Introduction

Smart Plant Communicator System With IBM Cloud

Overview

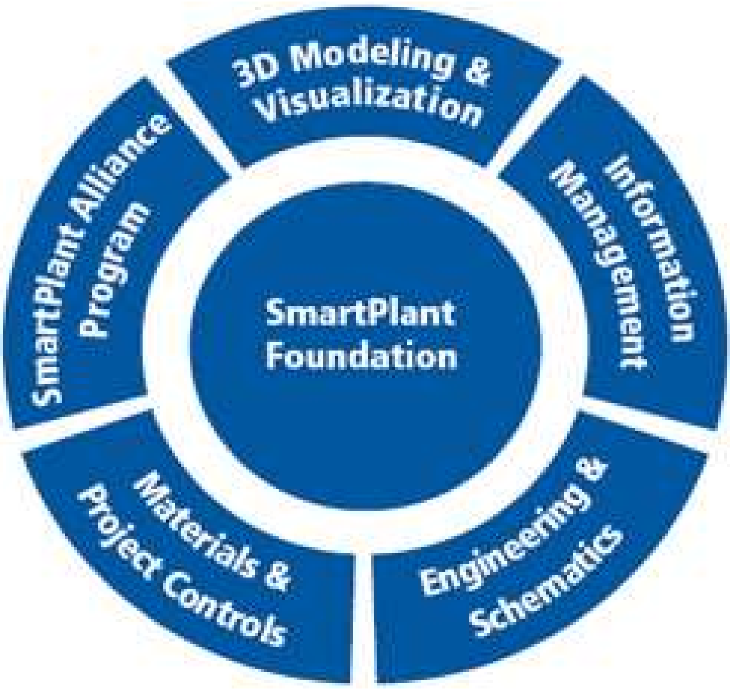
Smart Plant Communicator using IBM is seen as one of the most important tasks in any indoor and outdoor cultivation based environment. Integration of Python IDLE with plant monitoring makes planting/ floriculture easier. In this documentation, we discuss about the implementation of a smart plant communicator using IBM which makes use of the concept of python coding with the use of   a few IBM services which, proactively handles the plant monitoring system. The given implementation works along with a cloud based server and a mobile based device (ideally Android device) which helps the user to control and see the status of the plant which is being monitored by the software device. The given python code detects changes in the moisture, temperature, humidity ,distance ,water tank level and pH level in and around the plant, and performs a software based curation on the plant by providing necessary cultivation and maintaining soil moisture of the plant. Software curation is also integrated with IBM Node red service which are deployed in the cloud based server .For user based curation,the Android device provides user an option to display sensor data values and control the motor ON/OFF manually   based on the sensor data.

Purpose

Plant monitoring is seen as one of the most important tasks in any farming or agriculture based environment. With the inception of Ambient Intelligent systems, there have been a rise in ambient intelligent based devices-Smart Homes and other similar technologies involving IOT has evolved over the past few years Integration of such an ambient intelligent system with plant monitoring makes farming easier.

It has been widely used in todays generation as it can continuously monitor the status of the sensors and provide signal for taking necessary action.

It is used to observe the parameters such as humidity,temperature,ph value for a better yeild and makes us provide water to crop from time to time.Creating something that can grow plants and food for you!A living and breathing gift that lives for several decades and significantly benefits the planet.


3D Modeling & Visualization
Information Management
Engineering & Schematics
Materials & Project Controls
SmartPlant Alliance Program
SmartPlant Foundation

Literature Survey

Existing problem

The problem statement given to us is -

1. Send random sensor data temperature, humidity,PH levels in the plant which helps in the plant growth.
2. Receive data from the cloud to control the water pump if any person is detected based on the sensor value should greet them with welcome messages.
3. Use IBM text to speech service to greet the person with welcome messages also based on the different sensor values generate the voice output like plant is happy or sad.

Proposed solution

The solution for the above problem statement is -

1. write a python code to display the temperature,humidity and ph values to the cloud.

2. create a nodered by taking the sensor data and display in the web ui.

3. For the development of mobile app we will create 2 http nodes in the nodered ,moniter the data and control on off buttons.

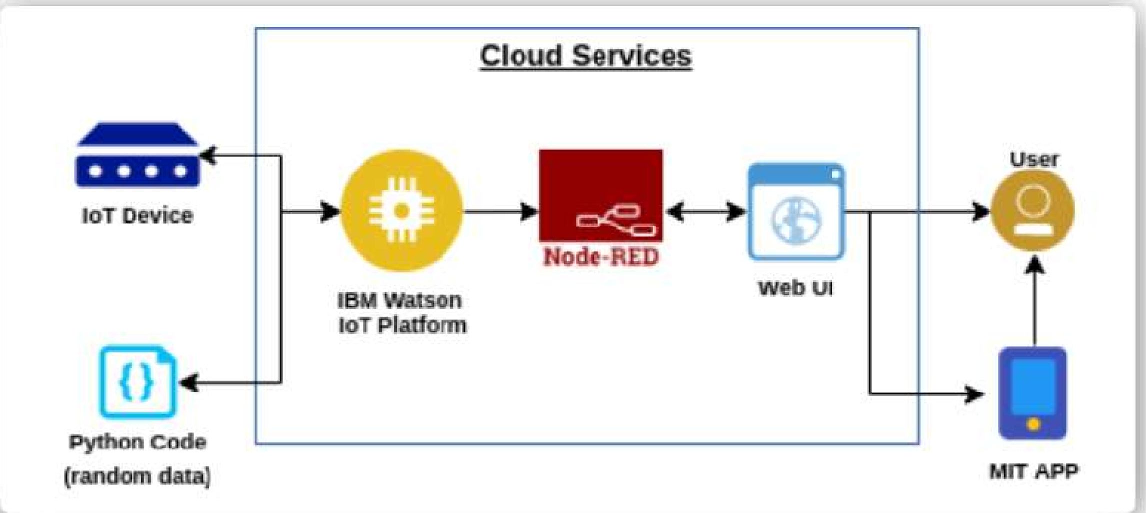
4. when a person is detected near the plant it is said to greet.

5. Depending on the temperature,humidity,moisture and ph values in the stipulated range plant is said to be happy or sad.

note:placing the button in nodered and controlling is an exception process and is benificial incase of any sensor failure.

Theoretical Analysis

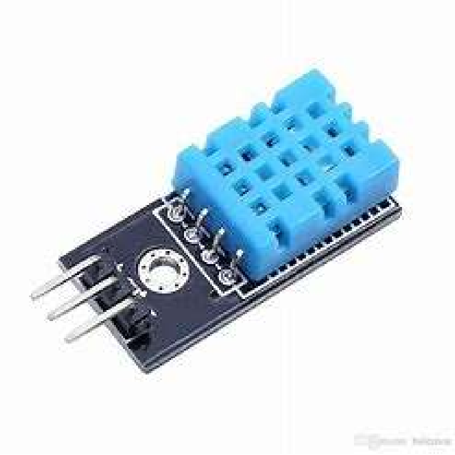
Block diagram



Hardware / Software designing

Hardware designing

TEMPERATURE AND HUMIDITY SENSOR (also known by DHT-11):It is a low cost humidity and temperature sensor with a single wire digital interface. The sensor is calibrated and doesn't require extra components so you can get right to measuring relative humidity and temperature.

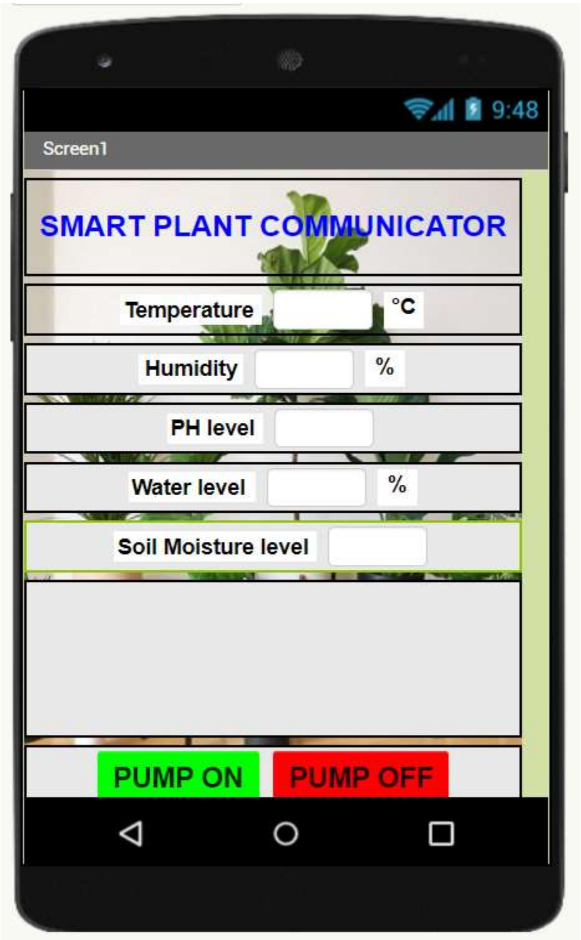


Ultrasonic sensor: An **ultrasonic sensor** is an instrument that measures the distance to an object using ultrasonic sound waves.It uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity.

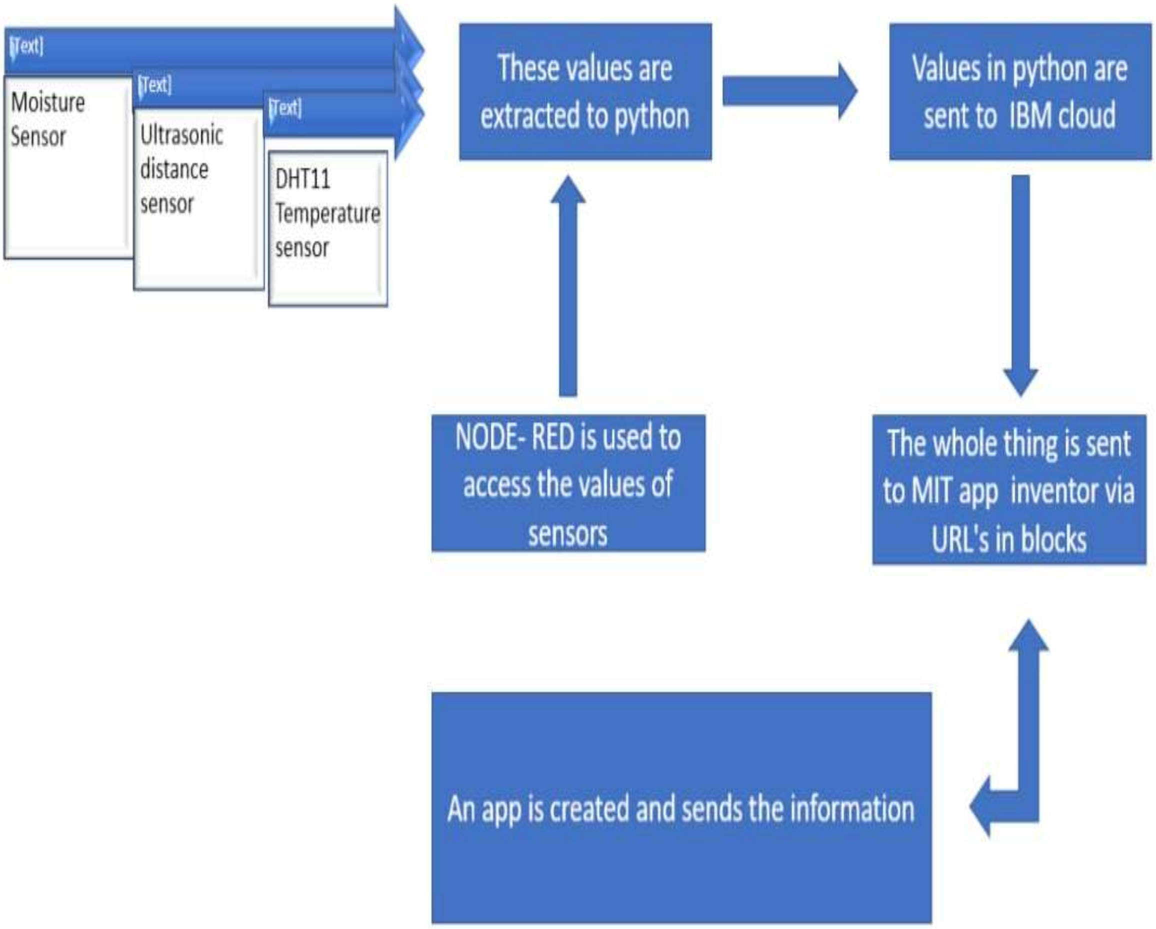
Moisture sensor:A Soil Moisture Sensor is one kind of low-cost electronic sensor that is used to detect the moisture of the soil. This sensor can measure the volumetric content of water inside the soil. This sensor is consists of mainly two parts, one is **Sensing Probs** and another one is the **Sensor Module.**

software designing

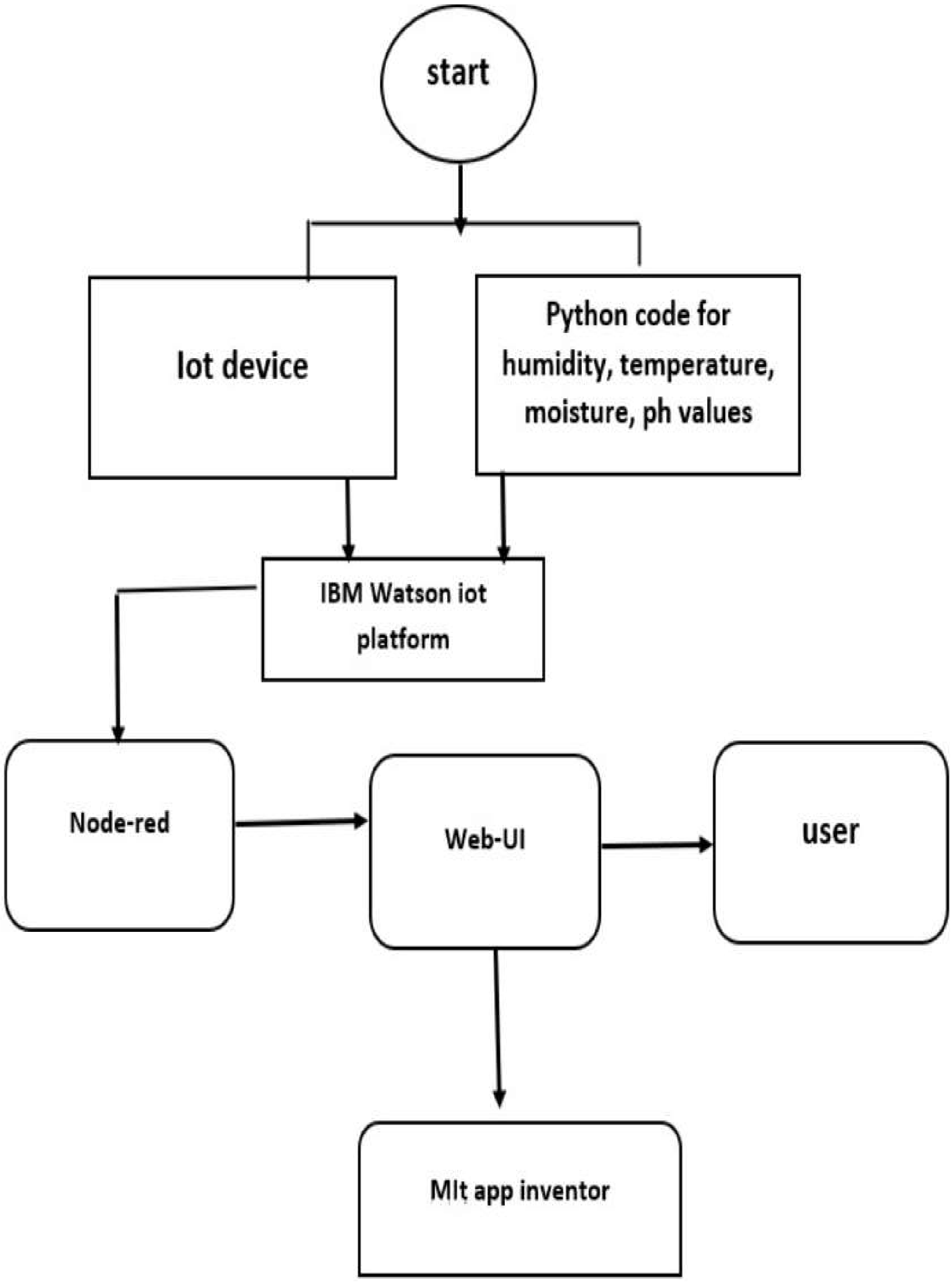
This is based on Internet of Things.software applications such as python tool,IBM cloud platform,Nodered block,MIT app inventer are used. It work on the mobile application and on the web server by uploading the data to the application available on PC. In mobile application we upload the wifi hotspot name and the password in our program based on python programming language. In web server on a particular IP address or the web page data is uploaded on the server and through the sensors data is uploaded on the web server. connectivity is also programmed in ESP32 by using the library availbale in the Arduino application. It is shown in the figure how the button is selected in the application and how the temperature, moisture,Ph value and humidity is uploaded on the web.



Experimental Investigations


[Text]
[Text]
[Text]
Moisture Sensor
Ultrasonic distance sensor
DHT11 Temperature sensor
These values are extracted to python
Values in python are sent to IBM cloud
NODE- RED is used to access the values of sensors
The whole thing is sent to MIT app inventor via URL's in blocks
An app is created and sends the information

Flowchart

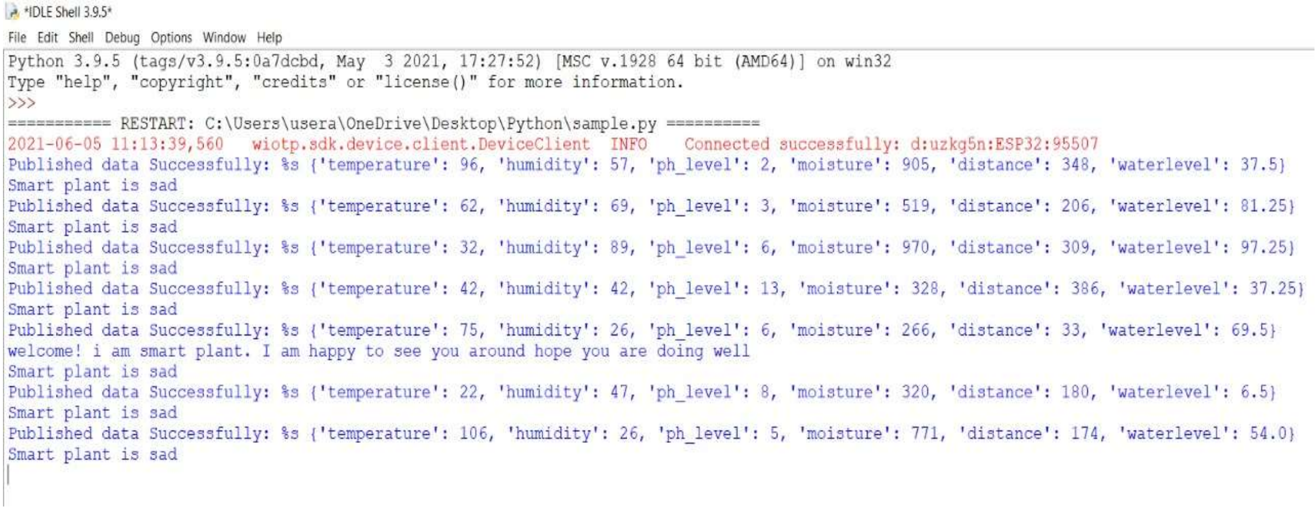

start
Iot device
Python code for humidity, temperature, moisture, ph values
IBM Watson iot platform
Node-red
Web-UI
user
MIt app inventor

Result

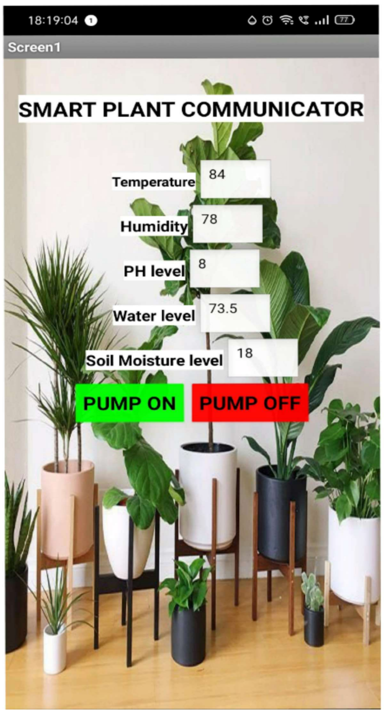
Firstly we achieved to extract the data from python(random sensor data).

secondly,we were able to constuct an app and communicate with it.

finally,we were able to get the expected output(audio output).



```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: C:\Users\usera\OneDrive\Desktop\Python\sample.py ==========
2021-06-05 11:13:39,560   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:uzkg5n:ESP32:95507
Published data Successfully: %s {'temperature': 96, 'humidity': 57, 'ph_level': 2, 'moisture': 905, 'distance': 348, 'waterlevel': 37.5}
Smart plant is sad
Published data Successfully: %s {'temperature': 62, 'humidity': 69, 'ph_level': 3, 'moisture': 519, 'distance': 206, 'waterlevel': 81.25}
Smart plant is sad
Published data Successfully: %s {'temperature': 32, 'humidity': 89, 'ph_level': 6, 'moisture': 970, 'distance': 309, 'waterlevel': 97.25}
Smart plant is sad
Published data Successfully: %s {'temperature': 42, 'humidity': 42, 'ph_level': 13, 'moisture': 328, 'distance': 386, 'waterlevel': 37.25}
Smart plant is sad
Published data Successfully: %s {'temperature': 75, 'humidity': 26, 'ph_level': 6, 'moisture': 266, 'distance': 33, 'waterlevel': 69.5}
welcome! i am smart plant. I am happy to see you around hope you are doing well
Smart plant is sad
Published data Successfully: %s {'temperature': 22, 'humidity': 47, 'ph_level': 8, 'moisture': 320, 'distance': 180, 'waterlevel': 6.5}
Smart plant is sad
Published data Successfully: %s {'temperature': 106, 'humidity': 26, 'ph_level': 5, 'moisture': 771, 'distance': 174, 'waterlevel': 54.0}
Smart plant is sad
```



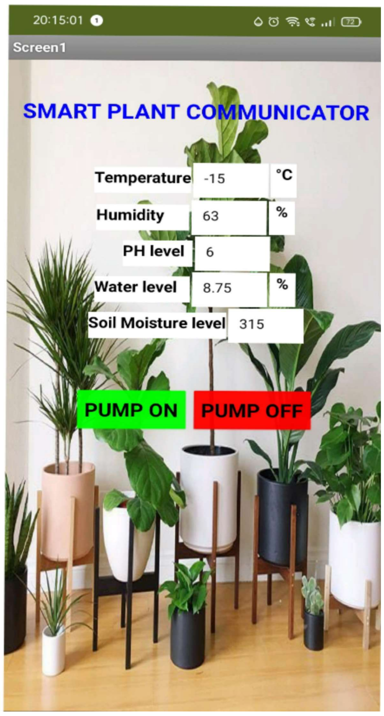
Temperature 84

Humidity 78

PH level 8

Water level 73.5

Soil Moisture level 18



Temperature -15 °C

Humidity 63 %

PH level 6

Water level 8.75 %

Soil Moisture level 315

Advantages

- Increased crop yeild.
- Improved data accuracy through better resolution of design inconsistencies
- Increased reliability through rule-driven, intelligent, graphical comparison.
- Reduced labor by decreasing the number of iterative cycles.
- helps to shorten time to market
- Improve compliance with safety and environmental regulations
- Effecient way to engineer to manage your instrumentation and control system.

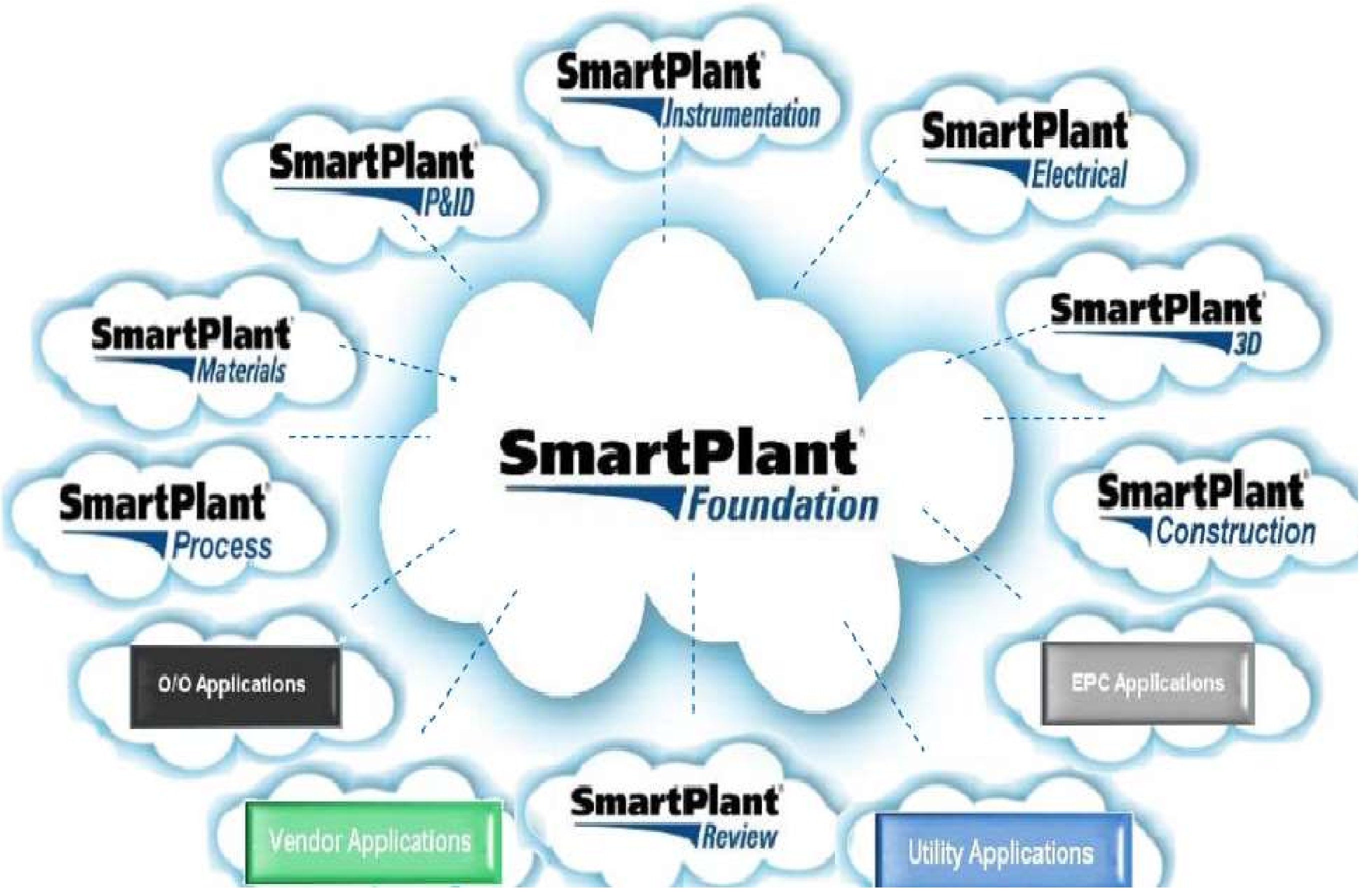
Disadvantages

- lack of accuracy in sandy soils
- little expensive
- knowledge needed to grow crops succesfuly
- depending upon the size of property you need more systems

Note:Due to pandemic we were unable to get hardware equipment so we initialised 2 buttons to control the pump.If there is a need to make it automated we could make changes in python and nodered accordingly(if moisture levels are low pump will automaticlly on).

Applications

With the advent of Cloud Computing, Mobile Access, Web Security, Refined Standards and new SmartPlant Applications the Future Vision of Engineering and Supply Chain automation data integration with SmartPlant Instrumentation looks very promising



Conclusion

The efficient automation on monitoring and control of the plants require new and revolutionary solutions. Wireless sensor networks can respond to requirement by offering an accurate and easily configurable monitoring system.In this work we are using the the temperature sensor and ultrasonic sensor with which, we could efficiently monitor the basic resources of the plant .This is prototype of the monitoring and control system for plants.Unlike other automated systems which,our model is more "Intelligent" to utilize the resources according to the changes in weather conditions. Our model has the capability to integrate with any mobile platform, Since the service is running on a
cloud based service it is scalable.

Future Scope

Future work would be focused more on increasing sensors on this system to fetch more data especially by integrating different modules in this system to enhance this agricultural Iot technology to full fledged agriculture .precision ready product.

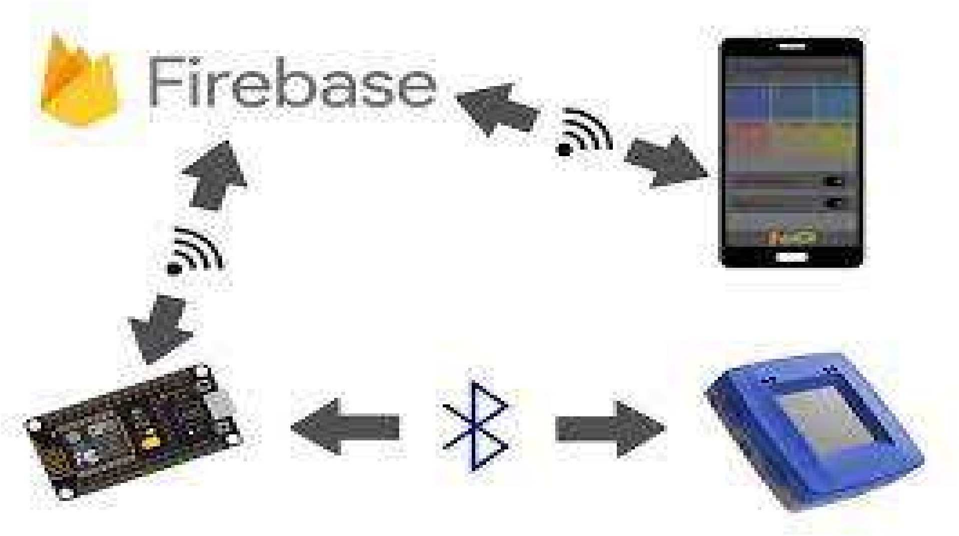
Bibliography

1.https://www.researchgate.net/publication/283123947_Smart_Plant_Monitoring_System

2.Engineering & Schematic Product Update (spi-ltuf.org)

3.https://www.slideshare.net/SjaakW

4.IOT Based Smart Garden Monitoring System (ijser.org)

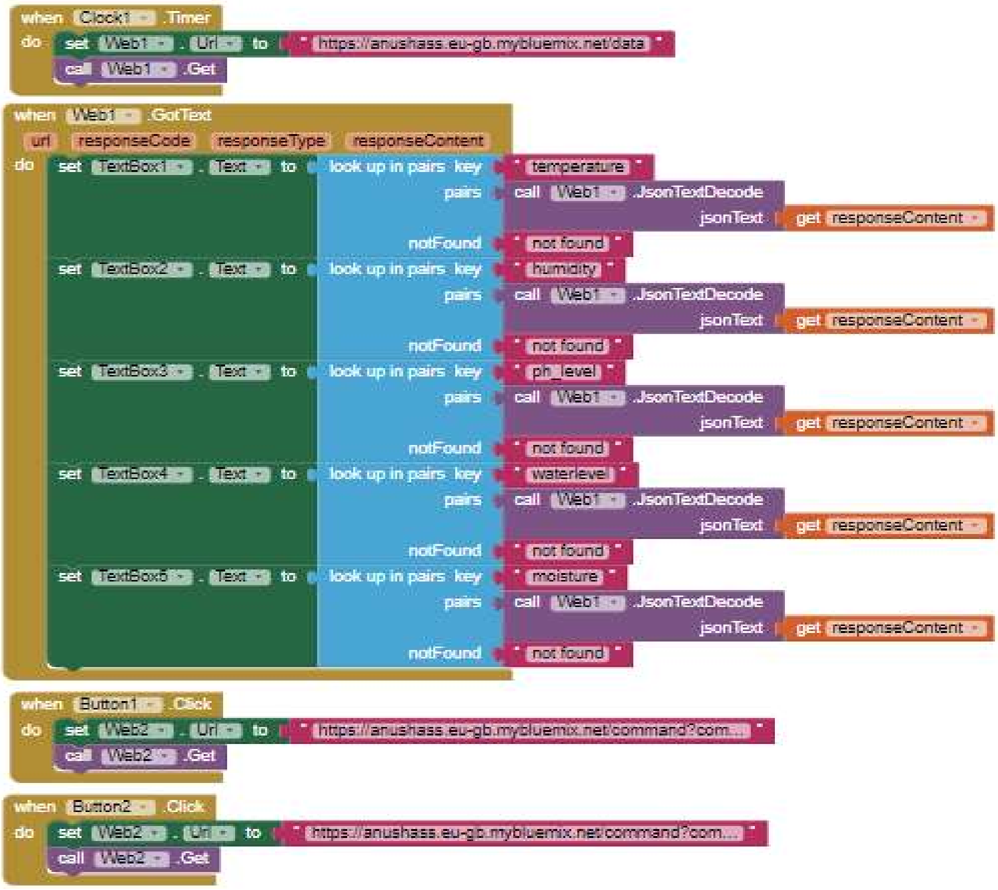


Appendix

Source code

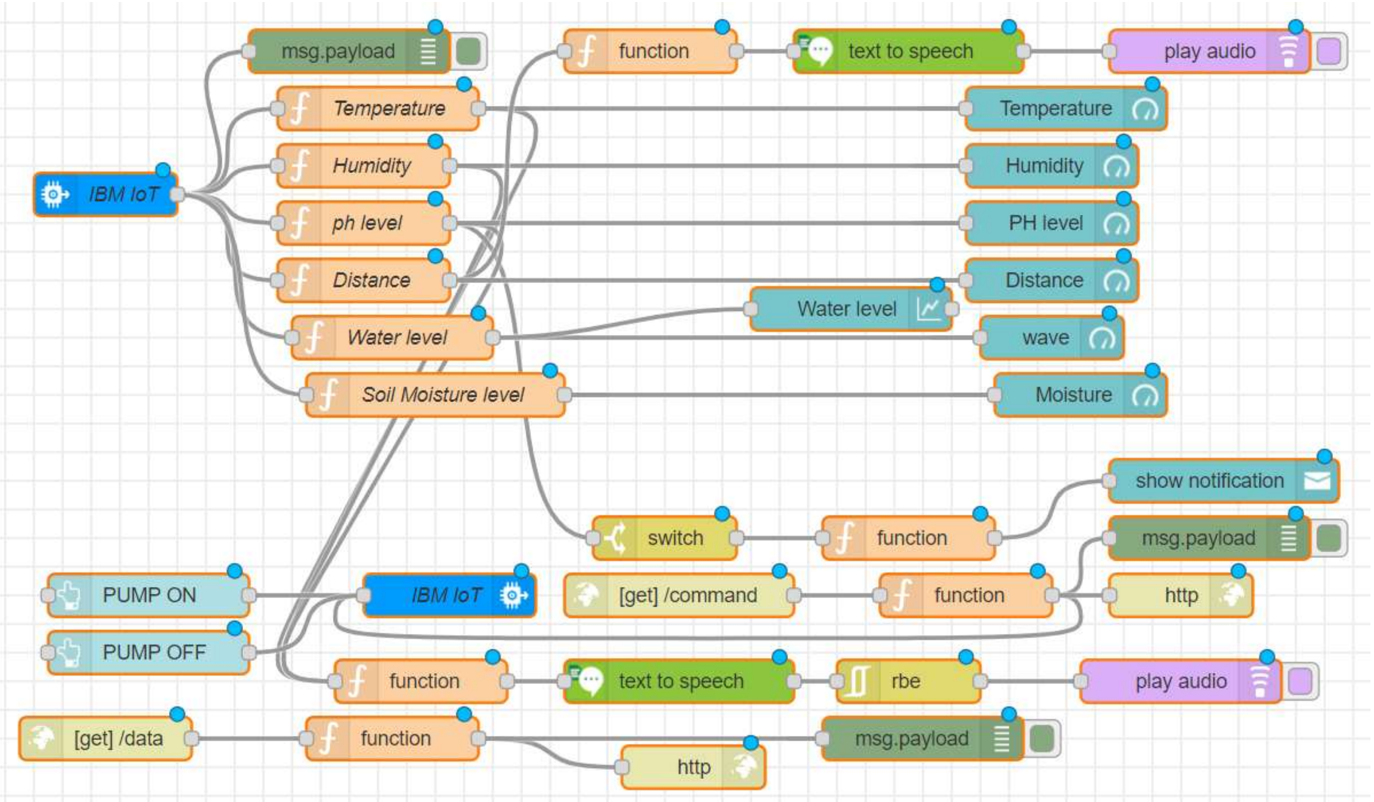
```
 import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "uzkg5n",
        "typeId": "ESP32",
        "deviceId":"95507"
    },
    "auth": {
        "token": "12345678"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data)
    m=cmd.data
# The manual operation  ON/OFF of water pump is required in case of any SENSOR FAILURE

    if(m=={"COMMAND":"PUMPON"}):
        print("PUMP ON")
    if(m=={"COMMAND":"PUMPOFF"}):
          print("PUMP OFF")
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    ph_level=random.randint(0,14)
    dis=random.randint(2,400)
    mo=random.randint(0,1023)
    level=random.randint(2,400)
    waterlevel = (level/400)*100
    myData={'temperature':temp, 'humidity':hum, 'ph_level':ph_level, 'moisture':mo, 'distance':dis, 'waterlevel':waterlevel}
    time.sleep(6)
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)

    if mo > 600 :
        print("PUMP ON")
    else:
        print("PUMP OFF")
# Testing whether the below print statements are in sync with the IBM text to speech service OR NOT

    if dis < 50 :
     print("welcome! i am smart plant. I am happy to see you around hope you are doing well")
     time.sleep(2)
    if((temp < 50) and (ph_level < 5.5)):
        if((hum > 50) and (hum < 70)):
          if((mo > 400) and (mo < 600)):
             print("Smart plant is happy")
             time.sleep(3)
    else:
             print("Smart plant is sad")
             time.sleep(3)
    client.commandCallback = myCommandCallback
    time.sleep(1)
client.disconnect()
```

MIT Block

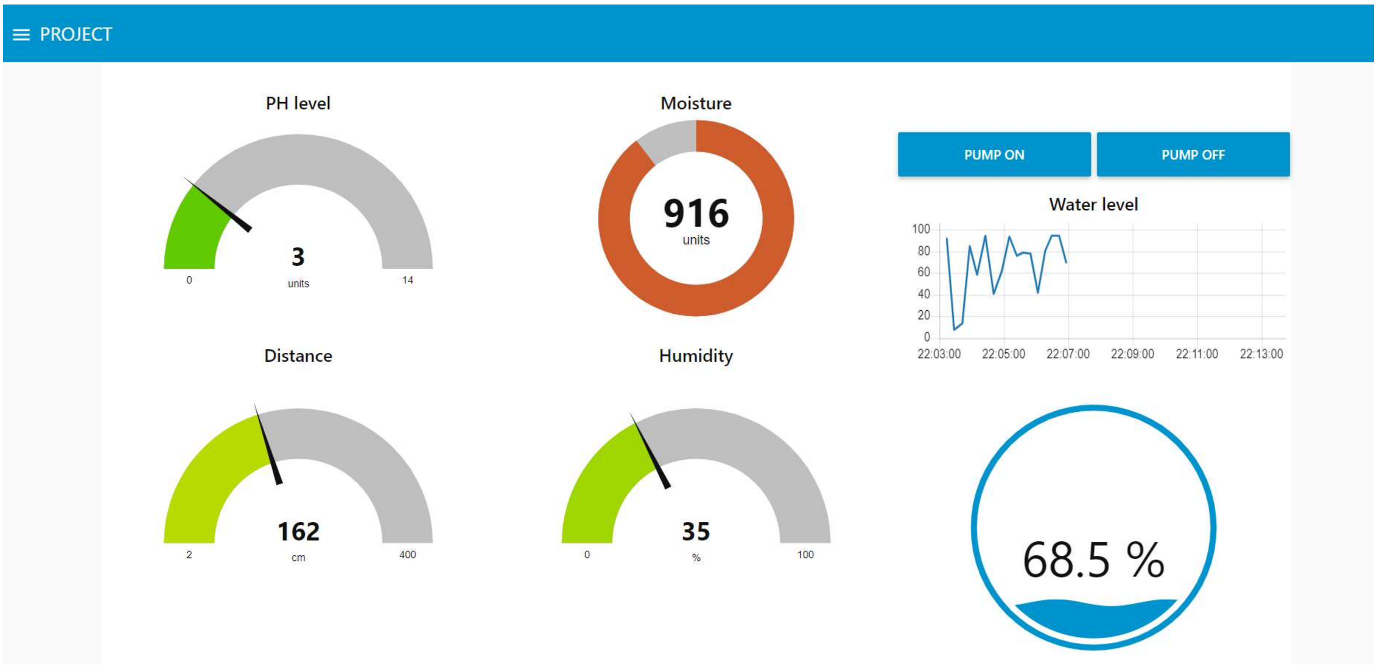


```
when Clock1 .Timer
do  set Web1 . Url to " https://anushass.eu-gb.mybluemix.net/data "
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key " temperature "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "
    set TextBox2 . Text to  look up in pairs  key " humidity "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "
    set TextBox3 . Text to  look up in pairs  key " ph_level "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "
    set TextBox4 . Text to  look up in pairs  key " waterleve| "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "
    set TextBox5 . Text to  look up in pairs  key " moisture "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "

when Button1 .Click
do  set Web2 . Url to " https://anushass.eu-gb.mybluemix.net/command?com... "
    call Web2 .Get

when Button2 .Click
do  set Web2 . Url to " https://anushass.eu-gb.mybluemix.net/command?com... "
    call Web2 .Get
```

Nodered block



UI output Screenshot



Team members-

G.Abhinav (19R11A04MO)

G.Guna sundeep(19R11A04M2)

P.Harika (19R11A04M3)

S.Anusha(19R11A04P1)