

Project Development Phase

Project Development Template (Sprint-8)

Date	19 November 2022
Team ID	PNT2022TMID29879
Project Name	Project - Real Time River Water Monitoring and Control system

Python code for 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('Celsius (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)
```