**A**

**PROJECT REPORT**

**ON**

IoT Based Water Level Monitoring

SUBMITTED IN

PARTIAL FULFILLMENT OF

**DIPLOMA IN EMBEDDED SYSTEM DESIGN (PG-DESD)**



**BY**

**Ashwini Wagh**

**AT**

**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**

**PUNE**

|  |  |
| --- | --- |
| **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE.** | |
|  | |
|  | |
| **CERTIFICATE** | |
|  | |
| This is to certify that the project | |
|  | |
| IoT Based Water Level Monitoring | |
| Has been submitted by | |
| **Ashwini Wagh** | |
| In partial fulfillment of the requirement for the Course of **PG Diploma in Embedded System Design (PG-DESD FEB - 2019)**as prescribed by The **CDAC** ACTS, PUNE. | |
|  | |
| Place: Pune | Date: 30 July 2019 |
|  | |
|  | **Authorized Signature** |
|  | |

**ACKNOWLEDGEMENT**

It gives me a great pleasure to submit Project-Seminar report. This is the only page where I have the opportunity to express my emotions and gratitude from the bottom of my heart.

I express my sincere thanks to my guide **Mr.Deep Ranjan Deb** for guiding me at every step in making of this project. He motivated us and boosted my confidence and I must admit that the work would not have been accomplished without his guidance and encouragement.

I would like to extend my special thanks to **Mr. Devendra Dhande Sir** and **Mr. Nilesh Ghule Sir** for spending their valuable time to go through my report and providing many helpful suggestions. Lastly I would like to thank all the staff member of electronics department and my friends without whom the project report would not have been completed.

**Ashwini Wagh(28193)**

**ABSTRACT**

Water is always a crucial part of everyday life. Due to global environmental situation, water management and conservation is vital for human survival. In recent times, there were huge needs of consumer based humanitarian projects that could be rapidly developed using Internet of Things (IoT) technology. Water source is fundamental for all and a significant factor in agriculture, cultivating and it’s a key for nature of our life. Monitoring water level of a water supply, For example, Lakes, River, Waterways and Pond etc., Plays a major part in rural and agricultural. Even it’s helpful for our every day needs. For example the amount of water drops under the edge level in a bore well, The motor pump may get affected due to dry running. There are many several alternative things wherever water level monitoring is an important task. This report proposes a prototype system design, implementation and description of needed devices and technologies to improve Internet of Things (IoT) based water level monitoring. Ultrasonic Sensor is used to measuring water level. The calculated values from the sensors can be processed by the Microcontrollers and uploaded to the internet through the Wi-Fi module (ESP 8266). Analysis can also be done on sensed data to pick out for the solutions. The invented system is used some IoT modules for accessing sensor data from the core controller to the cloud. The data which is obtained from the sensors can be shown on the internet and provides facilities for screening the data on mobile phones or web application.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **TITLE** | **Page No.** |
| **1.** | **INTRODUCTION** | 7 |
|  | 1.1 Introduction | 7 |
|  | 1.2 Overview of Project | 7 |
| **2.** | **LITERATURE SURVEY** | 8 |
| **3.** | **PROJECT ARCHITECTURE** |  |
|  | 3.1 Block Diagram | 9 |
|  | 3.2 Flow Chart | 11 |
| **4.** | **HARDWARE DESCRIPTION** |  |
|  | 4.1 Ultrasonic ranging module (HCSR-04) | 12 |
|  | 4.2 ESP8266 | 13 |
| **5.** | **SOFTWARE REQUIREMENTS AND SPECIFICATIONS** | 16 |
| **6.** | **SOURCE CODE EXPLAINATION** | 18 |
| **7.** | **TESTING** | 26 |
| **8.** | **FUTURE SCOPE** | 28 |
| **9.** | **CONCLUSION** | 29 |
| **10.** | **REFERNCES** | 30 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Section** | **Table Title** | **Page** |
| 4 | Electrical Parameters of HCSR04 | 12 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Section** | **Figure Title** | **Page** |
| 3 | Block Diagram | 9 |
| 3 | Flow chart | 11 |
| 4 | Pin diagram of ESP8266 | 14 |
| 4 | Functional Block Diagram | 14 |
| 6 | Activity Life Cycle | 24 |

**INTRODUCTION**

Water has gained an inestimable value in the recent past. This scenario arises mainly because of the various faces. Population, ageing infrastructure, dilapidated technologies and ground water contamination have been the fore-runners of the many challenges that water faces. These challenges have pushed drinking water to an extent where it has become an amenity rather than utility. This phenomenon of scanty drinking water demands a methodology which is both achievable and efficient. One such process is the over-head tank and quality monitor system. In this process a group of tanks situated in locality are considered as an entity. An ultrasonic sensor is used to detect the water level of the tank and to monitor the quality an Ultrasonic-sensor and is used. Observing the level of water in the tank we can regulate the amount of water that locality is using. Through this observation we could send extra amount of water to that locality if the demand is more or consequently curb the amount of water if the supply is more to avoid wastage of drinking water. Quality parameters help in purity check. If in case the results show the water is impure and turbid the information through the module is sent to the municipal department which employs workers to take sanitary measures of cleaning the tank so the water could regain its purity. The data from the sensor is collected and Smart water tank executes IoT, with which, the client can specifically screen and control the working of tank through the cell phone and from wherever in this world. The android application is made with the goal that the information assembled from the cloud is sent to app. From application civil division can send workers to clean the tank in view of the area.

**LITERATURE SURVEY**

Level controlling of water or liquid is a very active field

where many papers have addressed the fuzzy or neural

Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural Level controlling of water or liquid is a very active field

where many papers have addressed the fuzzy or neural

Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural networks control in the water level control system. It requires measurement and control of the water-level. Water is a limited resource and is essential for agriculture, industry and for creature’s existence on earth including human beings. Lots of people don’t realize the true importance of drinking enough water every day. More water are wasted by many uncontrolled way. This problem is quietly related to poor water allocation, inefficient use, and lack of adequate and integrated water management. Therefore, efficient use and water monitoring are potential constraint for home or office water management system. Every living thing on earth needs water to survive. Human bodies are made up of more than 60 percent water. We use clean water to drink, grow crops for food, operate factories, and for swimming, surfing, fishing and sailing. Water is vitally important to every aspect of our lives. Monitoring the quality of surface water will help protect our waterways from pollution. Farmers can use the information to help better manage their land and crops. Our local, state and national governments use monitoring information to help control pollution level.

As we see, now a days the safety control of large dams depends mainly on the measurement of some important quantities like absolute and relative displacements, strains and stresses in the concrete, discharges through the foundation, etc. and on visual inspections of the dam structures. In certain dams, the analysis of the measured data is compared with results of mathematical or physical models and is helpful in the structural safety assessment.

1. IoT Based Water level Monitoring For Smart Village

Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural networks control in the water level control system. It requires measurement and control of the water-level. Water is a limited resource and is essential for agriculture, industry and for creatures existence on earth including human beings. Lots of people don’t realize the true importance of drinking enough water every day. More water are wasted by many uncontrolled way. This problem is quietly related to poor water allocation, inefficient use, and lack of adequate and integrated water management. Therefore, efficient use and water monitoring are potential constraint for home or office water management system. Every living thing on earth needs water to survive. Human bodies are made up of more than 60 percent water. We use clean water to drink, grow crops for food, operate factories, and for swimming, surfing, fishing and sailing. Water is vitally important to every aspect of our lives. Monitoring the quality of surface water will help protect our waterways from pollution. Farmers can use the information to help better manage their land and crops. Our local, state and national governments use monitoring information to help control pollution level.

As we see, now a days the safety control of large dams depends mainly on the measurement of some important quantities like absolute and relative displacements, strains and stresses in the concrete, discharges through the foundation, etc. and on visual inspections of the dam structures. In certain dams, the analysis of the measured data is compared with results of mathematical or physical models and is helpful in the structural safety assessment.

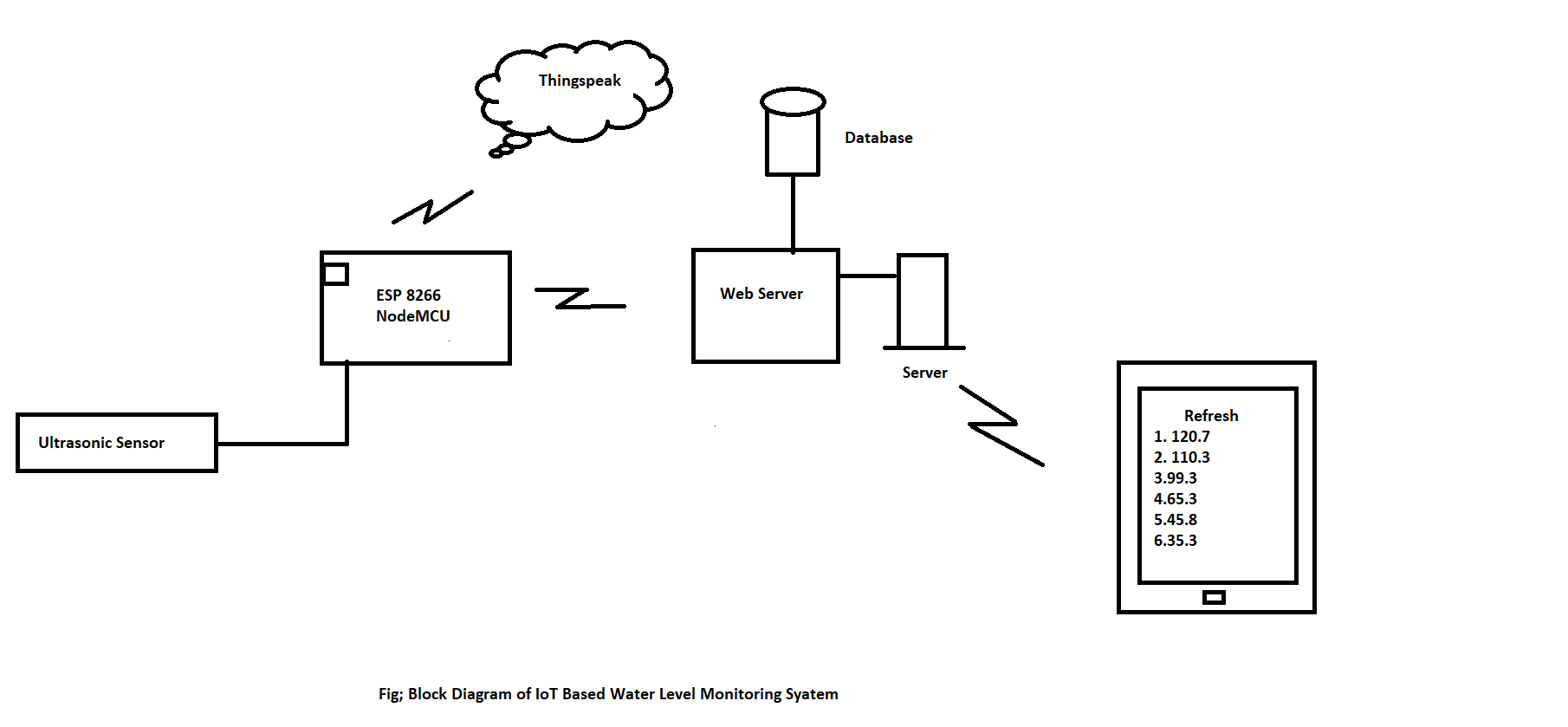
1. Water Level Monitoring and Management Using IoT.

Level controlling of water or liquid is a very active field where many papers have addressed the fuzzy or neural networks control in the water level control system. It requires measurement and control of the water-level. One of the easiest way to measure water level is using submersible pressure transducers (wet sensors) which are easy to install and require very little maintenance. For the very reason, they are often used for temporary installations and installations in remote locations. They are supposed to be installed in a fixed position and should remain fully submersed at all times. It works on the concept of application of hydrostatic pressure to a strain gauge, which converts mechanical movement into an electrical signal which is in turn measured by the station data logger and converted into pressure, level and discharge. As we see, now a days the safety control of large dams depends mainly on the measurement of some important quantities like absolute and relative displacements, strains and stresses in the concrete, discharges through the foundation, etc. and on visual inspections of the dam structures. In certain dams, the analysis of the measured data is compared with results of mathematical or physical models and is helpful in the structural safety assessment.

Artificial neural networks (ANNs) provide a quick and flexible means of creating models for river flow prediction, and have been shown to perform well in comparison with conventional methods. These networks can be used for characterizing the normal structural behavior of the dams by taking into account the actions to which the structure is subject to in the past. An artificial neural network is nothing but an interconnected group of nodes, similar to the vast network of neurons in a brain. Each circular node of this network represents an artificial network and an arrow represents flow from the output of one node to the input of another.

**PROJECT ARCHITECTURE**

**Block Diagram**



The Project will inform the user about the water level. In this ESP8266 acts as a client. The Ultrasonic Sensor which is connected to the ESP8266 is going to sense the duration of that pulse . The pulse will going to take the return back since it is transmitted .we have connected it into distance & the distance value is sent further to the web server by ESP8266. By making request onto the web server further web server will send to the database & from database android application will going to fetch the latest 20 values when user clicks on ‘Refresh’ button. Using ThingSpeak we have to show the level. Let’s go to the network detail.

We have used HCSR04 Ultrasonic sensor operates well in the range of 2cm to 400cm with frequency of 40 Hz. It is connected to the GPIO of ESP 8266. It has four pins Vcc, Gnd, TRIG & ECHO. Vcc requires 5V power supply for operating HCSR04. TRIG pin is configured as output which will going to send trigger in the burst of 8 pulses when it will hit the water pulses the pulses returns the turnaround time i.e. time since pulse is sent & returns back, the distance can be found by the formula,

Distance = Velocity of sound(340ms) \* Time/2

This distance is then sent to the server by sending request on the server. We are using express server. Express is Node.js web application framework that provides set of features for web & mobile when client makes request. Depending onthe request, server sends response to the client. While inserting values in database it will going to parse the distance value from the query entered in URL by executing insert command it will add the values on database module so that through http methods like get post we can send request onto the server & server will going to respond i.e. either going to insert the values on database or will fetch the data from the database. In our android application we have developed login page which will going to navigate to the page after user entered correct username & password. On second page as refresh button is clicked the application will going to fetch the data from the database & will show the latest twenty values of distance. We have shown graphical view of the container by using ThingSpeak which is an open source IoT Application API to store & retrieve data from things using HTTP protocol over Internet or via LAN.

We can use ThingSpeak to provide real time updates on our own webpage. In ThingSpeak we have to create channel then we have to pass the API key & channel Id in the ESP8266 code so that the values will be get loaded on ThingSpeak& we will get updated graphical presentation.

**Flowchart**

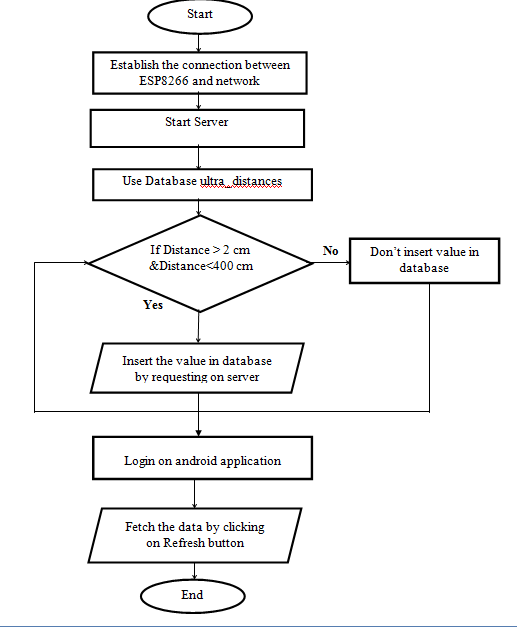


Fig. Flowchart of IoT based water level monitoring

**Hardware Description:**

1. **Ultrasonic Ranging Module HC - SR04:**

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves.  
The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

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

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

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

**Electric Parameter:**

|  |  |
| --- | --- |
| Working Voltage | DC5V |
| Working Current | 15mA |
| Working Frequency | 40Hz |
| Max Range | 4cm |
| Min Range | 2cm |
| Measuring Angle | 15 degree |
| Trigger Input Signal | 10uS TTL pulse |
| Echo Output Signal | Input TTL lever signal and the range in  Proportion |
| Dimension | 45\*20\*15mm |

Table Electrical Parameters of HCSR04

1. **ESP8266 (Wi-Fi module):**

The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications.

ESP8266EX (simply referred to as ESP8266) is a system-on-chip (SoC) which integrates a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules into a small package. It provides capabilities for 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2), general-purpose input/output (16 GPIO), Inter-Integrated Circuit (I²C), analog-to-digital conversion (10-bit ADC), Serial Peripheral Interface (SPI), I²S interfaces with DMA (sharing pins with GPIO), UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and pulse-width modulation (PWM). The processor core, called "L106" by Espressif, is based on Tensilica's Diamond Standard 106Micro 32-bit processor controller core and runs at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 32 KB instruction RAM, and 80 KB user data RAM. (Also, 32  KB instruction cache RAM and 16 KB ETS system data RAM.) External flash memory can be accessed through SPI. The silicon chip itself is housed within a 5 mm × 5 mm Quad Flat No-Leads package with 33 connection pads — 8 pads along each side and one large thermal/ground pad in the center.

**What is NodeMCU?**

NodeMCU is a firmware that allows you to program the ESP8266 modules with LUA script. 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. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language.

**Specification:**

* Voltage:3.3V.
* Wi-Fi Direct (P2P), soft-AP.
* Current consumption: 10uA~170mA.
* Flash memory attachable: 16MB max (512K normal).
* Integrated TCP/IP protocol stack.
* Processor: Tensilica L106 32-bit.
* Processor speed: 80~160MHz.
* RAM: 32K + 80K.
* GPIOs: 17 (multiplexed with other functions).
* Analog to Digital: 1 input with 1024 step resolution.
* +19.5dBm output power in 802.11b mode
* 802.11 support: b/g/n.

**Pin Diagram of ESP8266:**

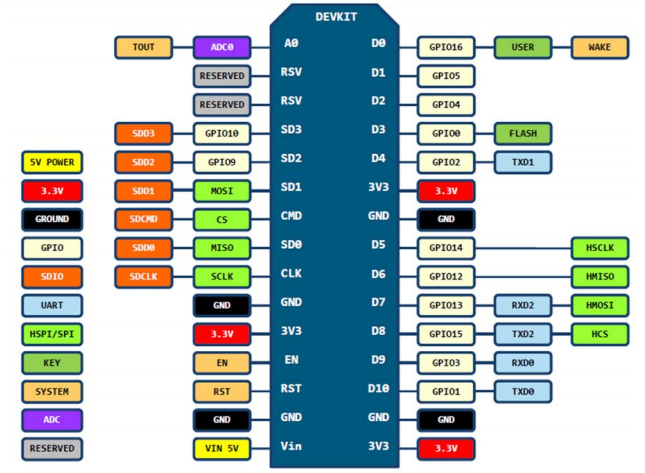


Fig 3. Pin Diagram of ESP8266

**Functional Description of ESP8266:**

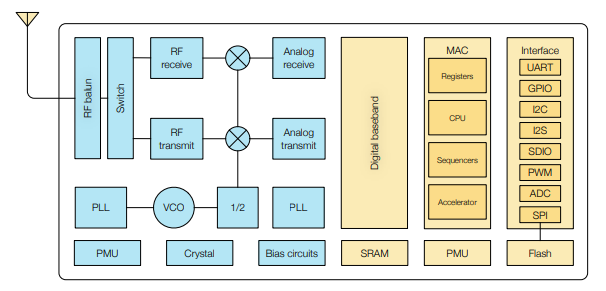


Fig 4. Functional Block Diagram

* CPU, Memory, and Flash:

CPU:

The ESP8266EX integrates a Tensilica L106 32-bit RISC processor, which achieves extra low power consumption and reaches a maximum clock speed of 160 MHz. The Real-Time Operating System (RTOS) and Wi-Fi stack allow 80% of the processing power to be available for user application programming and development. The CPU includes the interfaces as below:

• Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller, and can also be used to visit flash.

• Data RAM interface (dBus), which can be connected with memory controller.

• AHB interface which can be used to visit the register.

Memory :

ESP8266EX Wi-Fi SoC integrates memory controller and memory units including SRAM and ROM. MCU can access the memory units through iBus, dBus, and AHB interfaces. All memory units can be accessed upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor.

• RAM size < 50 kB, that is, when ESP8266EX is working under the Station mode and connects to the router, the maximum programmable space accessible in Heap + Data section is around 50 kB.

External Flash:

ESP8266EX uses external SPI flash to store user programs, and supports up to 16 MB memory capacity theoretically. The minimum flash memory of ESP8266EX is shown below:

• OTA disabled: 512 kB at least

• OTA enabled: 1 MB at least

**Software Requirement Specification:**

**Arduino IDE 1.8.2:**

The most basic way to use the ESP8266 module is to use serial commands, as the chip is basically a WiFi/Serial transceiver. However, this is not convenient. Hence we have used Arduino IDE 1.8.2 for flashing into ESP8266. One can use Arduino IDE 1.6.4 or greater.

The [Arduino](https://en.wikipedia.org/wiki/Arduino) integrated development environment ([IDE](https://en.wikipedia.org/wiki/Integrated_development_environment)) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2. The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) 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](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution.

**Android Studio:**

Android Studio is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s Android operating, built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains)'[IntelliJ IDE. A](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development). It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems. It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as the primary IDE for native Android application development.

The following features are provided in the current stable version:

* [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support
* Android-specific [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes
* [Lint](https://en.wikipedia.org/wiki/Lint_(software)) tools to catch performance, usability, version compatibility and other problems
* [ProGuard](https://en.wikipedia.org/wiki/ProGuard_(software)) integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations
* Support for building [Android Wear](https://en.wikipedia.org/wiki/Android_Wear) apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of [IntelliJ](https://en.wikipedia.org/wiki/IntelliJ) (and [CLion](https://en.wikipedia.org/wiki/CLion)) e.g. [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), and more with extensions, such as [Go](https://en.wikipedia.org/wiki/Go_(programming_language));and Android Studio 3.0 or later supports [Kotlin](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects [backport](https://en.wikipedia.org/wiki/Backporting) some Java 9 features. While IntelliJ that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

**ThingSpeak:**

**What is ThingSpeak?**

ThingSpeak is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs.

We will consider each of these features in detail below.

The core element of ThingSpeak is a ‘ThingSpeak Channel’. A channel stores the data that we send to ThingSpeak and comprises of the below elements:

* 8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.
* 3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
* 1 status field - A short message to describe the data stored in the channel.

To use ThingSpeak, we need to signup and create a channel. Once we have a channel, we can send the data, allow ThingSpeak to process it and also retrieve the same. Let us start exploring ThingSpeak by signing up and setting up a channel.

**Node.js**

Node.js is an [open-source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform)[JavaScript](https://en.wikipedia.org/wiki/JavaScript)[run-time environment](https://en.wikipedia.org/wiki/Runtime_system) that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting)—running scripts server-side to produce [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) content before the page is sent to the user's web browser.

## **Why Node.js?**

## A common task for a web server can be to open a file on the server and return the content to the client.

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

**Source Code Explanation:**

For programming Arduino we have used following libraries:

* ESP8266WiFi.h
* WiFiClient.h
* ESP8266HTTPClient.h
* ThingSpeak.h

In order to connect with ESP8266 we have to mention SSID(Service Set Identifier) and password. Define trigger pin and echo pin numbers.

WiFiClient client:

Create client that can connect to specified port and IP address.

We have to mention channel Id and write API key provided by ThingSpeak. In order to operate ultrasonic sensor, make trigger pin as output and echo pin as input. For communicating with Serial Monitor, make sure to use one of the baud rates listed in the menu at the bottom right corner of its screen by using serial.begin( ). Configure ESP8266 in station mode and then connect it the wifi by wifi.begin( ). wifi.begin( ) returns WL\_CONNECTED when connection is established.

To start ranging supply a short 10uS pulse to the trigger input. Read the echoPin which returns the sound wave travel time in microseconds. Calculate distance by formula:

Distance = high level time \* velocity (340M/S) / 2

Declare object of class HTTPClient which provides methods to create and send the HTTP request. After that, we call the begin method on the http object and pass the URL that we want to connect to and make the GET request. The HTTPGETmethod requests a representation of the specified resource. Requests using GET should only retrieve data and returns only status code. Finally, we call the end method. This is very important to close the TCP connection and thus free the resources. To send the values to ThingSpeak we have to provide channel ID, field label and write API key to write menthod of ThingSpeak class.

**Express Server:**

Express is a web application framework for Node.js that allows you to spin up robust APIs and web servers in a much easier and cleaner way. It is a lightweight package that does not obscure the core Node.js features. In this post we’ll go over how to setup a very basic web server.

Let’s go over each section of this code to explain how Express works.

const express = require('express');

const app = express();

The first line here is grabbing the main Express module from the package you installed. This module is a function, which we then run on the second line to create our app variable. You can create multiple apps this way, each with their own requests and responses.

app.get('/', (request, response) => {

response.send('Welcome to Iot Application');

});

This bite of code is where we tell our Express server how to handle a GET request to our server. Express includes similar functions for POST, PUT, etc. using app.post(...), app.put(...), etc.

These functions take two main parameters. The first is the URL for this function to act upon. In this case, we are targeting '/', which is the root of our website: in this case, localhost:3005.

The second parameter is a function with two arguments: request, and response. req represents the request that was sent to the server; We can use this object to read data about what the client is requesting to do. res represents the response that we will send back to the client. Here, we are calling a function on res to send back a response: 'Welcome to Iot Application'.

app.listen(3000, () => console.log('Server started listening on port 3005'));

Finally, once we’ve set up our requests, we must start our server! We are passing 3005 into the listen function, which tells the app which port to listen on. The function passed-in as the second parameter is optional, and runs when the server starts up. This just gives us some feedback in the console to know that our application is running.

**HTTP Using REST:**

RESTful APIs enable you to develop any kind of web application having all possible CRUD (create, retrieve, update, delete) operations. REST guidelines suggest using a specific HTTP method on a specific type of call made to the server.

**What is REST?**

Representational State Transfer (REST) is a [software architectural](https://en.wikipedia.org/wiki/Software_architecture) style that defines a set of constraints to be used for creating [Web services](https://en.wikipedia.org/wiki/Web_service). Web services that conform to the REST architectural style, called RESTful Web services (RWS), provide interoperability between computer systems on the [Internet](https://en.wikipedia.org/wiki/Internet). RESTful Web services allow the requesting systems to access and manipulate textual representations of [Web resources](https://en.wikipedia.org/wiki/Web_resource) by using a uniform and predefined set of [stateless](https://en.wikipedia.org/wiki/Stateless_protocol) operations.

## HTTP GET

Use GET requests to retrieveresource representation/information only – and not to modify it in any way. As GET requests do not change the state of the resource, these are said to be safemethods. Additionally, GET APIs should be idempotent, which means that making multiple identical requests must produce the same result every time until another API (POST or PUT) has changed the state of the resource on the server.

For any given HTTP GET API, if the resource is found on the server then it must return HTTP response code 200 (OK) – along with response body which is usually either XML or JSON content (due to their platform independent nature).

In case resource is NOT found on server then it must return HTTP response code 404 (NOT FOUND). Similarly, if it is determined that GET request itself is not correctly formed then server will return HTTP response code 400(BAD REQUEST).

**Concept of ConnectDB:**

function connectDB () {

Const connection = mysql.createConnection({

host: 'localhost',

user: 'root',

password: 'root',

database: 'times',

port: 3306

});

ConnectDB function is used to establish the connection between mysql i.e. database and server. ConnectDB function consist of name of the database i.e. “times” , localhost, and port number of mysql.

**MySQL:**

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

# **MySQL Features**

* Relational Database Management System (RDBMS): MySQL is a relational database management system.
* Easy to use: MySQL is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.
* It is secure: MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.
* Client/ Server Architecture: MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.
* Free to download: MySQL is free to use and you can download it from MySQL official website.
* It is scalable: MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.
* Compatibale on many operating systems: MySQL is compatible to run on many operating systems, like Novell NetWare, Windows\* Linux\*, many varieties of UNIX\* (such as Sun\* Solaris\*, AIX, and DEC\* UNIX), OS/2, FreeBSD\*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).
* Allows roll-back: MySQL allows transactions to be rolled back, commit and crash recovery.
* High Performance: MySQL is faster, more reliable and cheaper because of its unique storage engine architecture.
* High Flexibility: MySQL supports a large number of embedded applications which makes MySQL very flexible.
* High Productivity: MySQL uses Triggers, Stored procedures and views which allows the developer to give a higher productivity.

# **MySQL Create Database**

CREATE DATABASE database\_name;

**MySQL Select Database**

SELECT Database is used in MySQL to select a particular database to work with. This query is used when multiple databases are available with MySQL Server.

USE database\_name;

# **MySQL CREATE TABLE**

The MySQL CREATE TABLE command is used to create a new table into the database. A table creation command requires three things:

* Name of the table
* Names of fields
* Definitions for each field

CREATE TABLE table\_name (column\_name column\_type...);

# **MySQL INSERT Statement**

MySQL INSERT statement is used to insert data in MySQL table within the database. We can insert single or multiple records using a single query in MySQL.

INSERT INTO table\_name ( field1,field2,...fieldN )

VALUES

( value1, value2,...valueN );

# **MySQL SELECT Statement**

The MySQL SELECT statement is used to fetch data from the one or more tables in MySQL. We can retrieve records of all fields or specified fields.

SELECT expressions FROM tables;

# **MySQL ORDER BY Clause**

The MYSQL ORDER BY Clause is used to sort the records in ascending or descending order.

SELECT expressions

FROM tables

ORDER BY expression [ ASC | DESC ];

**Android**

**What is Activity?**

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with [setContentView(View)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View)).

While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with [R.attr.windowIsFloating](https://developer.android.com/reference/android/R.attr.html#windowIsFloating) set), [Multi-Window mode](https://developer.android.com/guide/topics/ui/multi-window) or embedded into other windows.

There are two methods almost all subclasses of Activity will implement:

* [onCreate(Bundle)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) is where you initialize your activity. Most importantly, here you will usually call [setContentView(int)](https://developer.android.com/reference/android/app/Activity.html#setContentView(int)) with a layout resource defining your UI, and using [findViewById(int)](https://developer.android.com/reference/android/app/Activity.html#findViewById(int)) to retrieve the widgets in that UI that you need to interact with programmatically.
* [onPause()](https://developer.android.com/reference/android/app/Activity.html#onPause()) is where you deal with the user pausing active interaction with the activity. Any changes made by the user should at this point be committed (usually to the [ContentProvider](https://developer.android.com/reference/android/content/ContentProvider.html) holding the data). In this state the activity is still visible on screen.

**Activity Lifecycle**

Activities in the system are managed as [activity stacks](https://developer.android.com/guide/components/activities/tasks-and-back-stack). When a new activity is started, it is usually placed on the top of the current stack and becomes the running activity -- the previous activity always remains below it in the stack, and will not come to the foreground again until the new activity exits. There can be one or multiple activity stacks visible on screen.

An activity has essentially four states:

* If an activity is in the foreground of the screen (at the highest position of the topmost stack), it is active or running. This is usually the activity that the user is currently interacting with.
* If an activity has lost focus but is still presented to the user, it is visible. It is possible if a new non-full-sized or transparent activity has focus on top of your activity, another activity has higher position in multi-window mode, or the activity itself is not focusable in current windowing mode. Such activity is completely alive (it maintains all state and member information and remains attached to the window manager).
* If an activity is completely obscured by another activity, it is stopped or hidden. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere.
* The system can drop the activity from memory by either asking it to finish, or simply killing its process, making it destroyed. When it is displayed again to the user, it must be completely restarted and restored to its previous state.
* The following diagram shows the important state paths of an Activity. The square rectangles represent callback methods you can implement to perform operations when the Activity moves between states. The colored ovals are major states the Activity can be in.

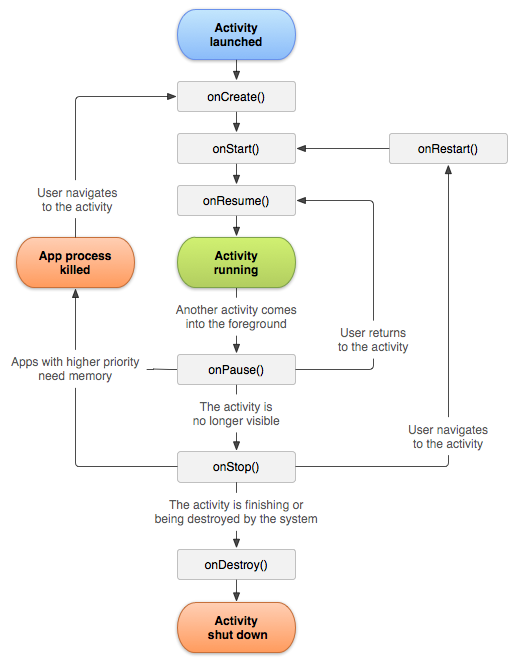


Fig 5. Activity Life Cycle

# **RecyclerView**

# A RecyclerView is : A View group for a scrollable container

# Ideal for long lists of similar items .Uses only a limited number of views that are re-used when they go off-screen. This saves memory and makes it faster to update list items as the user scrolls through data, because it is not necessary to create a new view for every item that appears.

# In general, the RecyclerView keeps as many item views as fit on the screen, plus a few extra at each end of the list to make sure that scrolling is fast and smooth.

# **Implementing a Recycler View**

# Implementing a RecyclerView requires the following steps:

# Add the RecyclerView dependency to the app's app/build.gradle file.

# Add the RecyclerView to the activity's layout

# Create a layout XML file for one item

# Extend RecyclerView.Adapter and implement onCrateViewHolder and onBindViewHolder methods.

# Extend RecyclerView.ViewHolder to create a view holder for your item layout.

# You can add click behavior by overriding the onClick method.

# In your activity, In the onCreate method, create a RecyclerView and initialize it with the adapter and a layout manager.

# **What is the use of an ion library in Android?**

The [ion library](https://github.com/koush/ion) by Koushik Dutta is an *Android Asynchronous Networking and Image Loading* library.

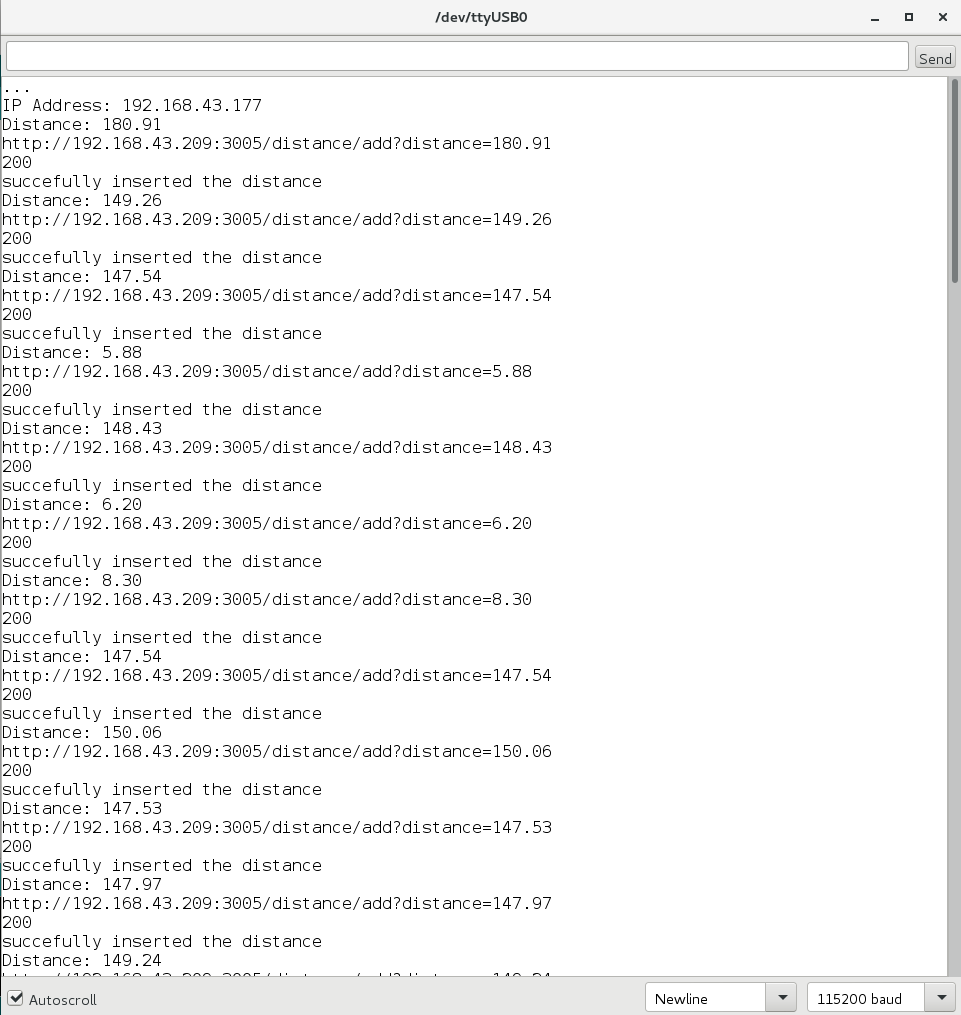
It uses ImageView or Bitmaps, JSON (via Gson), Strings, Files and Java types using Gson for asynchronous download. Apart from this:-

* It automatically cancels operations when the calling Activity finishes.
* Manages invocation back onto the UI thread.
* Uses the best/stablest connection from a server if it has multiple IP addresses.
* Supports file:/, http(s):/, and content:/ URIs

**TESTING**

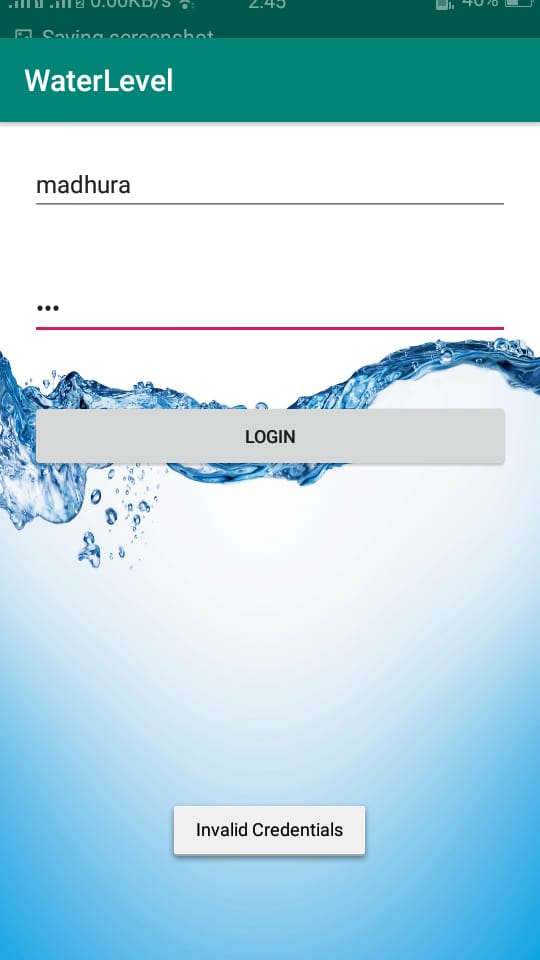
**Test Procedures:**

* 1. Check connectivity of ESP8266 with Wi-Fi whether the values are being inserted in database or not.



* 1. To test android applications login page

2.1 By entering incorrect password.



2.2 By entering correct username and password



**FUTURE SCOPE**

In future, the proposed system can be used to monitor and analyze water usage of the specific water source thus require developing such logic for the application. The system can also be used to collect and study the environmental data of water source and its surrounding area by integrating other sensor to the system. The study may include location data, water quality, temperature, humidity and various other factors. For example Arduino GPS shield can be integrated in the system to obtain location data of the water source dynamically.

* Power conservation.
* Automatic on/off switching operation.
* Wireless Communication.
* Can be implementing on any water source.
* Accuracy.
* Mobile Access.

**CONCLUSION**

We have been discussing about the elements of Smart phone and Water tank system. We actualized the discontinuous notice framework using the current tank system. There are a few preferences. Our connotation of this study is to become a line up to a flexible and competitively priced, simple configurable and most significantly, a transportable system which may solve our water wastage drawback. We have used ESP and Ultrasonic sensor which reduces cost effectively and makes this project economical. Also, this project doesn’t require special different tank for it, existing water tanks can be used. We have successfully implemented this project.

**REFERENCES**

1. DivyaKura,IOT based water tank control”[Article-Embedded for you Jan/Feb 2016.
2. N Vijaya Kumar, R Ramayas, “The real time monitoring of water quality in IOT environment”, IEEE sponsored 2nd international conference on innovations in information, embedded and communication systems (Iciiecs) 2015.
3. SaimaMaqbool ,Nidhi Chandra, “Real time wireless monitoring and control of water system using zigbee 802.15.14’ 5th international conference on computational intelligence and communication networks, 2013.
4. thinagaranperumall, md nasirsulaiman , leongintenet of things(IOT) enabled water system , IEEE 4th gobal conference on consumer electronic (GCCE), 2015.
5. Made Saraswathi ,EndrowednesKauntama, PonoMardjoko, Design and Construction of Water Level Management System Accessible Through SMS, IEEE Computer Society, 201299.
6. B. Dhivyapriya, C. Gulabsha, S. P. Maniprabha, G. kandaswamy, Dr. Chandhrasekharan, Gsm Based Water Tank Level Monitoring and Pump control system ,2016 Ijarmate.
7. ParchetVarma, Akshya Kumar, NiheshRathod,prathikjainmallikarjuns.renusubramaniyam, BhardhwajAmrutur,M,S.mohankumar , Rajesh Sundresan,IOT Based Water management system for a campus IEEE,IEEE First international Smart cities conference (isc2), 2015.
8. Asaad Ahmed mohammedahmedELTAIEB,ZhangJianmin”Automatic Water Level Control System”, international journal of science and research (IJSR) 2013.
9. SanamPudasiaini, Anujprathik, sukrtiDhakal, Milan Paudel ,”automatic Water Level Controller with Short Messaging Service (SMS)notification “,International journal of science and research publications, Volume 4,Issue 9, September 2014 1, ISSN2250-3153.
10. Jaytibhatt, jigneshpatoliya,IOT based water quality monitoring system, proceecings of 49th IRF international Conference , 21st February 2016 ,pune, india, ISBN: 978-93-85973-46-8.
11. In 2013, Raghavendhra.R,M.UttaraKumari, S.A. Hariprasad presented a paper on “implemenationof Simulated Water Level Cntroller”, International Journal of Advanced Reserch in Computer Science