, 'soilmoisture': 74, 'guest': 8}
Published data Successfully: %s {'temperature': 82, 'humidity': 57, 'soilmoisture': 74, 'guest': 0}
Published data Successfully: %s {'temperature': -1, 'humidity': 58, 'soilmoisture': 95, 'guest': 10}
Published data Successfully: %s {'temperature': 18, 'humidity': 62, 'soilmoisture': 95, 'guest': 9}
Published data Successfully: %s {'temperature': 68, 'humidity': 0, 'soilmoisture': 77, 'guest': 5}
Published data Successfully: %s {'temperature': 15, 'humidity': 92, 'soilmoisture': 97, 'guest': 3}
Published data Successfully: %s {'temperature': 43, 'humidity': 56, 'soilmoisture': 86, 'guest': 6}
Published data Successfully: %s {'temperature': 37, 'humidity': 77, 'soilmoisture': 46, 'guest': 8}
Published data Successfully: %s {'temperature': 120, 'humidity': 29, 'soilmoisture': 40, 'guest': 0}
Message received from IBM IoT Platform: PUMP ON
The pump is switched OFF
Published data Successfully: %s {'temperature': 27, 'humidity': 38, 'soilmoisture': 3, 'guest': 3}
Published data Successfully: %s {'temperature': 97, 'humidity': 9, 'soilmoisture': 36, 'guest': 6}
Published data Successfully: %s {'temperature': 32, 'humidity': 50, 'soilmoisture': 69, 'guest': 1}
Published data Successfully: %s {'temperature': 3, 'humidity': 20, 'soilmoisture': 28, 'guest': 8}
Published data Successfully: %s {'temperature': 19, 'humidity': 77, 'soilmoisture': 67, 'guest': 10}
Published data Successfully: %s {'temperature': 54, 'humidity': 17, 'soilmoisture': 6, 'guest': 7}
Published data Successfully: %s {'temperature': 123, 'humidity': 80, 'soilmoisture': 91, 'guest': 0}
Published data Successfully: %s {'temperature': 55, 'humidity': 11, 'soilmoisture': 9, 'guest': 2}
Published data Successfully: %s {'temperature': 16, 'humidity': 90, 'soilmoisture': 68, 'guest': 4}
Published data Successfully: %s {'temperature': 74, 'humidity': 42, 'soilmoisture': 32, 'guest': 5}
Message received from IBM IoT Platform: PUMP ON
The pump is switched OFF
Published data Successfully: %s {'temperature': 53, 'humidity': 56, 'soilmoisture': 57, 'guest': 7}
Published data Successfully: %s {'temperature': 90, 'humidity': 96, 'soilmoisture': 80, 'guest': 3}
Published data Successfully: %s {'temperature': 58, 'humidity': 71, 'soilmoisture': 21, 'guest': 1}
Published data Successfully: %s {'temperature': 38, 'humidity': 36, 'soilmoisture': 0, 'guest': 2}
Published data Successfully: %s {'temperature': 55, 'humidity': 18, 'soilmoisture': 21, 'guest': 5}
Published data Successfully: %s {'temperature': 52, 'humidity': 95, 'soilmoisture': 85, 'guest': 5}
Published data Successfully: %s {'temperature': 86, 'humidity': 54, 'soilmoisture': 21, 'guest': 10}
Published data Successfully: %s {'temperature': 93, 'humidity': 67, 'soilmoisture': 77, 'guest': 9}
Published data Successfully: %s {'temperature': 72, 'humidity': 82, 'soilmoisture': 50, 'guest': 10}
Published data Successfully: %s {'temperature': 6, 'humidity': 34, 'soilmoisture': 78, 'guest': 10}
Published data Successfully: %s {'temperature': 47, 'humidity': 62, 'soilmoisture': 93, 'guest': 3}
Published data Successfully: %s {'temperature': 122, 'humidity': 82, 'soilmoisture': 5, 'guest': 8}
Published data Successfully: %s {'temperature': 97, 'humidity': 52, 'soilmoisture': 92, 'guest': 5}
Published data Successfully: %s {'temperature': 112, 'humidity': 90, 'soilmoisture': 19, 'guest': 7}
```


Screen1
SMART PLANT COMMUNICATOR
Temperature
101
Humidity
66
SoilMoist level
88
Pump OFF
Pump ON

Advantages and Disadvantages :

Advantages

1. Reduced water consumption
2. Safe
3. No manpower required
4. Require smaller water sources
5. Reduce soil erosion and nutrient leaching

Disadvantages

1. The smart plant need availability on internet continuously. Rural part of the developing countries did not fulfill this requirements. Moreover internet is slower.
2. Fault sensor or data processing engines can cause faulty decisions which may lead to over use of water, fertilizers and other wastage of resources.

Conclusion :

The implementation of Smart Plant Communicator system using the Internet of Things has been verified to satisfactorily work by connecting different parameters of the soil to the cloud and was successfully controlled remotely through a mobile application. The system designed not only monitors the sensor data, like moisture, humidity, temperature and ultrasonic but also actuates other parameters according to the requirement, for example, if the water level in tank is reduced to a minimum value then the motor switch is turned on automatically to the water level of the tank reaches the maximum value.

Bibliography :

References

1. Kidd, CoryD. et,al. “The Aware Home: A Living Laboratory for UbiquitousComputing Research”. Springer Berlin Heidelberg (1999).
2. Antonio J. Jara, Miguel A. Zamora, and Antonio F. Skarmeta. “An internetof things—based personal device for diabetes therapy management in ambientassisted living (AAL).”. Personal Ubiquitous Comput. 15, 4 (April 2011),431-440.
3. M. Mancuso and F. Bustaffa, “TA Wireless Sensors Network for MonitoringEnvironmental Variables in a Tomato Greenhouse”.
4. Teemu Ahonen, Reino Virrankoski and Mohammed Elmusrati , “GreenhouseMonitoring with Wireless Sensor Network”. University of Vaasa.
5. TongKe, Fan. “Smart Agriculture Based on Cloud Computing and IOT.”Journal of Convergence Information Technology 8.2 (2013).
6. Zhang, Xiang Wen, Ran Chen, and Chun Wang. “Design for Smart Mon-itoring and Control System of Wind Power Plants.” Advanced MaterialsResearch. Vol. 846. 2014.
7. Chen, Joy Iong Zong, Yuan-Chen Chen, and Shien-Dou Chung. “Implemen-tation of a Greenhouse Crop Remote Monitoring System with IOT Technology.

Appendix :

Source code

```
M Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "ttmy0h",
        "typeId": "plant",
        "deviceId":"123654"
    },
    "auth": {
        "token": "12345678"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if m=="pump ON":
        print("The pump is switched ON")
    else:
        print("The pump is switched OFF")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    soilmoistLevel=random.randint(0,100)
    user=random.randint(0,10)
    if soilmoistLevel<65:
        print("Motor on")
    else:
        print("Motor off")
    myData={'temperature':temp, 'humidity':hum, 'soilmoistureLevel':soilmoistLevel, 'guest':user}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```