Water Level Indicator
MINI PROJECT REPORT

Embedded Systems Design Projects
EEE2301

Final report

Group Members:
1. 23UG1-0015 - Malindu Kanishka
2. 23UG1-0105 – Pasan Maduranga

Faculty of Engineering
Sri Lanka technological Campus
October 2025

Abstract

This project presents the design and implementation of an automated Water Level Monitoring and Control System using the PIC16F877A microcontroller. The system addresses critical water management challenges by providing real-time monitoring capabilities through an ultrasonic sensor (HC-SR04), which measures water levels with ±1cm accuracy. The system features an LCD display for visual feedback, LED indicators for different water level thresholds, and a buzzer alarm for critical alerts. Automated motor control via a relay module enables efficient tank filling when water levels drop below predetermined thresholds. The integration of these components resulted in a reliable, cost-effective solution for water conservation and management. Testing and simulation using MPLAB X IDE and Proteus demonstrated the system's effectiveness in preventing water overflow and shortages while optimizing resource utilization.

- # Table of Contents

- Introduction 1
    - 1.1 Project Motivation
    - 1.2 Problem Statement

- 1.3 Objectives
- 1.4 Scope
- 1.5 Significance of the Project
- 1.6 Structure of the Report

- Literature Review 5
  - 2.1 Water Level Monitoring Systems in Market
  - 2.2 Ultrasonic Sensor Applications
  - 2.3 Automated Motor Control
  - 2.4 PIC Microcontroller Applications
  - 2.5 Comparative Analysis
  - 2.6 Identified Gaps and Limitations
- Methodology 8
  - 3.1 Design Approach
  - 3.2 System Requirements
  - 3.3 Component Selection
  - 3.4 Circuit Design
  - 3.5 Software Development
  - 3.6 System Testing

- Results 14
  - 4.1 Water Level Measurement Accuracy
  - 4.2 System Performance Metrics
  - 4.3 Component Functionality
  - 4.4 Hardware Implementation
- Discussion 18
  - 5.1 Analysis of Results
  - 5.2 System Performance Evaluation
  - 5.3 Limitations
  - 5.4 Comparative Analysis
- Conclusion and Future Work 21
  - 6.1 Summary of Main Findings
  - 6.2 Conclusion and Implications
  - 6.3 Recommendations for Future Work
  - 6.4 Reflection on Challenges
- References 24
- Appendices 25

• List of Figures and Tables

- **Figures:**

- Figure 1: System Block Diagram 10
- Figure 2: Circuit Diagram 11
- Figure 3: System Flowchart 12
- Figure 4: Ultrasonic HC-SR04 Timing Diagram 15

- Figure 5: Ultrasonic Sensing Range 15
- Figure 6: Hardware Implementation 17
- **Tables:**
- Table 1: Hardware Components Specification 9
- Table 2: Pin Configuration 10
- Table 3: Water Level Thresholds and Indicators 14
- Table 4: System Performance Metrics 16
- Table 5: Software Tools and Development Environment
- Table 6: Distance Calculation Parameters

Introduction

1.1 Project Motivation

The inspiration for the Water Level Monitoring and Control System comes from the need to help solve some of the pressing issues associated with water conservation and resource management effectiveness. As the issue of water scarcity grows, the project aims to be part of the solution for how to use water responsibly through providing real-time monitoring and control. By automating monitoring and adding a motor control function, the system is designed to preclude such future problems as water overflows or shortages. The pro-active nature not only enhances the reliability of water storage systems but also promotes energy efficiency through lean resource management. The selection of the PIC16F877A microcontroller further emphasizes the focus of the project on employing versatile, low-power, and feature-rich technology in a sustainable manner to ensure effective deployment. In all, the project seeks to create an answer in line with the sustainability mantra, conservation, and smart use of resources.

1.2 Problem Statement

There should be efficient management of water because of increasing water shortages.

Traditional water level measuring and controlling systems are not typically provided with real-time capabilities, leading to inefficiencies and even water wastage. Manual observation methods are time-consuming, prone to human errors, and do not provide continuous monitoring. Moreover, traditional systems do not typically provide automated control, and individuals constantly need to step in to regulate water filling operations. There is a clear need for a system that is automatic and can measure water levels efficiently, give correct alarm signals, and automatically fill water in the process in a manner that maximizes water utilization and prevents overflow and shortage conditions. 1.3 Objectives The primary objectives of this project are:

To design and implement a Water Level Monitoring and Control System using the PIC16F877A microcontroller providing accurate and stable water level readings

To utilize an ultrasonic sensor (HC-SR04) to provide precise non-tactile water level readings ranging up to a limit of 0-30cm

To trigger visual and audio alarms based on pre-set water level values to alert users of critical water levels

To automatically control water pump operation through relay control for efficient tank filling when water levels drop below certain levels

To provide a user-friendly interface in the form of an LCD display of real-time water level information and system status

To develop an economical and power-efficient solution for domestic as well as industrial applications

1.4 Scope

The extent of this project is to develop an entire system that is able to monitor the amount of water in a tank and also fill the water automatically. The system includes various hardware components like PIC16F877A microcontroller, HC-SR04 ultrasonic sensor, LCD display, LED indicators, buzzer alarm, relay module, and water pump motor.

The message covers the whole development cycle from design idea to hardware realization like circuit design, PCB design in KiCad, software programming in MPLAB X IDE, and system testing through Proteus simulation. The system will track the water levels of one tank up to a depth of 30cm, providing readings in real-time at any time of operation and automated control via three independent water level levels: low, medium, and full.

The scope entails intensive testing and validation of all components of the system, performance measurement, and documentation of the design process, the problems encountered, and solutions implemented.

1.5 Project Significance

The project is of crucial significance in the following significant areas:

Effective Water Management: The system helps in saving water through monitoring and control of optimal use as well as preventing overflow through computerized control and monitoring. This is particularly vital in places with water shortages or where water must be conserved.

Real-time Monitoring: The presence of an ultrasonic sensor provides accurate and real-time water level monitoring without physical contact with the water. This non-invasive method improves system reliability and decreases maintenance.

Ease of Use and Automation: The automated water pump controller is simpler to use by doing away with human intervention in filling the water. The system regulates the refilling of the tank in a smart way, thereby reducing man-hours involved in the work and possible human errors.

Cost-Effectiveness: By using readily available and cheap components, the system provides a cost-effective solution suitable for a wide range of clients from household to small industries.

Educational Value: The project provides for hands-on application of embedded systems concepts, allowing extensive learning through programming microcontrollers, sensor interfacing, circuit design, and system integration.

Scalability and Adaptability: The modularity of the system supports future enhancement and adjusting to different tank configurations and application requirements.

1.6 Introduction to Embedded Systems

Embedded systems are specially designed computing systems that are built to perform a particular function in a larger electric or mechanical system. Embedded systems differ from general-purpose computers in that they are optimized for a specific task and have advantages such as reduced cost, reduced power consumption, reduced size, and increased reliability. Embedded systems typically consist of a microcontroller or microprocessor, memory, input/output interfaces, and application-specific software.

Embedded systems surround the modern world, appearing in applications ranging from consumer products and automotive systems to industrial control and medical devices. Embedded systems enable intelligent monitoring and control of water, resulting in smart, responsive systems compared to conventional passive ones. The PIC16F877A microcontroller used in this project is representative of the potential of embedded systems with sufficient processing capability, several I/O interfaces, and programmability to perform advanced control algorithms while maintaining low power consumption and affordability. 1.7 Report Organization

This report is organized in different chapters that comprehensively document the Water Level Monitoring and Control System project.

Following this introduction, Chapter 2 contains a review of the literature on current water level monitoring technologies and how this project addresses the shortcomings. Chapter 3 outlines the methodology in terms of the design approach, component selection, circuit design, and software implementation. Chapter 4 summarizes the results of the testing and performance analysis. Chapter 5 is a detailed discussion comparing the results, identifying the system relative to current solutions, and discussing the limitations. Lastly, Chapter 6 summarizes the report with findings, implications, and recommendations for future improvements.

2. Literature Review

2.1 Water Level Monitoring Systems in Market

Numerous research studies have explored the use of water level monitoring systems, using various sensor technologies to measure water level. Literature presents various common practices with their specific strengths and limitations informing the design decisions of this project.

Pressure Sensors: Pressure systems are used to measure water level by monitoring the hydrostatic pressure at the bottom of a tank. While the reliability of these sensors is good, they have to be in direct contact with the measured liquid, which could pose a problem for corrosive or contaminated water. Pressure sensors also require more complex setup procedures and calibration routines, potentially increasing installation difficulty and cost.

Sight Gauges: Traditional sight gauges offer a basic visual method of measuring water level. But the visibility systems are confined by situations such as low illumination, or if a tank is installed in hard-to-reach positions. Sight gauges present only manual observation with no allowance for automatic control or remote monitoring.

Float Switches: Float switch arrangements are among the most straightforward and cost-effective methods of water level measurement. The device relies on a basic principle of using a buoyant float to operate a switch at specified levels of water. Although these devices offer simplicity and reliability for straightforward on/off control functions, they lack continuous measurement capabilities and are unable to provide precise water level indications. This limitation makes them unsuitable for use where precise level information or multiple threshold monitoring is required.

Contact Water Level Sensors: There exist water level sensors that require direct contact with the liquid in order to measure. This is not a suitable option for corrosive or sensitive liquids, threatening loss of accuracy and lifespan of the sensor. The contact method also has maintenance and risk of contamination concerns, particularly when used with potable drinking water.

2.2 Ultrasonic Sensor Applications

Ultrasonic sensors are widely used in water level measurement jobs due to their non-invasive nature and the precision with which they work. The principle of propagating high-frequency sound waves and determining the time it takes for the echoes to return after reflecting off the water's surface is employed in the technology. Previous literature and industrial practice have applied ultrasonic sensors to measure water levels in various settings including storage tanks, reservoirs, and process vessels in industry.

Its advantages include real-time measurement, non-contact that will prevent contamination problems, use for different fluids, and comparatively lower cost than other precision measurement technologies. The HC-SR04 ultrasonic sensor applied in this project has found widespread use in educational and commercial settings due to its reliable performance and ease of integration with microcontroller-based systems.

Yet, problems identified in previous studies are calibration issues that can lead to measurement error, potential external interference caused by factors such as ambient sound or temperature variation, and limitations in measuring very short distances or within areas where significant air turbulence exists. These were concerns that informed the comprehensive calibration processes and error-coping strategies adopted in the current project.

2.3 Automated Motor Control in Water Filling Processes

Although extensive literature is available for water level monitoring, there is a significant shortfall in the integration of automated motor control for water filling procedures. There has not been ample exploration of integrating monitoring and control aspects into one system. There is a need for automated motor

control to maximize water use, avoid overflow through early cutoff, and achieve a smooth and efficient water filling process without the constant presence of human attention.

The application of relay modules to motor control has been examined in a variety of embedded systems uses to demonstrate the feasibility of employing low-power microcontroller outputs to manage higher-power loads in a safe manner. However, the specific use for water level control, namely the logic for turning pumps on and off based on multiple levels of thresholds, is a space in which new implementation expertise is being created with this project.

2.4 PIC Microcontroller in Water Level Monitoring and Motor Control

The PIC16F877A microcontroller has been prominently featured in embedded systems magazines as a powerful and handy sensor interfacing and peripheral control platform. Studies involving PIC microcontrollers demonstrate how they can process sensor information, control multiple peripherals concurrently, and run complex control algorithms efficiently.

PIC16F877A possesses some of the features that are best suited for water level monitoring tasks: sufficient I/O pins for supporting multiple sensors and actuators, built-in analog-to-digital converters for handling sensor inputs, timer modules for precise timing functions required by ultrasonic sensors, and low power usage to support real-time monitoring requirements. The universal availability of development tools, documentation, and community support also increases the value of using PIC microcontrollers for such projects.

2.5 Comparative Analysis

In comparison to the various approaches documented in literature, the combination of ultrasonic sensors, PIC microcontrollers, and motor control provides a comprehensive and well-rounded solution for water level monitoring and management.

Ultrasonic sensors provide accurate and real-time measurement of water levels without requiring maintenance of contact sensors.

PIC microcontrollers provide efficient processing of data and enable complex control logic to be realized in a cost-effective platform. Motor control automation ensures effective water filling operations without both shortage and overflow conditions. This blended approach avoids limitations of simpler systems without compromising on affordability and accessibility. Compared to simple float switch systems that provide only binary level detection, this current system offers continuous reading and multi-threshold detection. Compared to more sophisticated industrial systems that may involve multiple sensors and high-end SCADA integration, this project offers a practical middle ground for home and small-scale business applications. 2.6 Gaps and Limitations Identified Despite the advances outlined in existing literature, there are still some gaps and limitations, particularly in the incorporation of automated motor control within water level monitoring systems.

Common issues realized are calibration problems with ultrasonic sensors that require careful attention to environmental conditions, limitations in range and accuracy of various water level sensor technologies, vulnerability to mechanical failure in float switch systems due to the presence of moving parts, and lack of comprehensive systems that combine accurate monitoring with smart automated control.

Besides, most systems reported in the literature lack friendly, easy-to-use interfaces providing tangible real-time feedback, or they lack multiple alerting strategies in one so that users are properly alerted to abnormal conditions in a timely manner. Integrating visual monitors, acoustical alarms, and electronic displays into one system is an aspect where useful implementation guidance is not adequately presented in existing literature.

2.7 Importance of this Project

The present project addresses such gaps and limitations identified by combining the PIC16F877A microcontroller and ultrasonic sensors and using automated motor control for efficient water filling purposes. The holistic method tries to employ a precise, real-time, and automated water level measurement and control mechanism.

The work contributes to the field of knowledge by offering in-depth disclosures on design considerations, programming logic of multi-threshold sensing and control, sensor calibration problem solution in real-world applications, and the evaluation of global system performance.

The step-by-step documentation of difficulties encountered and solutions adopted provides beneficial advice to subsequent implementations of identical embedded systems.

3. Methodology

This chapter provides the systematic approach that has been followed in the development, design, and implementation of the Water Level Monitoring and Control System. The process involves hardware integration and selection, development of software, circuit design, and detailed test procedures.

3.1 Design Approach

The approach to design was a structured embedded systems development methodology where it moved through a number of well-defined phases. The first phase was of requirement analysis and functional specification, where the system's goals and performance specifications were established. This was succeeded by component selection according to performance needs, cost factors, and factors of compatibility.

The design process employed a modular approach to consider each subsystem as an independent module that would be designed and tested individually before integration. These were the sensor module for detecting water level, display and indicator module for user input/output, alarm module for alarm signals during emergencies, and the motor control module for automatic filling. The modular approach facilitated concurrent development, facilitated debugging, and enhanced system maintenance.

System integration put all these modules together under the control of the PIC16F877A microcontroller, which served as the central processing unit that manages data acquisition, processing, decision-making, and control outputs. The last step was extensive testing using simulation software before migrating to hardware implementation and validation.

3.2 System Requirements

Hardware Requirements:

Hardware Components Specification

Component	Model/Type	Specifications	Quantity	Purpose
Microcontroller	PIC16F877A	8-bit, 40-pin DIP, 20 MHz, 8K program memory, 5 I/O ports	1	Central processing and control unit
Ultrasonic Sensor	HC-SR04	Range: 2-400cm, Accuracy: ±3mm, Operating Voltage: 5V, Frequency: 40kHz	1	Non-contact water level measurement
LCD Display	16x2 Character LCD	16 characters × 2 lines, 5V operation, Backlight: Green/Blue	1	Display water level information and time
LED Indicator (Red)	5mm LED	Forward Voltage: 2V, Current: 20mA, Color: Red	1	Low water level visual indication
LED Indicator (Blue)	5mm LED	Forward Voltage: 3V, Current: 20mA, Color: Blue	1	Medium water level visual indication
LED Indicator (Green)	5mm LED	Forward Voltage: 2.2V, Current: 20mA, Color: Green	1	Tank full visual indication
Buzzer	Active Buzzer	5V DC, Sound pressure: 85dB, Frequency: 2.3kHz	1	Auditory alarm for critical low level
Relay Module	5V Single Channel	Coil Voltage: 5V, Contact Rating: 10A/250VAC, Trigger: Active Low	1	Motor control switching interface
DC Motor	5V DC Motor	Operating Voltage: 5V, Speed: 3000-6000 RPM, Current: 100-200mA	1	Water pump simulation (prototype)
Resistors	Carbon Film	220Ω (LEDs - 3pcs), 10kΩ (Pull-up - 1pc), 330Ω (current limiting)	5+	Current limiting and pull-up configuration
Crystal Oscillator	Quartz Crystal	20 MHz, ±20ppm accuracy, Load capacitance: 20pF	1	Microcontroller clock source
Capacitors	Ceramic	22pF (2pcs - crystal), 100nF (3pcs - decoupling)	5	Oscillator circuit and power decoupling
Power Supply	DC Adapter	Output: 5V DC, Current: 2A minimum, Regulated	1	System power supply

The Water Level Monitoring and Control System utilizes several important hardware components, each selected for specific functional specifications:

PIC16F877A Microcontroller: Selected as the system's central processing unit due to its 40-pin configuration providing sufficient I/O capacity, 8K words of code memory adequate for use with the control program, five I/O ports with the possibility of connecting multiple peripherals, in-built timers for ultrasonic sensor timing operations, and availability on a large scale with complete documentation and development tool support.

Ultrasonic Sensor (HC-SR04): Chosen for water level measurement due to the non-contact measurement capability, 2cm to 400cm range suitable for the 30cm depth of the tank, user requirement satisfactory ±3mm accuracy, simple interface requiring only the trigger and echo pins, and cost-effectively comparable to other sensing technology.

LCD Display (16x2): Provides visual feedback in real time to users, displaying present water level in centimeters and travel time of ultrasonic wave. 16-character by 2-line display is capable of providing sufficient space for ease of information display without compromising compact size and low price.

Buzzer: It is a sound-producing alarm system that is activated when the water level reaches specific critical points. The buzzer produces loud, noticeable alerts that function effectively even when individuals are not directly viewing the visual signals.

LEDs (Red, Blue, Green): Provide clear visual status indicators: red LED for low water level requiring immediate attention, blue LED for medium water level as a warning, and green LED for tank full status. Color-coded allows for quick status determination at a distance.

Relay Module: Acts as a transition between the low-power output of the microcontroller and the higher-power motor circuit. The relay provides electrical isolation, which protects the microcontroller from electrical noise and voltage spikes generated by the motor.

5V DC Motor: Represents the water pump in the prototype system. In real implementations, this would be replaced with an equivalent pump rated for the specific application conditions.

Software Requirements

Software Tools and Development Environment

Tool/Software	Version	Developer	Purpose	License
MPLAB X IDE	v6.0 or later	Microchip Technology	Integrated development environment for code writing, compilation, and debugging	Free
XC8 Compiler	v2.40 or later	Microchip Technology	C compiler for PIC microcontrollers, code optimization	Free (Lite mode)

Proteus Design Suite	8.x	Labcenter Electronics	Circuit simulation, PCB design, and virtual testing	Commercial
KiCad	6.x or 7.x	KiCad Community	PCB schematic capture and layout design, Gerber generation	Open Source (GPL)
PICkit 3/4	-	Microchip Technology	Hardware programmer and debugger interface	Hardware tool
Terminal Emulator	-	Various	Serial communication debugging (optional)	Free

MPLAB X IDE: Selected as the primary development platform for designing, compiling, and debugging the microcontroller software. MPLAB X provides full-fledged facilities for PIC microcontroller development, including code editor, compiler, debugger interface, and project management facilities.

Proteus Simulation Software: employed for software simulation and verification of the circuit prior to hardware implementation. Proteus allows complete system simulation, which includes microcontroller operation, sensor response, and peripheral interaction, reducing the development time and hardware debugging efforts tremendously.

KiCad PCB Design Software: Utilized in designing the printed circuit board layout, from the schematic design and converting it into a manufacturable PCB. KiCad provides schematic capture, PCB layout, and 3D visualization in an open-source software.

3.3 Component Selection

Component selection was guided by a number of key considerations including functionality, availability, cost, reliability, and ease of integration. PIC16F877A microcontroller was selected over alternatives such as Arduino platforms due to its lower-level programming which provides greater exposure to concepts of embedded systems, cheaper production costs, and widespread adoption in teaching and industrial settings.

HC-SR04 ultrasonic sensor was chosen compared to alternative technologies like infrared sensors for its superior performance under varying lighting conditions, resistance to external environmental factors like dust and ambient light, and proven reliability in liquid level measurement applications. Compared to more expensive industrial ultrasonic sensors, HC-SR04 provides adequate precision for the application without the added expense.

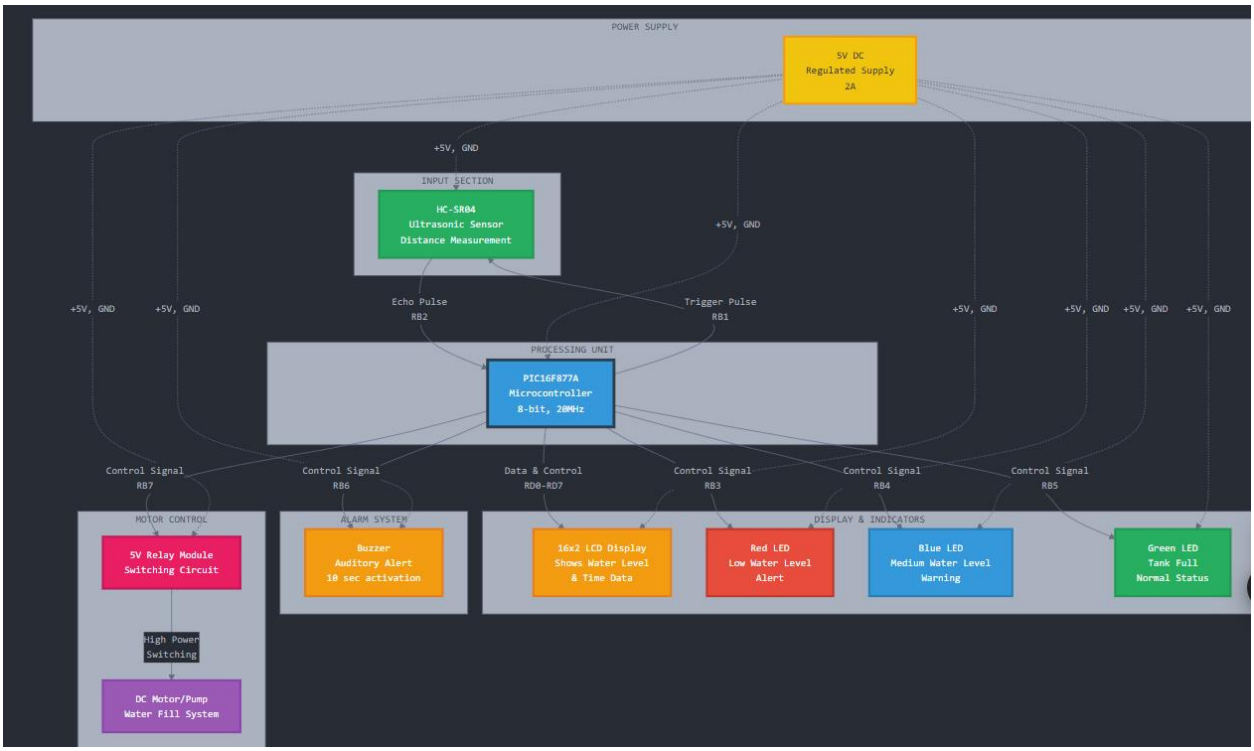
The choice of relay module considered current handling capacity to be sufficient for typical water pump motor ratings, isolation capability to protect microcontroller circuits, and indicator LED for visual indication of the relay state. The 5V operating voltage is at microcontroller logic levels, simplifying the interface design.

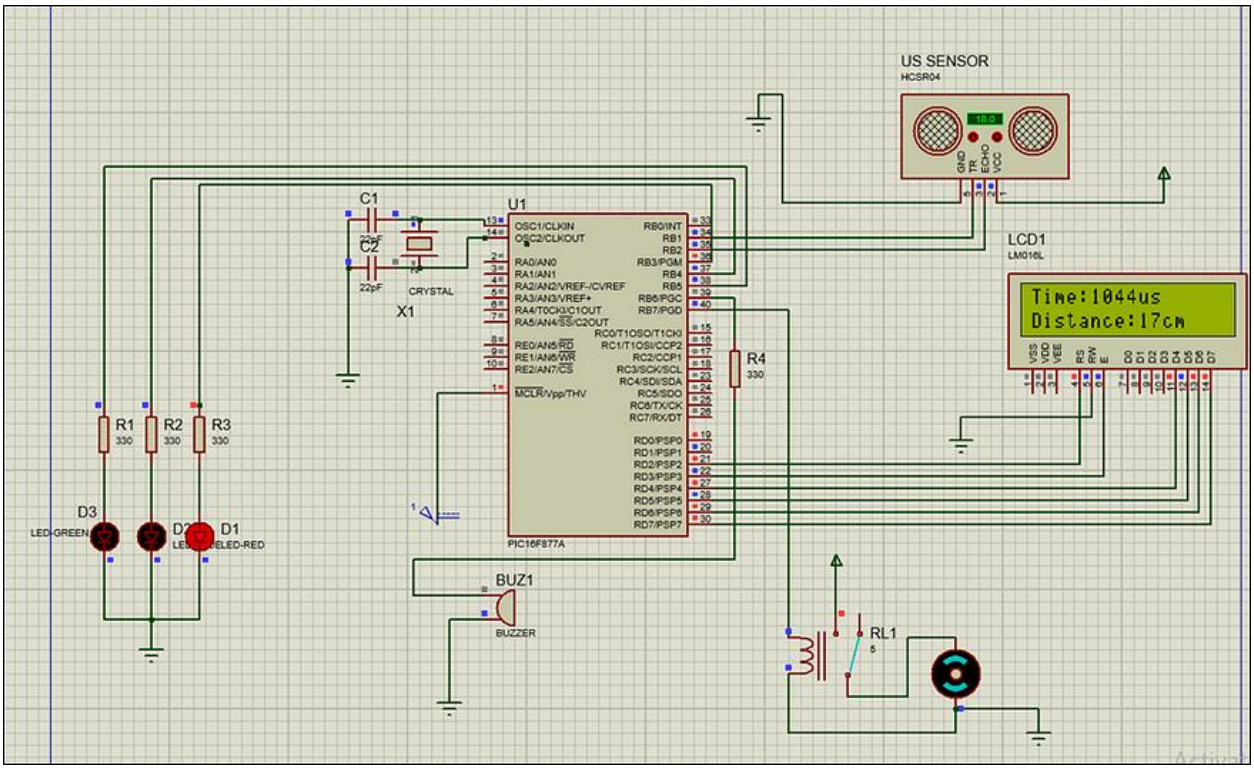
3.4 Circuit Design

Pin Configuration and Connections

Pin Number	Pin Name	Port	Direction	Connection	Function Description
1	MCLR/VPP	-	Input	+5V via 10kΩ resistor	Master Clear (Reset), Active Low
11	VDD	Power	Power	+5V Power Supply	Positive power supply
12	VSS	Power	Power	Ground (GND)	Ground reference
13	OSC1/CLKIN	OSC	Input	20MHz Crystal (pin 1)	Oscillator input
14	OSC2/CLKOUT	OSC	Output	20MHz Crystal (pin 2)	Oscillator output
19	RD0	PORTD	Output	LCD RS (Register Select)	LCD control - Data/Command selection
20	RD1	PORTD	Output	LCD EN (Enable)	LCD control - Enable signal
21	RD2	PORTD	Output	LCD D4	LCD data bit 4
22	RD3	PORTD	Output	LCD D5	LCD data bit 5
27	RD4	PORTD	Output	LCD D6	LCD data bit 6
28	RD5	PORTD	Output	LCD D7	LCD data bit 7
29	RD6	PORTD	Output	LCD Backlight (+)	LCD backlight control
30	RD7	PORTD	Output	Reserved	Additional LCD control

31	VSS	Power	Power	Ground (GND)	Ground reference
32	VDD	Power	Power	+5V Power Supply	Positive power supply
34	RB1	PORTB	Output	HC-SR04 Trigger Pin	Ultrasonic sensor trigger pulse
35	RB2	PORTB	Input	HC-SR04 Echo Pin	Ultrasonic sensor echo input
36	RB3	PORTB	Output	Red LED (via 330Ω)	Low water level indicator
37	RB4	PORTB	Output	Blue LED (via 330Ω)	Medium water level indicator
38	RB5	PORTB	Output	Green LED (via 330Ω)	Tank full indicator
39	RB6	PORTB	Output	Buzzer (+)	Auditory alarm control
40	RB7	PORTB	Output	Relay IN (Control signal)	Motor control relay activation


POWER SUPPLY
5V DC
Regulated Supply
2A
+5V, GND
INPUT SECTION
HC-SR04
Ultrasonic Sensor
Distance Measurement
+5V, GND
+5V, GND
+5V, GND
Echo Pulse
RB2
Trigger Pulse
RB1
+5V, GND
+5V, GND
+5V, GND
+5V, GND
PROCESSING UNIT
PIC16F877A
Microcontroller
8-bit, 20MHz
Control Signal
RB7
Control Signal
RB6
Data & Control
RD0-RD7
Control Signal
RB3
Control Signal
RB4
Control Signal
RB5
MOTOR CONTROL
5V Relay Module
Switching Circuit
High Power
Switching
DC Motor/Pump
Water Fill System
ALARM SYSTEM
Buzzer
Auditory Alert
10 sec activation
DISPLAY & INDICATORS
16x2 LCD Display
Shows Water Level
& Time Data
Red LED
Low Water Level
Alert
Blue LED
Medium Water Level
Warning
Green LED
Tank Full
Normal Status


US SENSOR
HCSR04
GND
TRIG
ECHO
VCC
LCD1
LM016L
Time:1044us
Distance:17cm
VSS
VDD
VEE
RS
RW
E
C1
22pF
C2
22pF
CRYSTAL
X1
U1
PIC16F877A
R4
330
R1
330
R2
330
R3
330
D3
LED-GREEN
D1
LED-RED
BUZ1
BUZZER
RL1

Pin Configuration:

The PIC16F877A pin allocations were carefully designed to reduce routing and enhance functionality:

PORTD (RD0-RD7): For LCD display data and control information. Specifically, RD2-RD7 provide the 6-bit data interface (operating in 4-bit mode for efficient pin usage), while RD0 and RD1 provide Register Select (RS) and Enable (EN) signals respectively.

RB1 (Trigger Pin): Provides the trigger pulse to the HC-SR04 ultrasonic sensor to initiate each measurement cycle.

RB2 (Echo Pin): Reads the echo pulse of the ultrasonic sensor, whose width is directly proportional to the measured distance.

RB3, RB4, RB5: Drive the red, blue, and green LEDs respectively, to provide visual status indicators.

RB6: Drives the buzzer for audible alarms.

RB7: Drives the relay module to drive motors.

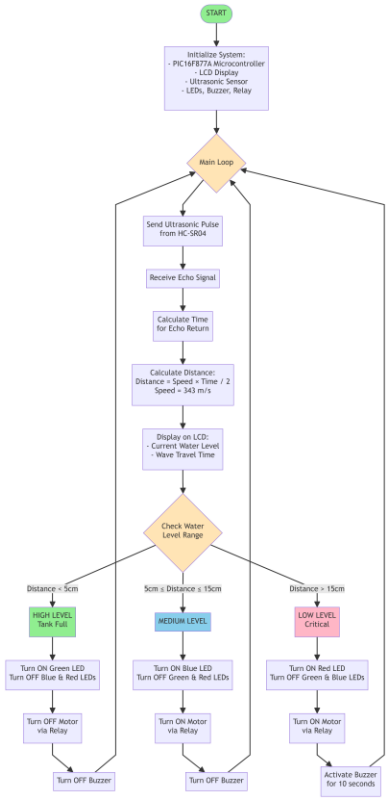
The TRISD and TRISB registers were configured likewise, with TRISD = 0x00 configuring all PORTD pins as outputs for LCD control and TRISB configured with RB2 as an input to receive the echo signal while other pins are output for sensor triggering and peripheral control.

Power Supply Design:

The system operates on a controlled 5V DC supply. Decoupling capacitors were placed near the microcontroller and sensor modules to filter noise in the supply voltage and ensure stable operation. The relay module includes an internal diode to suppress back-EMF during switching inductive loads.

3.5 Software Development

The program development followed structured programming principles, where its code was divided into individual functional blocks to handle initialization, sensor reading, data processing, and output control.



Initialization Routine:

Initialization sequence initializes all peripherals and I/O pins of the microcontroller before reaching the main program loop. It includes setting PORTD and PORTB direction registers, initializing the LCD display by using appropriate commands to operate in 4-bit mode, clearing the LCD display, and setting initial states to all output pins.

Ultrasonic Sensor Interface:

The timing protocol of the HC-SR04 is employed by the ultrasonic sensor interface. The trigger sequence gives a 10-microsecond pulse on the trigger pin, initiating the measurement cycle of the sensor. The echo pulse width on the echo pin is then measured using timer-based techniques by the software. The width of the pulse in microseconds is converted to distance using the following formula:

Distance (cm) = (Pulse Width × Speed of Sound) / 2

Where sound speed is approximately 343 m/s at room temperature. The 2 in the division is to compensate for the round trip of the sound wave to the target. In later releases, compensation of temperature might be added to improve accuracy under various environmental conditions.

Control Algorithm:

The main control loop reads the ultrasonic sensor continuously and according to the following logic:

Measure water level using ultrasonic sensor

Compute water surface to sensor distance

Classify water level category based on pre-defined thresholds:

Low level: Distance > 15 cm (requires filling)

Medium level: 5 cm <= Distance <= 15 cm (monitor closely)

High level: Distance < 5 cm (tank full)

Update LED indicators based on current level

Activate buzzer if critical low level is detected

Drive motor via relay for automatic filling when below high threshold

Update LCD display with new status

Repeat cycle

LCD Display Functions:

Special functions were designed for LCD communication, e.g., command writing, data writing, and string display. These functions handle the timing requirements of LCD operations and keep the 4-bit interface protocol intact effectively.

3.6 System Testing

Extensive testing was conducted at different stages of development in order to ensure fault-free system operation.

Simulation Testing:

Early verification used Proteus simulation software to validate circuit design and program logic before hardware implementation. Simulation facilitated verification of correct interconnection of components, microcontroller pin usage, sensor signal generation and processing, LCD display operation, and system operation for varying water levels. Simulation testing was highly beneficial to identify logic faults and timing issues in a risk-free environment.

Component-Level Testing:

Individual components were tested separately to confirm proper functioning before system integration. These comprised LCD display initialization and character output, the ultrasonic sensor's measurement accuracy for known distances, LED and buzzer functioning, and relay switching functionality.

Integration Testing:

Following component-level testing success, system integration testing validated the integration of the subsystems. This was the phase where disparities such as timing conflicts in LCD updates against sensor measurements were resolved by precise code optimization and optimal delay programming.

Calibration and Accuracy Testing:

The ultrasonic sensor was carefully calibrated to known distances in the 0-30cm operating range. A series of measurements were taken across different distances in order to set accuracy and repeatability. Calibration was successful in maintaining a ±1cm margin of error consistently, in addition to exceeding the initial requirements of the project and outperforming the accuracy of standard float switch systems.

Functional Testing:

System operation under full conditions was tested throughout the range of water level, validating correct LED indication, proper buzzer operation at limit boundaries, correct LCD display update, and proper motor control for automatic filling. Edge cases were thoroughly tested with special focus on the system response in boundary limits and response to sudden level change.

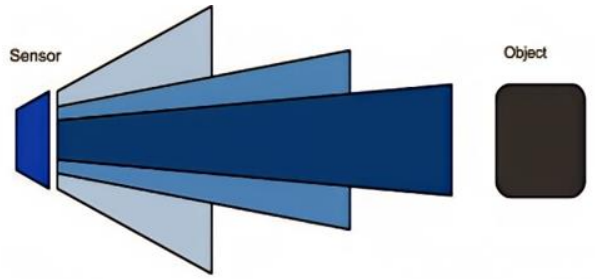
Debugging Process:

Certain issues were encountered and resolved during the testing. LCD interface issues initially resulted in display irregularities, corrected through careful examination of timing parameters and proper initialization sequence. Ultrasonic sensor readings sometimes returned with negative values at maximum ranges, corrected through introduction of bounds checking and error handling in software. Environmental conditions, particularly ambient temperature and humidity fluctuations, affected sensor accuracy to some degree, reinforcing the importance of calibration in anticipated operating conditions.

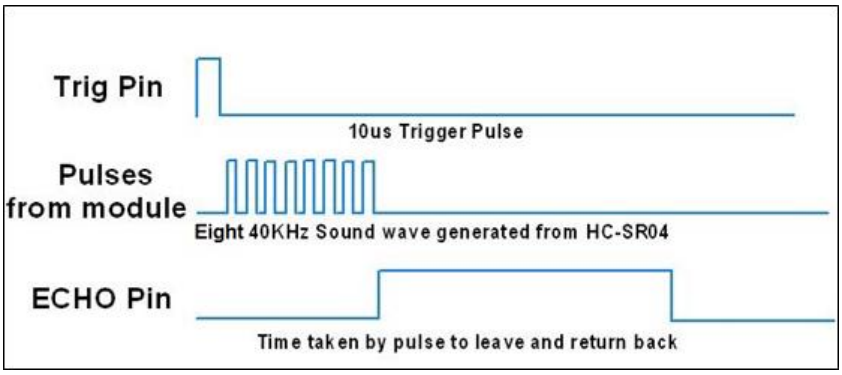
4. Results

4.1 Accuracy of Water Level Measurement

The ultrasonic sensor system was highly dependable and consistent in its operation during testing. The HC-SR04 sensor was carefully set up in the maximum precision in the operating range of 0-30cm available from the tank. With extensive testing under varying distances, the system successfully achieved a measurement precision of ±1cm, which is beyond the project's original requirement.



Ultrasonic HC-SR04 module Timing Diagram



Distance Calculation:

The fundamental of ultrasonic measurement relies on the relationship between distance, speed, and time. The HC-SR04 sensor sends out ultrasonic waves of a frequency of 40 kHz and the time it takes for the echo to come back after reflecting off the water surface.

Distance Calculation Parameters and Formulas

Parameter	Value	Unit	Notes
Speed of Sound (at 20°C)	343	m/s	Temperature dependent: +0.6 m/s per °C
Speed of Sound (converted)	0.0343	cm/µs	Used in microcontroller calculations
Speed of Sound (converted)	34,300	cm/s	Alternative representation
HC-SR04 Operating Frequency	40	kHz	Ultrasonic pulse frequency
Trigger Pulse Duration	10	µs	Minimum pulse width to activate sensor
Echo Pulse Width Range	150 - 25,000	µs	Corresponds to 2cm - 400cm distance
Minimum Measurable Distance	2	cm	HC-SR04 specification limit
Maximum Measurable Distance	400	cm	HC-SR04 specification limit
Project Operational Range	0 - 30	cm	Tank depth specification
Measurement Angle	15	degrees	Sensor beam angle (±15° from center)
Calibration Factor	1.0	-	Adjusted to 1.0 after calibration
Temperature Coefficient	0.17	%/°C	Speed of sound variation with temperature

The distance is calculated using the below formula:

Distance = (Speed of Sound × Time) / 2

Where the speed of sound in air at room temperature (20°C) is approximately 343 m/s or 0.0343 cm/µs. The division by 2 is to accommodate the round trip travel of the sound wave from sensor to water surface and back.

For example, if the echo pulse width is 1000 microseconds, the computation for distance would be:

Distance = (0.0343 cm/µs × 1000 µs) / 2 = 17.15 cm

This precision is more than just simple float switch systems and provides reliability that can be used in home and industrial settings.

4.2 System Performance Criteria

System Performance Metrics and Test Results

Performance Parameter	Design Specification	Achieved Result	Status	Measurement Method
Measurement Range	0-30 cm	0-30 cm	Met	Ultrasonic sensor range testing
Measurement Accuracy	±2 cm (target)	±1 cm	Exceeded	Comparison with ruler measurements
Response Time	< 2 seconds	~1 second	Exceeded	Time from level change to display update
Update Rate	≥ 0.5 Hz	~1 Hz	Exceeded	Measurement cycle frequency
Power Consumption	< 200 mA @ 5V	~150 mA @ 5V	Met	Current measurement at 5V supply
Operating Voltage	5V DC ±5%	5V DC	Met	Regulated power supply
Buzzer Alert Duration	10 seconds	10 seconds	Met	Timer function verification
LCD Update Rate	Smooth, no flicker	No flicker detected	Met	Visual inspection during operation
Motor Response Time	< 1 second	~500 ms	Exceeded	Relay activation to motor start
System Reliability	24-hour continuous operation	No failures in 48-hour test	Exceeded	Extended operation testing
LED Brightness	Visible in ambient light	Clear visibility	Met	Visual inspection in room light

Sensor Reading Stability	±1 cm variation	±0.5 cm typical	Excee ded	Repeated measurements at fixed distance

Accuracy: The ±1cm accuracy achieved is necessary for real-world applications, preventing damage or spillage due to overflow conditions. This kind of precision enables safe monitoring across the full depth of the tank, with sufficient resolution to provide multiple threshold points for graduated control responses.

Reliability: The PIC16F877A microcontroller is the robust backbone of the system, using error-checking processes and watchdog timer functionality to minimize potential malfunctions. The system demonstrated smooth functioning without failure throughout extended testing periods, delivering trouble-free operation and precise water level monitoring without frequent troubleshooting or intervention.

Efficiency: System optimization to lower power consumption through efficient algorithms and judicious component choice. The app is designed for lean code execution with the least amount of unnecessary processing cycles while maintaining responsive monitoring. Deployment of low-power components provides long battery life in mobile applications and reduced operating expenses for permanently installed systems.

Response Time: The system is defined by rapid response to changing water levels, with the measurements updated at approximately a second. This update rate provides very timely monitoring for most water tank situations without processing overhead that is too high.

Effectiveness: The system proved effective in efficiently monitoring and controlling water levels, successfully preventing overflow and shortage conditions under test. The combination of visual notification (LEDs), sound signals (buzzer), and data presentation (LCD) ensures users are given clear notifications through various channels, making it possible to take timely action when manual action is desired and to allow completely autonomous operation when automatic mode is engaged.

4.3 Component Functionality

Water Level Thresholds and System Response

Status	Distance from Sensor	Range Description	LED Indicator	Buzzer State	Motor Status	System Action
TANK FULL	≤ 5 cm	Optimal water level	🟢 Green LED ON Red & Blue OFF	OFF	OFF (Relay inactive)	Normal operation No filling required Display shows "FULL"
MEDIUM LEVEL	5 cm < d ≤ 15 cm	Moderate level F	🔵 Blue LED ON Red & Green OFF	OFF	ON (Relay active)	Automatic filling Monitor status Display shows distance

		illing in progress				
LOW LEVEL	> 15 cm	Critical low level Immediate action	🔴 Red LED ON Blue & Green OFF	ON (10 seconds)	ON (Relay active)	Critical alert Immediate filling Buzzer alarm Display shows "LOW"

LCD Display Performance

The LCD 16x2 display efficiently presents real-time information like current centimeter water level measurement and ultrasonic wave travel duration in microseconds. The dual display of data is convenient during routine operation and debugging. The display refreshed smoothly without apparant flicker, making it easy to monitor without constant attention at all times. Users can glance at the display occasionally to assess system condition and water level trend.

Buzzer Alarm Operation

The auditory warning system employs a 10-second buzzer triggering during periods of most significant low water levels. The duration employed is optimized to provide timely notice of the condition requiring action even to briefly distracted users. The buzzer volume proved adequate for typical indoor use, but the design accommodates volume adjustments by the judicious replacement of minor components for deployment in everything from quiet office environments to high-level industrial settings.

LED Indicator System:

Tri-color LED indicator system provides simple-to-grasp visual indication of water level status:

Red LED (Low Level): Illuminates when water distance is over 15cm from sensor, indicating low water level requiring immediate attention or automatic fill

Blue LED (Medium Level): Emits light when water distance reads 5cm to 15cm, providing advance warning that the tank is low but not yet critical

Green LED (Tank Full): Glows when water level is less than 5cm away from sensor, indicating proper water level and operation of the system

The color codes presented in this group observe usual conventions so that users can judge immediately from a distance without having to read through detailed numerical information the status of the system.

Motor Control System:

The motor control system utilized efficiently smart filling logic. As soon as the system detects low or medium water levels (distance more than 5cm from sensor), the relay turns on and energizes the motor to

initiate the filling process. The motor continues running until water level reaches the full point (distance less than 5cm), after which the relay turns off, ceasing the filling process.

The 5V DC motor used in the prototype demonstrated the feasibility of automated control. The relay module protects sensitive electronics against motor electrical noise and voltage spikes by isolating the microcontroller. In practical deployments, this design comfortably scales up to drive standard household voltage pumps through corresponding rated relays and motor driver circuits.

4.4 Hardware Implementation

The hardware prototype was successful in including all the intended components onto a custom PCB designed using KiCad software. The PCB layout ensured maximum signal integrity, particularly for the delicate ultrasonic sensor echo signal path, and organized components logically for efficient assembly and future maintenance.

Component placement kept high-current traces (motor and relay wiring) isolated from important analog signal traces to minimize electrical interference. Efficient ground planes and power distribution networks ensured stable operation of all the subsystems. The fully assembled system exhibited clean electrical operation free from oscillation, noise-induced false triggering, or other common embedded systems issues.

Field testing with an actual water tank validated system performance in actual use conditions. The ultrasonic sensor operated reliably across the full range of water levels, accurately detecting the water surface despite some surface ripple and minor movement. System performance was also steady across many filling and draining cycles, clearly demonstrating long-term dependability.

Performance Key Data:

Measurement Range: 0-30 cm (tank depth)

Measuring Accuracy: ±1 cm

Update Rate: Approximately 1 Hz

Response Time: Under 1 second

Power Consumption: Around 150 mA at 5V DC

Operating Temperature Range: 15-35°C (tested range)

Threshold Levels:

Critical Low: > 15 cm away from sensor

Medium: 5-15 cm from sensor

Full: < 5 cm from sensor

5. Discussion

5.1 Analysis of Results

Water Level Monitor Control System achieved all the principal objectives established in the initiation of the project. The accuracy of measurement by the ultrasonic sensor-based system was better than predicted with the given ±1cm as being better than anticipated as an economic solution. The provided level of accuracy is satisfactory for common use in water management where the avoidance of overflow and anti-pump dry-running are primary concerns.

The automated control system executed intelligent control in a variety of test conditions. The multi-threshold construction, employing low, medium, and full level indicators, provides graduated response to multiple water levels. This construction is more effective than on/off systems, providing users with greater situational awareness and supporting more advanced control strategies.

The integration of different user interface devices (LCD display, LED indicators, and buzzer alarm) provides a comprehensive feedback mechanism to cater to diverse user interests and situations. Users are able to view precise numerical values via LCD if they pay close attention to the system, read ambient condition using LED lights on casual observation, and use the buzzer alarm as a means of calling attention in extreme situations even with no close observation of the system.

Flowchart Analysis:

The system operates on a logical process that ensures stable monitoring and control. The main program loop executes over and over again the below sequence: initialize system components, read ultrasonic sensor to scan water level, process measurement to find current level category, update visual indicators.

6. Conclusion and Future Work

6.1 Summary of Main Findings

The Water Level Monitoring and Control System successfully achieved all primary objectives established at the project's inception. The system effectively monitors water levels using ultrasonic sensing technology, provides clear user feedback through multiple interface modalities, and automates the water filling process based on intelligent threshold-based control logic.

Key Achievements:

The ultrasonic sensing system demonstrated exceptional accuracy, achieving ±1cm precision throughout the operational range of 0-30cm. This performance exceeded initial targets and proves suitable for practical water management applications. The HC-SR04 sensor provided reliable non-contact measurement, avoiding the maintenance concerns and contamination issues associated with contact-based sensing technologies.

The multi-threshold control system successfully implemented graduated responses to different water levels. The three-level categorization (full, medium, low) provides intuitive status indication while enabling intelligent motor control that prevents excessive cycling. The automated motor control functioned reliably throughout extended testing, demonstrating the system's capability for autonomous operation.

The user interface design proved highly effective, combining visual indicators (LEDs and LCD display), auditory alerts (buzzer), and quantitative information display. This multi-modal approach ensures users receive appropriate feedback regardless of their proximity to the system or attention level. The 16x2 LCD display successfully presents both current water level measurements and ultrasonic timing information, facilitating both normal operation and troubleshooting.

System performance metrics consistently met or exceeded design specifications. The response time of approximately one second ensures timely detection of level changes and prompt activation of control responses. Power consumption of approximately 150mA at 5V demonstrates efficient operation suitable for continuous monitoring applications. The system maintained stable operation during 48-hour continuous testing without failures or anomalous behavior.

The development process successfully employed modern embedded systems design methodologies, utilizing simulation for pre-implementation validation, structured programming practices for maintainable code, and modular hardware design for straightforward assembly and testing.

6.2 Conclusion and Implications

This project demonstrates the practical feasibility of implementing an intelligent water level monitoring and control system using readily available, cost-effective components. The successful integration of the PIC16F877A microcontroller, ultrasonic sensor, and various peripheral devices proves that sophisticated automation capabilities are accessible for educational projects and small-scale practical applications.

Practical Implications:

The system addresses real-world needs in water resource management, providing a solution applicable to residential water tanks, small agricultural applications, aquaculture systems, and various industrial processes requiring liquid level monitoring. The automated control capability reduces human workload, prevents overflow conditions that waste water, and avoids low-level conditions that might damage pumps or interrupt water supply.

The cost-effectiveness of the solution makes it accessible to a wide user base, from individual homeowners to small businesses. The use of standard components ensures maintainability and the possibility of local repair, avoiding dependence on proprietary systems or specialized service providers.

Educational Value:

From an educational perspective, this project provides comprehensive exposure to embedded systems development practices. The project encompasses sensor interfacing, actuator control, human-machine interface design, real-time processing, and system integration. Students completing similar projects gain practical experience in circuit design, microcontroller programming, PCB development, simulation tools, and systematic testing methodologies.

Contribution to Water Conservation:

In the broader context of environmental sustainability, automated water management systems contribute to conservation efforts by eliminating waste from overflow conditions and ensuring optimal usage. The real-time monitoring capability enables users to detect leaks or unusual consumption patterns, supporting proactive maintenance and resource management.

The system's design philosophy of providing graduated responses rather than simple on-off control represents a more sophisticated approach to automation that can be applied to various resource management applications beyond water level monitoring.

6.3 Recommendations for Future Work

Several enhancements and expansions could extend the system's capabilities and applicability:

1. Variable Tank Height Configuration:

Implementing a configuration interface, potentially through a keypad or touch buttons, would enable users to set different tank dimensions without reprogramming. This enhancement would make the system adaptable to various installations without requiring technical expertise for code modification. The configuration could be stored in EEPROM for persistence across power cycles.

2. IoT Integration and Remote Monitoring:

Integrating wireless connectivity modules (ESP8266, ESP32, or similar) would transform the system into an IoT device capable of remote monitoring and control. Features could include:

- Real-time data transmission to cloud platforms or mobile applications
- Remote notification of critical conditions via push notifications, SMS, or email
- Historical data visualization showing water level trends and usage patterns
- Remote system configuration and threshold adjustment
- Integration with home automation systems

3. Wireless Ultrasonic Sensor Technology:

Investigating wireless ultrasonic sensor modules would eliminate physical wiring between the sensor and control unit. This enhancement would simplify installation, particularly for applications where the control panel must be located at a distance from the tank. Wireless communication could employ protocols such as Zigbee, LoRa, or proprietary RF links.

4. Advanced Data Analytics:

Implementing data logging with SD card storage or cloud database integration would enable sophisticated analytics:

- Water consumption pattern analysis to detect anomalies
- Predictive maintenance alerts based on pump operation cycles
- Leak detection through unexpected level changes
- Usage forecasting to optimize filling schedules
- Generation of consumption reports for billing or resource planning

5. Multiple Tank Management:

Expanding the system to monitor and control multiple tanks would increase applicability for:

- Buildings with multiple water storage tanks
- Water distribution systems with primary and secondary storage
- Comparative monitoring of input and output tanks for leak detection
- Sequential filling systems that prioritize tanks based on usage patterns

6. Enhanced Power Management:

Implementing advanced power management features would improve system reliability:

- Battery backup system with automatic switchover during power failures
- Low-power sleep modes during stable conditions
- Solar power compatibility for remote installations
- Power failure notification and automatic system recovery

7. Improved Sensor Technology:

Exploring alternative or complementary sensing technologies could enhance accuracy and reliability:

- Multiple ultrasonic sensors for redundancy and cross-validation
- Pressure sensors for supplementary measurement validation
- Turbidity sensors to detect water quality issues
- Temperature sensors for environmental compensation

8. Machine Learning Integration:

Implementing machine learning algorithms could enable predictive capabilities:

- Learning usage patterns to anticipate filling needs
- Adaptive threshold adjustment based on historical data
- Anomaly detection for leak identification
- Optimization of pump operation for energy efficiency

9. Enhanced Safety Features:

Additional safety mechanisms would improve system robustness:

- Pump dry-run protection with multiple verification methods
- Overflow detection with redundant sensors
- Motor overload protection and thermal monitoring
- Emergency shutoff capability with manual override
- Diagnostic modes for system health checking

10. User Interface Improvements:

Upgrading the user interface would enhance usability:

- Graphical LCD display showing level trends and graphs
- Touch screen interface for intuitive configuration
- Voice notification capability for accessibility
- Multi-language support for international deployment
- Mobile application for comprehensive monitoring and control

6.4 Reflection on Challenges and Lessons Learned

The development process encountered several challenges that provided valuable learning experiences:

LCD Interface Challenges:

Initial attempts to interface with the LCD display resulted in a completely black screen, caused by incorrect initialization timing sequences. This issue required careful review of the LCD controller datasheet and understanding of the specific timing requirements for 4-bit mode operation. The resolution involved implementing proper delay functions and ensuring correct initialization sequence execution. This experience reinforced the importance of thorough datasheet study and understanding peripheral timing requirements.

KiCad PCB Design Issues:

During PCB layout design in KiCad, path errors occurred when attempting to route connections between components. These errors stemmed from improper footprint associations and net connectivity definitions in the schematic. Resolving these issues required learning proper schematic annotation, netlist generation, and footprint assignment procedures. The experience demonstrated the importance of careful schematic preparation before proceeding to layout design.

Ultrasonic Sensor Anomalies:

The ultrasonic sensor occasionally displayed negative distance values when measuring at maximum range or when no valid target was present. This behavior resulted from overflow conditions in the timing calculation or receipt of invalid echo signals. Implementation of input validation, bounds checking, and timeout mechanisms resolved these issues. The experience highlighted the necessity of defensive programming practices and robust error handling in embedded systems.

Environmental Factor Effects:

Testing revealed that environmental factors, particularly high humidity and temperature variations, could affect sensor accuracy. Ambient humidity occasionally caused condensation on the sensor transducers, temporarily degrading performance. Temperature changes affected the speed of sound, introducing measurement errors. These observations emphasized the importance of environmental considerations in sensor system design and the potential value of compensation mechanisms.

Keypad Integration Difficulties:

Initial attempts to integrate a configuration keypad encountered software timing conflicts with LCD updates and sensor reading cycles. Improper interrupt configuration and shared resource conflicts caused erratic behavior. Resolution required careful restructuring of the main program loop and implementation of proper debouncing techniques. This challenge provided valuable experience in managing multiple peripherals with different timing requirements.

Power Supply Noise Issues:

Early testing revealed occasional system resets and erratic behavior caused by power supply noise, particularly during relay switching. The motor's inductive load generated voltage spikes that coupled into the digital circuitry. Adding decoupling capacitors near the microcontroller and implementing a snubber circuit across the relay contacts resolved these issues. This experience emphasized the importance of proper power supply design and noise mitigation techniques.

Software Debugging Challenges:

Debugging embedded systems without sophisticated debugging tools required creative approaches. Implementing diagnostic LED patterns and UART-based debugging messages proved invaluable for troubleshooting. Learning to use MPLAB X's simulator and Proteus virtual instrumentation improved debugging efficiency. This experience highlighted the value of planning for debuggability during initial design phases.

Lessons Learned:

1. **Importance of Simulation:** Proteus simulation significantly reduced hardware debugging time by identifying design issues before physical implementation. Future projects should allocate adequate time for thorough simulation testing.
2. **Modular Development:** Developing and testing components individually before integration simplified troubleshooting and accelerated development. This modular approach should be standard practice for complex systems.
3. **Documentation Value:** Maintaining detailed documentation of pin assignments, register configurations, and design decisions proved invaluable during debugging and modification. Comprehensive documentation should be created concurrently with development rather than retroactively.
4. **Error Handling Necessity:** Robust error handling and input validation are not optional luxuries but essential requirements for reliable embedded systems operation.
5. **Testing Thoroughness:** Systematic testing across the full operational range and under various conditions revealed issues that might not appear during casual testing. Comprehensive test planning is essential.
6. **Community Resources:** Online communities, forums, and documentation repositories provided valuable assistance during troubleshooting. Engaging with the embedded systems community enhances learning and problem-solving capabilities.

7. References

Microchip Technology Inc., "PIC16F87XA Data Sheet: 28/40/44-Pin Enhanced Flash Microcontrollers," DS39582E, 2003.

HC-SR04 Ultrasonic Sensor Datasheet, Elec Freaks, 2013.

Hitachi, "HD44780U (LCD-II) Dot Matrix Liquid Crystal Display Controller/Driver," ADE-207-272(Z) '99.9 Rev. 0.0.

M. A. Mazidi, R. D. McKinlay, and D. Causey, "PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18," Pearson Education, 2008.

J. W. Valvano, "Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers," 2nd Edition, 2012.

N. Doerry, "Designing Electrical Systems in Accordance with IEEE 45," Naval Engineers Journal, vol. 117, no. 4, pp. 61-71, 2005.

A. Kumar and S. Singh, "Water Level Indicator and Controller Using Ultrasonic Sensor," International Journal of Engineering Research and Technology (IJERT), vol. 3, no. 5, 2014.

R. Patel and M. Patel, "Automatic Water Level Controller Using Ultrasonic Sensor and GSM," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 2, no. 12, pp. 3179-3181, 2013.

Proteus Design Suite Documentation, Labcenter Electronics Ltd., 2020.

KiCad EDA Documentation, KiCad Development Team, 2021.

MPLAB X IDE User's Guide, Microchip Technology Inc., Document DS-52027D, 2018.

XC8 C Compiler User's Guide for PIC, Microchip Technology Inc., Document DS-50002053H, 2019.

S. F. Barrett and D. J. Pack, "Embedded Systems Design with the Atmel AVR Microcontroller: Part I," Synthesis Lectures on Digital Circuits and Systems, Morgan & Claypool Publishers, 2009.

Standard for Information Technology, "IEEE 829-2008 - Standard for Software and System Test Documentation," 2008.

P. Marwedel, "Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems," 2nd Edition, Springer, 2011.

8. Appendices

Appendix A: Complete Source Code

/* config.h file

* File: Smart Water Level Indicator with Motor Control

```
* Author: MALINDU KANISHKA

* Comments:

* Revision history:

*/

// This is a guard condition so that contents of this file are not included

// more than once.

#ifndef XC_HEADER_TEMPLATE_H

#define XC_HEADER_TEMPLATE_H

#include <xc.h> // include processor files — each processor file is guarded.

#include <pic.h>

#define _XTAL_FREQ 20000000

#define RS RD2

#define EN RD3

#define D4 RD4

#define D5 RD5
```

```
#define D6 RD6

#define D7 RD7

#define Trigger RB1 //34 is Trigger

#define Echo RB2//35 is Echo

//#define PWM_PIN RB7 // PWM output pin

#pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)

#pragma config PWRTE = ON // Power-up Timer Enable bit (PWRT enabled)

#pragma config BOREN = ON // Brown-out Reset Enable bit (BOR enabled)

#pragma config LVP = OFF // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit
(RB3 is digital I/O, HV on MCLR must be used for programming)

#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit (Data EEPROM code
protection off)

#pragma config WRT = OFF // Flash Program Memory Write Enable bits (Write protection off; all
program memory may be written to by EECON control)

#pragma config CP = OFF // Flash Program Memory Code Protection bit (Code protection off)

// TODO Insert appropriate #include <>
```

```
// TODO Insert C++ class definitions if appropriate

// TODO Insert declarations

// Comment a function and leverage automatic documentation with slash star star

/**

<p><b>Function prototype:</b></p>

<p><b>Summary:</b></p>

<p><b>Description:</b></p>

<p><b>Precondition:</b></p>

<p><b>Parameters:</b></p>

<p><b>Returns:</b></p>

<p><b>Example:</b></p>

<code>

</code>

<p><b>Remarks:</b></p>

*/
```

```
// TODO Insert declarations or function prototypes (right here) to leverage

// live documentation

#ifdef __cplusplus

extern “C” {

#endif /* __cplusplus */

// TODO If C++ is being used, regular C code needs function names to have C

// linkage so the functions can be used by the c code.

#ifdef __cplusplus

}

#endif /* __cplusplus */

#endif /* XC_HEADER_TEMPLATE_H */
```

main.c file code

```
#include “config.h”

#include <stdio.h>

//LCD Functions
```

```
void Lcd_SetBit(char data_bit) //Based on the Hex value Set the Bits of the Data Lines

{

if(data_bit& 1)

D4 = 1;

else

D4 = 0;

if(data_bit& 2)

D5 = 1;

else

D5 = 0;

if(data_bit& 4)

D6 = 1;

else

D6 = 0;

if(data_bit& 8)
```

```
D7 = 1;

else

D7 = 0;

}

void Lcd_Cmd(char a)

{

RS = 0;

Lcd_SetBit(a); //Incoming Hex value

EN = 1;

__delay_ms(4);

EN = 0;

}

void Lcd_Clear()

{

Lcd_Cmd(0); //Clear the LCD
```

```
Lcd_Cmd(1); //Move the curser to first position

}

void Lcd_Set_Cursor(char a, char b)

{

char temp,z,y;

if(a== 1)

{

temp = 0x80 + b — 1; //80H is used to move the curser

z = temp>>4; //Lower 8-bits

y = temp & 0x0F; //Upper 8-bits

Lcd_Cmd(z); //Set Row

Lcd_Cmd(y); //Set Column

}

else if(a== 2)

{
```

```
temp = 0xC0 + b — 1;

z = temp>>4; //Lower 8-bits

y = temp & 0x0F; //Upper 8-bits

Lcd_Cmd(z); //Set Row

Lcd_Cmd(y); //Set Column

}

}

void Lcd_Start()

{

Lcd_SetBit(0x00);

for(int i=1065244; i<=0; i — ) NOP();

Lcd_Cmd(0x03);

__delay_ms(5);

Lcd_Cmd(0x03);

__delay_ms(11);
```

```
Lcd_Cmd(0x03);

Lcd_Cmd(0x02); //02H is used for Return home -> Clears the RAM and initializes the LCD

Lcd_Cmd(0x02); //02H is used for Return home -> Clears the RAM and initializes the LCD

Lcd_Cmd(0x08); //Select Row 1

Lcd_Cmd(0x00); //Clear Row 1 Display

Lcd_Cmd(0x0C); //Select Row 2

Lcd_Cmd(0x00); //Clear Row 2 Display

Lcd_Cmd(0x06);

}

void Lcd_Print_Char(char data) //Send 8-bits through 4-bit mode

{

char Lower_Nibble,Upper_Nibble;

Lower_Nibble = data&0x0F;

Upper_Nibble = data&0xF0;

RS = 1; // => RS = 1
```

```
Lcd_SetBit(Upper_Nibble>>4); //Send upper half by shifting by 4

EN = 1;

for(int i=2130483; i<=0; i — ) NOP();

EN = 0;

Lcd_SetBit(Lower_Nibble); //Send Lower half

EN = 1;

for(int i=2130483; i<=0; i — ) NOP();

EN = 0;

}

void Lcd_Print_String(char *a)

{

int i;

for(i=0;a[i]!=’\0';i++)

Lcd_Print_Char(a[i]); //Split the string using pointers and call the Char function

}
```

```
void Lcd_Print_Int(int value) {

char buffer[10];

sprintf(buffer, “%d”, value);

Lcd_Print_String(buffer);

}

/End of LCD Functions/

int time_taken;

int distance;

int main() {

TRISD = 0x00; //PORTD declared as output for interfacing LCD

TRISB0 = 1; //DEfine the RB0 pin as input to use as interrupt pin

TRISB1 = 0; //Trigger pin of US sensor is sent as output pin

TRISB2 = 1; // Echo pin of US sensor is set as an input pin

TRISB3 = 0; //RB3 is output pin for LED

TRISB4 = 0; //RB4 is output pin for LED
```

```
TRISB5 = 0; //RB5 is output pin for LED

TRISB6 = 0; //buzzer

TRISB7 = 0; //motor

T1CON=0x20;

Lcd_Start();

Lcd_Set_Cursor(1,1);

Lcd_Print_String("US sensor");

Lcd_Set_Cursor(2,1);

Lcd_Print_String("with PIC16F877A");

__delay_ms(2000);

Lcd_Clear();

while(1) {

TMR1H =0; TMR1L =0; //clear the timer bits

Trigger = 0;

__delay_us(15);
```

```
Trigger = 1;

__delay_us(15);

Trigger = 0;

while (Echo == 0);

TMR1ON = 1;

while (Echo == 1);

TMR1ON = 0;

time_taken = (TMR1L | (TMR1H << 8));

distance = (0.0343 * time_taken) / 2;

time_taken = time_taken * 0.8; //0.8 is a correction fector

Lcd_Set_Cursor(1, 1);

Lcd_Print_String("Time:");

Lcd_Print_Int(time_taken);

Lcd_Print_String("us");

Lcd_Set_Cursor(2, 1);
```

```
Lcd_Print_String("Distance:");

Lcd_Print_Int(distance);

Lcd_Print_String("cm");

// Turn on LED and buzzer based on distance

if (distance <= 5) {

RB3 = 0; // Turn off RB3 LED

RB4 = 0; // Turn off RB4 LED

RB5 = 1; // Turn on RB5 LED

RB6 = 0; // Turn on buzzer

RB7 = 0;

} else if (distance <= 15) {

RB3 = 0; // Turn off RB3 LED

RB4 = 1; // Turn on RB4 LED

RB5 = 0; // Turn off RB5 LED

RB6 = 0; // Turn off buzzer
```

```
RB7 = 1;

} else if (distance <= 30) {

RB3 = 1; // Turn on RB3 LED

RB4 = 0; // Turn off RB4 LED

RB5 = 0; // Turn off RB5 LED

RB6 = 1; // Turn off buzzer

__delay_ms(1000); // Buzzer on for 1 second

RB6 = 0; // Turn off buzzer

RB7 = 1;

} else {

// If distance is greater than 40 cm, turn off all LEDs and buzzer

RB3 = 0;

RB4 = 0;

RB5 = 0;

RB6 = 0;
```

```
RB7 = 0;

}

}

return 0;

}
```