Design and Development of Sensor system for monitoring of the Quality of Recycled Water

Annapareddy Tarun

EE15B008

DEPARTMENT OF Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Report of Creative engineering Project
ID4100

Abstract

Water is essential to human life and the health of the environment. Water quality is important not only to protect public health, water provides ecosystem habitats, is used for farming, fishing and mining, and contributes to recreation and tourism. Water quality issues influence human and environmental health, so the more we monitor our water the better we will be able to recognize and prevent contamination problems.

Monitoring your water quality by having it tested regularly is an important part of maintaining a safe and reliable source. Testing the water allows a knowledgeable approach to address the specific problems of a water supply. Monitoring helps ensure that the water source is being properly protected from potential contamination, and that an appropriate treatment system is selected and is operating properly. This will assist you in making informed decisions about your water and how you use it, ensure you are using water suitable for your intended agricultural use, ensure that your drinking water is safe, help determine the effectiveness of your water treatment system. A design is proposed to address such issues. The output specifies the quality of water and turbidity. A prototype of the design has been developed and tested in the laboratory. Test results validate the efficacy of the technique presented.

Such a system can be used at both industrial level as well as household level. Ensuring the user is aware of the type of water by its turbidity is the aim of this project. Since it is cost effective and easy to use such a model can be applied to a larger extent.

Problem Statement:

Design and Development of a Water Quality Monitoring System that could provide reliable information to the user.

Principle:

WATER QUALITY IS ANALYSED BASED ON TURBIDITY AS PARAMETER.
The project deals with design and development of a system for monitoring water quality. Its operation is based on the principle that the intensity of the light scattered by the suspended matter is proportional to its concentration.

THEORY OF SCATTERING

A directed beam of light remains relatively undisturbed when transmitted through absolutely pure water, but even the molecules in a pure fluid will scatter light to a certain degree. Therefore, no solution will have a zero turbidity. In samples containing suspended solids, the manner in which the sample interferes with light transmittance is related to the size, shape and composition of the particles in the solution and to the wavelength (color) of the incident light.

Effect of size of the particle and wavelength of the incident light on scattering:

a. Large particles scatter long wavelengths of light more effectively than they scatter short wavelengths.
b. Small particles scatter short wavelengths of light more effectively than large particles but have less effect on the scatter of longer wavelengths

METHODS TO MEASURE TURBIDITY

Turbidity is caused by suspended and colloidal particles in water. Particles can be inorganic and organic: clay, silt, mud, silica, rust, calcium carbonate, algae, bacteria, organic material. The particles absorb and scatter light.

In Turbidimetry, light passing through the sample is measured.
In Nephelometry, light scattered by suspended particles in a sample is measured.It is more sensitive for very dilute suspensions.

The analytical methods employ common electric photometers that can measure light intensity. The turbidimeter or nephelometer mainly consists of four parts – light source, optical components (e.g. slit), sample compartment and a photocell for the measurement of light either transmitted through the sample or scattered from the suspended particles in the sample. The photocell detects light and an electronic amplifier measures the light intensities.

Turbidity Units

The standard Turbidity unit is called Nephelometric Turbidity Unit (NTU) because of the use of nephelometric method of measurement.

Typical Turbidity Values for different waters

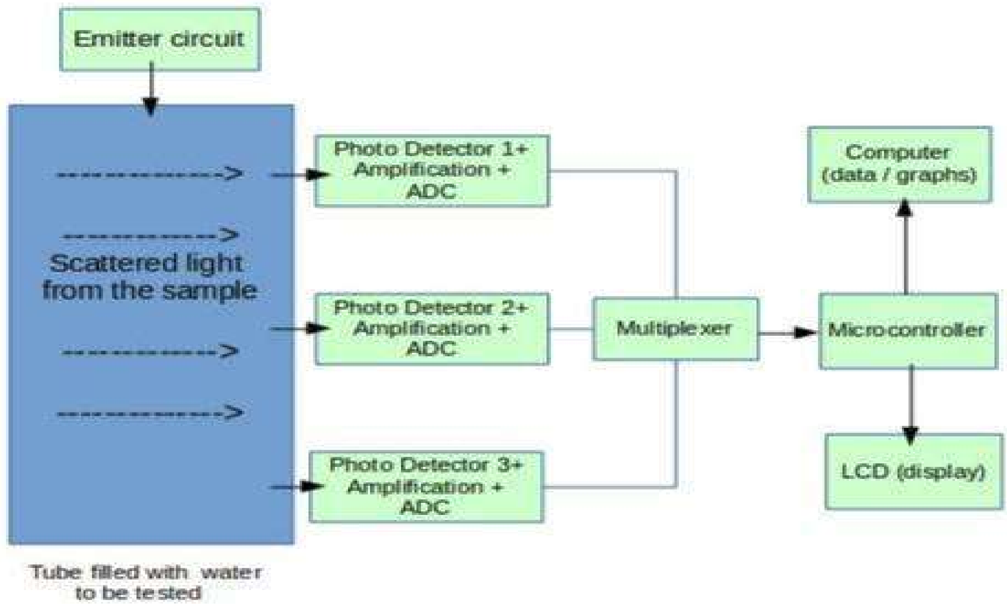
Drinking water quality standards are determined according to World Health Organization guidelines for drinking-water quality as well as other pertinent organizations. These organizations set the standards for drinking water quality parameters and indicate which microbiological, chemical and indicator parameters must be monitored and tested regularly in order to protect the health of the consumers and to make sure the water is wholesome and clean.

Type of water	NTU
Surface Water (Rainy season)	1000 and above
Treated tap water	1 to 2
De-mineralised water	0.1 to 0.5
Water filtered through membrane filter	0.05 or less

Types of water and their turbidity

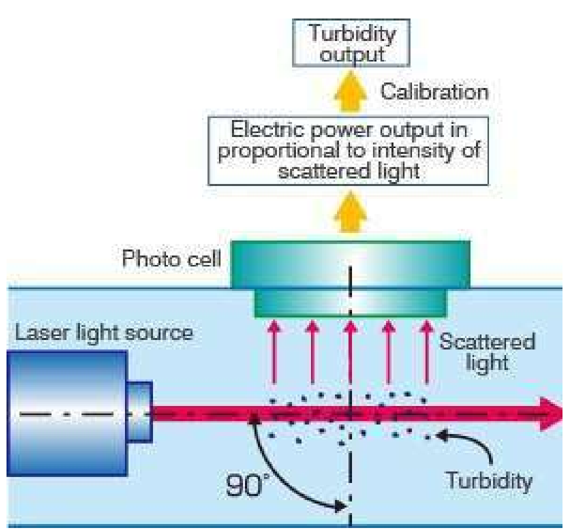
A Turbidity of 1 NTU for drinking water is accepted and above 5 NTU consume

Experimental Setup:



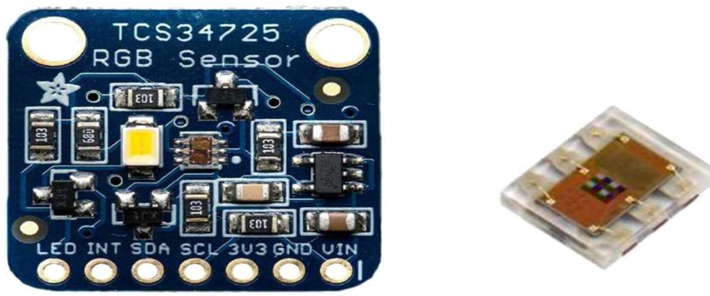
WORKING OF THE MODEL

The light emitted from the laser passed through the sample. The scattered light from the particles inside the sample responsible for the turbidity is then detected by the TCS34725 sensor and corresponding digital output is fed to the Arduino through I2C expander. Thereafter the results can be taken from the LCD or from the computer



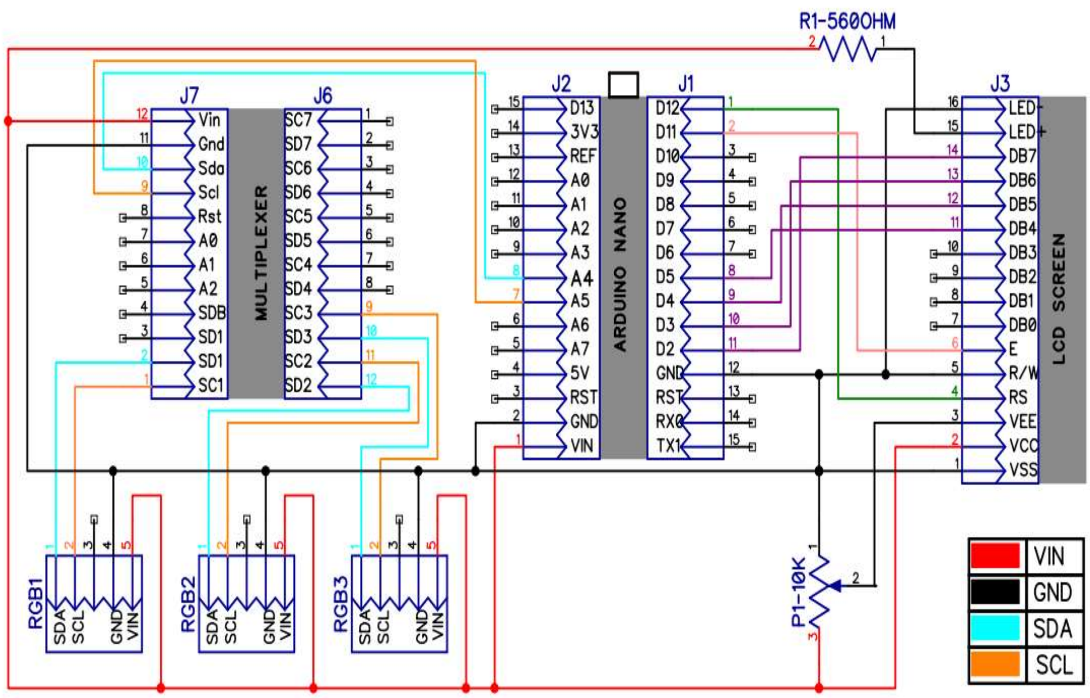
SENSOR MODULE

The device provides a digital return of red, green, blue (RGB), and clear light sensing values. An IR blocking filter, integrated on-chip and localized to the color sensing photodiodes, minimizes the IR spectral component of the incoming light and allows color measurements to be made accurately

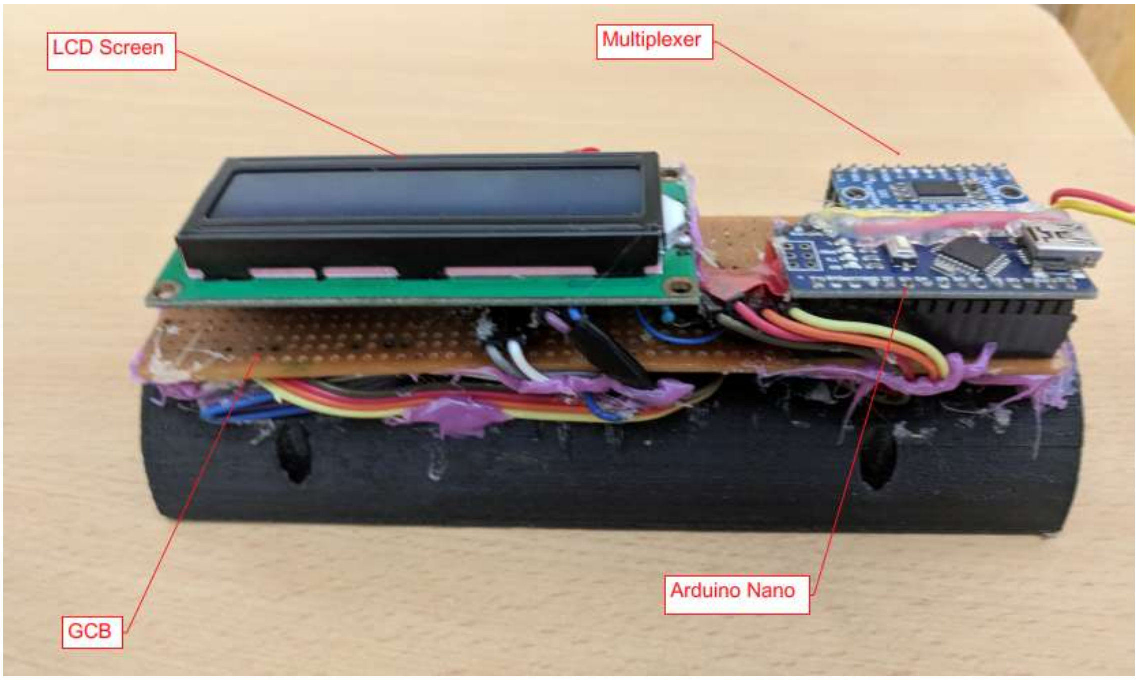


TCS3472

The circuit:

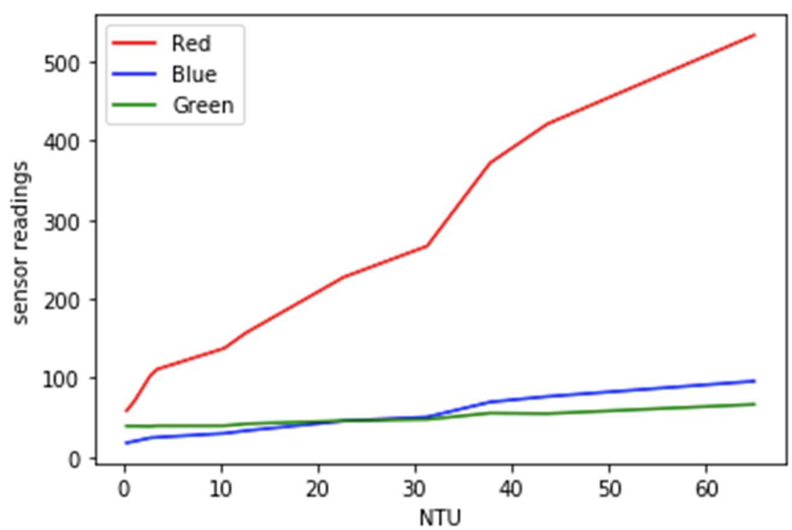


The circuit diagram depicting connections of colour sensors with a multiplexer, Arduino Nano and LCD screen



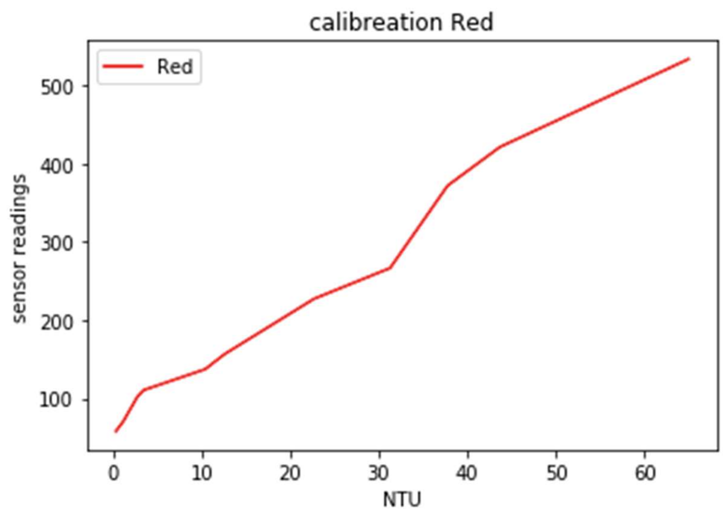
Circuit parts showing all components

Readings and Results



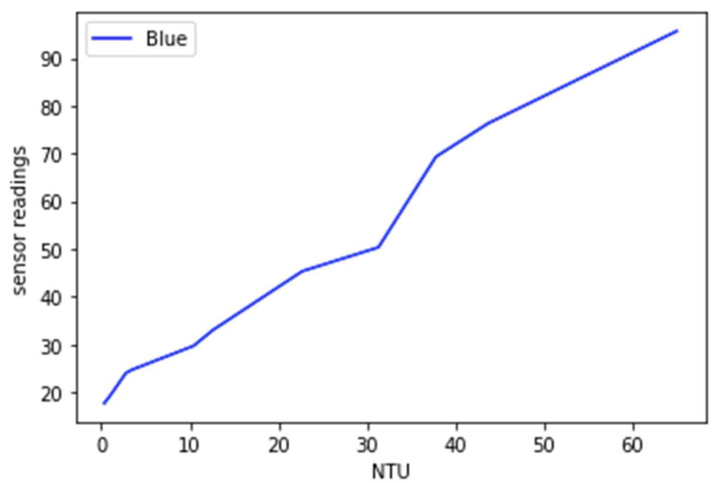
Relation of Turbidity with average of RBG values from all 3 sensors

The above plot shows how average vales of R, G,B are varying with NTU. This gives comparison between each of the colour values .range of R is too good and continuously increasing compared to others can be used for results



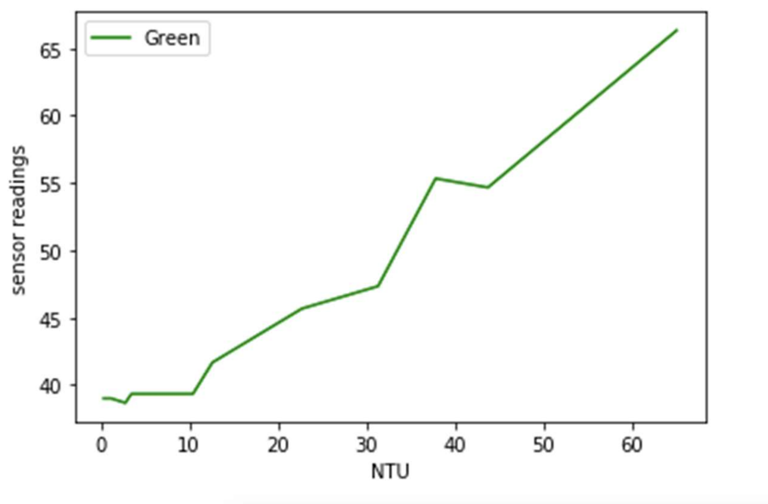
Variation of red color with NTU Values

This plot shows that R vales are continuously increasing with NTU .and plot is almost linear



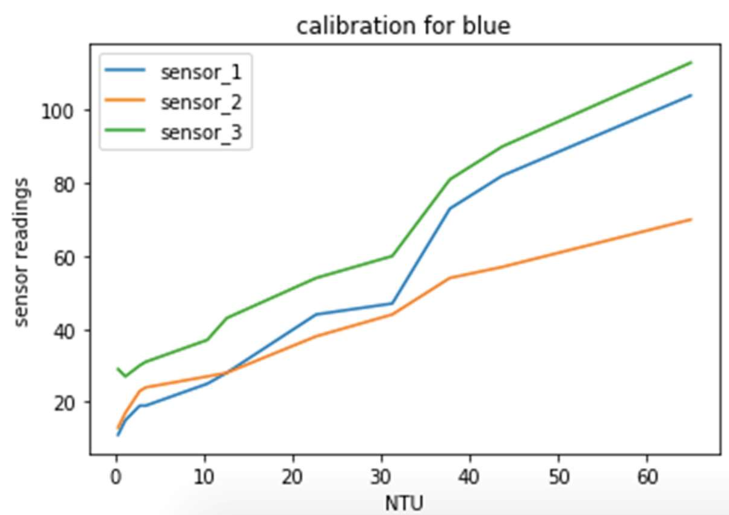
Variation of blue color with NTU Values

This plot describes that B vales are continuously increasing with NTU but is range is too small compared to R

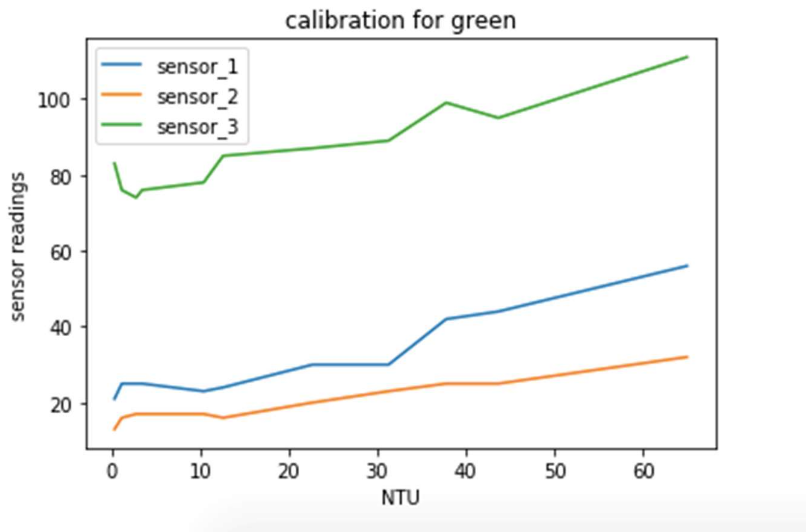


Variation of green color with NTU Values

This plot is not useful as G is not increasing continuously and the range for G is also too small

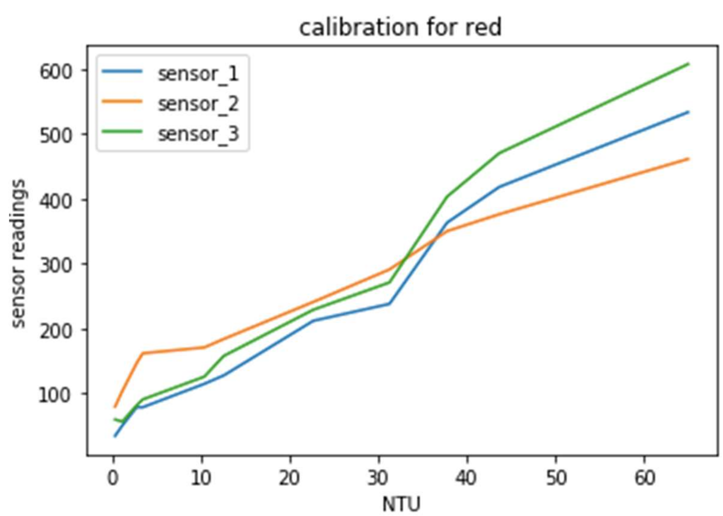


This plot gives how B values of individual sensors are increasing



Calibration curves with green color

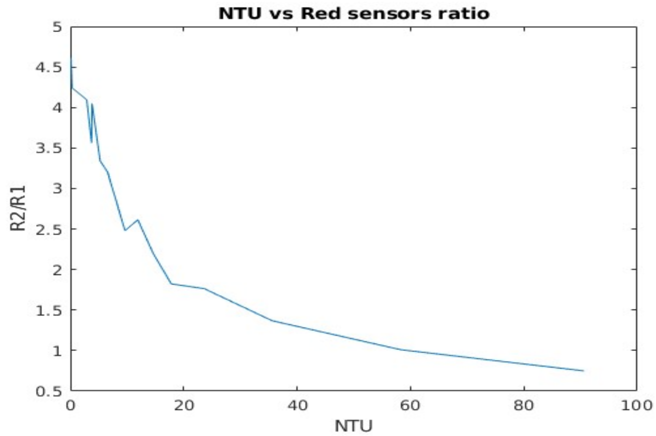
This plot gives how G values of various sensors are increasing. Correlation between them



Calibration curves with Green color

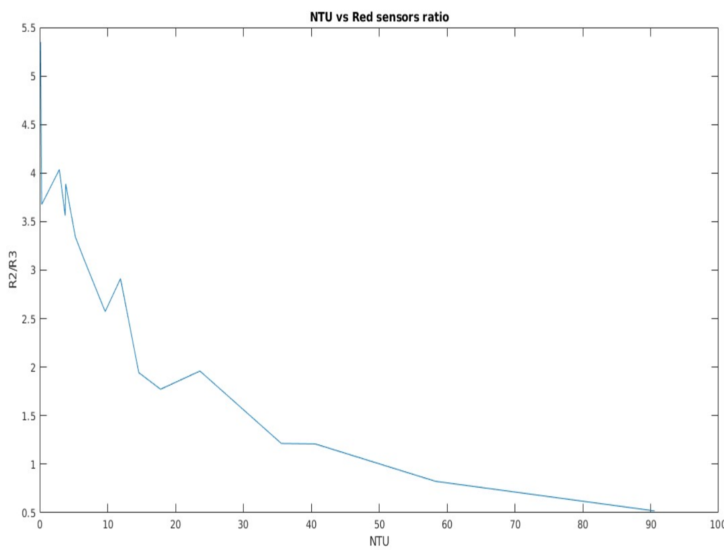
This plot gives how individual R values as varying

In the real life application, we observed that R, G, B values are changing (Amplified due to surrounding light) so, we have moved our factor from Average to Ratio. Red sensors are the best ratio that is exponentially decreasing and useful for our purpose



R2/R1 vs standard NTU

Ratio of average Red value of sensor 2 to sensor1. It is exponentially decreasing



R2/R3 vs standard NTU

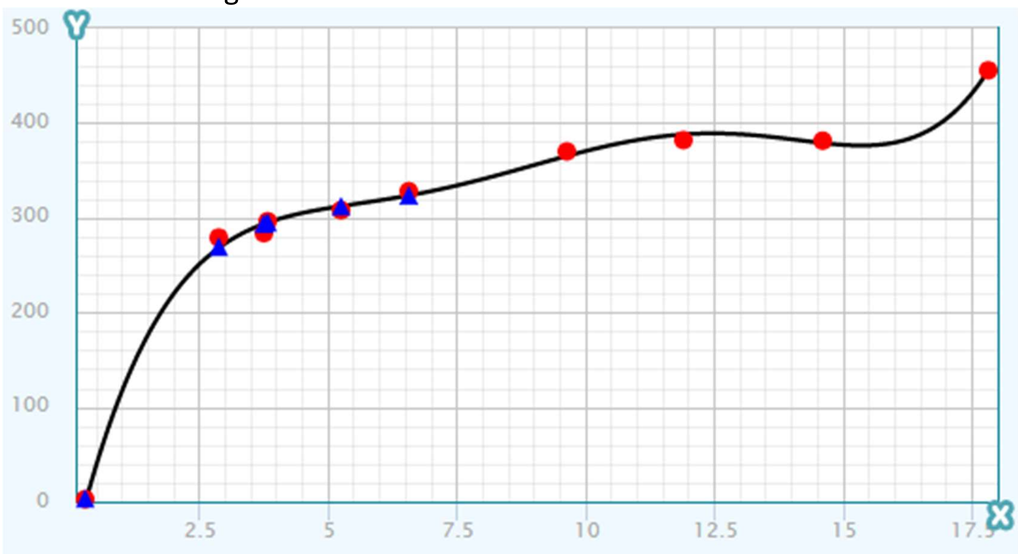
Ratio of average Red value of sensor 2 to sensor3. It is exponentially decreasing

ERROR In Calibration:

As we are dealing with red values. Let's calculate the error in red, by using curve fitting

y = -57.91423 + 227.859*x - 56.32649*x^2 + 6.811235*x^3 - 0.383248*x^4 + 0.008038626*x^5

where Y: Value from sensor
X: NTU reading



Curve fitting for Red

NTU	Sensor Reading	Equation	Error %
0.29	2.62	3	14%
2.89	279	269	3%
3.75	283.67	293	3.5%
3.82	296.3	295	0.3%
5.25	308	302	1.9%
6.57	328	323	1.5%
9.64	370	365	1.3%
11.9	382	388	1.5%
14.6	381.33	378	0.7%
17.8	455.67	456	0.2%

Readings

Std Value		R	G	B	R Avg	G Avg	B Avg	R2/R1	R2/R3	R1/R3
blANK		123	9	21	265.33	20.33333333	39.33333333	4.609756098		1.160377358
		567	30	75					5.349056604	
		106	22	22						
0.29	S1	123	14	23	262.33	24	40.66666667	4.243902439		0.8661971831
	S2	522	30	71					3.676056338	
	S3	142	28	28						
2.89	S1	137	15	26	279	24.33333333	44	4.094890511		0.9856115108

	S2	561	31	79					4.035971223	
	S3	139	27	27						
3.75	S1	153	15	28	283.67	24.33333333	44.33333333	3.562091503		1
	S2	545	30	76					3.562091503	
	S3	153	28	29						
3.82	S1	146	16	28	296.33	26	46.33333333	4.047945205		0.9605263158
	S2	591	33	81					3.888157895	
	S3	152	29	30						
5.25	S1	173	17	32	308.00	26.33333333	48.66666667	3.341040462		1
	S2	578	32	81					3.341040462	
	S3	173	30	33						
6.57	S1	188	18	34	328.00	27.66666667	51.66666667	3.20212766		0.9690721649
	S2	602	34	85					3.103092784	
	S3	194	31	36						
9.64	S1	250	21	45	370.33	29.33333333	58.33333333	2.48		1.037344398
	S2	620	34	87					2.572614108	
	S3	241	33	43						
11.9	S1	254	22	45	382.00	30.33333333	59.33333333	2.614173228		1.114035088
	S2	664	36	92					2.912280702	
	S3	228	33	41						
14.6	S1	264	22	46	381.33	30.33333333	59.33333333	2.200757576		0.8829431438
	S2	581	32	80					1.943143813	
	S3	299	37	52						
17.8	S1	355	28	63	455.67	34.66666667	72.33333333	1.822535211		0.9726027397
	S2	647	35	91					1.77260274	
	S3	365	41	63						
23.6	S1	391	29	68	477.67	34.66666667	74	1.764705882		1.110795455
	S2	690	36	96					1.960227273	
	S3	352	39	58						
35.6	S1	473	34	82	551.00	40	87	1.367864693		0.8874296435
	S2	647	35	92					1.213883677	
	S3	533	51	87						

40.6	S1	525	36	91	587.33	40.33333333	91.33333333	1.28952381		0.9375
	S2	677	35	96					1.208928571	
	S3	560	50	87						
58.3	S1	580	39	99	625.67	42.33333333	96	1.010344828		0.8157524613
	S2	586	31	83					0.8241912799	
	S3	711	57	106						
90.6	S1	686	46	118	730.33	48.33333333	113.3333333	0.749271137		0.6922300706
	S2	514	27	74					0.5186680121	
	S3	991	72	148						

Table showing variation in color values with standard turbidity values for the 1st experiment

Classification Design

Sample NTU Range	R Range	G Range	B Range	R2/R1 Range	R2/R3 Range
<1 NTU	250-263	20-24	30-41	>4.24	4.2-5

Table for Range of outputs for water sample <1 NTU

Sample NTU Range	R Range	G Range	B Range	R2/R1 Range	R2/R3 Range
1-5 NTU	265-300	24.3-26.7	40.6-48.6	3.34-4.2	3.34-4.03

Table for Range of outputs for water sample 1-5 NTU

Sample NTU Range	R Range	G Range	B Range	R2/R1 Range	R2/R3 Range
6-15 NTU	328-384	27.6-30.3	51.6-60	3.2-2.2	3.1-1.9

Table for Range of outputs for water sample 6-15 NTU

Sample NTU Range	R Range	G Range	B Range	R2/R1 Range	R2/R3 Range
>15 NTU	455-700	34-48	72-118	1.8-0.7	1.7-0.7

Table for Range of outputs for water sample >15 NTU

Sample NTU Range	R Range	G Range	B Range	R2/R1 Range	R2/R3 Range
No water	<260	<15	<30.5	4.06	5.34

Table for Range of outputs without water

Test Code

Code for getting values

```
/*calibrating sensors*/


#include <Wire.h>

#include "Adafruit_TCS34725.h"

#define MUX_Address 0x70

#include<LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);


Adafruit_TCS34725 tcs_1 = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_101MS,

                    TCS34725_GAIN_60X);


Adafruit_TCS34725 tcs_2 = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_101MS,

                    TCS34725_GAIN_60X);

Adafruit_TCS34725 tcs_3 = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_101MS,

                    TCS34725_GAIN_60X);

uint16_t red_1, green_1, blue_1, clear_1;

uint16_t red_2, green_2, blue_2, clear_2;

uint16_t red_3, green_3, blue_3, clear_3;


int ind=0;

float myarray[9];

float avg1=0,avg2=0;

void tcaselect(uint8_t i2c_bus) {

   if (i2c_bus > 7) return;

   Wire.beginTransmission(MUX_Address);

   Wire.write(1 << i2c_bus);

   Wire.endTransmission();

}
```

```
void setup()
{
   lcd.init();                     // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(1,0);
  lcd.print("Code on");
  lcd.setCursor(1,1);
  lcd.print("setting up");
  delay(2000);

  digitalWrite(2,HIGH);
  Serial.begin(9600);
 // lcd.begin(16, 2);
  Serial.println("CODE ON!!");

  tcaselect(1); if (tcs_1.begin())
  {
    Serial.println(" DETECTED FIRST SENSOR");
  }
  if (!tcs_1.begin())
  {
    Serial.println("sensor1 not found");
  }

  tcaselect(2);
  if (tcs_2.begin())
  {
    Serial.println(" DETECTED SECOND SENSOR");
```

```
  }
  else {
    Serial.println("sensor2 not found");
  }
  tcaselect(3);
  if (!tcs_3.begin())
  {
    Serial.println("sensor3 not found");
  }
  if (tcs_3.begin())
  {
    Serial.println(" DETECTED THIRD SENSOR");
  }
  for (int i = 0; i < 9 ; i++)
  {
    myarray[i]=0;
  }
}


void loop()
{
  tcaselect(1);
  tcs_1.getRawData(&red_1, &green_1, &blue_1, &clear_1);

  tcaselect(2);
  tcs_2.getRawData(&red_2, &green_2, &blue_2, &clear_2);
  tcaselect(3);
  tcs_3.getRawData(&red_3, &green_3, &blue_3, &clear_3);

  myarray[0]+=red_1;
```

```
myarray[1]+=red_2;
myarray[2]+=red_3;
myarray[3]+=green_1;
myarray[4]+=green_2;
myarray[5]+=green_3;
myarray[6]+=blue_1;
myarray[7]+=blue_2;
myarray[8]+=blue_3;
ind++;

if(ind == 50)
{
 // Wire.beginTransmission(MUX_Address);
 digitalWrite(2,LOW);
 delay(5000);
//   lcd.setCursor(2, 0);
 //lcd.print("working");
  avg1=(myarray[1])/(myarray[0]);
  avg2=(myarray[1]/myarray[2]);
  Serial.print(avg1);
  Serial.print("- ");
  Serial.println(avg2);
  if(avg2>=4.5)
  {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("ADD WATER!!!");
  }
  else if(avg1>4.05 || avg2>4.05)
  {
    //lcd.clear();
```

```
  lcd.setCursor(0, 0);
  lcd.print("Drinking water..");
  lcd.setCursor(0, 1);
  lcd.print("<1NTU");
}
else if((avg1>=3 && avg1<=4.05)||(avg2>=3 && avg2<4))
{
 // lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Tap Water..");
  lcd.setCursor(0, 1);
  lcd.print("1-6NTU");
}
else if((avg1>=2 && avg1<3)||(avg2>=2 && avg2<3))
{
 // lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Dirty Water..");
  lcd.setCursor(0, 1);
  lcd.print("7-15NTU");
}
else
{
 // lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Waste water..");
  lcd.setCursor(0, 1);
  lcd.print(">15NTU");
}
for(int i=0;i<9;i++)
  myarray[i]=0;
```

```
  ind=0;

  // Wire.begin();

  digitalWrite(2,HIGH);

 }

}
```

Code for calibration

```
/*calibrating sensors*/

#include <Wire.h>
#include "Adafruit_TCS34725.h"
#define TCAADDR 0x70
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

Adafruit_TCS34725 tcs_1 = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_101MS,
                    TCS34725_GAIN_60X);
Adafruit_TCS34725 tcs_2 = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_101MS,
                    TCS34725_GAIN_60X);
Adafruit_TCS34725 tcs_3 = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_101MS,
                    TCS34725_GAIN_60X);
uint16_t red_1, green_1, blue_1, clear_1;
uint16_t red_2, green_2, blue_2, clear_2;
uint16_t red_3, green_3, blue_3, clear_3;

int ind=0;
int myarray[9];

void tcaselect(uint16_t i)

{
 if (i > 7) return;
 Wire.beginTransmission(TCAADDR);
 Wire.write(1 << i);
 Wire.endTransmission();
}

void setup()
{
 Serial.begin(9600);
 lcd.begin(16, 2);
 lcd.clear();
```

```
  lcd.setCursor(0, 0);
  lcd.print("SETTING UP!");
//  delay(5000);

  Wire.begin();
  Serial.println("CODE ON!!");
  tcaselect(1); if (tcs_1.begin())
  {
    Serial.println(" DETECTED FIRST SENSOR");
  }
  if (!tcs_1.begin())
  {
    Serial.println("sensor1 not found");
  }
  tcaselect(2);
  if (tcs_2.begin())
  {
    Serial.println(" DETECTED SECOND SENSOR");
  }
  else {
    Serial.println("sensor2 not found");
  }
  tcaselect(3);
  if (!tcs_3.begin())
  {
    Serial.println("sensor3 not found");
  }
  if (tcs_3.begin())
  {
    Serial.println(" DETECTED THIRD SENSOR");
  }
  for (int i = 0; i < 9 ; i++)
  {
    myarray[i]=0;
  }
}


void loop()
{
  tcaselect(1);
  tcs_1.getRawData(&red_1, &green_1, &blue_1, &clear_1);
  tcaselect(2);
  tcs_2.getRawData(&red_2, &green_2, &blue_2, &clear_2);
  tcaselect(3);
  tcs_3.getRawData(&red_3, &green_3, &blue_3, &clear_3);
```

```
 myarray[0]+=red_1;
 myarray[1]+=green_1;
 myarray[2]+=blue_1;
 myarray[3]+=red_2;
 myarray[4]+=green_2;
 myarray[5]+=blue_2;
 myarray[6]+=red_3;
 myarray[7]+=green_3;
 myarray[8]+=blue_3;
 ind++;

 if(ind ==50)
{
  for(int i=0;i<9;i++)
   Serial.println(myarray[i]/50);
 }
}
```

CONCLUSION:

The device is of low cost and easy to use. The construction materials are easily obtainable. This system can be used as a part of a low cost sensor network to provide water quality information to consumers. If installed at the consumer end will provide water quality information to the user both at industrial and household levels. Data collected can be put up on a website for security.

We can use communication modules like sim900 to transmit messages to required authority on water quality and analyse them to make good conclusions. IOT modules like Nodemcu can be used for storing data on server and run Machine learning modes to to analyse data and take corresponding actions.