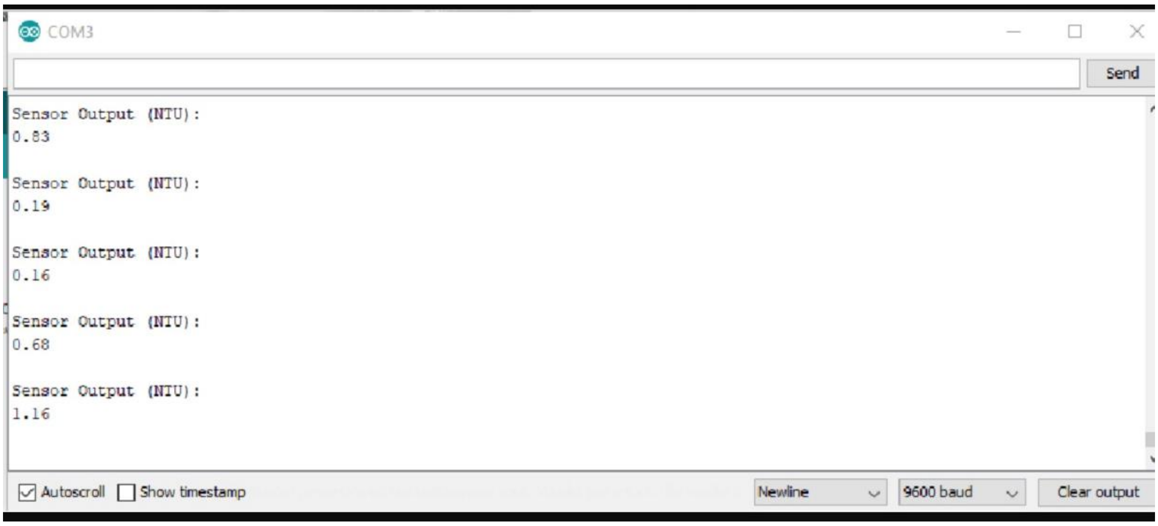
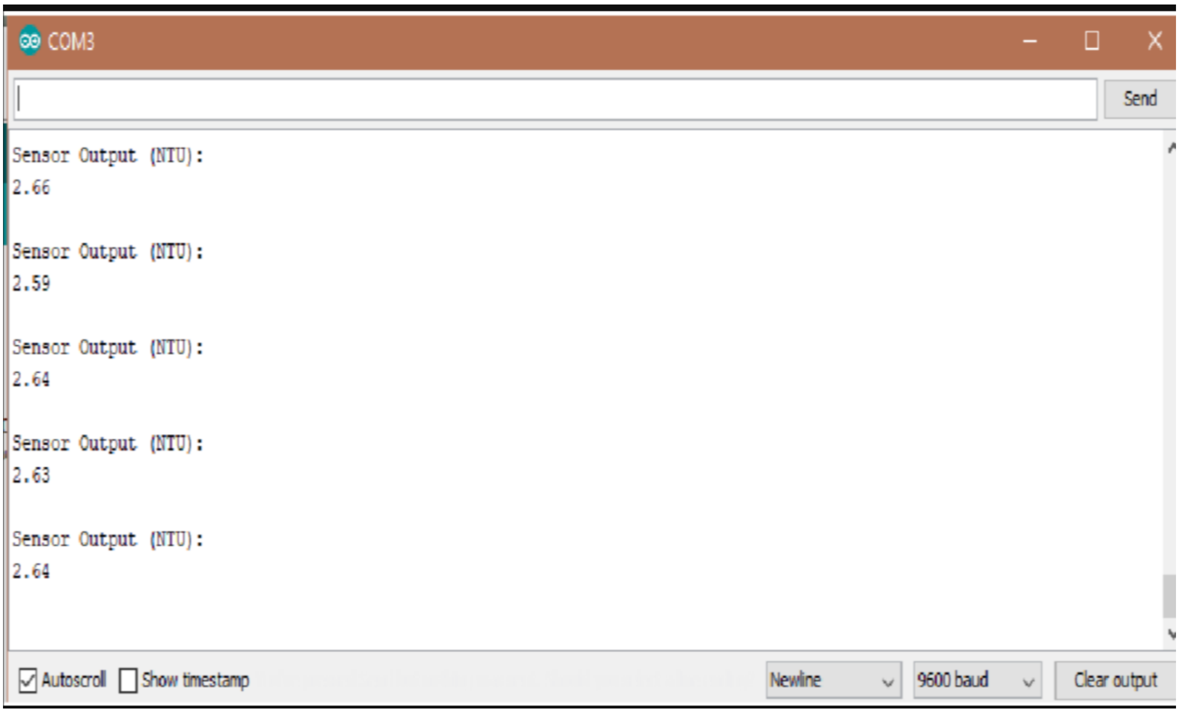
# DEVELOPMENT PHASE - SPRINT 4

| | |
|---|---|
| Date | 10 November 2022 |
| Team ID | PNT2022TMID15074 |
| Project Name | IOT Based Real – time River Water Quality Monitoring and Control System |

## OUTPUT TESTCASES:

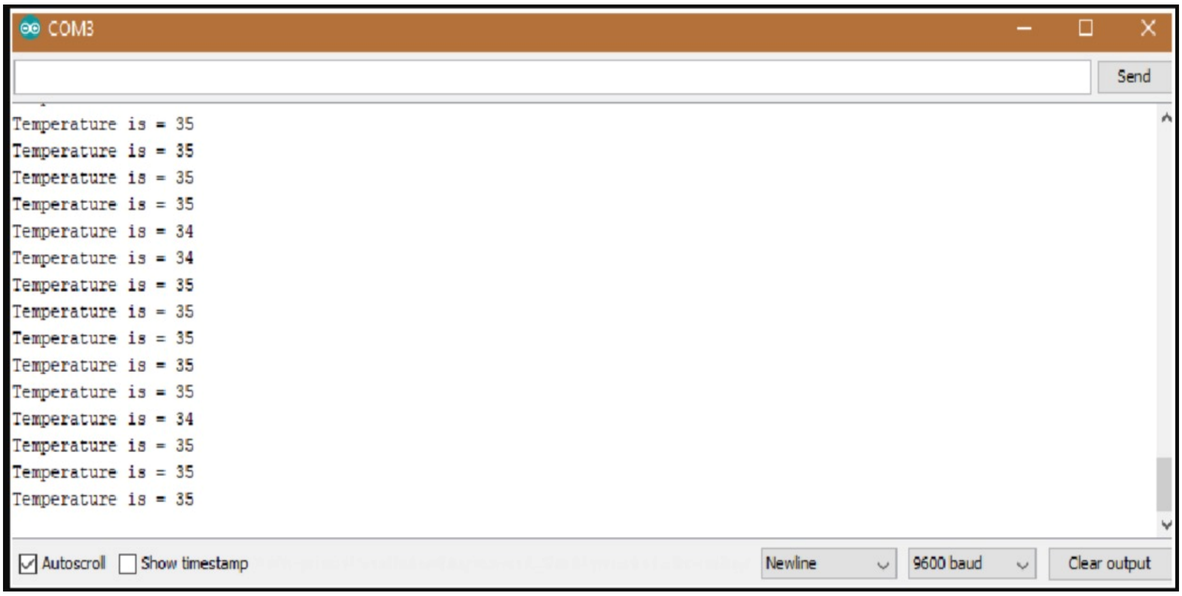


```
Sensor Output (NTU):
0.83

Sensor Output (NTU):
0.19

Sensor Output (NTU):
0.16

Sensor Output (NTU):
0.68

Sensor Output (NTU):
1.16
```



```
Sensor Output (NTU):
2.66

Sensor Output (NTU):
2.59

Sensor Output (NTU):
2.64

Sensor Output (NTU):
2.63

Sensor Output (NTU):
2.64
```

## OUTPUT:



```
Temperature is = 35
Temperature is = 35
Temperature is = 35
Temperature is = 35
Temperature is = 34
Temperature is = 34
Temperature is = 35
Temperature is = 35
Temperature is = 35
Temperature is = 35
Temperature is = 35
Temperature is = 34
Temperature is = 35
Temperature is = 35
Temperature is = 35
```

## 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(Celcius);
 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:
 print("Keyboard Interrupt")
 ser.close()
 break()
t = np.arange(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(Celcius);
 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=",")
 writer.writerow([time.time(),decoded_bytes])
 except:
 print("Keyboard Interrupt")
 ser.close()
 break()
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2*np.pi*t)
plt.plot(t, s)
```