Python code for Water-Level and Light-Intensity

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

#Provide your IBM Watson Device Credentials
organization ="kbfeya"
deviceType = "IOTDEVICE"
deviceId = "1010"
authMethod = "token"
authToken = "07_13*11&83"



# Initialize the device client.
WL=0
LI=0
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])



      if cmd.data['command']=='switchon':
print("SWITCH ON IS RECEIVED")
```

```
elifcmd.data['command']=='switchoff':
print("SWITCH OFF IS RECEIVED")

    if cmd.command == "setInterval":
        if 'interval' not in cmd.data:
print("Error - command is missing required information: 'interval'")
        else:
            interval = cmd.data['interval']
elifcmd.command == "print":
        if 'message' not in cmd.data:
print("Error - command is missing required information: 'message'")
        else:
            print(cmd.data['message'])

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:
        WL=29.58
        LI=35.46
        #Send Water-Level & Light Intensity  to IBM Watson
        data =jsondata={"d":{ 'waterlevel': WL, 'lightintensity': LI}}
        print (data)
        def myOnPublishCallback():
           print ("Published Water Level = %s %%" % WL, "Light Instensity = %s %%" % LI, "to IBM Watson")

        success = deviceCli.publishEvent("Data", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
print("Not connected to IoTF")
time.sleep(1)

deviceCli.commandCallback = myCommandCallback

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

Python Program Output:

```
Published Water Level = 25.56 % Light Instensity = 34.78 % to IBM Watson
{'d': {'waterlevel': 25.56, 'lightintensity': 34.78}}
Published Water Level = 25.56 % Light Instensity = 34.78 % to IBM Watson
{'d': {'waterlevel': 25.56, 'lightintensity': 34.78}}
Published Water Level = 25.56 % Light Instensity = 34.78 % to IBM Watson
{'d': {'waterlevel': 25.56, 'lightintensity': 34.78}}
Published Water Level = 25.56 % Light Instensity = 34.78 % to IBM Watson
Command received: switchon
SWITCH ON IS RECEIVED
```

{'d': {'waterlevel': 25.56, 'lightintensity': 34.78}}

Published Water Level = 25.56 % Light Instensity = 34.78 % to IBM Watson

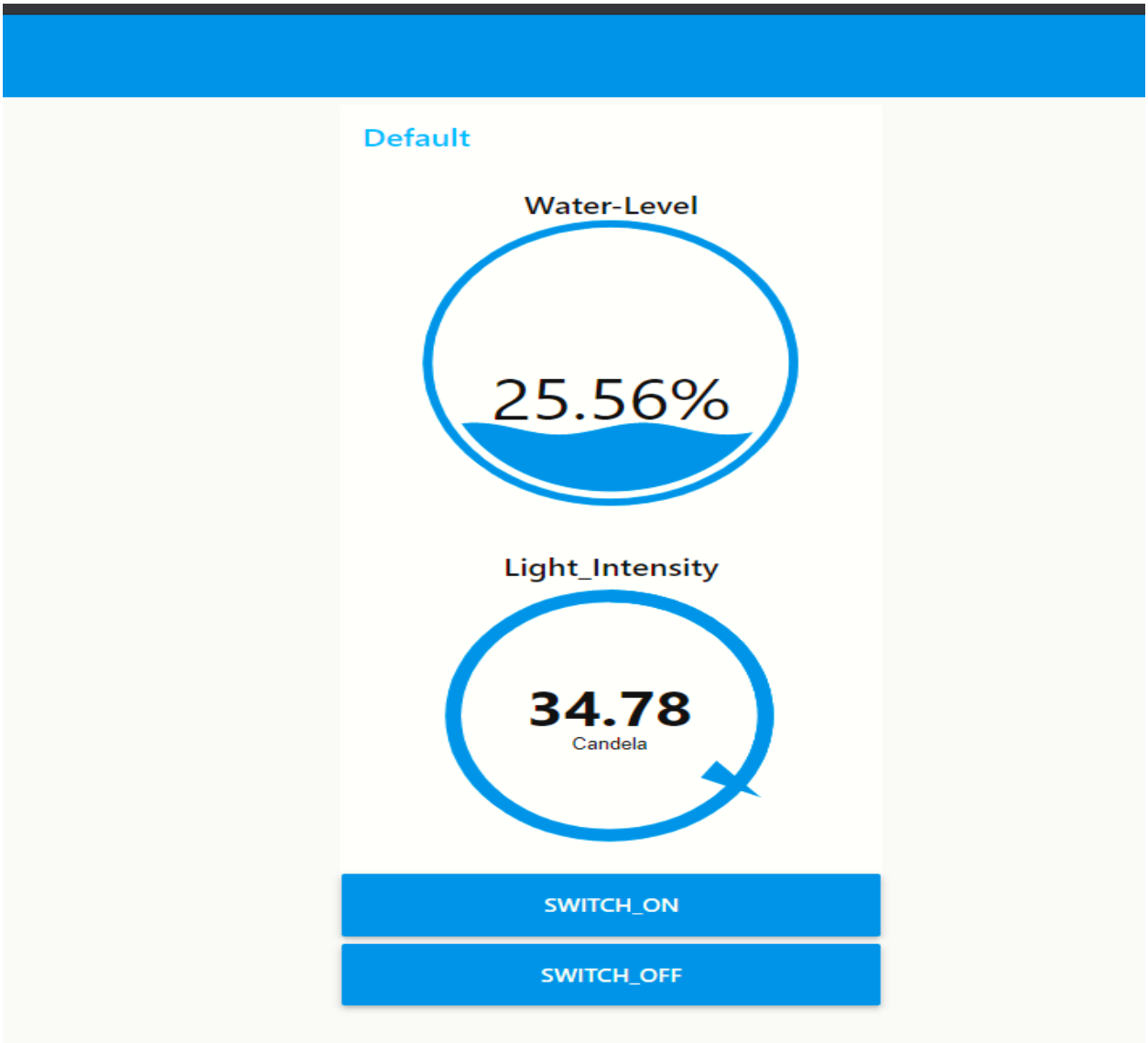
{'d': {'waterlevel': 25.56, 'lightintensity': 34.78}}

Published Water Level = 25.56 % Light Instensity = 34.78 % to IBM Watson

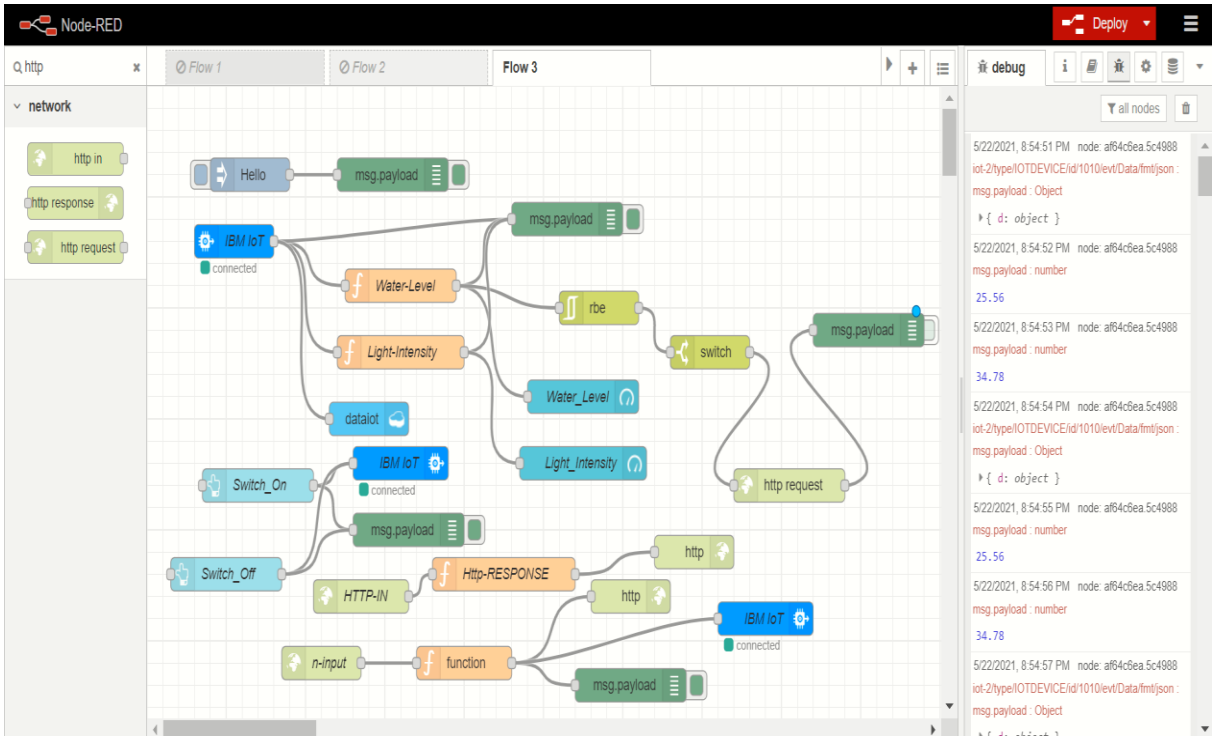
Command received: switchon

SWITCH ON IS RECEIVED

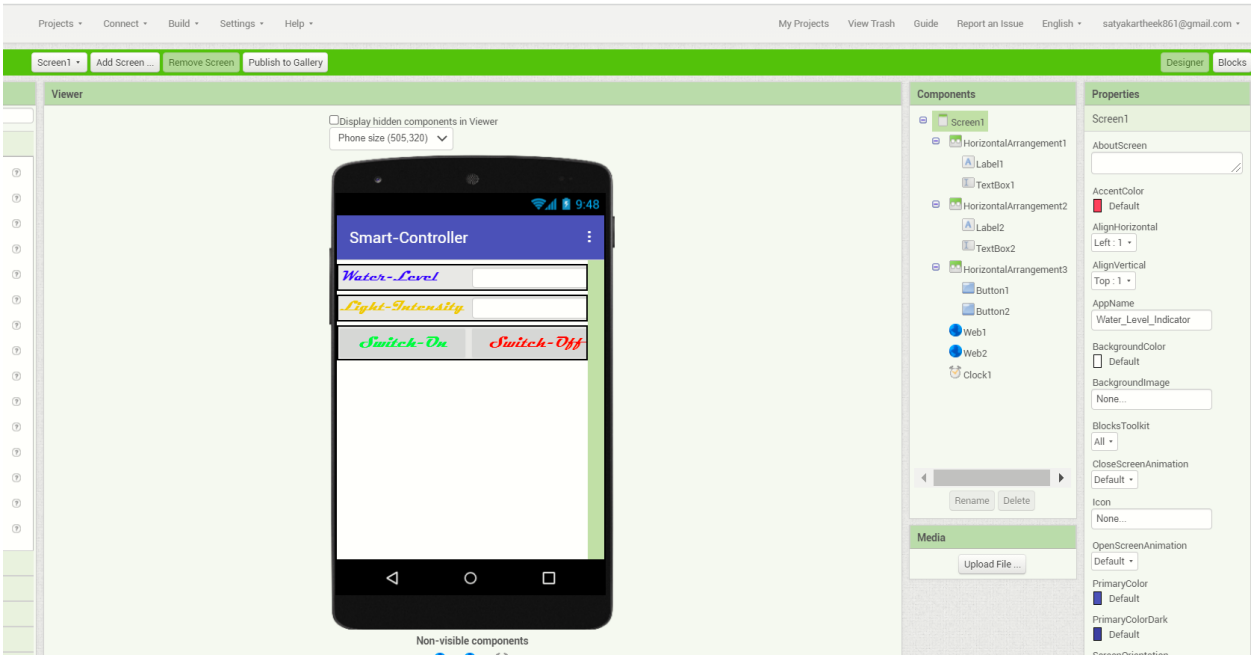
User Interface Image :



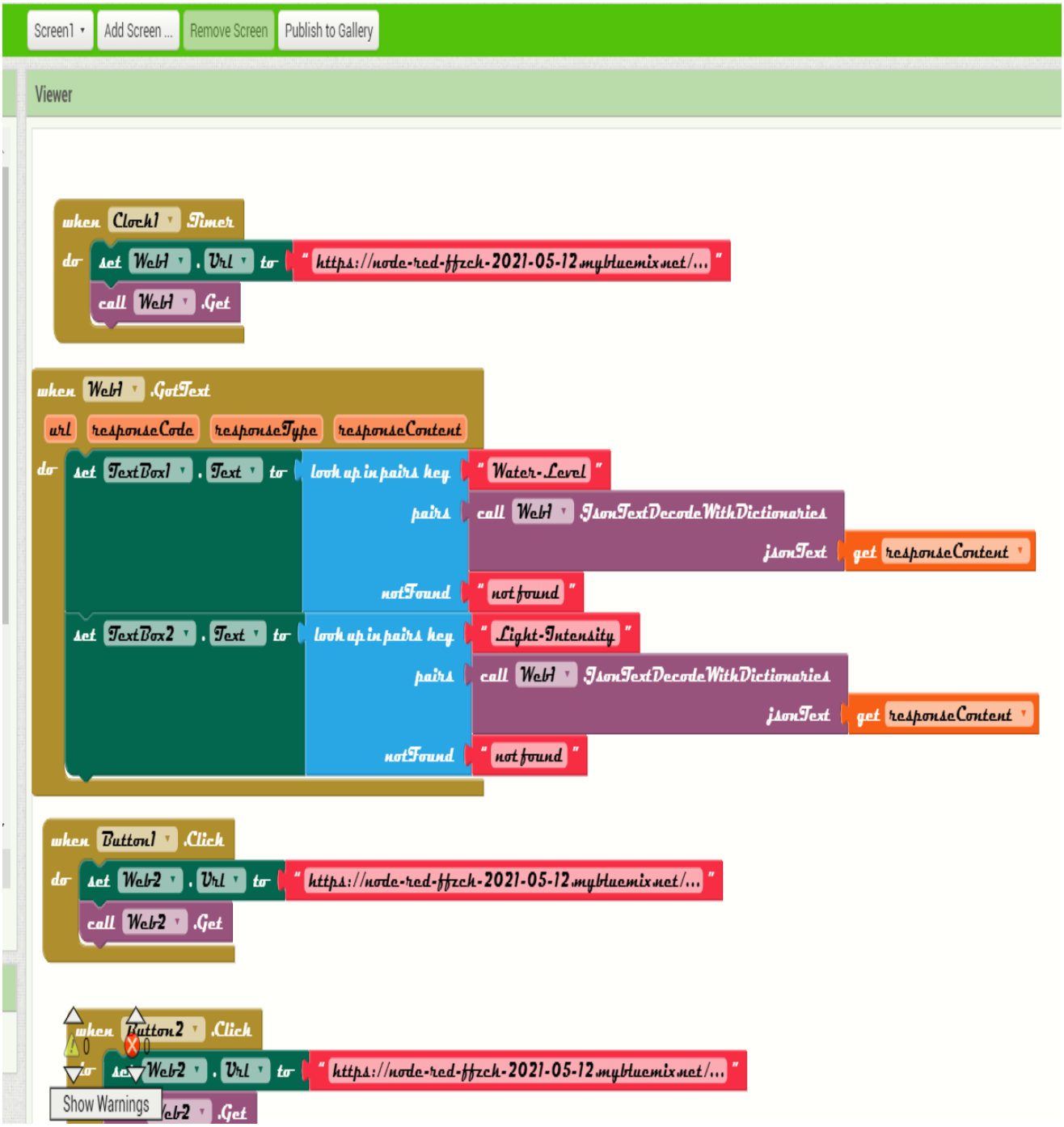
Node-Red Connections:



MIT DESIGN:



MIT BLOCKS


Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Viewer
when Clock1 .Timer
do set Web1 . Url to " https://node-red-ffzch-2021-05-12.mybluemix.net/... "
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set TextBox1 . Text to look up in pairs key " Water-Level "
pairs call Web1 JsonTextDecodeWithDictionaries
jsonText get responseContent
notFound " not found "
set TextBox2 . Text to look up in pairs key " Light-Intensity "
pairs call Web1 JsonTextDecodeWithDictionaries
jsonText get responseContent
notFound " not found "
when Button1 .Click
do set Web2 . Url to " https://node-red-ffzch-2021-05-12.mybluemix.net/... "
call Web2 .Get
when Button2 .Click
set Web2 . Url to " https://node-red-ffzch-2021-05-12.mybluemix.net/... "
Show Warnings