SPRINT – 1 DELIVERY

Date	November 9, 2022
Team ID	PNT2022TMID36352
Project Name	Real-Time River Water Quality Monitoring and Control System

PYTHON PROGRAM:-

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

Provide your IBM Watson Device Credentials

```
organization = "dymr4l"  # repalce it with organization ID deviceType
= "NodeMCU"  # replace it with device type
deviceId = "2002"  # repalce with device id
authMethod = "token"
authToken = "Nirmal@2002"  # repalce with token


def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data)
if cmd.data['command'] == 'lighton':
      print("LIGHT ON")              elif
cmd.data['command'] == 'lightoff':
      print("LIGHT OFF")


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()

deviceCli.connect()
```

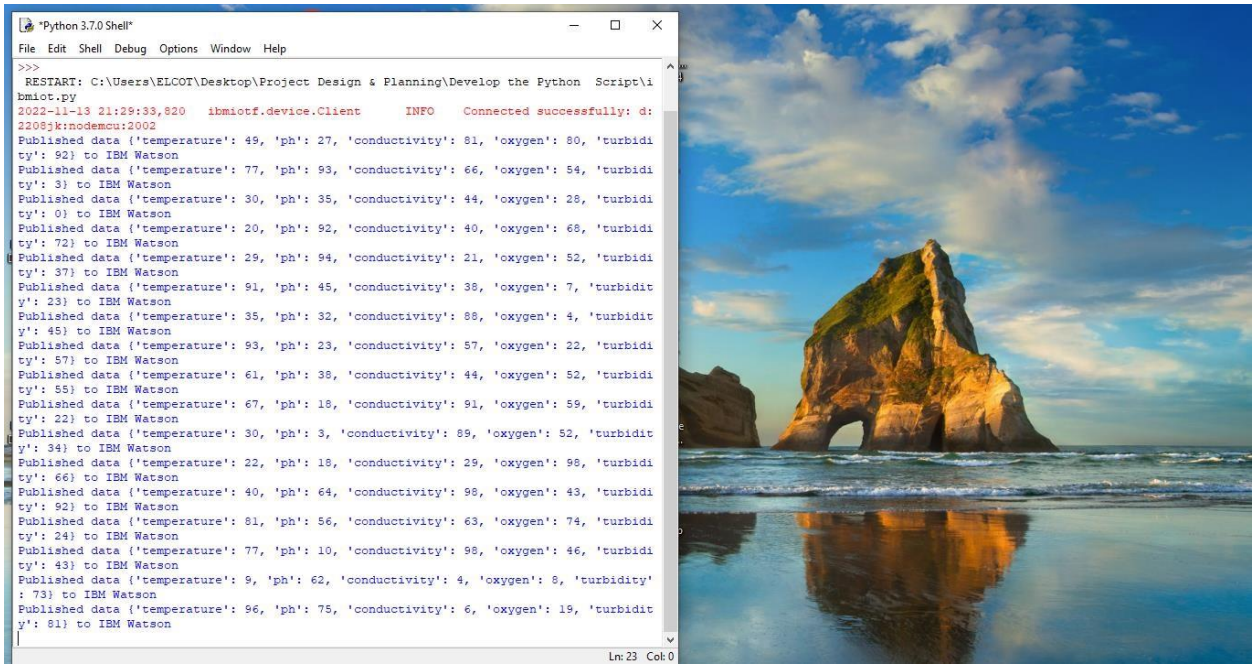
```
while True:     pH =
random.randint(0,100)     conductivity =
random.randint(0,100)     T =
random.randint(0,100)     oxygen =
random.randint(0,100)     turbidity =
random.randint(0,100)
```

 # Send Temperature & Humidity to IBM Watson

```
data = {'temperature': T,'ph':pH,'conductivity':conductivity,'oxygen':oxygen,"turbidity":turbidity}


# print data          def myOnPublishCallback():
print("Published data",data, "to IBM Watson")


success = deviceCli.publishEvent("event", "json", data, 0,
myOnPublishCallback)     if not success:          print("Not connected to IoTF")
time.sleep(5)

deviceCli.commandCallback = myCommandCallback
```

 # Disconnect the device and application from the cloud

OUTPUT:



CODE FOR ARDUINO:

```
#include <OneWire.h>

#include <DallasTemperature.h> #define ONE_WIRE_BUS 5
```

```
OneWire oneWire(ONE_WIRE_BUS); DallasTemperature
sensors(&oneWire); float Celcius=0; float Fahrenheit=0; float
voltage=0; const int analogInPin = A0; int sensorValue = 0;
unsigned long int avgValue; float b; int buf[10],temp; void
setup(void)
{

Serial.begin(9600); sensors.begin(); int sensorValue = analogRead(A1);
voltage = sensorValue * (5.0 /
1024.0);
} void loop(void) { sensors.requestTemperatures();
Celcius=sensors.getTempCByIndex(0);
                                        Fahrenheit=sensors.toFahrenheit(C
elcius); for(int i=0;i<10;i++) { buf[i]=analogRead(analogInPin); delay(10);
} for(int i=0;i<9;i++) { for(int j=i+1;j<10;j++)
{     if(buf[i]>buf[j])     {     temp=buf[i];     buf[i]=buf[j];
      buf[j]=temp;
}

 }      }      for(int      i=2;i<8;i++)      avgValue+=buf[i];      float
pHVol=(float)avgValue*5.0/1024/6; float phValue = -5.70 * pHVol +
21.34;

Serial.println(phValue);
Serial.print("pH");



Serial.print(" C ");
Serial.print(Celcius);


Serial.print(voltage); Serial.print("V"); delay(10000);
```

}

CODE IMPLEMENTATION:

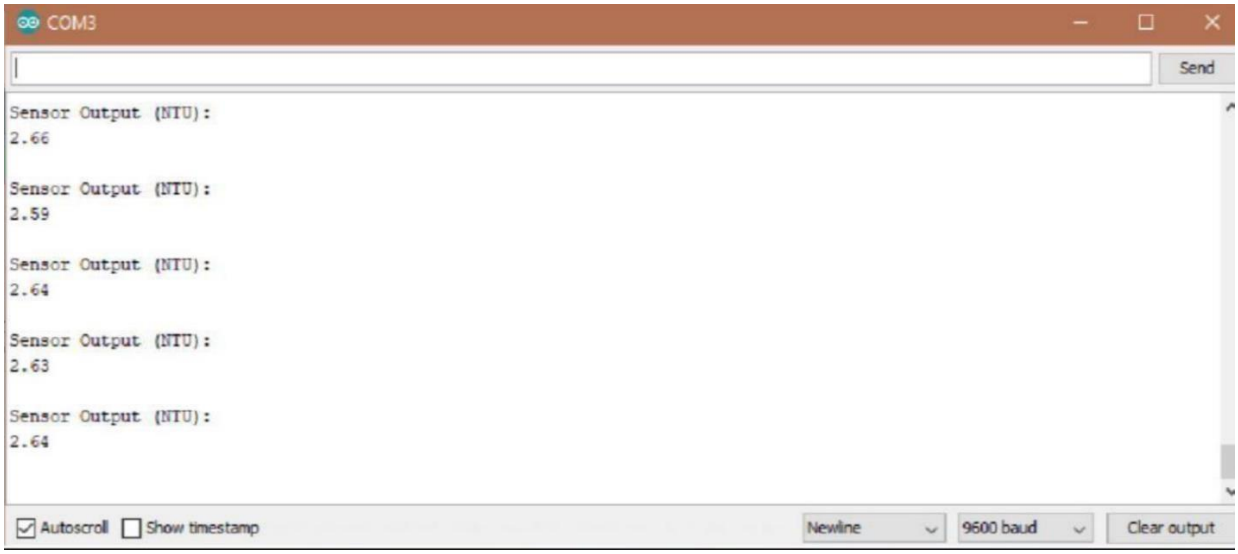
```
import serial import time import csv import numpy as np import matplotlib.pyplot as plt ser = serial.Serial('/COM6',9600) ser_bytes = ser.readline(10) print (ser_bytes) ser.flushInput() while True: try:

 ser_bytes = ser.readline() decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8")) print(decoded_bytes) temp = float(decoded_bytes(1:3)) turb = float(decoded_bytes(4:6)) pH = float(decoded_bytes(6:8)) with open("test_data.csv","a") as f:

writer = csv.writer(f,delimiter=",")

writer.writerow([time.time(),decoded_bytes]) except:
{ if(buf[i]>buf[j]) { temp=buf[i]; buf[i]=buf[j]; print("Keyboard Interrupt") ser.close() break() t = np.arange(0.0, 2.0, buf[j]=temp; 0.01) s = 1 + np.sin(2*np.pi*t) plt.plot(t, s) plt.xlabel('time (s)') plt.ylabel('Celsisus (C)')
```

```
plt.title('Temperature') plt.grid(True) plt.savefig("Temperature.png")
plt.show()
Serial.begin(9600); sensors.begin(); int sensorValue = analogRead(A1);

voltage = sensorValue * (5.0 / 1024.0);

}

void loop(void)

{

sensors.requestTemperatures();

Celcius=sensors.getTempCByIndex(0);

                                        Fahrenheit=sensors.toFahrenheit(C

elcius); for(int i=0;i<10;i++)

{

buf[i]=analogRead(analogInPin); delay(10);

}

for(int i=0;i<9;i++)

{

for(int j=i+1;j<10;j++)

{

if(buf[i]>buf[j])

{

temp=buf[i]; buf[i]=buf[j]; buf[j]=temp;

}

n = 256

X = np.linspace(-np.pi, np.pi, 256, endpoint=True) C,S = np.cos(X),

np.sin(X) plt.plot(X, C) plt.plot(X,S) plt.show() print ("Visualization

of real time sensor Data.") print("/n") while True:
```

```
try:

ser_bytes = ser.readline() decoded_bytes =

    float(ser_bytes[0:len(ser_bytes)2].decode("utf-8")) print(decoded_bytes) temp =

float(decoded_bytes(1:3)) turb = float(decoded_bytes(4:6)) pH = float(decoded_bytes(6:8))

with open("test_data.csv","a") as f: writer = csv.writer(f,delimiter=",")
```

```
np.arange(0.0, 2.0, 0.01) s = 1 + np.sin(2*np.pi*t) plt.plot(t, s)
plt.title('Temperature') plt.grid(True) plt.savefig("Temperature.png")
plt.show()
```

ARDUINO OUTPUT:



```
plt.title('Temperature') plt.grid(True) plt.savefig("Temperature.png")
plt.show()
```

CODE FOR ARDUINO:

```
#include <OneWire.h>

#include <DallasTemperature.h> #define ONE_WIRE_BUS 5

OneWire oneWire(ONE_WIRE_BUS); DallasTemperature

sensors(&oneWire); float Celcius=0; float Fahrenheit=0; float

voltage=0; const int analogInPin = A0; int sensorValue = 0;

unsigned long int avgValue; float b; int buf[10],temp; void

setup(void)

{


Serial.begin(9600); sensors.begin(); int sensorValue = analogRead(A1);

voltage = sensorValue * (5.0 /

1024.0);

} void loop(void) { sensors.requestTemperatures();

Celcius=sensors.getTempCByIndex(0);

                                        Fahrenheit=sensors.toFahrenheit(C

elcius); for(int i=0;i<10;i++) { buf[i]=analogRead(analogInPin); delay(10);

} for(int i=0;i<9;i++) { for(int j=i+1;j<10;j++)
```

```
{ if(buf[i]>buf[j]) { temp=buf[i]; buf[i]=buf[j];
buf[j]=temp;
}

} } for(int i=2;i<8;i++) avgValue+=buf[i]; float
pHVol=(float)avgValue*5.0/1024/6; float phValue = -5.70 * pHVol +
21.34;

Serial.println(phValue);
Serial.print("pH");



Serial.print(" C ");
Serial.print(Celcius);


Serial.print(voltage); Serial.print("V"); delay(10000);
}
```

CODE IMPLEMENTATION:

```
import serial import time import csv import numpy as np import matplotlib.pyplot as plt ser =
serial.Serial('/COM6',9600) ser_bytes = ser.readline(10) print (ser_bytes) ser.flushInput() while
True: try:

ser_bytes = ser.readline() decoded_bytes = float(ser_bytes[0:len(ser_bytes)-
2].decode("utf-8")) print(decoded_bytes) temp =
float(decoded_bytes(1:3)) turb = float(decoded_bytes(4:6)) pH =
float(decoded_bytes(6:8)) with open("test_data.csv","a") as f:
writer = csv.writer(f,delimiter=",")
writer.writerow([time.time(),decoded_bytes]) except:
{ if(buf[i]>buf[j]) { temp=buf[i]; buf[i]=buf[j]; print("Keyboard
Interrupt") ser.close() break() t = np.arangebuf[j]=temp; (0.0, 2.0,
```

```
0.01) s = 1 + np.sin(2*np.pi*t) plt.plot(t, s) plt.xlabel('time (s)')
plt.ylabel('Celsisus (C)')
```

```
plt.title('Temperature') plt.grid(True) plt.savefig("Temperature.png")
plt.show()
Serial.begin(9600); sensors.begin(); int sensorValue = analogRead(A1);

voltage = sensorValue * (5.0 / 1024.0);

}

void loop(void)

{

sensors.requestTemperatures();

Celcius=sensors.getTempCByIndex(0);

                                        Fahrenheit=sensors.toFahrenheit(C

elcius); for(int i=0;i<10;i++)

{

buf[i]=analogRead(analogInPin); delay(10);

}

for(int i=0;i<9;i++)

{

for(int j=i+1;j<10;j++)

{

if(buf[i]>buf[j])

{

temp=buf[i]; buf[i]=buf[j]; buf[j]=temp;

}

n = 256

X = np.linspace(-np.pi, np.pi, 256, endpoint=True) C,S = np.cos(X),

np.sin(X) plt.plot(X, C) plt.plot(X,S) plt.show() print ("Visualization

of real time sensor Data.") print("/n") while True:
```

try:

ser_bytes = ser.readline() decoded_bytes = float(ser_bytes[0:len(ser_bytes)2].decode("utf-8")) print(decoded_bytes) temp = float(decoded_bytes(1:3)) turb = float(decoded_bytes(4:6)) pH = float(decoded_bytes(6:8))

with open("test_data.csv","a") as f: writer = csv.writer(f,delimiter=",")

np.arange(0.0, 2.0, 0.01) s = 1 + np.sin(2*np.pi*t) plt.plot(t, s)

plt.title('Temperature') plt.grid(True) plt.savefig("Temperature.png")
plt.show()

ARDUINO OUTPUT: