

WATER QUALITY MONITORING AND WASTEMANAGEMENT USING IOT

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Engineering Degree in Electronics and
Communication Engineering

by

**J.MANI PAVAN(37130164)
G.TRINADH(37130152)**



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600119

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **J.MANI PAVAN (37130164)** and **G. TRINADH (37130152)** who carried out the project entitled “WATER QUALITY MONITORING AND WASTE MANAGEMENT USING IOT” under our supervision from October 2020 to April 2021.

Internal Guide

DR.N.R.KRISHNAMOORTHY M.E. Ph.D.

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **J.MANI PAVAN (37130164)** and **G.TRINADH (37130152)** hereby declare that the Project Report entitled **“WATER QUALITY MONITORING AND WASTEMANAGEMENT USING IOT”** done by us under the guidance of **Dr. N.R KRISHNAMOORTHY, M.E., Ph.D** submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1)

2)

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D. Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. N.R.KRISHNAMOORTHY M.E., Ph.D., Department of Electronics and Instrumentation Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We express our gratitude to our parents for their constant encouragement and support for the completion of the project.

ABSTRACT

ABSTRACT:

In order to ensure the safe supply of the drinking water the quality needs to be monitor in real time. In our project, we design and development of a low cost system for real time monitoring of the water quality in IOT(internet of things). The system consists of several sensors which will be used to measuring physical and chemical parameters of the water. The parameters such as PH, and prototype setup for temperature, turbidity, conductivity, dissolved oxygen of the water can be measured. The measured values from the sensors can be processed by the Arduino to core controller. The NodeMCU model can be used as a core controller And NodeMCU is act as a server to send data to cloud Finally, the sensor data can be viewed on internet using cloud computing.

TABLE OF CONTENTS

CHAPTER.NO	TITLE	PAGE.NO
	ABSTRACT	V
	LIST OF FIGURES	VIII
1	INTRODUCTION	1
	1.1 IMPORTANCE OF WATER QUALITY	1
	1.2 VARIOUS TYPES OF WATER CONTAMINANTS	2
	1.3 INTRODUCTION TO IOT	2
	1.4 HOW IOT WORKS	4
	1.5 BENEFITS OF IOT	5
	1.6 HISTORY OF IOT	6
	1.7 APPLICATIONS	7
2	LITERATURE SURVEY	9
3	EXISTING SYSTEM	11
	3.1 EXISTING SYSTEM METHODOLOGIES	11
	3.2 PROBLEM STATEMENT	11
	3.3 OBJECTIVE	12
4	PROPOSED SYSTEM	13
	4.1 PROPOSED METHODOLOGIES	13
	4.2 ADVANTAGES	13
	4.3 BLOCK DIAGRAM	14

CHAPTER.NO	TITLE	PAGE.NO
	4.4 COMPONENTS AND FUNCTIONING	14
	4.5 HARDWARE REQUIREMENTS	15
	4.6 SOFTWARE REQUIREMENTS	38
5	RESULT AND DISCUSSION	45
	5.1 DATA PROCESSED IN SERVER	45
	5.2 IDEAL VIEW	45
	5.3 OUTPUT ANALYSIS	46
6	CONCLUSION AND FUTURE WORKS	49
	6.1 CONCLUSION	49
	6.2 FUTURE SCOPE	49
	REFERENCES	50

LIST OF FIGURES

FIGURE NO	TITLE	PAGE.NO
1.1	Internet of things	3
1.2	Internet of things cloud	4
1.3	Example of IOT	5
4.1	Block diagram of water quality monitoring system	14
4.2	Nodemcu	15
4.3	NodeMCU Development Board/kit v0.9 (Version1)	16
4.4	NodeMCU Development Board/kit v1.0 (Version2)	17
4.5	Turbidity sensor	18
4.6	Turbidity sensor interface	20
4.7	Measurement Principle of Turbidity sensor	21
4.8	pH sensor interface	24
4.9	Level sensor module	27
4.10	DHT11 sensor	30
4.11	Ultrasonic sensor	32
4.12	Relay module	33
4.13	Piezoelectric disk beeper	35
4.14	Buzzer	36
4.15	DC MOTOR	37
4.16	Ar/duino IDE	38
4.17	Arduino IDE Sketch	40
5.1	Demonstration	45
5.2	Ideal view with water	46
5.3	Data in cloud	48

CHAPTER 1

INTRODUCTION

1.1 IMPORTANCE OF WATER QUALITY :

- Good water quality is one of the most primary factors considering the health of every living thing in the biosphere and to the aquatic ecosystem. At present, due to the unexpected growth of industries water and water bodies get spoiled and affected because of industrial wastage in various distribution networks across the globe.
- The nuclear reactor wastage affects the sea and aquatic based living things. When human beings consume those seafood, indirectly health is getting affected. Use of hazardous chemicals in manufacturing industries and agriculture cause severe water pollution as they are directly let into nearby rivers, lakes, and ponds without getting treated to remove harmful toxins and compounds.
- Increasing water pollution badly affects marine life and their habitats. Poor people who do not have access to clean drinking water are forced to drink untreated water which makes them prone to water- related disease which affects their health.
- The accelerated evaporation of ponds is due to anthropogenic actions such as sewage disposal, and fertilizer run-off from agricultural lands. At a certain level polluted water can be dangerous to crops and reduce the fertility of the soil, thus harming the overall agricultural sector and the country as well.
- Water bodies and the nearby wetlands are affected and damaged due to the following factors: Urbanization: Wetlands near urban centers are under increasing developmental pressure for residential, industrial and commercial facilities. Urban wetlands are essential for saving public water supplies.

1.2 Various Types of Water Contaminants:

The following are general categories of various water contaminants and examples of each:

- **Physical** contaminants primarily impact the physical appearance or other physical properties of water. Examples of physical contaminants are sediment or organic material suspended in the water of lakes, rivers and streams from soil erosion.
- **Chemical** contaminants are elements or compounds. These contaminants may be naturally occurring or man-made. Examples of chemical contaminants include nitrogen, bleach, salts, pesticides, metals, toxins produced by bacteria, and human or animal drugs.
- **Biological** contaminants are organisms in water. They are also referred to as microbes or microbiological contaminants. Examples of biological or microbial contaminants include bacteria, viruses, protozoan, and parasites.
- **Radiological** contaminants are chemical elements with an unbalanced number of protons and neutrons resulting in unstable atoms that can emit ionizing radiation. Examples of radiological contaminants include cesium, plutonium and uranium.

1.3 Introduction to IOT:

- The term Internet of Things generally refers to scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate, exchange and consume data with minimal human intervention. There is, however, no single, universal definition.
- **Enabling Technologies:** The concept of combining computers, sensors, and networks to monitor and control devices has existed for decades. The recent confluence of several technology market trends, however, is bringing the Internet of Things closer to widespread reality. These include Ubiquitous Connectivity, Widespread Adoption of IP-based Networking, Computing Economics, Miniaturization, Advances in Data

- **Connectivity Models:** IoT implementations use different technical communications models, each with its own characteristics. Four common communications models described by the Internet Architecture Board include: Device-to-Device, Device-to-Cloud, Device-to-Gateway, and Back-End Data-Sharing. These models highlight the flexibility in the ways that IoT devices can connect and provide value to the user.

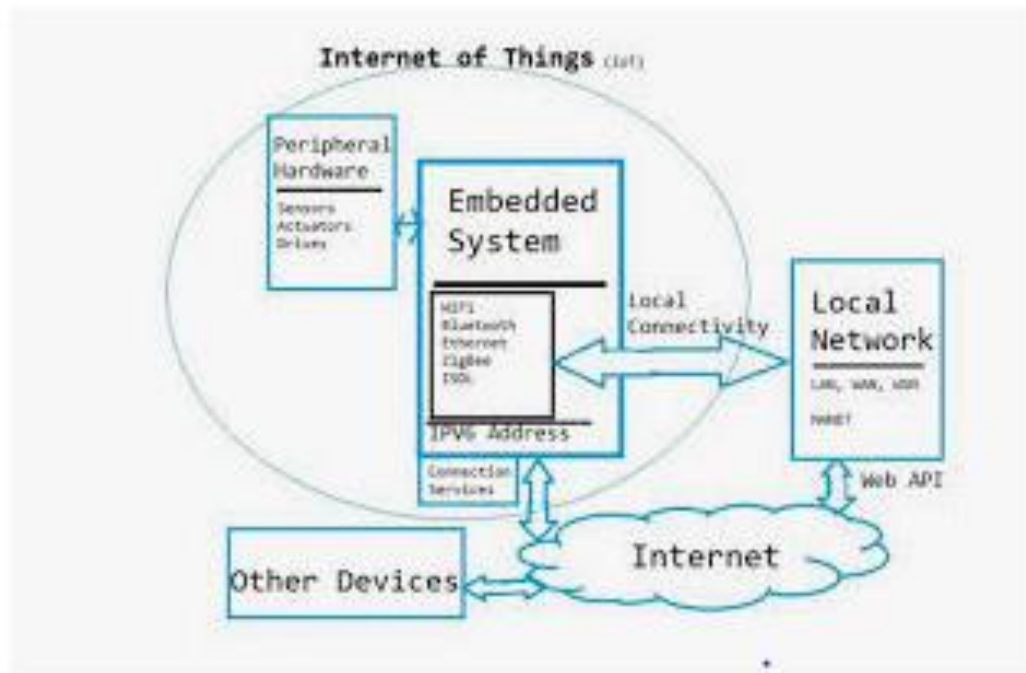


Fig.1.1 Internet of things

IoT devices are implemented using both hardware and software components. Dedicated hardware components are used to implement the interface with the physical world, and to perform tasks which are more computationally complex. Microcontrollers are used to execute software that interprets inputs and controls the system. This module discusses the roles of both the hardware and software components in the system. The functions of common hardware components are described and the interface between the software and hardware through the microcontroller is explained. IoT devices often use an operating system to support the interaction between the software and the microcontroller. We will define the role of an operating system in an IoT device and how an IoT operating system differs from a standard one.

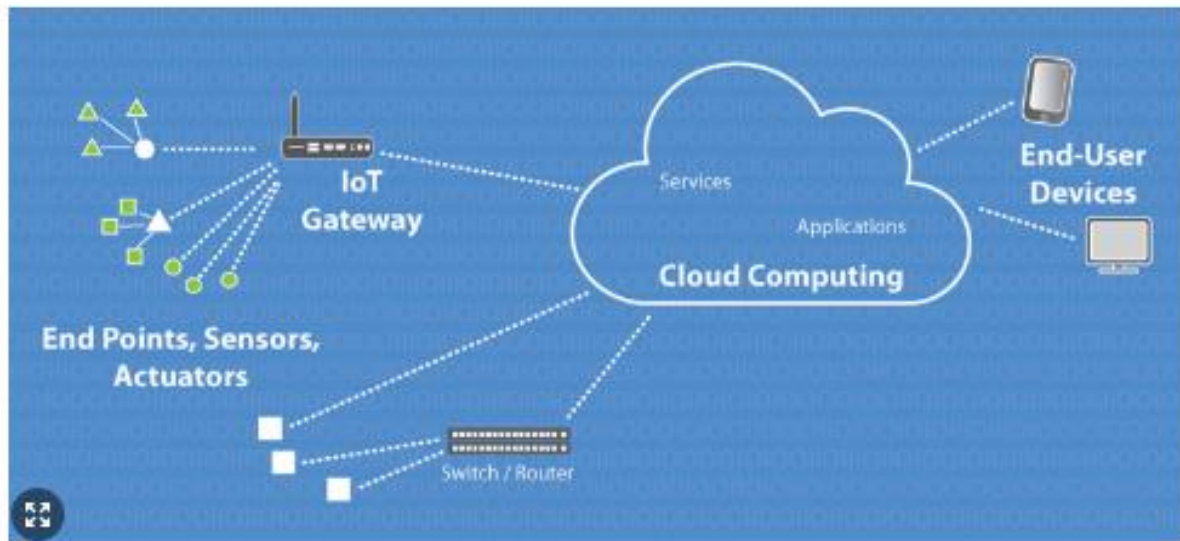


Fig.1.2 Internet of things cloud

1.4 How IoT works

An IoT ecosystem consists of web-enabled smart devices that use embedded processors, sensors and communication hardware to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed .

Example of an IoT system

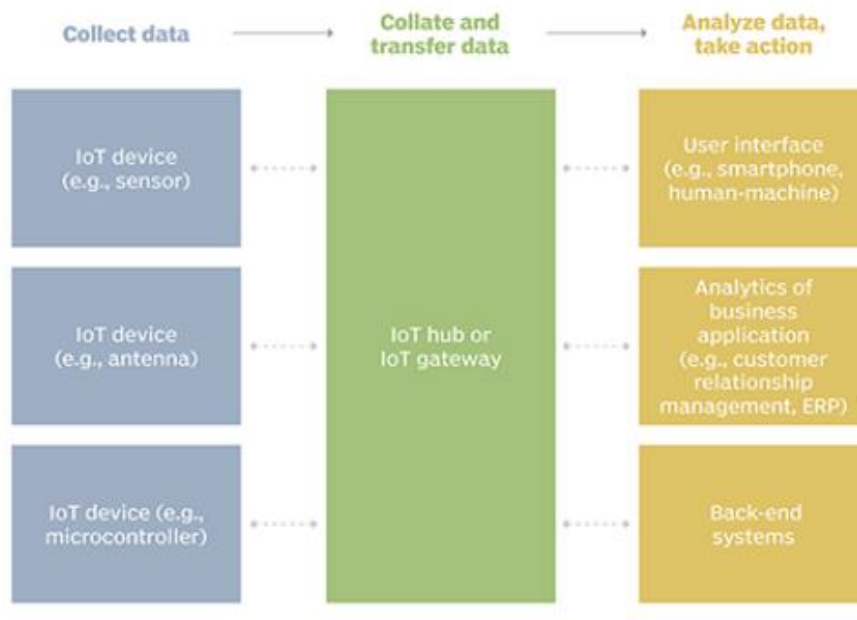


Fig.1.3 Example of IOT

1.5 Benefits of IoT:

The internet of things offers a number of benefits to organizations, enabling them to:

- monitor their overall business processes;
- improve the customer experience;
- save time and money;
- enhance employee productivity;
- integrate and adapt business models;
- make better business decisions; and
- generate more revenue.

IoT encourages companies to rethink the ways they approach their businesses, industries and markets and gives them the tools to improve their business strategies.

Consumer and enterprise IoT applications

There are numerous real-world applications of the internet of things, ranging from consumer IoT and enterprise IoT to manufacturing and industrial IoT (IIoT). IIoT applications span numerous verticals, including automotive, telco, energy and more. In the consumer segment, for example, smart homes that are equipped with smart thermostats, smart appliances and connected heating, lighting and electronic devices can be controlled remotely via computers, smartphones or other mobile devices.

Wearable devices with sensors and software can collect and analyze user data, sending messages to other technologies about the users with the aim of making users' lives easier and more comfortable. Wearable devices are also used for public safety -- for example, improving first responders' response times during emergencies by providing optimized routes to a location or by tracking construction workers' or firefighters' vital signs at life-threatening sites.

In healthcare, IIoT offers many benefits, including the ability to monitor patients more closely to use the data that's generated and analyze it. Hospitals often use IIoT systems to complete tasks such as inventory management, for both pharmaceuticals and medical instruments.

1.6 HISTORY of IIoT:

As of 2016, the vision of the Internet of things has evolved due to a convergence of multiple technologies, including ubiquitous wireless communication, real-time analytics, machine learning, commodity sensors, and embedded systems. This means that the traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things (IIoT). The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold. Mark Weiser's seminal 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of IIoT.

In 1994 Reza Raji described the concept in IEEE Spectrum as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1996 several companies proposed solutions like Microsoft's at Work or Novell's NEST. However, only in 1999 did the field start gathering momentum. Bill Joy envisioned Device to Device (D2D) communication as part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999. The concept of the Internet of things became popular in 1999, through the Auto-ID Centre at MIT and related market-analysis publications. Radio-frequency identification (RFID) was seen by Kevin Ashton (one of the founders of the original Auto-ID Centre) as a prerequisite for the Internet of things at that point. Ashton prefers the phrase "Internet for Things."

If all objects and people in daily life were equipped with identifiers, computers could manage and inventory them. Besides using RFID, the tagging of things may be achieved through such technologies as near field communication, barcodes, QR codes and digital watermarking. In its original interpretation, one of the first consequences of implementing the Internet of things by equipping all objects in the world with minuscule identifying devices or machine-readable identifiers would be to transform daily life. For instance, instant and ceaseless inventory control would become ubiquitous. A person's ability to interact with objects could be altered remotely based on immediate or present needs, in accordance with existing end-user agreements. For example, such technology could grant motion-picture publishers much more control over end-user private devices by remotely enforcing copyright restrictions and digital rights management, so the ability of a customer who bought a Blu-ray disc to watch the movie could become dependent on the copyright holder's decision, similar to Circuit City's failed DIVX.

1.7 APPLICATIONS:

According to Gartner, Inc. (a technology research and advisory corporation), there will be nearly 20.8 billion devices on the Internet of things by 2020. ABI Research estimates that more than 30 billion devices will be wirelessly connected to the Internet of things by 2020. As per a 2014 survey and study done by Pew Research Internet Project, a large majority of the technology experts and engaged Internet users who responded—83 percent—agreed with the notion that the Internet/Cloud of Things, embedded and wearable computing

As such, it is clear that the IOT will consist of a very large number of devices being connected to the Internet. In an active move to accommodate new and emerging technological innovation, the UK Government, in their 2015 budget, allocated £40,000,000 towards research into the Internet of things. The former British Chancellor of the Exchequer George Osborne, posited that the Internet of things is the next stage of the information revolution and referenced the inter-connectivity of everything from urban transport to medical devices to household appliances. The ability to network embedded devices with limited CPU, memory and power resources means that IOT finds applications in nearly every field. [48] Such systems could be in charge of collecting information in settings ranging from natural ecosystems to buildings and factories, thereby finding applications in fields of environmental sensing and urban planning.

On the other hand, IOT systems could also be responsible for performing actions, not just sensing things. Intelligent shopping systems, for example, could monitor specific users' purchasing habits in a store by tracking their specific mobile phones. These users could then be provided with special offers on their favourite products, or even location of items that they need, which their fridge has automatically conveyed to the phone. Additional examples of sensing and actuating are reflected in applications that deal with heat, water, electricity and energy management, as well as cruise-assisting transportation systems. Other applications that the Internet of things can provide is enabling extended home security features and home automation. The concept of an "Internet of living things" has been proposed to describe networks of biological sensors that could use cloud-based analyses to allow users to study DNA or other molecules. However, the application of the IOT is not only restricted to these areas. Other specialized use cases of the IOT may also exist. An overview of some of the most prominent application areas is provided here.

CHAPTER 2

LITERATURE SURVEY

In 2018 Jyotirmaya Ijaradar and Subhasish Chatterjee have discussed the novel approach for Real Time Water Quality Monitoring System. In this they have recommended that the Water contamination is probably the greatest dread for the green globalization. To forestall the water contamination, first we need to appraise the water boundaries like pH, turbidity, conductivity and so on, as the varieties in the estimations of these boundaries point towards the presence of poisons. They have utilized implanted framework programming in Raspberry Pi and cell organizations. In this framework there is no mix with web application.

Mr.Kiran Patil , Mr.Sachin Patil, Mr. Sanjay Patil and Mr.Vikas Patil in 2015 they have talked about the strategies in Monitoring of Turbidity, PH and Temperature of Water Based on GSM and the disadvantages of the customary technique for testing Turbidity, PH and Temperature is to gather tests physically and afterward send them to research center for examination. Notwithstanding, it has been not able to satisfy the needs of water quality checking today. So a bunch of Monitoring of Turbidity, PH and Temperature of Water quality has been created by utilizing the GSM advancements.

Bodepudi SrinivasaRao and U.Jyothi Kameswari in 2012 have built up a Monitoring System of Aquacluture with Automatic Control System Using ARM 7. The significance of water quality in Aquaculture and plan a natural elements observing framework with Automatic control framework. In this framework they have utilized a PC to settle the framework, information preparing and examination by utilizing the assistance of control boundaries. Framework computerization has been utilized in it.

Dongxian He , Daoliang Li , Jie Bao , Hu Juanxiu , and Shaokun Lu had built up a Water-Quality Dynamic Monitoring System Based on Web-worker inserted Technology for Aquaculture In this they have portrayed the Water-quality observing innovation and It has been taken an incredible consideration in hydroponics due to expanding sea-going items and effect of food handling horrendous occurrences. To meet the innovation prerequisites of water-quality observing in multi-boundary, dynamic, and organized checking, a water-quality powerful observing framework was created to screen various boundaries of water-quality and it is finished by utilizing installed framework.

KUMURA Takahiro, SUZUKI Naofumi, TAKAHASHI Masatake, TOMINAGA Shin, MORIOKA Sumio and Ivan Stoianov in 2015 talked about the novel methodology for Smart Water Management Technology with Intelligent Sensing and ICT for the Integrated Water Systems In this they have examined that the expanding interest for water emerging from worldwide populace development and urbanization as of late is focusing on the water supply as far as possible. Then again, water foundation, for example, pipes has been weakening because of maturing. Because of raise in new advancements they have adjusted the more seasoned renditions of water the board. Also, in this framework they have utilized GSM and GPRS innovation.

Harshwardhan, V.Kharpate and T.H.Nagrare in 2013 have done an examination on Comparative Study on Embedded Systems for Drinking Water Testing dependent on WSN In this they have talked about that "as of now, water quality observing is the fundamental part to our nation's hydroponics. In this paper, various methodologies for a continuous water quality climate observing framework are talked about dependent on remote sensor organization to secure information of ecological variables of water quality". The principle Limitations for this distribution is it is finished by utilizing ZigBee based remote sensor organization.

CHAPTER 3

EXISTING SYSTEM

3.1 EXISTING SYSTEM METHODOLOGIES:

- The parameters involved in the water quality monitoring such as the pH level, turbidity and temperature is measured in real time by the sensors that send the data to the base station or control/monitoring room.
- As the monitoring is intended to be carried out in a remote area with limited access, signal or data from the sensor unit will then be transmitted wirelessly to mail.
- The water purification of the tank is done by purifier system.
- The application of wireless sensor network (WSN) for a water quality monitoring is composed of a number of sensor nodes with a networking capability.
- The application of wireless sensor network (WSN) for a water quality monitoring is composed of a number of sensor nodes with networking capability. Such monitoring system can be setup emphasizing on the aspects of low cost, easy ad hoc installation, easy handling and maintenance.
- The use of wireless system for monitoring purpose will not only reduce the overall monitoring system cost in terms of facilities setup.

3.2 PROBLEM STATEMENT:

- Water tank monitoring systems are only provided for PC environment.
- Manual operations.
- We need a pc to monitor the water level.
- High cost.
- Low efficiency.
- Complex installations.

3.3 OBJECTIVE:

- The objectives of this project firstly, using the IOT technology in day-to-day makes the life better and smarter.
- To monitor the quality of the water and controlling the flow of the water in water plant industry.
- And also to monitor the quality of the water and controlling the flow of the water in water supply at municipal corporation, agriculture fields, aquatic ponds ,etc.
- To monitor the various parameter of the water such as PH level, water level, Temperature of the water, Turbidity and controlling the plant using Wi-Fi technology.

CHAPTER 4

PROPOSED SYSTEM

4.1 PROPOSED METHODOLOGIES:

- In our proposed system, we monitor water tanks on a smart phone. The Automatic water level controller ensures no overflows or dry running of pump there by saves electricity and water.
- We are going to monitor the quality of the water and controlling the flow of the water in water plant industry.
- The various parameter of the water such as PH level, water level, Temperature of the water, Turbidity and controlling the plant using Bluetooth technology.
- We can able to control the flow of the water using the water floe sensor when any parameters exceeds it's maximum level
- We can able to control the fire on that areas means we easily control by firefighting robot.

4.2 ADVANTAGES:

- Easy installation.
- Low maintenance.
- Compact elegant design.
- Flexible for many industries.
- Monitor it from far away.
- Low power consumption.

4.3 BLOCK DIAGRAM:

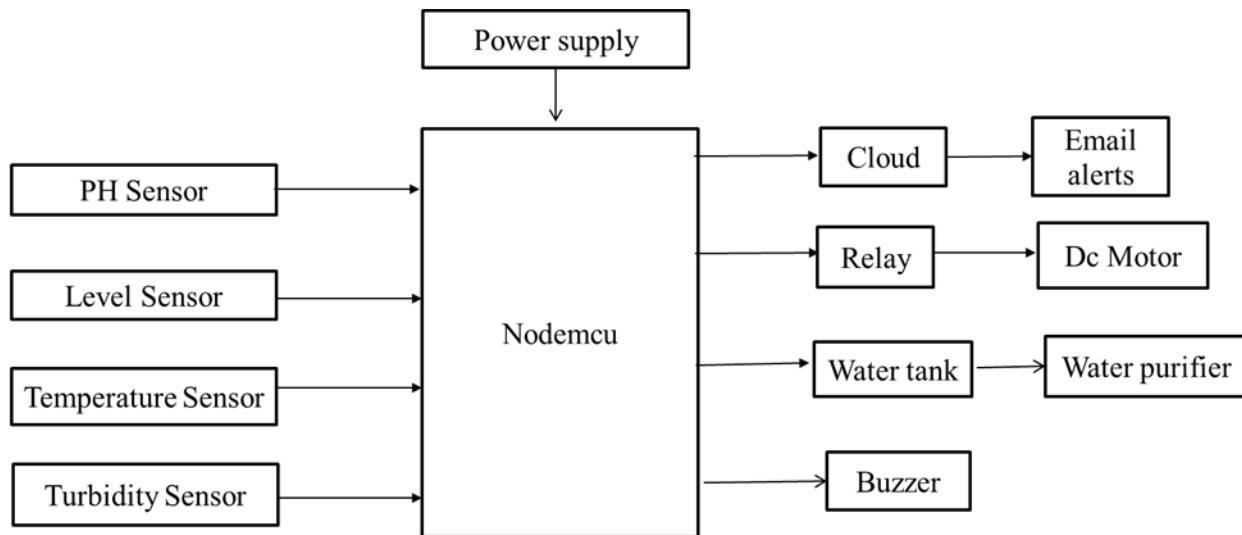


Fig.4.1 Block diagram of water quality monitoring system

4.4 COMPONENTS AND FUNCTIONING

The components used for doing this project were

HARDWARE REQUIEMENTS

- Nodemcu
- Level sensor
- PH Sensor
- Temperature sensor
- Buzzer
- Turbidity sensor
- Ultrasonic sensor
- DC motor
- Relay

SOFTWARE REQUIREMENTS:

- Arduino IDE
- Embedded C

4.5 HARDWARE REQUIREMENTS:

NODEMCU

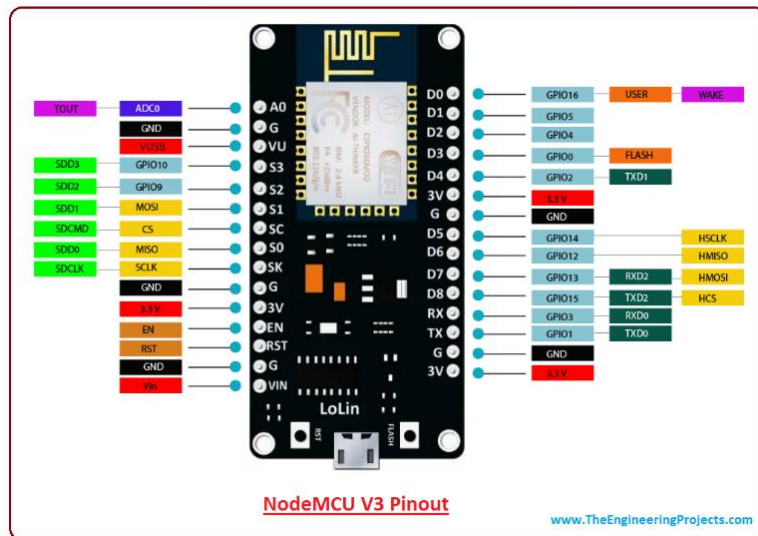


Fig 4.2 Nodemcu

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.^{[6][7]} The term “NodeMCU” by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications. NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month,

Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

NodeMCU is an open source LUA based firmware developed for ESP8266 wifi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e. NodeMCU Development board.

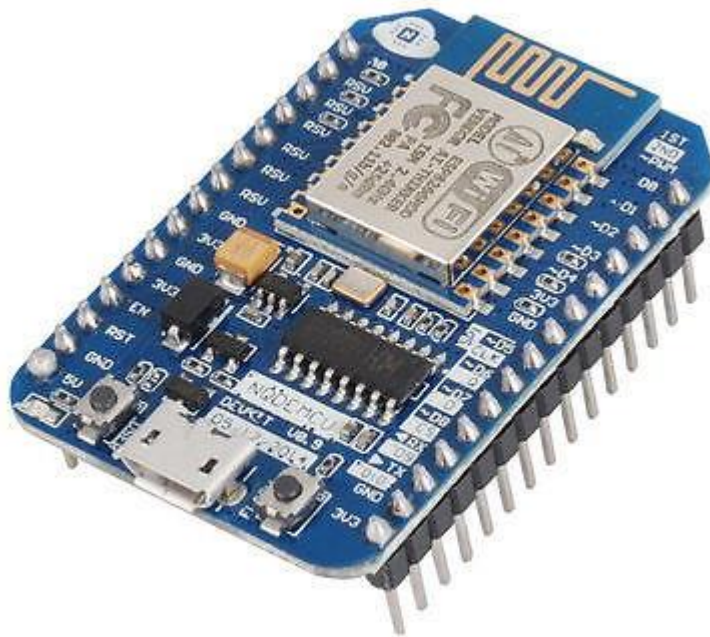


Fig 4.3 NodeMCU Development Board/kit v0.9 (Version1)

Since NodeMCU is open source platform, their hardware design is open for edit/modify/build. NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WiFi Module. There is Version2 (V2) available for NodeMCU Dev Kit i.e. NodeMCU Development Board v1.0 (Version2), which usually comes in black colored PCB.



Fig 4.4 NodeMCU Development Board/kit v1.0 (Version2)

NodeMCU Dev Kit has **Arduino like** Analog (i.e. A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols i.e. UART, SPI, I2C etc. -Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

Here is another way of developing NodeMCU with a well-known IDE i.e. Arduino IDE. We can also develop applications on NodeMCU using Arduino development environment. This makes easy for Arduino developers than learning new language and IDE for NodeMCU.

We need to code in C/C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE. Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executes sequentially. It will not build binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get execute.

In Arduino IDE when we write and compile code, ESP8266 toolchain in background creates binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware.

That's the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware. Since Arduino IDE compile and upload/writes complete firmware, it takes more time than ESPlorer IDE.

Turbidity sensor:



Fig.4.5 Turbidity sensor

Turbidity, the measure of suspended solids in liquids, is utilized as a measure of water quality and can be leveraged as a way for processors to slash waste, improve sustainability and control consumables. Turbidity Meters are engineered to detect the instant a liquid media reaches a pre-defined specification.

As concerns about the impact of industries on the environment grow, turbidity sensors are an ideal way to reduce waste and improve product yield. An inline turbidity meter should be installed at advantage points in the process to facilitate immediate detection of liquid media. Application examples for turbidity meters include:

- Phase separation of products (for example whey – cream – milk)
- Phase changes
 - product-to-product, product-to-water and water-to-cleaning agent
- CIP-return line (monitoring of pre-rinse water)
- Yeast harvest in breweries
- Quality control
- Leak detection of filter and gaskets
- Raw receiving bay

Introduction

The gravity Arduino turbidity sensor detects water quality by measuring the levels of turbidity, or the opaqueness. 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 TSS 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 liquid 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.

Specification

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
 - Analog output: 0-4.5V
- Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5°C~90°C

- Storage Temperature: -10°C~90°C
- Weight: 30g
- Adapter Dimensions: 38mm*28mm*10mm/1.5inches *1.1inches*0.4inches

Connection Diagram

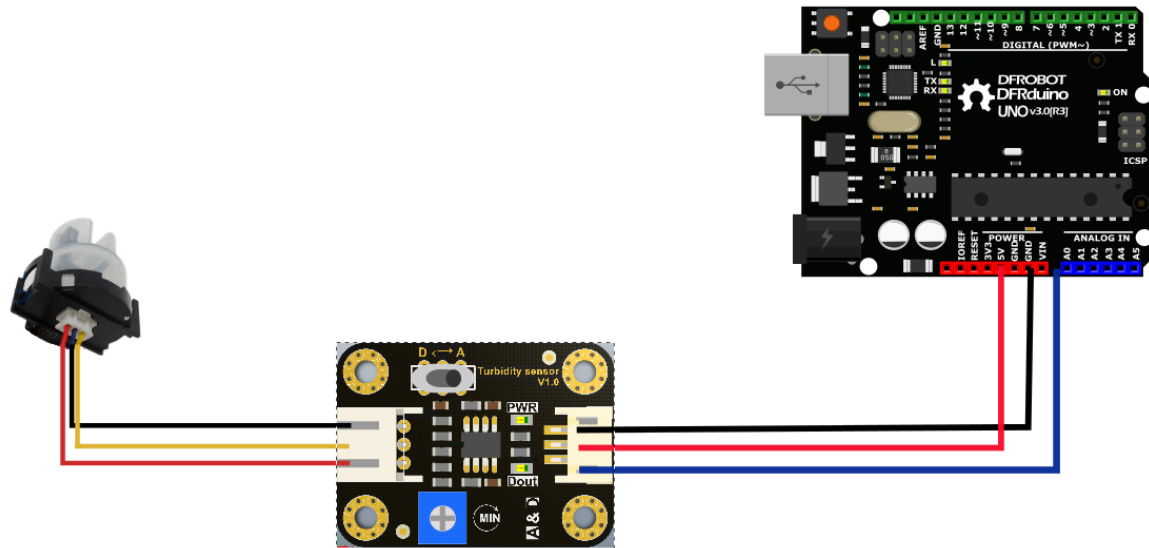


Fig.4.6 Turbidity sensor interface

Interface Description:

"D/A" Output Signal Switch

- "A": Analog Signal Output, the output value will decrease when in liquids with a high turbidity
- "D": Digital Signal Output, high and low levels, which can be adjusted by the threshold potentiometer

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

With the use of LED light sources, turbidity meters determine the level of particulate matter in water or other fluids. Anderson-Negele, our turbidity partner, defines turbidity as “the phenomenon where by a specific portion of a light beam passing through a liquid medium is

reflected by undissolved particles. The sensor measures the light that is reflected by these particles to determine their concentration in the liquid.” For example, purified water would have nearly zero undissolved particles, while ice cream mix has a high concentration.

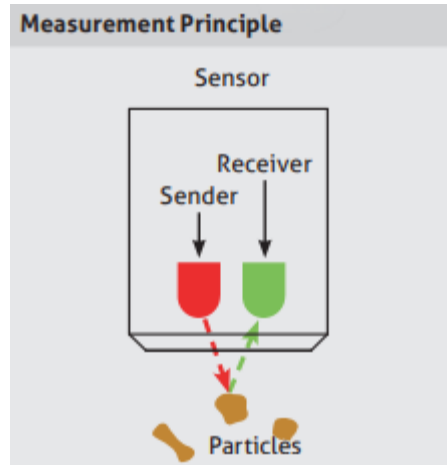


Fig.4.7 Measurement Principle of Turbidity sensor.

Measuring Principle of the Relative Turbidity Meter

An infrared diode irradiates infrared light into the media. Particles in the media reflect the irradiated light which is detected by the receiver diode (backscatter principle). The electronics calculate the relative turbidity of the media according to the received signal.

Turbidity Meter Options

Our turbidity partner, Anderson-negelies-sensors are designed and suitable for inline turbidity measurement. The four-beam turbidity meter is specifically designed to meet the requirements of the fluid food and beverage industries. Sanitary by design, Anderson-Negele turbidity meters meet the highest expectations of accuracy, durability, and cleanability. For high turbidity applications typically found in dairy processing, the ITM-51 Turbidity Meter proves to be a cost-effective insertion sensor for:

- optimizing final flush
- product changeover
- phase transition
- CIP pre rinse control

When low turbidity measurement is needed, the ITM-4 4-beam Turbidity Sensor provides the needed resolution for (COW) water reuse, chill media monitoring and other high purity applications.

Turbidity Meters Produce Rapid ROI

Turbidity meters are a simple and affordable way to generate big payback for your process. From reducing CIP consumables to boosting yield through improved changeovers, turbidity meters are an ideal option for sustainability-minded producers.

pH Sensors:

In the process world, pH is an important parameter to be measured and controlled. The pH of a solution indicates how acidic or basic (alkaline) it is. The pH term translates the values of the hydrogen ion concentration - which ordinarily ranges between about 1 and 10×10^{-14} gram-equivalents per litre - into numbers between 0 and 14. On the pH scale a very acidic solution has a low pH value such as 0, 1, or 2 (which corresponds to a large concentration of hydrogen ions; 10×10^0 , 10×10^{-1} , or 10×10^{-2} gram-equivalents per litre) while a very basic solution has a high pH value, such as 12, 13, or 14 which corresponds to a small number of hydrogeions (10×10^{-12} , 10×10^{-13} , or 10×10^{-14} gram-equivalents per litre). A neutral solution such as water has a pH of approximately 7. A pH measurement loop is made up of three components, the pH sensor, which includes a measuring electrode, a reference electrode, and a temperature sensor; a preamplifier; and an analyser or transmitter.

A pH measurement loop is essentially a battery where the positive terminal is the measuring electrode and the negative terminal is the reference electrode. The measuring electrode, which is sensitive to the hydrogen ion, develops a potential (voltage) directly related to the hydrogen ion concentration of the solution. The reference electrode provides a stable potential against which the measuring electrode can be compared. When immersed in the solution, the reference electrode potential does not change with the changing hydrogen ion concentration. A solution in the reference electrode also makes contact with the sample solution and the measuring electrode through a junction, completing the circuit. Output of the measuring electrode changes with temperature (even though the process remains at a constant pH), so a temperature sensor is necessary to correct for this change in output. This is done in the analyser or transmitter software.

The pH sensor components are usually combined into one device called a combination pH electrode. The measuring electrode is usually glass and quite fragile. Recent developments have replaced the glass with more durable solid-state sensors. The preamplifier is a signal-conditioning device. It takes the high-impedance pH electrode signal and changes it into a low impedance signal which the analyser or transmitter can accept. The preamplifier also strengthens and stabilizes the signal, making it less susceptible to electrical noise.

The sensor's electrical signal is then displayed. This is commonly done in a 120/240 V ac-powered analyser or in a 24 V dc loop-powered transmitter. Additionally, the analyser or transmitter has a man machine interface for calibrating the sensor and configuring outputs and alarms, if pH control is being done. Keep in mind, application requirements should be carefully considered when choosing a pH electrode. Accurate pH measurement and the resulting precise control that it can allow, can go a long way toward process optimisation and result in increased product quality and consistency. Accurate, stable pH measurement also controls and often lowers chemical usage, minimising system maintenance and expense.

A system's pH electrodes require periodic maintenance to clean and calibrate them. The length of time between cleaning and calibration depends on process conditions and the user's accuracy and stability expectations. Overtime, electrical properties of the measuring and reference electrode change. Calibration in known-value pH solutions called buffers will correct for some of these changes. Cleaning of the measuring sensor and reference junction will also help. However, just as batteries have a limited life, a pH electrode's lifetime is also finite. Even in the "friendliest" environments, pH electrodes have to be replaced eventually.

Interfacing pH Sensors:



Fig.4.8 pH sensor interface

Ph sensors, interface and calibration

The pH electrode is essentially a simple single cell battery. The voltage is directly proportional to the hydrogen ion concentration surrounding the electrode. The pH is the logarithm of the hydrogen ion concentration.

The ideal pH electrode:

- Zero volts output at neutral pH (=7.0)
- Positive voltage in acids, $\text{pH} < 7$
- Negative voltages in bases, $\text{pH} > 7$
- Total realistic pH range is 0 to 14.
- Generates -59.16 millivolts per pH unit at room temperature (="Nernst potential"). Note that this is a negative slope--higher pH, lower voltage.
- the full scale range is ± 0.414 volts. ($\pm 0.05916 \times 7$), at 25 degC.
- Temperature coefficient of the Nerst potential is -0.001984 mV per °C. That makes the slope -54.2 millivolts per pH unit at 0 degrees Celsius, and -74.04 millivolts per pH unit at 100 degrees Celsius.

The above values depend somewhat on the construction of the individual electrode, and its aging. That is why it is necessary to calibrate and standardize the pH monitoring and recording instrument from time to time, depending on the conditions it is subjected to. "Standardize" means to adjust the offset so that the instrument reads zero in neutral (pH 7) solution. "Calibrate" means to trim the slope of the pH/mV response to the the correct value for the electrode at that point in time. The instrument should probably have automatic temperature compensation, to adjust the slope in response to different calibration and working temperatures.

The kicker from the electronic standpoint is that the output impedance of the pH electrode is extremely high. The electrode acts like voltage source, however, there is a 10 to 50 mega ohm resistor in series with the voltage. Any voltmeter that measures the output of a pH electrode has to have extremely high input impedance, 1 tera ohm or more. Even many digital multimeters, which have 10 or 20 mega ohms of input resistance, will load down a pH electrode and give a reading that is much lower than it should be.

It is the glass membrane of the probe that is responsible for the high resistance. It a special glass with tiny "pores" that cannot support much electrical current. The usual approach to pH electrodes is to amplify and buffer the signal, with an MOS or CMOS input operational amplifier. When properly constructed with attention to the input circuit layout, it can easily achieve the necessary high input resistance.

LEVEL MEASUREMENT

In industry, liquids such as water, chemicals, and solvents are used in various processes. The amount of such liquid stored can be found by measuring level of the liquid in a container or vessel. The level affects not only the quantity delivered but also pressure and rate of flow in and out of the container Level sensors detect the level of substances like liquids, slurries, granular materials, and powders. The substance to be measured can be inside a container or can be in its natural form (e.g. a river or a lake). The level measurement can be either continuous or point values. Wide spectrum of sensors is available in the market and commonly, they are classified based on the specific application of the sensor.

Sensor used for measuring humidity is termed as humidity sensor, the one used for measurement of pressure is called pressure sensor, sensor used for measurement of displacement is called position sensor and so on though all of them may be using the similar sensing principle. In a similar fashion, the sensor used for measurement of fluid levels is called a level sensor.

Quite obvious from its name, level sensors are used to measure the level of the free-flowing substances. Such substances include liquids like water, oil, slurries, etc as well as solids in granular/powder form (solids which can flow). These substances tend to get settled in the container tanks due to gravity and maintain their level in rest state. Level sensors measure their level against a pre-set reference.

They are important not only in nuclear plants but in lot many applications. Every car, truck and motorcycle is equipped with a fuel level sensor to measure the amount of gasoline left in the fuel tank. In addition, there are sensors for level measurement of engine oil, brake / power steering fluid, cooling water, windshield cleaning liquid, etc. Industrial applications include liquid level sensing in water treatment tanks, transport and storage tanks, in the petrochemical industry for liquids such as petrol, etc. Liquid level measurement is important in household applications for devices such as automated coffee machines, water dispensers, juice squeezers, water evaporators, steamers, fridges and freezers, boilers, heating systems, dishwashers, washing machines, steam irons, etc

In short, level sensors are one of the very important sensors and play very important role in variety of consumer/ industrial applications. As with other type of sensors, level sensors are available or can be designed using variety of sensing principles. Selection of an appropriate type of sensor suiting to the application requirement is very important.



Fig.4.9 Level sensor module

This is an Analog pH Meter Kit with industrial real-time online electrode, specially designed for Arduino and other Microcontrollers. It uses an industry electrode and has built-in simple, convenient, practical connection and long life (up to 1 year), which makes it very suitable for long term online monitoring. It has an LED which works as the Power Indicator, a BNC connector and PH sensor interface. To use it, just connect the pH sensor with BNC connector, and then plug the PH interface into the analog input port of any microcontroller or Arduino controller. If programmed, you will get the pH value easily. Or can measure the voltage on the multi-meter.

PH indicates the sample's acidity but is actually a measurement of the potential activity of hydrogen ions (H^+) in the sample. PH measurements run on a scale from 0 to 14, with 7.0 considered neutral. Solutions with a pH below 7.0 are considered acids. Solutions with a pH above 7.0, up to 14.0 are considered bases. All organisms are subject to the amount of acidity of stream water and function best within a given range.

This industrial pH electrode is made of sensitive glass membrane with low impedance. It can be used in a variety of PH measurements with fast response and excellent thermal stability. It has good reproducibility, is difficult to hydrolysis, and can eliminate basic alkali error. In 0pH to 14pH range, the output voltage is linear. The reference system which consists of the Ag/AgCl gel electrolyte salt bridge has a stable half-cell potential and excellent anti-pollution performance. The ring PTFE membrane is not easy to be clogged, so the electrode is suitable for long-term online detection

Continuous level sensors measure the level to determine the exact amount of substance in a continuous manner.

Point-level sensors indicate whether the substance is above or below the sensing point. This is essential to avoid overflow or emptying of tanks and to protect pumps from dry run.

From the application point of view the considerations are:

- Price
- Accuracy
- Response rate
- Ease of calibration
- Physical size and mounting of the instrument
- Monitoring or control of continuous or discrete levels

Level measurements are broadly classified in two groups:

- Direct methods
- Indirect methods

In direct methods, the level is indicated directly by means of simple mechanical devices. The measurement is not affected by changes in material density. Few examples are:

- Dip Stick
- Resistance Tapes
- Sight Glass
- Floats
- Ultrasonic
- Radar

In Indirect methods, the level is converted in a measurable signal using a suitable transducer. Change in the material affects the measurement. A corrective factor must be used in recalibrating the instrument. Few examples are:

- Hydrostatic head methods
- Load cell
- Capacitance
- Conductivity

Applications:

Level Probes are used for measuring level of

1. Liquids
2. Powdered and granular solids
3. Liquid metals at very high temperature
4. Liquefied gases at very low temperature

Humidity sensor:

Measurement and control of temperature and relative humidity finds applications in numerous areas. These days devices are available which have both temperature and humidity sensors with signal conditioning, ADC, calibration and communication interface all built inside them. The use of such smart sensors greatly simplify the design and reduces the overall cost. We discussed in past about Humidity and temperature measurement. These sensors are capable of measuring both temperature and relative humidity and provide fully calibrated digital outputs. While SHT1x/SHT7x are very accurate sensors, they are still expensive for hobbyists use. This article discusses the DHT11 sensor which also provides calibrated digital outputs for temperature and humidity but is relatively lot cheaper than the Sensirion sensors. The DHT11 sensor uses a proprietary 1-wire protocol which we will be exploring here and implementing with the PIC16F628A microcontroller that will receive the temperature and humidity values from the sensor and display them on a 16×2 character LCD.

About DHT11 sensor

The DHT11 sensor comes in a single row 4-pin package and operates from 3.5 to 5.5V power supply. It can measure temperature from 0-50 °C with an accuracy of $\pm 2^{\circ}\text{C}$ and relative humidity ranging from 20-95% with an accuracy of $\pm 5\%$. The sensor provides fully calibrated digital outputs for the two measurements. It has got its own proprietary 1-wire protocol, and therefore, the communication between the sensor and a microcontroller is not possible through a direct interface with any of its peripherals. The protocol must be implemented in the firmware of the MCU with precise timing required by the sensor.

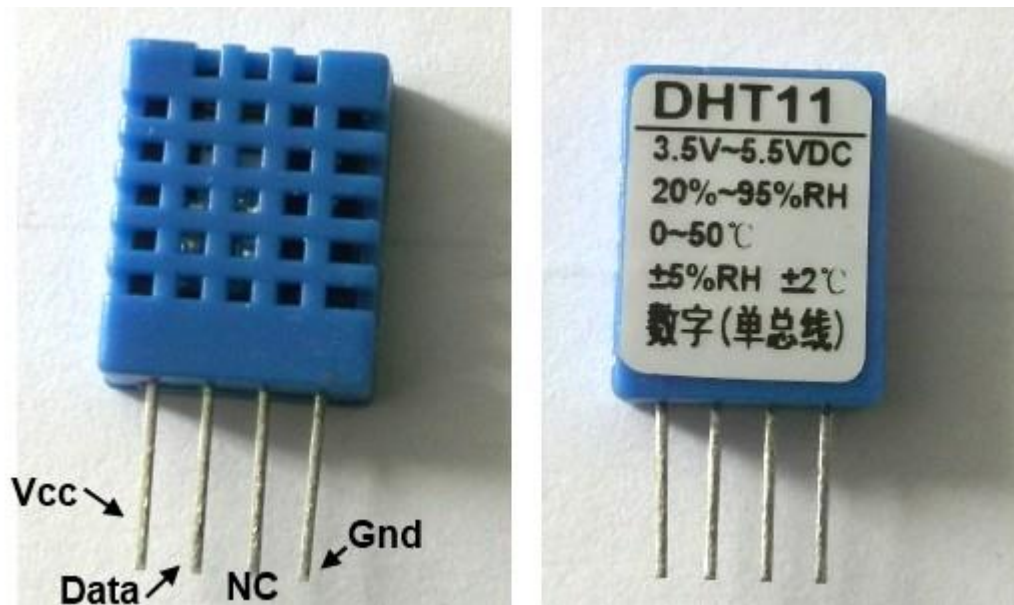
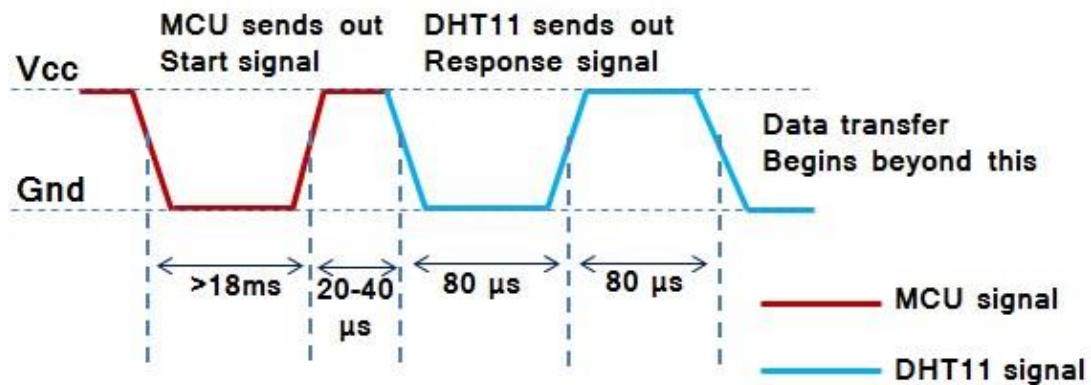


Fig.4.10 DHT11 sensor

DHT11 sensor comes in a single row 4-pin package. The following timing diagrams describe the data transfer protocol between a MCU and the DHT11 sensor. The MCU initiates data transmission by issuing a “**Start**” signal. The MCU pin must be configured as output for this purpose. The MCU first pulls the data line low for at least 18 ms and then pulls it high for next 20-40 μ s before it releases it. Next, the sensor responds to the MCU “**Start**” signal by pulling the line low for 80 μ s followed by a logic high signal that also lasts for 80 μ s. Remember that the MCU pin must be configured to input after finishing the “**Start**” signal. Once detecting the response signal from the sensor, the MCU should be ready to receive data from the sensor. The sensor then sends 40 bits (5 bytes) of data continuously in the data line. Note that while transmitting bytes, the sensor sends the most significant bit first.



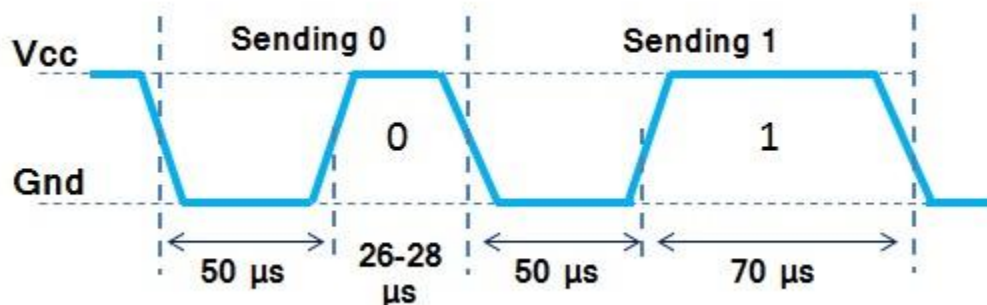
"Start" and "Response" signals The 40-bit data from the sensor has the following structure.

Data (40-bit) = Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp. + Checksum Byte

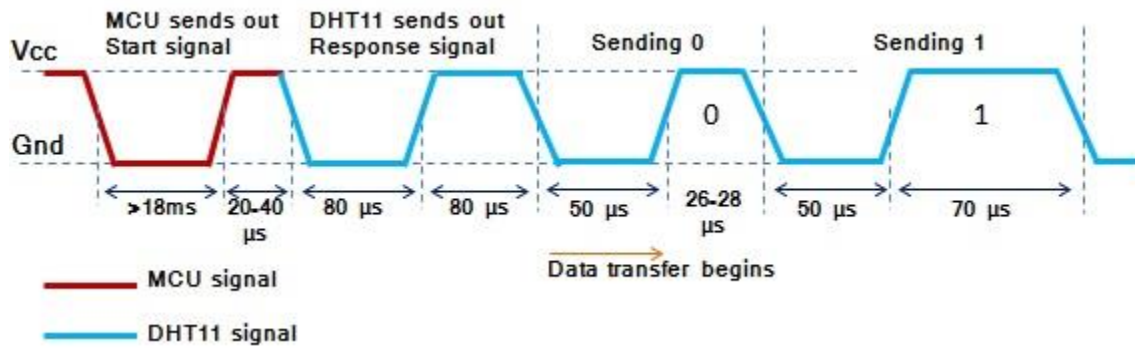
For DHT11 sensor, the decimal bytes of temperature and humidity measurements are always zero. Therefore, the first and third bytes of received data actually give the numeric values of the measured relative humidity (%) and temperature (°C). The last byte is the checksum byte which is used to make sure that the data transfer has happened without any error. If all the five bytes are transferred successfully then the checksum byte must be equal to the last 8 bits of the sum of the first four bytes, i.e.,

Checksum = Last 8 bits of (Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp.)

Now let's talk about the most important thing, which is signalling for transmitting "0" and "1". In order to send a bit of data, the sensor first pulls the line low for 50 µs. Then it raises the line to high for 26-28 µs if it has to send "0", or for 70 µs if the bit to be transmitted is "1". So it is the width of the positive pulse that carries information about 1 and 0.



Timing difference for transmitting "1s" and "0s"



Start, Response and Data signals in sequence

At the end of the last transmitted bit, the sensor pulls the data line low for 50 μs and then releases it. The DHT11 sensor requires an external pull-up resistor to be connected between its Vcc and the data line so that under idle condition, the data line is always pulled high. After finishing the data transmission and releasing the data line, the DHT11 sensor goes to the low-power consumption mode until a new “**Start**” signal arrives from the MCU.

Ultrasonic sensor:



Fig.4.11 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 (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target). In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver.

The formula for this calculation is $D = \frac{1}{2} T \times C$ (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:

$$D = 0.5 \times 0.025 \times 343$$

or about 4.2875 meters.

Relay module:

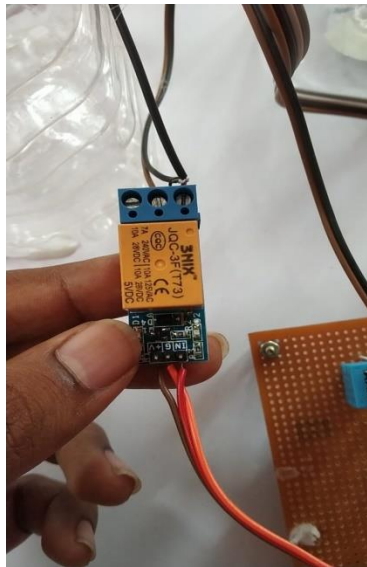


Fig.4.12 Relay module

A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit.

A simple relay consists of wire coil wrapped around a soft iron core, or solenoid, an iron yoke that delivers a low reluctance path for magnetic flux, a movable iron armature and one or more sets of contacts. The movable armature is hinged to the yoke and linked to one or more set of the moving contacts. Held in place by a spring, the armature leaves a gap in the

magnetic circuit when the relay is de-energized. While in this position, one of the two sets of contacts is closed while the other set remains open.

When electrical current is passed through a coil, it generates a magnetic field that in turn activates the armature. This movement of the movable contacts makes or breaks a connection with the fixed contact. When the relay is de-energized, the sets of contacts that were closed, open and breaks the connection and vice versa if the contacts were open. When switching off the current to the coil, the armature is returned, by force, to its relaxed position. This force is usually provided by a spring, but gravity can also be used in certain applications. Most power relays are manufactured to operate in a quick manner.

For distribution of power in high current applications, GEP Power Products is the industry leader in high power relay module design and manufacturing. Rated up to 70 amps, GEP's power relay modules are designed for seamless integration in high power distribution applications. The convenient integral mounting brackets provide easy installation and accessibility. With endless options such as terminal position assurance available for wire retention, GEP Power Products' power distribution solutions and off-road industry knowledge are second to none.

Buzzer:

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play."

Electromechanical:

The electric buzzer was invented in 1831 by Joseph Henry. They were mainly used in early doorbells until they were phased out in the early 1930s in favor of musical chimes, which had a softer tone.

Piezoelectric:

Piezoelectric buzzers, or piezo buzzers, as they are sometimes called, were invented by Japanese manufacturers and fitted into a wide array of products during the 1970s to 1980s. This advancement mainly came about because of cooperative efforts by Japanese manufacturing companies. In 1951, they established the Barium Titanate Application Research Committee, which allowed the companies to be "competitively cooperative" and bring about several piezoelectric innovations and inventions.

Types

Electromechanical

Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz (the contacts buzz a line frequency if powered by alternating current) Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

Mechanical

A joy buzzer is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.

Piezoelectric



Fig.4.13 Piezoelectric disk beeper

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click a ring.



Fig.4.14 Buzzer

Interior of a readymade loudspeaker, showing a piezoelectric-disk-beeper (With 3 electrodes ... including 1 feedback-electrode (the central, small electrode joined with red wire in this photo), and an oscillator to self-drive the buzzer.

A piezoelectric buzzer/beeper also depends on acoustic cavity resonance or Helmholtz resonance to produce an audible beep.

Modern applications

While technological advancements have caused buzzers to be impractical and undesirable, there are still instances in which buzzers and similar circuits may be used. Present day applications include:

- Novelty uses
- Judging panels
- Educational purposes
- Annunciator panels
- Electronic metronomes
- Game show lock-out device
- Microwave ovens and other household appliances
- sporting events such as basketball games
- Electrical alarms
- Joy buzzer (mechanical buzzer used for pranks)

DC MOTOR:

The pumping of water is a basic and practical technique, far more practical than scooping it up with one's hands or lifting it in a hand-held bucket. This is true whether the water drawn from a fresh source, moved to a needed location, purified, or used for irrigation, washing, or sewage treatment, or for evacuating water from an undesirable location. Regardless of the outcome, the energy required to pump water as shown in figure 4.20, is an extremely demanding component of water consumption. All other processes depend or benefit either water descending from a higher elevation or some pressurized plumbing system.

The ancient concept of the aqueduct took simple and eloquent advantage of maintaining elevation of water for as long and far a distance as possible. Thus, as water moves over great distances, it retains a larger component of its potential energy by spending small portions of this energy flowing down a slight gradation. A useful aqueduct system ultimately depends on a fresh water source existing at a higher elevation than the location where the water can be of use. Gravity does all the work. In all other instances, pumps are necessary.



Fig: 4.15 DC MOTOR

where it can be of use. A fresh water source in a lower stream, river, pond, or lake is often pumped to higher ground for irrigation, livestock, cooking, cleaning or other uses by humans, who quite naturally need fresh water. This will purify mostly fresh water and the treatment of largely contaminated water refer endlessly to pumping.

A coil pump is a low lift pump is composed of a tube, shaped as a coil and mounted on a rotating axle powered by an engine or an animal capable of turning the axle around rapidly. Due to the rotation, water then picked up by the tube and pumped upwards in the hose. The coil pump, as many low lift pumps, commonly used for irrigation purposes.

4.6 SOFTWARE REQUIREMENTS:

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: 4.16 Arduino IDE

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.

The 6 Buttons

While more advanced projects will take advantage of the built-in tools in the IDE, most projects will rely on the six buttons found below the menu bar.



The button bar

1. The **check mark** is used to verify your code. Click this once you have written your code.
2. The **arrow** uploads your code to the Arduino to run.
3. The **dotted paper** will create a new file.
4. The **upward arrow** is used to open an existing Arduino project.
5. The **downward arrow** is used to save the current file.
6. The far right button is a **serial monitor**, which is useful for sending data from the Arduino to the PC for debugging purposes.



Fig: 4.17 Arduino IDE Sketch

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.^[5] The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware

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.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

EMBEDDED C:

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems. So, in this article, we will see some of the Basics of Embedded C Program and the Programming Structure of Embedded C. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability. Before digging in to the basics of Embedded C Program, we will first take a look at what an Embedded System is and the importance of Programming Language in Embedded Systems.

Programming Embedded Systems:

As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).

All these devices have one thing in common: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to define how the device actually works.

Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc. From the above statement, it is clear that the Software part of an Embedded System is equally important to the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).

There are many programming languages that are used for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high-level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc. In the process of making a better embedded system, the programming of the system plays a vital role and hence, the selection of the Programming Language is very important.

Factors for Selecting the Programming Language:

The following are few factors that are to be considered while selecting the Programming Language for the development of Embedded Systems.

- **Size:** The memory that the program occupies is very important as Embedded Processors like Microcontrollers have a very limited amount of ROM.
- **Speed:** The programs must be very fast i.e. they must run as fast as possible. The hardware should not be slowed down due to a slow running software.
- **Portability:** The same program can be compiled for different processors.
- **Ease of Implementation**
- **Ease of Maintenance**
- **Readability**

Earlier Embedded Systems were developed mainly using Assembly Language. Even though Assembly Language is closest to the actual machine code instructions, the lack of portability and high amount of resources spent on developing the code, made the Assembly Language difficult to work with.

There are other high-level programming languages that offered the above mentioned features but none were close to C Programming Language.

Introduction to Embedded C Programming Language:

Before going in to the details of Embedded C Programming Language and basics of Embedded C Program, we will first talk about the C Programming Language. The C Programming Language, developed by Dennis Ritchie in the late 60's and early 70's, is the most popular and widely used programming language. The C Programming Language provided low level memory access using an uncomplicated compiler (a software that converts programs to machine code) and achieved efficient mapping to machine instructions.

The C Programming Language became so popular that it is used in a wide range of applications ranging from Embedded Systems to Super Computers. Embedded C Programming Language, which is widely used in the development of Embedded Systems, is an extension of C Program Language. The Embedded C Programming Language uses the same syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc.

The extension in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc. There is actually not much difference between C and Embedded C apart from few extensions and the operating environment. Both C and Embedded C are ISO Standards that have almost same syntax, datatypes, functions, etc. Embedded C is basically an extension to the Standard C Programming Language with additional features like Addressing I/O, multiple memory addressing and fixed-point arithmetic, etc.

Keywords in Embedded C:

A Keyword is a special word with a special meaning to the compiler (a C Compiler for example, is a software that is used to convert program written in C to Machine Code). For example, if we take the Keil's Cx51 Compiler (a popular C Compiler for 8051 based Microcontrollers) the following are some of the keywords:

- bit
- sbit
- sfr
- small
- large

These are few of the many keywords associated with the Cx51 C Compiler along with the standard C Keywords.

Data Types in Embedded C

Data Types in C Programming Language (or any programming language for that matter) help us declaring variables in the program. There are many data types in C Programming Language like signed int, unsigned int, signed char, unsigned char, float, double, etc. In addition to these there few more data types in Embedded C. The following are the extra data types in Embedded C associated with the Keil's Cx51 Compiler.

- bit
- sbit
- sfr
- sfr16

CHAPTER 5

RESULT & CONCLUSION

5.1 Data Processed in Server

After setting up all the connection as for demonstration as shown in the figure 5.1 and the sensors senses the data and send the data to the server through Wi-Fi module as code embedded in the Node MCU functioning the data analysis graph shown in the server.

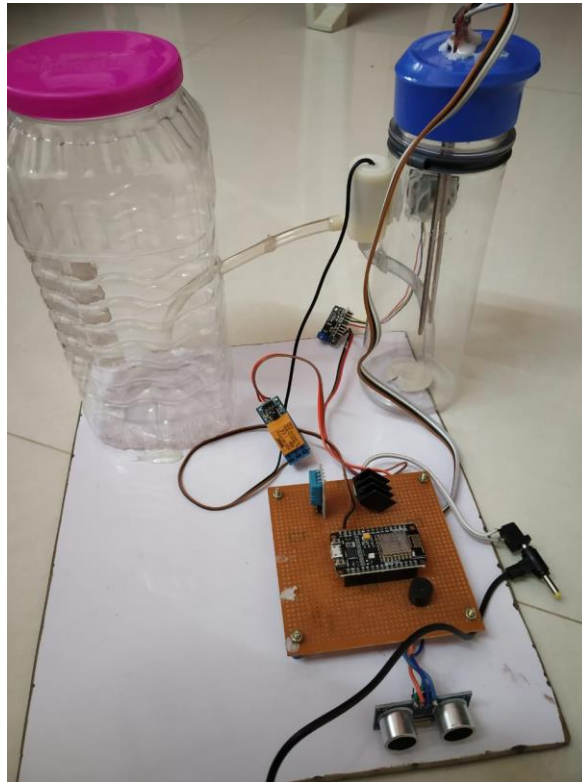


Fig: 5.1: Demonstration

The Data received by the sensors sent into the server. Connecting to the power supply the data been processed into the server. Sensor's values analysis made in the Ubidots cloud.

5.2 IDEAL VIEW

The images shown below used for Ideal view showing the working and Data Processed

in the server. The field with both empty tank and water filled in it conditions were applied accordingly and data been processed. If any sudden change in water level and in other parameters of water such as ph, temperature, etc., alert email will be terminated to the user or or the respective water quality lead and water management can take care of the water quality by providing necessary purification techniques and hence the water purification will be done before sending the water into to the various water bodies and also we can monitor the health of the various aquatic animals by measuring the different water parameters.

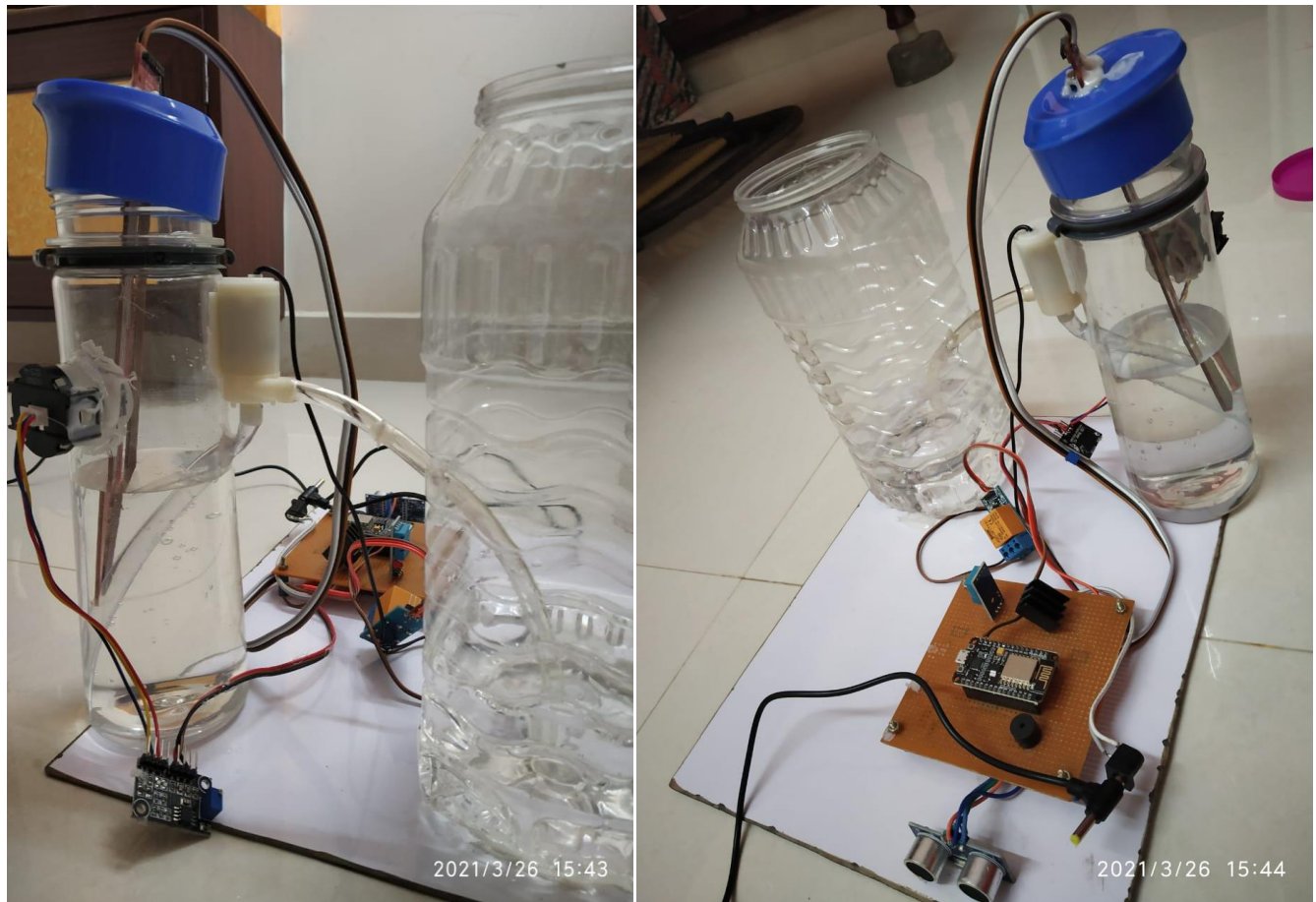


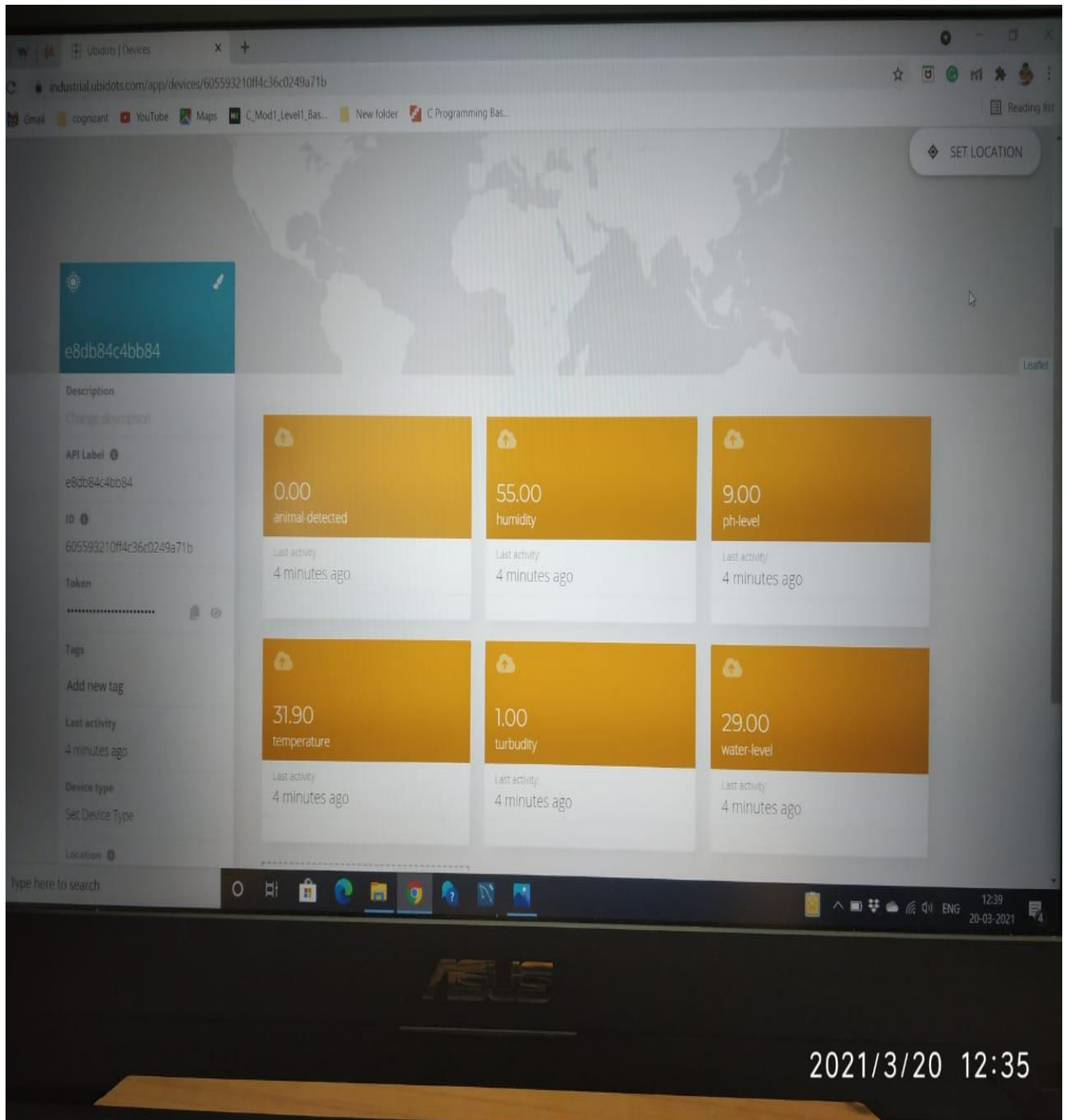
Fig: 5.2: Ideal view with water

5.3 OUTPUT ANALYSIS

The data in the Ubidots will have the generated data from the sensors. The data in the cloud shows the date, time and values of Temperature, Humidity, ph, turbidity, water level, for smart water quality monitoring. Hence this report useful for analysis and predictions for Aquatic Farmers, water corporations, water industry's, etc., As shown in figure 5.4, is the data analysis

on the server. The Fields in the Server were Temp, Humidity, Object detected, ph, Turbidity, water level.

SCREEN SHOTS OF DATA UPLOADED IN THE CLOUD:



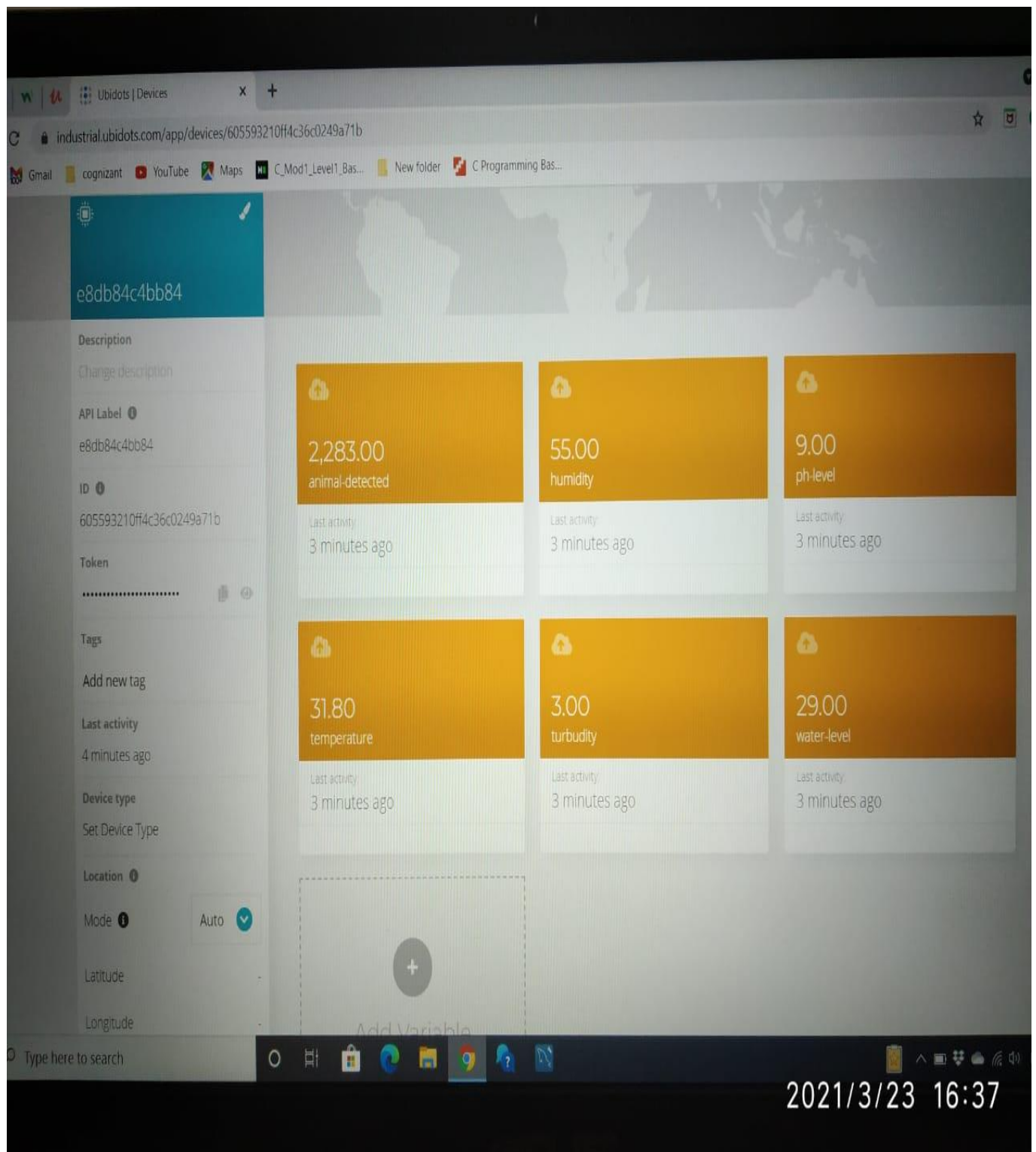


Fig: 5.3 Data in cloud

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 CONCLUSION:

Hence, the parameters involved in the water quality monitoring such as the pH level, turbidity and temperature is measured in real time by the sensors that send the data to the IOT cloud. As the monitoring is intended to be carried out in a remote area with limited access, signal or data from the sensor unit will then be transmitted wirelessly to IOT cloud. The water purification of the tank is done by the external purifier. The application of wireless sensor network (WSN) for a water quality monitoring is composed of a number of sensor nodes with a networking capability. Such monitoring system can be setup emphasizing on the aspects of low cost, easy adhoc installation, easy handling and maintenance. The use of wireless system for monitoring purpose will not only reduce the overall monitoring system cost in terms of facilities setup but also it provides efficient results for the parameters involving in water quality monitoring.

6.2 FUTURE SCOPE:

As, we all know the Water is very essential for the each and every living organism in the world. It is very essential to monitor the quality of water which we regularly use for many ways in our day to day life. The future scope of this particularly inspire the people that they can process or take up the water quality monitoring using this Smart techniques with IoT measures as it makes the things to work smarter and better. The smart Water Quality monitoring will help for the people those you are lacking or busy in their daily needs can refer the water quality by their smartphone. This makes the productivity in maintaining the quality to the aquatic animals growth as well. The further modification that this can afford by making an Application of the data processed. That it is useful for illiteracy people. So, the data been processed and can enhance and have great impact to the smart innovations in large manner.

REFERENCES

1. Jayti Bhatt, Jignesh Patoliya,, “IoT Based Water Quality Monitoring System”, vol. 4, no. 4, pp. 762-766, 2016.
2. Geetha S, Gouthami S, “Internet of Things Enabled Real Time water Quality Monitoring System”, Smart Water International Journal for aqua – Smart ICT for Water, vol. 2, no.1, pp. 3-11, 2017.
3. . Kumar RK, Mohan MC, Venkatesh pandian s, Kumar MM, Eswaran R, “Solar Based Advanced Water Quality Monitoring System Using Wireless Sensor Network’, International Journal of Science Engineering and Technology Research, vol. 3, no.3, pp. 385-389, 2014.
4. Goib Wiranto, Grace A, Mambu, Hiskia, I Dewa Putu Hermida, Slamet Widodo, “Design of Online Data Measurement and Automatic Sampling System for Continuous Water Quality Monitoring”, Proceedings of 2015 IEEE International Conference on Mechatronics and Automation, pp. 2331-2335, 2015.
5. Y. L. Xiong, "Application research of biological monitoring technology in environmental monitoring", Resource. Economization Environ. Protection, vol. 12, no. 6 pp. 93-98, 2016.
6. Fei Yuan, Yi-fan Huang, Xin Chen “A Biological Sensor System Using Computer Vision for Water Quality Monitoring ” October 2018. IEEE Access PP(99):1-1
7. j.j.Emery, D.B.Matchett “Waste Management Strategies for industries.”-2003
8. Shrivastava, A.K., A Review on copper pollution and its removal from water bodies by pollution control Technologies, IJEP 29(6): 552-560, 2009.
9. Journal of environmental Management, vol. 88, issue 3, August 2008, pp. 437-447.
- 10.Industrial wastewater reuses potential – internet (web)