**MINI PROJECT**

**(B.Tech. Second year 1st Sem Electrical Engg. Department)**

****

|  |  |  |
| --- | --- | --- |
| **SMART WATER LEVEL INDICATOR WITH AUTOMATIC PUMP SWITCHING ACTION** | | |
| **Group No** | **Sl.**  **No.** | **Name**  **(In alphabetical order) Roll No. Signature** |
| **22** | **1** | **Aishwarya Raj 510619114** |
| **2** | **Diptarka Roy 510619112** |
| **3** | **Moulindu Mandal 510619110** |

**Forwarded by Countersigned by**

**\_\_\_\_\_\_\_\_\_\_\_\_\_ ----------------------- Prof. D Ganguly HOD, E.E. Department (Convenor, DUGC)**

**Abstract**

The drinking water crisis in India is reaching alarming proportions especially in cities. Hence it is of extreme importance to preserve water by preventing wastage of water in overhead tanks of households due to inability of residents to monitor the water level of their overhead tanks. As a result, the water pumps keep running even when the overhead tank is overflowing. This amounts to huge waste of precious water and precious electricity. Our project offers a simple and effective solution to this serious problem by continuous monitoring the water level in the overhead tank and controlling the motor of the water pump by switching it on when the tank is empty and switching it off when the tank is full. We have made it a smart device with the use of IOT (Internet of Things) and enabling the users to monitor and control the water level of the overhead tanks of their houses from any location via smartphone or laptop**.**

**Acknowledgments**

We express our sincere gratitude and heartfelt thanks to our mentor Prof. **ABHIJIT CHATRABARTI** for developing the idea for this project and helping us progress along the way by his valuable input whenever and wherever necessary.

A special vote of thanks to **Prof. Prasid Syam,** our respected H.O.D for his incessant support, guidance and encouragement throughout the project.

Last but not the least, a word of thanks to our dear friends and esteemed seniors for their continuous cooperation. help and support.

**Table of contents**

Abstract…………………………………………………………………………………. Acknowledgements……………………………………………………………………..

Chapter 1: Introduction and Objectives………………………………………………… Chapter 2: System Design …………………………………………………………….. Chapter 3: Design Specifications and List of components used ……………………… Chapter 4: Implementation and Testing………………………………………………. Chapter 5: Comparison: Conventional WLI vs our IOT based WLI Chapter…………

6: Conclusion and Recommendation…………………………………………. Chapter 7: Appendixes ………………………………………………………………….

**Chapter-1**

**Introduction and Objectives**

Water is one of the most important substances on earth. All plants and animals must have water to survive. If there was no water there would be no life on earth. Apart from drinking it to survive, people have many other uses for water. Overhead water tanks are used for Domestic water storage and commercial water storage purposes. They are generally placed over the rooftop of any house, building or apartment. These tanks circulate the water through its distributary channels or pipes to the taps. Generally, most of the houses depend upon the overhead tanks as the main source of water. One of the commonly seen situations in every house is that the overflow from the tank made people switch off the pump. Otherwise, they should keep monitoring the tank when the pump is ON and switch off the pump as soon as the tank is full.

What happens when the client or user is not aware of water overflow or he is not at home while water is overflowing, as a result water resources are being wasted only because of improper management standards that we follow. Efficient usage techniques should be employed to gain better control on resource consumption. Water wastage is a serious issue that must be considered. Every drop of water counts when it comes to human survival on this planet as we only have a little amount of water available for us.as per the records only 3 percent of the water is available for the inhabitation the remaining 97 percent of the water is situated in the oceans.

We can’t use sea water by any means because it is salty. So, 3 percent of the water is present in the underground and rivers. Bore wells and pumps are used to extract water from the underground and stored in the tanks or sumps. Such an important and lifesaving resource is being wasted by us. We can develop an automatic system which can monitor on behalf of us. Currently there are many products which can solve this problem but the implementation and maintenance are very difficult. This model works by automatically controlling the water supply pump and communicating wirelessly to the client on his mobile or any device connected to the Internet.

In this model we are going to use a NODEMCU. It is just like any other microcontroller board, with a specifically designed API and software which makes programming it very easy. Embedded systems allow us to develop a standalone system which can solve certain problems. Embedded system has the capability to run automatically. It reduces the human involvement in the problem solving once it is assembled and deployed.

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, store, and also control the data in various

electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems are mainly involved in our real life for several devices like microwave, calculators, TV remote control, home security and neighbourhood traffic control systems. Embedded systems alone will have limitless opportunities when it comes to solving real world problems. What happens when we provide an internet community to every system and every device that is called an IoT. Internet of things facilitates internet connectivity to every device making the device accessible from remote areas or from the internet. Connectivity allows sharing of large data with the server where the decision-making system resides. All the nodes can be controlled by the master and everything else works as a slave in the network. IOT devices are a part of the larger concept of home automation, also known as domoticz. Large smart home systems utilize a main hub or controller to provide users with a central control for all of their devices. These devices can include lighting, heating and air conditioning, media and security systems.

**Problem Statement:**

Most of the time People generally switch on the motor when their taps go dry and switch off the motor when the tank starts overflowing. This results in unnecessary wastage of water and sometimes non-availability of water in an emergency. Sometimes people forget to switch OFF the pump by involving in their day-to-day activities. This results in wastage of both water and power resources. This is the serious problem that must be considered because the global scale of power and water resource wastage will be high.

**Objectives:**

The main intention of this project is to control the water management activity automatic by using float sensors which can detect the water level and controllers which can further turn ON or OFF the pump based on the water level.

● The system which decreases human intervention in maintaining the overhead tanks which is capable of effective water usage.

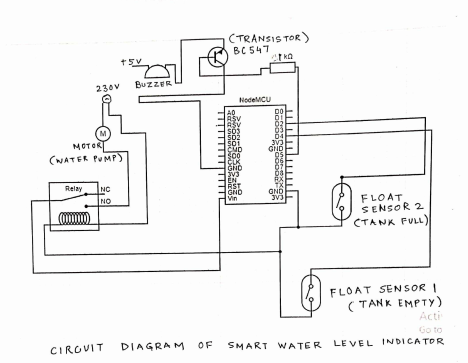
● Autonomous system which guarantees overflow prevention and optimal power usage.

● To deploy a system which when connected with the Internet, can provide the water level of the overhead tank and the status (On/Off) of the water pump to the client on his Smartphone.

**Chapter-2**

**System Design:**

Automatic water monitoring system solves the problem of water overflow by installing two float sensors in the overhead water tank one at the lowest level (about 10 % of tank capacity) to operate when tank is empty and other on the highest level of the tank to operate when tank is full (about 90% of tank capacity). These float sensors interface directly with the NodeMCU and send signals to NodeMCU according to water level. NodeMCU is programmed to send signal and energise the relay to start and stop the water pump. NodeMCU is also connected to a Buzzer through a transistor which produces sound when the tank is full and also when the tank is empty. NodeMCU can be suitably programmed and easily connected to WiFi to provide the data to the user on internet. The circuit diagram of the system is given below:



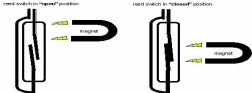
**Chapter-3**

**List of components used:**

Following are the hardware components used in the system:

**1-Float Sensors (REED-TYPE NO SWITCH): An Innovative Solution to Naked Probe**

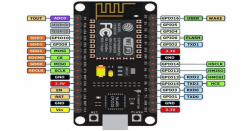
We are using float sensors for water sensing to do away with inserting naked electrical probes into tanks or water bodies which often have adverse side effects.



Float Sensors are often used in Water sensing. It’s a device that opens or closes the circuit depending on the liquid level of tank. We are using a reed switch type float switch. It is basically an electromagnetic switch. A magnet moves up and down the stem body of the switch depending upon the water level. When the magnet passes over the reed switch it exerts an electromagnetic force on it closing on opening the switch as required.

2-**NodeMCU**

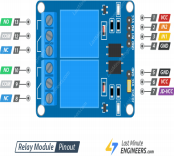
1-At first, we were geared towards using Arduino microcontroller for our project due to its overwhelming popularity. However, a bit of looking into soon revealed that it was better to use NODE MCU as it was targeted towards small scale IOT projects and performed better than Arduino in terms of WiFi Connectivity. Node MCU is a firmware and development board based on the TENSILICA 32-bit RISC CPU Xtensa LX106 microcontroller. It operates on a minimal 3.3V but can take 7-12V input. The pin diagram (16 digital, 1 Analog) is shown. For our project which heavily relies on IOT mechanisms this board has been a boom.



****

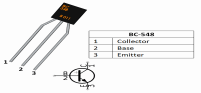
**3--DC Relay Module:**

We have used a DC relay in our model as a Relay Module. An AC relay switches based upon impedance but a DC relay does so based on resistance. The NodeMCU circuit provides DC signals to a relay to act as a circuit breaker for a much higher 220V AC circuit. The Jd-Vcc pin is simply used to provide a separate source to power the electromagnet of the relay in order to reduce the load on NodeMCU which is not used to output large currents or voltages. There is no risk of back emf from a DC relay.

**4- Transistor (BC 548):**

Transistor (BC 548) has been used to drive a Buzzer with 1k resistor at its base.



**5-Buzzer:**

A Piezo-electric buzzer has been used in the circuit to act as an alarm both when the tank is empty and when tank is full.

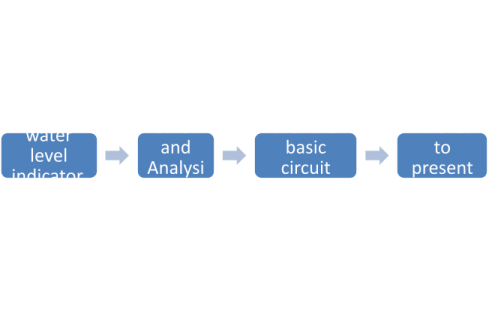


**6-Resistor:**

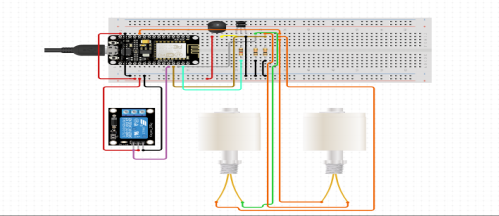
Resistor is used at the base of the transistor to protect transistor and NodeMCU. 

. **Chapter-4**

**Implementation and Testing**

The project was conceived, designed, implemented and tested in the following manner:

The system was implemented by connecting the components according to the circuit diagram given previously. The schematic diagram is given below: This has been made using the Web App of circuitio.io and has not been plagiarized

**Working of system:**

We installed the Arduino IDE and set it up using the following tutorial on [Instructables](https://www.instructables.com/Steps-to-Setup-Arduino-IDE-for-NODEMCU-ESP8266-WiF/) website.

**Explanation of the Code:**

## In this project we would be using the Wi-Fi functionality of the ESP8266 NodeMCU, for which we need to include the ESP8266WiFi library.

## 

## The official [documentation](https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html) states - “The [Wi-Fi library for ESP8266](https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi) has been developed based on [ESP8266 SDK](https://github.com/espressif/ESP8266_NONOS_SDK), using the naming conventions and overall functionality philosophy of the [Arduino Wi-Fi library](https://www.arduino.cc/en/Reference/WiFi). Over time, the wealth of Wi-Fi features ported from ESP8266 SDK to [esp8266 / Arduino](https://github.com/esp8266/Arduino) outgrew [Arduino Wi-Fi library](https://www.arduino.cc/en/Reference/WiFi) and it became apparent that we would need to provide separate documentation on what is new and extra.”

## The next two variables are for connecting the NodeMCU Wi-Fi module to the home Wi-Fi router and contain the ssid and password of the router.

## 

## We then create a Wi-Fi server.

## 

## The [official website](https://www.arduino.cc/en/Reference/WiFiServer) says that the function – “Creates a server that listens for incoming connections on the specified port.” The syntax is “server(port);”. In our case the port is 80.

## Below code declares two functions that if called will turn ON/OFF the motor by switching the relay from HIGH to LOW states as well as changing the motorStatus so that the website displays the appropriate state of the motor (i.e., whether it is switched ON or OFF).

## 

## Then we start setting up the NodeMCU using the “setup()” function. We start off by setting up code for serial communication with the computer. The output can be seen in the Serial Monitor within the Arduino IDE by pressing ctrl+shift+M.

## 

## “Serial.begin()” function starts the serial communication between the computer and the microcontroller. “Serial.println()” function prints the string given to it as argument with a new line. The “delay()” function is used for implementing a delay between the execution of the previous line of code and the next line of code.

## We then move on to setting up the Wi-Fi connection using the “WiFi.begin()” function with ssid and password initialised earlier as the arguments.

## 

## We then print sequential numbers per second showing that the microcontroller is attempting to connect to the Wi-Fi router.

## 

## After successful connection we print that the connection is established and we print out the local IP address.

## 

## “WiFi.localIP()” function returns the local IP address which is displayed to the Se/rial Monitor.

## Then we start the server and print that “Server started”.

## 

## The “server.begin()” is used to start the server. And we display that the server has started.

## Finally, we pinmodes using the “pinMode()” function which takes in two arguments of which the first is the integer value of the pin whose mode is being declared and the second is its mode. The mode for the input pins (i.e., the Float Sensors) is set as INPUT\_PULLUP and that if the output pin is set as OUTPUT.

## Also, we call the motorOFF() function, so that the motor remains switched OFF when the microcontroller sets itself up. Finally, we set a delay before starting the main program contained in the “loop()” function. Here we notice that both “loop()” and “setup()” functions return nothing and hence have return type “void”.

## Next, we have the “loop()” function which runs the code contained inside its scope on a continuous loop. As mentioned before its return type is “void”.

## Inside the “loop()” function we first digitally read the float sensor values. The corresponding values read by upper sensor (“FloatSensorHigh”) and the lower sensor (“FloatSensorLow”) are assigned to upperLimit and lowerLimit respectively.

## 

## Next, we create two Boolean variables “tooHigh” and “too Low”. The variable “tooHigh” is true when both the float sensors are floating (i.e., when they are LOW). The other variable “tooLow” is true when both the float sensors are not floating (i.e., when they are HIGH).

## If the water level is “tooHigh” then we turn off the motor. And If the water level is “tooLow” then we turn on the motor.

## 

## Next, we see if a client is connected to the server or not. If client is not connected then we return the “loop()” function and the rest of the code does not run. However, if the client is connected the next lines run where in we initialise a String variable “request” which parses the request made by the client and assigns it as a string.

## “Server.available()” function checks if a client is connected to the server, which is assigned to the “client” variable. The “client.flush()” function discards any bytes that have been written to the client but not yet read.

## We read the request and turn on the motor if the client has clicked on turn ON the motor and turn it off if the client has clicked on the turn OFF the motor followed by a delay.

## 

## Finally, we prepare the response string and send it to the client. We use the “client.print()” function to send it to the client and print it on the display.

## The HTML, CSS and JavaScript code sent to the client is saved in the remote GitHub repository in the [index.html](https://github.com/DiptarkaRoy/NodeMCU_MiniProject/blob/main/index.html) file.

## For converting the HTML file containing CSS and JavaScript to C string, we used the [text to C/C++ string converter](https://tomeko.net/online_tools/cpp_text_escape.php?lang=en). This website made our work a lot easier.

**Chapter-5**

**Comparison: Conventional WLI vs our IOT based WLI**

Conventional water level indicator and controller currently available or based on old design and in our model, we have improved upon it by incorporating latest technology and making it more accurate rugged robust lo power consuming and more importantly very user friendly. Following the major advantages of our small water level indicator over conventional water level indicator are listed below.

|  |
| --- |
| **Conventional Water Level Indicator (WLI) Our Smart Water Level Indicator** |
| 1- This system requires naked Probes to be  1-Our system uses float sensors which can  inserted into the water tank which get  handle thousands of opening and closing  corroded due to dissolves minerals in water  iterations without getting corroded. Thus it  requires very low maintenance. |
| 2-The design of the system involves use of  2-This system uses NODEMCU  discrete component like Transistors and LEDs  (microcontroller) which can be customized and  etc which have no programming facility and  programmed according to the requirement of  internet connectivity.  the user and data can be viewed remotely by  the user. |
| 3- This system sounds buzzer when the tank is  3- No manual intervention required as the  full and requires manual intervention for  switching process is fully automated.  switching on/Off the water pump. |

4-This system has very limited scope of improvement.

4- This system is very flexible and with slight modification it can be used in industries as water/liquid level sensor in boilers, containers etc. as fuel level indicator in vehicles and

|  |
| --- |
| measuring water level in areas having heavy  rains and floods. |

## Appendix:

## References:

## <https://tomeko.net/online_tools/cpp_text_escape.php?lang=en>

## <https://www.arduino.cc/en/Reference/WiFiServer>

## <https://www.arduino.cc/en/software>

## <https://www.electronicwings.com/nodemcu/getting-started-with-nodemcu-using-esplorer>

## <https://www.instructables.com/How-to-Program-NodeMCU-on-Arduino-IDE/>

## <https://techtutorialsx.com/2016/10/15/esp8266-http-server-serving-html-javascript-and-css/>

## <https://diyprojects.io/bootstrap-create-beautiful-web-interface-projects-esp8266/#.YJ5Mv6gzZPZ>