

Sprint-3

```
void send_sms(){
    lcd.clear();
    lcd.setCursor(2,0);//set cursor (column by row) indexing from 0
    lcd.print("ATTENTION!!! ");
    lcd.setCursor(2,1);
    lcd.print("SENDING SMS ");
    String temp;
    String lev;
    String phm;
    String turb;
    int turbidityValue = analogRead(A1);
    float turbidityV = turbidityValue/100;

    int buf[10];                //buffer for read analog
    for(int i=0;i<10;i++)        //Get 10 sample value from the sensor for
smooth the value
    {
        buf[i]=analogRead(SensorPin);
        delay(10);
    }
    for(int i=0;i<9;i++)        //sort the analog from small to large
    {
        for(int j=i+1;j<10;j++)
        {
            if(buf[i]>buf[j])
            {
                int temp=buf[i];
                buf[i]=buf[j];
                buf[j]=temp;
            }
        }
    }
    avgValue=0;
    for(int i=2;i<8;i++)        //take the average value of 6
center sample
        avgValue+=buf[i];
    float phValue=(float)avgValue*3.8/1030/6; //convert the analog into
millivolt
    phValue=3.3*phValue+Offset;

    if(sensors.getTempCByIndex(0) > 40){
        temp = String("HIGH");
    }
    if(sensors.getTempCByIndex(0) >= 10 && sensors.getTempCByIndex(0) <=
40){
        temp = String("NORMAL");
    }
    if(sensors.getTempCByIndex(0) < 10){
        temp = String("LOW");
    }

    if(sonar.ping_result / US_ROUNDTRIP_CM > 8){
        lev = String("LOW");
    }
}
```

```

    }
    if(sonar.ping_result / US_ROUNDTRIP_CM >= 5 && sonar.ping_result /
US_ROUNDTRIP_CM <= 8){
        lev = String("NORMAL");
    }
    if(sonar.ping_result / US_ROUNDTRIP_CM < 5){
        lev = String("HIGH");
    }

    if(phValue >= 7.30){
        phm = String("ALKALINE");
    }
    if(phValue >= 6.90 && phValue <= 7.19){
        phm = String("NORMAL");
    }
    if(phValue < 6.89){
        phm = String("ACIDIC");
    }

    if(turbidityV >= 6 && turbidityValue/100 <= 9){
        turb = String("CLEAN");
    }
    if(turbidityV < 6){
        turb = String("DIRTY");
    }

    mySerial.begin(19200); //Default serial port setting for the GPRS
modem is 19200bps 8-N-1
    mySerial.print("\r");
    digitalWrite(buzzer, LOW);
    digitalWrite(blueled, LOW);
    digitalWrite(greenled, LOW);
    digitalWrite(redled, LOW);
    delay(1000); //wait for a second while the modem
sends an "OK"
    mySerial.print("AT+CMGF=1\r"); //Because we want to send the SMS in
text mode
    delay(1000);
    mySerial.print("AT+CMGS=\"+233540518223\" \r"); //Start accepting the
text for the message
    delay(1000);
    mySerial.print(temp);
    mySerial.print(" \r");
    mySerial.print("WATER TEMPERATURE= \r"); //The text for the message
    mySerial.print(sensors.getTempCByIndex(0));
    mySerial.print("*C\r");
    mySerial.println("\r");
    mySerial.print(lev);
    mySerial.print(" \r");
    mySerial.print("WATER LEVEL= \r"); //The text for the message
    mySerial.print(sonar.ping_result / US_ROUNDTRIP_CM);
    mySerial.print("cm\r");
    mySerial.println("\r");
    mySerial.print(phm);

```

```

mySerial.print(" \r");
mySerial.print("WATER PH VALUE= \r");    //The text for the message
mySerial.print(phValue);
mySerial.println("\r");
mySerial.print(turb);
mySerial.print(" \r");
mySerial.print("WATER TURBIDITY= \r");    //The text for the message
mySerial.print(turbidityV);
mySerial.print("NBT\r");

digitalWrite(greenled, HIGH);
digitalWrite(tempgreenled, HIGH);
digitalWrite(levgreenled, HIGH);
digitalWrite(turbgreenled, HIGH);
delay(3000);
/*lcd.clear();
lcd.setCursor(5,0);//set cursor (column by row) indexing from 0
lcd.print("SMS SENT ");
lcd.setCursor(2,1);
lcd.print("SUCCESSFULLY ");*/
mySerial.write(0x1A); //Equivalent to sending Ctrl+Z
}

void send_sms1(){
  lcd.clear();
  lcd.setCursor(2,0);//set cursor (column by row) indexing from 0
  lcd.print("ATTENTION!!! ");
  lcd.setCursor(2,1);
  lcd.print("SENDING SMS ");
  String temp;
  String lev;
  String phm;
  String turb;
  int turbidityValue = analogRead(A1);
  float turbidityV = turbidityValue/100;

  int buf[10];                //buffer for read analog
  for(int i=0;i<10;i++)        //Get 10 sample value from the sensor for
smooth the value
  {
    buf[i]=analogRead(SensorPin);
    delay(10);
  }
  for(int i=0;i<9;i++)          //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        int temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }
}

```

```

    }
    avgValue=0;
    for(int i=2;i<8;i++) //take the average value of 6
center sample
        avgValue+=buf[i];
    float pHValue=(float)avgValue*3.8/1030/6; //convert the analog into
millivolt
    pHValue=3.3*PValue+Offset;

    if(sensors.getTempCByIndex(0) > 40){
        temp = String("HIGH");
    }
    if(sensors.getTempCByIndex(0) >= 10 && sensors.getTempCByIndex(0) <=
40){
        temp = String("NORMAL");
    }
    if(sensors.getTempCByIndex(0) < 10){
        temp = String("LOW");
    }

    if(sonar.ping_result / US_ROUNDTRIP_CM > 8){
        lev = String("LOW");
    }
    if(sonar.ping_result / US_ROUNDTRIP_CM >= 5 && sonar.ping_result /
US_ROUNDTRIP_CM <= 8){
        lev = String("NORMAL");
    }
    if(sonar.ping_result / US_ROUNDTRIP_CM < 5){
        lev = String("HIGH");
    }

    if(PValue >= 7.30){
        phm = String("ALKALINE");
    }
    if(PValue >= 6.90 && PValue <= 7.19){
        phm = String("NORMAL");
    }
    if(PValue < 6.89){
        phm = String("ACIDIC");
    }

    if(turbidityV >= 6 && turbidityValue/100 <= 9){
        turb = String("CLEAN");
    }
    if(turbidityV < 6){
        turb = String("DIRTY");
    }

    mySerial.begin(19200); //Default serial port setting for the GPRS
modem is 19200bps 8-N-1
    mySerial.print("\r");
    digitalWrite(buzzer, LOW);
    digitalWrite(blueled, LOW);
    digitalWrite(greenled, LOW);

```

```

    digitalWrite(redled, LOW);
    delay(1000); //wait for a second while the modem
sends an "OK"
    mySerial.print("AT+CMGF=1\r"); //Because we want to send the SMS in
text mode
    delay(1000);
    mySerial.print("AT+CMGS=\"+233265188849\"\r"); //Start accepting the
text for the message
    delay(1000);
    mySerial.print(temp);
    mySerial.print(" \r");
    mySerial.print("WATER TEMPERATURE= \r"); //The text for the message
    mySerial.print(sensors.getTempCByIndex(0));
    mySerial.print("*C\r");
    mySerial.println("\r");
    mySerial.print(lev);
    mySerial.print(" \r");
    mySerial.print("WATER LEVEL= \r"); //The text for the message
    mySerial.print(sonar.ping_result / US_ROUNDTRIP_CM);
    mySerial.print("cm\r");
    mySerial.println("\r");
    mySerial.print(pHm);
    mySerial.print(" \r");
    mySerial.print("WATER PH VALUE= \r"); //The text for the message
    mySerial.print(pHValue);
    mySerial.println("\r");
    mySerial.print(turb);
    mySerial.print(" \r");
    mySerial.print("WATER TURBIDITY= \r"); //The text for the message
    mySerial.print(turbidityV);
    mySerial.print("NBT\r");

```

```

    digitalWrite(greenled, HIGH);
    digitalWrite(tempgreenled, HIGH);
    digitalWrite(levgreenled, HIGH);
    digitalWrite(turbgreenled, HIGH);
    delay(3000);
    lcd.clear();
    lcd.setCursor(5,0); //set cursor (column by row) indexing from 0
    lcd.print("SMS SENT ");
    lcd.setCursor(2,1);
    lcd.print("SUCCESSFULLY ");
    mySerial.write(0x1A); //Equivalent to sending Ctrl+Z
}

```

```

void PHblink() {
    digitalWrite(blueled, HIGH);
    delay(1000);
    digitalWrite(blueled, LOW);
    delay(1000);
    digitalWrite(blueled, HIGH);
    delay(1000);
    digitalWrite(blueled, LOW);
    delay(1000);
}

```

```

    digitalWrite(blueled, HIGH);
    delay(1000);
    digitalWrite(blueled, LOW);
    delay(1000);
    digitalWrite(blueled, HIGH);
    delay(1000);
    digitalWrite(blueled, LOW);
    delay(1000);
}

void TEMPblink() {
    digitalWrite(tempblueled, HIGH);
    delay(1000);
    digitalWrite(tempblueled, LOW);
    delay(1000);
    digitalWrite(tempblueled, HIGH);
    delay(1000);
    digitalWrite(tempblueled, LOW);
    delay(1000);
    digitalWrite(tempblueled, HIGH);
    delay(1000);
    digitalWrite(tempblueled, LOW);
    delay(1000);
    digitalWrite(tempblueled, HIGH);
    delay(1000);
    digitalWrite(tempblueled, LOW);
    delay(1000);
}

void LEVblink() {
    digitalWrite(levblueled, HIGH);
    delay(1000);
    digitalWrite(levblueled, LOW);
    delay(1000);
    digitalWrite(levblueled, HIGH);
    delay(1000);
    digitalWrite(levblueled, LOW);
    delay(1000);
    digitalWrite(levblueled, HIGH);
    delay(1000);
    digitalWrite(levblueled, LOW);
    delay(1000);
    digitalWrite(levblueled, HIGH);
    delay(1000);
    digitalWrite(levblueled, LOW);
    delay(1000);
}

void TURBblink() {
    digitalWrite(turbblueled, HIGH);
    delay(1000);
    digitalWrite(turbblueled, LOW);
    delay(1000);
    digitalWrite(turbblueled, HIGH);

```

```
delay(1000);  
digitalWrite(turbblueled, LOW);  
delay(1000);  
digitalWrite(turbblueled, HIGH);  
delay(1000);  
digitalWrite(turbblueled, LOW);  
delay(1000);  
    digitalWrite(turbblueled, HIGH);  
delay(1000);  
digitalWrite(turbblueled, LOW);  
delay(1000);  
}
```