DEVELOPMENTPHASE-SPRINT4

Date	11 November 2022
TeamID	PNT2022TMID25057
ProjectName	IOTBasedReal –time River Water Quality Monitoring and ControlSystem

OUTPUTTESTCASES:

OUTPUT:
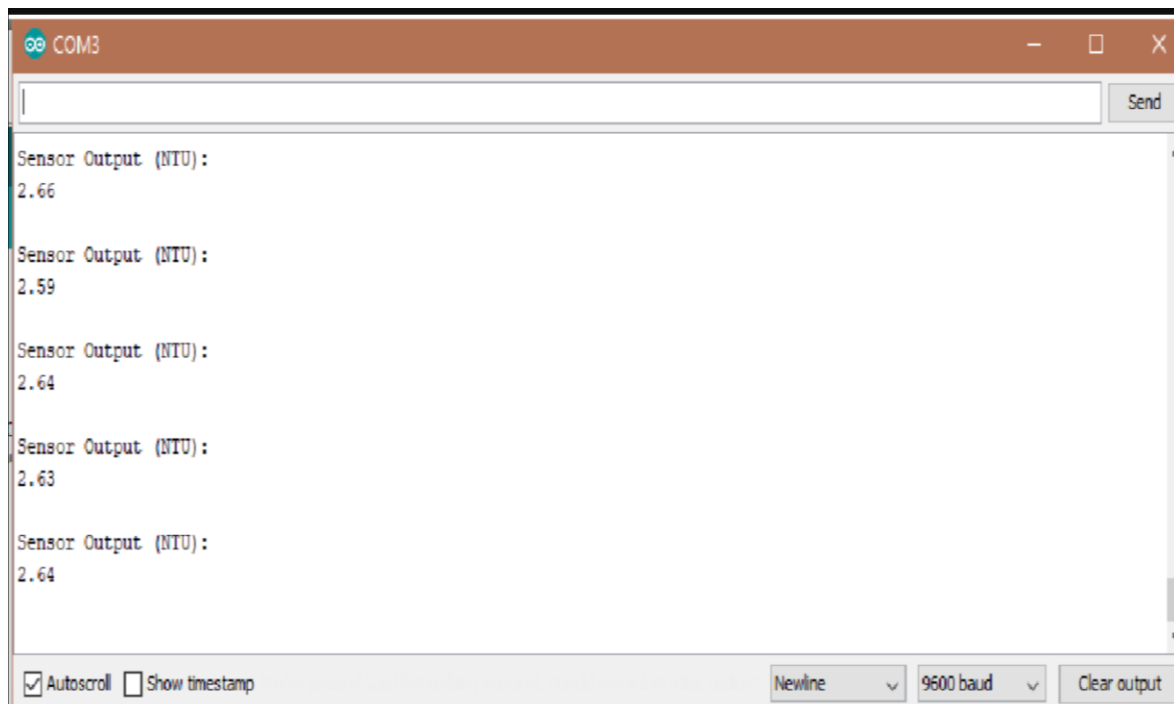

```
COM3                                                                    —    □    ×

                                                                             Send

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

☑ Autoscroll  ☐ Show timestamp            Newline  ∨  9600 baud  ∨  Clear output
```
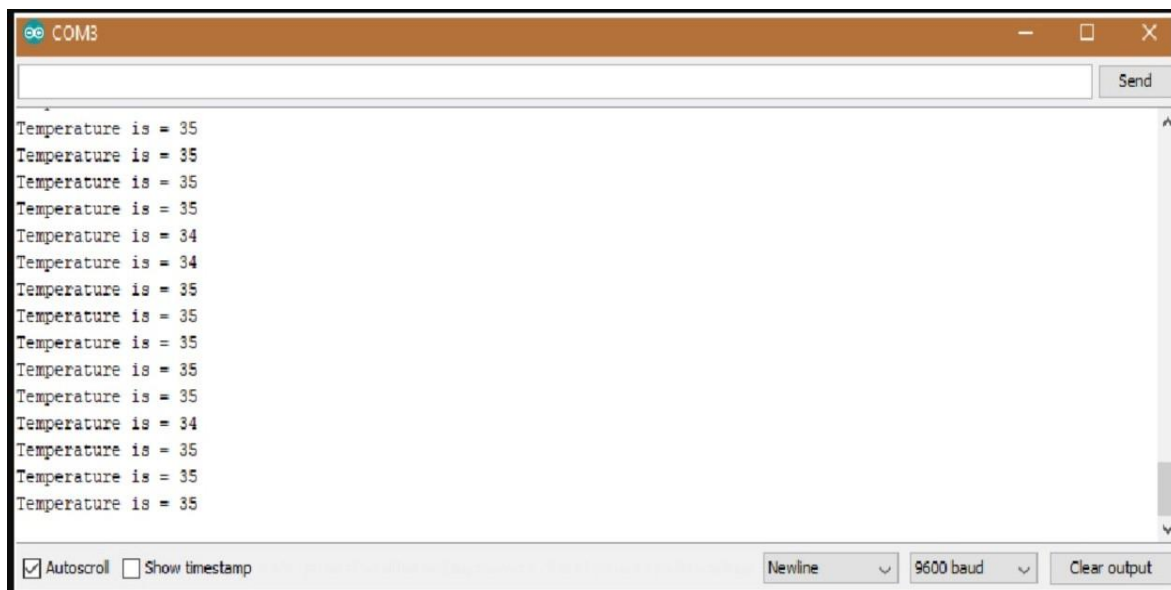
CODEFORARDUINO:

```
#include
<OneWire.h>
          #include<DallasTemperature.h>
          #defineONE_WIRE_BUS5
          OneWireoneWire(ONE_WIRE_BUS);
          DallasTemperaturesensors(&oneWire);
          floatCelcius=0;
          floatFahrenheit=0;
          floatvoltage=0;
          constintanalogInPin= A0;
          intsensorValue=0;
          unsignedlongintavgValue;
          floatb;
          intbuf[10],temp;
          voidsetup(void)
          {

           Serial.begin(9600);
           sensors.begin();
           intsensorValue=analogRead(A1);
           voltage=sensorValue*(5.0/1024.0);
          }
          voidloop(void)
          {
           sensors.requestTemperatures();
           Celcius=sensors.getTempCByIndex(0);
```

```
              Fahrenheit=sensors.toFahrenheit(Celcius);
              for(inti=0;i<10;i++)
             {
              buf[i]=analogRead(analogInPin);
              delay(10);
             }
             for(inti=0;i<9;i++)
             {
              for(intj=i+1;j<10;j++)
              {
              if(buf[i]>buf[j])
              {
              temp=buf[i];
              buf[i]=buf[j];
              buf[j]=temp;
              }
               }
             }
             for(inti=2;i<8;i++)
             avgValue+=buf[i];
             floatpHVol=(float)avgValue*5.0/1024/6;
             floatphValue=-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);
             }
```

CODEIMPLEMENTATION:

```
  import
  serial
          importtime
          importcsv
          importnumpyasnp
          importmatplotlib.pyplotasplt
          ser=serial.Serial('/COM6',9600)
          ser_bytes=ser.readline(10)
          print(ser_bytes)
```

```python
ser.flushInput()
whileTrue:
 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))
 withopen("test_data.csv","a")asf:
 writer=csv.writer(f,delimiter=",")
 writer.writerow([time.time(),decoded_bytes])
 except:
 print("KeyboardInterrupt")
 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();
 intsensorValue=analogRead(A1);
 voltage=sensorValue*(5.0/1024.0);
}
voidloop(void)
{
 sensors.requestTemperatures();
 Celcius=sensors.getTempCByIndex(0);
 Fahrenheit=sensors.toFahrenheit(Celcius);
 for(inti=0;i<10;i++)
{
 buf[i]=analogRead(analogInPin);
 delay(10);
}
for(inti=0;i<9;i++)
{
 for(intj=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("VisualizationofrealtimesensorData.")
print("/n")
whileTrue:
 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))
 withopen("test_data.csv","a")asf:
 writer=csv.writer(f,delimiter=",")
 writer.writerow([time.time(),decoded_bytes])
 except:
 print("KeyboardInterrupt")
 ser.close()
 break()
t=np.arange(0.0,2.0, 0.01)
s=1 +np.sin(2*np.pi*t)
plt.plot(t,s)
```