**Automatic Water level indicator & control**

**Table of contents**

1. About Automatic water level indicator & control
   * [1.1 Abrstact](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#11-abstract)
   * [1.2 Introduction](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#12-introduction)
   * [1.3 working of the system](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#13-working-of-the-system)
   * [1.4 SWOT Analysis](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#14-swot-analysis)
   * [1.5 4W & 1H](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#15-4w--1h)
2. Requirements
   * [2.1 HIGH Level requirements](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#21-high-level-requirements)
   * [2.2 LOW Level requirements](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#22-low-level-requirements)
3. Flow chart, Block diagram and Explanation
   * [3.1 Flow chart](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#31-flow-chart)
   * [3.2 Block Diagram](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#32-block-diagram)
   * [3.3 components used](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#33-componenets-used)
   * [3.4 Explanation of components](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#34-explanation-of-components)
   * [3.5 Schematic Diagram](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#35-schematic-diagram)
4. Test plans
   * [4.1 High level test plan](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#41-high-level-test-plans)
   * [4.2 Low lwvwl test plan](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#42-low-level-test-plans)
5. [Advantages](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#5-advantages)
6. [Applications](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#6-applications)

[Reference](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#reference)

[Code](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#code)

[OUTPUT](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#output)

* **1. About Automatic water level indicator & control**
  1. **Abstract**

Most of the people in reidental areas face the problem of running out of water and overflow of water in water tank due to excess supply of water. It becomes difficult for users to judge the level of water in water tanks. When the pump is turned ON, users will not realise that the water tank is filled, Which may result in overflow. Waater level indicator and controller system is used to sort out the issues associated with water tank. It is also possible to check the level of the water using sensor so that whenever the water goes below, Pump gets turned ON automatically. Also when there is overflow of water in water tank it uses sensor to detect the water level so that if the water level goes above, the pump gets turned OFF utomatically. This system prevents wastage of water.

**1.2 Introduction**

* Water is the important natural resource that should be efficiently used. Our project Automatic Water Level indicator & Controller uses the Arduino and Ultrasonic senser. The Basic idea of the project is to ensure the proper use of water and reduce the wastage of water.
* This basic idea can be implemented in many fields and purposes such as irrigation in agriculture land, water pump controlling water usage monitoring etc
* Automatic Water level indicator and controller, the name itself indicates that an electronic device or circuit kit used for controlling the water level can be termed as a Automatic water level indicator & controller.
* It is difficult to know the level of water in the overhead tank such that wastage of water can happen frequently. \*To conserve water, avoid overflow of water in the overhead tank which may cause loss of water, loss of electrical power, etc.,. \*Thus, an ultrasonic water level indicator & controller using Arduino is an innovative electronics project application for controlling water level.
* Automatic Water Level indicator Controller using the Arduino project will help in automatically controlling the water motor by sensing the water level in a tank.
* This article explains how to detect and control the water level in an overhead tank or any other container.
* This system monitors the water level of the tank and automatically switches ON the motor whenever the tank is empty.

**1.3 Working of the System**

\* In this Arduino based automatic water level indicator and controller project we are going to measure the water level by using ultrasonic sensors.

\* Basic principal of ultrasonic distance measurement is based on ECHO.

\* When sound waves are transmitted in environment then they return back to the origin as ECHO after striking on any obstacle.

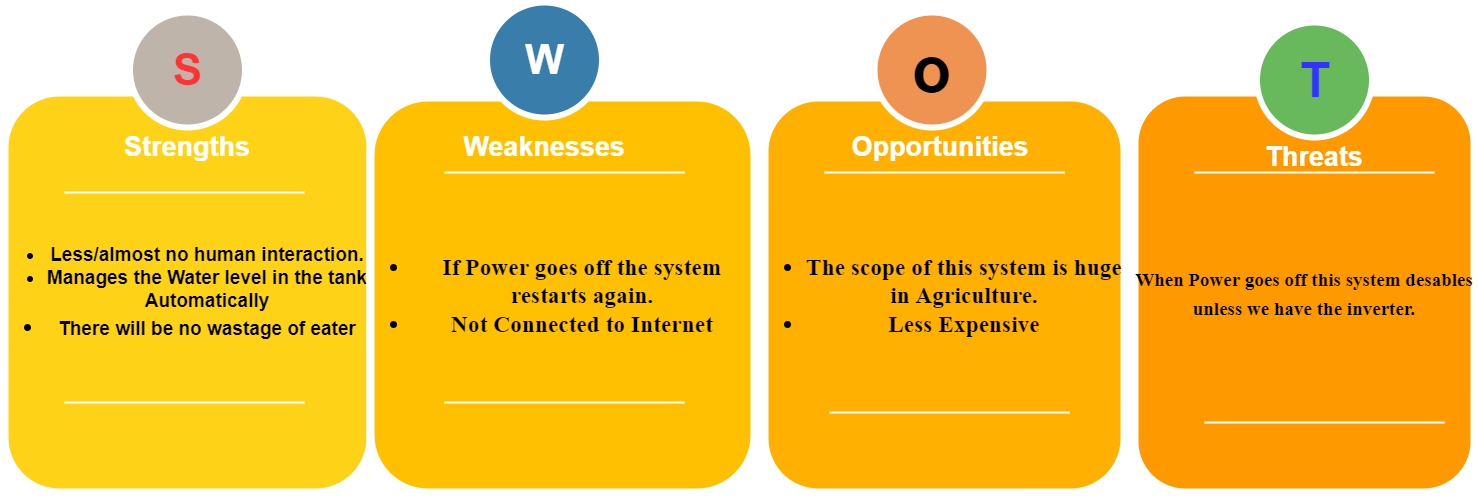
\* So we have to only calculate its traveling time of both sounds means outgoing time and returning time to origin after striking on any obstacle.

\* And after some calculation we can get a result that is the distance.

\* This concept is used in our water controller project where the water motor pump is automatically turned on when water level in the tank becomes low.

[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**1.4 SWOT Analysis**

[](https://user-images.githubusercontent.com/98537406/155819233-2f1a0379-3d83-4d67-87e4-a3b72dd8a1ac.png)

[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**1.5 4W & 1H**

* What:
  + Automatic water level indicator & controller is an module that was created to automatically control a motor, which helps to ensure a constant reserve of water in a storage tank.
  + These Automatic water level indicator & controllers are used to automatically fill the over-head tank when it starts or has become empty as well as monitor the water level in it.
* Why:
  + A water level controller helps save money by limiting the waste of water and electricity.
  + These devices accurately regulate how much energy is used to protect against any unnecessary water/electricity usage.
* Where:
  + It can be installed in homes, hotels, and hospitals.The power used by this controller is low and negligible.
  + It can be installed in metro cities where the drinking water is the only water used for all purposes, which keeps the drinking water from being wasted.
* When: \*Automatic water level indicator & controllers switch the motor on whenever the water level drops below a certain level and shuts the motor off when the water rises well above a fixed level.
* How:
  + Implementing the simulation by using SimulIDE software and Arduino IDE sourse Code.

[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**2. Requirements**

**2.1 HIGH Level requirements**

| **RID** | **Discreption** |
| --- | --- |
| HLR1 | Water level Indicator & controller shall be automated. |
| HLR2 | System shall show the status of the water level in the tank. |
| HLR3 | System shall switch ON the motor |
| HLR3 | System shall switch OFF the motor |

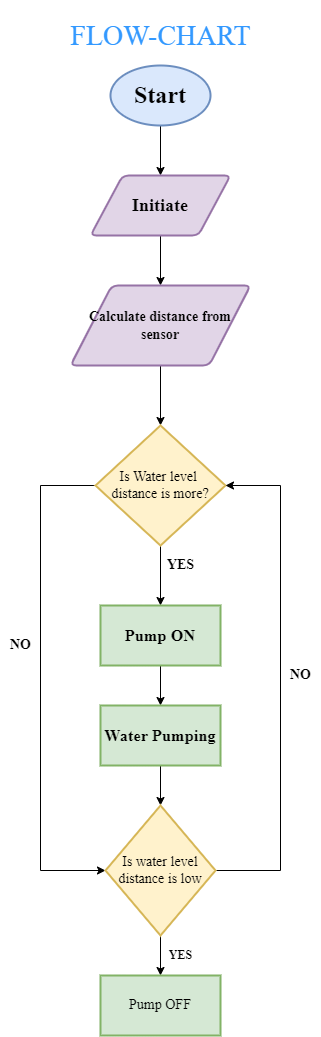
**2.2 LOW Level requirements**

| **RID** | **Discreption** |
| --- | --- |
| LLR1 | Ultrasonic sensor shall calculate the Level of the water in overhead tank |
| LLR2 | LCD shall display the Water level and Motor status. |
| LLR3 | Motor shall be switched ON, on the following condition : \* When the water level of the overhead tank is lower than the lowest sense level. |
| LLR4 | Motor shall be switched OFF, on the following condition: \* When the water level of the overhead tank reaches the highest sense level. |

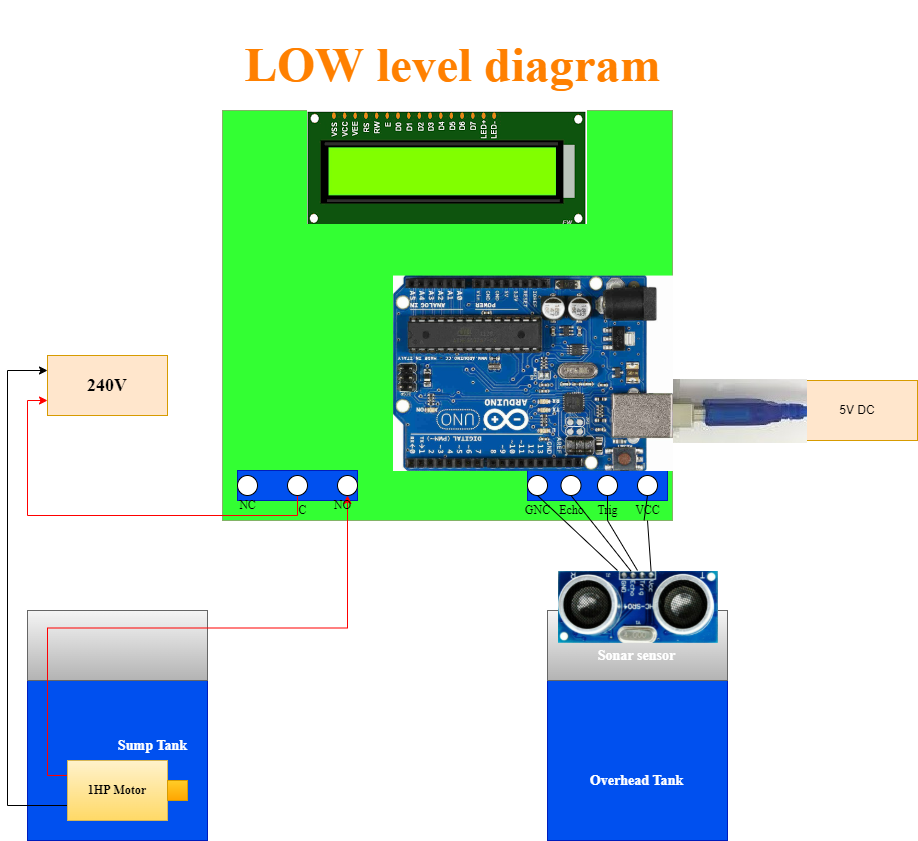
[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**3. Flow chart, Block diagram and Explanation**

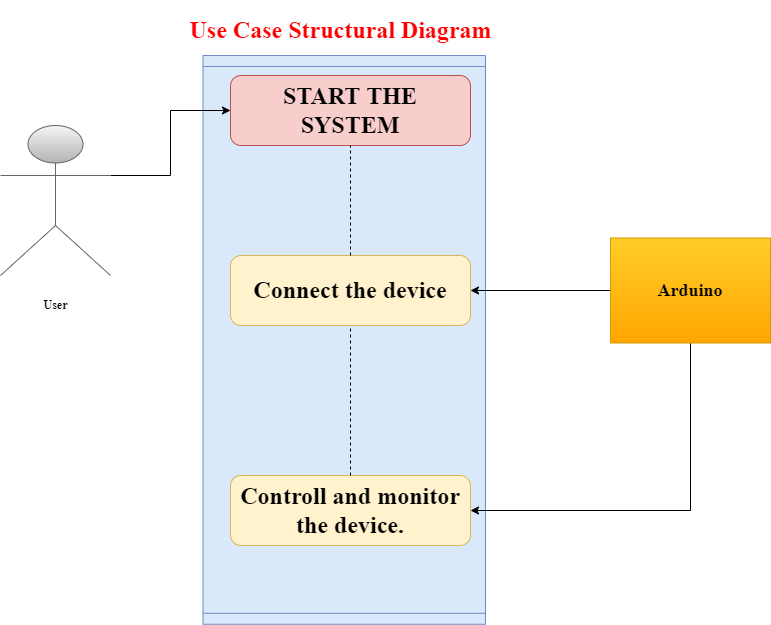
**3.1 Flow chart**

[](https://user-images.githubusercontent.com/98537406/155820422-ae0ec75d-fe6d-4d64-939a-be945e17d177.png)

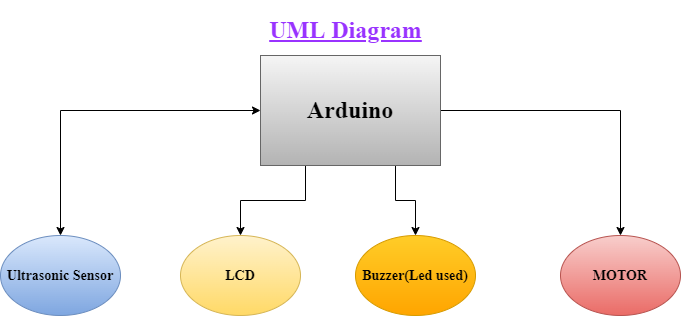
**3.1.1LOW level deiagram**

[](https://user-images.githubusercontent.com/98537406/156926009-f85d1b68-255f-4679-a4b5-cab9dc579c87.png)

**3.1.2 USECASE Structural Diagram**

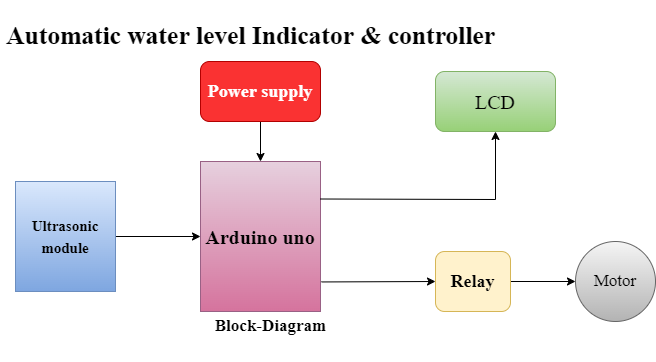
[](https://user-images.githubusercontent.com/98537406/156926329-d2a974bb-051c-434a-8e34-ba4d04b272ec.png)

**3.1.3 UML Diagram**

[](https://user-images.githubusercontent.com/98537406/156926432-3eb515a2-3a13-49b9-be86-5f8d07c8b191.png)

[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**3.2 Block Diagram**

[](https://user-images.githubusercontent.com/98537406/155879177-b5b8ba02-53be-4e79-a60d-701e2c4c67af.png)

**3.3 Componenets used**

* Arduino.
* Ultrasonic Sensor.
* LCD.
* LED.
* Relay.
* DC motor.

[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**Software used**

- Simulation (`SimulIDE`)

- Code (`Arduino IDE`)

**3.4 Explanation of components**

1. Arduino:

[](https://user-images.githubusercontent.com/98537406/155877578-41ca9a18-7c54-4932-9705-28ec1ab634e3.png)

* Arduino UNO has the micro-controller ATmega328 embedded in it.
* It has 14 digital I/O pins out of which 6 provide PWR output.
* It is an open-source and provides prototype platform.
* It also has a 16MHX crystal oscillator attach to it.
* Inaddition to above features, it also has an USB connection, a power jack, an ICSP, header and reset button.
* It can simply be connect to a computer using an USB cable or power it with an AC or a DC adapter or a bettery.

1. Ultrasonic Sensor:

[](https://user-images.githubusercontent.com/98537406/155877878-0c855093-d640-40ef-8022-4f2e6245cd0e.png)

- It is basically a distance sensor and is used for detecting the distance.

- It has two ultrasonic transmitters namely the receiver and the control circuit.

- The transmitter emits a high frequency ultrasonic sound wave which bounces off from any solid object and receiver receives it as an echo.

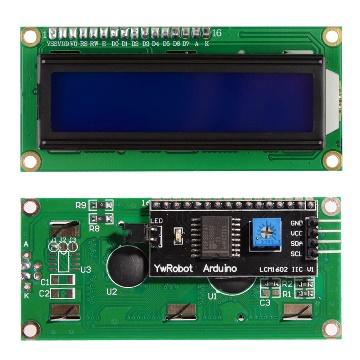
- The echo is then processed by the difference between the transmitter and receiver signal.

- This time can subsequently be used to measure the distance between the sensor and the reflecting object.

- It has an ultrasonic frequency of 40 KHz and acurracy is near to 0.3 cm.

| **Discreption** | **Range** |
| --- | --- |
| Operating Voltage | 5V |
| Operating Current | 15mA |
| Operating frequency | 40KHz |
| Measuring angle | 15 Degree |
| Ranging accuracy | 3mm |
| Trigger input signal | 10 MicroSeconds |
| Device dimension | 45x20x15mm |
|  |  |

1. LCD:

[](https://user-images.githubusercontent.com/98537406/155878092-81772001-d662-4d11-9b78-c0d8645b9e89.png)

* + LCD csreen is a flat-panel display, electronic visual display that uses the light-modulating properties of liquid crystal and LCD's are available to display arbitrary images, Preset words, digits and 7-segment dispalys.

1. LED:

[](https://user-images.githubusercontent.com/98537406/155878183-74f5e449-fc88-4ff9-b24b-449a4b04d48f.png)

* + LED is a light-emitting diode.
  + It is a semi-conductor diode that emits light when conducting current.

1. Relay:

[](https://user-images.githubusercontent.com/98537406/155878245-7bd76c56-5154-4df1-b3e3-4d0529aed879.png)

* + In order to isolate two circuit electrically and to connect them magnetically relays are used.
  + They are very useful in switching from one circuit to another when they are completely seperated.
  + The relays comprise of an input and an output section.
  + The input section has a coil which produces magnetic field when a small voltage from an electrical circuit is applied.
  + This applied voltage is known as the operating voltage.

1. DC motor:

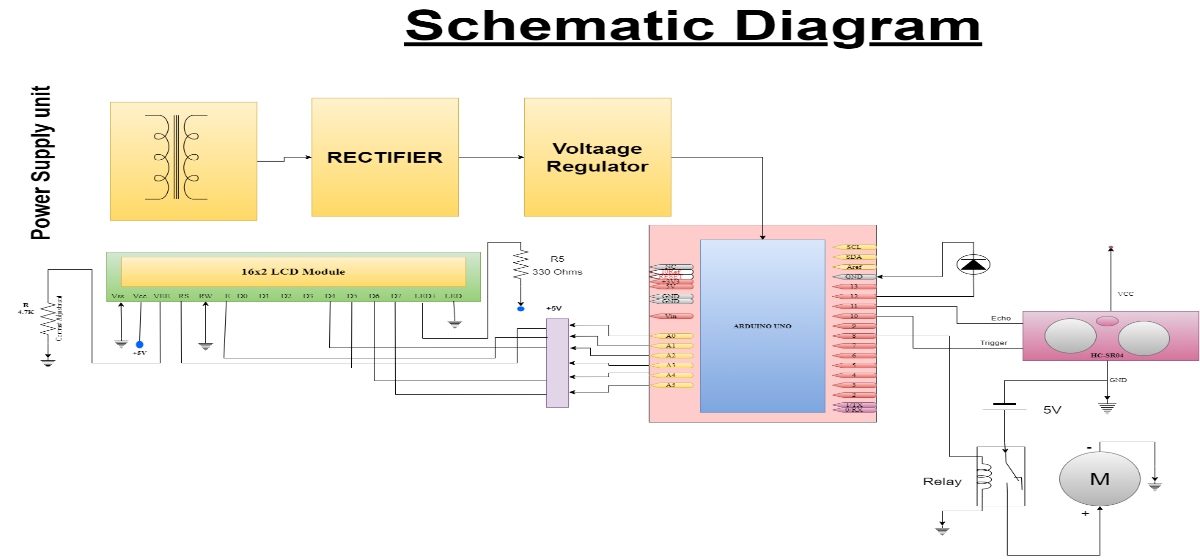
[](https://user-images.githubusercontent.com/98537406/155878364-4bb38804-a7dc-48d6-892c-9b04c99f39bf.png)

- A DC motor is a class of electrical motors that converts direct current electrical energy into mechanical energy.

- DC motors use magnetic fields that occur from the electrical currents generated, which powers the movement of a rotar fixed within the output shaft.

- The output torque and speed depends upon both the electrical input and the design of the motor.

**3.5 Schematic Diagram**

[](https://user-images.githubusercontent.com/98537406/157217159-1c7749eb-78c2-4034-8256-93a75405ff02.png)

[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**4. Test Plans**

**4.1 High level test plans**

| **RID** | **Discreption** | **Expected O/p** | **Actual O/p** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| HLT1 | Ultra sonic sensor | Calculate the Distance | Calculate the distance | Pass |
| HLT2 | Buzzer(Led Used) When water Tank is Full | It will make sound(Led will glow) | It will make sound(Led will glow) | Pass |
| HLT3 | When water Tank is Full | Motor gets turned OFF | Motor gets turned OFF | Pass |
| HLT4 | Water Level is Low | Motor gets turned ON | Motor gets turned ON | Pass |
| HLT5 | Water Level is Low | It will remain silent(Led will not glow) | It will remain silent(Led will not glow) | Pass |
| HLT6 | LCD | Displays the Distance and Water level | Displays the Distance and Water level | Pass |

**4.2 Low level test plans**

| **RID** | **Discreption** | **Expected O/p** | **Actual O/p** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| LLT1 | Ultra sonic sensor Calculate the Distance Example 14.23cm | Water tank Full & Motor turned off | Water tank Full & Motor turned off | Pass |
| LLT2 | Ultra sonic sensor Calculate the Distance Example 374.32cm | LOW Water level & Motor turned on | LOW Water level & Motor turned on | Pass |

[**Back to Contents**](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**5. Advantages**

* Easy installation.
* Low maintenance.
* Compact and elegant design.
* Users can control the required level of water in over head tank.
* Avoids wastage of water from tanks.
* It can maintain exact preset water levels.
* Being automatic saves man power.

**6. Applications**

* Used in buildings where the manual monitoring is difficult.
* Used in industries to control the water level and in chemical mixing etc.
* It can be installed in metro cities where the drinking water is the only water used for all purposes which keeps the drinking water from being wasted.
* Fuel tank level gauging.
* Oil tank level control.
* Pool water level control.

**Reference**

* [Study1](https://www.thehindu.com/news/cities/Delhi/40-per-cent-of-water-supply-gets-wasted-Study/article16836247.ece)
* [Study2](https://www.ijraset.com/fileserve.php?FID=14330)
* [Study3](https://www.academia.edu/34342283/AUTOMATIC_WATER_LEVEL_CONTROL)

**Neaws on wastage of water**

[over-head-tank-water-wastage](https://www.deccanherald.com/city/bwssb-targets-overhead-tank-water-wastage-in-new-move-773145.html)

[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**Code**

To program Arduino for Automatic Water level indicator and controller, first we define all the pin that we are going to use in the project for interfacing external devices like Relay , LCD , Buzzer etc.

#include<LiquidCrystal.h>

#define trigger 10

#define echo 11

#define motor 8

#define buzzer 12

LiquidCrystal LCD(A0,A1,A2,A3,A4,A5);

float time=0,distance=0;

int temp=0;

Then we initialize all the devices that we used in project in void setup{}

// put your setup code here, to run once:

pinMode(trigger,OUTPUT);

pinMode(echo,INPUT);

pinMode(motor,OUTPUT);

pinMode(buzzer,OUTPUT);

LCD.begin(16, 2);

LCD.setCursor (0,0);

LCD.print("Water level");

LCD.setCursor (0,1);

LCD.print("Indicator");

delay(2000);

LCD.clear();

Now initialize the ultrasonic sensor module and read time of sending and receving time of ultrasonic waves or sound by using pulseIn(pin). Then perform calculations and display the result on 16x2 LCD by using appropriate function. In voide loop{}

digitalWrite(trigger,LOW);

delayMicroseconds(2);

digitalWrite(trigger,HIGH);

delayMicroseconds(10);

digitalWrite(trigger,LOW);

delayMicroseconds(2);

time=pulseIn(echo,HIGH);

distance=time\*340/20000;

LCD.setCursor (0,0);

LCD.print("Water Space in");

LCD.setCursor (0,1);

LCD.print(" Tank is: ");

LCD.print(distance);

LCD.print("Cm");

delay(1000);

LCD.clear();

After it we check conditions if water tank is full or water level is LOW, and take action accordingly

if(distance<40 && temp==0)

{

digitalWrite(motor,LOW);

digitalWrite(buzzer,HIGH);

LCD.setCursor (0,0);

LCD.print("Water Tank Full");

LCD.setCursor (0,1);

LCD.print("Motor Turned OFF");

LCD.print(millis());

delay(3000);

digitalWrite(buzzer,LOW);

delay(1000);

LCD.clear();

temp=1;

}

else if(distance<40 && temp==1)

{

digitalWrite(motor,LOW);

LCD.setCursor(0,0);

LCD.print("Water Tank Full");

LCD.setCursor(0,1);

LCD.print("Motor Turned OFF");

delay(1000);

LCD.clear();

}

else if(distance>100)

{

digitalWrite(motor,HIGH);

LCD.setCursor(0,0);

LCD.print("LOW Water Level");

LCD.setCursor(0,1);

LCD.print("Motor Turned ON");

delay(1000);

temp=0;

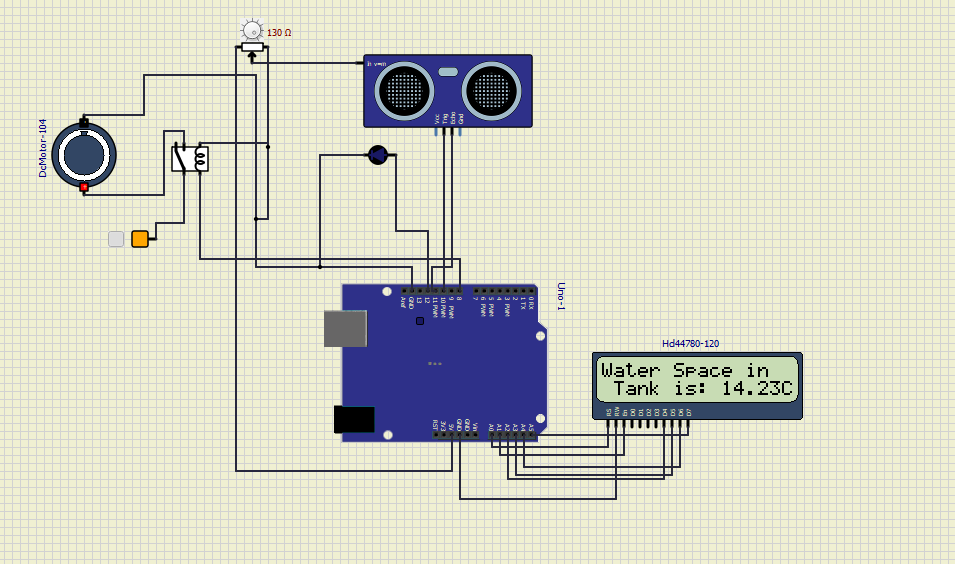
LCD.clear();

}

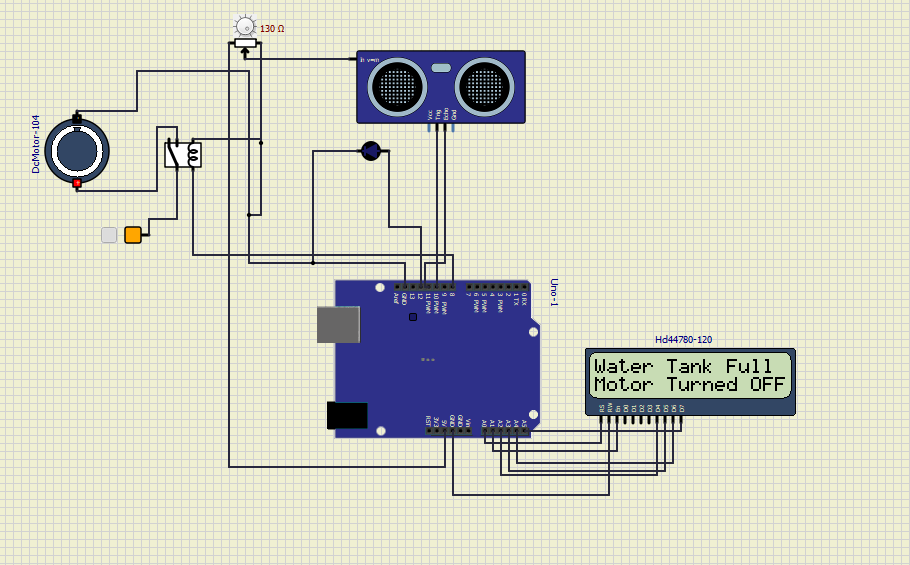
[Back to Contents](https://github.com/Reddy426/M2_EmbSys/tree/Master/PROJECT/6_Report#table-of-contents)

**OUTPUT**

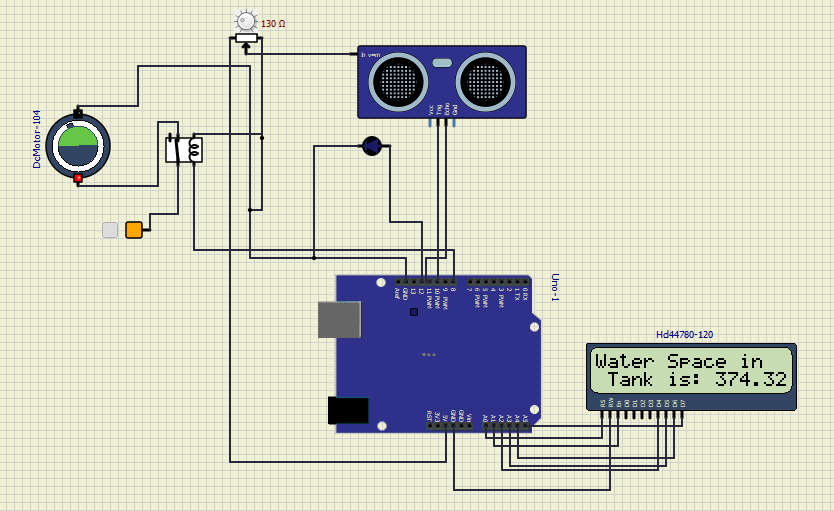
* Distance < 40

[](https://user-images.githubusercontent.com/98537406/156934666-b29cc5ae-83f4-431f-a4db-11f2870d607e.PNG)

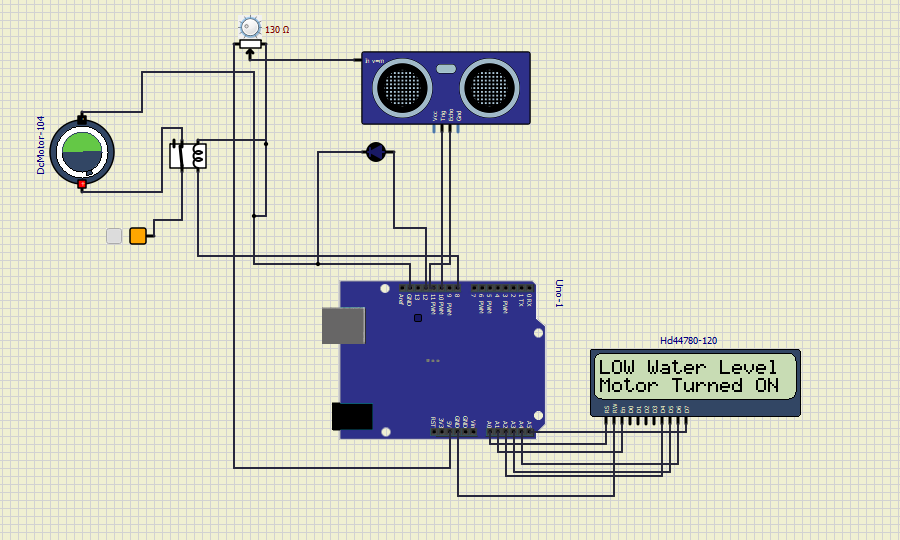
* Water level is HIGH (Full) & Motor is OFF at distance < 40

[](https://user-images.githubusercontent.com/98537406/156934737-39cc898c-54b3-46ad-9e4a-f3759ca02f8c.PNG)

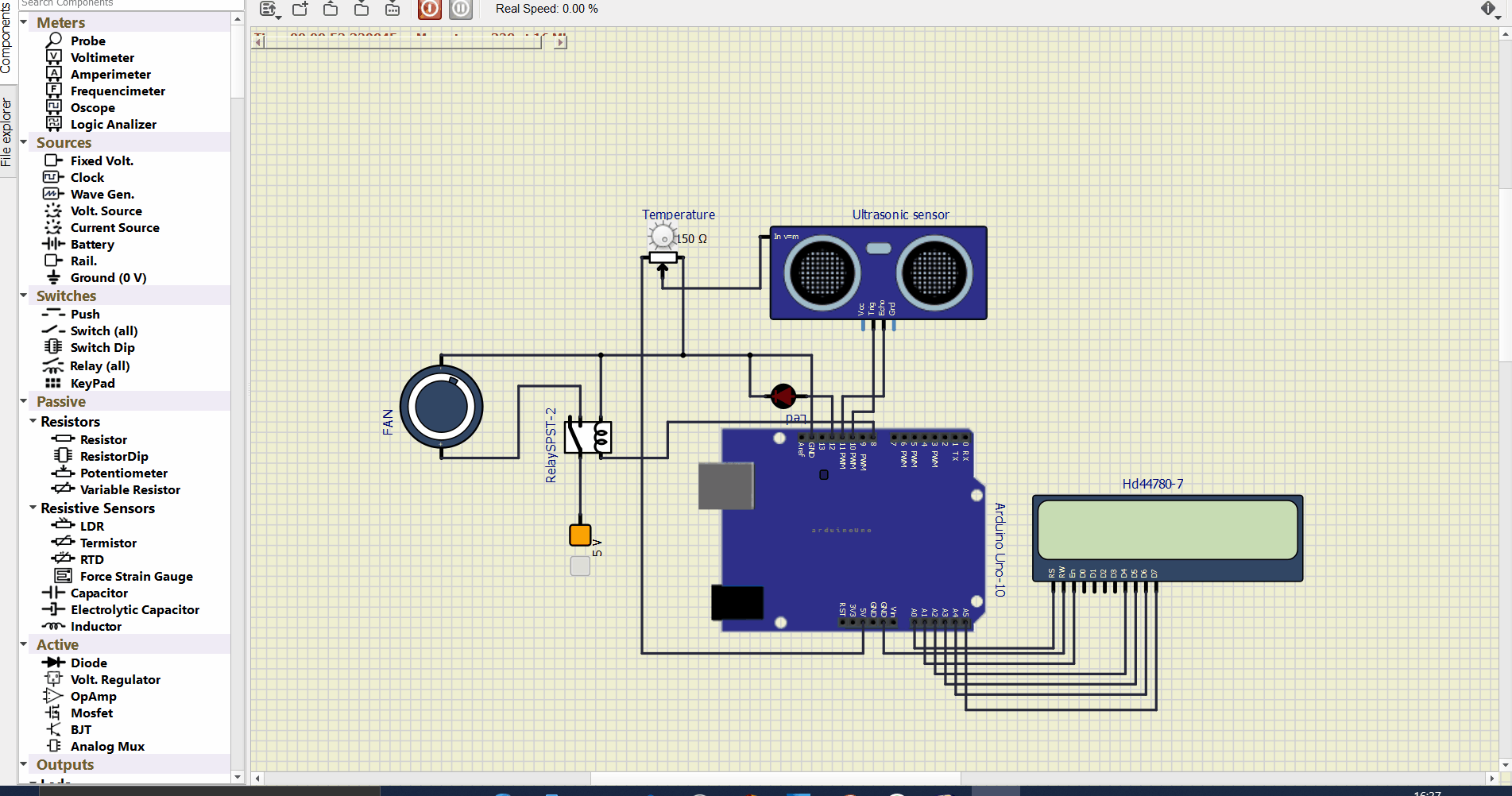
* Distance > 100

[](https://user-images.githubusercontent.com/98537406/156934770-96f476af-000d-42a0-ba6f-cc601129bdff.PNG)

* Water level is LOW & Motor is ON at distance > 100

[](https://user-images.githubusercontent.com/98537406/156934810-81f27cac-1ad4-46d9-8c72-f684875b515b.PNG)

**video**

[](https://user-images.githubusercontent.com/98537406/157227662-214a623e-2b3d-4156-be73-d197bf3f17ff.mp4)