CONTENTS

List of Figures

List of Screens

List of Abbreviations

1. SDLC - Software Development Life Cycle
2. UML - Unified Modeling Language
3. ICCCIS - International Conference on Computing, Communication and Intelligent Systems
4. ICECA - International Conference of Electronics, Communications and Aerospace Technology
5. SRS – Software Requirement Specification
6. DFD Data Flow Diagram
7. RFID – Radio Frequency Identification

ABSTRACT

Water is one of the essential parts of life. Water pollution is one of the big problems to the world. In order to ensure the safe supply of the drinking and useful water for different purposes like agricultural, the water should be monitored. This paper presents a design of a low cost system for real time monitoring of the water quality and quantity of water in IOT (internet of things). The system having of several sensors is used to measuring physical of the water. The parameters flow sensor of the water can be measured. The measured values from the sensors can be processed by the controller. The Arduino model can be used as a controller. Finally, the sensor data can be shown on internet using WIFI system. A cloud server was configured as data saving and analysis. This data can be used in future research and development.

Chapter 1

INTRODUCTION

1.1 Introduction

Environment around us consists of five key elements. These are soil, water, climate, natural vegetation and land forms. Among these water the most essential element for human to live. It is also important for the survival of other living habitants. Whether it is used for drinking, domestic use, and food production or recreational purposes, safe and readily available water is must for public health. So it is highly imperative for us to maintain water quality balance. Otherwise it would severely damage the health of the humans and at the same time affect the ecological balance among other species. In the 21st century providing pure drinking water is becoming a major challenge worldwide. International governing bodies such as United Nations (UN) and World Health Organization (WHO) also recognized human right to sufficient, continuous, safe, and acceptable, physically accessible, and affordable water for personal and domestic use. According to research of WHO 844 million people lack even a basic drinking –water service, including 159 million people who are dependent on surface water. Impure drinking can cause life threatening disease such diarrhea, cholera, dysentery, typhoid, and polio. The research alarmingly estimates that every year diarrhea alone is causing around death of five lakh people. Figure 1 illustrates how water crisis becoming an epidemic in twenty first century. Now a day's Internet of things is a revolutionary technological phenomenon. It is shaping today's world and is used in different fields for collecting, monitoring and analysis of data from remote locations. Internet of things integrated network if everywhere starting from smart cities, smart power grids, and smart supply chain to smart wearable. Though internet of things is still under applied in the field of environment it has huge potential. It can be applied to detect forest fire and early earthquake, reduce air population, monitor snow level, prevent landslide and avalanche etc. Moreover it can be implemented in the field of water quality monitoring and controlling system. We can design a water quality monitoring system in smart city where there will be a network of devices connected to remote stations and the parameters from the water sources will be stored in a microcontroller via WSN. City dwellers can easily get notified about of the quality of the water via SMS or they can view it on webpage and also local authority can take necessary actions

1.2 Objective of Project

The population of our world is growing rapidly and percentages of drinkable water sources are dropping down very fast. As a result of that meeting the need of ever growing people, major water sources like rivers, ponds, seas, canals etc are being filled up and large industrial in fractures are made. According to World Health Organization (WHO) by 2025, half of the world’s population will be living in water-stressed areas. For the developing country like Bangladesh, where people cannot afford high cost water purifier this system can provide affordable solution to water crisis.

1.3 Problem Definition

Smart Water Monitoring System for Real-Time Water Quality and Usage Monitoring. The Smart Water Quality meter checks the purity of portable water that the consumer receives, by measuring five qualitative parameters of water viz. pH, temperature, turbidity, dissolved oxygen and conductivity.

Chapter 2

LITERATURE SURVEY

Nikhil Kedia entitled “Water Quality Monitoring for Rural Areas-A Sensor Cloud Based Economical Project.” Published in 2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India. This paper highlights theentire water quality monitoring methods, sensors, embedded design, and information dissipation procedure, role of government, network operator and villagers in ensuring proper information dissipation. It also explores the Sensor Cloud domain. While automatically improving the water quality is not feasible at this point, efficient use of technology and economic practices can help improve water quality and awareness among people.[1]

Jayti Bhatt,Jignesh Patoliya entitled “Real Time Water Quality Monitoring System”.This paper describes to ensure the safe supply of drinking water the quality should be monitored in real time for that purpose new approach IOT (Internet of Things) based water quality monitoring has been proposed. In this paper, we present the design of IOT based water quality monitoring system that monitor the quality of water in real time. This system consists some sensors which measure the water quality parameter such as pH, turbidity, conductivity, dissolved oxygen, temperature. The measured values from the sensors are processed by microcontroller and this processed values are transmitted remotely to the core controller that is raspberry pi using Zigbee protocol. Finally, sensors data can view on internet browser application using cloud computing.

[2] Michal Lom, Ondrej Pribyl, Miroslav Svitek entitled “Industry 4.0 as a Part of Smart Cities”. This paper describes the conjunction of the Smart City Initiative and the concept of Industry 4.0. The term smart city has been a phenomenon of the last years, which is very inflected especially since 2008 when the world was hit by the financial crisis. The main reasons for the emergence of the Smart City Initiative are to create a sustainable model for cities and preserve quality of life of their citizens. The topic of the smart city cannot be seen only as a technical discipline, but different economic, humanitarian or legal aspects must be involved as well. In the concept of Industry 4.0, the Internet of Things (IoT) shall be used for the development of so–called smart products. Subcomponents of the product are equipped with their own intelligence. Added

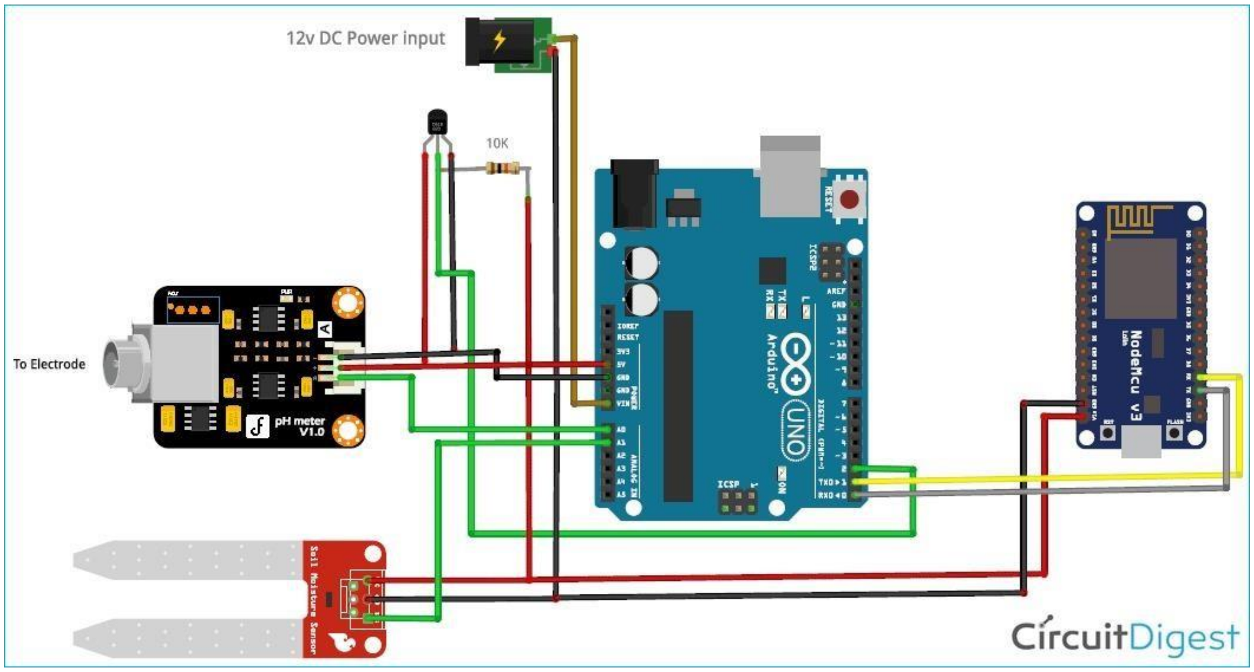
intelligence is used both during the manufacturing of a product as well as during subsequent handling, up to continuous monitoring of the product lifecycle (smart processes). Other important aspects of the Industry 4.0 are Internet of Services (IoS), which includes especially intelligent transport and logistics (smart mobility, smart logistics), as well as Internet of Energy (IoE), which determines how the natural resources are used in proper way (electricity, water, oil, etc.). IoT, IoS, IoP and IoE can be considered as an element that can create a connection of the Smart City Initiative and Industry 4.0 – Industry 4.0 can be seen as a part of smart cities.

2.1 Related Work

The construction of a pH sensor is shown above. The pH Sensor looks like a rod usually made of a glass material having a tip called "Glass membrane". This membrane is filled with a buffer solution of known pH (typically pH = 7). This electrode design ensures an environment with the constant binding of H+ ions on the inside of the glass membrane. When the probe is dipped into the solution to be tested, hydrogen ions in the test solution start exchanging with other positively charged ions on the glass membrane, which creates an electrochemical potential across the membrane which is fed to the electronic amplifier module which measures the potential between both electrodes and converts it to pH units. The difference between these potentials, the voltage displayed on the pH sensor, determines the pH value based on the Nernst equation.

Circuit Diagram

Circuit diagram for this Arduino based Smart Water Quality Monitoring System is given below:



2.2 Requirement Analysis

Arduino UNO - 550

NodeMCU - 420

Gravity Analog pH sensor - 2500

DS18B20 temperature sensor - 250

Soil moisture sensor - 150

Power supply - 350

Jumpers - 200

Breadboard – 150

Temperature range - 55 to 125°C

bit selectable resolution - 9-12 bit

1-Wire interface

Unique 64-bit address enables multiplexing

Accuracy - ±0.5°C

Operating Voltage - 3-5 VDC

Conversion time - 750ms at 12-bit

Chapter 3

SYSTEM ANALYSIS & DESIGN

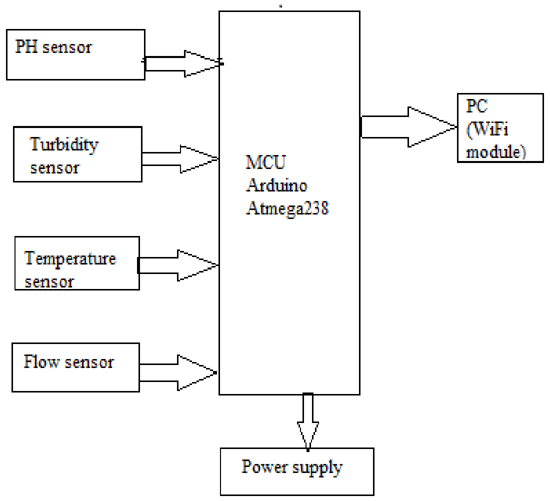


Fig1: Block diagram of our project

In this proposed block diagram consist of several sensors (temperature, pH, turbidity, flow) is connected to core controller. The core controller are accessing the sensor values and processing them to transfer the data through internet. Ardunio is used as a core controller. The sensor data can be viewed on the internet wi-fi system.

PH Sensor: The pH of a solution is the measure of the acidity or alkalinity of that solution. The pH scale is a logarithmic scale whose range is from 0-14 with a neutral point being 7. Values above 7 indicate a basic or alkaline solution and values below 7 would indicate an acidic solution. It operates on 5V power supply and it is easy to interface with arduino.The normal range of pH is 6 to 8.5.

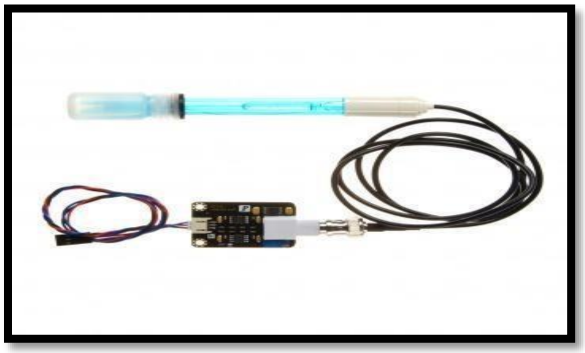


Fig2: pH sensor

Temperature sensor: Water Temperature indicates how water is hot or cold. The range of DS18B20 temperature sensor is -55 to +125 °C. This temperature sensor is digital type which gives accurate reading

Fig 3: Temperature sensor

Arduino Uno: Arduino is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Fig 4: Arduino uno

ultrasonic sensor : An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound

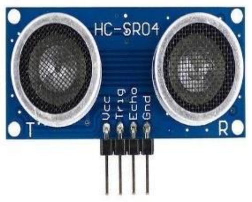


Fig 5: **ultrasonic sensor**

infrared (IR) sensor :An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation.

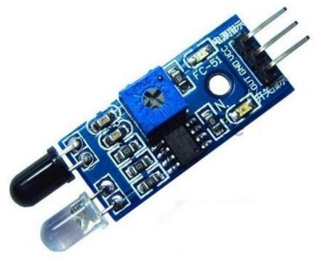


Fig 6: **infrared (IR) sensor**

Gravity Analog pH Sensor : DFRobot Gravity Analog pH Sensor is specifically designed to measure the pH of the solution and reflect the acidity or alkalinity. ...
The gravity analog pH sensor is used in various applications such as aquaponics, aquaculture, and environmental water testing.

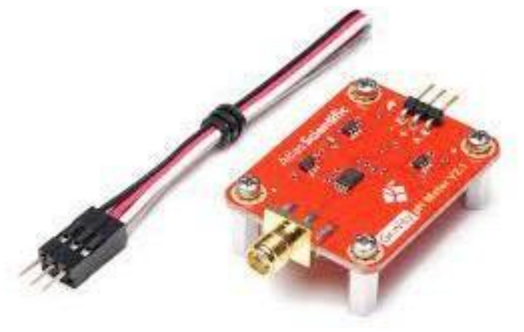


Fig 7 : Gravity Analog pH Sensor

3.1 Type of SDLC Model Used:

Waterfall Model - Design
Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.
The following illustration is a representation of the different phases of the Waterfall Model.

The sequential phases in Waterfall model are −

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the

customer environment.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

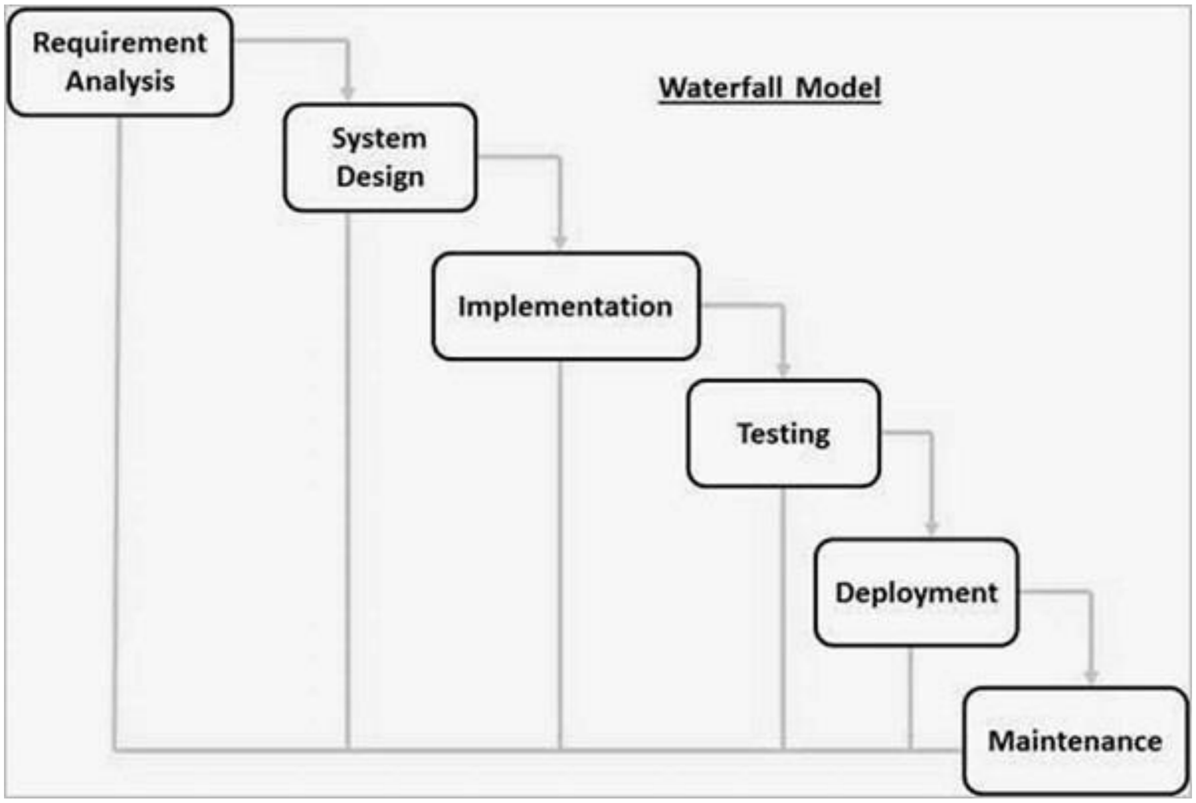
The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.

- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

3.2 UML Diagrams



Chapter 4

SYSTEM IMPLEMENTATION

4.1 Implementation Details

Programming for Soil Monitoring using IoT

There are two parts of programming in this Smart Water Monitoring System using IoT. In the first part, Arduino is programmed and in the second part, NodeMCU will be programmed. Complete code for both Arduino and NodeMCU along with the video is given at the end of this tutorial. The stepwise description of the code is given below.

Arduino Programming:

First of all, include all the header files, which will be required throughout the code. Here we are using onwire.h and DallasTemperature.h library for a DS18B20 temperature sensor. This can be downloaded from the links given and included in the Arduino library. Similarly, we are using ArduinoJson.h library for sending JSON data from the transmitter to the receiver side.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ArduinoJson.h>
```

Next, define the connection pin of Arduino, where the output pin of the DS18B20 sensor will be connected, which is digital pin 2 in my case. Then, objects for onewire class and DallasTemperature class are defined which will be required in the coding for temperature measurement.

```
OneWire oneWire(2);
DallasTemperature temp_sensor(&oneWire);
```

Next, the calibration value is defined, which can be modified as required to get an accurate pH value of solutions.

```
float calibration_value = 21.34;
```

Then a JSON Object is defined which will be required for sending parameters from the Transmitter part to the Receiver part.

```
StaticJsonBuffer<1000> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();
```

Inside loop(), read 10 sample Analog values and store them in an array. This is required to smooth the output value.

```
for(int i=0;i<10;i++)
 {
 buffer_arr[i]=analogRead(A0);
 delay(30);
 }
```

Then, we have to sort the Analog values received in ascending order. This is required because we need to calculate the running average of samples in the later stage.

```
 for(int i=0;i<9;i++)
 {
 for(int j=i+1;j<10;j++)
 {
 if(buffer_arr[i]>buffer_arr[j])
 {
 temp=buffer_arr[i];
 buffer_arr[i]=buffer_arr[j];
 buffer_arr[j]=temp;
 }
 }
 }
```

Finally, calculate the average of a 6 centre sample Analog values. Then this average value is converted into actual pH value and stored in a variable.

```
 for(int i=2;i<8;i++)
 avgval+=buffer_arr[i];
 float volt=(float)avgval*5.0/1024/6;
 float ph_act = -5.70 * volt + calibration_value;
```

To send a command to get the temperature values from the sensor, requestTemperatures() function is used.

temp_sensor.requestTemperatures();

Now, analog values from the soil moisture sensor are read and this is mapped to percentage using map() function as shown below:

int moisture_analog=analogRead(A1);

int moist_act=map(moisture_analog,0,1023,100,0);

Finally, the parameters which are to be sent to NodeMCU, are inserted into JSON objects and they are sent via serial communication using root.printTo(Serial) command.

root["a1"] = ph_act;

root["a2"] = temp_sensor.getTempCByIndex(0);

root["a3"] = moist_act;

root.printTo(Serial);

Serial.println("");

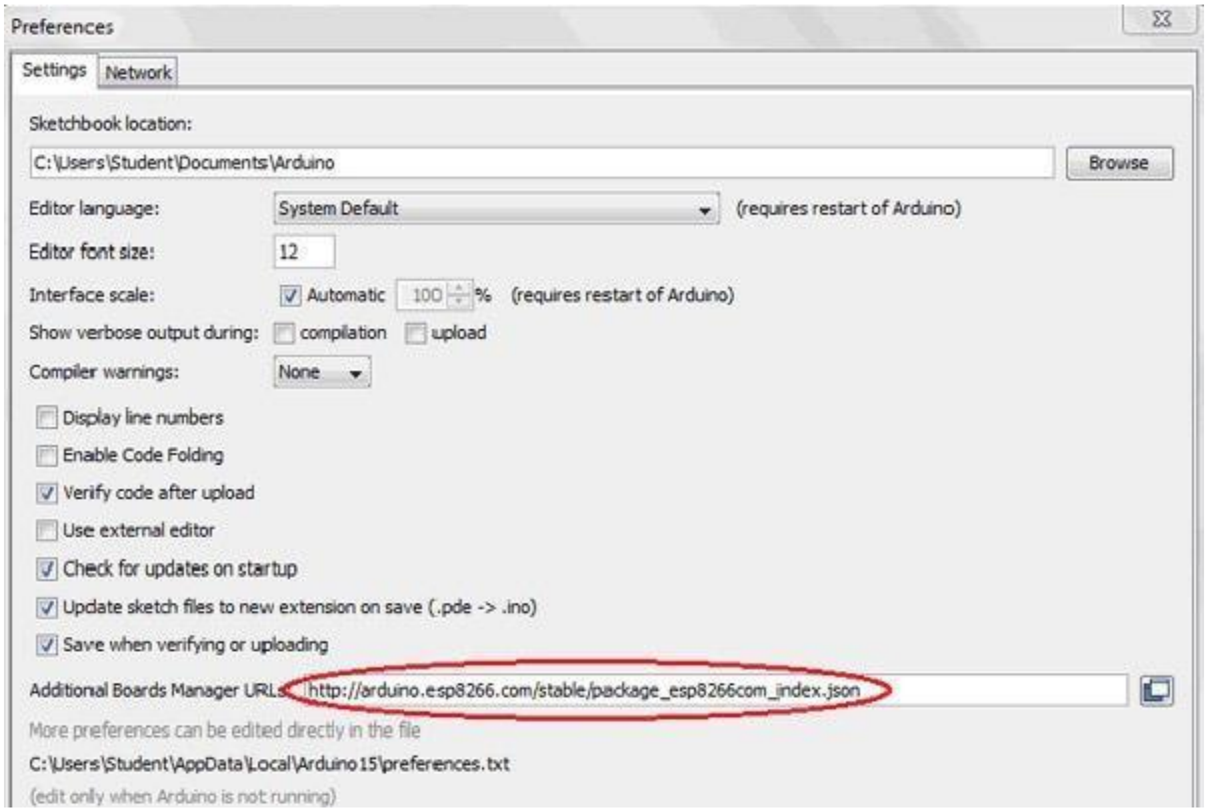
Programming NodeMCU

After the successful completion of the hardware setup and Arduino programming, now it’s time to program the NodeMCU.

Note: Remove the Transmitter and Receiver connections between Arduino and NodeMCU before uploading the code.

To upload code in NodeMCU, follow the steps below:

1. Open Arduino IDE, then go to File–>Preferences–>Settings.


Preferences

Settings | Network

Sketchbook location: C:\Users\Student\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☑ Verify code after upload

☐ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

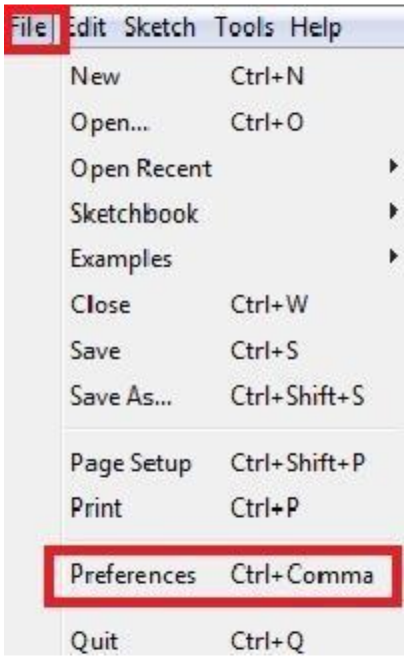
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

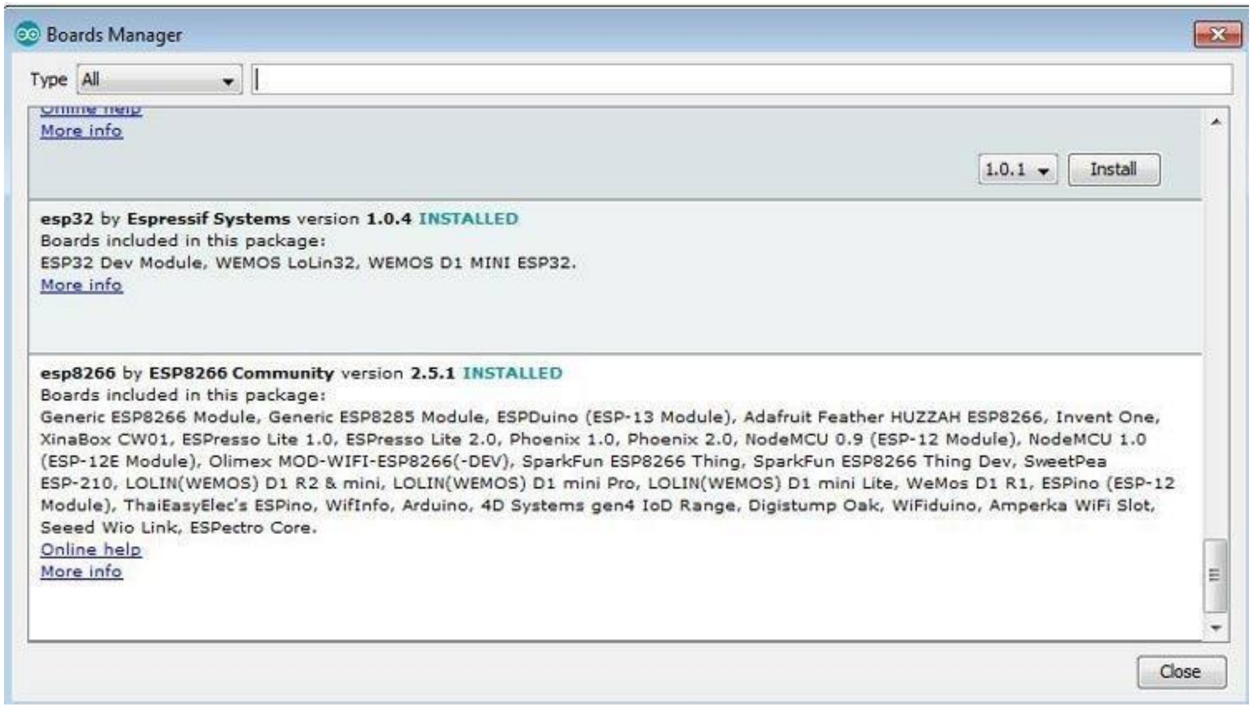
More preferences can be edited directly in the file

C:\Users\Student\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)



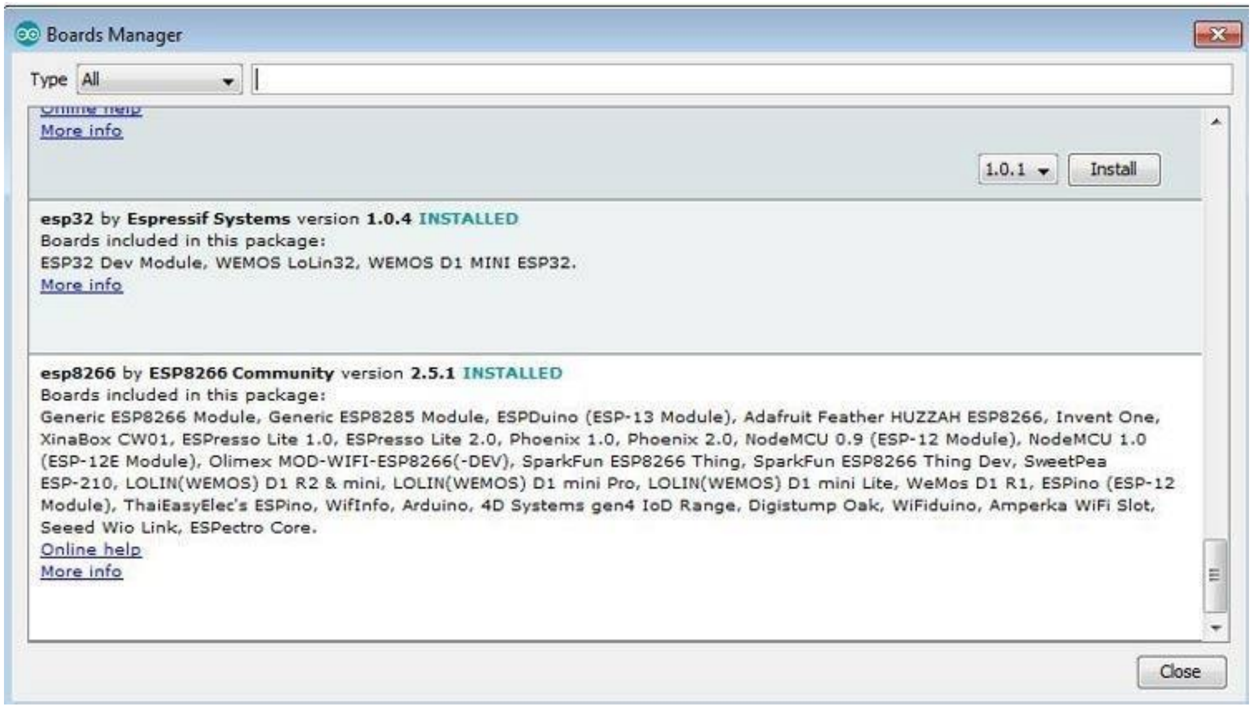
2. Type https:// arduino.esp8266.com/stable/package_esp8266com_index.json in the 'Additional Board Manager URL' field and click 'Ok'.



esp32 by Espressif Systems version 1.0.4 INSTALLED
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More info

esp8266 by ESP8266 Community version 2.5.1 INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

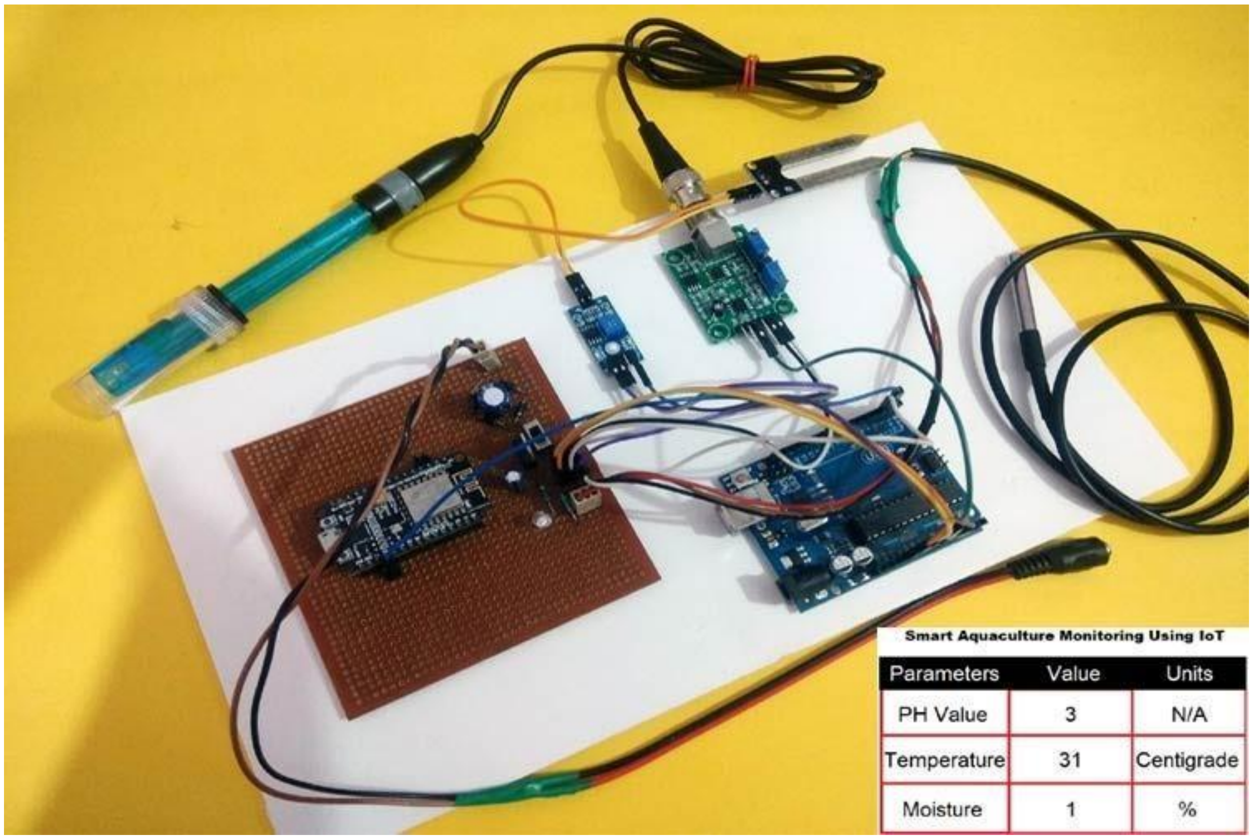
3. Now go to Tools > Board > Boards Manager. In the Boards Manager window, Type ESP8266 in the search box, select the latest version of the board, and click on install.



esp32 by Espressif Systems version 1.0.4 INSTALLED
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More info

esp8266 by ESP8266 Community version 2.5.1 INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

4.2 Snapshots



Smart Aquaculture Monitoring Using IoT

Parameters	Value	Units
PH Value	3	N/A
Temperature	31	Centigrade
Moisture	1	%

4.3 Coding

```
#include<ESP8266WiFi.h>
#include<WiFiClient.h>
#include<ESP8266WebServer.h>
#include <ArduinoJson.h>
```

Then create the ESP8266WebServer class object with the name server and default port number 80.

```
ESP8266WebServer server (80);
```

Now, declare network credentials i.e SSID and password. It is required to connect our NodeMCU to the internet.

```
const char* ssid = "admin";
const char* password = "12345678";
```

Then, to connect NodeMCU to the internet, call WiFi.begin and pass network SSID and password as its arguments. Check for the successful network connection using WiFi.status() and after a successful connection, print a message in Serial Monitor with IP address.

```
Serial.begin(115200);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
  {
  delay(500);
  Serial.print(".");
  }
  Serial.print(WiFi.localIP());
```

In the next step, an HTML page for this Water Monitoring System using IOT is created as shown below, which has an HTML table to show pH value, Temperature, and soil moisture. The HTML page is stored in a string variable so that it can be sent back on client request using the server.send() function.

```
page = "<html><head><title>IoT Design Pro</title></head><style type=\"text/css\">";
    page += "table{border-collapse: collapse;}th {background-color:  green ;color: white;}table,td
{border: 4px solid black;font-size: x-large;";
    page += "text-align:center;border-style: groove;border-color:
rgb(255,0,0);}</style><body><center>";
    page += "<h1>Smart Aquaculture Monitoring using IoT</h1><br><br><table style=\"width:
1200px;height: 450px;\"><tr>";
    page += "<th>Parameters</th><th>Value</th><th>Units</th></tr><tr><td>PH
Value</td><td>"+String(data1)+"</td><td>N/A</td></tr>";
    page +=
"<tr><td>Temperature</td><td>"+String(data2)+"</td><td>Centigrade</td></tr><tr><td>Mois
ture</td><td>"+String(data3)+"</td><td>%</td>";
    page += "<meta http-equiv=\"refresh\" content=\"3\">";
    server.send(200, "text/html", page);
```

Now, check for valid JSON data reception, from the Transmitter side. If no valid data is found, a message will be displayed on the serial monitor.

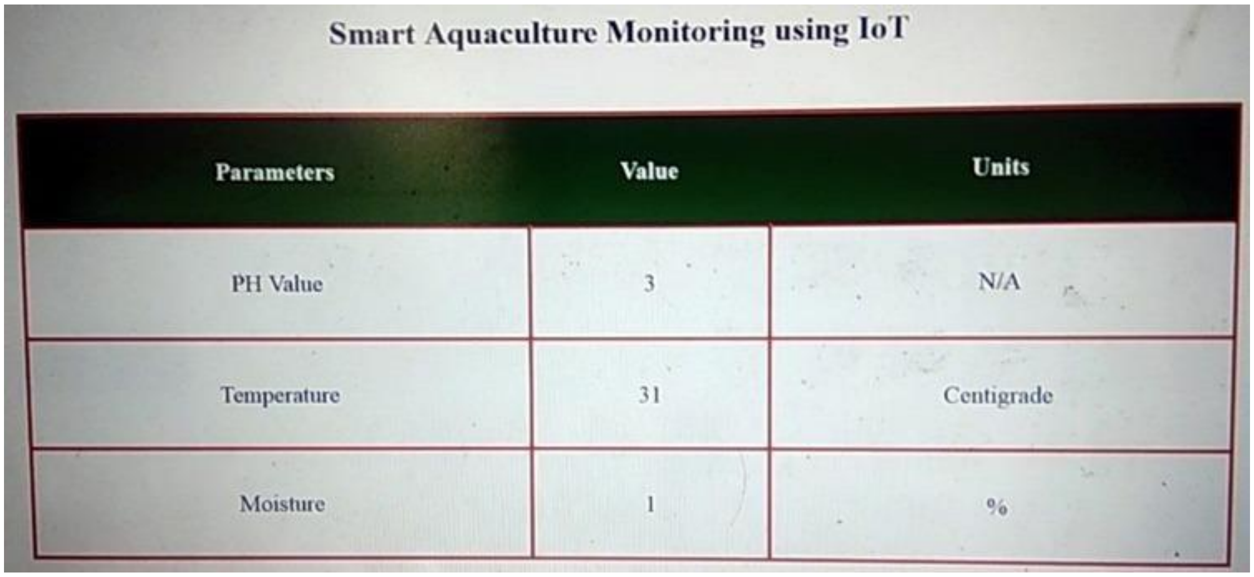
```
if (root == JsonObject::invalid())
  {
    return;
    Serial.println("invalid");
  }
```

Otherwise, data received are assigned to individual variables and appended in the HTML webpage for real-time display.

```
root["a1"] = ph_act;
root["a2"] = temp_sensor.getTempCByIndex(0);
root["a3"] = moist_act;
```

At the end of the loop, to handle the client request, we have to call server.handleClient(). It will handle new requests and check them.

```
server.handleClient();
```



Smart Aquaculture Monitoring using IoT

Parameters	Value	Units
PH Value	3	N/A
Temperature	31	Centigrade
Moisture	1	%

```
# Arduino Code
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ArduinoJson.h>
OneWire oneWire(2);
DallasTemperature temp_sensor(&oneWire);
float calibration_value = 21.34;
int phval = 0;
unsigned long int avgval;
int buffer_arr[10], temp;
void setup()
{
  Serial.begin(9600);
  temp_sensor.begin();
}
StaticJsonBuffer<1000> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();
void loop() {
  for (int i = 0; i < 10; i++)
  {
    buffer_arr[i] = analogRead(A0);
    delay(30);
  }
  for (int i = 0; i < 9; i++)
  {
    for (int j = i + 1; j < 10; j++)
    {
      if (buffer_arr[i] > buffer_arr[j])
      {
        temp = buffer_arr[i];
        buffer_arr[i] = buffer_arr[j];
        buffer_arr[j] = temp;
      }
    }
  }
  avgval = 0;
  for (int i = 2; i < 8; i++)
    avgval += buffer_arr[i];
  float volt = (float)avgval * 5.0 / 1024 / 6;
  float ph_act = -5.70 * volt + calibration_value;
  temp_sensor.requestTemperatures();
  int moisture_analog=analogRead(A1);
  int moist_act=map(moisture_analog,0,1023,100,0);
  root["a1"] = ph_act;
  root["a2"] = temp_sensor.getTempCByIndex(0);
  root["a3"] = moist_act;
  root.printTo(Serial);
  Serial.println("");
}
```

Chapter 5

CONCLUSION

5.1 CONCLUSION

Monitoring of Turbidity, PH & Temperature of Water makes use of water detection sensor with unique advantage and existing GSM network. 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.

5.2 FUTURE SCOPE:

- In future we use IOT concept in this project
- Detecting the more parameters for most secure purpose
- Increase the parameters by addition of multiple sensors
- By interfacing relay we controls the supply of water

Chapter 6

REFERENCES

[1] Nikhil Kedia, Water Quality Monitoring for Rural Areas- A Sensor Cloud Based Economical Project, in 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2015. 978-1-4673-6809-4/15/$31.00 ©2015 IEEE

[2] Jayti Bhatt, Jignesh Patoliya, Iot Based Water Quality Monitoring System, IRFIC, 21feb,2016.

[3] Michal lom, ondrej priby & miroslav svitek, Internet 4.0 as a part of smart cities, 978-1-5090-1116-2/16/$31.00 ©2016 IEEE

[4] Zhanwei Sun, Chi Harold Liu, Chatschik Bisdikia_, Joel W. Branch and Bo Yang, 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks

[5] (SECON), 978-1-4673-1905-8/12/$31.00 ©2012 IEEE

[6] Sokratis Kartakis, Weiren Yu, Reza Akhavan, and Julie A. McCann, 2016 IEEE First International Conference on Internet-of-Things Design and Implementation, 978-1-4673-9948-7/16 © 2016IEEE

[7] Mithaila Barabde, shruti Danve, Real Time Water Quality Monitoring System, IJIRCCE, vol 3, June 2015.

[8] Akanksha Purohit, Ulhaskumar Gokhale, Real Time Water Quality Measurement System based on GSM , IOSR (IOSR-JECE) Volume 9, Issue 3, Ver. V (May - Jun. 2014)

[9] Eoin O’Connell, Michael Healy, Sinead O’Keeffe, Thomas Newe, and Elfed Lewis, IEEE sensors journal, vol. 13, no. 7, July 2013, 1530-437x/$31.00 © 2013 IEEE

[10] Nidal Nasser, Asmaa Ali, Lutful Karim, Samir Belhaouari, 978-1-4799- 0792-2/13/$31.00 ©2013 IEEE