PROJECT DEVELOPMENT PHASE

PROJECT DEVELOPMENT DELIVERY OF SPRINT 2

Date	10-11-2022
Team ID	PNT2022TMID31710
Project Name	Real-Time River Water Quality Monitoring and Control System

SPRINT DESCRIPTION:

In this Sprint, we discuss about the complete Python Code Simulation.

CODE EXPLANATION:

The below described code is what we have developed for connecting with IBM IoT cloud. The code also connects with the Node- Red Service and displays the output frequently. Once the Code is simulated, the code runs with the output of temperature, humidity and pH value. This code links with the IBM IoT Platform and then to Node-Red, Finally the result is displayed in our Application.

PYTHON CODE:

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


#Provide your IBM Watson Device
Credentialsorganization = "udjkcs"
deviceType = "1234"
deviceId = "1234567"
authMethod = "token"
authToken =
"123456789"

# Initialize GPIO


def myCommandCallback(cmd):
   print("Command received: %s" %
   cmd.data['command'])status=cmd.data['command']
   if
      status=="lighton
      ":print ("led is
      on")
   elif status ==
      "lightoff":print ("led
      is off")
   else :
      print ("please send proper

   command")#print(cmd)
```

```
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(0,10
      0)pH=random.randint(0,14)

      data = { 'temp' : temp, 'Humid': Humid ,'pH' :
      pH }#print data
      def myOnPublishCallback():
         print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,
         "pHValue =
%s" % pH,  "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
clouddeviceCli.disconnect()
```