

Development of Water Pollution Monitoring System

Presented By

- Namrata Thakre
- Shweta Lanjewar
- Natasha Ramtekkar
- Himanshu Saha
- Vijayalaxmi Yelchalwar



Guided By

Ms. Snehal P. Dongre

Department of Information Technology

**S. B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT AND
RESEARCH, NAGPUR**

An Autonomous Institute, Affiliated to RTMNU, Nagpur

Contents

- Introduction
- Literature Survey
- Problem Statement and Objectives
- System Design
- Technology to be Used
- Proposed Work
 - Module Description
 - Screenshots
- Advantages & Applications
- Plan of Work
- References

Introduction

- The water quality monitoring is very important now a days. As there are various resources available but still no progress. In most cases the water gets contaminated due to different reasons and one of the major reason is industrial waste. This needs to be stopped somewhere.
- Conventional water quality checking methods include sampling and laboratory techniques. These strategies are however not cost effective and time consuming which eventually leads to delayed in detection of impurities and reaction to those contaminants in water. So, there should be more efficient and productive checking strategies to monitor the quality of water.
- To cure this real time monitoring of water quality should be done. So, the RC boat is used for continuously moving on water surface to identify water quality. The different sensors are interfaced to the controller viz. water temperature sensor, air temperature & humidity sensor, Ph sensor, Turbidity sensor.

Literature Survey

Websites / Paper / Article /APP/ Website

Topic: Arduino Based Water Quality
Monitoring System

Authors: Wong Jun Hong, Norazanita
Shamsuddin

researchgate.net | January 2021

Topic: Internet of things enabled real time
water quality monitoring system

Authors: S. Geeta & S. Gouthami
Journal of Springer Open | 2017

Topic:-Water Quality Monitoring for Rural
Areas

Authors: Nikhil Kedia
2015 IEEE

Reviews / Findings

1. Wireless data transfer required.
2. Proper Data Analysis Required.

1. Stationary System.
2. Geographic data required for mapping of polluted streams.

- 1 . This paper focuses on a sensor-actuator system which later shifts to Sensor-Cloud model.

Websites / Paper / Article /APP/ Website

Topic:-Design and implementation of a cost-effective water quality evaluation system

Authors: Md. Omar Faruq, Injamamul Hoque Emu, Md. Nazmul Haque ,Maitry Dey ,N. K. Das , Mrinmoy Dey
IEEE (2017)

Article:-Water Pollution: Everything You Need to Know

Author: Melissa Denchak | May 2018

Reviews / Findings

Water parameter such as temperature, turbidity, and pH displayed on LCD screen.

Details about water pollution. Its types , causes and ways to prevent water pollution.

Problem Statement & Objectives

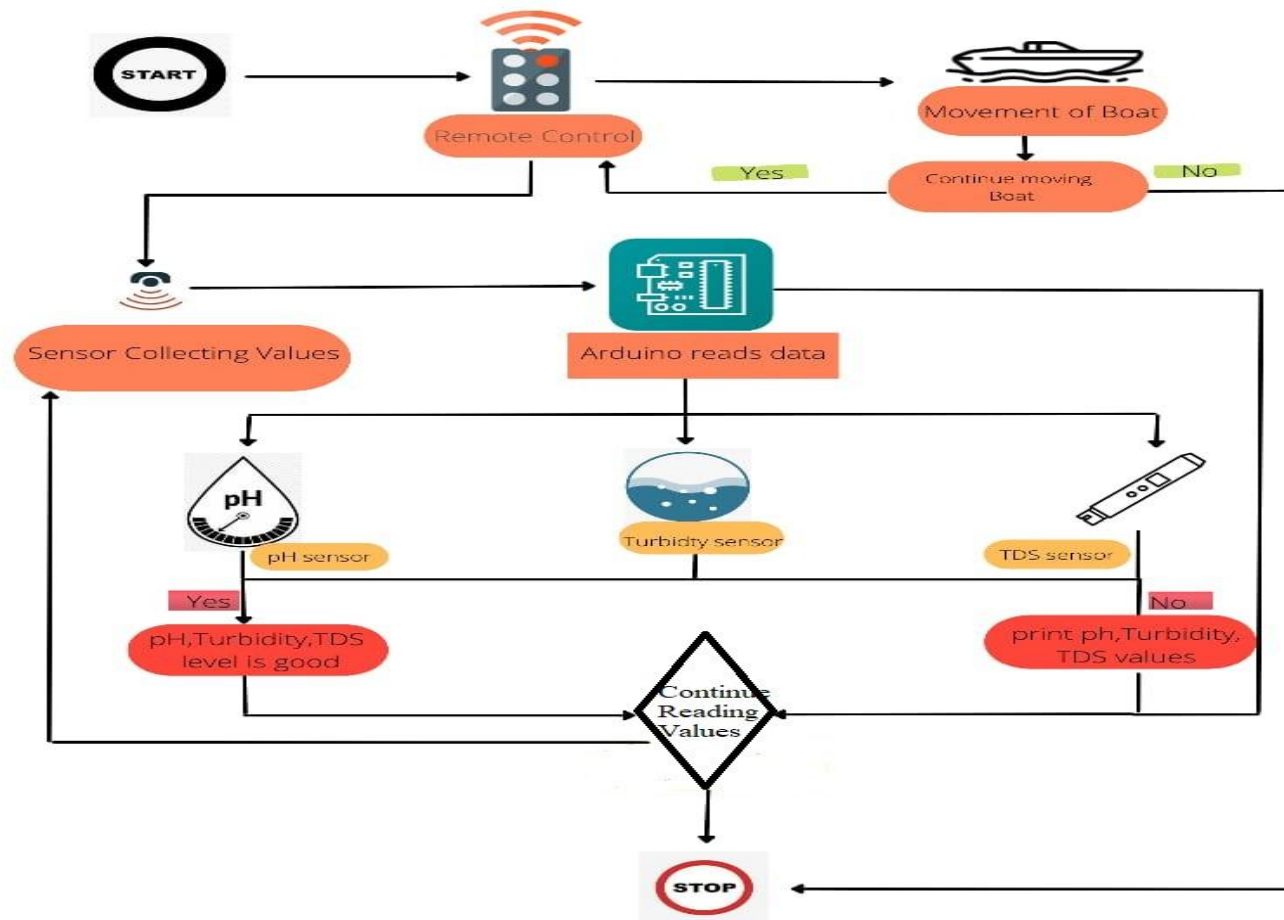
Problem Statement

The quality of drinking water plays a crucial role in the health of animals and human beings. Lakes and reservoirs, canals one of the major sources of drinking water. Water quality monitoring of these water bodies requires a lot of effort as operators need to get in a boat with all sensors and manually check the entire water body. The current manual method for monitoring of the water quality requires Money, Time consuming and labor intensive. So, we are designing a solution for easy water quality checking of vast water bodies with ease.

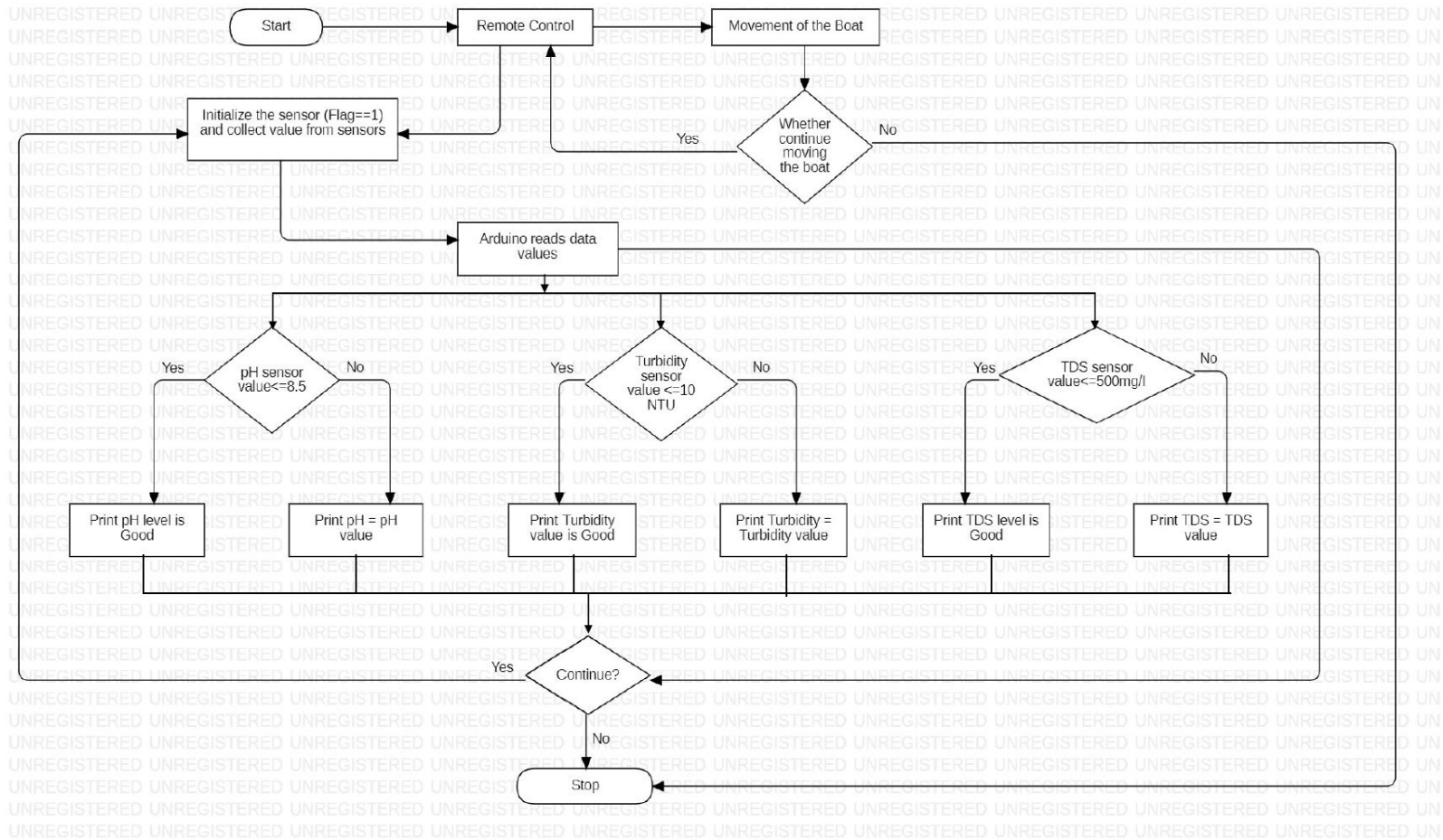
Objectives

- To study, water quality and importance of water quality monitoring.
- To develop a system with Ease and convenience of usage.
- To improve data collection system in survey of water quality for large water bodies

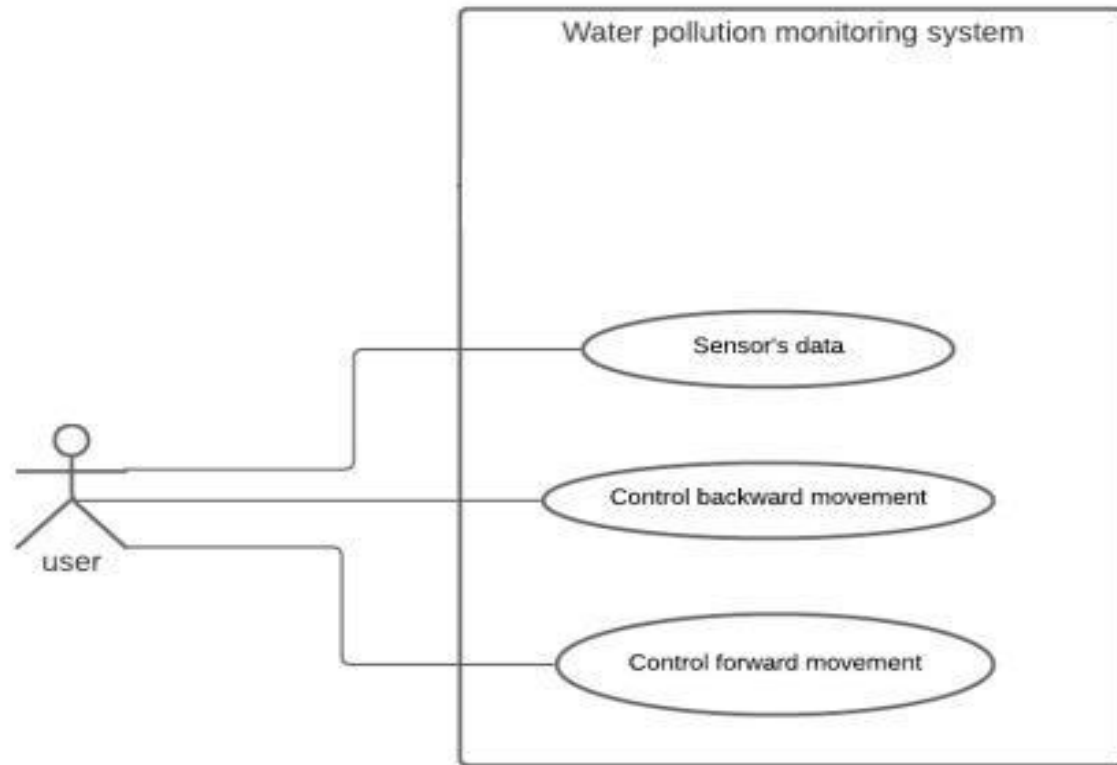
System Architecture



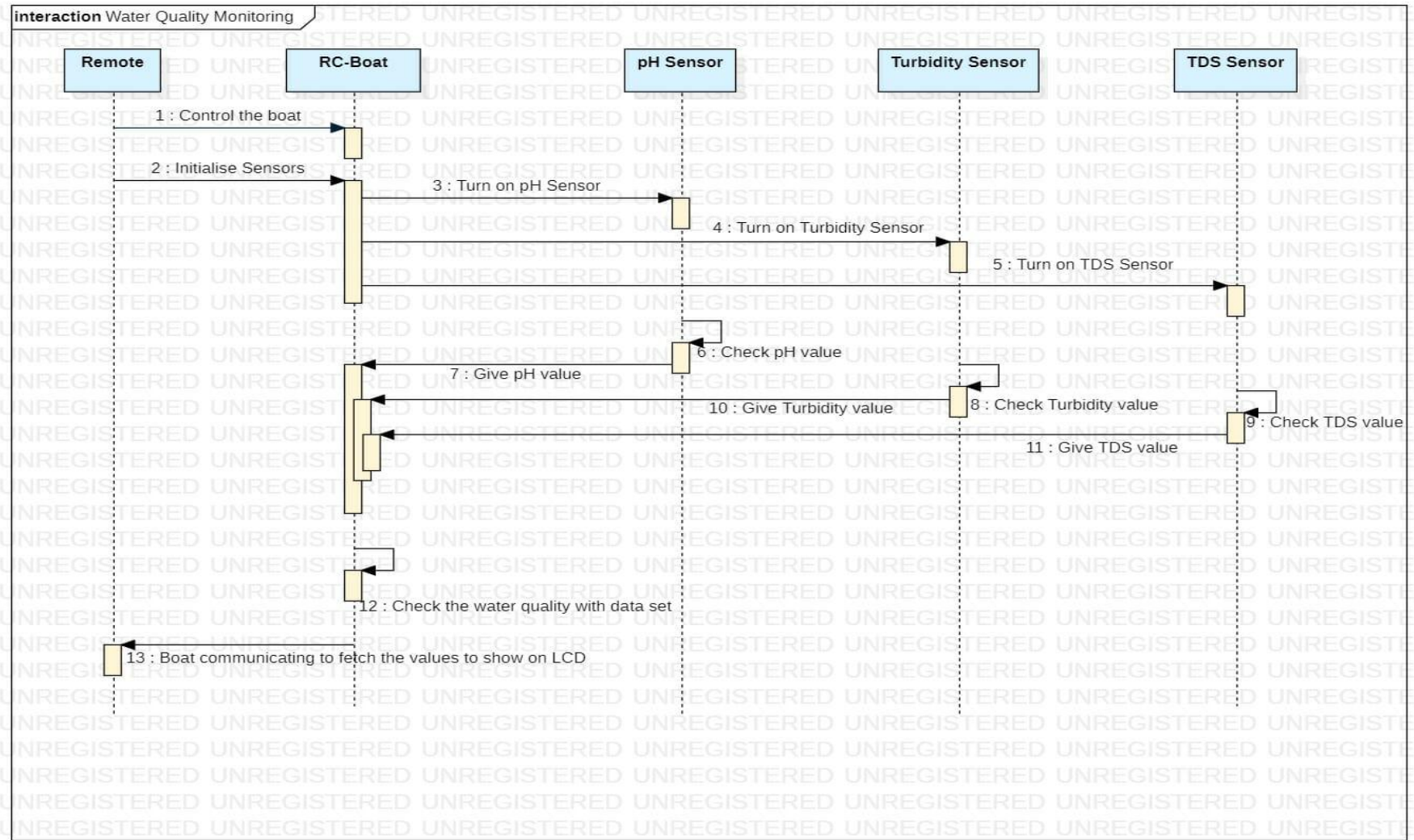
System Design: Flowchart



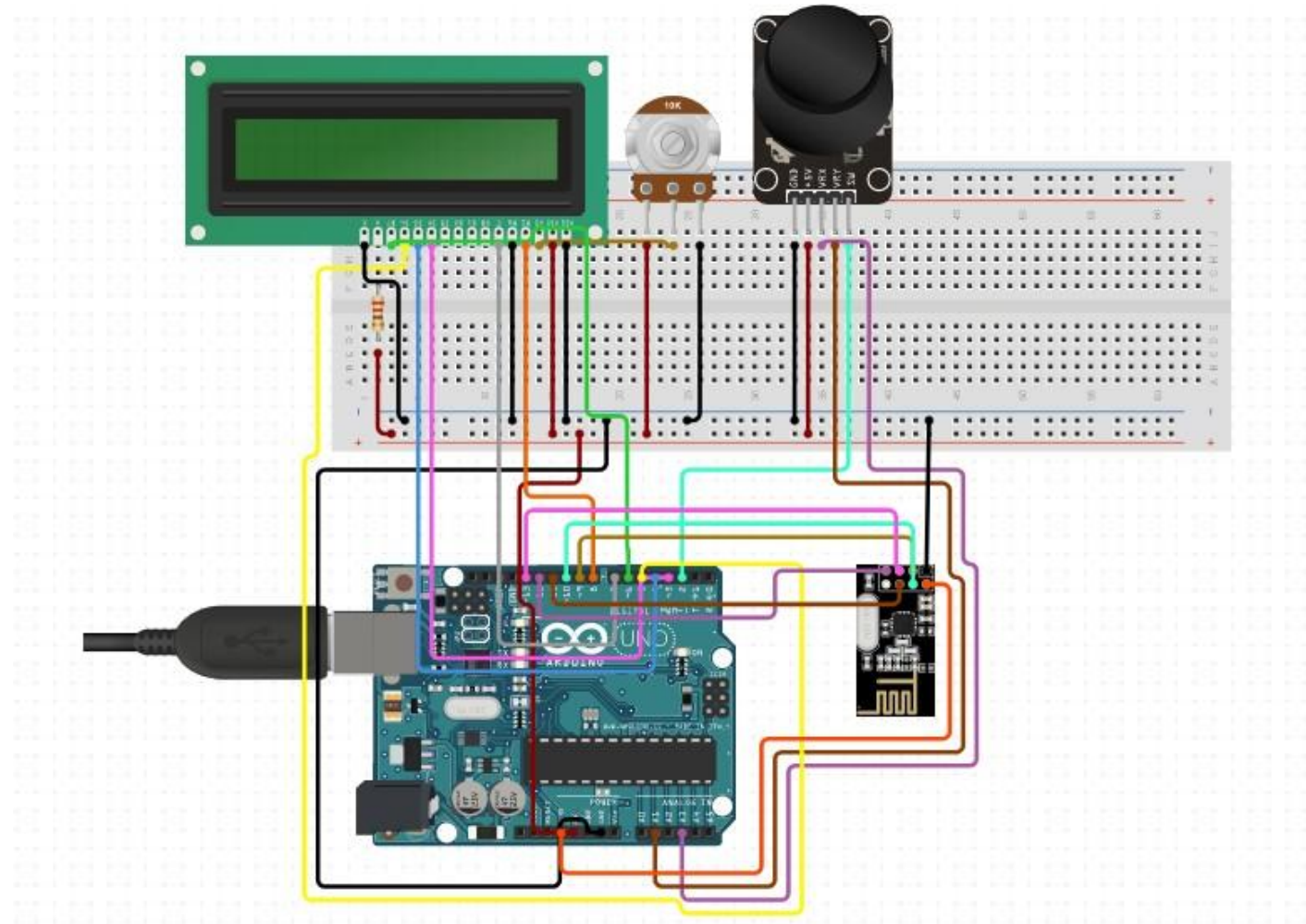
Use case Diagram



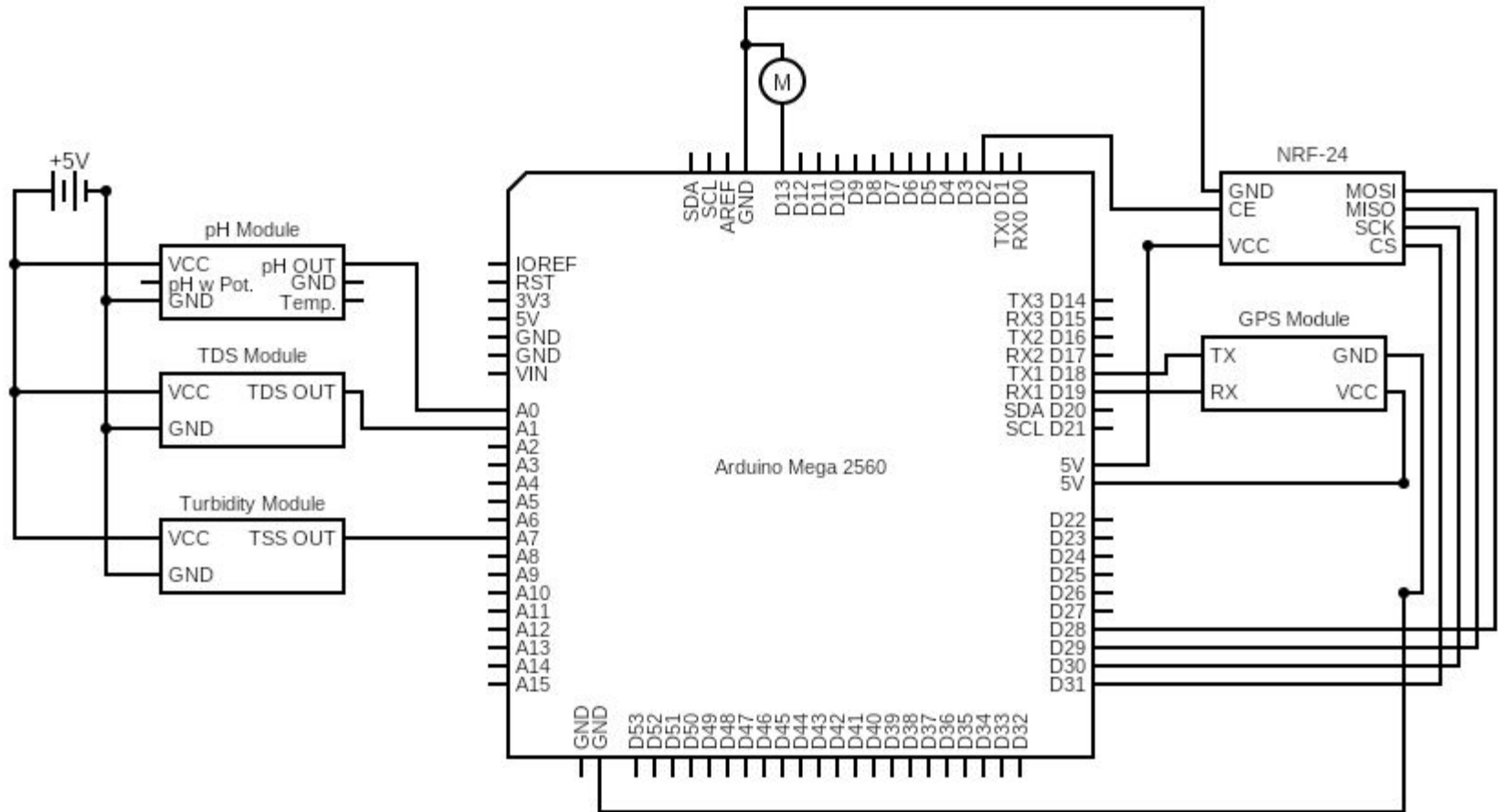
Sequence Diagram



Circuit Diagram



Circuit Diagram



Remote Control Boat Circuit

Technology to be Use

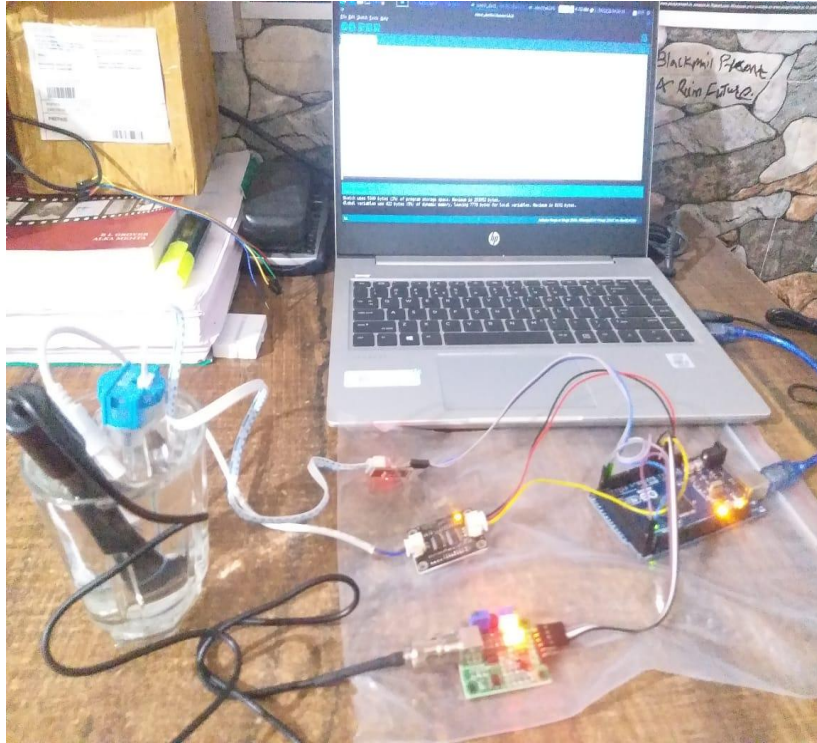
- **Front End:**
LCD Screen Or Arduino Serial Monitor(while testing & Troubleshooting).
- **Back End:**
C, C++.
- **Library/API/Framework:**
TinyGPSPlus.h, SoftwareSerial.h, LiquidCrystal.h, SPI.h, nRF24L01.h, RF24.h.
- **Hardware used:**
NRF24L01, Arduino Mega, NEO-6M GPS, Turbidity Sensor with Module, Analog PH Sensor Kit & TDS Sensor kit.
- **Training Environment:**
Arduino IDE, GitHub.

Proposed Work

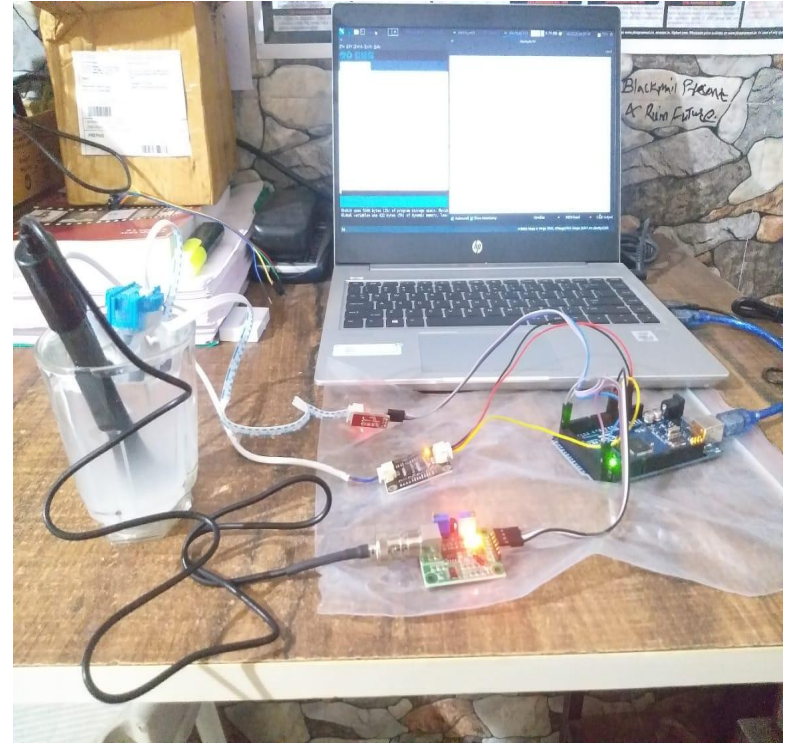
- Module 1: Sensors Testing & Study.
- Module 2: Integration with Remote Control(RC) System.
- Module 3: Manufacturing RC Boat & Remote Control.

Screenshots

MODULE 1



Without impurities



With Impurities

Screenshots

The screenshot displays the Arduino IDE interface. The left pane shows a sketch named 'sketch_dec06a' with C++ code for a water quality sensor. The right pane shows the serial monitor output for the device '/dev/ttyACM0' at 9600 baud. The code defines pins for Turbidity (A3) and TDS (A0), sets calibration values, and implements a median filter for the TDS sensor. The serial monitor shows a continuous stream of data including timestamps, pH values, TDS values in ppm, and Turbidity values.

```
File Edit Sketch Tools Help
sketch_dec06a
int sensorPin = A3; // Turbidity

float calibration = 25.20; //change this value to calibra
const int analogInPin = A1; // pH
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10],temp;

#define TdsSensorPin A0 // TDS
#define VREF 5.0 // analog reference vol
#define SCOUNT 30 // sum of sample point
int analogBuffer[SCOUNT]; // store the analog val
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0,copyIndex = 0;
float averageVoltage = 0,tdsValue = 0,temperature = 25;
int getMedianNum(int bArray[], int iFilterLen)
{
    int bTab[iFilterLen];
    for (byte i = 0; i<iFilterLen; i++)
        bTab[i] = bArray[i];
    int i, j, bTemp;
    for (j = 0; j < iFilterLen - 1; j++)
    {
        for (i = 0; i < iFilterLen - i - 1; i++)
            if (bTab[i] > bTab[i+1])
                swap(bTab[i], bTab[i+1]);
    }
    return bTab[j];
}

Done uploading.
Sketch uses 5340 bytes (2%) of program storage space. Maximum is 65536 bytes.
Global variables use 422 bytes (5%) of dynamic memory, leaving 1818 bytes for local variables. Maximum is 2048 bytes.

09:58:49.323 -> pH Value = 14.52
09:58:49.823 -> TDS Value:352ppm
09:58:54.820 -> Turbidity Value : 36
09:58:55.220 -> pH Value = 13.97
09:58:55.720 -> TDS Value:352ppm
09:59:00.713 -> Turbidity Value : 35
09:59:01.113 -> pH Value = 14.01
09:59:01.613 -> TDS Value:352ppm
09:59:06.613 -> Turbidity Value : 35
09:59:07.012 -> pH Value = 13.69
09:59:07.512 -> TDS Value:352ppm
09:59:12.509 -> Turbidity Value : 32
09:59:12.912 -> pH Value = 13.69
09:59:13.445 -> TDS Value:352ppm
09:59:18.444 -> Turbidity Value : 32
09:59:18.844 -> pH Value = 13.43
09:59:19.344 -> TDS Value:352ppm
09:59:24.341 -> Turbidity Value : 32
09:59:24.741 -> pH Value = 13.15
09:59:25.241 -> TDS Value:352ppm
09:59:30.239 -> Turbidity Value : 32
09:59:30.638 -> pH Value = 12.95
09:59:31.138 -> TDS Value:352ppm
09:59:36.135 -> Turbidity Value : 32
09:59:36.568 -> pH Value = 12.50
09:59:37.068 -> TDS Value:352ppm
09:59:42.067 -> Turbidity Value : 32
09:59:42.467 -> pH Value = 12.48
09:59:42.967 -> TDS Value:352ppm
09:59:47.965 -> Turbidity Value : 32
09:59:48.365 -> pH Value = 12.30
09:59:48.864 -> TDS Value:357ppm

Autoscroll Show timestamp Newline 9600 baud Clear output
54 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/ttyACM0
```

Arduino Serial Monitor (With impurities)

Screenshots

The screenshot displays the Arduino IDE interface. The left pane shows a sketch named 'sketch_dec06a' with C++ code for a water quality sensor. The right pane shows the serial monitor output for the '/dev/ttyACM0' port, displaying real-time sensor data including pH, TDS, and Turbidity values with timestamps. The status bar at the bottom indicates the board is an 'Arduino Mega or Mega 2560, ATmega2560 (Mega 2560)' on the '/dev/ttyACM0' port.

```
File Edit Sketch Tools Help
sketch_dec06a
int sensorPin = A3; // Turbidity

float calibration = 25.20; //change this value to calibra
const int analogInPin = A1; // pH
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10],temp;

#define TdsSensorPin A0 // TDS
#define VREF 5.0 // analog reference vol
#define SCOUNT 30 // sum of sample point
int analogBuffer[SCOUNT]; // store the analog val
int analogBufferIndex = 0,copyIndex = 0;
float averageVoltage = 0,tdsValue = 0,temperature = 25;
int getMedianNum(int bArray[], int iFilterLen)
{
    int bTab[iFilterLen];
    for (byte i = 0; i<iFilterLen; i++)
        bTab[i] = bArray[i];
    int i, j, bTemp;
    for (j = 0; j < iFilterLen - 1; j++)
    {
        for (i = 0; i < iFilterLen - j - 1; i++)
            if (bTab[i] > bTab[i+1])
            {
                bTemp = bTab[i];
                bTab[i] = bTab[i+1];
                bTab[i+1] = bTemp;
            }
    }
    return bTab[j];
}

Done uploading.

Sketch uses 5340 bytes (2%) of program storage space. Maximum is 65536 bytes.
Global variables use 422 bytes (5%) of dynamic memory, leaving 1818 bytes for local variables.

09:57:03.029 -> pH Value = 11.40
09:57:03.529 -> TDS Value:0ppm
09:57:08.526 -> Turbidity Value : 58
09:57:08.926 -> pH Value = 11.49
09:57:09.425 -> TDS Value:0ppm
09:57:14.421 -> Turbidity Value : 58
09:57:14.821 -> pH Value = 11.58
09:57:15.321 -> TDS Value:0ppm
09:57:20.321 -> Turbidity Value : 58
09:57:20.721 -> pH Value = 11.54
09:57:21.221 -> TDS Value:0ppm
09:57:26.221 -> Turbidity Value : 58
09:57:26.621 -> pH Value = 11.85
09:57:27.121 -> TDS Value:0ppm
09:57:32.153 -> Turbidity Value : 58
09:57:32.553 -> pH Value = 11.74
09:57:33.052 -> TDS Value:0ppm
09:57:38.051 -> Turbidity Value : 58
09:57:38.451 -> pH Value = 11.95
09:57:38.951 -> TDS Value:0ppm
09:57:43.951 -> Turbidity Value : 58
09:57:44.351 -> pH Value = 11.66
09:57:44.851 -> TDS Value:0ppm
09:57:49.847 -> Turbidity Value : 58
09:57:50.247 -> pH Value = 12.00
09:57:50.744 -> TDS Value:0ppm
09:57:55.744 -> Turbidity Value : 58
09:57:56.177 -> pH Value = 11.73
09:57:56.676 -> TDS Value:0ppm
09:58:01.675 -> Turbidity Value : 58
09:58:02.073 -> pH Value = 12.05
09:58:02.571 -> TDS Value:0ppm

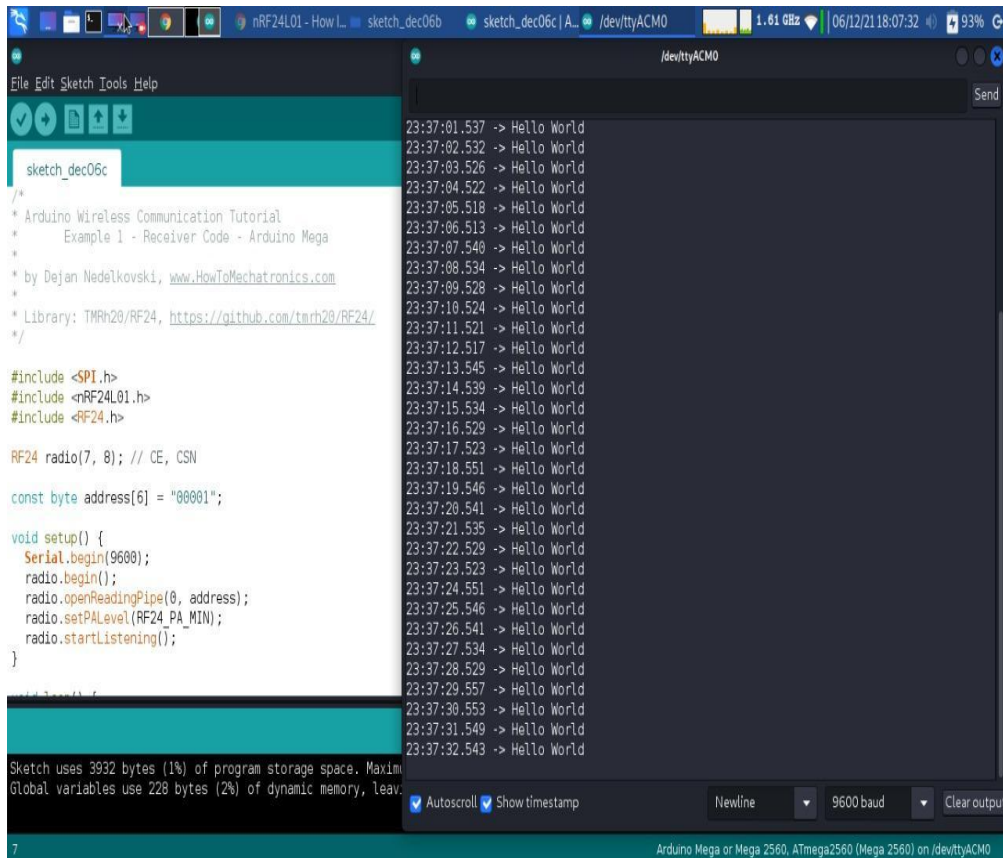
Autoscroll Show timestamp Newline 9600 baud Clear output

54 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/ttyACM0
```

Arduino Serial Monitor (Without impurities)

Screenshots

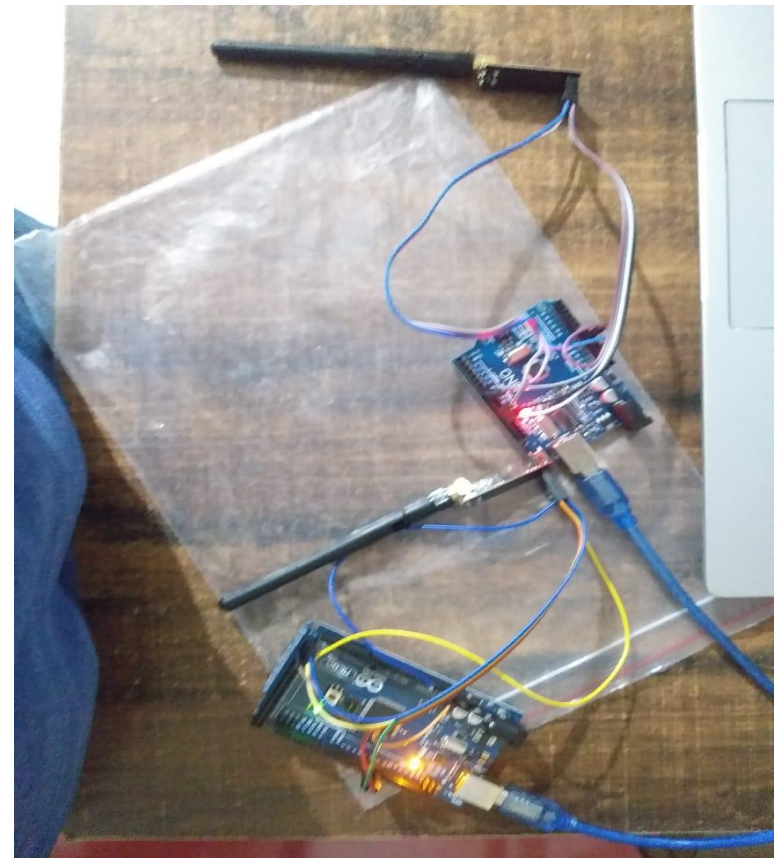
MODULE 2



The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the following code:

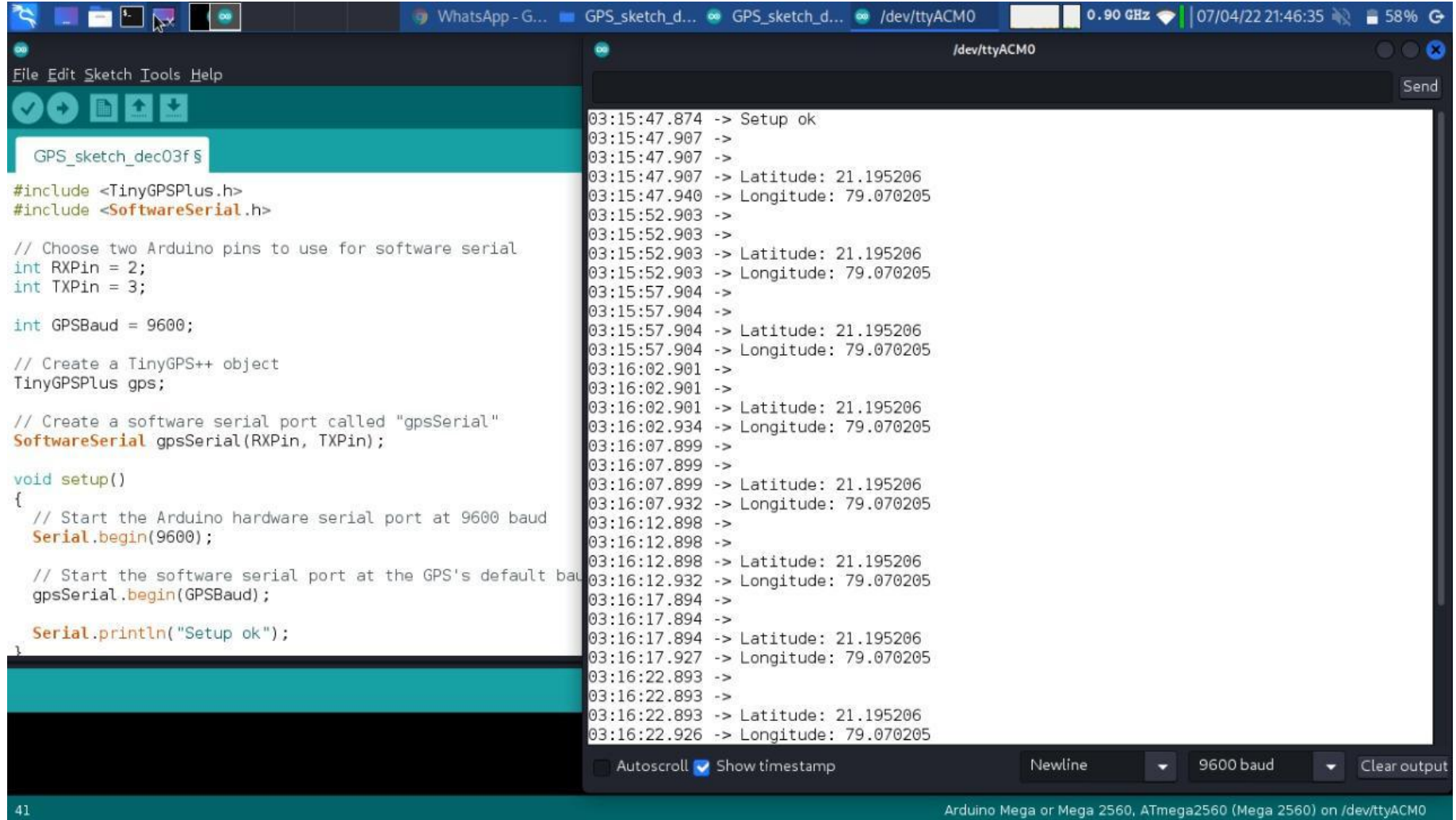
```
/*  
 * Arduino Wireless Communication Tutorial  
 * Example 1 - Receiver Code - Arduino Mega  
 * by Dejan Nedelkovski, www.HowToMechatronics.com  
 * Library: TMRh20/RF24, https://github.com/tmrh20/RF24/  
 */  
  
#include <SPI.h>  
#include <RF24L01.h>  
#include <RF24.h>  
  
RF24 radio(7, 8); // CE, CSN  
  
const byte address[6] = "000001";  
  
void setup() {  
  Serial.begin(9600);  
  radio.begin();  
  radio.openReadingPipe(0, address);  
  radio.setPALevel(RF24_PA_MIN);  
  radio.startListening();  
}  
  
void loop() {  
  if (radio.available()) {  
    char message[16];  
    radio.read(&message, sizeof(message));  
    Serial.println(message);  
  }  
}
```

The serial monitor on the right displays a continuous stream of "Hello World" messages, each preceded by a timestamp. The status bar at the bottom indicates "Sketch uses 3932 bytes (1%) of program storage space. Maximum Global variables use 228 bytes (2%) of dynamic memory, leaving 1024 bytes free." and "Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/ttyACM0".



Arduino serial Monitor (Transmitter and Receiver Module)

Screenshots



The screenshot displays the Arduino IDE interface. The left pane shows the sketch code for 'GPS_sketch_dec03f.s', which includes TinyGPSPlus and SoftwareSerial libraries, sets pins 2 and 3 for RX and TX, and configures a 9600 baud rate. The right pane shows the serial monitor output for '/dev/ttyACM0', displaying a series of timestamps and coordinates. The status bar at the bottom indicates the board is an 'Arduino Mega or Mega 2560, ATmega2560 (Mega 2560)' on '/dev/ttyACM0'.

```
File Edit Sketch Tools Help
GPS_sketch_dec03f.s
#include <TinyGPSPlus.h>
#include <SoftwareSerial.h>

// Choose two Arduino pins to use for software serial
int RXPin = 2;
int TXPin = 3;

int GPSBaud = 9600;

// Create a TinyGPS++ object
TinyGPSPlus gps;

// Create a software serial port called "gpsSerial"
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup()
{
  // Start the Arduino hardware serial port at 9600 baud
  Serial.begin(9600);

  // Start the software serial port at the GPS's default baud
  gpsSerial.begin(GPSBaud);

  Serial.println("Setup ok");
}

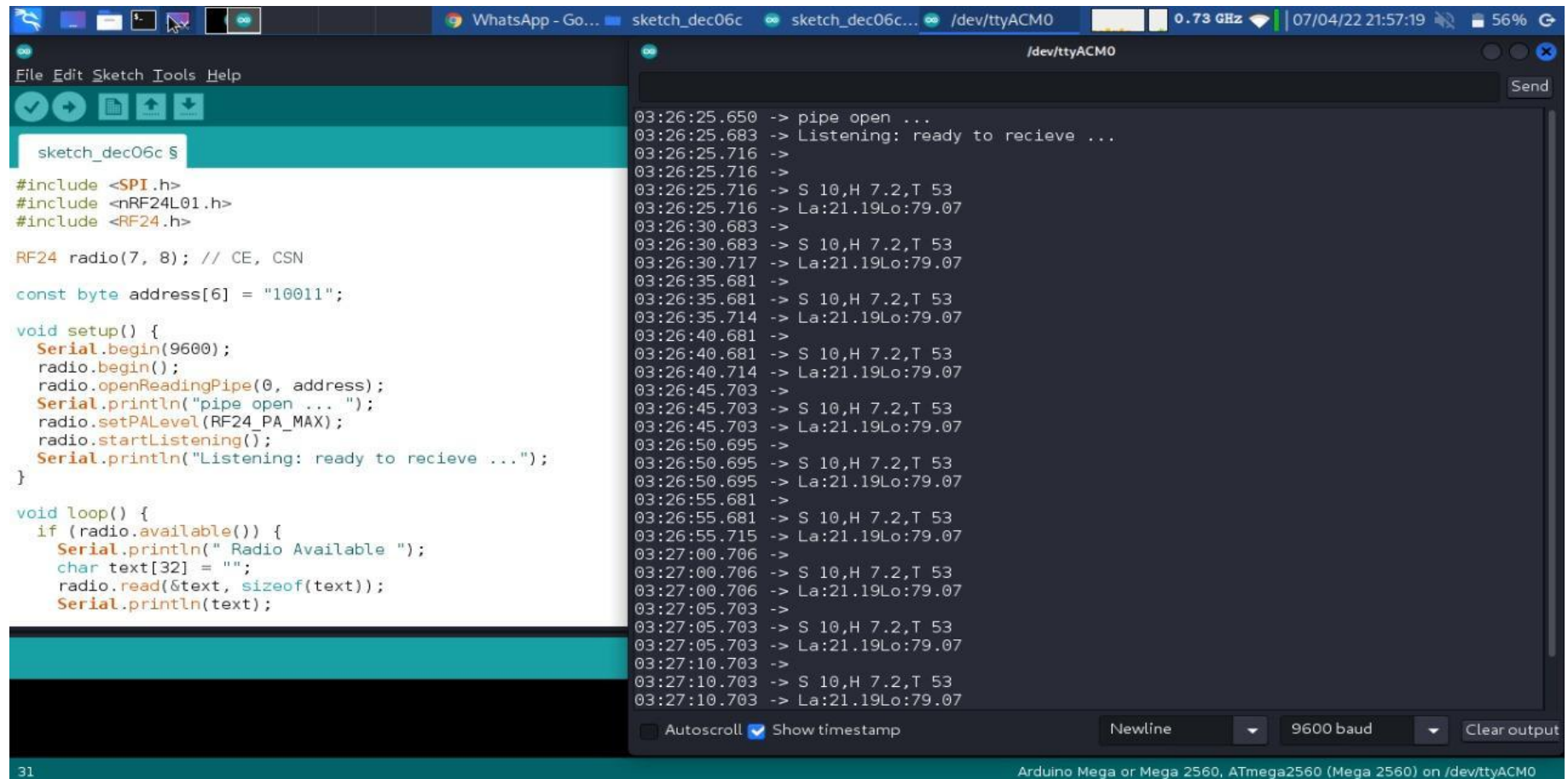
03:15:47.874 -> Setup ok
03:15:47.907 ->
03:15:47.907 ->
03:15:47.907 -> Latitude: 21.195206
03:15:47.940 -> Longitude: 79.070205
03:15:52.903 ->
03:15:52.903 ->
03:15:52.903 -> Latitude: 21.195206
03:15:52.903 -> Longitude: 79.070205
03:15:57.904 ->
03:15:57.904 ->
03:15:57.904 -> Latitude: 21.195206
03:15:57.904 -> Longitude: 79.070205
03:16:02.901 ->
03:16:02.901 ->
03:16:02.901 -> Latitude: 21.195206
03:16:02.934 -> Longitude: 79.070205
03:16:07.899 ->
03:16:07.899 ->
03:16:07.899 -> Latitude: 21.195206
03:16:07.932 -> Longitude: 79.070205
03:16:12.898 ->
03:16:12.898 ->
03:16:12.898 -> Latitude: 21.195206
03:16:12.932 -> Longitude: 79.070205
03:16:17.894 ->
03:16:17.894 ->
03:16:17.894 -> Latitude: 21.195206
03:16:17.927 -> Longitude: 79.070205
03:16:22.893 ->
03:16:22.893 ->
03:16:22.893 -> Latitude: 21.195206
03:16:22.926 -> Longitude: 79.070205

Autoscroll Show timestamp Newline 9600 baud Clear output
41 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/ttyACM0
```

Arduino serial Monitor (GPS Module)

Screenshots

MODULE 3



The screenshot displays the Arduino IDE interface. The left pane shows a sketch named 'sketch_dec06c' with the following code:

```
File Edit Sketch Tools Help
sketch_dec06c $
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(7, 8); // CE, CSN

const byte address[6] = "10011";

void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
  Serial.println("pipe open ... ");
  radio.setPALevel(RF24_PA_MAX);
  radio.startListening();
  Serial.println("Listening: ready to recieve ...");
}

void loop() {
  if (radio.available()) {
    Serial.println(" Radio Available ");
    char text[32] = "";
    radio.read(&text, sizeof(text));
    Serial.println(text);
  }
}
```

The right pane shows the serial monitor output for the device '/dev/ttyACM0'. The output consists of timestamped log messages:

```
03:26:25.650 -> pipe open ...
03:26:25.683 -> Listening: ready to recieve ...
03:26:25.716 ->
03:26:25.716 ->
03:26:25.716 -> S 10,H 7.2,T 53
03:26:25.716 -> La:21.19Lo:79.07
03:26:30.683 ->
03:26:30.683 -> S 10,H 7.2,T 53
03:26:30.717 -> La:21.19Lo:79.07
03:26:35.681 ->
03:26:35.681 -> S 10,H 7.2,T 53
03:26:35.714 -> La:21.19Lo:79.07
03:26:40.681 ->
03:26:40.681 -> S 10,H 7.2,T 53
03:26:40.714 -> La:21.19Lo:79.07
03:26:45.703 ->
03:26:45.703 -> S 10,H 7.2,T 53
03:26:45.703 -> La:21.19Lo:79.07
03:26:50.695 ->
03:26:50.695 -> S 10,H 7.2,T 53
03:26:50.695 -> La:21.19Lo:79.07
03:26:55.681 ->
03:26:55.681 -> S 10,H 7.2,T 53
03:26:55.715 -> La:21.19Lo:79.07
03:27:00.706 ->
03:27:00.706 -> S 10,H 7.2,T 53
03:27:00.706 -> La:21.19Lo:79.07
03:27:05.703 ->
03:27:05.703 -> S 10,H 7.2,T 53
03:27:05.703 -> La:21.19Lo:79.07
03:27:10.703 ->
03:27:10.703 -> S 10,H 7.2,T 53
03:27:10.703 -> La:21.19Lo:79.07
```

The bottom status bar indicates the hardware is an 'Arduino Mega or Mega 2560, ATmega2560 (Mega 2560)' connected to '/dev/ttyACM0'.

Arduino serial Monitor (Receiver Module)

Advantages & Applications

- **Advantages:-**

- Ease and convenience of usage
- Instantaneous data
- Improved accuracy of measurements
- Cost saving

- **Applications:-**

- It can be used to survey the pollution levels of Large Water bodies.
- It can be used to enumerate the water pollution levels near the industrial as an alert system for high pollution levels.

Plan of Work

Work	Time (Days)	Status
Requirement Gathering (Project)	10	Done
Analysis (Project)	5	Done
Requirement gathering & Analysis (Module1)	5	Done
Implementation (Module1)	10	Done
Testing (Module1)	2	Done
Requirement gathering & Analysis(Module2)	2	Done
Implementation (Module2)	10	Done
Testing (Module2)	3	Done
Report Generation & Submission	7	Done
Requirement gathering & Analysis(Module3)	3	Done
Implementation (Module3)	20	Done
Testing (Module3)	5	Done
Report Generation & Submission	10	Done

References

❑ Papers:

- [1] Hong, Wong J., Norazanita Shamsuddin, Emeroylariffion Abas, Rosyzie A. Apong, Zarifi Masri, Hazwani Suhaimi, Stefan H. Gödeke, and Muhammad N.A. Noh 2021. "Water Quality Monitoring with Arduino Based Sensors" *Environments* 8, no. 1: 6.
- [2] Geetha, S., Gouthami, S. Internet of things enabled real time water quality monitoring system. *Smart Water* 2, 1 (2016).
- [3] Nikhil Kedia, Water Quality Monitoring for Rural Areas- A Sensor Cloud Based Economical Project, in 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2015. 978-1-4673-6809-4/15/\$31.00 ©2015 IEEE
- [4] Md. Omar Faruq, Injamamul Hoque Emu, Md. Nazmul Haque¹, Maitry Dey, N.K. Das, Mrinmoy Dey Design and implementation of a cost-effective water quality evaluation system IEEE Region 10 Humanitarian Technology Conference, Dhaka, Bangladesh (2017), pp. 860-863
- [5] A Catalyst for Water Quality Improvements: (U.S. Environmental Protection Agency [EPA]. National Nonpoint Source Program, 2 Nov , 2020)

- Websites:

[1]<https://www.nrdc.org/stories/water-pollution-everything-you-need-know>

[2]https://www.epa.gov/sites/production/files/2016-10/documents/nps_program_highlights_report-508.pdf

[3]<https://borgenproject.org/water-quality-brunei/>

[4]<http://water.usgs.gov/edu/turbidity.html>

Published Papers/Achievements

LIST OF PUBLICATION/PARTICIPATIONS

Sr. No.	Title	Event Name / JournalName/ Conference	Date	Remark
1	Development of WaterPollution Monitoring System	International Journal Of Advanced Research In Science, CommunicationAnd Technology (IJARSCT)	01/01/22	Participated
2	Development of WaterPollution Monitoring System	International Conference On Tech Trends In Science& Engineering (ICTTSE)	25/02/22	Participated

Links:

GitHub:<https://github.com/MasterSaha/Development-of-Water-Pollution-Monitoring-System>

Published Paper: <https://doi.org/10.48175/IJARSCT-2268>

Thank You