Project Report

On

ARDUINO BASED SMART IRRIGATION SYSTEM USING IOT

By:

Adepu Rohithkhanna

Badavath Vamshi

as an interns at

https://smartinternz.com/rsip2020

ON:

INTERNET OF THINGS

ARDUINO BASED SMART IRRIGATION SYSTEM USING IOT:

An automated irrigation system for efficient water management and intruder detection system has been proposed. Soil Parameters like soil moisture, pH, Humidity are measured and the Pressure sensor and the sensed values are displayed in LCD.

ABSTRACT

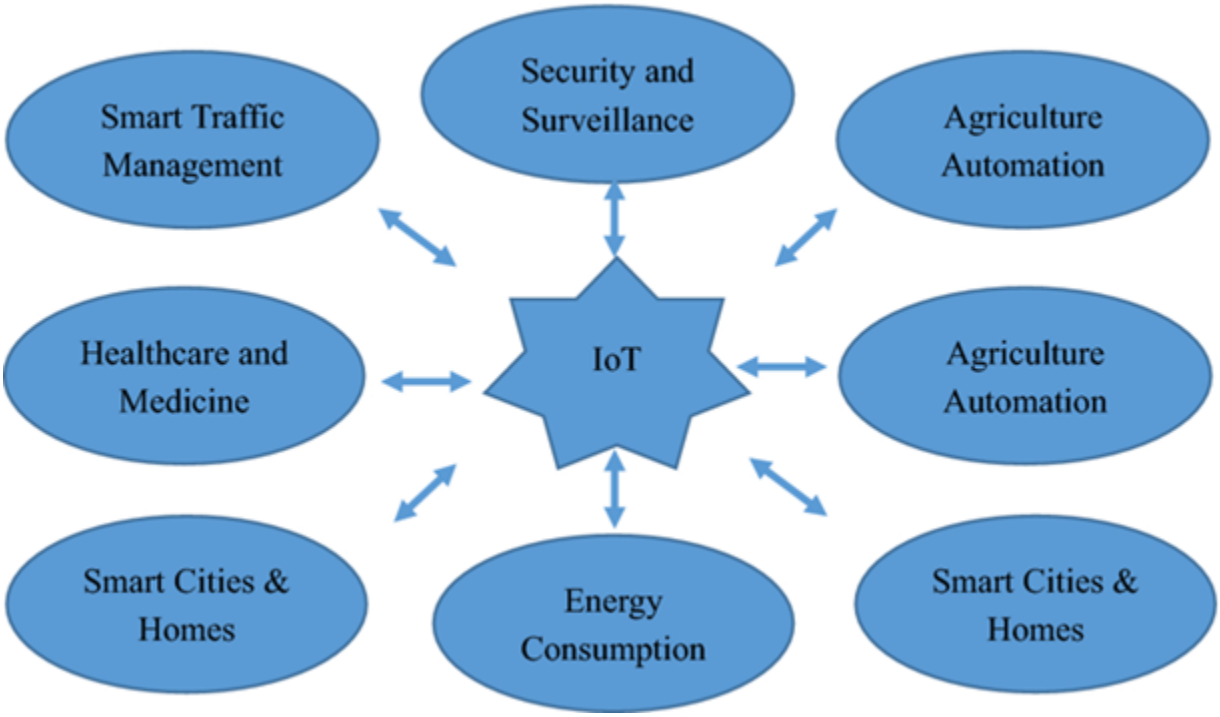
The project presents the use of correct soil moisture sensors which helps to ease out the pain to monitor and keep records about the changes in soil moisture. Using the Arduino Mega micro controller with Light-Depended Resistor sensor, moisture sensor and temperature sensor, temperature are measured and analyzed. The soil for a certain duration, provides information related to the moisture status of

the soil. The Arduino Mega will collect and process the data received from the Sensors. When a threshold moisture level of the soil is reached, the water will supply accordingly. This is essential because water must be provided to the plant at a particular time for a good yield. This project is highly use for farmers, Nursery professionals by eradicating traditional or manual method of irrigation system.

INTRODUCTION

Internet of Things can be defined as the collection of two terms: one is Internet, which is defined as networks of networks which can connect billions of users with some standard internet protocols. Internet connect several different sectors and department while using different technologies. Several devices like mobile, personal systems and business organizations are connected to Internet. The second term is Thing, this term is basically mean to these devices or objects which turn into intelligent objects. More over this it is also a part of all objects of this real world. If we want to define IOT then we cannot define it precisely and concisely but Vermesan et al. defined the Internet of Things as simply an interaction between the physical and digital worlds. The digital world interacts with the physical world using a plethora of sensors and actuators .

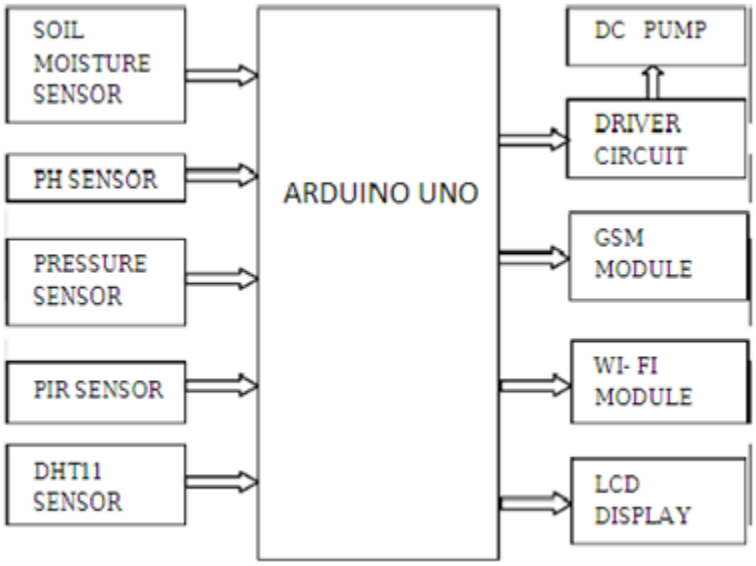
LITERATURE SURVEY:



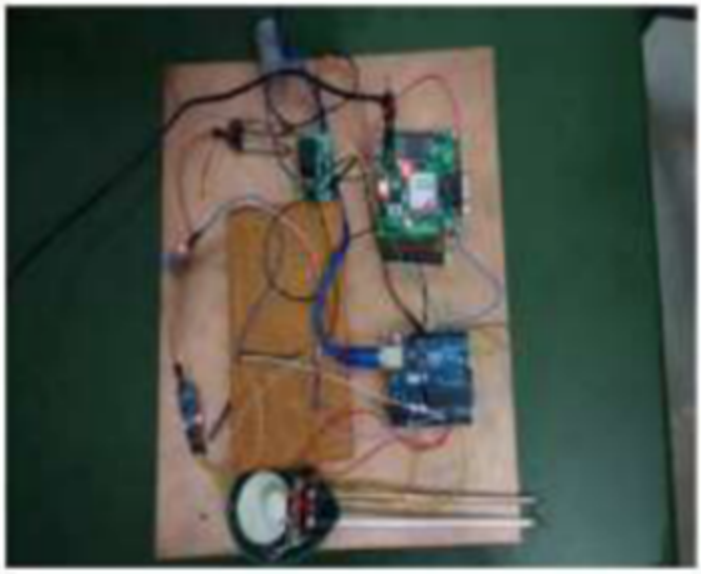
THEORTICAL ANALYSIS:-

IoT has a multidisciplinary vision to provide its benefit to several domains such as environmental, industrial, public/private, medical, transportation etc. Different researchers have explained the IoT differently with respect to specific interests and aspects. The potential and power of IoT can be seen in several application domains. Illustrates few of the application domains of IoTs potentials

Block diagram:



HARDWARE DESIGNING:



ADVANTAGES

1. **Data**: The more the information, the easier it is to make the right decision. Knowing what to get from the grocery while you are out, without having to check on your own, not only saves time but is convenient as well.

2. **Tracking**: The computers keep a track both on the quality and the viability of things at home. Knowing the expiration date of products before one consumes them improves safety and quality of life. Also, you will never run out of anything when you need it at the last moment.
3. **Time**: The amount of time saved in monitoring and the number of trips done otherwise would be tremendous.
4. **Money**: The financial aspect is the best advantage. This technology could replace humans who are in charge of monitoring and maintaining supplies.

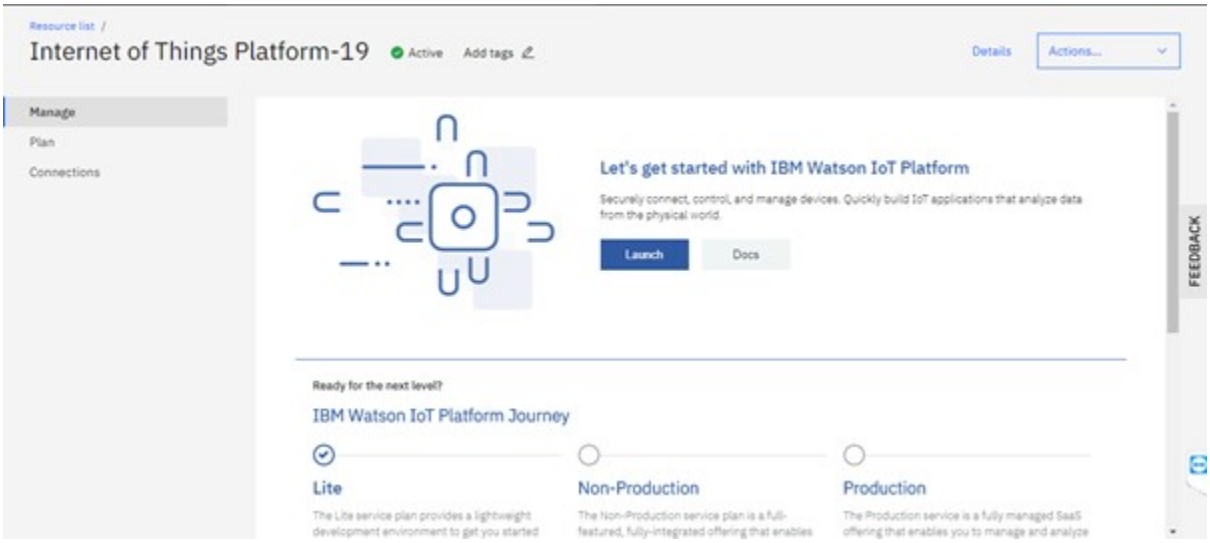
DISADVANTAGES

1. **Compatibility**: As of now, there is no standard for tagging and monitoring with sensors. A uniform concept like the USB or Bluetooth is required which should not be that difficult to do.
2. **Complexity**: There are several opportunities for failure with complex systems. For example, both you and your spouse may receive messages that the milk is over and both of you may end up buying the same. That leaves you with double the quantity required. Or there is a software bug causing the printer to order ink multiple times when it requires a single cartridge.
3. **Privacy/Security**: Privacy is a big issue with IoT. All the data must be encrypted so that data about your

financial status or how much milk you consume isn't common knowledge at the work place or with your friends. 4. **Safety**: There is a chance that the software can be hacked and your personal information misused. The possibilities are endless. Your prescription being changed or your account details being hacked could put you at risk. Hence, all the safety risks become the consumer's responsibility.

Designing Procedure:-

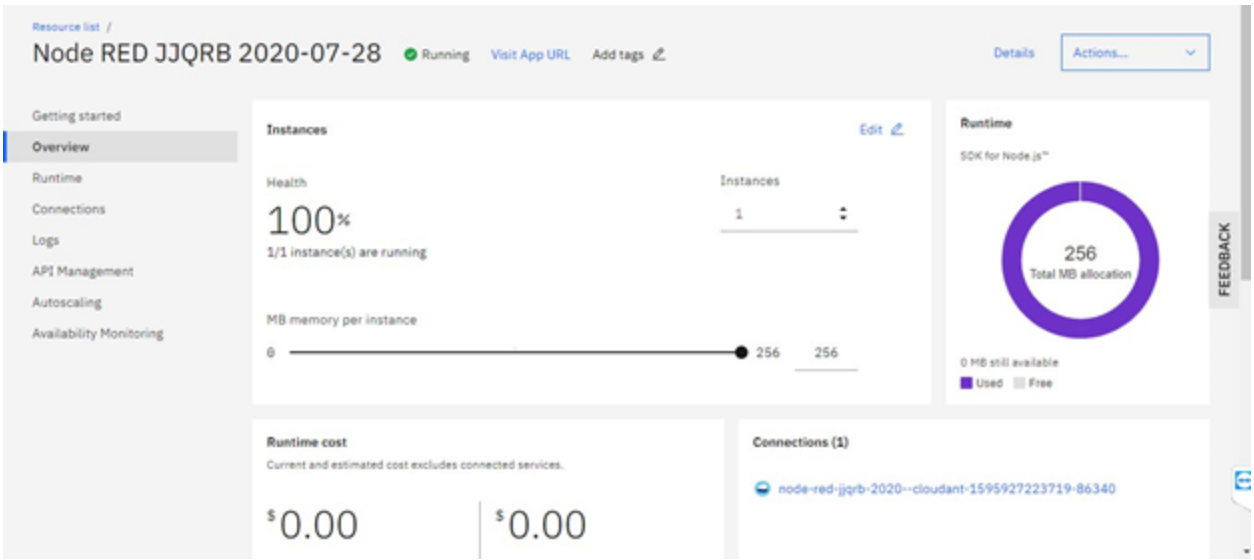
1. Sign-in to your IBM cloud account from the link cloud.ibm.com. There, go to Catalog and search for IoT in the search bar. Then select Internet of Things platform and subscribe for the desired plan and click create. Now, in the menu, go to Resource List - Services - Internet of Things Platform and then click Launch, as shown below:

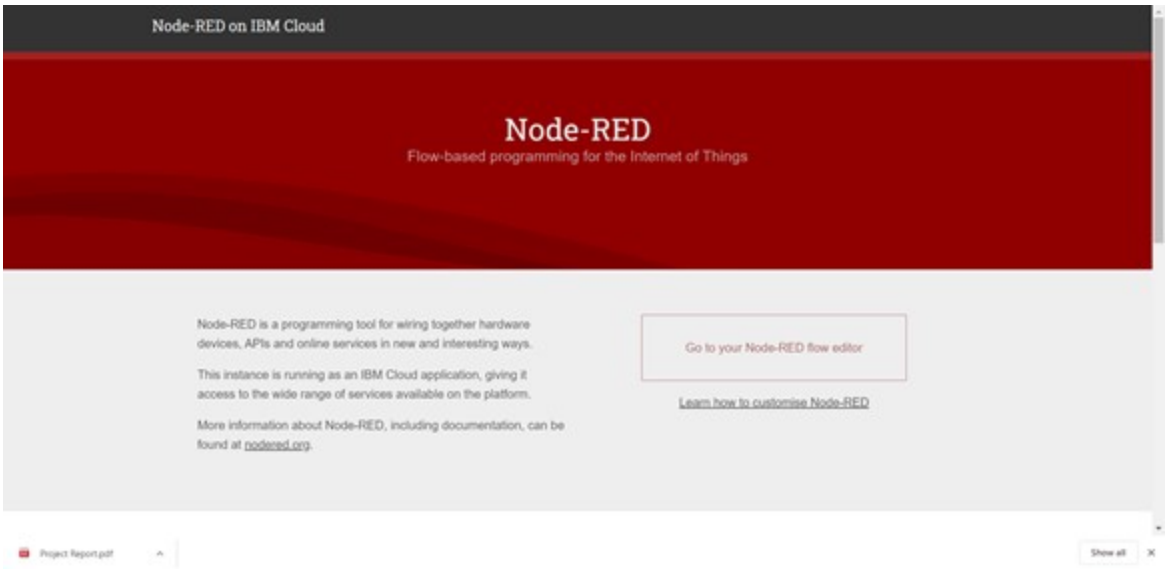


1. Now in the Watson IoT platform, click on the Add Device button at the top right corner, as shown below, to create a device to get the soil conditions from the sensor (simulator).

Make a note of the device credentials given during the device on for further uses.

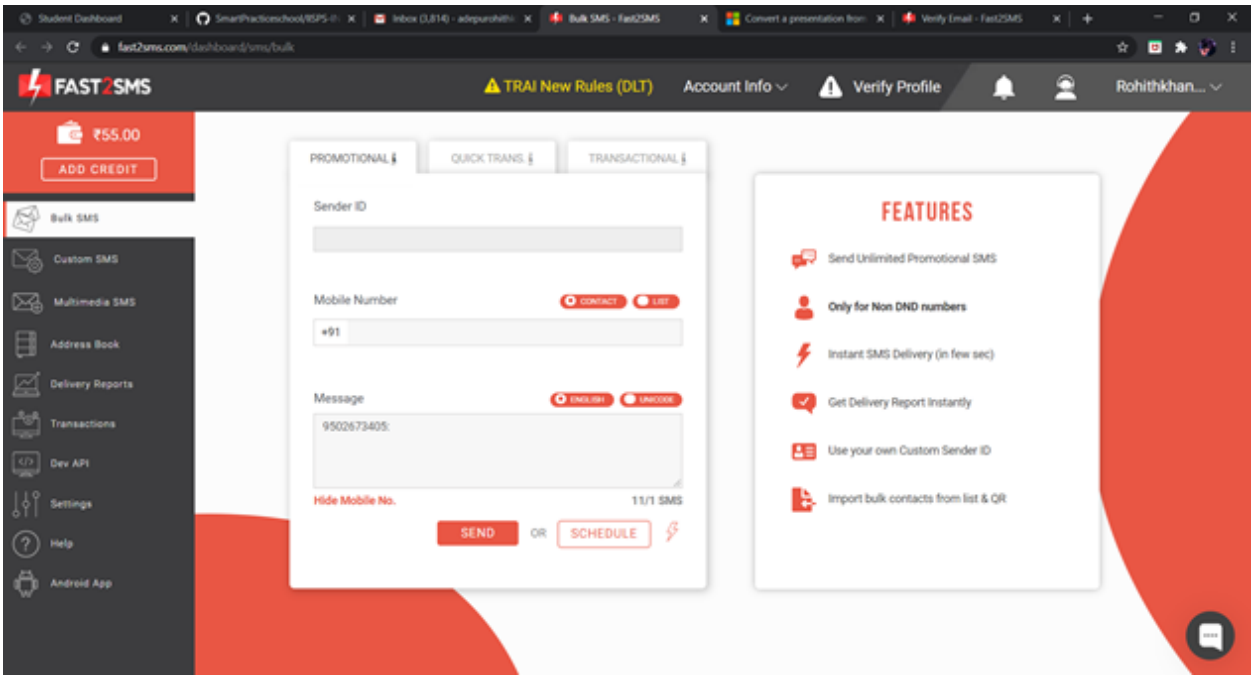
(3) Now, create Node-Red as shown here. click on Visit App url.Then you can see Main Page of Node-Red .





create a fast2sms account:(for sending alert messages):- now
create a fast2sms account to send alert messages touser

search fast2sms in bowser ,open that website and create an account there



code snippet for sending sensor data to the watsoniot platform and for sending the alert messages to the user

note: we don't have sensors to send the data to the cloud, so we sendsensor data with python code

The following code is the code used for this task

IN THE BELOW CODE ENTER THE CREDENTIALS OF THE DEVICE THAT U CREATED IN THE WATSON IOT PLATFORM

PYTHON CODE:

#try to use jupyter notebook while executing the program wait for atleast 40 seconds for the

entire program to run

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
#Provide your IBM Watson Device Credentials
organization = "h41b00"
deviceType = "rasberrypi"
deviceId = "9502673405"
authMethod = "token"
authToken = "9949213286"


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

        if cmd.data['command']=='motoron':
                print("Motor ON IS RECEIVED")


        elif cmd.data['command']=='motoroff':
                print("MOTOR OFF IS RECEIVED")


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"
```

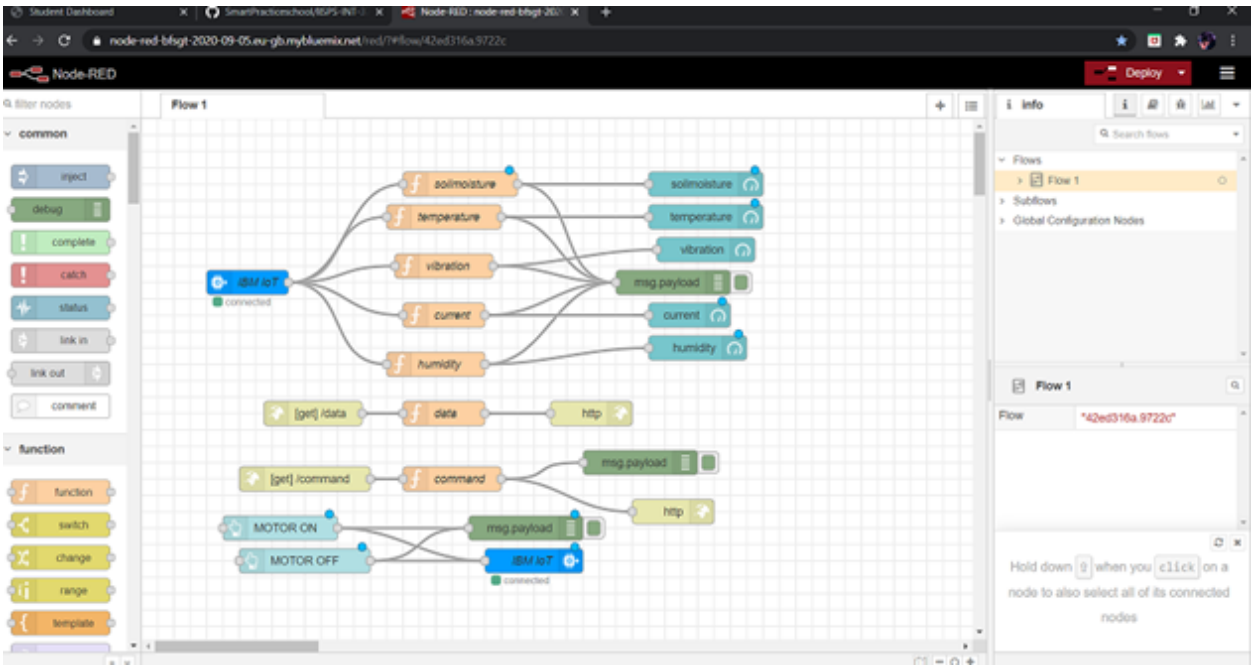
10 times
deviceCli.connect()

CREATE THE NODERED FLOW TO GET DATA FROM THE DEVICE AND HTTP REQUEST TO COMMUNICATE

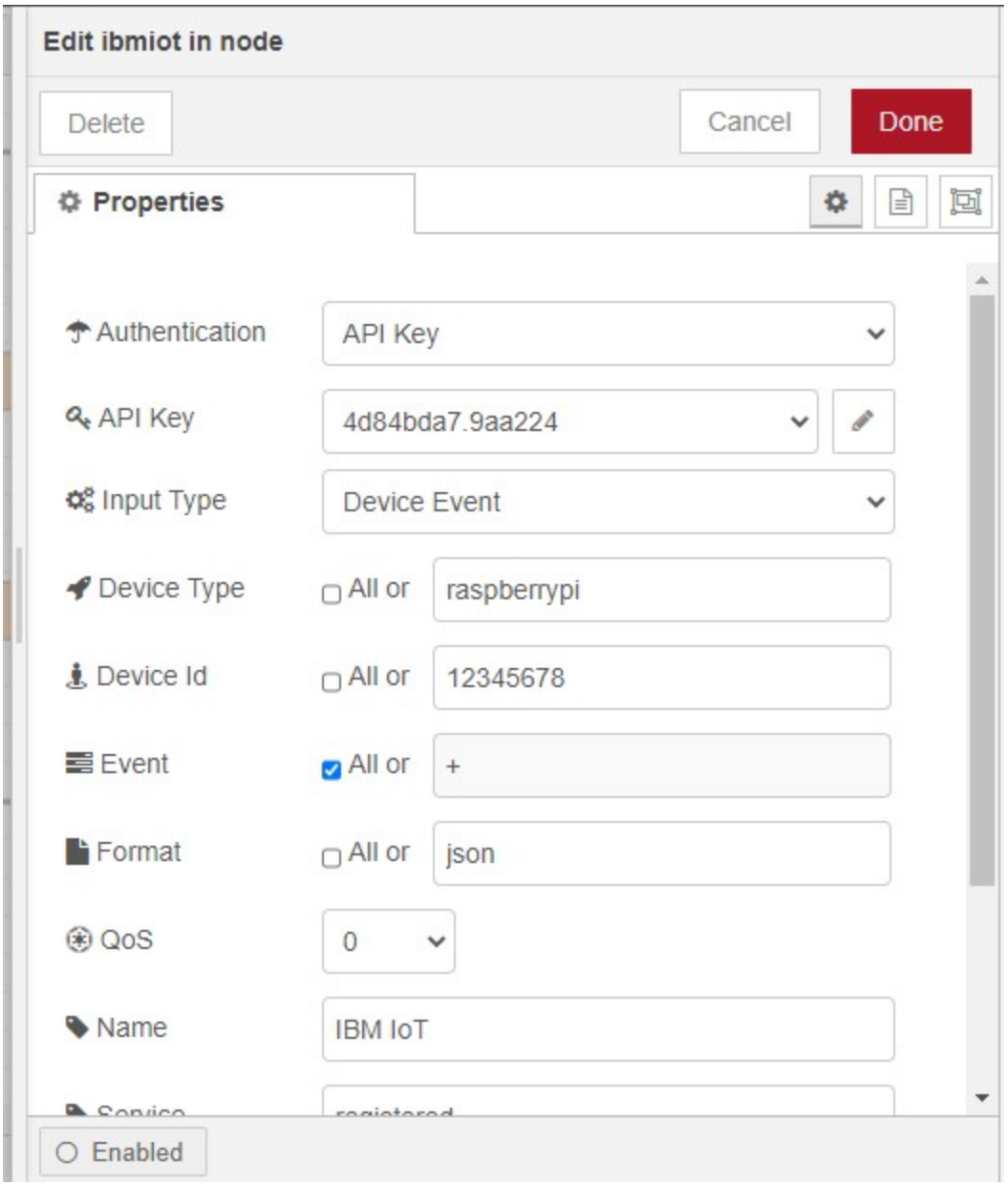
WITH THE MOBILE APP

we need to create flows to do this taskFLOW 1:

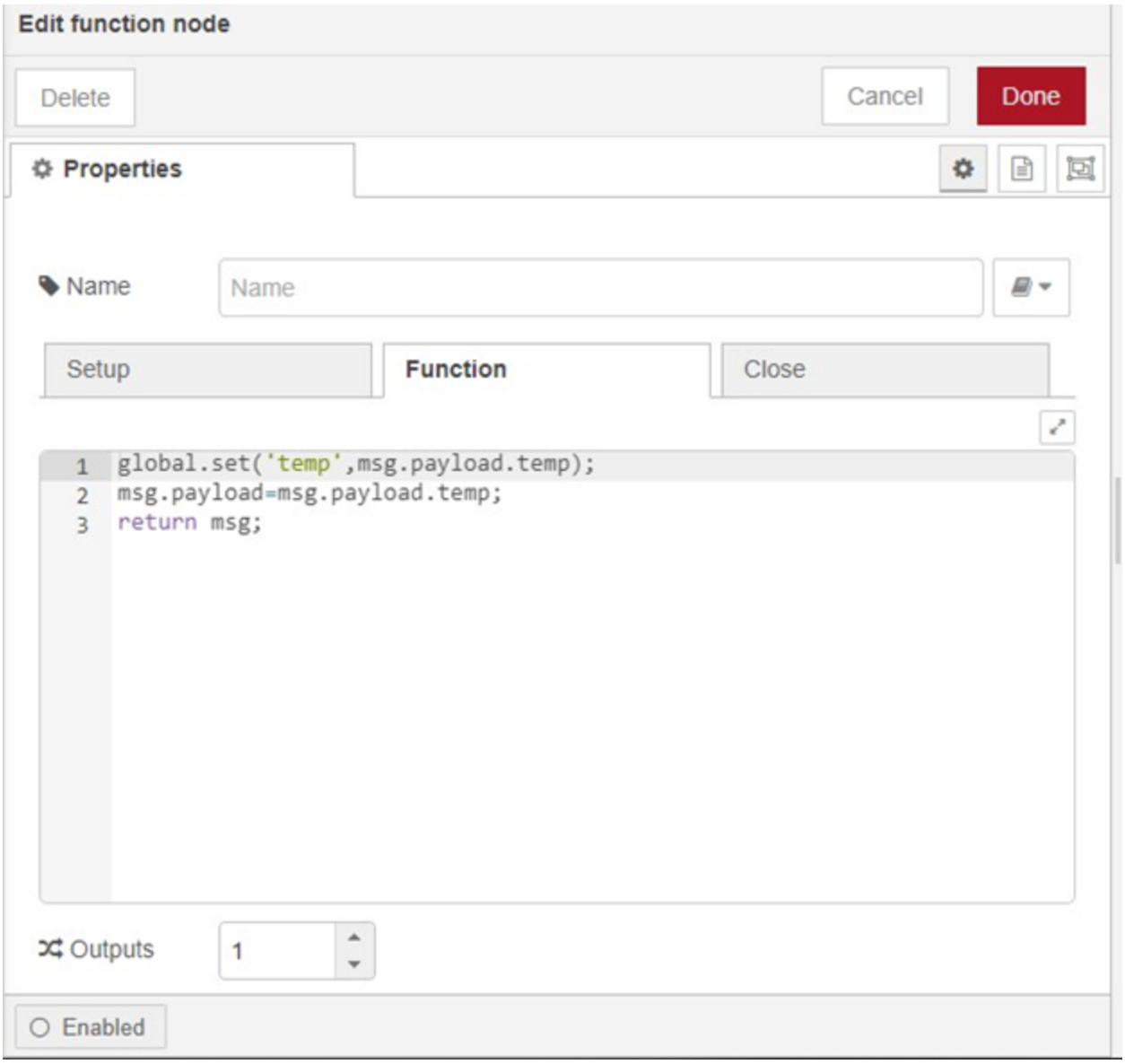
to get data from the device



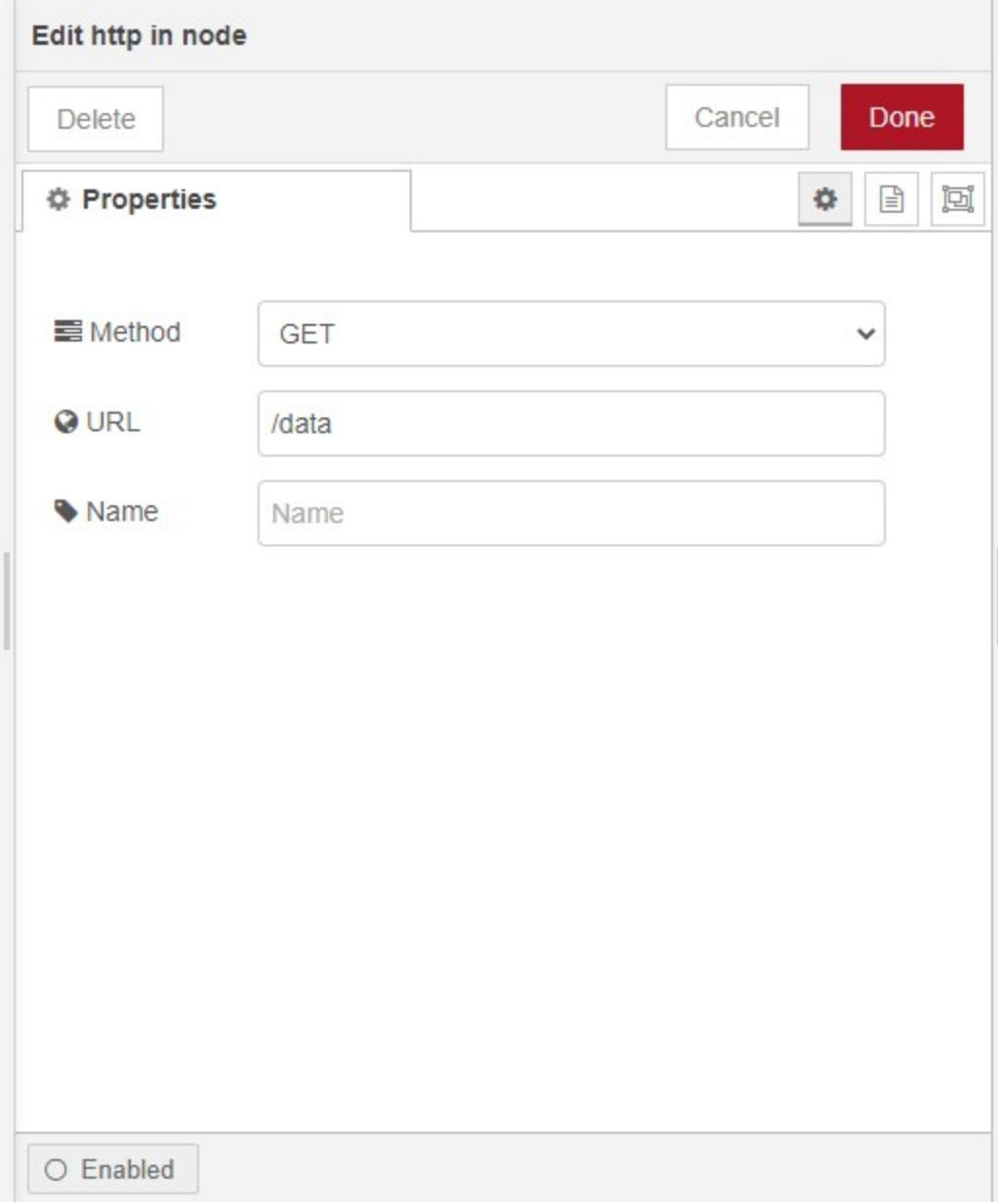
to connect ibmiot device node to the device double click on the node and enter the device credentials of the device that u have in your ibm iot watson platform



now the data that comes from the device is combined you need to parse the data and display data individuallycode the function node like this

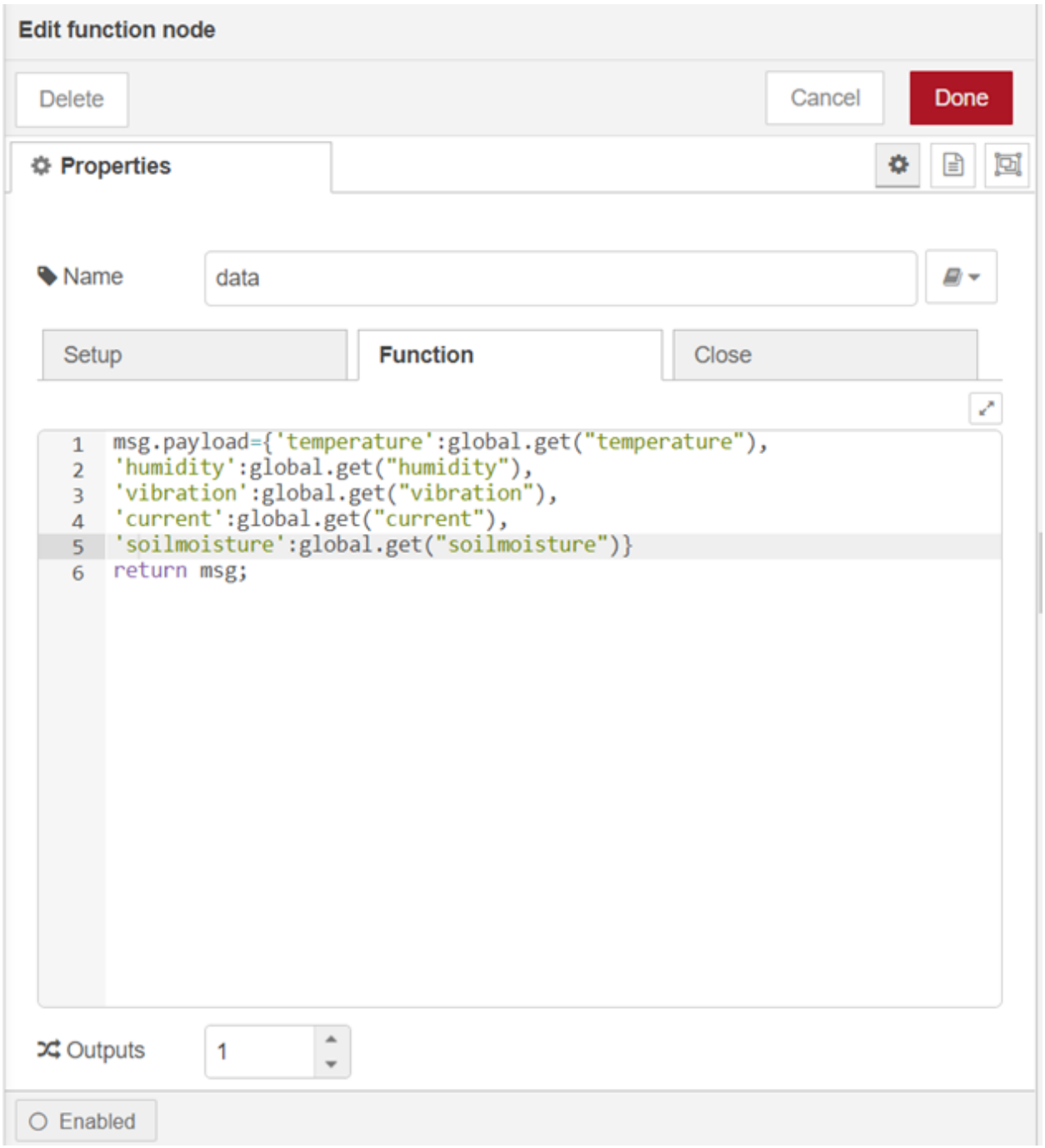


configure httpin node like this

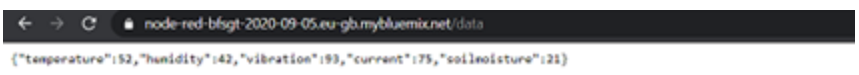


BY THIS FLOW WE ARE SENDING THE DATA TO SERVER

the data has been sent to the server looks like this



THE WEB APP UI WILL BE LIKE THIS



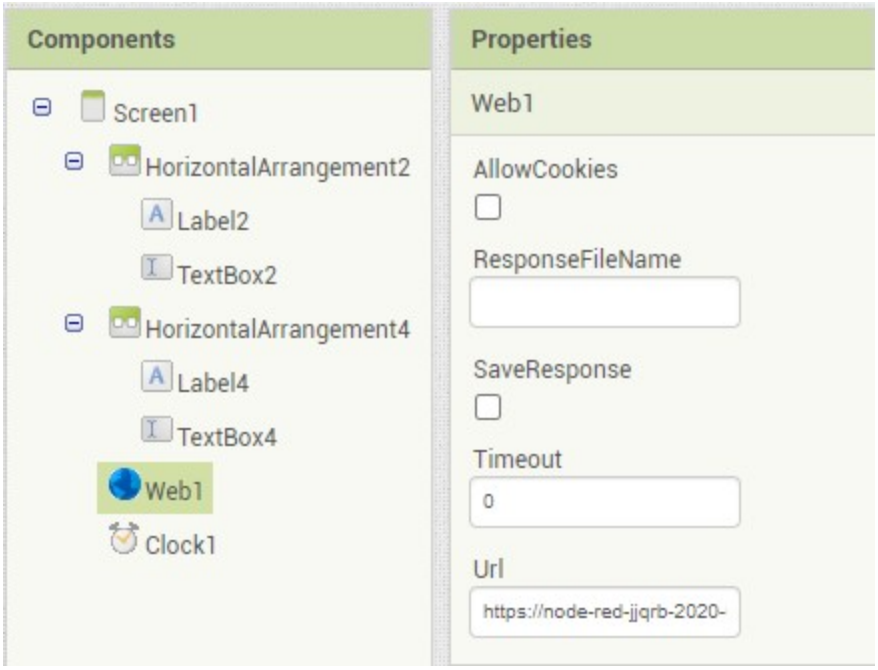
CREATE A MOBILE APP USING MIT APP INVENTOR & CONFIGURE IT TO GET DATA FROM CLOUD :

search mit app inventor in browser and open it

click on create apps on dashboard & login with ur google accountgive the name of your project without spacesconfigure the ui of your app like this

NOW DROP THE WEB FROM CONNECTIVITY ON TO THE BOARD &

ENTER THE URL IN THE WEB



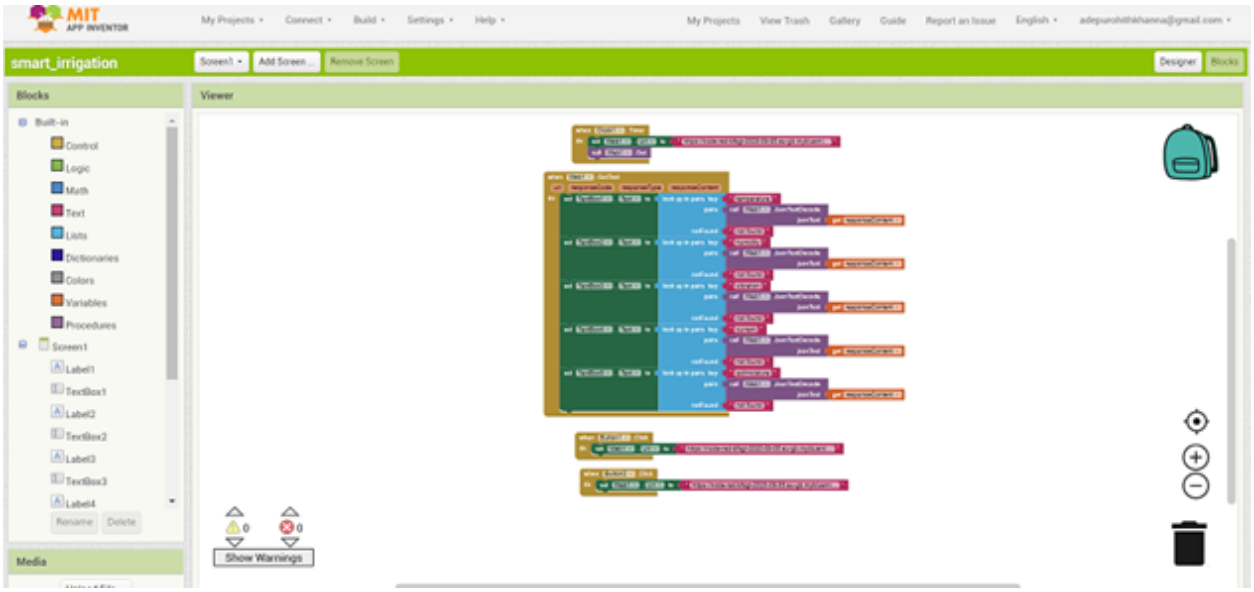
NOTE:

The app receives the data from the url that u enter In the web so you should enter the url that receives data from ibm device

enter this url

now click on the blocks on the top right corner of the screen and start arranging the blocks to create back end of the app

set the blocks in manner of text boxes



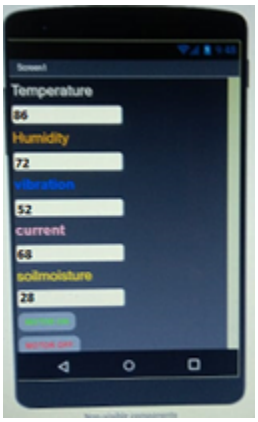
these blocks are there to decode data that is in the form of json

& display then in their respective text boxes

now click on build option on the top of dashboard & download the apk file

& install it in your mobile

app on mobile will be like this



PYTHON CODE:

#try to use jupyter notebook while executing the program wait for atleast 40 seconds for the entire program to run

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
#Provide your IBM Watson Device Credentials
organization = "h41b00"
deviceType = "rasberrypi"
deviceId = "9502673405"
authMethod = "token"
authToken = "9949213286"


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

        if cmd.data['command']=='motoron':
                print("Motor ON IS RECEIVED")


        elif cmd.data['command']=='motoroff':
                print("MOTOR OFF IS RECEIVED")


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:

        hum=random.randint(0, 100)
        #print(hum)
        temp =random.randint(0, 100)
        soilmoisture =random.randint(0,100)
        vibration =random.randint(0, 100)
        current =random.randint(0, 100)
        #Send Temperature & Humidity to IBM Watson
        data = { 'Temperature' : temp, 'Humidity': hum, 'vibration':vibration, 'current':current,
'soilmoisture':soilmoisture }
        #print (data)
        def myOnPublishCallback():
           print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % hum, "vibration = %s %%"
% vibration, "current = %s %%" % current,"soilmoisture = %s %%" % soilmoisture, "to IBM Watson")

        success = deviceCli.publishEvent("Weather", "json", data, qos=0, on_publish=myOnPublishCallback)
        if(temp<50):
                r =
requests.get('https://www.fast2sms.com/dev/bulk?authorization=RcIXU7uQ3HSYPhaeWLf0JkoGmbsF6w
2rl9KAMtDqg5xTz48dBpn0kS7eNGOyZEdfwDKJQa8CXzLFPqR5&sender_id=FSTSMS&message=Temperat
ure is low...please switch on the motor.&language=english&route=p&numbers=9704207699')
                print(r.status_code)
        if(hum<50):
                r =
requests.get('https://www.fast2sms.com/dev/bulk?authorization=RcIXU7uQ3HSYPhaeWLf0JkoGmbsF6w
2rl9KAMtDqg5xTz48dBpn0kS7eNGOyZEdfwDKJQa8CXzLFPqR5&sender_id=FSTSMS&message=Humidity
is low ....please switch on the motor.&language=english&route=p&numbers=9704207699')
                print(r.status_code)
        if(vibration<50):
                r =
requests.get('https://www.fast2sms.com/dev/bulk?authorization=RcIXU7uQ3HSYPhaeWLf0JkoGmbsF6w
2rl9KAMtDqg5xTz48dBpn0kS7eNGOyZEdfwDKJQa8CXzLFPqR5&sender_id=FSTSMS&message=The motor
vibration is too high...&language=english&route=p&numbers=9704207699')
```

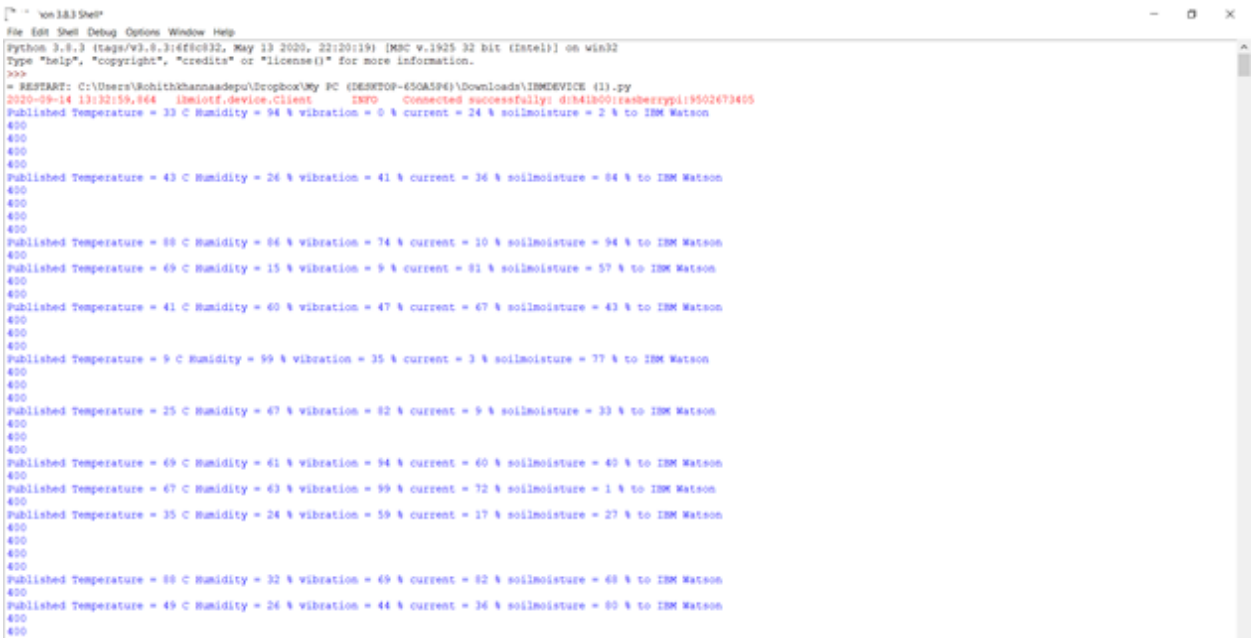
```
            print(r.status_code)
        if(soilmoisture<50):

r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=RcIXU7uQ3HSYPhaeWLf0JkoGmbsF6w2rI9KAMtDqg5xTz48dBpn0kS7eNGOyZEdfwDKJQa8CXzLFPqR5&sender_id=FSTSMS&message=The soilmoisture is too low...please switch on the motor.&language=english&route=p&numbers=9704207699')
            print(r.status_code)
        if(current<50):
            r = requests.get('https://www.fast2sms.com/dev/bulk?authorization=RcIXU7uQ3HSYPhaeWLf0JkoGmbsF6w2rI9KAMtDqg5xTz48dBpn0kS7eNGOyZEdfwDKJQa8CXzLFPqR5&sender_id=FSTSMS&message=The Cylinder is going to be empty.Please book it soon.&language=english&route=p&numbers=9704207699')
            print(r.status_code)
        if not success:
           print("Not connected to IoTF")
        time.sleep(2)

        deviceCli.commandCallback = myCommandCallback

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

OUTPUT:



CONCLUSION

Recent advancements in IoT have drawn attention of researchers and developers worldwide. IoT developers and researchers are working together to extend the technology on large scale and to benefit the society to the highest possible level. However, improvements are possible only if we consider the various issues and shortcomings in the present technical approaches. In this survey article, we presented several issues and challenges that IoT developer must take into account to develop an improved model. Also, important application areas of IoT is also discussed where IoT developers and researchers are engaged. As IoT is not only providing services but also generates a huge amount of data. Hence, the importance of big data analytics is also discussed which can provide accurate decisions that could be utilized to develop an improved IoT system.

FUTURE SCOPE

This paper aims at developing the Smart Irrigation System Using IoT Technology with an objective of automating the total irrigation system which provide adequate water required by crop by monitoring the moisture of soil and climate condition in order to prevent the wastage of water resource.

SENSORS	SENSOR VALUE RANGE
Soil Moisture	-10°c to +85°c
Pressure	±102 mmH_2O
DHT11	Temp -55°c to +150°c Humidity 40%
pH Sensor	6.5 to 7.5

APPENDIX:

System	Moisture Sensor	Temperature Sensor	LDR	Self-irrigation
GSM based Agriculture Monitoring and Controlling System	Yes	No	No	No
Plant Monitoring System	Yes	Yes	Yes	No
Plant Moisture Monitoring System	Yes	No	No	No
Agriculture Monitoring System	Yes	Yes	No	Yes (User must command)
Plant Communicator based on Arduino	Yes	Yes	Yes	Yes