ONLINE WATER QUALITY MONITORING SYSTEM USING IOT

A PROJECT REPORT

Submitted by

JANANI.P (110318106014)

THAMILARASI.C (110318106041)

SHANMUGA PRIYA.K (110318106702)

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY

TIRUTTANI

ANNA UNIVERSITY: CHENNAI 600 025

JUNE-2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**ONLINE WATER QUALITY MONITORING SYSTEM USING IOT**" is the Bonafide work of **"JANANI.P (110318106014), THAMILARASI.C (110318106041), and SHANMUGA PRIYA.K (110318106702)"** who carried out the project work under my supervision.

SIGNATURE
Dr.P.SIVAKUMAR, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor
Electronics & Communication Engineering
GRT Institute of Engineering & Technology
Tiruttani.

SIGNATURE
Mr.S.SENTHILKUMAR,M.TECH.,(Ph.D.),
SUPERVISOR
Associate Professor
Electronics &Communication Engineering
GRT Institute of Engineering & Technology
Tiruttani.

Submitted for the Project Viva-Voce examination held on _____

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

First and foremost we place this project work on the feet of **GOD ALMIGHTY** and **OUR PARENTS** who is the power of strength in each step of progress towards the successful completion of our project.

Our sincere thanks to honorable chairman, **Shri.G.RAJENDRAN** and Managing Director **Shri.G.R.ANANTHAPADMANABHAN** for creating a wonderful atmosphere inside the campus.

We express our thanks to our Dean, **Dr.K.S.MEENAKSHISUNDARAM, Ph.D.**(MANAGEMENT&AGRI.EXT),**M.B.A.**(HR&MARKETING),**M.Sc.**(AGRI .EXT), **PGDBM, PGDJMC, PGDTD, DST.,** for his great support and blessings.

We express our thanks to our principal, **Dr.S.ARUMUGAM, M.E., Ph.D.,** and our Administrative officer **Mr.K.SASIKUMAR** who helped us in providing required facilities to complete this project.

We like to express our deep gratitude to **Dr.P.SIVAKUMAR, M.E., Ph.D.,** Professor, Head of the Department of Electronics and Communication Engineering, for giving the opportunity and facilities to complete our work in time.

We like to express our special gratitude to the project coordinator **Mr.C.E.MOHANKUMAR, M.E.,** Assistant professor, Department of Electronics and Communication Engineering, for his kind co-operation and guidance in doing our project work.

We would like to express our deep sense of profound gratitude and thanks to our project supervisor **Mr. S. SENTHILKUMAR, M.TECH., (Ph.D.),** Associate Professor, Department of Electronics and Communication Engineering, for his enthusiastic guidance, suggestions and constant encouragement in completing project. Finally, we thank our faculty members and friends for their moral support and valuable suggestions in this project.

ABSTRACT

Water pollution is one of the biggest fears for the green globalization. In order to ensure the purity of the water the quality needs to be monitored in real time. Smart solutions for monitoring water pollutions are getting more and more significant these days with sensors, communication, and internet of things (IoT) Technology. The proposed system consists of several sensors which is used to measure physical and chemical characteristics of the water. The parameters such as temperature, PH, Air Quality and Level of the water can be measured using sensors and monitored using Arduino Uno Microcontroller. Finally, the user gets to see the above-mentioned parameters in the mobile application.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TCP/IP	Transmission Control Protocol/Internet Protocol
SWQM	Smart Water Quality Monitoring
WQM	Water Quality Monitoring
WSN	Wireless Sensor Networks
USB	Universal Serial Bus
IOT	Internet of Things
TDS	Total Dissolved Solids
LCD	Liquid Crystal Display
PCB	Printed Circuited Board
pH	Potential of Hydrogen
DO	Dissolved Oxygen

CHAPTER 1

INTRODUCTION

1.1. WATER QUALITY:

Fresh water is a world resource that is a gift of a nature and important to farming, manufacturing and domestical uses on the Earth. Currently water faces new real-world problems. The high use of chemicals in the manufacturing, constructions, fertilizers in the farms and also directly discharged polluted water from industries into the nearby water bodies have made a huge contribution to the global water quality reduction, which has become an important problem.

When lethal materials move into the water sources like ponds, rivers and lakes, gets dissolved and suspends in the water or gets deposited on the bed. Pollution will degrade the quality and purity of water. Ensuring pure and safer water is really challenging due to undue sources of chemicals and contaminates. Pollution of water can be numerous ways; one of the main reasons for pollution industrial discharges and city sewage. Secondary source of the pollution are pollutants that enter the water from the soils or from atmosphere via rain or from ground water system.

1.2. OVERVIEW OF WATER QUALITY MONITORING SYSTEM:

In the 21st century, there were lots of inventions, but at the same time were pollutions, global warming and so on is being formed. Because of this there is no good quality of water. Water quality monitoring in real-time faces challenges because of global warming, limited water resources and growing population etc. So there is a need for developing better methodologies to monitor the water quality parameters in real-time. The water quality parameters pH, temperature, turbidity,

dissolved oxygen and other parameters should be monitored carefully for maintaining the quality of water. Some remote sensing systems have also been developed for monitoring ambient water quality in riverine, estuarine and coastal water bodies.

1.3. MOTIVATION BEHIND THE PROJECT:

The traditional method for monitoring of the water quality was done by taking water samples and sent to the laboratory to be tested manually by analytical methods. Although by this method the chemical, physical, and biological agents of the water can be analysed, it has several drawbacks.

- Firstly, it is a time consuming and labour intensive.
- Secondly, the cost for this is controlled.

Compared to the water quality testing techniques, sensor-based water quality testing has many advantages such as accurate, high sensitivity, good selectivity, speed, fast response and low cost etc.

1.4. ORGANISATION OF THIS REPORT:

Remaining of this report is organised as follows. In chapter 2, Literature Survey we discuss about the related works and problems formed from these methods are discussed. In chapter 3, Water Quality Monitoring System we discuss about the prototype of the proposed system. In chapter 4, System Analysis we discuss about the design and implementation of the proposed system. In chapter 5, Results and Discussion we discuss about the outcome of this project such as prototype of the proposed system and mobile application to the water quality parameters. Finally, in chapter 6 conclusion and future work we discuss about our work and also discuss about extensions of the project with monitoring some more parameters.

CHAPTER 2
LITERATURE SURVEY

A literature survey provides a description, summary, and critical evaluation of the works in relation to the research problem being investigated. It also provides background knowledge about the topic and allows us to give evidence as to why the current research is being completed. In this chapter we have reviewed some of the recently published article about water quality monitoring and formulation of problem statement from the drawbacks of the existing system.

The authors SathishPasika, Sai TejaGandlahave published the paper **"SMART WATER QUALITY MONITORING SYSTEM WITH COST-EFFECTIVE USING IOT"** in the year 2021.

The proposed system consists of several sensors to measure various parameters such as pH value, the turbidity in the water, level of water in the tank, temperature and humidity of the surrounding atmosphere. And also, the Microcontroller Unit (MCU) interfaced with these sensors and further processing is performed at Personal Computer (PC). The obtained data is sent to the cloud by using IoT based ThinkSpeak application to monitor the quality of the water.

The authors VarshaLakshmikantha, AnjithaHiriyannagowd, AkshayManjunath, Aruna Patted, JagadeeshBasavaiah, Audre Arlene Anthony have published the paper **"IOT BASED SMART WATER QUALITY MONITORING SYSTEM"** in the year 2021.

The proposed system gives us a detailed review of the latest works that were implemented in the arena of smart water pollution monitoring systems is presented.

This paper proposes a cost effective and efficient IoT based smart water quality monitoring system which monitors the quality parameters uninterruptedly. The developed model is tested with three water samples and the parameters are transmitted to the cloud server for further action.

The authors G. NitishSatya Sai, Reddy Sudheer, Kondreddy Sai Manikanta, Sri Ganesh ArjulaBandiNarasimha Rhave published the paper **"IOT BASED WATER QUALITY MONITORING SYSTEM"** in the year 2021.

The proposed system measures five water parameters like Potential of Hydrogen (pH), Total Dissolved Solids (TDS), Temperature, Turbidity and Flow Rate. All this data is transmitted to the Amazon Web Services (AWS) cloud platform hosted in Firebase, and made available to both the admin dashboard and User mobile application (App). If any deviations occur in water quality, then the user will get the notification and he/she can take a precise decision. Every end-user is provided with a mobile app that engages them to see their usage and water quality. Users can be charged based on the usage of water and the app provides them the facility to pay for the water used and the admin can control all the Smart Valves in a grid when needed. Admin Dashboard consists of all the functionalities required to manage users, identify leakages, and check historical data.

The authors Halima Haque, KashshafLabeeb, RabeaBasriRiha, Md. Nasfikur R. Khanhave published the paper **"IOT BASED WATER QUALITY MONITORING SYSTEM BY USING ZIGBEE PROTOCOL"** in the year 2021.

The proposed system says about the damages caused by water and what can be done to resolve those issues by involving the Internet of things (IoT). The use of IoT based solution, focused mainly on water quality monitoring. In order to

support the issue, an IoT-based water quality checking network has been introduced that continuously monitors and evaluates the quality of water and tries to distinguish whether it is up to the mark for general use. This also includes the use of specific sensors that calculates the various parameters of the quality of water which includes conductivity and dissolved oxygen (DO), turbidity, pH, and temperature. The values from the sensors have been measured and calculated using the microcontrollers. Then these processed remote values have been transmitted to the raspberry pi, the central controller which uses the Zigbee protocol. Lastly, all the data from the sensors are then accessible via cloud computing through any browser, on request.

The authors AjithJerom B., R. Manimegalai, R. Manimegalai have published the paper **"AN IOT BASED SMART WATER QUALITY MONITORING SYSTEM USING CLOUD"** in the year 2020.

In proposed system shows an IoT Based Smart Water Quality Monitoring System using Cloud and Deep Learning is proposed to monitor the quality of the water in water-bodies. The design and development of a low-cost system for real-time monitoring of water quality using the Internet of Things (IoT) is essential. The system monitors the quality of water relentlessly with the help of IoT devices, such as, NodeMCU. The in-built Wi-Fi module is attached in NodeMCU which enables internet connectivity transfers the measured data from sensors to the Cloud. The prototype is designed in such a way that it can monitor the number of pollutants in the water. The results are stored in the Cloud, deep learning techniques are used to predict whether the water suitable or not.

The authors YashwanthGowda K. N, Vishali C, Sumalatha S.J And Spoorth G.B have published the paper **"REAL-TIME WATER QUALITY MONITORING SYSTEM"** in the year 2020.

The main goal of this system is to build a Sensor- based Water Quality Monitoring System. pH, Temperature and Turbidity sensors, Arduino board technique is very high due to the operation cost, labor cost and equipment cost, and it is difficult to make critical decisions in the real time. Sensor is an ideal detecting device which can convert non-power information to electrical signals which can easily be processed or transformed.

The authors MoniraMukta, Samia Islam, Surajit Das Barman, Ahmed Wasif Reza, M Saddam Hossain Khanhave published the paper **"IOT BASED SMART WATER QUALITY MONITORING SYSTEM"** in the year 2019.

This system represents an IoT (Internet of things) based smart water quality monitoring (SWQM) system that aids in continuous measurement of water condition based on four physical parameters i.e., temperature, pH, electric conductivity and turbidity properties. Four sensors are connected with Arduino-uno in discrete way to detect the water parameters. The extracted data from the sensors are transmitted to a desktop application developed in NET platform and compared with the WHO (World Health Organization) standard values. Based on the measured result, the proposed SWQM system can successfully analyze the water parameters using fast forest binary classifier to classify whether the test water sample is drinkable or not.

The authors Nageswara Rao Moparthi, Ch. Mukesh, P. VidyaSagarhave published the paper **"WATER QUALITY MONITORING SYSTEM USING IOT"** in the year 2018.

The objective of this water quality monitoring system using internet of things is to find the quality of the water i.e., how the pH content varies and sending message to the corresponding authorities. They have implemented this project at municipal water tanks and drinking water reservoir. For that they are using an Arduino board for finding pH value and GSM module for message technique. They have also used a led display to have continuous observation on water parameters. Finally, the user gets message of pH value of water further they extend this project by sending the sensor data to cloud for global monitoring of water quality.

2.1. PROBLEM FORMULATION:

We find that the water quality monitoring is difficult from this survey. The water quality results take longer time to evaluate the result. And the monitoring of water is done manually which is difficult and also time-consuming process. It leads to spread of water diseases. The values of parameters are not monitored in the mobile application and cannot be accessed anytime and anywhere.

CHAPTER 3
WATER QUALITY MONITORING SYSTEM

In this chapter we discuss how the proposed water quality monitoring system overcomes the problem identified with the existing systems. In the proposed system we implement multiple sensors for monitoring the quality of water automatically with accurate details of the water quality.

3.1. PROPOSED WATER QUALITY MONITORING SYSTEM:

Water plays an important role for all the living beings. So, the physical, chemical and biological characteristics of water are monitored. The parameters like pH, temperature and the quality of air where the water is present are monitored. The normal pH value of water is 7.0-7.5. If the pH value is more than 7 then the water is alkaline. If the pH value is less than 7 then the water is acidic. The ultrasonic sensor is used to find the depth. The motor driver and pump motor are used for pumping out the water from the place where the above-mentioned parameters are not in normal range. LCD displays the value of the parameters. All the sensors and motor driver are connected to the Arduino UNO microcontroller and ESP8266 Wi-Fi module is used for accessing the parameter values of the system in mobile application named “water quality”.

Fig 3.1 shows the Arduino UNO microcontroller is used to interface with the sensors and to the communication devices. The pH sensor is used to monitor the water pH level like acidity or alkaline. The ultrasonic sensor provides the water level in the tank and also in the sump. The LCD is used to display the updated value from the sensors. The IOT module ESP8266 is used to update the

information of sensors to the cloud. The end result can be monitored in the mobile application.

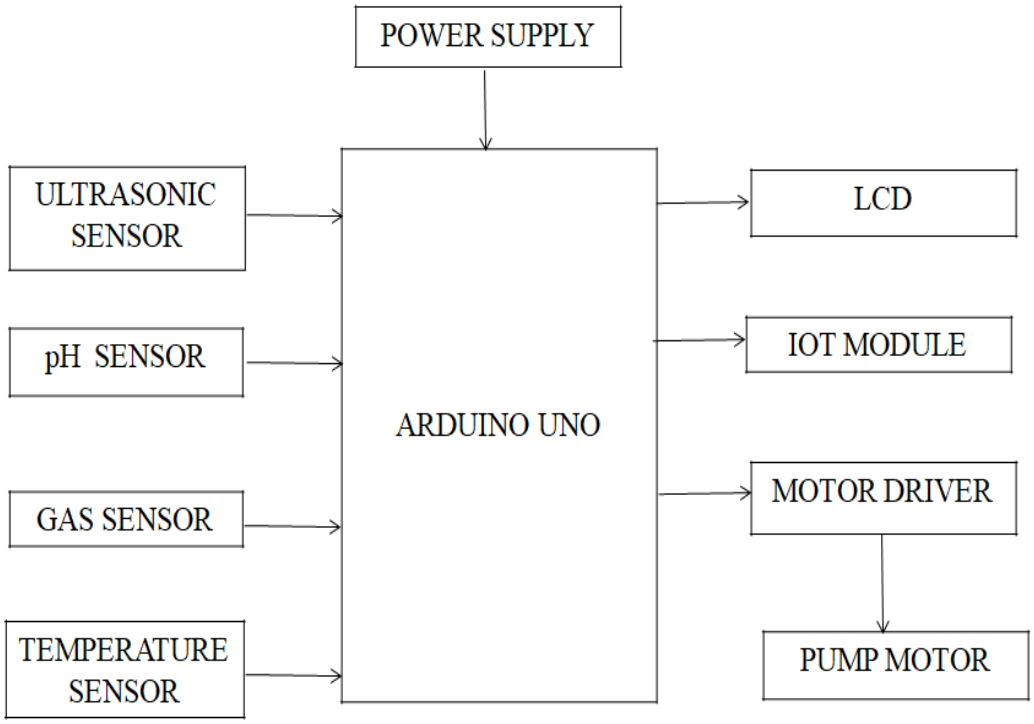

Fig 3.1 – WATER QUALITY MONITORING SYSTEM

The water pump motor is activated based on the pH level by automatically. MQ135 Gas Sensor module is for sensing Air Quality which is having Digital as well as Analog output. For measuring water temperature waterproof temperature sensors are used.

3.2. ADVANTAGES:

- It provides accurate details of the quality parameters of water and it is easy and convenient to use.
- It can be accessed in any place and the monitoring can be done if the corresponding person is not present at the place.

- It helps farmers to monitor the quality of water which is provided to the plants.
- It helps in monitoring the quality of air in the places like well, sump and in industries where large tank is used.
- It helps to monitor the quality of water for aquaculture and also monitors the quality of the air around it.

3.3. REQUIREMENT ANALYSIS:

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

3.4. HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENT:

- Arduino UNO
- ESP8266
- pH Sensor (PE 2.0)
- Ultrasonic Sensor (HC-SR04)
- Gas Sensor (MQ-135)

- Temperature Sensor (DS18B20)
- Motor Driver (L293D)
- Pump Motor
- LCD16x2
- 12v Adapter

SOFTWARE REQUIREMENT:

- Compiler: Arduino IDE
- Language: Embedded C Language

CHAPTER 4

SYSTEM DESIGN

In this chapter we concentrate on the various software and hardware requirements that help us to build the water quality monitoring system efficiently. We discuss about the components used for system design and the design implementation of the proposed system in detail.

4.1. ARDUINO UNO:

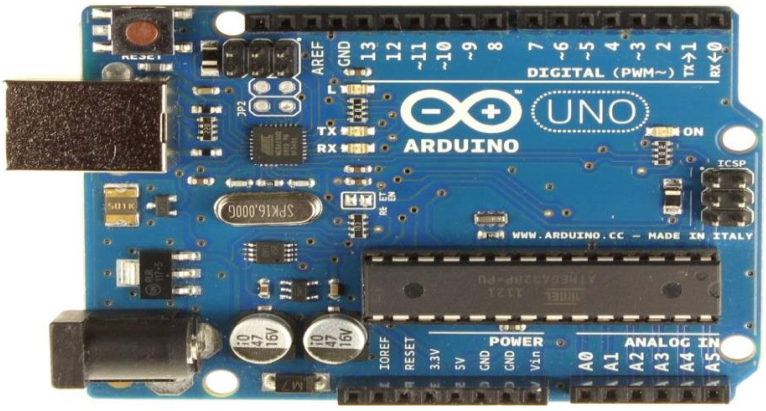

Fig 4.1 – Arduino UNO

As the Fig 4.1 shows it has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards.

Table 4.1 - Technical Specifications of Arduino UNO

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32KB of which 0.5KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

4.2. ESP8266:

ESP8266 (also called ESP8266 Wireless Transceiver) is a cost-effective, easy-to-operate, compact-sized & low-powered **Wi-Fi module**, which supports both TCP/IP and Serial Protocol. It is considered the most widely used Wi-Fi module because of its low cost and small size. It runs at operating voltage of 3V and can handle maximum voltage of around 3.6 V. ESP8266 Wi-Fi module can easily be interfaced with microcontrollers board (i.e. Arduino UNO) via Serial Port. There are numerous breakout boards available based on ESP8266 Wi-Fi Module (i.e. ESP8266 NodeMCU V3).

ESP8266 Pin Diagram:

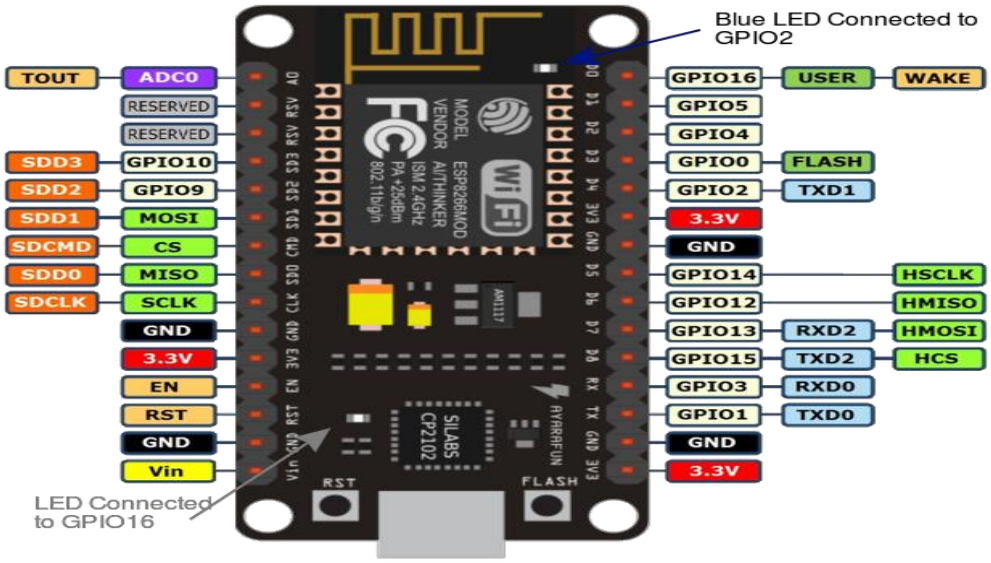

Fig 4.2 – ESP8266 Pin Diagram

From Fig 4.2 **ESP8266 Pin out** which consists of 8 pins in total, which are given in below table along with their operation:

Table 4.2 - Pin Configuration of ESP8266

No.	Pin Name	Working
1	RX	Serial Receiver Pin
2	Vcc	Power Pin (+3.3 V; can handle up to 3.6 V)
3	GPIO 0	General-purpose I/O No. 0
4	RST	Reset
5	CH_PD	Chip power-down
6	GPIO 2	General-purpose I/O No. 2
7	TX	Serial Transmitter Pin
8	GND	Ground

Each pin comes with a specific function associated with it where Vcc and GND are voltage source and ground respectively. RX and TX are used for communication where TX is dedicated for data transmission and RX is used receiving data. It is also known as a system-on-chip (SoC) and comes with a 32-bit Tensilica microcontroller, antenna switches, RF balun, power amplifier, standard digital

peripheral interfaces, low noise receive amplifier, power management module and filter capability. The processor is based on Tensilica Xtensa Diamond Standard 106Micro and runs at 80 MHz. It incorporates 64 KB boot ROM, 80 KB user data RAM and 32 KB instruction RAM. It supports Wi-Fi 802.11 b/g/n around 2.4 GHz and other features including 16 GPIO, Inter-Integrated Circuit (I^2C), Serial Peripheral Interface (SPI), 10-bit ADC, and I^2S interfaces with DMA. External QSPI flash memory is accessed through SPI and supports up to 16 MB and 512 KB to 4 MB is initially included in the module. It is a major development in terms of wireless communication with little circuitry and contains onboard regulator that helps in providing 3.3V consistent power to the board. It supports APSD which makes it an ideal choice for VoIP applications and Bluetooth interfaces.

Applications of ESP8266 Wi-Fi Module:

- Wireless Web Server
- Geolocation using ESP8266
- Pressure Sensors on Railway Tracks
- Air Pollution Meter
- Temperature logging system
- World's smallest IoT project
- Wi-Fi controlled robot
- Humidity and temperature monitoring
- M2M using ESP8266

4.3. pH SENSOR:

The Analog pH Sensor Kit is specially designed for Arduino controllers and has a built-in simple, convenient, and practical connection and features. It has an LED that works as the Power Indicator, a BNC connector, and a PE2.0 sensor interface. To use this, we need to connect the pH sensor with the BNC connector, and plug

the PE 2.0 interface into the analog input port of any Arduino controller. It pre-programmed, so we will get the pH value easily.

pH Sensor:

Fig 4.3 – pH Sensor

Table 4.3 - Technical Specifications of pH Sensor

Input Supply voltage (VDC)	5V
Module Size (mm)	50 x 47 x 16
Measuring Range	0 -14 pH
Measuring Temperature	0 -50
Accuracy	0.01 pH
Response Time	1min
Cable Length (cm)	75
pH sensor size (mm)	150, 12

4.4. ULTRASONIC SENSOR:

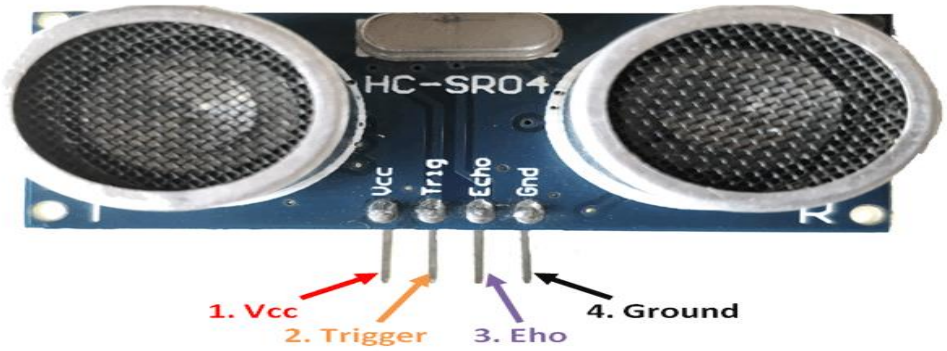

Fig 4.4 – Ultrasonic Sensor

As Fig 4.1.4 shows the Ultrasonic ranging module HC - SR04 which provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules include ultrasonic transmitters, receiver and control circuit.

The basic principle of work:

(1) Using IO trigger for at least 10us high level signal,

(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.

(3) If the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning. Test distance = (high level time×velocity of sound (340M/S) / 2,

Table 4.4 - Electric Parameter of Ultrasonic Sensor

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm

Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

4.5. GAS SENSOR:

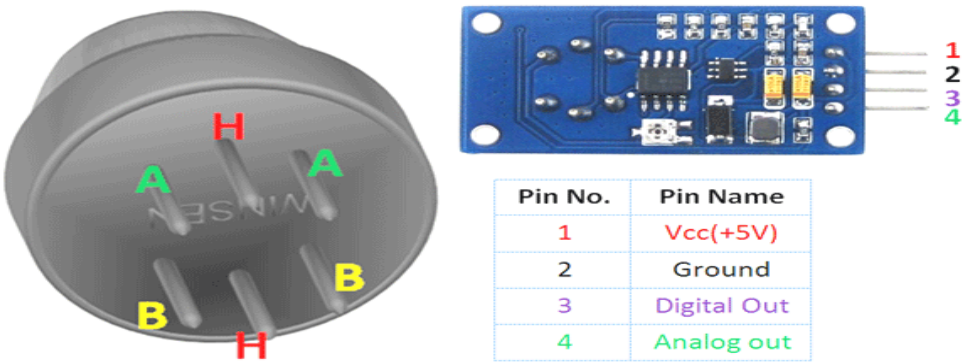

Fig 4.5 – Gas Sensor

Fig 4.5 shows the MQ-135 gas sensor which consists of 4 pins in total, which are given in below table along with their operation:

Table 4.5 - Pin Configuration of Gas Sensor

Pin No:	Pin Name:	Description
For Module		
1	Vcc	Used to power the sensor, Generally the operating voltage is +5V.
2	Ground	Used to connect the module to system ground.

3	Digital Out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer.
4	Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas.
For Sensor		
1	H -Pins	Out of the two H pins, one pin is connected to supply and the other to ground
2	A-Pins	The A pins and B pins are interchangeable. These pins will be tied to the Supply voltage.
3	B-Pins	A pins and B pins are interchangeable. One pin will act as output while the other will be pulled to ground.

We use MQ-135 gas sensor because it has wide detecting scope, fast response and high sensitivity. It has a stable and long life and the operating voltage is +5V. It detects/measures NH3, NOx, alcohol, Benzene, smoke, CO2, etc.

- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)
- Preheat duration 20 seconds
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

Applications:

- Used to detect leakage/excess of gases like Ammonia, nitrogen oxide, alcohols, aromatic compounds, Sulphide and smoke.
- Air quality monitors.

4.6. DS18B20 TEMPERATURE SENSOR:

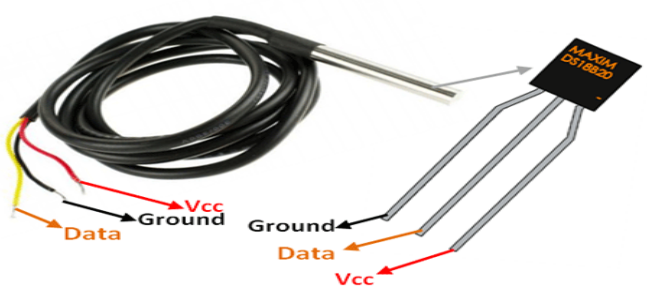

Fig 4.6 – Temperature Sensor

Fig 4.6 show the DS18B20 Temperature sensor which consists of 3 pins in total, which are given in below table along with their operation:

Table 4.6 - Pin Configuration of Temperature Sensor

Pin No	Pin Name	Description
1	Ground	Connect to the ground of the circuit
2	Vcc	Powers the Sensor, can be 3.3V or 5V
3	Data	This pin gives output the temperature value which can be read using 1-wire method

The DS18B20 is a 1-wire programmable Temperature sensor from maxim integrated. It is widely used to measure temperature in hard environments like in chemical solutions, mines or soil etc. The constriction of the sensor is rugged and also can be purchased with a waterproof option making the mounting process easy. The operating voltage is from 3V to 5V. It can measure a wide range of temperature. Each sensor has a unique 64-bit address and requires only one pin of the MCU to transfer data so it a good choice for measuring temperature at multiple points without compromising much of our digital pins on the microcontroller.

4.7. MOTOR DRIVER:

Fig 4.7 –Pin Diagram of L293D

Fig 4.7 shows the pin diagram of L293D Motor Driver which consists of 16 pins in total, which are given in below table along with their operation:

Table 4.7 - L293D Pin Configuration

Pin Number	Pin Name	Description
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits
8	Vcc2 (Vs)	Connected to Voltage pin for running motors (4.5V to 36V)
9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2

15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also, the speed and direction of these two motors can be controlled independently. So, we have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Microcontrollers like Arduino, PIC, ARM etc.

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, and it is a set-up which is used to run motors both in clock wise and anti-clockwise direction.

Applications:

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

4.8. PUMP MOTOR:

Fig 4.8 – Pump Motor

The Fig 4.8 is a water pump motor is a tough thermoplastic body; it is widely used for water priming pump, automotive pump, and experiment pump and so on.

Specification:

- Rated Voltage: DC 12V
- Load: Water
- Water absorption: 1L-1.2L/min
- Current (With load): Less than 320mA
- Flow: 2.0LPM
- Total Size: D27 x 75mm

- Water Hole Diameter: 6.5mm
- Maximum pressure: More than 360mmHg
- Noise: Less than <60dB

4.9. LCD 16X2:

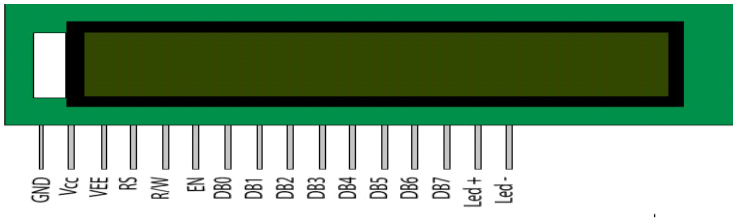

Fig 4.9 – LCD Display

As Fig 4.9 shows a LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and soon.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

4.10 SOFTWARE REQUIREMENTS:

4.10.1 ARDUINO IDE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

- Writing Sketches:

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right and corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Verify Checks your code for errors compiling it.

Upload Compiles your code and uploads it to the configured board..

New Creates a new sketch.

Open Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Save Saves your sketch.

Serial Monitor Opens the serial monitor.

Additional commands are found within the five menus: File**,** Edit**,** Sketch**,** Tools**,** and Help**.** The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

- File:

- *New* Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

- *Open* Allows loading a sketch file browsing through the computer drives and folders.

- *Open Recent* Provides a short list of the most recent sketches, ready to be opened.

- *Sketchbook* Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

- *Examples* Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- *Close* Closes the instance of the Arduino Software from which it is clicked.
- *Save* Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- *Save as...* Allows saving the current sketch with a different name.
- *Page Setup* It shows the Page Setup window for printing.
- *Print* Sends the current sketch to the printer according to the settings defined in Page Setup.
- *Preferences* Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- *Quit* Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time we start the IDE.

➢ Edit:

Undo/Redo Goes back of one or more steps you did while editing; when we go back, we may go forward with Redo.

Cut Removes the selected text from the editor and places it into the clipboard.

Copy Duplicates the selected text in the editor and places it into the clipboard.

Copy for Forum Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax colouring.

Copy as HTML Copies the code of our sketch to the clipboard as HTML, suitable for embedding in web pages.

Paste Puts the contents of the clipboard at the cursor position, in the editor.

Select All Selects and highlights the whole content of the editor.

Comment/Uncomment Puts or removes the // comment marker at the beginning of each selected line.

Increase/Decrease Indent Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

Find Opens the Find and Replace window where we can specify text to search inside the current sketch according to several options.

Find Next Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

Find Previous Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

- Sketch:

  Verify/Compile Checks our sketch for errors compiling it; it will report memory usage for code and variables in the console area.

Upload Compiles and loads the binary file onto the configured board through the configured Port.

Upload Using Programmer This will overwrite the bootloader on the board; we will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows us to use the full capacity of the Flash memory for our sketch. Please note that this command will NOT burn the fuses. To do so a *Tools -> Burn* Bootloader command must be executed.

Export Compiled Binary Saves a .hex file that may be kept as archive or sent to the board using other tools.

Show Sketch Folder Opens the current sketch folder.

Include Library Adds a library to our sketch by inserting #include statements at the start of our code. Additionally, from this menu item we can access the Library Manager and import new libraries from .zip files.

Add File... Adds a supplemental file to the sketch (it will be copied from its current location). The file is saved to the data subfolder of the sketch, which is intended for assets such as documentation. The contents of the data folder are not compiled, so they do not become part of the sketch program.

- Tools:

  Auto Format This formats our code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

Archive Sketch Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

Fix Encoding & Reload Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

Serial Monitor Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

Board Select the board that we're using.

Port This menu contains all the serial devices (real or virtual) on our machine. It should automatically refresh every time we open the top-level tools menu.

Programmer For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally we won't need this, but if we're burning a bootloader to a new microcontroller, we will use this.

Burn Bootloader The items in this menu allow us to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega microcontroller (which normally comes without a bootloader). Ensure that we've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

- Help:

  Here we find easy access to a number of documents that come with the Arduino Software (IDE). We have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection.

  Find in Reference This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

- Sketchbook:

  The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in our sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time we run the Arduino software, it will automatically create a directory for our sketchbook. We can view or change the location of the sketchbook location from with the Preferences dialog.

- Tabs, Multiple Files, and Compilation:

  Allows us to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

  Before compiling the sketch, all the normal Arduino code files of the sketch (.ino, .pde) are concatenated into a single file following the order the tabs are shown in. The other file types are left as is.

- Uploading:

  Before uploading our sketch, we need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an UNO or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5**,** COM7**,** or higher (for a USB board) - to find out, we look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once we've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, we'll need to press the reset button on the board just before starting the upload. On most boards, we'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

  When we upload a sketch, we're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on our board. It allows us to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

- Libraries:

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with our sketch. Because libraries are uploaded to the board with our sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of our code.

- Serial Monitor:

This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to serial.begin in our sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun our sketch) when we connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if our sketch needs a complete management of the serial communication with control characters, we can use an external terminal program and connect it to the COM port assigned to your Arduino board.

- Language Support:

Since version 1.0.1, the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by our operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If we would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select our preferred language from the menu, and restart the software to use the selected language. If our operating system language is not supported, the Arduino Software (IDE) will default to English.

We can return the software to its default setting of selecting its language based on our operating system by selecting System Default from the Editor Language drop-down. This setting will take effect when we restart the Arduino Software (IDE). Similarly, after changing our operating system's settings, we must restart the Arduino Software (IDE) to update it to the new default language.

4.10.2. EMBEDDED C LANGUAGE:

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems.

Embedded C programming typically requires nonstandard extensions to the C language in order to support enhanced microprocessor features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. The C Standards Committee produced a Technical Report, most recently revised in 2008 and reviewed in 2013, providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as fixed-point arithmetic, named address spaces and basic I/O hardware addressing. Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, data type declaration, conditional statements (if,

switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

4.11. DESIGN AND IMPLEMENTATION OF WATER QUALITY MONITORING SYSTEM:

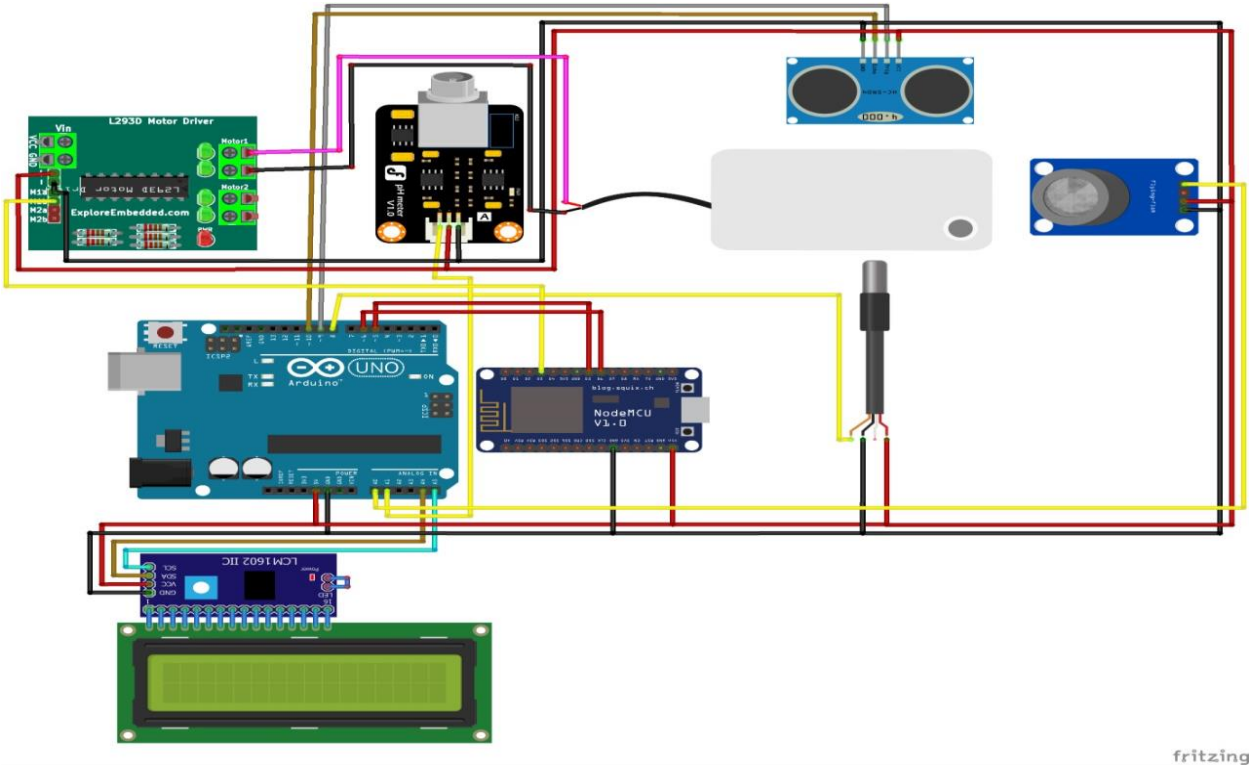

Fig 4.11 – Circuit Diagram of Water Quality Monitoring

From the Fig 4.3 we can see the circuit diagram of water quality monitoring. From the power pins the 5V is given as a supply to all and Gnd is given as a supply to all components. The Analog pin A0 is connected to Gas sensor, A1 is connected to pH sensor, A4 is connected to SDA and the A5 is connected to SCL. The Digital pins 5 and 6 are B5 and B6 Node MCU, pin 8 is connected to temperature sensor, pin 9 is connected to trig and pin 10 is connected to Echo of the ultrasonic sensor. The pin B3 of Node MCU is connected to L293D motor, and Motor 1 is connected to pump which is immersed in the water.

CHAPTER 5

RESULTS AND DISCUSSION

In this chapter we discuss about the results of the proposed water quality monitoring system. The hardware prototype and the mobile application results are discussed below.

Fig 5.1 – Water Quality Monitoring

The Fig 5.1 shows the setup of Arduino UNO and sensors for monitoring the water quality. The Arduino is connected with ESP8266 Wi-Fi module and Sensors like pH, Ultrasonic, Gas and Temperature for monitoring the water quality.

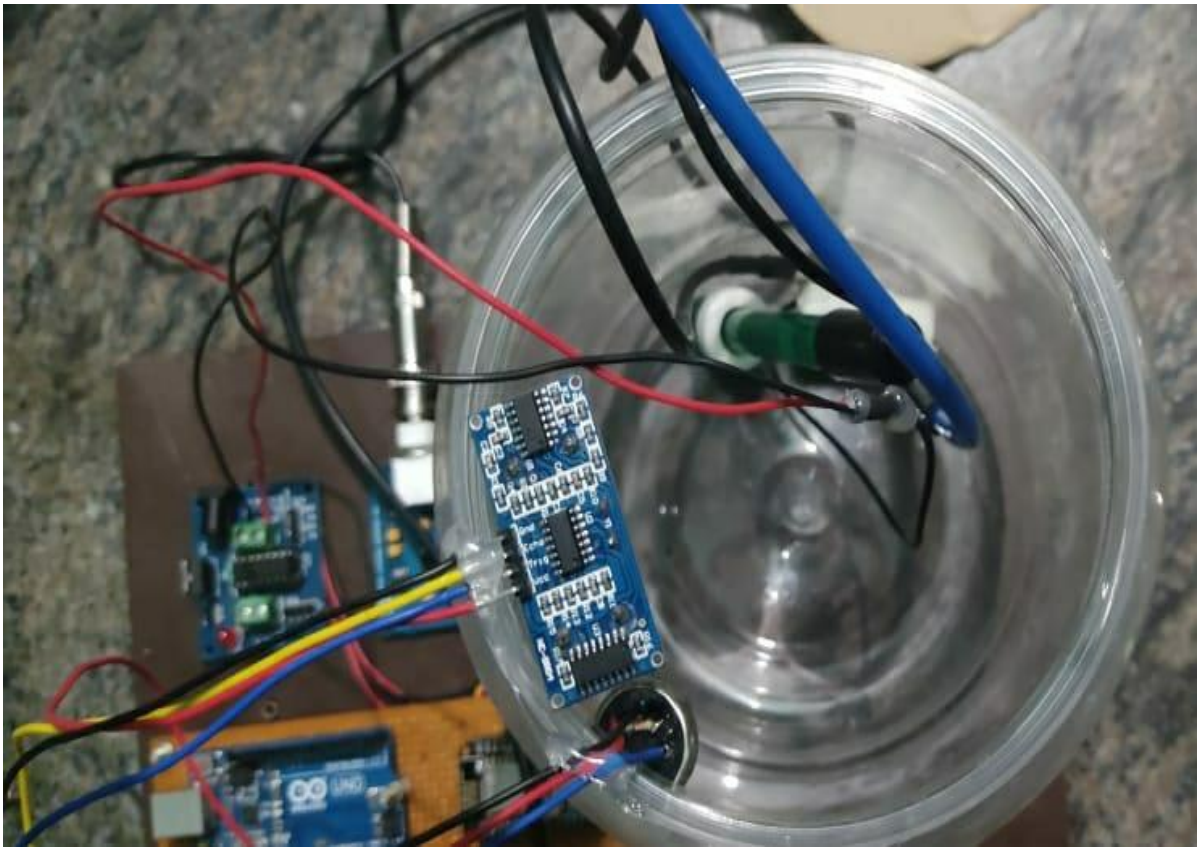

Fig 5.2 –Top View of Water Quality Monitoring

The Fig 5.2 shows the temperature and pH sensor immersed in water tank and the ultrasonic and gas sensors on the top of the water tank. And the pump motor is also immersed inside the tank for pumping out the water if the parameters show the values high or low then the desired range.

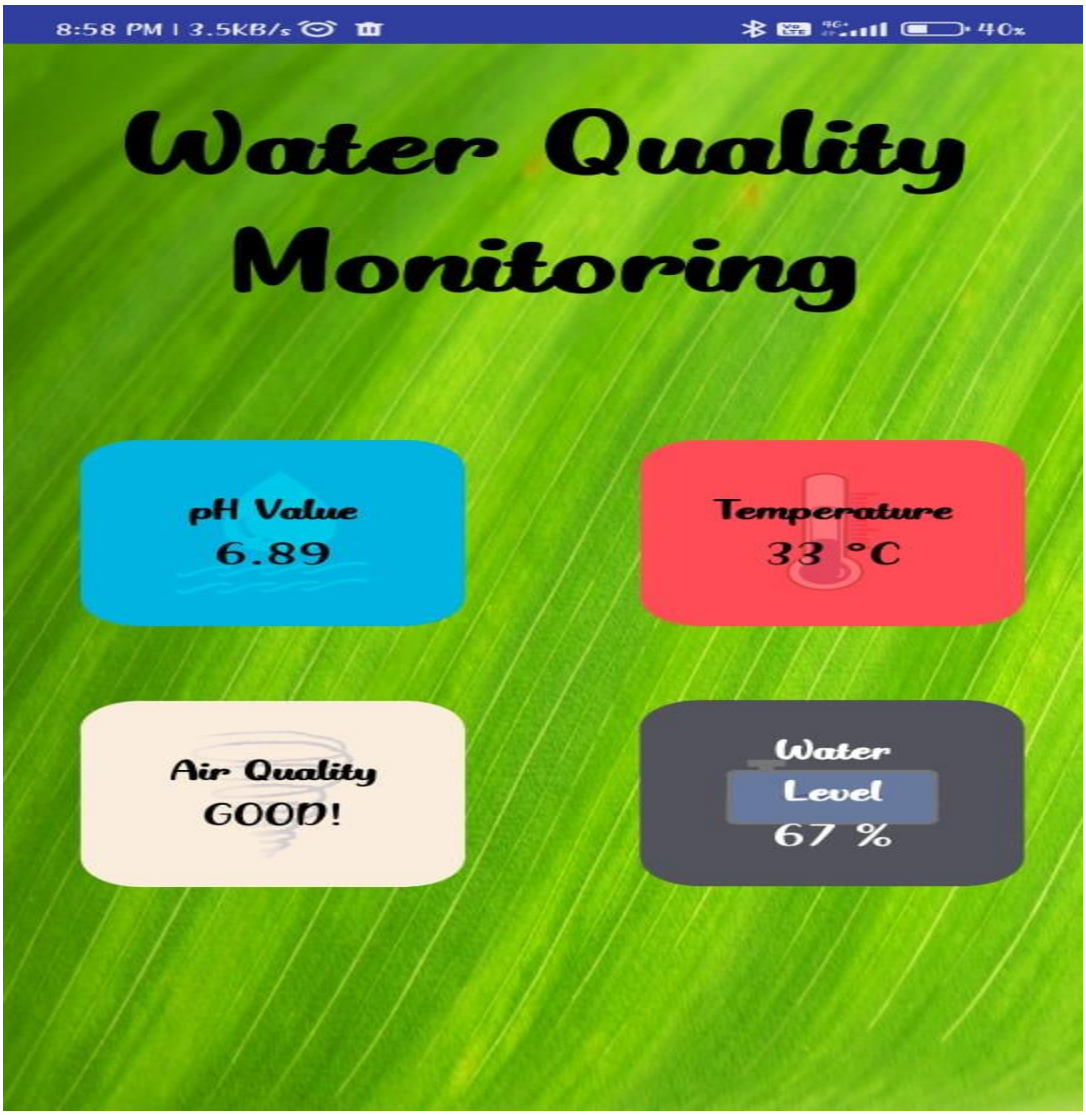

Fig 5.3 – Mobile Application Output

Fig 5.3 shows the result of mobile application where we can see the values of measured parameters such as measured values of pH, Temperature, Air Quality and level of water.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The system proposed is cost-efficient and inexpensive IoT solution for real-time water quality monitoring (WQM). The developed system is having Arduino UNO and ESP8266 are interfaced with several sensors successfully. An efficient algorithm is developed in real-time, to track water quality. The measured pH value of water ranges from 6.5 to7.5. A mobile-based application is used to monitor the parameters such as pH value, temperature, air quality and level of water in the tank through the mobile application. Further, these measured parameters are monitored in mobile application.

This project can be extended to analyze several other parameters like electrical conductivity, free residual chlorine, and dissolved oxygen in the water.

APPENDIX:

```
#include <SoftwareSerial.h>

#include <ArduinoJson.h>

SoftwareSerialnodemcu(5, 6);//rx/tx

#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE_WIRE_BUS 8  // pin changable to any digital

OneWireoneWire(ONE_WIRE_BUS);

DallasTemperatureTemperature_Sensor(&oneWire);

intTemperature_Val;

constinttrigPin = 9; // analog

constintechoPin = 10; // analog

long duration;

float distance;

intWater_Level;

floatWater_Max = 20;

#define SensorPin A1

unsigned long intavgValue;

float b;
```

```
intbuf[10],temp;

floatphValue;

intGasLevel_Good_Min = 0;

intGasLevel_Good_Max = 750;

#include <SPI.h>

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <Fonts/FreeSans9pt7b.h>

#include <Fonts/FreeMonoOblique9pt7b.h>

#define AQ_Sen A0

intGasLevel = 0;

String quality ="";

void setup()

{

Serial.println("Program started");

  {

pinMode(8, INPUT);pinMode(7, OUTPUT);

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
```

```
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600);
  }
    {
Temperature_Sensor.begin();
    }
      {
pinMode(A0,INPUT);
Serial.println("Ready");
      }
  {
pinMode(AQ_Sensor,INPUT);
        }
nodemcu.begin(9600);
delay(1000);
}
void loop()
{
{
```

```
StaticJsonBuffer<1000>jsonBuffer;
JsonObject& data = jsonBuffer.createObject();
US_Sensor();
Temp_Sensor();
PH_Sensor();
AQ_Sensor();
  //------------------------------Assign collected data to JSON Object
  //data["Distance: "] = distance;
data["Water Level: "] = Water_Level;
data["Temperature: "] = Temperature_Val;
data["Air Quality: "] = quality;
data["PH Value: "] = phValue;
  //------------------------------Send data to NodeMCU
data.printTo(nodemcu);
jsonBuffer.clear();
delay(500);
{
if (phValue< 6.5 || phValue> 7.5)
    {
```

```
digitalWrite(7, HIGH);
Serial.println("Motor ON");
    }
if (phValue>= 6.5 &&phValue<=7.5)
    {
digitalWrite(7, LOW);
Serial.println("Motor OFF");
    }
delay(50);
}
}
}
voidUS_Sensor()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

```
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
Serial.print("Distance: ");
Serial.println(distance);
  {
if (distance >= 5 && distance <=20)
{ Water_Level = ((Water_Max - distance)/.15);
Serial.print("Water Level: ");
Serial.println(Water_Level);
    //Firebase.setFloat ("Water Level",Water_Level);
    }
if (distance < 5)
{ Water_Level = 100;
Serial.print("Water Level: ");
Serial.println(Water_Level);
   //Firebase.setFloat ("Water Level",Water_Level);};
    }
if (distance > 20)
{ Water_Level = 0;
```

```
Serial.print("Water Level: ");
Serial.println(Water_Level);
   //Firebase.setFloat ("Water Level",Water_Level);};
    }
  }
}
voidTemp_Sensor()
{
Temperature_Sensor.requestTemperatures();
Temperature_Val = Temperature_Sensor.getTempCByIndex(0);
Serial.print("Temperature: ");
Serial.print(Temperature_Val);
Serial.println(" C");
delay(50);
}
voidAQ_Sensor()
{
GasLevel = analogRead(AQ_Sensor);
```

```
if(GasLevel>=GasLevel_Good_Min||GasLevel<=  GasLevel_Good_Max){quality = "GOOD!";}

if(GasLevel<GasLevel_Good_Min ||  GasLevel>GasLevel_Good_Max){quality = "Poor!";}

 //if (GasLevel>880 &&GasLevel<=920){quality = "Very_bad!";}

 // else{quality = "Toxic!";}

Serial.print("GasLevel");

Serial.println(GasLevel);

Serial.print("Air Quality: ");

Serial.println(quality);

delay(50);

}

voidPH_Sensor()

{

for(inti=0;i<10;i++)          //Get 10 sample value from the sensor for smooth the value

  {

buf[i]=analogRead(SensorPin);

delay(10);

  }
```

```
for(inti=0;i<9;i++)       //sort the analog from small to large
{ for(int j=i+1;j<10;j++)
{ if(buf[i]>buf[j])
{ temp=buf[i];
buf[i]=buf[j];
buf[j]=temp;
      }
    }
  }
avgValue=0;
for(inti=2;i<8;i++) //take the average value of 6 center sample
avgValue+=buf[i];
phValue=(float)avgValue*5.0/1024/6; //convert the analog into millivolt
 // phValue=3.5*phValue;                    //convert the millivolt into pH value
phValue=1+(3.5*phValue);
Serial.print("PH Value: ");
 // phValue = (roundf(phValue*1000))/10;
Serial.println(phValue);
  //float phdeci2 = (roundf(phValue*10))/10;
```

```
 // Serial.println(phdeci2);

delay(50);

}
```

REFERENCES:

[1] A. Jerom B., R. Manimegalai and R. Manimegalai, "An Iot Based Smart Water Quality Monitoring System Using Cloud," 2020 International Conference on Emerging Trends in Information Technology and Engineering (Ic-Etite), 2020, Pp. 1-7, Doi: 10.1109/Ic-Etite47903.2020.450.

[2] Cloete, Niel Andre, Malekian, Reza, Nair, Lakshmi, 2014. Design of smart sensors for Real-time water quality monitoring, Department of electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa. IEEE J. 13, 1–16.

[3] Daigavane, Vaishnavi V., Gaikwad, M.A., 2017. Water quality monitoring system based on IoT. Adv. Wireless Mobile Commun.. ISSN 10, 1107–1116.

[4] Das, Brinda, Jain, P.C., 2017. Real-time water quality monitoring system using the internet of things. Int. Conf. Comput. Commun. 78–82.

[5] G. Nitish Satya Sai, Reddy Sudheer, Kondreddy Sai Manikanta, Sri Ganesh ArjulaBandi Narasimha Rhave Iot Based Water Quality Monitoring System IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC),2020.

[6] He Donge, Zhang, Li-Xin, 2012. The water quality monitoring system based on wireless sensor network. In: Report: Mechanical and Electronic Information Institute, China University of GeoScience, Wu Hen, China.

[7] Lambrou, Theofanis P., Anastasiou, Christos C., Panayiotou, Christos G.,Polycarpou, Marios M., 2014. A low-cost sensor network for real-time

monitoring and contamination detection in drinking water distribution systems. IEEE Sensor. J. 8,2765–2772.

[8] Meng, F., Fu, G., Butler, D., 2017. Cost-effective river water quality management using integrated real-time control technology. Environ. Sci. Technol. 51, 9876–9886.

[9] Moparthi, Nageswara Rao, Mukesh, Ch, VidyaSagar, P., 2018. Water quality monitoring system using IoT. 4th International Conference on Advances in Electrical, Electronics, Information, Communication, and Bio-Informatics.

[10] Omar Faruq, Md., Hoque Emu, Injamamul, Nazmul Haque1, Md., Dey, Maitry, Das, N.K.,Dey, Mrinmoy, 2017. Design and implementation of a cost-effective water quality evaluation system. In: IEEE Region 10 Humanitarian Technology Conference, Dhaka, Bangladesh, pp. 860–863.

[11] Prasad, A.N., Mamun, K.A., Islam, F.R., Haqva, H., 2015. Smart water quality monitoring system. In: The University of the South Pacific. 2nd Asia-Pacific World congress on Computer Science and Engineering IEEE Conference.

[12] SathishPasika, Sai Teja GandlahaveSmart Water Quality Monitoring System with Cost-Effective Using Iot Department of Electronics and Communication Engineering Chaitanya Bharathi Institute of Technology, Hyderabad (TS), India 2021.

[13] Shafi, Uferah, Mumtaz, Rafia, Anwar, Hirra, Mustafa Qamar, Ali, Khurshid, Hamza, 2018.Surface Water Pollution Detection Using the Internet of Things. School of Electrical Engineering and Computer Science, National University of Science and Technology, IEEE Conference, pp. 92–96.

[14] Siddula, Sai Sreekar, Babu, Phaneendra, Jain, P.C., 2018. Water level monitoring and management of dams using IoT. In: IEEE, EE Department Shiv Nadar University.

[15] Siregar, Baihaqi, Menen, Krisna, Efendi, Syahril, Andayani, Ulfi, 2017. Monitoring quality standard of waste water using wireless sensor network technology for smart environment. In: The International Conference on ICT for Smart Society (ICISS).

[16] Srishaila Mallikarjuna Swamy, P.M., Mahalakshmi, G., 2017. Real-Time Monitoring of Water quality using the smart sensor. JETER J. 4, 139–144.

[17] Sugapriyaa, Tha, Rakshaya, S., Ramyadevi, K., Ramya, M., Rashmi, P.G., 2018. Smart water quality monitoring system for real-time applications. Int. J. Pure Appl. Math.118, 1363–1369.

[18] Varsha Lakshmikantha, AnjithaHiriyannagowd, AkshayManjunath, Aruna Patted, JagadeeshBasavaiah, Audre Arlene Anthony Iot Based Smart Water Quality Monitoring System Department of Electronics and Communication Engineering, Vidya vardhaka College of Engineering, Mysuru, India, 2021.

[19] Whittle, A.J., Allen, M., Preis, A., Iqbal, M., 2013. Sensor Networks for Monitoring and Control of Water Distribution Systems. Coventry University, Coventry, UK. MIT article.

[20] ZinMyint, Cho, Gopal, Lenin, Lin Aun, Yan, 2017. Reconfigurable smart water quality monitoring system in an IoT environment. IEEE ICIS 435–440.