WATER PH LEVEL MONITORING WITH IOT

A MINI PROJECT REPORT

Submitted by

KUSHINDER D	**180701122**
MAGESH K	**180701133**
MAKESHWARAN M	**180701134**
MOHAMAD NIBRAS	**180701141**
PRASHANTH S	**180701169**

In partial fulfilment for the award of the degree Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, THANDALAM

(An Autonomous Institution)

CHENNAI 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

MARCH 2021

BONAFIDE CERTIFICATE

Certified that this project report **"WATER PH LEVEL MONITORING"** is the bonafide work of **"KUSHINDER.D(180701122), MAGESH.K(180701133) ,MAKESHWARAN.M (180701134), MOHAMAD NIBRAS(180701141), PRASHANTH. S(180701169)",** carried out the under my supervision.

SIGNATURE

Dr.R.G.SAKTHIVELAN, M.E,PhD

HEAD OF DEPARTMENT

Professor,

Department of CSE,

Rajalakshmi Engineering College,

Chennai - 602 105.

SIGNATURE

Mr. S.SURESH KUMAR M.E

SUPERVISOR

Associate Professor,

Department of CSE,

Rajalakshmi Engineering College,

Chennai - 602 105.

Submitted to Project and Viva Examination held on_____.

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We are extremely grateful to our principal Dr.S.N.MURUGESAN, M.E., Ph.D., for giving us a valuable support and encouragement throughout the duration of this course. We wish to thank Dr. R.G.SAKTHIVELAN, M.E., Ph.D., Head of the Department, Department of Computer Science and Engineering, Rajalakshmi Engineering College, for extending all facilities to us to work on this project. We would like to express our sincere appreciation and gratitude to our supervisor and guide Mr.S.SURESH KUMAR Associate Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College for his guidance, constant encouragement, and support. His meticulous attention and creative thinking have beena source of inspiration for us throughout this project. We also extend our sincerethanks to all faculty members and supporting staff for their direct and indirect involvement in successful completion of the project. All endeavours over a long period can be successful only with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them. Above all, we express our heartfelt thanks to our parents and family members who have dedicated their life to our well-being.

TABLE OF CONENTS

3 ARCHITECTURE DESIGN

CHAPTER 1

INTRODUCTION

In the 21st century, there were lots of inventions, but at the same time were pollutions, global warming and so on are being formed, because of this there is no safe drinking water for the world's pollution. Nowadays, water quality monitoring in real time faces challenges because of global warming limited water resources, growing population, etc. Hence there is need of developing better methodologies to monitor the water quality parameters in real time[1]. The water quality parameters pH measures the of hydrogen ions. It shows the water is acidic or alkaline. Pure water has 7pH value, less than 7pH has acidic, more than 7pH has alkaline. The range of pH is 0-14 pH. For drinking purpose it should be 6.5-8.5pH. Turbidity measures the large number of suspended particles in water that is invisible. Higher the turbidity higher the risk of diarrheoa, collera. Lower the turbidity then the water is clean. Temperature sensor measures how the water is, hot or cold. Flow sensor measures the flow of water through flow sensor. The traditional methods of water quality monitor involves the manual collection of water samples from different locations.

1.1 OVERVIEW OF THE PROJECT

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.

1.2 OBJECTIVE OF THE PROJECT

Pure water is considered a neutral—located right in the middle of the scale—with a pH of 7. Unfortunately, not all water is pure and is rarely ever at this actual level. So why do the pH levels of water change? There are many contributing factors that can effect the acidity and alkalinity of water. The first and most prominent of these is the bedrock and soil composition in which the water is located. Depending on the rock type, acidity in the water might be neutralized. On the other hand, plant growth and organic material located near the water can actually increase acidity due to the emission of carbon dioxide when decomposition occurs. Other factors include the dumping of chemicals, acid precipitation, and coal mine drainage.

pH levels in drinking water are monitored extremely closely and are required to be kept in a range of 6.5–8.5 by the Environmental Protection Agency (EPA). Though, if not regulated carefully, the effects of drinking acidic or alkaline water could be harmful. When water is below a pH level of 7, it has a corrosive quality to it. This means it could contain iron, copper, lead, or zinc from plumping and various other metal fixtures. It will have a bitter and metallic taste to it. High alkaline water doesn't pose health risks but does cause aesthetic problems. Formation of scale or precipitate on piping, fixtures, dishes, and utensils will occur. The taste will have a baking soda-like taste and will have a slippery feel.

1.3 PURPOSE OF THE PROJECT

The purpose of this project is to design and implement such a system that detects any liquid with a pH level not advised to have and report it to the owner of machine or the users. The detection system would help reduce fatalities or health issues due to water diseases by giving immediate results. In order to further reduce response time, usage of Wi-Fi is done which will help scale back response time and therefore reduce giving results much more faster..

In this project we are utilizing Arduino Uno ,sensors like pH sensor(to detect pH value in analog & NodeMCU(to supply the values from Arduino to our smartphone) in seconds. This will help users to know if the liquid has the perfect pH.

1.4 SCOPE OF THE PROJECT

The scope lies in the optimal time taken for the pH detection and the value to be sent immediately to the users and the owner. The optimality is got by using the online message gateway. This helps to overcome the delay and errors caused due to remote and connectivity less regions of the world.

The major focus is on the pH sensors. These determine the respective thresholds, which in turn governs the water purity level,.

The scope of the project also lies in creating a multipurpose algorithm that not only detects pH Level but additionally also alerts or sends the value to users and the owner. The in-built system algorithm is looking at an efficient which are accurate and quick.

CHAPTER 2

SYSTEM OVERVIEW

2.1 HARDWARE COMPONENT DETAILS

Arduino Uno:

The Arduino Uno is one kind of microcontroller board based on ATmega328, and Uno is an Italian term which means one. Arduino Uno is named for marking the upcoming release of microcontroller board namely Arduino Uno Board 1.0. This board includes digital I/O pins-14, a power jack, analog i/ps-6, ceramic resonator-A16 MHz, a USB connection, an RST button, and an ICSP header. All these can support the microcontroller for further operation by connecting this board to the computer. The power supply of this board can be done with the help of an AC to DC adapter, a USB cable, otherwise a battery. This article discusses what is an Arduino Uno microcontroller, pin configuration, Arduino Uno specifications or features, and applications.

NodeMCU:

The NodeMCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It

supports UART, SPI, and I2C interface.

2.2 SENSOR DETAILS

pH Sensor: Analog pH meter is specifically designed to measure the pH of the solution and reflect the acidity or alkalinity. It is commonly used in various applications such as aquaponics, aquaculture, and environmental water testing.

As an upgraded version of pH Meter, this product greatly improves the precision and user experience. The onboard voltage regulator chip supports the wide voltage supply of 3.3~5.5V, which is compatible with 5V and 3.3V main control board. The output signal filtered by hardware has low jitter. The software library adopts the two-point calibration method, and can automatically identify two standard buffer solutions (4.0 and 7.0), so simple and convenient

Liquid pH Sensor Module V2.0

2.3 PROTOCOL DETAILS:

The following describes the protocols and standards for the layers of the IP stack and an additional adaptation layer, which is used for communication between smart objects.

(1) Physical and MAC Layer*:*

The IEEE 802.15.4 protocol is designed for enabling communication between compact and inexpensive low power embedded devices that need a long battery life. It defines standards and protocols for the physical and link (MAC) layer of the IP stack. It supports low power communication along with low cost and short range communication. In the case of such resource constrained environments, we need a small frame size, low bandwidth, and low transmit power.

Transmission requires very little power (maximum one milliwatt), which is only one percent of that used in WiFi or cellular networks. The protocol also supports short 16-bit link addresses to decrease the size of the header, communication overheads, and memory requirements.

(2) Adaptation Layer.

IPv6 is considered the best protocol for communication in the IoT domain because of its scalability and stability. Such bulky IP protocols were initially not thought to be suitable for communication in scenarios with low power wireless links such as IEEE 802.15.4.

Specifically, the adaptation layer performs the following three optimizations in order to reduce communication overhead:

(i) Header compression

6loWPAN defines header compression of IPv6 packets for decreasing the overhead of IPv6. Some of the fields are deleted because they can be derived from link level information or can be shared across packets.

(ii) *Fragmentation*:

the minimum MTU size (maximum transmission unit) of IPv6 is 1280 bytes. On the other hand, the maximum size of a frame in IEEE 802.15.4 is 127 bytes. Therefore, we need to fragment the IPv6 packet. This is done by the adaptation layer.

(iii) Link layer forwarding

6LoWPAN also supports mesh under routing, which is done at the link layer using link level short addresses instead of in the network layer. This feature can be used to communicate within a 6LoWPAN network.

(3) Network Layer.

The network layer is responsible for routing the packets received from the transport layer. The IETF Routing over Low Power and Lossy Networks (ROLL) working group has developed a routing protocol (RPL) for Low Power and Lossy Networks (LLNs).

For such networks, RPL is an open routing protocol, based on distance vectors. It describes how a destination oriented directed acyclic graph (DODAG) is built with the nodes after they exchange distance vectors. A set of constraints and an objective function is used to build the graph with the best path [53]. Theobjective function and constraints may differ with respect to their requirements. For example, constraints can be to avoid battery powered nodes or to prefer encrypted links. The objective function can aim to minimize the latency or the expected number of packets that need to be sent.

(4) Transport Layer.

TCP is not a good option for communication in low power environments as it has a large overhead owing to the fact that it is a connection oriented protocol. Therefore, UDP is preferred because it is a connectionless protocol and has low overhead.

(5) Application Layer.

The application layer is responsible for data formatting and presentation. Here Message Queue Telemetry Transport: MQTT is a publish/subscribe protocol that runs over TCP. It was developed by IBM primarily as a client/server protocol. The clients are publishers/subscribers and the server acts as a broker to which clients connect through TCP. Clients can publish or subscribe to a topic. This communication takes place through the broker whose job is tocoordinate subscriptions and also authenticate the client for security. MQTT is alightweight protocol, which makes it suitable for IoT applications. But because of the fact that it runs over TCP, it cannot be used with all types of IoT applications. Moreover, it uses text for topic names, which increases its overhead.

Further, for power conservation, there is an offline procedure for clients who are in a sleep state. Messages can be buffered and later read by clients when they wake up. Clients connect to the broker through a gateway device, which resides within the sensor network and connects to the broker.

2.4 SOFTWARE REQUIREMENTS

C++:

C++ is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a compiled language. C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

C:

C is a general-purpose programming language that is extremely popular, simple, and flexible to use. It is a structured programming language that is machine-independent and extensively used to write various applications, Operating Systems like Windows, and many other complex programs like Oracle database, Git, Python interpreter, and more.

It is said that 'C' is a god's programming language. One can say, C is a base for the programming. If you know 'C,' you can easily grasp the knowledge of the other programming languages that uses the concept of 'C'

It is essential to have a background in computer memory mechanisms because it is an important aspect when dealing with the C programming language.

2.5 PROGRAMMING LANGUAGE DETAILS

If IoT devices are expected to do complex tasks, C++ is chosen over C. C++ comes with added abilities like data abstraction, classes and objects. C++ creates compact and faster runtime code. Line of code can be compiled into a couple of instructions leading to high runtime speeds and low energy consumption and is therefore suitable for writing IoT and embedded system code.

According to C++ developer, Bjarne Stroustrup, there is still no other language that makes it better than C++ when it comes to specialized hardware to be used for Internet of Things. C++ is designed to handle both hardware and complexity simultaneously. It has apparent advantage of running seamlessly with systems with a few hundred kilobytes of memory. And there are not many languages that can work within such a framework.

C is considered the most useful for IoT devices because it doesn't require a lot of processing power. C++ is an alternative if the IoT device requires more complex tasks, think thermostats and smart toasters rather than devices that detect moisture or heat. Java is another general purpose language that is useful for IoT devices that require a lot of interfacing and calculation, since it is more portable than C++, lightweight (for a high level language), and more commonly taught.

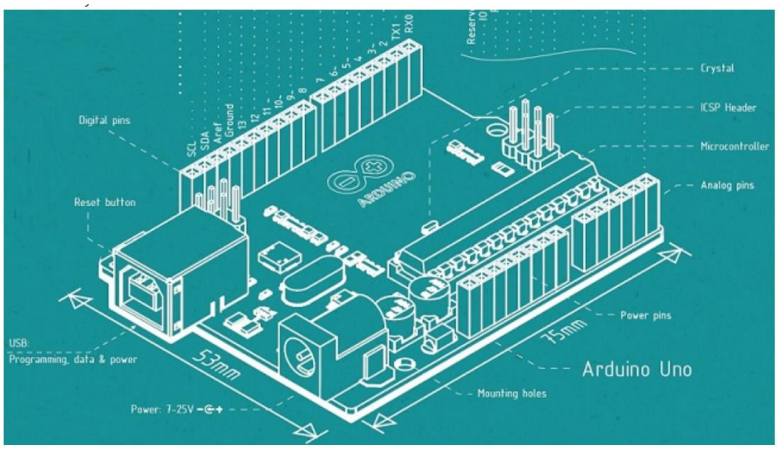
CHAPTER 3
ARCHITECTURE DESIGN

3.1 ARDUINO UNO ARCHITECTURE

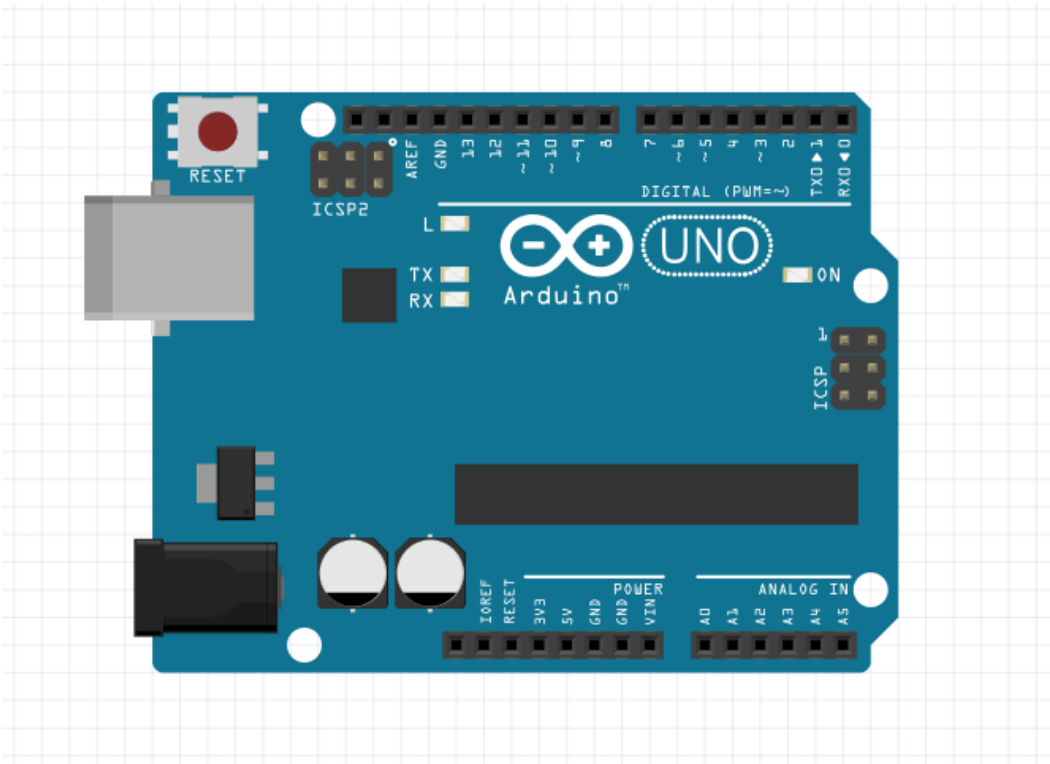
Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Arduino UNO:

It is an open-source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality.The software used for Arduino devices is called IDE (Integrated Development Environment) which is free to use and required some basic skills to learn it. It can be programmed using C and C++ language.Some people get confused between Microcontroller and Arduino. While former is just an on system 40 pin chip that comes with a built-in microprocessor and later is a board that comes with the microcontroller in the base of the board, bootloader and allows easy access to input-output pins and makes uploading or burning of the program very easy.



- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- UART: 1
- I2C: 1
- SPPI: 1
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g



Voltages:

Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V), which are unconfigurable. The remaining pins are all general purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant.

Power and Ground Pins:

- VIN (sometimes labelled "9V"). 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. Note that different

boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input.

- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.
- GND. Ground pins.

Other Pins

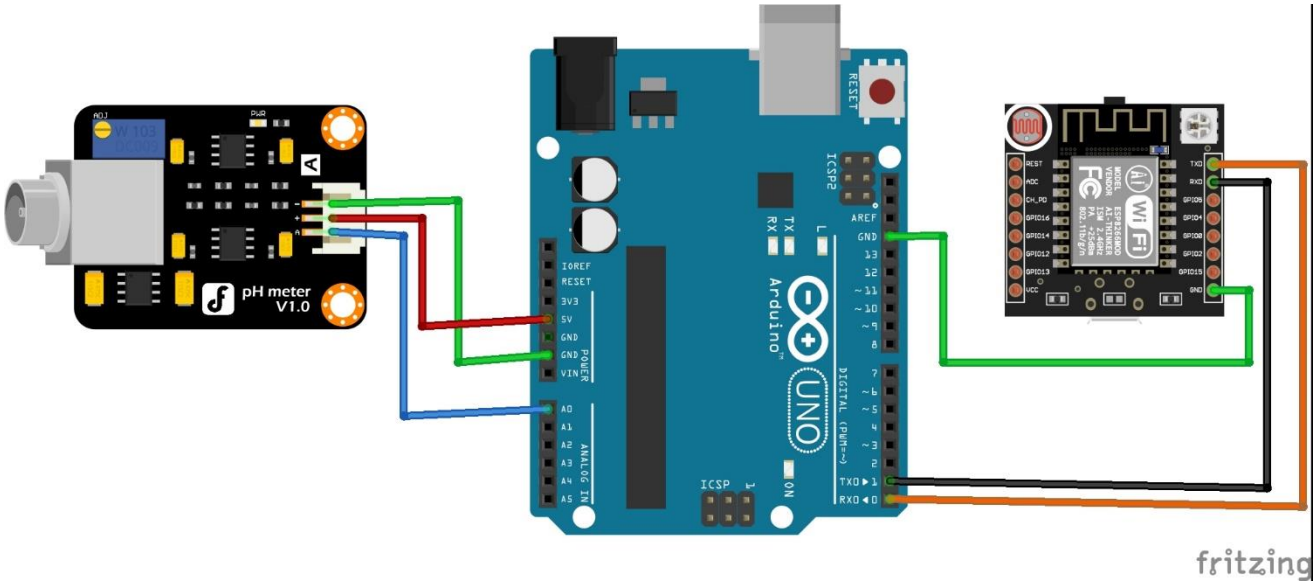
- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Digital Pins

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- External Interrupts: 2 and 3. 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: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- BT Reset: 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.

- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. On the Diecimila and LilyPad, 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.

3.2 CIRCUIT DIAGRAM



In this circuit diagram, we show how the wired connections are made with the components to the uno. Here it is shown how the main components are provided with power and ground based on the Arduino pin configuration. The RX and TX of NodeMCU is pinned with TXD and RXD of Uno. pH sensors ground is connected with arduino's ground, Power supply of 5V is connected

SYSTEM ARCHITECTURE:

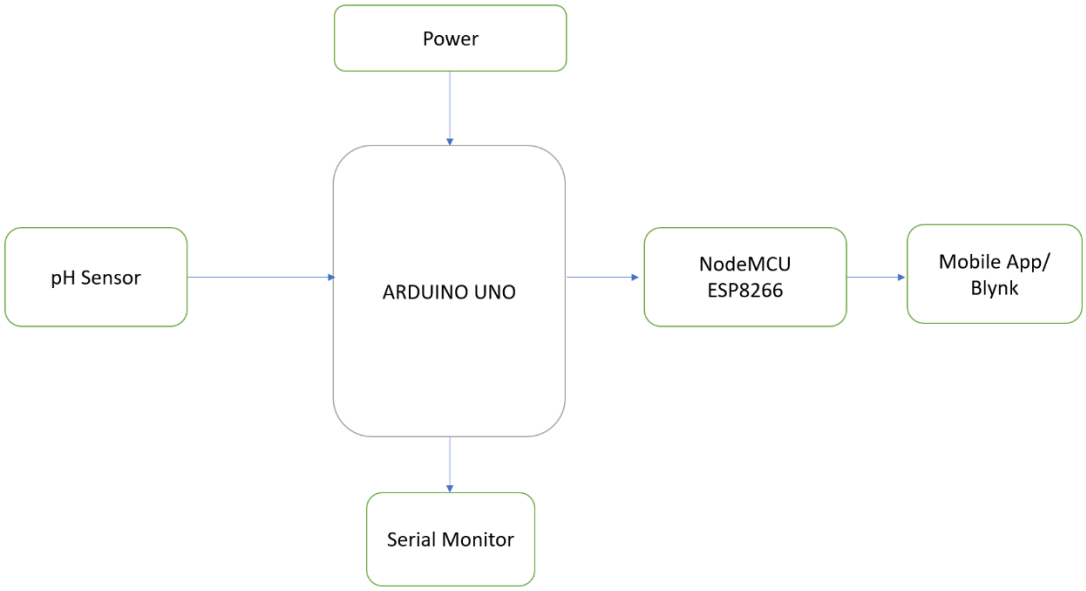


Figure 1: General Architecture

As in the Figure-1 General architectural diagram, above the sensor have been connected with The Meter in one end and the module is connected with Arduino Uno Board. Every sensor(s) value(s) is/are stored in the server and also the threshold is calculated instantly at all times. The value is passed to a phone via using NodeMCU WiFi module through internet. The system is mainly focused to know the average pH value.

1. pH sensor: that detects the ph value of any liquid mainly water as we focus on water for this project

ACTIVITY DIAGRAM:

Figure 2: Activity Diagram

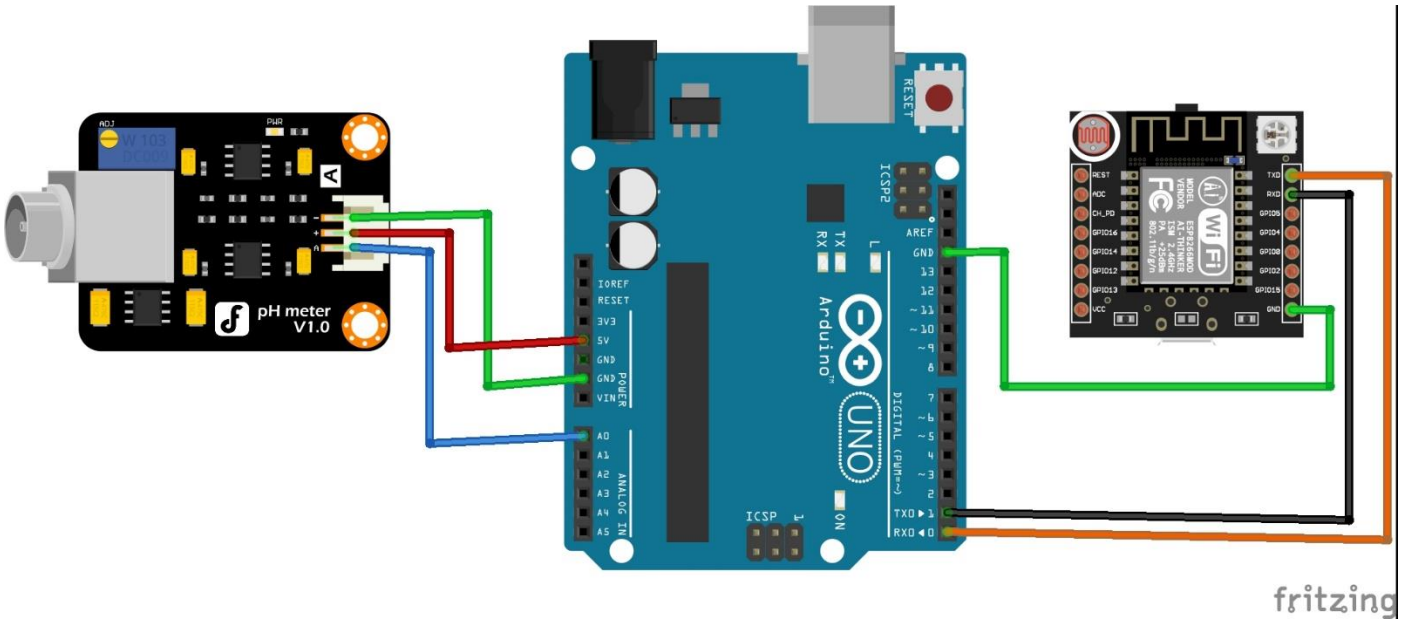
The Figure-2 explains the behaviour of the system based on the parameters provided by the sensors. This value is sent to a smartphone.

CHAPTER 4

WORKING PRINCIPLE

4.1 EXPANATION OF WORKING PRINCIPLE

In this project, The ph sensor machine is put into a volume of liquid or water and the value is calculated in analog and is sent to the Arduino board which is connected to a system. The value then is displayed in serial monitor. But the change done here is to get results in our very own hands that is a smart phone. The nodemcu is used to send the values to a smartphone through wifi to an App called Blynk.



The following steps take place in pH detection

- Step 1: Get some amount of liquid
- Step 2: Power on all the modules.
- Step 3: Put the sensor into the liquid
- Step 4: The code is ran in arduino
- Step 5: pH values are displayed in monitor
- Step 6: NodeMCU connects with Phone
- Step 7: Values are sent to phone via wifi
- Step 8: Stop the process.

CHAPTER 5
EXPERIMENTAL RESULT & ANALYSIS

5.1 COMPARISON OF EXISTING PRODUCT

Several research works have been conducted in recent times to develop intelligent and efficient systems to identify and monitor water
parameters. For real time monitoring of water quality and delivery, setup based on sensors is proposed. The recommended setup focused
on the low cost, lightweight implementation. This setup is appropriate for large amount categorizations enabling an approach to water
purchaser, water distributors and water supremacists.

In the modern world the problem of the efficient water supply is extremely important because the water resources are widely exploited
and water is used in different fields of human activities. Water is an essential need for human survival and therefore there must be
mechanism put in place to forcefully test the quality of water that made available for drinking in town and city. In this paper, we present
a setup for Water Quality Monitoring System based on Internet Of Things that continuously measures the water parameter i.e. pH,
turbidity, temperature and total dissolved solids. Four sensors are connected to NodeMCU to measure the water parameters.
Extracted data from the sensor is sent to the android application i.e. BLYNK and according to the pre-defined set of standard values, if
there is a mismatch in water parameters system will generate an alert message to the remote user. The data updated on the application can
be accessed or get back at your fingertips.

5.2 FEATURES

Current project features:

- Optimal response time to get values
- Warning if pH value is more.
- Faster results.
- Result in our own smartphone
- Internet enabled
- App with better UI

5.3 CODING

pHmeter

```
#include <stdlib.h>
#include <SoftwareSerial.h>
SoftwareSerial nodemcu(2,3);

#define SensorPin 0          //pH meter Analog output to Arduino Analog
Input 0
unsigned long int avgValue;  //Store the average value of the sensor
feedback
float b;
int buf[10],temp;

// for float value to string converstion
int f;
  float val; // also works with double.
  char buff2[10];
  String valueString = "";
  String Value = "";

void setup()
{
  pinMode(13,OUTPUT);
  Serial.begin(9600);
  nodemcu.begin(9600);
}
void loop()
{
  for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for
smooth the value
  {
    buf[i]=analogRead(SensorPin);
    delay(10);
```

```
  }
  for(int i=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(int i=2;i<8;i++)                      //take the average value of
6 center sample
    avgValue+=buf[i];
  float phValue=(float)avgValue*5.0/1024/6; //convert the analog into
millivolt
  phValue=3.5*phValue;                      //convert the millivolt into
pH value

     Value =  dtostrf(phValue, 4, 2, buff2);  //4 is mininum width, 6 is
precision
   valueString = valueString + Value +",";
   Serial.println(valueString);
   nodemcu.println(valueString);
  valueString = "";
 delay(1000);

}
```

Sketchnode:

```
/*
 * lcd.print(x,y,"messsage");
 * where x is a symbol position(0 to 15)
 * y is a line number (0 or 1)
 * lcd.clear();
 */

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <SimpleTimer.h>

WidgetLCD lcd(V2);
String data;
String I;
char auth[] = "HN4Q0oFe86vlsN_jlRDfGnfAS1fCYJwg";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Nibras";
char pass[] = "qwerty123";

SimpleTimer timer;

String myString; // complete message from arduino, which consistors of
snesors data
char rdata; // received charactors
```

```
// This function sends Arduino's up time every second to Virtual Pin
(1).
// In the app, Widget's reading frequency should be set to PUSH. This
means
// that you define how often to send data to Blynk App.
void myTimerEvent()
{
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V1, millis() / 1000);

}



void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

    timer.setInterval(1000L,sensorvalue1);

}

void loop()
{

   if (Serial.available() == 0 )
   {
  Blynk.run();
  timer.run(); // Initiates BlynkTimer
   }
```

```
  if (Serial.available() > 0 )
  {
    rdata = Serial.read();
    myString = myString+ rdata;
   Serial.print(rdata);
    if( rdata == '\n')
    {
I = getValue(myString, ',', 0);
  myString = "";
 // Serial.println(I);
lcd.print(0,0,"pH Value:");
    }
  }

}


void sensorvalue1()
{
data = data + I;
lcd.print(0,0,"pH Value:");
 lcd.print(0,1,data);
 data = "";

}



String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;
```

```
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) :
"";
}
```

CHAPTER 6
CONCLUSION & FUTURE WORK

6.1 CONCLUSION

In this paper, a prototype water monitoring system using IoT is presented. .For this some sensors are used. The collected data from the all the sensors are used for analysis purpose for better solution of water problems. The data is sends to the cloud server via Wi-Fi module ESP8266. So this application will be the best challenger in real time monitoring & control system and use to solve all the water related problems.

The main purpose of this work is to observe the quality of water samples by designing an intelligent water quality monitoring setup implemented in IoT platform that can detect water's physical and chemical parameters. The interfacing is done between the system and the sensor network on a single chip solution wirelessly. For the monitoring process, the system is achieved with reliability and feasibility by verifying the four parameters of water. The time interval of monitoring might be changed depending upon the necessity of water resources .The time is reduced, and the cost is low in this environmental management

6.2 FUTURE WORK

In modern cities there are water diseases spreading each and every day and also in agriculture and other purposes

The water used is of wrong Ph. It is to be acknowledged that the pH is also an important factor to be taken care of

Daily. So to get the pH value of the liquids each and everyone use daily it is advisable that this will help us in getting the results much faster than expected.

The system can monitor water quality automatically, and it is low in cost and does not require people on duty. So the water quality testing is likely to be more economical, convenient and fast. The system has good flexibility. Only by replacing the corresponding sensors and changing the relevant software programs, this system can be used to monitor other water quality parameters. The operation is simple. The system can be expanded to monitor hydrologic, air pollution, industrial and agricultural production and so on. It has widespread application and extension value. By keeping the embedded devices in the environment for monitoring enables self protection (i.e., smart environment) to the environment. To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, we can bring the environment into real life i.e. it can interact with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi.

6.3 REFERENCES

- Thinagaran Perumal1, 1Md Nasir Sulaiman, 2Leong.C.Y, “Internet of Things (IoT) Enabled Water Monitoring System “,2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)
- Perumal, T.; Sulaiman, M.N.; Mustapha, N.; Shahi, A.; Thinaharan, R., "Proactive architecture for Internet of Things (IoTs) management in smart homes," Consumer Electronics
- Asaad Ahmed Mohammedahmed Eltaieb, Zhang Jian Min, “Automatic Water Level Control System”, International Journal of Science and Research (IJSR)2013