MINI PROJECT REPORT

ON

"WATER LEVEL DETECTION SYSTEM"

Submitted in

Partial Fulfillment of requirements for the Award of Degree

of

Bachelor of Technology

In

Computer Science and Engineering

By

(Group Number:39)

Akshat Tiwari (1816410042)
Ankur Sachan (1816410056)
Anubhav Srivastava (1816410064)

Under the supervision of
Kumar Saurabh
(Asst. Professor)

Pranveer Singh Institute of Technology.
Kanpur - Agra - Delhi National Highway - 2
Bhauti -Kanpur - 209305.
(Dr. A.P.J. Abdul Kalam Technical University)

CONTENT

OBJECTIVE

The objective of this project is to notify user the amount of water present in the overhead water tank.

The transmitter circuit makes use of an ultrasonic sensor and Arduino board to measure the water level in terms of distance. This data is sent to the receiver circuit using RF communication to calculate the distance at what level the water is present and produce the output percentage accordingly and reflect in the screen.

fig (1): Echo illustration

This project helps in reduction in wastage of water, it can be implemented in very large scale i.e. as a part of river flow management system or in small scale i.e. households.

All components are interfaced with the Arduino and works by automation as per uploaded. 0% is the condition of the tank where the quantity of water present in it is null and finally 100% is maximum condition. We have to monitor and maintain the tank when the water in it is pouring in it.

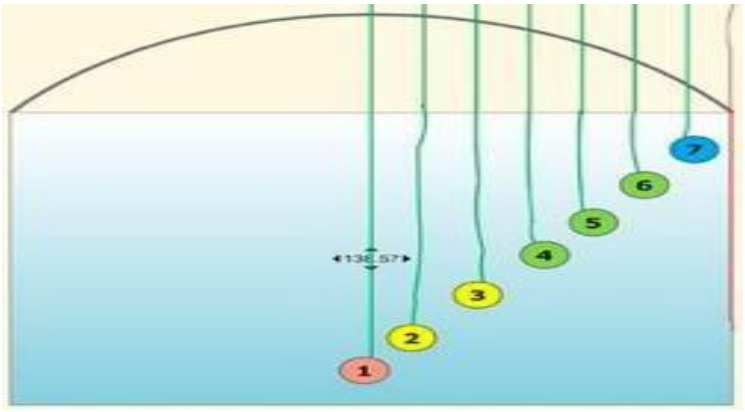
INTRODUCTION

These days almost all the technologies use some kind of computer based control system. In this computer system are used to control hardware in flexible way. These control systems can be implemented for optimizing the river flow in order to minimize the flood caused by the water overflow.

The water level management has always been a major issue, so new methods are required to manage the water level. Here our proposed system works better than existing system.

Existing System:

The existing system works with many number of wires let us take a tank which is divided into percentages like 10%,20%, . . . 100%. Every wire is of different size kept at different percentages of the tank [3]. If the water present in the tank is 10% the water touches only one wire so LCD returns the amount of water present in it. This take much time in implementation. As water touches the wires may get damaged.

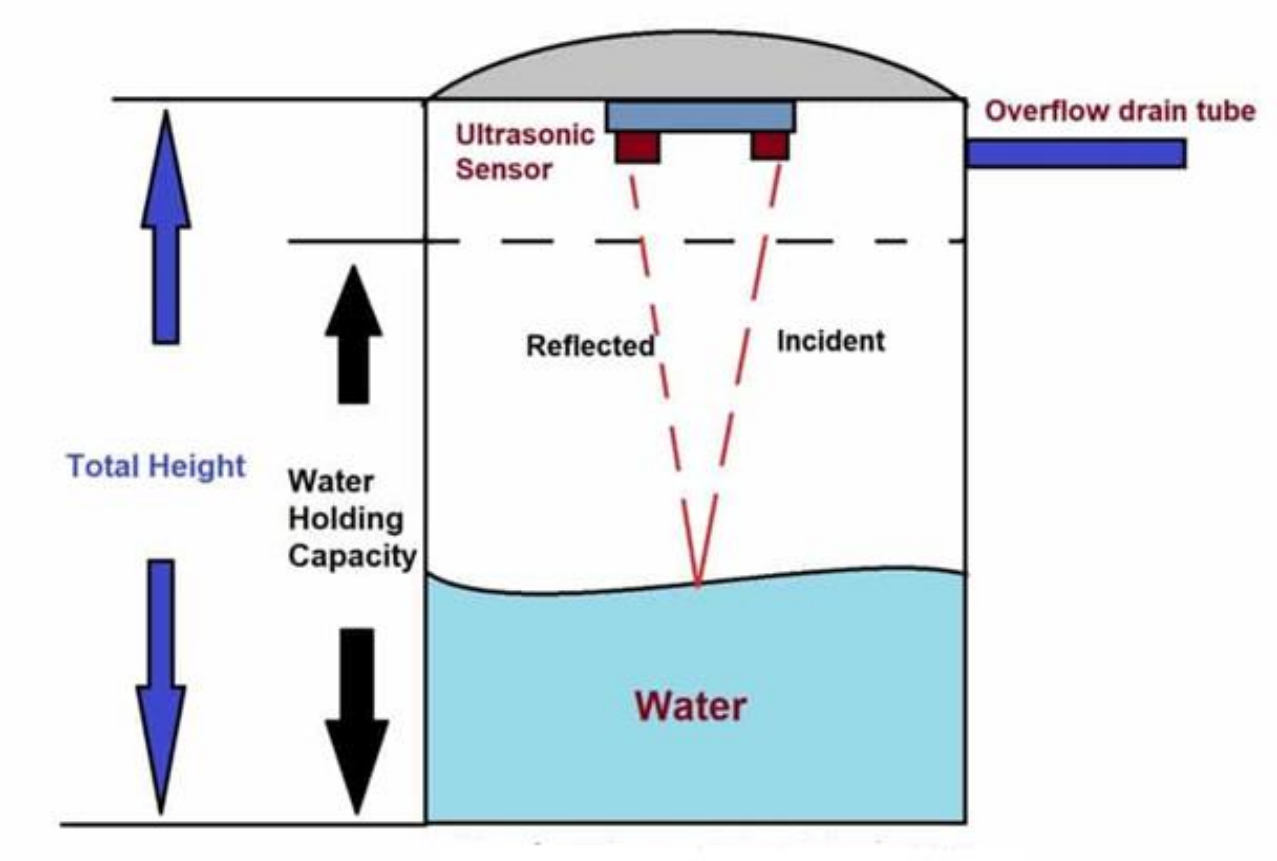


The above figure shows the existing water level management system here in this system they hang the wires into water and each wire is connected to an LED. here the water is touching all 7 wires so the wire 7 returns that water in tank is maximum the LED connected to the wire 7 will glow up, likely when the water level is only touching wire 1 similarly the connected LED will glow up and we will know the percentage of water present in the tank.

Proposed system:

This proposed system works under automation and contains the components like Arduino, ultrasonic sensor, motor pump, relay, led, buzzer and LCD in which each component has its own functionality but Arduino looks like heart of the project as all the components are interfaced with Arduino. The ultrasonic sensor plays major role in

determining the water level present in the tank. This sensor is fitted under the lid of the tank and uses the concept called “echo”. The sensor contains two small openings called trig and echo. The trig works like a small speaker in which it sends the ultrasonic waves and the echo acts as a small microphone in which it receives the reflected waves which are sent by trig and this echo returns the distance. So by this we are known of water level in the tank through LCD. Another advantage of this proposed system is motor automation. We have designed a code that whenever the tank is getting low about of 10% the motor automatically starts and stops when the tank reaches to 90% so here no one required for controlling to motor this is the main advantage of proposed system. Since no component touches the water there is no chance of damage to the components whereas in the existing system which contains wires that are placed in water can get damaged after sometime.



The figure shows working of our project “water level detection system”. Here the ultrasonic sensor will first calculate the distance of the blank vessel and the distance is stored and after every data read.
When we pour water into the vessel as the water level increases it will calculate the distance and compute the calculation and show the percentage of the water present in the vessel it can be vary from 0% to 100%.
The output is delayed by 3 microseconds that means after every 3 microseconds it will calculate the distance and perform the calculation and show the output in the console window.

FEASIBILITY STUDY

ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service,or environment is accessible by as many people as possible .It is very cost effective and work for the welfare of the nature.

MAINTAINABILITY:

In software engineering, maintain ability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations when the tank level is given if we want to implement it the large scale area then we need high quality if sensors which provide accurate and good data through which the calculation can be done.

PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependent assemblies would have to be configured in such case.

VALIDATION:

It is the process of checking that a software system meets specifications and that it fulfils its intended purpose.

It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle.

Software validation checks that the software product satisfies or fits the intended use (high-level checking) i.e. the software meets the user requirements, not as specification arte facts or as needs of the who will operate the software only; but as the needs of all the stakeholders**.**

Why use Water Level indicator?

- **Saves Power**

  By using our project, we can save power. I can be used in places where there is problem of Load Shedding. As it is automatically controlled, it limits the amount of electricity. As today energy conservation is the utmost need, using one of these devices is useful.

- **Saves Money**

  Now, as we know that automatic water level controller conserves power, it saves money as well. Water regulation is optimized using this device that means wastes electricity and wasted water is kept at a minimum. This saves a huge amount of money along with manpower.

- **Works Automatically**

  The utmost advantage of water level controller is that it can work on its own. It is because of relay and timer switches that there is no need to operate them manually. This means that a lot of human work is reduced.

- **Maximizes Water**

  Additionally, water usage can be maximized with a water level controller. Often, water pumps get more use during the middle of the day. A water level controller is helpful because it automatically provides more water during the day.

TECHNOLGY USED

Arduino:

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.

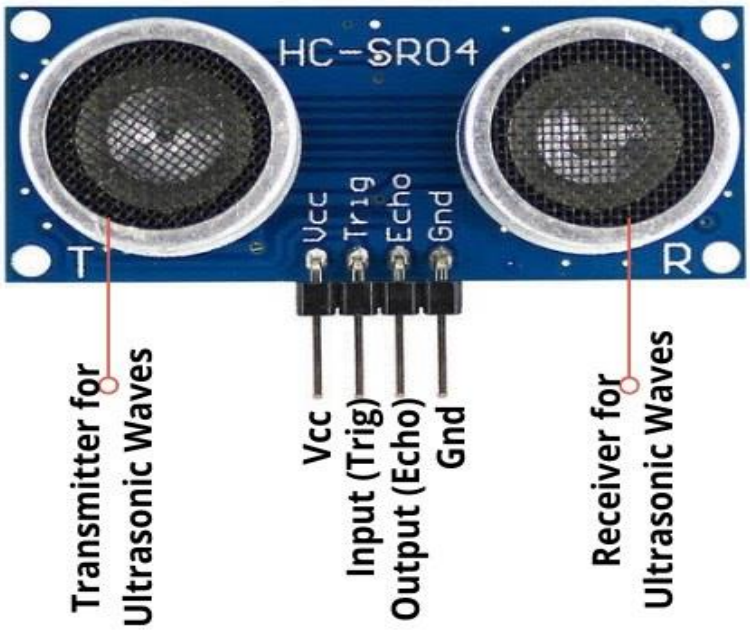
The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.
It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.

It is 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.

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases.

The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.
While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

HC-SR04 ultrasonic sensor:



An ultrasonic transducer is a device that convert energy into ultrasound or sound waves above the normal range of human hearing. Ultrasonic sensor generates high frequency sound waves and evaluates the echo which is received back by the sensor. The physical shape of ultrasonic sensor is shown in above figure.

It is an ultrasonic sensor, also known as an ultrasonic transducer that is based on a transmitter and receiver and mainly used to determine the distance from the target object. The amount of time it takes to send and receive waves will determine how far the object is placed from the sensor.

It mainly depends on the sound waves working on "non-contact" technology. The required distance of the target object is measured without any damage, giving you accurate and precise details. This sensor comes with a range between 2cm to 400cm and is used in a wide range of applications including speed and direction measurement, wireless charging, humidifiers, medical ultrasonography, sonar, burglar alarms, and non-destructive testing.

- HC-SR04 is an ultrasonic sensor mainly used to determine the distance of the target object.

- It measures accurate distance using a non-contact technology - A technology that involves no physical contact between sensor and object.

- Transmitter and receiver are two main parts of the sensor where former converts an electrical signal to ultrasonic waves while later converts that ultrasonic signals back to electrical signals
- These ultrasonic waves are nothing but sound signals that can be measured and displayed at the receiving end.

- Following table shows the main features of this ultrasonic sensor.

Parameter	Value
Main Parts	Transmitter & Receiver
Technology Used	Non-Contact Technology
Operating Voltage	5 V
Operating Frequency	4 MHz
Detection Range	2cm to 400cm
Measuring Angle	30°
Resolution	3mm
Operating Current	<15mA
Sensor Dimensions	45mm x 20mm x 15mm

It gives precise measurement details and comes with accuracy (resolution) around 3mm, terming there might be a slight difference in the calculated distance from the object and the actual distance.

The HC-SR04 Ultrasonic sensor is an ultrasonic transducer that comes with 4 pin interface named as Vcc, Trigger, Echo, and Ground. It is very useful for accurate distance measurement of the target object and mainly works on the sound waves. As we connect the module to 5V and initialize the input pin, it starts transmitting the sound waves which then travel through the air and hit the required object. These waves hit and bounce back from the object and then collected by the receiver of the module

No.	Pin Name	Pin Description
1	VCC	The power supply pin of the sensor that mainly operates at 5V DC.
2	Trig Pin	It plays a vital role to initialize measurement for sending ultrasonic waves. It should be kept high for 10us for triggering the measurement.
3	Echo Pin	This pin remains high for short period based on the time taken by the ultrasonic waves to bounce back to the receiving end.
4	Ground	This pin is connected to ground.

Distance is directly proportional to the time these waves require to come back at the receiving end. The more the time taken, more the distance will be. The waves will be generating if the Trig pin is kept High for 10 µs. These waves will travel at the speed of sound, creating 8 cycle sonic burst that will be collected in the Echo pin. The echo pin remains turned on for the time these waves take to travel and bounce back to the receiving end. This sensor is mainly incorporated with Arduino to measure the required distance. Following formula is used to calculate the distance of the object.

$$S = (V \times t)/2$$

Distance (cm)= duration*0.034/2;

Where S is the required distance, V is the speed of sound and t is the time sound waves take to come back after hitting the object. We need to divide the value by 2 because time will be double as the waves travel and bounce back from the initial point. Dividing it by 2 will give the actual distance of the target object.

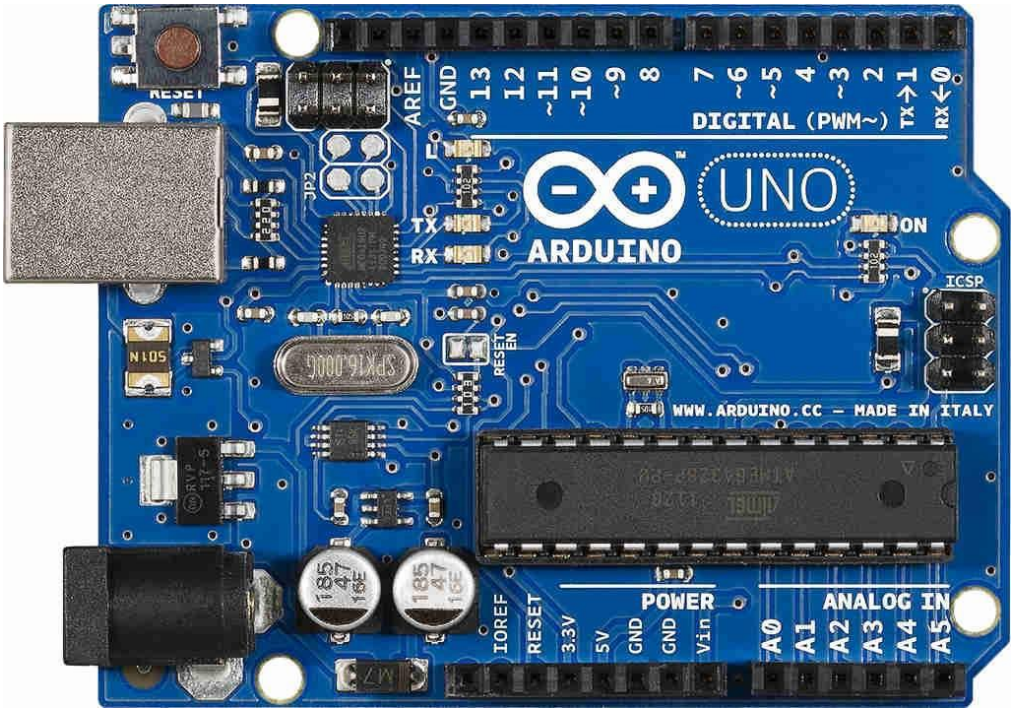
Applications of the HC-SR04 SENSOR:

- Speed and direction measurement
- Wireless charging
- Humidifiers
- Medical ultrasonography
- Burglar alarms
- Embedded system
- Depth measurement
- Non-destructive testing

Technical specifications:

- Microcontroller: Microchip ATmega328P [7]
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- UART: 1
- I2C: 1
- SPPI: 1
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

Arduino UNO:



This is the Arduino Uno R3. In addition to all the features of the previous board, the Uno now uses an ATmega16U2 instead of the 8U2 found on the Uno (or the FTDI found on previous generations). This allows for faster transfer rates and more memory. No drivers needed for Linux or Mac (inf file for Windows is needed and included in the Arduino IDE), and the ability to have the Uno show up as a keyboard, mouse, joystick, etc.

The Uno R3 also adds SDA and SCL pins next to the AREF. In addition, there are two new pins placed near the RESET pin. One is the IOREF that allow the shields to adapt to the voltage provided from the board. The other is a not connected and is reserved for future purposes. The Uno R3 works with all existing shields but can adapt to new shields which use these additional pins.

The Arduino Uno is a microcontroller board based on the ATmega328. Arduino is an open-source, prototyping platform and its simplicity makes it ideal for hobbyists to use as well as professionals.

The Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 microcontroller chip programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Arduino Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform

Pin Description

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 – 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.

TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

General pin functions:

LED:- There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.

VIN:- The input voltage to the Arduino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V:- This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

3V3:- A 3.3 volts supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND:- Ground pins.

IOREF:- This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.

Reset: Typically used to add a reset button to shields that block the one on the board

Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers.

The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows a .inf file is required. Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows serial communication on any of the Uno's digital pin

HARDWARE REQUIREMENTS

- System : Pentium 4, Intel Core i3, i5, i7 and 1 GHz Minimum
- RAM : 512Mb or above
- Hard disk : 500 MB or above
- Input Device : Keyboard and Mouse
- Output Device : Monitor or PC
- Board used : Arduino uno
- Jumper wires
- Ultrasonic Sensor
- Tank for detection of the water level

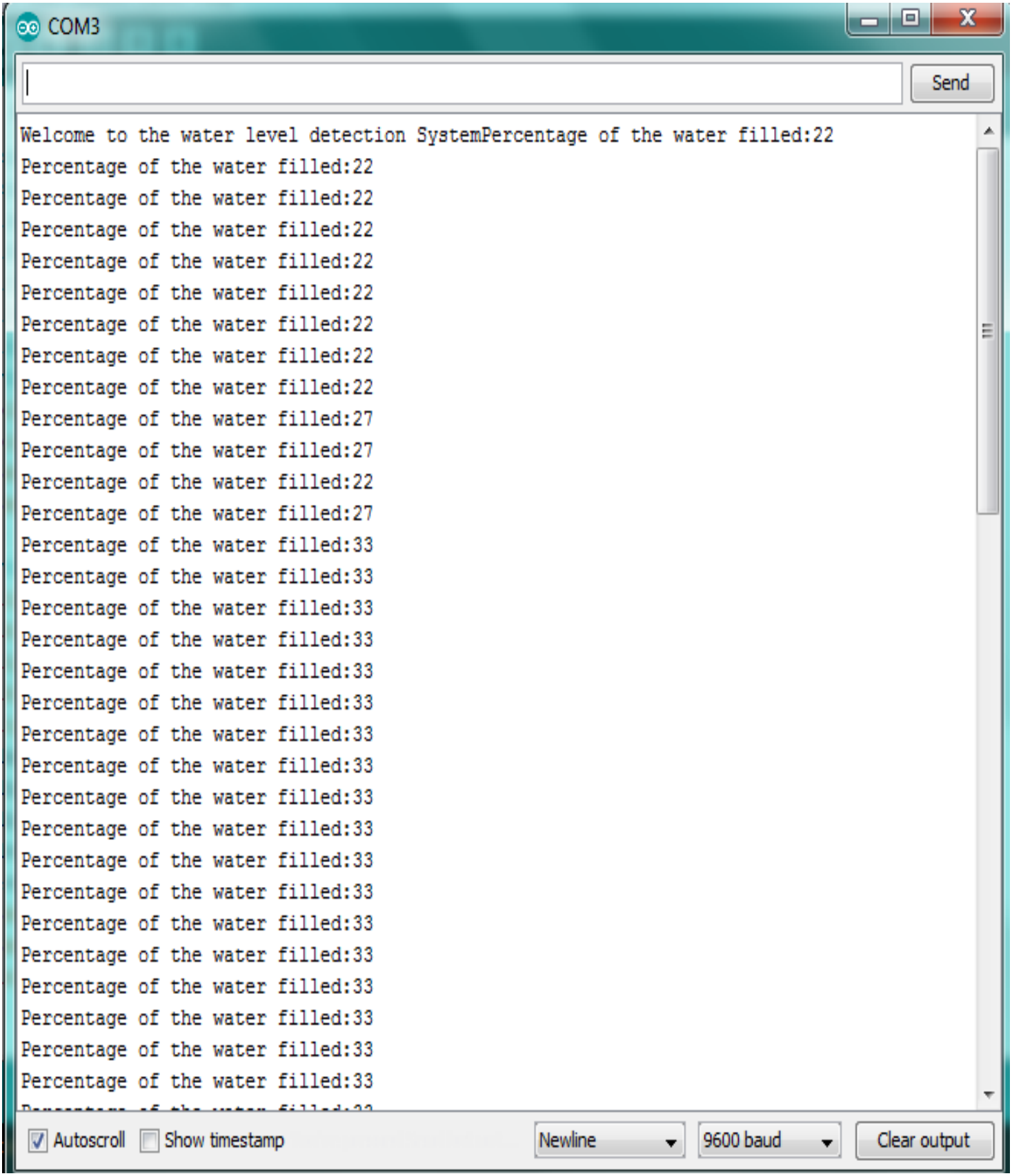
SOFTWARE REQUIREMENTS

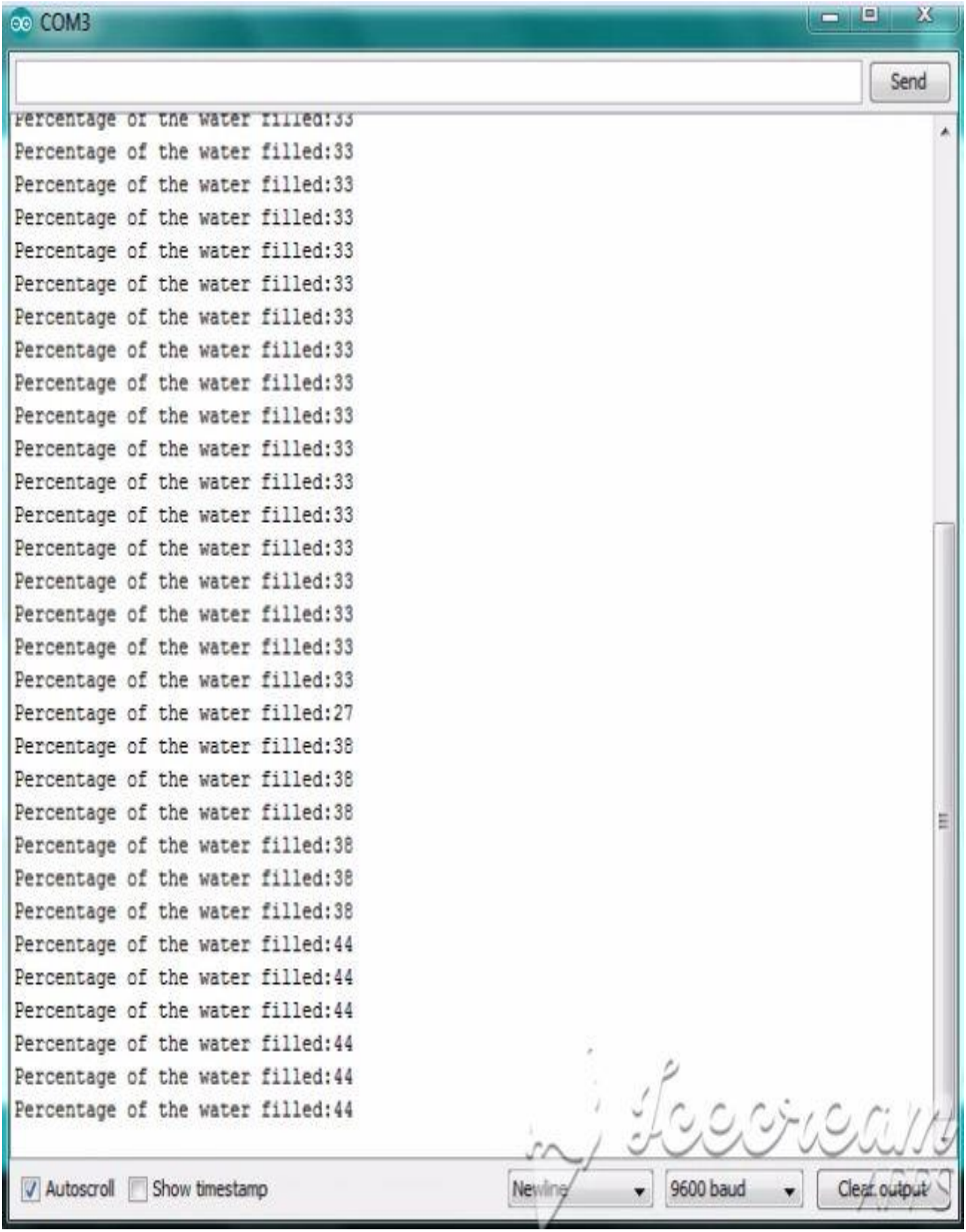
- Operating System : Windows 7, 10 or Higher Version
- Platform : The **Arduino** Integrated Development Environment
- Programming Language: C language

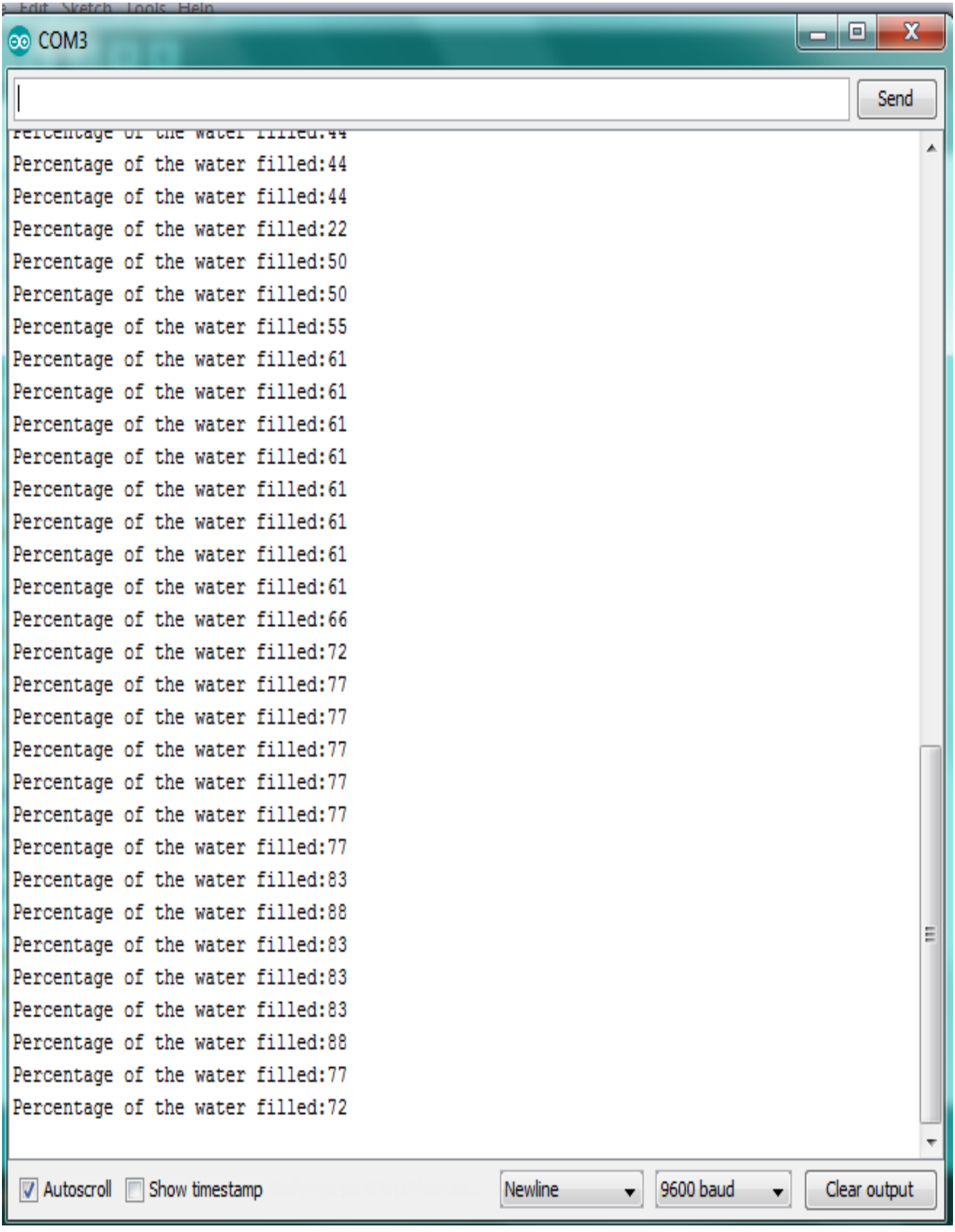
CODING

```
const int trigPin = 7;
const int echoPin = 8;
long duration;
int distanceCm1;
int distanceCm;
int d;
int e;
void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
Serial.print("Welcome to the water level detection System");
}
void loop() {
//the loop function will run until we close the terminal window we may call it as an infinite loop.
digitalWrite(trigPin, LOW);
delayMicroseconds(2);          //Here we are giving a delay of 2 microsecond so that it can calculate the rest.
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);         //Here we are giving a delay of 2 microsecond so that it can calculate the rest.
digitalWrite(trigPin, LOW);                                  //Here we are making the trig pin low.
duration = pulseIn(echoPin, HIGH);  //Here we are taking input that is we are taking the distance of the tank
distanceCm1=18;                                               //This is the distance of our tank
distanceCm= duration*0.034/2;                                 //Here we are calculating the distance
d=distanceCm1-distanceCm;
e=d*100/distanceCm1;                          // here we calculating the percentage of the water filled
Serial.print("Percentage of the water filled:");        //here we are displaying the percentage of the water filled
Serial.println(e);
delay(1000);                    //here we are giving the delay so that after every 1 second it will check the
level of the water and display on the output screen
}
```

Output Window:




COM3
Send
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:33
Percentage of the water filled:27
Percentage of the water filled:38
Percentage of the water filled:38
Percentage of the water filled:38
Percentage of the water filled:38
Percentage of the water filled:38
Percentage of the water filled:44
Percentage of the water filled:44
Percentage of the water filled:44
Percentage of the water filled:44
Percentage of the water filled:44
Percentage of the water filled:44
Autoscroll
Show timestamp
Newline
9600 baud
Clear output


Edit Sketch Tools Help
COM3
Send
Percentage of the water filled:44
Percentage of the water filled:44
Percentage of the water filled:22
Percentage of the water filled:50
Percentage of the water filled:50
Percentage of the water filled:55
Percentage of the water filled:61
Percentage of the water filled:61
Percentage of the water filled:61
Percentage of the water filled:61
Percentage of the water filled:61
Percentage of the water filled:61
Percentage of the water filled:61
Percentage of the water filled:61
Percentage of the water filled:66
Percentage of the water filled:72
Percentage of the water filled:77
Percentage of the water filled:77
Percentage of the water filled:77
Percentage of the water filled:77
Percentage of the water filled:77
Percentage of the water filled:77
Percentage of the water filled:83
Percentage of the water filled:88
Percentage of the water filled:83
Percentage of the water filled:83
Percentage of the water filled:83
Percentage of the water filled:88
Percentage of the water filled:77
Percentage of the water filled:72
Autoscroll
Show timestamp
Newline
9600 baud
Clear output

CONCLUSION

Prototype of Water Level Detection System has been tested and reasonably good performance is shown based on the test result. The main contribution of this performance is the ping sensor calibration by adjusting calculation of distance based on an actual data. Testing need to be carried out for the real fluctuated water surface condition to get the system performance in the real condition.

We have successfully run a test on different tanks and the water level of each tank is shown to us on the output screen and when we remove the water it will also show the decrease in the percentage of the water level hence our project runs on every possibility we can calculate the distance and percentage accurately in different climate such as when the water is cold, hot, warm or moderate temperature.

The water level data is successfully displayed locally or remotely, therefore this prototype can be used as a part of the bigger system, such as, river flow management system which controls the stream to minimize the flood.

The receiver can act as a water level data feeder that can be transmit the data remotely to the server.

Since computer is used as a part of receiver module, therefore more sophisticated system can be developed to display and analysis time series water level data, instead of only displaying the current.

Suggestions:

- The better radio frequency module can be used, in order to reach longer distances.
- This tool can be developed to measure water depth using the other type of ultrasonic sensors such as srf02, srf04 or srf08
- The better display applications can be developed for example by displaying time series data in graphic form
- The better receiver module can be developed to receive the water level data from multiple transmitter modules.
- The repeater can be employed to reach a longer distance, especially for remote area.

FUTURE SCOPE

- Automatic water level monitoring system has a good scope in future especially for agriculture sector. There are any areas where we need water level controller. It could be agricultural fields, overhead tanks. We can make this project wireless by using NRF transmitter and receiver. We can also add Ethernet shield so that we can get all the information using mobile phones and control it accordingly.

- It can also be implemented as a part of river flow management system.

- We can upgrade it by installing new modules such as wifi or sms module. We can use a resistor which can help to automate the water level indicator and if we can set the value that when the water level is below 15% Then the water pump will start automatically and when the tank is full then water pump will automatically shut down and the level of the water tank will depict on the screen of the mobile phone because we can use sms module so that the owner of the home or institute can keep a track of the consumption.

- When we install an automation system then it will be cost effective and very helpful to every segment of the people.

REFRENCES

- Budiharto Widodo, ”Perancangan Sistem dan Aplikasi Mikrokontroler”, Elex Media Komputindo, Jakarta, 2005
- NPTEL
- Tim Lab. Mikroprosesor, “Pemrograman Mikrokontroler AT89S51”, Andi Offset, Yogyakarta, 2007;
- URL: itp.nyu.edu/physcomp/sensors/Reports/Par allaxUltrasonikDistanceSensor/14 Februari 2006
- Lingga Wardhana,“Belajar Sendiri Mikrokontroler AVR Seri ATMega8535 Simulasi, Hardware, dan Aplikasi”, Andi Offset, Jakarta
- URL:http://www.innovativeelectronics.com/inn ovative_electronics/download_fles/artikel/AN7 3.pdf
- URL: http://depokinstruments.com/2010/02/08/t eknik-pengendalian-lcd-krakter-i/#more-585/8 Februari 2010
- Tim Peneliti dan Pengembangan Wahana Komputer, "Teknik Antarmuka Mikrokontroler dengan Komputer Berbasis DELPHI", Salemba Infotek, Jakarta, 2006
- URL:http://www.scribd.com/doc/14779388/Pe mrograman-menggunakan-Delphi-70/25 Januari 2009
- URL:http://terminaltechno.blog.uns.ac.id/2009/11/07/pengkabelan-port-serial-port-usb-danport-serial-rs-232/15