# TURMET

# PROJECT REPORT


Group Members

Haris Suhail (4787)

Muhammad Oneeb Ul Haq Khan (7692)

Sameed Qureshi (6790)

## Table of Contents

# 1.0 - Introduction

Water is essential to almost everything that happens around us. Water is essential to life, and everything else that allows for it to prevail. For different processes, there are different requirements for water conditions. One of the most important parameters to judge water's ability to harbor and propagate life is its turbidity. Turbidity is the measure of the haziness or cloudiness of water due to suspended solid particles. To provide an effective, reliable and easy-to-use method, our group has made a Turbidimeter, or as it is more fondly called: "TURMET".

## 1.1 - Motivation

High levels of water turbidity is becoming an increasingly common global phenomenon. Due to our relentless and inconsiderate treatment of waste products from our countless industries, water is constantly being contaminated. Fresh water bodies are especially falling victim to this. As it happens fresh water bodies are home to a large percentage of all the species present today, as well as an incredibly significant sustainer of human civilizations, cities, countries, life and industries, and high levels of water turbidity threaten all of these. Turbidity is also an indicator of the amount of bacteria and viruses present in the water, and therefore an indicator of whether is not the water is capable of human consumption (usually the upper limit for drinking water is 5 NTU). All over the globe a huge population drinks water without having any knowledge of what turbidity is and how that affects them – and if they are they still consume water with turbidity levels more than advised, and a large portion of this population suffers from many water-borne diseases. To tackle this problem, i.e. a device that not only creates awareness of water turbidity but is also user friendly and cost effective in terms of measuring this parameter, our group had decided to come up with TURMET.

## 1.2 - Background

There are several devices available in the market capable of measuring turbidity. Such devices use one of the numerous methods that may be used to measure turbidity. One method that is commonly used is the ratio method. TURMET is a turbidimeter employing the ratio method.

However, TURMET is capable of doing more than the available devices. It is much smaller in size and much less in weight making it much more portable. It uses a powerful STM32 microcontroller so that values are obtained much more rapidly. This allows a real time turbidity monitoring system to be developed, and TURMET, accompanied by its android application of the same name, is just that.

The ability of TURMET to monitor flowing water and plot a graph makes it something a little more than those numerous devices available in the market.

# 2.0 - Project Design and Implementation

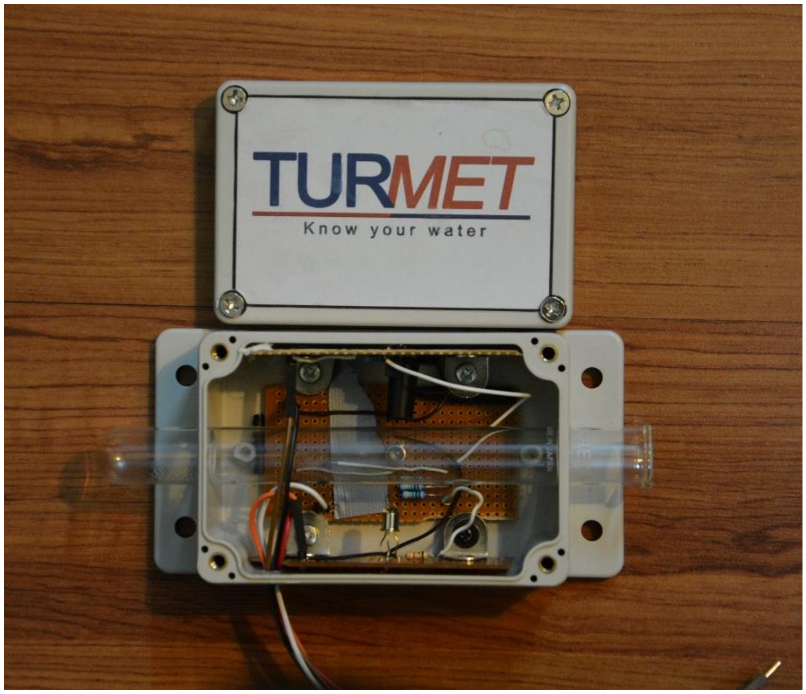



Figure 1 : Final product form

## 2.1 - Components Used

- STM32 (Microcontroller)
- LED
- 2N2222 Transistors
- ST-1KL3B Phototransistors
- HC-05 Bluetooth Module
- IP66 Waterproof casing

## 2.2 - Working Principle

There are more than numerous methods to calculate the Turbitidy of water. After research on a number of different methods, we chose one which was both accurate, and easily implementable. The two-detector measurement system, also known as the scatter light ratio to transmitted light method was chosen. This was mainly due to its high reliability and its ability to give an output in Nephalometric Turbidity Units (NTU).

This method, instead of using just one light detector to measure the amount of light passing through the water, uses two detectors placed at 90 degrees to each other. How does this work? Turbid water i.e. water with solid particles, causes light to scatter. The greater the turbidity of water, the greater will be the scattering of light. The phototransistors detect these amounts of light and send their readings to the microcontroller which then carries out the calculations.

The basic working principle is illustrated in the Figure 2 below:

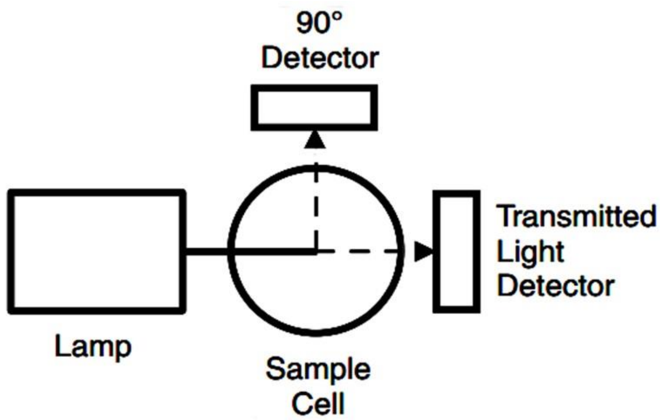



Figure 2

A light proof mechanical structure was then made and the circuit was setup in it. The light detector measuring the transmitted light was made less sensitive as much more light passes through. The scattered light detector was made more sensitive as much less scattered light is received.

Finally, the device was giving raw data that needed to be calibrated. This calibration was done using a commercially available Turbidimeter – HACH 2100P.

## 2.3 - Light Detecting

The component used to detect the amount of light was the **Kodenshi ST-1KL3B**, which is a high sensitivity NPN Silicon **phototransistor**. A phototransistor has an exposed base that amplifies the light that it comes in contact with. This causes a relatively high current to pass through the phototransistor. As the current spreads from the base to the emitter, the current is concentrated and converted into voltage.

The phototransistor has various advantages over the photodiode and the light dependant resistor (LDR). These advantages were the requirements for out project, which was the reason this component was chosen over other optical sensors. The ST-1KL3B phototansistor offers the following features:

a) More sensitive than the photodiode and the LDR
b) Provides voltage output unlike the LDR
c) Faster – provides almost instantaneous results
d) Reliable - preffered for outdoor usage
e) Low cost

As for the light source in our project, a Red LED was used. The reason behind this can be understood by looking at the figure 3 below. It shows the relative sesitivity of the ST-1KL3, and as it can be seen, the phototransistor is most sensitive for wavelengths of light that correspond to Red light.

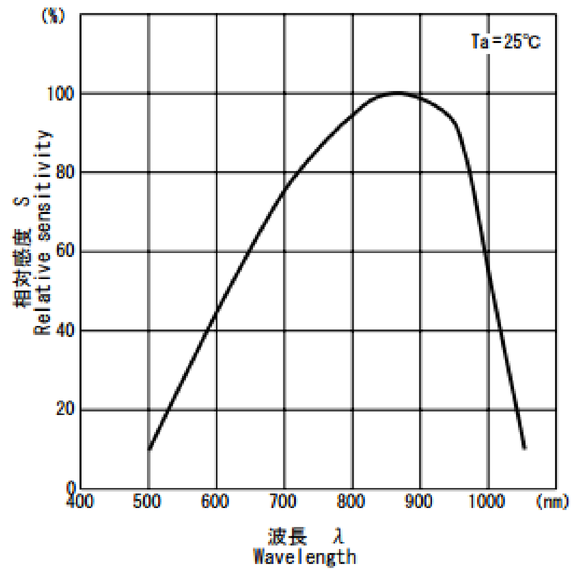



Figure 3

For proper and accurate operation of the phototransistors, it is required the light source must be placed directly in front. In order to fulfill this requirment, we used a plastic covering to make sure the LED light was concentrated directly towards the phototransistor.

## 2.4 - Device Casing

The basic idea behind our device was to make it portable and easy to use in various conditions. In order to accomplish this, we enclosed out circuitry in an IP66 Waterproof plastic box, making it resistant to dust and jets of water, as well as making it light proof. The box measured 100 X 68 X 50mm/3.94" x 2.68" x 1.97" (L X W X H).

Being waterproof and dust resistant allows our device to be installed outdoors, whether it be on the household water supply input, or at the output of a water filter, or even to check the turbidity of pumped ground water. This set up also allows our device to be immersed into water to check the turbidity of surface water of a water body.

## 2.5 - Microcontroller STM32 Nucleo F103b

The brain behind our device is the powerful STM32 microcontroller, the Nucleo F103b. It receives voltages from the two phototransistors, and uses its ADCs (Analog to Digital converters) to give us digital values which are then used to calculate the Turbidity. Its fast ADC conversion allowed us to get readings from the phototransistors at an impressively high speed, making the calculation time of our final value minimum while keeping it within an allowed range of accuracy.

## 2.6 - The ADC

### 2.61 - The basics

The STM32 Nucleoboard used has 2 ADCs and 15 channel per ADC. Other than the standard ADC channels, there is also a channel for the temperature sensor. The ADCs are 12 bit ADCs with a conversion range of 0 – 3.6V. That is, they can have a minimum step size of:

$$\frac{3.6}{2^{12}} = 0.879mV$$

The conversion speed of the ADCs is rated at 1us. This allows for rapid measurement of voltage values required for the continuous monitoring of turbidity values.

### 2.62 - ADC conversion modes

There are a variety of conversion modes allowed by the STM32 microcontroller. A scan conversion mode enables the ‘scanning’ of all the channels that are activated for a particular ADC. For example, a ADC with 10 active channels in scan conversion mode will give the value of each channel one by one.

Secondly, a continuous conversion mode makes it possible for the programmer to make the ADC work independently at the back end, only interacting with it through interrupts.

Lastly, there is the discontinuous conversion mode that was used in this project, since the main loop of the code involved the ADC. The ADC must be restarted after each value.

### 2.63 - Removing ADC reading fluctuations

As mentioned before, the ADC has a very high precision. This makes it output values susceptible to thermal noise and etc. So, in order to reduce such fluctuations, the code is written such as to take the average of a large number of ADC values (namely 400 values). This considerably reduces the voltage fluctuations.

## 2.7 - Independent watch dog timer (IDWG)

An independent watch dog timer is a safety precaution. If the code gets stuck (for example in an infinite loop), the IWDG resets the microcontroller. Such type of safety precautions are necessary in any commercial product as such products must be able to recover from malfunctions without external help.

### 2.8 - UART Communication and Bluetooth Low Energy (BLE)

The STM32 allows 3 fully duplex UART channels. They can operate in a variety of modes and speeds. In this application, the UART1 and UART2 were used. The following table shows the use of each:

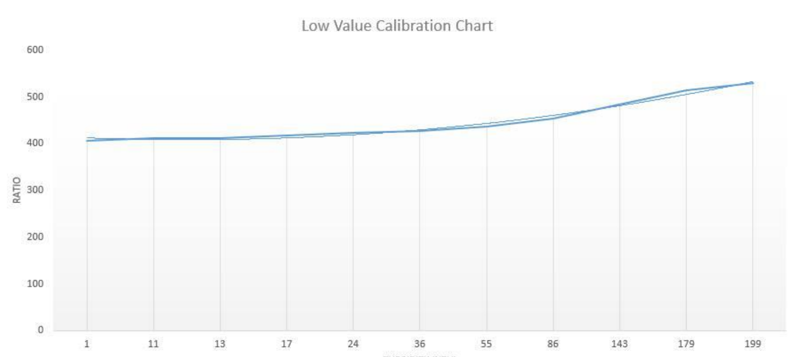
| UART | Baud Rate | Mode | Purpose |
|---|---|---|---|
| **UART1** | 9600 | Asynchronous | To communicate with BLE module |
| **UART2** | 115200 | Asynchronous | For debugging and passing messages to PC |

The 9600 baud rate, despite much lower than the maximum capacity, was adequate as the STM32 sends the data to the BLE module once every 800ms. UART1 is used in fully duplex mode. This allows it to not only send data to the BLE module (which is further sent by the BLE module to the app) but also receive data from it. This feature will be helpful in expanding and the upgrading the project as the STM32 would be able to receive commands from the android application through the BLE module.

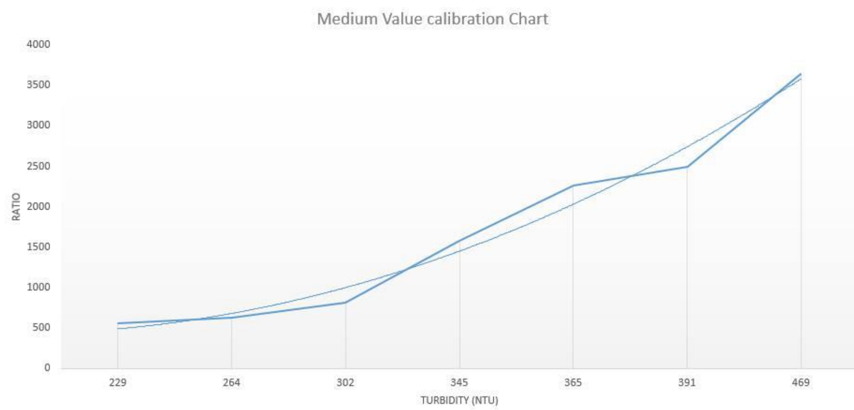
### 2.9 - Calibration

A commercially available turbidimeter was used to obtain the values of turbidity corresponding to the raw data being obtained from the TURMET device. A chart of the data was plotted and linear regression was used to extrapolate the data and determine the value of turbidity for between the calibration data points. The chart plotted showed three distinct regions:

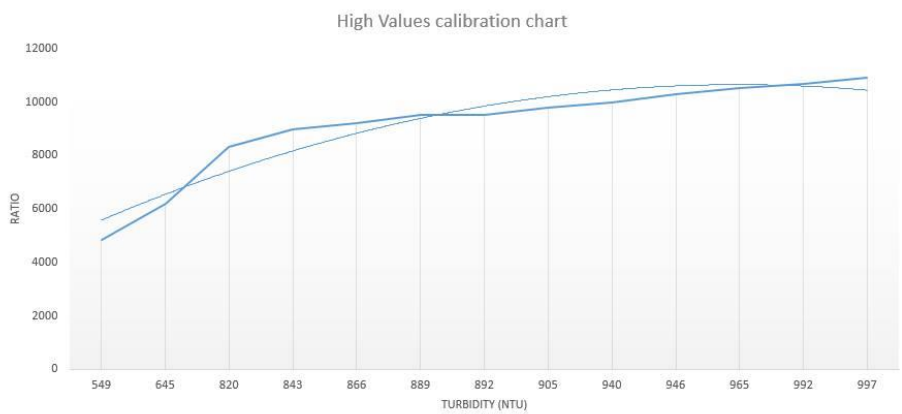
The low turbidity region:




The medium value turbidity region:




The high value turbidity region:




The same algorithm was used individually for each of the three regions.

$$Y' = b_{yx} X + a_{yx}$$

$$b_{yx} = r_{xy} \frac{\sigma_y}{\sigma_x} = \frac{N \sum XY - (\sum X \sum Y)}{N \sum X^2 - (\sum X)^2}$$

$$a_{yx} = \overline{Y} - b_{yx} \overline{X}$$

These formulas enabled us to estimate the value of turbidity in between the data points.


## 3.0 – Android Application

For allowing a user-friendly experience while measuring water turbidity, we opted to create an android application. TURMET and its app, of the same name, work in synch to allow for real time turbidity data to be provided to the device user.

The TURMET application starts at the home screen, where the user has the option of either viewing the real time turbidity values, or another of viewing the trend of turbidity over a period of time by means of a line graph.

The four activities of the application are discussed below:

### 3.1 - Main Activity

Main Activity is the first screen that the users come across when using the application. This activity provides an introductory animation to the device "TURMET", it's logo, and it's slogan. After the animation ends, the next activity is called.

### 3.2 – TURMET Menu Activity

The second activity called is the TURMET Menu Activity. The main purpose of creating this activity was to allow the users to navigate within the app, and choose the means by which they would like to receive their turbidity information. From here the users can either choose "Real Time Values", or they can choose "Real Time Graph"

### 3.3 – Real Time Values

The Real Time Values activity is the heart of the whole application as it does the most crucial and imperative tasks, as well as provides the most important data (around which this whole project is centered). When this activity is created, an onCreate function is called. This function sets up the starting conditions of the activity as required. In this case, the function sets the layout of the activity, and it prepares the activity to be ready to receive values from the device via Bluetooth.

The onCreate function checks if Bluetooth has been enabled, if it hasn't it proceeds to do so, and subsequently it checks whether the phone is connected to the Bluetooth device – and, again, if it hasn't it proceeds to do so. The main bulk of problems faced in the application was faced within this small step of establishing Bluetooth connectivity. One downside of having such a large number of operations in the onCreate was that there was a noticeable lag when starting the activity.

Once Bluetooth is up and running, we call upon our handler function which, as the name suggests, deals with our messages, which in this case are received messages. The received string, being transmitted by the device, is forwarded to be displayed in a Textview for the user. The value from the string is also converted into an integer and passed to a variable “reading” of the Real Time Graph activity.

To make sure that the transmitted and received values are the same, we add a few checks and balances: the delay between the sending/receiving of values on either end is set 800ms, the device always transmits 4 characters, and the app only displays the string of values when the received string exceeds 3 characters.

### 3.4 – Real Time Graph

For allowing the users to see a visible trend in the turbidity values of water, we established a graphing function which plots the values of turbidity against time. The value of turbidity is obtained by the handler from the previous activity, which assigns the turbidity value from the string being received.

## 4.0 – Results

After the device was calibrated, it was able to give the turbidity readings close to the actual value. The worst case error was 10%.

The device gives steady readings of turbidity for a homogenous sample. If the sample is settling, then the values fall as the sample clears.

The device gives one reading every 800ms. This is adequate speed to plot a graph and to monitor the turbidity continuously.

Environmental conditions may cause an error in the readings. Since the condition vary unpredictably, the correction can has to be applied whenever the device is used in a new environment. This is done by using a known clear sample of less than 0.5 NTU turbidity and calibrating it by pressing the blue button on the Nucleoboard.

## 5.0 – Challenges

We faced numerous challenges while making our project, however by maintaining a spirit of perseverance we managed to overcome all of them.

- Lack of information on the Nucleoboard on Kiel
- Making the device housing light-proof
- Fixing the circuitry precisely (especially the LED and the phototransistors)
- Creating an Android application without sufficient prior knowledge of the Java language
- Establishing Bluetooth connectivity with the device and the application

## 6.0 - Possible Improvements

The device can be further reduced in size by decreasing the space between the light source and the detectors, as well as designing it for a vessel of smaller size. However, this would require even more accurate calibration and may result in flawed results due to reflection of light from the glass body of the vessel.

The device can be made completely wireless by using an RF module. This would allow the device to send data to the microcontroller wirelessly and make it independent of any wire-connected microcontroller. This would, however, add to the cost of the device. It may also result in a reduced data speed depending on the transmission rate of the RF module.

## 7.0 – Conclusion

TURMET has been a successful endeavor, which has allowed us to develop a wide range of new skills, i.e. research in new fields, Android programming, product finishing, use of new and emerging controllers and boards, etc. The culmination of our efforts have allowed us to make a cheap, light-weight, accurate and easy-to-use alternative to the turbidimeters being used in chemistry and other relevant laboratories across the globe.

## 8.0 – Bibliography

1. Help in most android application related queries:
   stackoverflow.com

2. Help in understanding android related functions:
   developer.android.com

3. Setting up a line chart:
   https://www.youtube.com/watch?v=a20EchSQgpw