# **Automatic Water Pump Controller**

Submitted in partial fulfilment of the requirements

of the degree of Bachelors in Engineering

by

Mr. Kshitij S. Mayekar

Mr. Tanmay K. Dangle

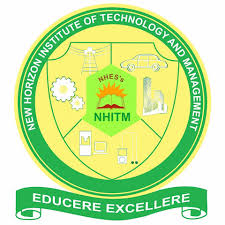
Mr. Tushar R. Pawar

Mr. Soham D. Sawant

(Roll No.10,2,14,18)

Supervisor (s):

(Dr. Anu Malhan)



New Horizon Institute of Technology and Management

Department of Electronics and Electrical Engineering

(2019-2020)

## CERTIFICATE

This is to certify that the project entitled **“Automatic Water Pump Controller”** is a bonafide work of **“Kshitij S. Mayekar, Tanmay K. Dangle, Tushar R. Pawar, Soham D. Sawant” submitted** to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **“Bachelor of Engineering”**.

Dr. Anu Malhan Dr. Niranjan Kulkarni Dr. Prashant Deshmukh

Guide Head of Department Principal

Thesis Approval for Project Report Approval for T. E.

# This project report entitled (**Automatic Water Pump Controller**) by (**Kshitij S. Mayekar, Tanmay K. Dangle, Tushar R. Pawar, Soham D. Sawant**) is approved for the degree of Electronics & Electrical Engineering.

Examiners

1.---------------------------------------------

2.---------------------------------------------

Date:

Place:

# **Declaration**

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

--------------------- ----------------------- ----------------------- ----------------------

(Signature) (Signature) (Signature) (Signature)

-------------------- ----------------------- ---------------------- ----------------------

Tushar R. Pawar Kshitij S. Mayekar Tanmay K. Dangle Soham D. Sawant

Date:

**INDEX**

Chapter 1: - Introduction

Chapter 2: - Objective

Chapter 3: - Principle of working

Chapter 4: - Components used

Chapter 5: - Programming

Chapter 6: - Advantages

Chapter 7: -Applications

Chapter 8: - Conclusion

Chapter 9: - Future projections

Chapter 10: - References

**Chapter 1: - Introduction**

The project “Automatic water pump controller” is design to monitor the level of liquid in the tank. It has an automatic pumping system attached to it so as to refill the tank once the liquid gets to the lower threshold, while witching of the pump once the liquid gets to higher threshold. Sustainability of available water resource in many regions of the world is now a dominant issue. Therefore, efficient use and water monitoring are potential constraint for home or office water management system.

**Chapter 2: - Objective**

The goal or objective of which the designed device is expected to accomplish is to build an automatic water pump controller. In this project sensors are placed at different level of the tank and with the aid of the sensors, the microcontroller monitors the level of liquid at any particular point in time, some of the objectives are:

* To design an automatic water monitoring system
* To avoid wastage of water
* To incorporate an interactive medium between the end user and the machine.
* To prevent the over usage of pumping machine and prevent it from wear and tear.

**Chapter 3: - Principle of working**

**Block Diagram**

Let’s learn the Designing and Working of this project using given block diagram.

PUMP

RELAY

LCD

ARDUINO

BUTTON

SONAR SENSOR

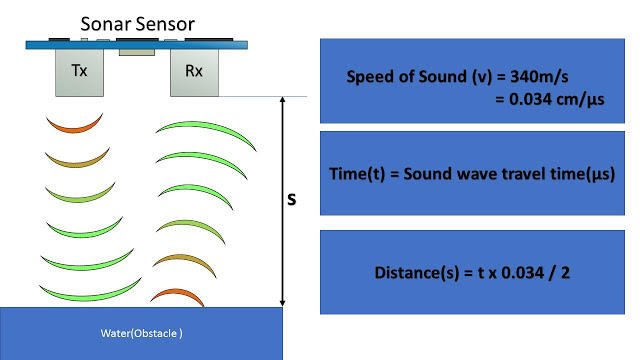
LED

The first block is **Arduino UNO R3**, Arduino is the brain of this project. It will take input from the sensors and control all other units according to the value received.

The second block is **16x2 LCD display**. This unit will display the Water Level in percentage as well as in Bar Diagram, it will also show the Pump status. This section will also notify us whenever the Sump tank is empty.

The third block is the **Sonar Sensor**. This is used to measure the water level present on the overhead water tank.

Let’s understand the working principle of the Sonar sensor.



Sonar Sensor emits an ultrasound at 40 kilohertz, which travels through the air, and if there is an object or obstacle on its path, it will bounce back to the module.

Arduino will use the echo pin, present on Ultrasonic sensor to measure sound wave travel time in microseconds.

Considering the travel time and the speed of the sound, you can calculate the distance using the formula shown here.

The fourth block is the Sump Water level sensor. These are two copper wires which are dipped in to the Sump Water Tank. And Analog pin present on the Arduino Nano will be used to sense the presence of water.

The fifth block is the push button, this is used to measure the Tank height at the time of installation of this Circuit. This can further used if you wish to replace the water tank with a new one.

The sixth block is the Internal Relay plush Water Pump. Arduino will control the Water pump using the Internal relay. The relay present on the circuit can be used to start any kind of 1 HP single phase Water pump without starters.

The Seventh block is LED which is used an indication for the water level.

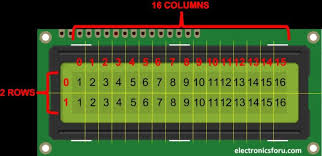
The Eight block is the 10k ohms potentiometer which is used for adjusting the contrast of LCD display.

**Chapter 4: - Components used**

**Arduino UNO**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo.The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

**16\*2 LCD Display**

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in colour or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as present words, digits, and seven-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of many small pixels, while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the colour of the backlight, and a character negative LCD will have a black background with the letters being of the same colour as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance.

**5V Relay**

A relay is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations.

Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance telegraph circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

The traditional form of a relay uses an electromagnet to close or open the contacts, but other operating principles have been invented, such as in solid-state relays which use semiconductor properties for control without relying on moving parts. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called protective relays.

Latching relays require only a single pulse of control power to operate the switch persistently. Another pulse applied to a second set of control terminals, or a pulse with opposite polarity, resets the switch, while repeated pulses of the same kind have no effects. Magnetic latching relays are useful in applications when interrupted power should not affect the circuits that the relay is controlling.

**10kΩ Resistor**

Resistors are used extensively throughout electrical and electronic circuits. Resistor devices may provide a fixed, variable, or adjustable value of resistance. Adjustable resistors are referring to as rheostats, or potentiometers. Resistor values are expressed in Ohms, the electric resistance unit.

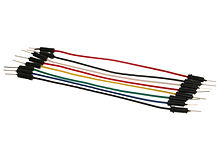
**Pin Header Connector Strip**



A pin header (often abbreviated as PH, or simply header) is a form of electrical connector. It consists of one or more rows of male pins typically spaced 2.54 millimetres (0.1 in) apart, but common sizes also include 5.08 millimetres (0.2 in), 5.00 millimetres (0.197 in), 3.96 millimetres (0.156 in), 2.00 millimetres (0.079 in), 1.27 millimetres (0.05 in) and 1.00 millimetre (0.04 in). The distance between pins is commonly referred as pitch in the electronic community.

In the past, a pin header was known as a Berg connector, but the term fell out of favour because pin headers are manufactured by many companies.

**Jumper wire**



A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

**Ultrasonic sensor**

Ultrasonic transducers or ultrasonic sensors are a type of acoustic sensor divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

In a similar way to radar and sonar, ultrasonic transducers are used in systems which evaluate targets by interpreting the reflected signals. For example, by measuring the time between sending a signal and receiving an echo the distance of an object can be calculated. Passive ultrasonic sensors are basically microphoning that detect ultrasonic noise that is present under certain conditions.

Ultrasonic probes and ultrasonic baths apply ultrasonic energy to agitate particles in a wide range of materials

**Chapter 5: - Programming**

const int trig\_pin = 6;

const int echo\_pin = 7;

const int led = 8;

double duration;

int distance;

long total;

const int relay = 10;

float a;

int percent;

#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin

// with the arduino pin number it is connected to

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

byte line[8] = {

0b11111,

0b11111,

0b11111,

0b11111,

0b11111,

0b11111,

0b11111,

0b11111

};

byte smiley[8] = {

0b00000,

0b00000,

0b01010,

0b00000,

0b00000,

0b10001,

0b01110,

0b00000

};

byte smiley1[8] = {

0b00000,

0b00000,

0b00000,

0b00000,

0b00000,

0b10001,

0b01110,

0b00000

};

byte arrow[8] = {

0b01000,

0b00100,

0b11110,

0b11111,

0b11110,

0b00100,

0b01000,

0b00000

};

byte arrow1[8] = {

0b00000,

0b00000,

0b00000,

0b10000,

0b00000,

0b00000,

0b00000,

0b00000

};

byte arrow2[8] = {

0b00000,

0b00000,

0b10000,

0b11000,

0b10000,

0b00000,

0b00000,

0b00000

};

byte arrow3[8] = {

0b00000,

0b10000,

0b11000,

0b11100,

0b11000,

0b10000,

0b00000,

0b00000

};

byte arrow4[8] = {

0b10000,

0b01000,

0b11100,

0b11110,

0b11100,

0b01000,

0b10000,

0b00000

};

#include<EEPROM.h>

int present = 0;

int past = 0;

int counter = 1;

void setup() {

pinMode(trig\_pin, OUTPUT);

pinMode(echo\_pin, INPUT);

pinMode(led, OUTPUT);

pinMode(relay, OUTPUT);

digitalWrite(relay, HIGH);

pinMode(13, INPUT);

lcd.begin(16, 2);

Serial.begin(9600);

Serial.setTimeout(10);

lcd.createChar(0, line);

lcd.createChar(1, smiley);

lcd.createChar(2, smiley1);

lcd.createChar(3, arrow);

lcd.createChar(4, arrow1);

lcd.createChar(5, arrow2);

lcd.createChar(6, arrow3);

lcd.createChar(7, arrow4);

lcd.setCursor(0, 0);

lcd.print("hello!!");

lcd.setCursor(5, 0);

lcd.write(byte(1));

delay(1000);

lcd.setCursor(5, 0);

lcd.write(byte(2));

delay(1000);

lcd.setCursor(5, 0);

lcd.write(byte(1));

delay(1000);

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

}

void loop() {

void show();

void show1();

void show2();

lcd.clear();

int sensorValue = analogRead(A0);//set the depth of your tank by using the potentiometer

int val = map(sensorValue, 0, 1023, 0, 500);//convert 0 to 1023 into 0 to 500 cm

total = val;

digitalWrite(trig\_pin, LOW);

delayMicroseconds(2);

digitalWrite(trig\_pin, HIGH);

delayMicroseconds(10);

digitalWrite(trig\_pin, LOW);

duration = pulseIn(echo\_pin, HIGH);

distance = 0.034 \* duration / 2;

a = total - distance;

percent = ((a / total) \* 100) + 8;//8% to avoid over flow and prevent sensor from water

if (percent > 100)

{

percent = 100;

}

if (percent < 0)

{

percent = 0;

}

present = digitalRead(13);

if (present == HIGH && present != past)

{

counter++;

}

past = present;

if (counter % 2 == 0) //if button pressed it enters in to measurement mode to measure the depth of the tank

{ lcd.clear();

lcd.write("measurement mode");

Serial.println("\n measurement mode");

lcd.setCursor(0, 1);

lcd.print("Distance:");

Serial.println("distance:");

lcd.print(distance);

lcd.print("cm");

Serial.print(distance);

Serial.print("cm");

delay(200);

lcd.clear();

}

else

{

if (percent < 20)

{

digitalWrite(relay, LOW);

}

if (percent >= 95 && present != HIGH)

{

digitalWrite(led, HIGH);

digitalWrite(relay, HIGH);

Serial.println("alert");

show();

show2();

delay(500);

lcd.clear();

show1();

delay(500);

}

else

{ digitalWrite(led, LOW);

}

if (percent < 99 && digitalRead(13) != HIGH)

{ show();

show2();

delay(200);

// put your main code here, to run repeatedly:

}

counter = 1;

}

}

void show()

{

lcd.setCursor(5, 1);

lcd.print("Depth:");

lcd.print(total);

lcd.print("cm");

lcd.setCursor(0, 1);

lcd.print(percent);

lcd.print("%");

lcd.setCursor(0, 0);

lcd.print("level");

lcd.setCursor(5, 0);

lcd.write(byte(4));

lcd.setCursor(5, 0);

lcd.write(byte(5));

lcd.setCursor(5, 0);

lcd.write(byte(6));

lcd.setCursor(5, 0);

lcd.write(byte(7));

lcd.setCursor(5, 0);

lcd.write(byte(3));

lcd.setCursor(6, 0);

lcd.write(byte(4));

lcd.setCursor(6, 0);

lcd.write(byte(5));

lcd.setCursor(6, 0);

lcd.write(byte(6));

lcd.setCursor(6, 0);

lcd.write(byte(7));

lcd.setCursor(6, 0);

lcd.write(byte(3));

}

void show2()

{

if (percent > 10)

{

lcd.setCursor(7, 0);

lcd.write(byte(0));

}

if (percent > 20)

{

lcd.setCursor(8, 0);

lcd.write(byte(0));

}

if (percent > 30)

{

lcd.setCursor(9, 0);

lcd.write(byte(0));

}

if (percent > 40)

{

lcd.setCursor(10, 0);

lcd.write(byte(0));

}

if (percent > 50)

{

lcd.setCursor(11, 0);

lcd.write(byte(0));

}

if (percent > 60)

{

lcd.setCursor(12, 0);

lcd.write(byte(0));

}

if (percent > 70)

{

lcd.setCursor(13, 0);

lcd.write(byte(0));

}

if (percent > 80)

{

lcd.setCursor(14, 0);

lcd.write(byte(0));

}

if (percent > 90)

{

lcd.setCursor(15, 0);

lcd.write(byte(0));

}

}

void show1()

{

lcd.blink();

lcd.setCursor(5, 1);

lcd.print("tank full");

delay(50);

lcd.noBlink();

}

**Chapter 6: - Advantages**

* Automatic operation saves you manual labour time.
* Consumes a small amount little energy, perfect for on-going operations.
* Indicates water levels in any type of storage tank or body of liquid.
* A water alarm is loud so you can easily hear it.

**Chapter 7: -Applications**

* Can be used in water tanks to control water levels
* Automatically turn ON/OFF pumps
* Can be used in factories, commercial complexes, apartments, home,
* Fuel tank level gauging
* Oil tank level control
* High & low-level alarms
* Pool water level control
* Life station switches
* Leachate level control
* Cooling tower water level control
* Sewage pump level control
* Remote monitoring liquid
* Water level control
* Pump controller
* Stream level monitoring
* Sump pump
* Tsunami warning and sea level monitoring
* Process batch control & monitoring
* Irrigation control

**Chapter 8: - Conclusion**

* This system is very beneficial in rural as well as urban area.
* It helps in efficient utilization of available water sources.
* If used on a large scale, it can provide a major contribution in the conservation of water for us and future generations.

In these days, when Earths reserve of consumable water is decreasing every moment, every drop has its value. Water level controller is a simple yet effective way to prevent wastage of water. Its simplicity in design and low-cost components make it an ideal piece of technology for the common man.

**Chapter 9: - Future Projection**

This project can be used to control water flow. However, there is no way of knowing whether the source of water flow, which in case is the underground tank, actually has water or not. If no water source is present, then the submersible pump would start running unnecessarily and overheat itself. This could be care by implementing another sensor. Also, the rate of water input must be always be equal to or greater than the rate of water output. To make this happen we could use a speed regulator. If these issues are taken care of then a more efficient and reliable performance can be achieved.

**Chapter 10: - References**

**WEBSITES: -**

* https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/index.jsp
* https://www.arduino.cc/
* https://www.powerzone.com/resources/glossary/centrifugal-pump
* https://www.elprocus.com/step-step-guide-build-electronic-circuit/