A Project Report

on

**IMPLEMENTATION OF REAL TIME MONITORING SYSTEM FOR AQUACULTURE**

***Submitted in partial fulfillment of the requirement for the award of degree of***

**BACHELOR OF TECHNOLOGY**

*in*

**ELECTRONICS AND COMMUNICATION ENGINEERING**

***Submitted by***

|  |  |  |
| --- | --- | --- |
| **Y.HARITHA NAGA SAI** | **-** | **15FE1A04G5** |
| **R.RAMAKANTH SINGH** | **-** | **15FE1A04C4** |
| **U.RAGHAVENDRA RAO** | **-** | **15FE1A04F3** |
| **K.AJAY** | **-** | **16FE5A0419** |

***Under the esteemed guidance of***

**Mr. G .HARISH M .Tech.,**

Assistant Professor



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**VIGNAN’S LARA INSTITUTE OF TECHNOLOGY & SCIENCE**

**(An ISO 9001:2008 Certified, Approved by AICTE, Affiliated to JNTU, KAKINADA)**

**VADLAMUDI-522213, GUNTUR Dist., ANDHRA PRADESH.**

**2015 - 2019**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**VIGNAN’S LARA INSTITUTE OF TECHNOLOGY & SCIENCE**

**(An ISO 9001:2008 Certified, Approved by AICTE, Affiliated to JNTU, KAKINADA)**

**VADLAMUDI-522213, GUNTUR Dist, ANDHRA PRADESH**



**CERTIFICATE**

This is to certify that main project work entitled **“IMPLEMENTATION OF REAL TIME MONITORING SYSTEM FOR AQUACULTURE”** is a bonafide work done by **Y.HARITHA NAGA SAI (15FE1A04G5), R.RAMAKANTH SINGH (15FE1A04C4), U.RAGHAVENDRA RAO (15FE1A04F3) and K.AJAY (16FE5A0419),**under my guidance and submitted in partial fulfillment of requirement for award of degree of **Bachelor of Technology** in **Electronics and Communication Engineering** by **Jawaharlal Nehru Technological University**, **Kakinada.**

**Project Guide Head of the Department**

**Mr. G .Harish., M. Tech, Mr. M. Suman., M. Tech,**

Assistant Professor Assistant Professor

**External Examiner**

**ACKNOWLEDGEMENT**

We are most thankful to our parents who stood as pillars of motivation and for the way they influenced and molded our lives.

We are more thankful to our chairman **Dr. Lavu Rathaiah** who helped us to have a technical incubation by providing the required infrastructure.

We are very much thankful to our principal **Dr. K.PHANEENDRA KUMAR** who extended a timely help at each and every step of our academic career.

We are very thankful to our Head of the department **Mr. M.Suman** an amicable person who supported us very much and helped to a maximum extent and made this project successful.

We are very thankful to our beloved coordinators **Mrs. P. V. N. LAKSHMI, Mr. B. HARISH** and **Mr. S. NAGARAJU** for inspiring all the way and arranging all the facilities and resources needed for project. Their efforts in this aspect are beyond the preview of the acknowledgement.

We are heartily thankful to our project guide **Mr. G.HARISH, Assistant Professor,** the person with vibrant knowledge and amicable by nature who laid a best guide lines and for the efforts made by him to make the project a successful one.

Finally, we are thankful to each and every faculty members both technical and non-technical, friends and all the persons who helped us directly or indirectly in making our project a successful one.

***Project Associates***

**Y.HARITHA NAGA SAI (15FE1A04G5)**

**R.RAMAKANTH SINGH (15FE1A04C4)**

**U.RAGHAVENDRA RAO (15FE1A04F3)**

**K.AJAY (16FE5A0419)**

**DECLARATION**

We hereby declare that the work described in this project work, entitled “**IMPLEMENTATION OF REAL TIME MONITORING SYSTEM FOR AQUACULTURE**” which is submitted by us in partial fulfillment for the award of **Bachelor of Technology** (**B.Tech**) in the Department of **Electronics & communication Engineering** to the Vignan’s Lara Institute of Technology and science affiliated to Jawaharlal Nehru Technological University Kakinada,, Andhra Pradesh, is the result of work done by us under the guidance of **Mr. G.HARISH,** Assistant Professor, in the department ECE**.**

The work is original and has not been submitted for any Degree/Diploma of this or any other university.

**Y.HARITHA NAGA SAI (15FE1A04G5) R.RAMAKANTH SINGH (15FE1A04C4) U.RAGHAVENDRA RAO (15FE1A04F3)**

**K.AJAY (16FE5A0419)**

**Place:**Vadlamudi.

**Date:**

**ABSTRACT**

Aquaculture is one of the flourishing sectors in India as it contributes nearly 1.07% of the GDP. Technological significance has been a great support for settling on choice in various fields especially in aquaculture. In recent years commercial aquaculture is facing many problems due to sudden climatic fluctuations which leads to changes in water quality parameters. At present aqua farmers are depending on manual testing to know the quality of water. This will consume time and inaccurate because water quality parameters may alter with time. In order to overcome this problem knowledge based real time monitoring system for aquaculture was implemented.

The proposed model mainly focuses on constant monitoring of water quality parameters from time to time in order to take preventive measures before actual damage was done. The proposed architecture consists of four modules. They are power module ,sensor module ,microcontroller module, output module.

Sensors are interfaced to the arduino mega board. Threshold values for the parameters are predefined in the code. When there is a variation in the parameter values then the arduino sends commands to the GSM module. Then it establishes a network and sends message to the mobile number specified in the code.

**INDEX**

**CONTENTS**  **PAGE NO.**

**ABSTRACT i**

**LIST OF FIGURES vi**

**LIST OF TABLES vii**

**NOMENCLATURE viii**

**CHAPTER1: INTRODUCTION 01-03**

* 1. Introduction to aqua monitoring 01
  2. Introduction to turbidity 01 1.2.1 Causes 02

1.2.2 Measurement 03

1.3 Introduction to pH 03

**CHAPTER 2: LITERATURE SURVEY 04-05**

2.1 Technology model of aquaculture production system 04

2.2 Design and deployment of aqua monitoring system using

wireless sensor networks and IAR Kick 04

2.3 Water quality monitoring system based on IOT 05

2.4 Knowledge based real time monitoring system for

aquaculture using IOT 05

**CHAPTER 3 : DESIGN OF AQUAMONITORING SYSTEM 06-12**

3.1 Existing system 06

3.1.1 Manual testing 06

3.1.2 Drawbacks of manual testing 07

3.2 Proposed system 07

3.2.1 Objective 07

3.2.2 Block diagram 08

3.2.3 Flow chart 09

3.2.4 Working 10

3.2.5Tools Required 10

3.2.5.1 Hardware tools 10

3.2.5.2 Software tools 10

3.2.6 Applications 11

3.2.7 Advantages 11

3.2.8 Tables related to water quality parameters 12

**CHAPTER 4: HARDWAREAND SOFTWARE DESCRIPTION 13-31**

4.1Arduino Mega 2560 13

4.1.1 Overview 13

4.1.2 Power 13

4.1.3 Memory 14

4.1.4 Input and output 14

4.1.5 Communication 15

4.1.6 Programming 16

4.2 GSM module 16

4.2.1 Specifications 17

4.2.2 Components in the module 17

4.2.3 Pin configuration 17

4.3 Ultrasonic sensor 18

4.3.1 Sensor Features 18

4.3.2 Working 19

4.3.3 Sensor usage 19

4.3.4 Applications 20

4.4Turbidity sensor 20

4.4.1 Introduction 21

4.4.2 Interface description 21

4.4.3 Operation 21 4.5 pH sensor 22

4.5.1 Introduction 22

4.5.2 Sensor features 22

4.5.3 Important notes 23

4.6 LCD Module( 2x16 character) 24

4.7Arduino IDE 25

4.7.1 Connecting the arduino 27

4.7.2 Preparing the board 27

4.7.3 Loading code 28

4.8 Embedded C programming 28

4.8.1 Introduction to Embedded C programming 28

4.8.2 Embedded systems programming 29

4.8.3 Difference between C programming and Embedded C programming 30

4.8.4 Programming using Embedded C 31

**CHAPTER 5: RESULTS AND DISCUSSION 32**

**CONCLUSION 37**

**FUTURE SCOPE 38**

**REFERENCES 39**

**APPENDIX 41**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Figure Name** | **Page No.** |
| Fig. 3.1 | Collecting water samples for quality measurement | 06 |
| Fig. 3.2 | Block diagram of proposed system | 08 |
| Fig. 3.3 | Flowchart of proposed system | 09 |
| Fig. 4.1 | Arduino Mega 2560 | 13 |
| Fig. 4.2 | GSM module | 16 |
| Fig. 4.3 | Ultrasonic sensor | 18 |
| Fig. 4.4 | Turbidity sensor | 20 |
| Fig. 4.5 | pH sensor | 22 |
| Fig. 4.6 | LCD diagram | 24 |
| Fig. 4.7 | Arduino IDE page | 26 |
| Fig. 5.1 | Required setup | 32 |
| Fig. 5.2 | Displaying turbidity value | 33 |
| Fig. 5.3 | Displaying pH value | 34 |
| Fig. 5.4 | Displaying water level | 35 |
| Fig. 5.5 | Message in user mobile | 36 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
| Table 3.1 | Water quality parameters with threshold range | 12 |
| Table 3.2 | Causes and effects of water quality parameters | 12 |

**NOMENCLATURE**

|  |  |
| --- | --- |
| **ACRONYM** | **DESCRIPTION** |
| GSM | Global System for Mobile communication |
| NTU | Nephelometric Turbidity Units |
| GPM | Gallons Per Minute |
| GDP | Gross Domestic Product |
| LCD | Liquid Crystal Display |
| IDE | Integrated Development Environment |
| SIM | Subscriber Identity Module |
| ICSP | In Circuit Serial Program |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction to Aquamonitoring:**

Water monitoring in ponds is the daily observation and analysis of water parameters to ensure provisional water quality and quantity by adopting good water quality management strategies and as well ensure it supply in adequacy through the appropriate channels of flow as well it acceptability for fish culture. Fishes are predisposed to diseases and other problems due to poor water quality in ponds, and daily water monitoring in ponds is a panacea to many pond problems. To carry out proper water monitoring, the right equipment and manpower must be in place with considerable expertise on the intricacies involved, in order to get accurate data for complete and accurate results.

Water sources and supplies are imperative in fish culture, as it is a precursor for good water quality. There are two main sources of water to ponds: 1. Ground water: Spring and well or borehole water. 2. Surface water: Rain and run-off water, natural water course, irrigation canals and drainage canals. Ground water supplies are reliable as they have their source from aquifers lying at different depths. However, major water supplies in most fish ponds within the country today is through boreholes and natural water course and people recourse to natural water course because it is cheap compared to the high cost of construction and pumping.

* 1. **Introduction to turbidity:**

Turbidity is the cloudiness or [haziness](https://en.wikipedia.org/wiki/Haze) of a [fluid](https://en.wikipedia.org/wiki/Fluid) caused by large numbers of individual [particles](https://en.wikipedia.org/wiki/Particle_(ecology)) that are generally invisible to the [naked eye](https://en.wikipedia.org/wiki/Naked_eye), similar to [smoke](https://en.wikipedia.org/wiki/Smoke) in [air](https://en.wikipedia.org/wiki/Air). The measurement of turbidity is a key test of [water quality](https://en.wikipedia.org/wiki/Water_quality).

[Fluids](https://en.wikipedia.org/wiki/Fluids) can contain suspended solid matter consisting of particles of many different sizes. While some suspended material will be large enough and heavy enough to settle rapidly to the bottom of the container if a liquid sample is left to stand (the [settable solids](https://en.wikipedia.org/wiki/Settling)), very small particles will settle only very slowly or not at all if the sample is regularly agitated or the particles are [colloidal](https://en.wikipedia.org/wiki/Colloid). These small solid particles cause the liquid to appear turbid.

* + 1. **Causes:**

Turbidity in open water may be caused by growth of [phytoplankton](https://en.wikipedia.org/wiki/Phytoplankton). Human activities that disturb land, such as [construction](https://en.wikipedia.org/wiki/Construction), [mining](https://en.wikipedia.org/wiki/Mining) and [agriculture](https://en.wikipedia.org/wiki/Agriculture), can lead to high [sediment](https://en.wikipedia.org/wiki/Sediment) levels entering water bodies during rain storms due to [storm water](https://en.wikipedia.org/wiki/Storm_water) [runoff](https://en.wikipedia.org/wiki/Surface_runoff). Areas prone to high bank erosion rates as well as [urbanized](https://en.wikipedia.org/wiki/Urbanized) areas also contribute large amounts of turbidity to nearby waters, through stormwater [pollution](https://en.wikipedia.org/wiki/Water_pollution) from [paved](https://en.wikipedia.org/wiki/Pavement_(material)) surfaces such as roads, bridges and parking lots. Some industries such as [quarrying](https://en.wikipedia.org/wiki/Quarry), [mining](https://en.wikipedia.org/wiki/Coal_mine) and coal recovery can generate very high levels of turbidity from colloidal rock particles.

In drinking water, the higher the turbidity level, the higher the risk that people may develop [gastrointestinal diseases](https://en.wikipedia.org/wiki/Gastrointestinal_diseases). This is especially problematic for immune compromised people, because contaminants like [viruses](https://en.wikipedia.org/wiki/Virus) or [bacteria](https://en.wikipedia.org/wiki/Bacteria) can become attached to the suspended solids. The suspended solids interfere with water disinfection with [chlorine](https://en.wikipedia.org/wiki/Chlorine) because the particles act as shields for the virus and bacteria. Similarly, suspended solids can protect bacteria from [ultraviolet (UV) sterilization](https://en.wikipedia.org/wiki/Ultraviolet_germicidal_irradiation) of water.

In water bodies such as [lakes](https://en.wikipedia.org/wiki/Lake), rivers and [reservoirs](https://en.wikipedia.org/wiki/Reservoirs), high turbidity levels can reduce the amount of light reaching lower depths, which can inhibit growth of submerged [aquatic plants](https://en.wikipedia.org/wiki/Aquatic_plants) and consequently affect species which are dependent on them, such as [fish](https://en.wikipedia.org/wiki/Fish) and [shellfish](https://en.wikipedia.org/wiki/Shellfish). High turbidity levels can also affect the ability of fish gills to absorb dissolved oxygen. This phenomenon has been regularly observed throughout the [Chesapeake Bay](https://en.wikipedia.org/wiki/Chesapeake_Bay) in the eastern United States.

For many [mangrove](https://en.wikipedia.org/wiki/Mangrove) areas, high turbidity is needed in order to support certain species, such as to protect juvenile fish from predators. For most mangroves along the eastern coast of [Australia](https://en.wikipedia.org/wiki/Australia), in particular [Moreton Bay](https://en.wikipedia.org/wiki/Moreton_Bay), turbidity levels as high as 600 [Nephelometric](https://en.wikipedia.org/wiki/Nephelometer) Turbidity Units (NTU) are needed for proper [ecosystem](https://en.wikipedia.org/wiki/Ecosystem) health.

Turbidity (or haze) is also applied to transparent solids such as glass or plastic. In plastic production, haze is defined as the percentage of light that is deflected more than 2.5° from the incoming light direction.

* + 1. **Measurement:**

The most widely used measurement unit for turbidity is the [Formazin](https://en.wikipedia.org/wiki/Formazine) Turbidity Unit (FTU). [ISO](https://en.wikipedia.org/wiki/International_Organization_for_Standardization) refers to its units as FNU (FormazinNephelometric Units). [ISO 7027](https://en.wikipedia.org/wiki/ISO_7027) provides the method in water quality for the determination of turbidity. It is used to determine the [concentration](https://en.wikipedia.org/wiki/Concentration) of suspended particles in a sample of water by measuring the incident light scattered at right angles from the sample. The scattered light is captured by a [photodiode](https://en.wikipedia.org/wiki/Photodiode), which produces an electronic signal that is converted to a turbidity. [Open source hardware](https://en.wikipedia.org/wiki/Open_source_hardware) has been developed following the ISO 7027 method to measure turbidity reliably using an [Arduino](https://en.wikipedia.org/wiki/Arduino) microcontroller and inexpensive [LEDs](https://en.wikipedia.org/wiki/LED).

* 1. **Introduction to pH:**

pH ([/piːˈeɪtʃ/](https://en.wikipedia.org/wiki/Help:IPA/English)) is a [logarithmic scale](https://en.wikipedia.org/wiki/Logarithmic_scale) used to specify the [acidity](https://en.wikipedia.org/wiki/Acidity) or [basicity](https://en.wikipedia.org/wiki/Basicity) of an [aqueous solution](https://en.wikipedia.org/wiki/Aqueous_solution). It is approximately the negative of the base 10 [logarithm](https://en.wikipedia.org/wiki/Logarithm) of the [molar concentration](https://en.wikipedia.org/wiki/Molar_concentration), measured in units of [moles](https://en.wikipedia.org/wiki/Mole_(unit)) per liter, of [hydrogen ions](https://en.wikipedia.org/wiki/Hydrogen_ion). More precisely it is the negative of the base 10 logarithm of the [activity](https://en.wikipedia.org/wiki/Activity_(chemistry)) of the hydrogen ion. At 25 °C, solutions with a pH less than 7 are acidic and solutions with a pH greater than 7 are [basic](https://en.wikipedia.org/wiki/Basic_(chemistry)). The neutral value of the pH depends on the temperature, being lower than 7 if the temperature increases. [Pure water](https://en.wikipedia.org/wiki/Properties_of_water) is neutral, pH 7 at (25 °C), being neither an acid nor a base. Contrary to popular belief, the pH value can be less than 0 or greater than 14 for very strong acids and bases respectively.

Measurements of pH are important in [agronomy](https://en.wikipedia.org/wiki/Agronomy), medicine, chemistry, water treatment, and many other applications.

The pH scale is [traceable](https://en.wikipedia.org/wiki/Traceable) to a set of standard solutions whose pH is established by international agreement.[[3]](https://en.wikipedia.org/wiki/PH#cite_note-covington-3) Primary pH standard values are determined using a [concentration cell with transference](https://en.wikipedia.org/wiki/Galvanic_cell), by measuring the potential difference between a [hydrogen electrode](https://en.wikipedia.org/wiki/Hydrogen_electrode) and a standard electrode such as the [silver chloride electrode](https://en.wikipedia.org/wiki/Silver_chloride_electrode). The pH of aqueous solutions can be measured with a [glass electrode](https://en.wikipedia.org/wiki/Glass_electrode) and a [pH meter](https://en.wikipedia.org/wiki/PH_meter), or an [indicator](https://en.wikipedia.org/wiki/PH_indicator).

There are three current theories used to describe [acid–base reactions](https://en.wikipedia.org/wiki/Acid%E2%80%93base_reaction): Arrhenius, Bronsted-Lowry and Lewis when determining pH.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Technology Model of Aquaculture Production System:**

The high market demand has led to the rapid growth in fish farming. The young generation are inexperienced in determining the estimated results of fish farming and the preparation of fish pond during the period of fish farming. These need a complete guide as their reference which includes the knowledge of fish farming. The main objective of this project is to develop a practical design of real pond appropriate with aquaculture technology and fish farming production. There are three parts of study in this project which include fish farming cage, growth of fish and water quality of fish farming pond. Few of experiments were carried out involved the collection data in terms of growth of fish and parameters of water quality.

**2.2 Design and Deployment of Aqua Monitoring System Using Wireless Sensor Networks and IAR-Kick:**

In Aquaculture, the yields (shrimp, fish etc.) depend on the water characteristics of the aquaculture pond. For maximizing fish yields, the parameters which are to be kept at certain optimal levels in water are dissolved oxygen, temperature, salinity, turbidity, pH level, alkalinity and hardness, ammonia and nutrient levels. These parameters can vary a lot during the period of a day and can rapidly change depending on the external environmental conditions. Hence it is necessary to monitor these parameters with high frequency, if not continuously, for timely analysis and action. This need accurate real- time information system and performance in order to maximize their potential. Wireless sensor networks are used to monitor aqua farms for relevant parameters, such as pH levels, humidity, dissolved oxygen levels, water temperature, ammonia levels etc. This system consists of two modules which are transmitter station and receiver station. The transmitter station consists of sensor nodes such as pH, humidity, and temperature inside and outside of water, and also microcontrollers, GSM, analog/digital converters. The receiver station consists of GSM module for receiving the sensing data from transmitter through GSM network. The receiver station receives data through the com port and stores in PC in order to achieve human-computer interface. The graphical user interface was designed, so that farmers and investigators can observe, investigate and analyze the related data. The user interface allows us to convey the analyzed data in the form of a message to the farmers in their respective local languages to their Mobile Phones and alerts them in unhygienic environmental conditions. With this even semi-literate farmers can interact with the system and can understand the information in order to take suitable actions.

**2.3 Water Quality Monitoring System Based on IOT:**

Water pollution is one of the biggest fears for the green globalization. In order to ensure the safe supply of the drinking water the quality needs to be monitor in real time. In this paper we present a design and development of a low cost system for real time monitoring of the water quality in IOT(internet of things).The system consist of several sensors is used to measuring physical and chemical parameters of the water. The parameters such as temperature, PH, turbidity, flow sensor of the water can be measured. The measured values from the sensors can be processed by the core controller. The arduino model can be used as a core controller. Finally, the sensor data can be viewed on internet using WI-FI system.

**2.4 Knowledge Based Real Time Monitoring System for Aquaculture Using IoT:**

Internet of things is one of the rapidly growing fields for delivering social and economic benefits for emerging and developing economy. The field of IOT is expanding its wings in all the domains like medical, industrial, transportation, education, mining etc. Now-a-days with the advancement in integrated on chip computers like arduino, Raspberry pi the technology is reaching the ground level with its application in agriculture and aquaculture. Water quality is a critical factor while culturing aquatic organism.

**CHAPTER 3**

**DESIGN OF AQUAMONITORING SYSTEM**

**3.1 Existing system**

**3.1.1Manual testing :**

To gain more information on particular body of water, AES conducts a water quality analysis to determine several key water quality parameters. Based of these results, AES can then implement specific recommendations in order to improve your water quality.



Fig. 3.1: Collecting water samples for quality measurement

pH is a measure of whether water is acidic or basic. Fish have an average blood pH of 7.4, so pond water with a pH close to this is optimum. An acceptable range would be 6.5 to 9.0. Fish can become stressed in water with a pH ranging from 4.0 to 6.5 and 9.0 to 11.0. Fish growth is limited in water pH less than 6.5, and reproduction ceases and fry can die at pH less than 5.0. Death is almost certain at a pH of less than 4.0 or greater than 11.0. Pond water pH fluctuates throughout the day due to photosynthesis and respiration by plants and vertebrates. Typically, pH is highest at dusk and lowest at dawn. This is because nighttime respiration increases carbon dioxide concentrations that interact with water producing carbonic acid and lowering pH. This can limit the ability of fish blood to carry oxygen.

The beginning aquaculturist usually underestimates the quantity of water required for commercial production. It is generally accepted that a minimum rate of 13 gallons per minute (gpm) is required for each surface acre of ponds. With this in mind, a 100- acre fish farm will need to have wells capable of producing 1,300 gpm of water. Such large volumes are required to replace water lost to evaporation and seepage. In addition, the farmer may have several ponds to fill quickly during the spawning season. In raceway culture, it is advisable to have a minimum flow rate of 500 gpm. Even water re-circulating systems that recycle water require large quantities of water. If a 100,000 gallon capacity water re-circulating operation exchanges 10 percent of the water daily, it will require 10,000 gallons of water per day.

**3.1.2 Drawbacks of manual testing:**

* Consumes more time.
* Parameters may vary with time.
* Not reliable.
* Accuracy is less.
* Does not provide real time monitoring.
* No scope for generation of early warning.
* Increases burden on farmer.
* No availability of proper equipment.

To overcome the above problems the project real time monitoring system for aquaculture was implemented.

**3.2 Proposed system**

**3.2.1 Objective:**

The main objective of the proposed system is to design a smart monitoring and control device for aquaculture. At present aqua farmers are depending on manual testing to know the quality of water. This will consume time and inaccurate because water quality parameters may alter with time. In order to overcome this problem, technology should be brought to aqua culture which increases the productivity and minimize the losses by constant monitoring of water quality parameters.

The proposed work uses an integrated on chip computer arduino mega 2560 controller. Several sensors are mounted to sense the data and the data is transferred to the aqua farmer through GSM. This is reliable and reduces the risk for aquatic life.

**3.2.2Block Diagram:**

POWER

SUPPLY

ULTRASONIC

SENSOR

PH

SENSOR

GSM

MODULE

ARDUINO

MEGA

2560

TURBIDITY

SENSOR

MOBILE

LCD

Fig.3.2 Block diagram of proposed system

**3.2.3 Flow chart:**

START

INITIALIZE THE ARDUINO

SENSED DATA FROM SENSORS IS SENT TO ARDUINO

YES

CHECK WHETHER THE PARAMETERS ARE WITHIN THE THRESHOLD LEVEL

NO

SEND ALERT MESSAGE TO USER MOBILE

Fig.3.3 Flowchart of proposed system

**3.2.4Working:**

The GSM module is connected to the arduino micro controller. The sensors required to measure the water parameters are connected to the analog pins of the arduino micro controller. In the arduino IDE software ,the code is written in embedded C language .In the code the threshold values are set . Mobile number to which the message has to be sent is specified.

The libraries required for the modules are downloaded and included in the arduino software. Update the drivers for the arduino mega board. The code is compiled and then uploaded to the arduino board. Insert the SIM in the GSM module. If the LED on the GSM blinks once in a second then it indicates that the module is searching for a network. If the LED blinks once for three seconds then it indicates that the network has been established. If the LED blinks twice in a second then it indicates that it is connected to gprs.

Place the sensors in the water. When the parameters of water vary when compared with the threshold values then the arduino sends commands to the GSM module. It then establishes a network and sends message to the mobile number which is specified in the code.

**3.2.5Tools Required:**

**3.2.5.1Hardware Tools:**

* + - Arduino Mega 2560
    - GSM module
    - Ultrasonic sensor
    - pH sensor
    - Turbidity sensor
    - LCD display

**3.2.5.2Software Tools:**

* + - Arduino IDE

**3.2.6Applications:**

* Aquaponics
* Fish ponds
* Shrimp breeding
* Aquariums
* Used to monitor the quality of water in wells
* To check the purity of drinking water
* To evaluate the safe drinking water content in water scarce regions

**3.2.7Advantages:**

* Healthy growth of fish
* Aquaponics grow well
* Accurate
* Less cost
* Time consumption
* Informs on mobile
* Improves reliability
* Real time monitoring
* Generates early warning
* Simple circuit

**3.2.8 Tables related to water quality parameters:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Water Quality Parameters** | **Range** |
| 1 | Dissolved Oxygen(DO) | (4-10) ppm |
| 2 | Ammonia | * + 1. Ppm |
| 3 | pH | (7.5-8.5) ppm |
| 4 | Temperature | 22-33 c |
| 5 | Salt | (0-2) ppt |
| 6 | Carbonates | (20-40) ppm |
| 7 | Bicarbonates | (150-300) ppm |
| 8 | Nitrates | * + 1. Ppm |
| 9 | Sour gas | * + 1. Ppm |

Table 3.1 Water quality parameters with threshold range

|  |  |  |
| --- | --- | --- |
| **Water Quality Parameter** | **Below Threshold** | **Above Threshold** |
| Ph | 1.Agriculture lime  2. pvc lime | 1.Sugar  2.Lemon salt  3.Gypsum  4.Probiotics |
| Salt | Impractical | Pumping fresh water into the tank |
| Hardness | Impractical | EDTA solution 1kg per acre |
| Ammonia | Impractical | Fresh water into the pond maintaining high DO, lowering feed rates |
| Unionized Ammonia | Impractical | Geolites (20 kg per acre), yucca products(300 gm per acre) |
| Nitrate | Impractical | Salt is used |
| Dissolved Oxygen | Turning on aerators, oxygen pills | Impractical |

Table 3.2 Causes and effects of water quality parameters

**CHAPTER 4**

**HARDWAREAND SOFTWARE DESCRIPTION**

**4.1Arduino Mega 2560**



Fig .4.1Arduino Mega 2560

**4.1.1 Overview:**

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), 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 a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the ArduinoDuemilanove or Diecimila.

**4.1.2 Power:**

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The Mega2560 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.

The power pins are as follows

The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

**4.1.3 Memory:**

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM

**4.1.4 Input and Output:**

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode() , digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

● Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

● External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

● PWM: 0 to 13. Provide 8-bit PWM output with the analogWrite() function.

● SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

● LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

● I2C: 20 (SDA) and 21 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function. There are a couple of other pins on the board:

● AREF. Reference voltage for the analog inputs. Used with analogReference()

. ● Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button

**4.1.5 Communication :**

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Mega2560's digital pins. The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. For SPI communication, use the SPI library.

**4.1.6 Programming:**

The Arduino Mega can be programmed with the Arduino software (download). For details, see the reference and tutorials. The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In Circuit Serial Programming) header; see these instructions for details

**4.2GSM module**



Fig.4.2 GSM module

SIM800L is a miniature cellular module which allows for GPRS transmission, sending and receiving SMS and making and receiving voice calls. Low cost and small footprint and quad band frequency support make this module perfect solution for any project that require long range connectivity. After connecting power module boots up, searches for cellular network and login automatically. On board LED displays connection state (no network coverage - fast blinking, logged in - slow blinking).

This module have two antennas included. First is made of wire (which solders directly to NET pin on PCB) - very useful in narrow places. Second - PCB antenna - with double sided tape and attached pigtail cable with IPX connector. This one have better performance and allows to put your module inside a metal case - as long the antenna is outside.

**4.2.1 Specification:**

* Supply voltage: 3.8V - 4.2V
* Recommended supply voltage: 4V
* Power consumption:
* sleep mode < 2.0mA
* idle mode < 7.0mA
* GSM transmission (avg): 350 mA
* GSM transmission (peek): 2000mA
* Module size: 25 x 23 mm
* Interface: UART (max. 2.8V) and AT commands
* SIM card socket: microSIM (bottom side)
* Supported frequencies: Quad Band (850 / 950 / 1800 /1900 MHz)
* Antenna connector: IPX
* Status signaling: LED
* Working temperature range: -40 do + 85 ° C

**4.2.2 Components in the module:**

* SIM800L module
* goldpin headers
* wire antenna
* PCB antenna with pigtail and IPX connector

**4.2.3 Pin configuration:**

* RING (not marked on PBC, first from top, square) - LOW state while receiving call
* DTR - sleep mode. Default in HIGH state (module in sleep mode, serial communication disabled). After setting it in LOW the module will wake up.
* MICP, MICN - microphone (P + / N -)
* SPKP, SPKN - speaker (P + / N -)
* NET - antenna
* VCC - supply voltage
* RESET - reset
* RXD - serial communication
* TXD - serial communication
* GND - ground

**4.3 Ultrasonic sensor**



Fig.4.3 Ultrasonic sensor

**4.3.1 Sensor Features:**

* Operating voltage: +5V
* Theoretical  Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

**4.3.2 Working:**

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

**Distance = Speed × Time**

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

**4.3.3 Sensor usage:**

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

  Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

**4.3.4 Applications:**

* Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
* Used to measure the distance within a wide range of 2cm to 400cm
* Can be used to map the objects surrounding the sensor by rotating it
* Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water.

**4.4 Turbidity sensor**



Fig.4.4 Turbidity sensor

**4.4.1 Introduction:**

The turbidity sensor detects water quality by measuring the levels of turbidity. It uses light to detect suspended particles in water by measuring the light transmittance and scattering rate, which changes with the amount of total suspended solids (TSS) in water. As the TTS increases, the liquid turbidity level increases. Turbidity sensors are used to measure water quality in rivers and streams, wastewater and effluent measurements, control instrumentation for settling ponds, sediment transport research and laboratory measurements. This sensor provides analog and digital signal output modes. The threshold is adjustable when in digital signal mode. You can select the mode according to your MCU.

**4.4.2 Interface Description:**

1. "D/A" Output Signal Switch

2. "A": Analog Signal Output, the output value will decrease when in liquids with a high turbidity

3. "D": Digital Signal Output, high and low levels, which can be adjusted by the threshold potentiometer

4. Threshold Potentiometer: you can change the trigger condition by adjusting the threshold potentiometer in digital signal mode.

**4.4.3 Operation:**

The sensor operates on the principle that when light is passed through a sample of water, the amount of light transmitted through the sample is dependent on the amount of soil in the water. As the soil level increases, the amount of transmitted light decreases. The turbidity sensor measures the amount of transmitted light to determine the turbidity of the wash water. These turbidity measurements are supplied to the washer controller, which makes decisions on how long to wash in all the cycles. These decisions are made based on a comparison between clean water measurements (taken at the beginning of the wash cycle) and the wash water turbidity measurement taken at the end of each wash cycle. By measuring the turbidity of the wash water, the washing machine can conserve energy on lightly soiled loads by only washing as long as necessary. This will result in energy savings for the consumer.

**4.5 pH sensor**

****

Fig.4.5 pH sensor

**4.5.1 Introduction:**

The Model PHE-45P pH Sensor measures the pH of aqueous solutions in industrial and municipal process applications. It is designed to perform in the harshest of environments, including applications that poison conventional pH sensors. All seals are dual o-ring using multiple sealing materials. The sensor is designed for use with the Omega PHTX-45 Monitor/Analyzer.

**4.5.2 Sensor Features:**

• A high volume, dual junction saltbridge is utilized to maximize the in-service lifetime of the sensor. The annular junction provides a large surface area to minimize the chance of fouling. Large electrolyte volume and dual reference junctions minimize contamination of the reference solution. The saltbridge is replaceable.

• The reference element of the sensor is a second glass pH electrode immersed in a reference buffer solution. This glass reference system greatly increases the range of sensor applications.

• An integral preamplifier is encapsulated in the body of the sensor. This creates a low impedance signal output which ensures stable readings in noisy environments and increases the maximum possible distance between sensor and transmitter to 3,000 feet (914 meters).

• System diagnostics warn the user in the event of electrode breakage, loss of sensor seal integrity or integral temperature element failure.

• Pt1000 RTD. The temperature element used in the PHE-45P sensor is highly accurate and provides a highly linear output.

**4.5.3 Important Notes:**

1. The PHE-45P process electrode is made of glass and can break if not handled properly. Should the electrode ever break, USE CAUTION when handling the sensor to avoid serious cuts.

2. The glass electrode must be wetted at all times to ensure proper functionality. PHE-45P sensors are shipped with a fluid-filled cap over the electrode to enable immediate use (remove cap before installing, save for storage and shipping purposes). Electrodes that have dried out for any reason should be hydrated for 24 hours to restore full functionality.

3. Hydrofluoric acid (HF) will dissolve conventional glass electrodes. Please contact the factory when the process application involves this or any other questionable substance.

The Model PHE-45P Sensor has a built-in preamplifier and comes standard with 15 feet of 6 conductor (only 5 are used) double shielded cable. The cable is permanently attached to the sensor, and a PEEK cordgrip is used to seal around the cable. Nevertheless, the cable should always be kept as clean and dry as possible.

Take care to route sensor cable away from AC power lines, adjustable frequency drives, motors, or other noisy electrical signal lines. Do not run signal lines in the same conduit as AC power lines. Run signal cable in dedicated metal conduit if possible. For optimum electrical noise protection, run an earth ground wire to the ground terminal in the transmitter.

### 4.6 LCD Module (2X16 character):

Dot matrix LCD modules is used for display the parameters and fault condition.16 characters 2 lines display is used. It has controller which interface data’s and LCD panel. Liquid crystal displays (LCD’s) have materials, which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal. An LCD consists of two glass panels, with the liquid crystal material sandwiched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed polymeric layers are present in between the electrodes and the liquid crystal molecules to maintain a defined orientation angle.

One each polarizer’s are pasted outside the two glass panels. These polarizer’s would rotate the light rays passing through them to a definite angle, in a particular direction When the LCD is in the off state, light rays are rotated by the two polarizes and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned on a specific direction.

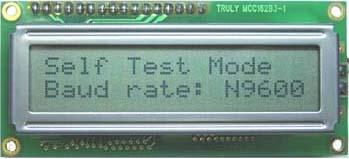


Fig.4.6 LCD Diagram

The LCD's are lightweight with only a few millimeters thickness. since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations .

Each [pixel](https://en.wikipedia.org/wiki/Pixel) of an LCD typically consists of a layer of [molecules](https://en.wikipedia.org/wiki/Molecule) aligned between two [transparent](https://en.wikipedia.org/wiki/Transparency_(optics)) [electrodes](https://en.wikipedia.org/wiki/Electrode), and two [polarizing](https://en.wikipedia.org/wiki/Polarizer) [filters](https://en.wikipedia.org/wiki/Filter_(optics)) (parallel and perpendicular), the axes of transmission of which are (in most of the cases) perpendicular to each other. Without the [liquid crystal](https://en.wikipedia.org/wiki/Liquid_crystal) between the polarizing filters, light passing through the first filter would be blocked by the second (crossed) polarizer. Before an [electric field](https://en.wikipedia.org/wiki/Electric_field) is applied, the orientation of the liquid-crystal molecules is determined by the alignment at the surfaces of electrodes. In a twisted nematic (TN) device, the surface alignment directions at the two electrodes are perpendicular to each other, and so the molecules arrange themselves in a [helical](https://en.wikipedia.org/wiki/Helix) structure, or twist. This induces the rotation of the polarization of the incident light, and the device appears gray. If the applied voltage is large enough, the liquid crystal molecules in the center of the layer are almost completely untwisted and the polarization of the [incident light](https://en.wikipedia.org/wiki/Incident_light) is not rotated as it passes through the liquid crystal layer. This light will then be mainly polarized [perpendicular](https://en.wikipedia.org/wiki/Perpendicular) to the second filter, and thus be blocked and the [pixel](https://en.wikipedia.org/wiki/Pixel) will appear black. By controlling the voltage applied across the liquid crystal layer in each pixel, light can be allowed to pass through in varying amounts thus constituting different levels of gray. Color LCD systems use the same technique, with color filters used to generate red, green, and blue pixels

**4.7 Arduino IDE:**

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

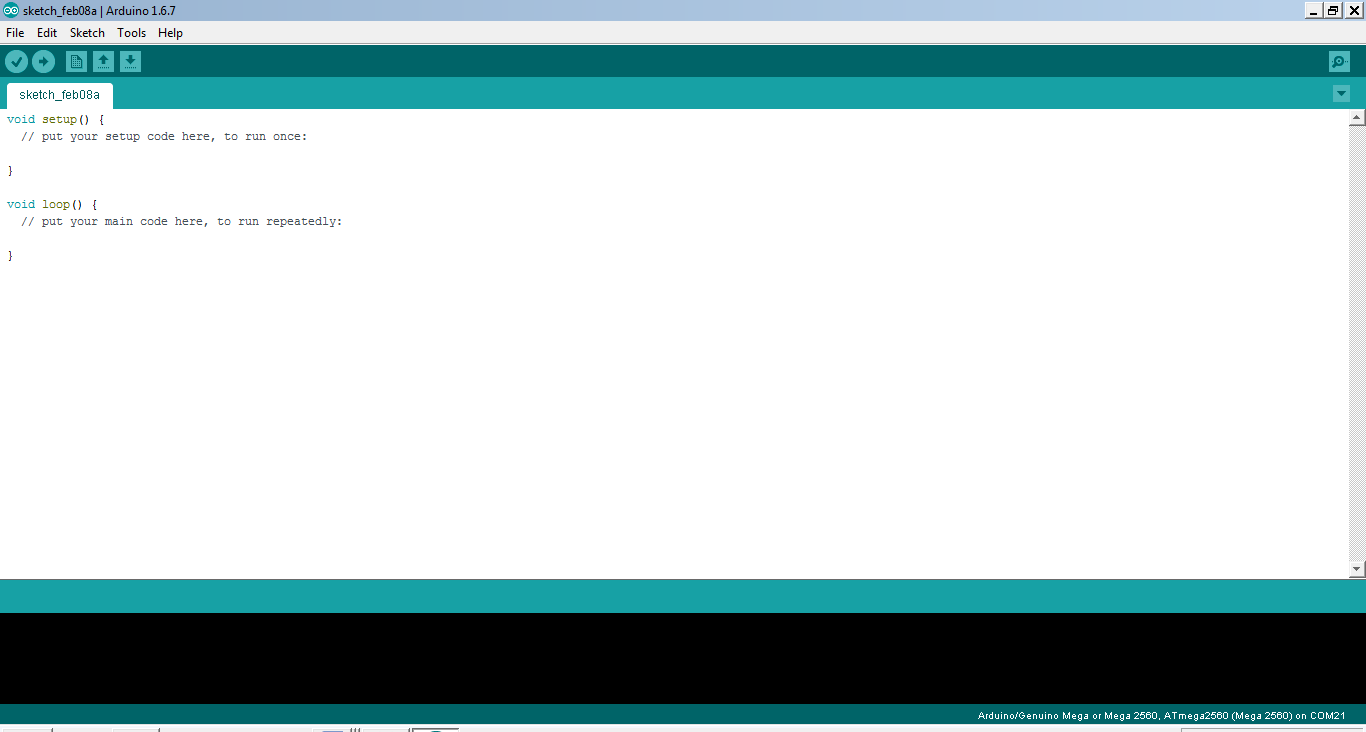


Fig.4.7Arduino IDE page

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an “Arduino language.” However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device.

To begin, download the Arduino IDE from [the Arduino website](http://arduino.cc/en/Main/Software). Make sure to select the right version for your Operating System (OS). For a full getting started guide for each OS, please refer to [the Arduino guide.](http://arduino.cc/en/Guide/HomePage) Once the arduino.zip file has been downloaded, extract the file to a folder somewhere on your computer. There is no install - simply open the folder and double click the .exe.

**4.7.1 Connecting the arduino:**

Connecting an Arduino board to your PC is quite simple. On Windows:

* Plug in the USB cable - one end to the PC, and one end to the Arduino board.
* When prompted, select "Browse my computer for driver" and then select the folder to which you extracted your original Arduino IDE download.
* You may receive an error that the board is not a Microsoft certified device - select “Install anyway.”
* Your board should now be ready for programming.

When programming your Arduino board it is important to know what COM port the Arduino is using on your PC. On Windows, navigate to Start->Devices and Printers, and look for the Arduino. The COM port will be displayed underneath.

Alternatively, the message telling you that the Arduino has been connected successfully in the lower-left hand corner of your screen usually specifies the COM port is it using.

**4.7.2 Preparing the board:**

Before loading any code to your Arduino board, you must first open the IDE. Double click the Arduino .exe file that you downloaded earlier. A blank program, or "sketch," should open.

The Blink example is the easiest way to test any Arduino board. Within the Arduino window, it can be found under File->Examples->Basics->Blink.

Before the code can be uploaded to your board, two important steps are required.

**1.** Select your Arduino from the list under Tools->Board. The standard board used in RBE 1001, 2001, and 2002 is the Arduino Mega 2560, so select the "Arduino Mega 2560 or Mega ADK" option in the dropdown.

**2.** Select the communication port, or COM port, by going to Tools->Serial Port.

If you noted the COM port your Arduino board is using, it should be listed in the dropdown menu. If not, your board has not finished installing or needs to be reconnected.

**4.7.3 Loading the code:**

The upper left of the Arduino window has two buttons: A checkmark to Verify your code, and a right-facing arrow to Upload it. Press the right arrow button to compile and upload the Blink example to your Arduino board.

The black bar at the bottom of the Arduino window is reserved for messages indicating the success or failure of code uploading. A "Completed Successfully" message should appear once the code is done uploading to your board. If an error message appears instead, check that you selected the correct board and COM port in the Tools menu, and check your physical connections.

If uploaded successfully, the LED on your board should blink on/off once every second. Most Arduino boards have an LED prewired to pin 13.

It is very important that you do not use pins 0 or 1 while loading code. It is recommended that you do not use those pins ever.

Arduino code is loaded over a serial port to the controller. Older models use an [FTDI](http://www.ftdichip.com/) chip which deals with all the USB specifics. Newer models have either a small AVR that mimics the FTDI chip or a built-in USB-to-serial port on the AVR micro-controller itself.

**4.8 Embedded C programming**

**4.8.1 Introduction to Embedded C programming:**

Looking around, we find ourselves to be surrounded by various types of [embedded systems](http://www.engineersgarage.com/articles/embedded-systems). Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor functioning inside it. Associated with each processor is the embedded software. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul. It is the embedded software which primarily governs the functioning of embedded systems.

During infancy years of microprocessor based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check correct execution of the program. Some ‘very fortunate’ developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well.

As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.

  Initially C was developed by Kernighan and Ritchie to fit into the space of 8K and to write (portable) operating systems. Originally it was implemented on UNIX operating systems. As it was intended for operating systems development, it can manipulate memory addresses. Also, it allowed programmers to write very compact codes. This has given it the reputation as the language of choice for hackers too.

As assembly language programs are specific to a processor, assembly language didn’t offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn’t find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Even though C might have lost its sheen as mainstream language for general purpose applications, it still is having a strong-hold in embedded programming. Due to the wide acceptance of C in the embedded systems, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of embedded systems using C.

Subsequent sections will discuss what is Embedded C, features of C language, similarities and difference between C and embedded C, and features of embedded C programming.

**4.8.2 Embedded systems programming:**

Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows:

·          Embedded devices have resource constraints(limited ROM, limited RAM, limited stack space, less processing power)

·         Components used in embedded system and PCs are different; embedded systems typically uses smaller, less power consuming components. Embedded systems are more tied to the hardware.

Two salient features of Embedded Programming are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language.  Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of languages:

         1. Machine Code

         2. Low level language, i.e., assembly

         3. High level language like C, C++, Java, Ada, etc.

         4. Application level language like Visual Basic, scripts, Access, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

**4.8.3 Difference between C programming and Embedded C programming:**

Though C and embedded C appear different and are used in different contexts, they have more similarities than the differences. Most of the constructs are same; the difference lies in their applications.

C is used for desktop computers, while embedded C is for microcontroller based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory. However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code exceeds the limit, the system is likely to crash.

Compilers for C (ANSI C) typically generate OS dependantexecutables. Embedded C requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications.

Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications.

Embedded systems often do not have a console, which is available in case of desktop applications.

So, what basically is different while programming with embedded C is the mindset; for embedded applications, we need to optimally use the resources, make the program code efficient, and satisfy real time constraints, if any. All this is done using the basic constructs, syntaxes, and function libraries of ‘C’.

**4.8.4 Programming using Embedded C:**

Embedded C use most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch. case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc. In addition, there are some specifics to embedded C which are mentioned below:

Embedded programming requires access to underlying hardware, i.e., timers, memory, ports, etc. In addition, it is often needed to handle interrupts, manage job queues, etc. As C offers pointers and bit manipulation features, they are extensively used for direct hardware access.

**CHAPTER 5**

**RESULTS AND DISCUSSION**

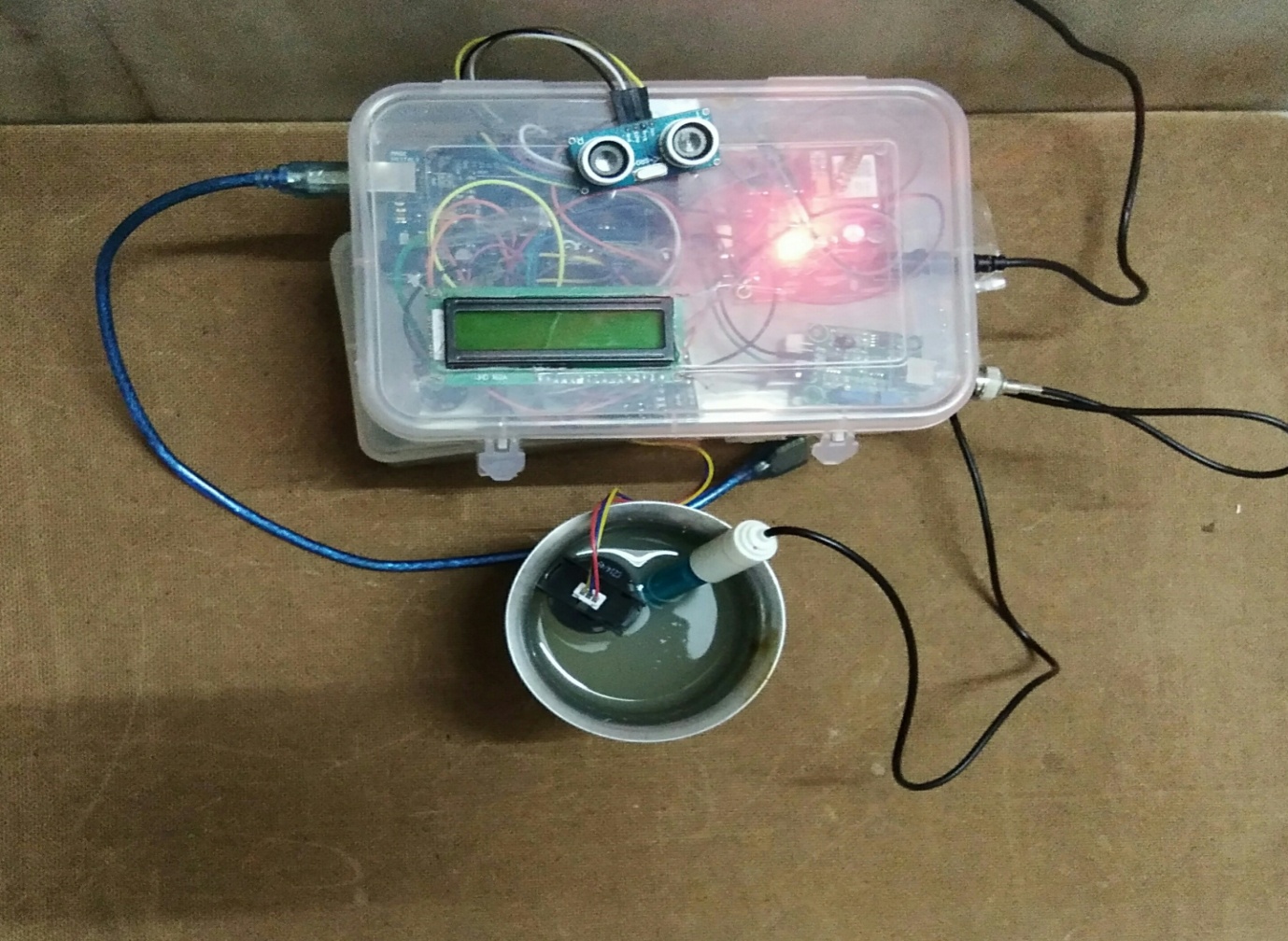
****

Fig.5.1Required setup

The required setup shows a container with water, arduino mega board, LCD, GSM module, pH sensor, ultrasonic sensor and turbidity sensor. pH sensor and turbidity sensor are placed in the water. The ultrasonic sensor is placed above the water level. LCD is used to display the values of different parameters measured by the sensors.

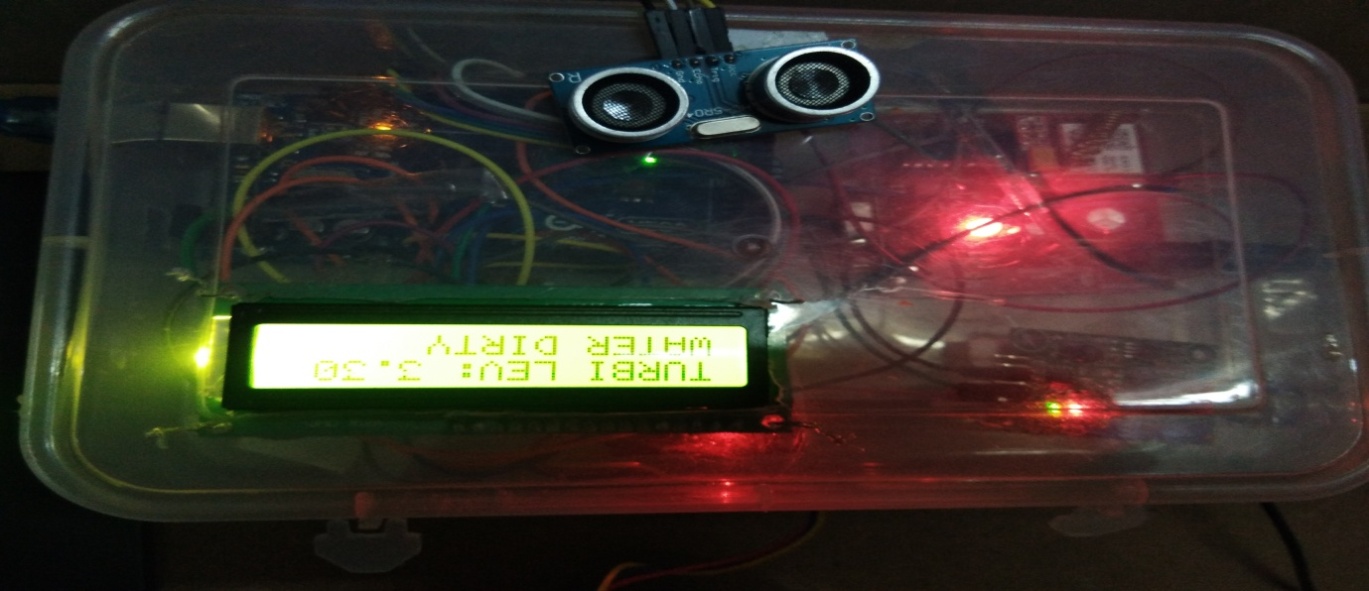


Fig.5.2 Displaying turbidity value

LCD shows the turbidity value of water. If the turbidity value is less than 3.5 then the water is considered to be dirty. Turbidity is the cloudiness or haziness of a fluid caused by large numbers of individual particles that are generally invisible to the naked eye, similar to smoke in air. The significance of excessive turbidity in water on fish and other aquatic life begins by modifying the temperature structure of lakes. Bottom temperatures are generally lower in turbid lakes or ponds than in clear ones. In many lakes, lower temperature means lower productivity. Turbidity also interferes with the penetration of light. This reduces photosynthesis and thereby decreases the primary productivity upon which the fish food organisms depend. As a consequence fish production is reduced.

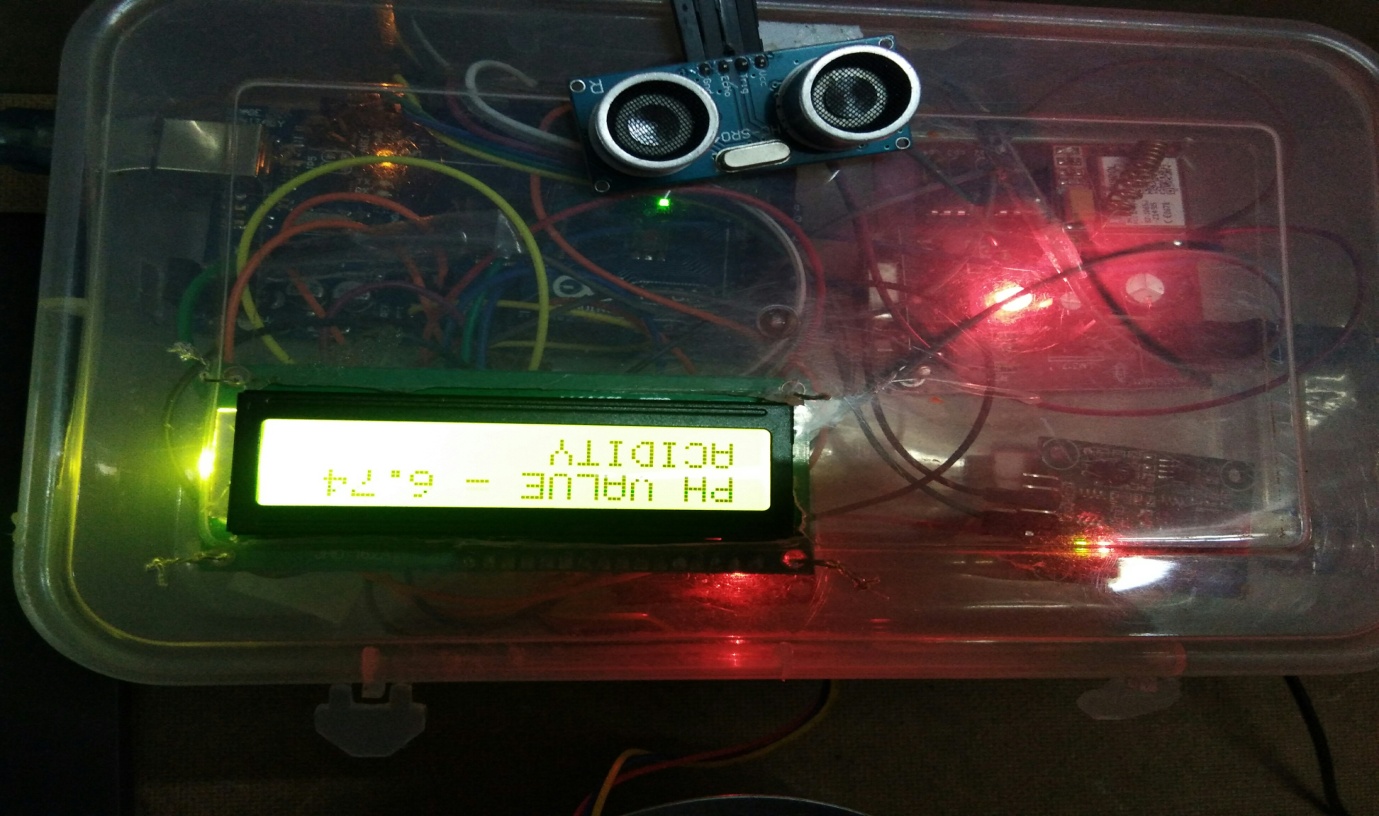


Fig.5.3 Displaying pH value

The LCD displays the pH value. If the pH value is less than 6.8 then the water is considered as acidic. pH is a logarithmic scale used to specify the acidity or basicity of an aqueous solution. It is approximately the negative of the base 10 logarithm of the molar concentration, measured in units of moles per liter, of hydrogen ions. pH stands for ‘power of hydrogen’. Acidic water robs fish and other aquatic species of sodium in the blood and oxygen in the tissues. Additionally, it affects the functioning of fish gills. acidic water also poisons fish eggs, as they will not hatch if water pH is too low

.



Fig.5.4Displaying water level

The LCD displays the water level. If the distance between the ultrasonic sensor and the water surface is more than 10 cm then the water level is considered to be low .When the water level is low then there will be a decrement in the dissolved oxygen content. Adequate dissolved oxygen is necessary for good water quality .Oxygen is a necessary element to all forms of life. Natural stream purification processes require adequate oxygen levels in order to provide for aerobic life forms, as dissolved oxygen levels in water drop below 5.0 mg/ll, aquatic life is put under stress. The lower the concentration, the greater the stress. Oxygen levels that remain below 1-2 mg/l for a few hours can result in large fish kills

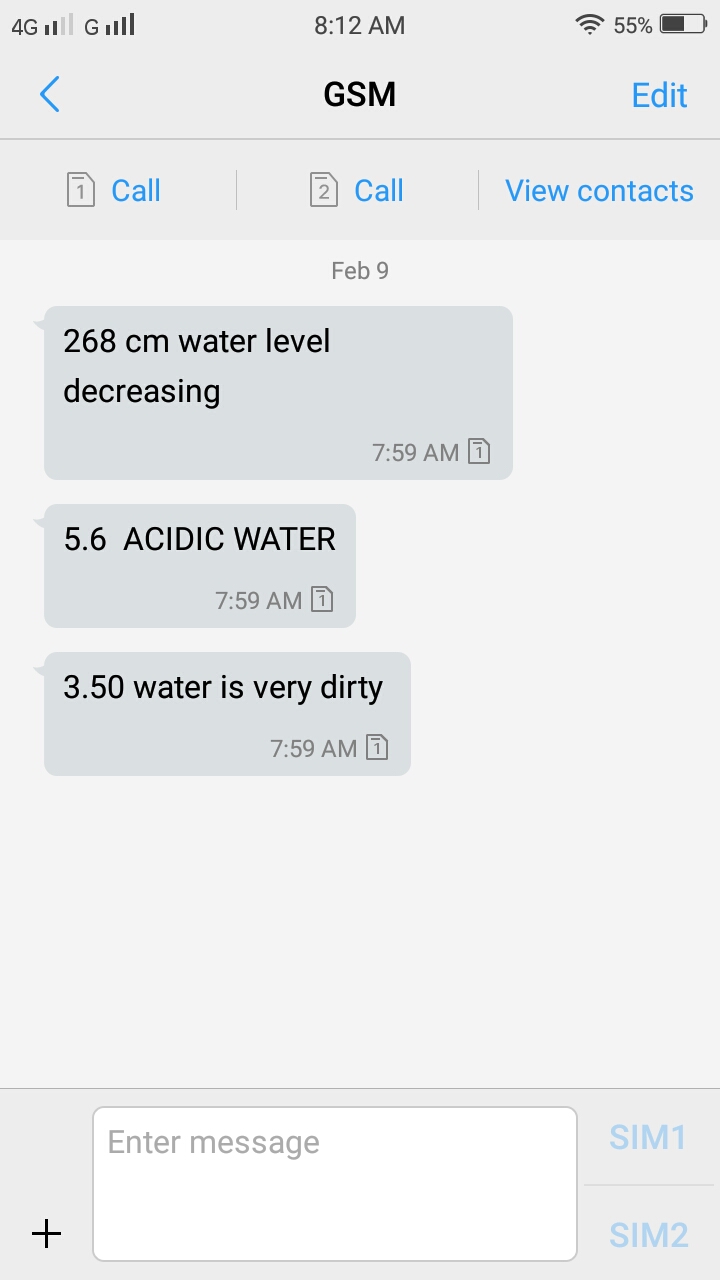


Fig.5.5 Message in user mobile

The above figure shows the message which is sent through the GSM module to the user mobile. When there is a variation in the parameters the GSM sends the warning message to the user.

**CONCLUSION**

The project, implementation of real time monitoring system for aquaculture helps the farmers for accurate and reliable monitoring of water quality parameters. The real time monitoring helps to take pro-active measures before necessary damage was done. Though the initial cost is high there will be no additional cost and maintenance once it gets installed. Further there is no need for manual testing periodically. It saves time and energy. The project is used for reducing the risk from climatic fluctuations and ensures growth and health for aquatic life.

**FUTURE SCOPE**

Future works should be enhancing the system remote access to the sensor nodes using internet and data transmission for further analysis. More network performance metrics need to be studied and evaluated to make the system more robust and scalable. Water quality monitoring with a larger network is envisaged for grossly polluted river stretches. More parameters can be taken into account, ,thereby further improving the process of examining the water resource.

**REFERENCES**

[1] S.Kayalvizhi, Koushik Reddy G, Vivek Kumar P, VenkataPrasanth N, “Cyber Aqua Culture Monitoring System Using ArdunioAnd Raspberry Pi,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 4, Issue 5,Pg:2320- 3765; May 2015.

[2] Suresh BabuChandanapalli,Sreenivasa Reddy E and Rajya Lakshmi D, “Design and Deployment of Aqua Monitoring System Using Wireless Sensor Networks and IAR-Kick,” Jouranl of Aquaculture Research &Development ,Volume 5 • Issue 7 • 1000283

[3] Daudi S. Simbeye and Shi Feng Yang, “Water Quality Monitoring and Control for Aquaculture Based on Wireless Sensor Networks,” JOURNAL OF NETWORKS, VOL. 9, NO. 4, APRIL 2014.

[4] Changhui Deng, YanpingGao, Jun Gu, Xinying Miao, “Research on the Growth Model of Aquaculture Organisms Based on Neural Network Expert System,” Sixth International Conference on Natural Computation (ICNC 2010) ; pg.no 1812-1815, SEPTEMBER 2010.

[5] Wen-TsaiSung,Jui-HoChen,Da-ChiunHuang,YiHaoJu,“MultisensorsRealtime Data Fusion Optimization for IOT Systems,” 2014 IEEE International Conference on Systems, Man, and Cybernetics October 5-8, 2014, San Diego,CA,USA.

[6] WeerasakCheunta, NitthitaChirdchoo and KanitthaSaelim, “Efficiency Improvement of an Integrated Giant Freshwater-White Prawn Farming in Thailand Using a Wireless Sensor Network,” Research Gate,DECEMBER2014.

[7] Jui-Ho Chen, Wen-Tsai Sung and Guo-Yan Lin, “Automated Monitoring System for the FishFarm Aquaculture Environment,” 2015 IEEE International Conference on Systems, Man, and Cybernetics.

[8] Mr. KiranPatil,Mr. SachinPatil,Mr. SachinPatil and Mr. VikasPatil,“Monitoring of Turbidity,pH& Temperature of Water Based on GSM,”International Journal for Research in Emerging Science and Technology,Volume-2,Issue-3,March-2015.

[9] Pradeep Kumar M,MonishaJ,Pravenisha R, Praiselin V, Suganya Devi K, “The Real Time Monitoring of Water Quality in IoT Environment,” International Journal of Innovative Research in Science,Engineering and Technology,Vol -5,Issue-6, March-2016.

[10] AkankshaPurohit, UlhasKumarGokhale,“Real Time Water Quality Measurement System based on GSM,” IOSR Journal of Electronics and CommnicationEngineering,Vol. 9,Issue 3,PP 63-67,May-2014.

[11] SharudinMohdShahrulZharif, “ IntillegentAquafarm System via SMS,” Diss. UniversitiTeknologi PETRONAS,2007.

[12] SheetalIsrani ,HarshalMeharkure , ParagYelore, “Application of IoT based System for Advance Agriculture in India,”International Journal of Innovative research in Computer and Communication Engineering Vol. 3,Issue. 11,November 2015.

[13] NikeshGondchawar , Prof. Dr. R. S. Kawitkar, “IoT based smart Agriculture, ” International Journal of advanced research in Computer and Communication Engineering, Vol.5,Issue. 6,June 2016

**APPENDIX**

**CODE:**

#include <SoftwareSerial.h.>

#include <Wire.h>

#include <LiquidCrystal.h>

constintrs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystallcd(rs, en, d4, d5, d6, d7); //lcd pins assign

SoftwareSerialmySerial(22, 23); //gsm (RX,TX) in arduino

intsensorPin = A2; //turbidity pin

float volt;

floatntu;

floatround\_to\_dp( float in\_value, intdecimal\_place );

#define trigPin 9 //ultrasonic pins grey

#define echoPin 8 //white

int temp=0;

long duration, distance;

floatphValue;

constintanalogInPin = A0; // ph sensor pin

intsensorValue = 0;

unsigned long intavgValue;

float b;

intbuf[10];

void setup(void) {

Serial.begin (9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

lcd.begin(16, 2); //initiliselcd with num of coloums 16 ,by row 2.

lcd.clear(); //clears lcd just incase there is anytin been displayed

mySerial.begin(9600); //gsm serial port

delay(100);

}

void loop(){

ultra();

turbidity();

ph();

SendMessage();

SendMessage1();

//SendMessage2();

// SendMessage3();

//SendMessage4();

}

//////////////////////////ultrasonic loop//////////////

void ultra() {

Serial.println(" ");

lcd.begin(16, 2); //initiliselcd with num of coloums 16 ,by row 2.

lcd.clear(); //clears lcd just incase there is anytin been displayed

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("TAKING READINGS");

lcd.setCursor(0,1);

lcd.print("FROM ULTRA SENSOR");

delay(2000);

Serial.println("Taking Readings from ultra sonic Sensor");

digitalWrite(trigPin, LOW); // Added this line

delayMicroseconds(2); // Added this line

digitalWrite(trigPin, HIGH);

delayMicroseconds(10); // Added this line

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1;

if(distance<10 && temp==0)

{

lcd.clear( );

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("WATER LEVEL:");

lcd.setCursor(12,0);

//delay(1000);

lcd.print(distance);

lcd.setCursor(14,0);

Serial.print("DISTANCE: ");

Serial.print(distance);

lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0

lcd.print("FULL WATER");

Serial.println(" cm");

Serial.println("water is level");

delay(3000);

}

else if(distance>10 || distance<40)

{

lcd.clear();

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("WATER LEVEL:");

lcd.setCursor(12,0);

// delay(1000);

lcd.print(distance);

lcd.setCursor(14,0);

Serial.print("DISTANCE: ");

Serial.print(distance);

Serial.println(" cm");

Serial.println("LOW Water Level");

lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0

lcd.print("LOW WATER LEVEL");

SendMessage();

delay(3000);

temp=0;

}

}

////////////////////////////ultra message//////////////////////

voidSendMessage()

{

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+919505715233\"\r"); // Replace x with mobile number

delay(1000);

mySerial.print(distance);// The SMS text you want to send

mySerial.print(" cm water level decreasing" );// The SMS text you want to send

delay(100);

mySerial.println((char)26);// ASCII code of CTRL+Z

delay(1000);

}

////////////////////////////turbidity sensor////////////////////////

void turbidity()

{

Serial.println(" ");

lcd.begin(16, 2); //initiliselcd with num of coloums 16 ,by row 2.

lcd.clear(); //clears lcd just incase there is anytin been displayed

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("TAKING READINGS");

lcd.setCursor(0,1);

lcd.print("FROM TURBI SENSOR");

delay(1000);

Serial.println("Taking Readings from turbidity Sensor");

volt = 0;

for(inti=0; i<800; i++)

{

volt += ((float)analogRead(sensorPin)/1023)\*5;

}

volt = volt/800;

volt = round\_to\_dp(volt,1);

if(volt < 3.5)

{

ntu = 3000;

lcd.clear( );

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("TURBI LEV:");

lcd.setCursor(11,0);

lcd.print(volt);

lcd.setCursor(14,0);

lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0

lcd.print("WATER DIRTY");

Serial.print(" turbidity value: ");

Serial.println(volt);

//Serial.println(" V");

SendMessage1();

// Serial.print(ntu);

//Serial.println(" NTU");

Serial.println(" water is dirty");

delay(1000);

}else{

ntu = -1120.4\*square(volt)+5742.3\*volt-4353.8;

lcd.clear( );

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("TURBI LEV:");

lcd.setCursor(11,0);

lcd.print(volt);

lcd.setCursor(14,0);

Serial.print(" turbidity value: ");

Serial.println(volt);

// Serial.println(" V");

//Serial.print(ntu);

//Serial.println(" NTU");

lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0

lcd.print("WATER IS CLEAN");

Serial.println(" water is clean");

delay(1000);

}

delay(1000);

}

floatround\_to\_dp( float in\_value, intdecimal\_place )

{

float multiplier = powf( 10.0f, decimal\_place );

in\_value = roundf(in\_value \* multiplier ) / multiplier;

returnin\_value;

}

///////////////////////////////////////turbidity message///////////////////////////

void SendMessage1()

{

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+919505715233\"\r"); // Replace x with mobile number

delay(1000);

mySerial.print(volt);// The SMS text you want to send

mySerial.print(" water is very dirty " );// The SMS text you want to send

delay(1000);

mySerial.println((char)26);// ASCII code of CTRL+Z

delay(1000);

}

///////////////////ph sensor///////////////

voidph(){

lcd.begin(16, 2); //initiliselcd with num of coloums 16 ,by row 2.

lcd.clear(); //clears lcd just incase there is anytin been displayed

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("TAKING READINGS");

lcd.setCursor(0,1);

lcd.print("FROM PH SENSOR");

delay(2000);

Serial.println(" ");

Serial.println("Taking Readings from ph Sensor");

for(inti=0;i<10;i++)

{

buf[i]=analogRead(analogInPin);

delay(10);

}

for(inti=0;i<9;i++)

{

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++)

avgValue+=buf[i];

floatpHVol=(float)avgValue\*5.0/1024/6;

floatphValue = -5.70 \* pHVol + 21.34;

if (phValue>=7.30){

lcd.clear( );

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("PH VALUE =");

lcd.setCursor(11,0);

lcd.print(phValue);

lcd.setCursor(14,0);

delay(2000);

Serial.print("PH VALUE = ");

Serial.println(phValue);

// SendMessage2();

lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0

lcd.print("ALKALINE");

Serial.println("water is alkaline");

delay(1000);

}

else if (phValue< 6.89){

Serial.print("PH VALUE = ");

Serial.println(phValue);

SendMessage3();

lcd.clear( );

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("PH VALUE =");

lcd.setCursor(11,0);

lcd.print(phValue);

lcd.setCursor(14,0);

lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0

lcd.print("ACIDITY");

Serial.println("water acidity high");

delay(1000);

}

else if (phValue>= 6.90 &&phValue<= 7.29 ){

Serial.print("PH VALUE = ");

Serial.println(phValue);

// SendMessage4();

lcd.clear( );

lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0

lcd.print("PH VALUE =");

lcd.setCursor(11,0);

lcd.print(phValue);

lcd.setCursor(14,0);

lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0

lcd.print("NEUTRAL");

Serial.println("water is neutral");

delay(1000);

}

delay(1000);

}

//////////////////ph sensor message///////////////////

/\*void SendMessage2()

{

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+919505715233\"\r"); // Replace x with mobile number

delay(1000);

mySerial.print(phValue);// The SMS text you want to send

mySerial.print(" ALKALINE WATER" );// The SMS text you want to send

delay(100);

mySerial.println((char)26);// ASCII code of CTRL+Z

delay(100);

}\*/

void SendMessage3()

{

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+919505715233\"\r"); // Replace x with mobile number

delay(1000);

mySerial.print(phValue);// The SMS text you want to send

mySerial.print(" ACIDIC WATER" );// The SMS text you want to send

delay(100);

mySerial.println((char)26);// ASCII code of CTRL+Z

delay(100);

}

/\*void SendMessage4()

{

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+919505715233\"\r"); // Replace x with mobile number

delay(1000);

mySerial.print(phValue);// The SMS text you want to send

mySerial.print(" NEUTRAL WATER" );// The SMS text you want to send

delay(100);

mySerial.println((char)26);// ASCII code of CTRL+Z

delay(100); }\*/