Princess Sumaya University for Technology
King Abdullah II School for Engineering
Computer Engineering Department

22442 Embedded Systems

Smart Water Tank

Dima Sawalmeh 20200296
Fares Alabdullat 20200256
Mohammad Ali 20200829

Instructor

Prof. Belal Sababha

Spring semester 2024

Contents

Abstract

The Smart Water Tank project utilizes the PIC 16F877A microcontroller to create an automated water management system designed to optimize water usage and prevent wastage. The system incorporates a set of ultrasonic sensors to continuously monitor the water level in the tank. The data collected by the sensors is processed by the PIC 16F877A, which controls a relay-driven pump mechanism to maintain the desired water level. Additionally, the system includes an LCD display to provide real-time information about the tank's water level and status indicators for easy monitoring.

This project demonstrates the practical application of embedded systems in solving real-world problems by providing an efficient, reliable, and cost-effective solution for water management. The smart water tank system not only prevents overflow and dry run conditions but also promotes sustainable water usage by ensuring that water is used judiciously. Through this project, the potential of microcontroller-based solutions in enhancing the functionality and intelligence of everyday systems is showcased, highlighting the benefits of integrating technology into resource management practices.

Introduction

In today's world, efficient water management is a critical concern due to the increasing scarcity of water resources and the growing demand for sustainable solutions. Traditional water tank systems often suffer from issues such as overflow, dry runs, and inefficient water usage, leading to water wastage and potential damage to the infrastructure. To address these challenges, smart water tank systems have emerged as a viable solution, leveraging modern technology to optimize water usage and management.

This project focuses on the development of a smart water tank system using the PIC 16F877A microcontroller. The PIC 16F877A, a popular choice in embedded systems, offers robust performance and versatile interfacing capabilities, making it ideal for this application. The smart water tank system aims to automate the monitoring and control of water levels, ensuring efficient water utilization and minimizing wastage. By integrating various sensors and control mechanisms, the system can provide real-time feedback and take proactive measures to maintain optimal water levels, thereby enhancing overall efficiency and reliability.

Project descriotion and components

To design our embedded system we used the following components:

1. PIC16F877A

Microchip Technology's PIC16F877A is an 8-bit microcontroller that is well-known for its extensive application in embedded systems and electronic applications. With 35 single-word instructions, its RISC (Reduced Instruction Set Computing) Harvard architecture offers effective performance. The microcontroller has 368 bytes of RAM, 256 bytes of EEPROM for data retention, and 14 KB of Flash memory for program storage. It uses frequencies as high as 20 MHz and provides 33 I/O pins for flexible connectivity. Three timers (Timer0, Timer1, and Timer2), an eight-channel, 10-bit Analog-to-Digital Converter (ADC), and support for the USART, SPI, and I2C communication protocols are some of the important features. Its usefulness is increased by extra features like watchdog timing, PWM outputs, and in-circuit debugging capabilities. Community support makes it ideal for applications in automation, robotics, and industrial control systems.

2. sharp IR distance sensor (20-80 c)

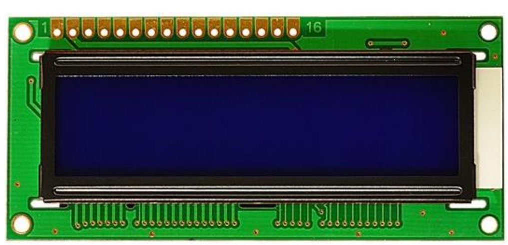
The Sharp IR Distance Sensor (GP2Y0A21YK0F) is a multipurpose part that uses infrared light to measure distances between 20 and 80 cm. Its analog voltage output matches the distance that is sensed, therefore integrating it with different microcontrollers is simple. This sensor, which normally operates between 4.5 and 5.5 volts, is renowned for its excellent accuracy and dependability in measuring distances, as well as its low susceptibility to outside influence. It is the perfect option for integration into a variety of applications, including robots, proximity sensing projects, and obstacle avoidance systems, due to its small size and low power consumption.

3. Buzzer

An audio signaling device known as a buzzer is utilized in electronic applications to provide sound alerts or alarms. It uses mechanical, electromagnetic, or piezoelectric methods to transform electrical signals into sound. The buzzer can be interfaced with a PIC microcontroller by connecting it to one of the microcontroller's digital output pins, which enables the microcontroller to regulate the buzzer's function. The buzzer can be turned on or off by the PIC by flipping this output pin high or low, producing sound as desired. With this configuration, the microcontroller may respond to different inputs or conditions by producing auditory feedback or alerts, which makes it a crucial part of applications like timers, alarms, and interactive user interfaces. The simplicity of interfacing a buzzer with a PIC microcontroller, combined with the PIC's programmable capabilities, facilitates the easy integration of auditory signals into a wide range of projects.

4. 16×2 LCD

Popular alphanumeric display modules, such as the 16x2 LCD, are perfect for showing text and simple symbols in electronic projects since they can display up to 16 characters per line on two lines. Usually, a parallel interface with numerous data and control lines is used to interface an LCD with a PIC microcontroller. The LCD receives data and commands from the PIC to regulate what is shown. With this configuration, the microcontroller can show data in real time, including user prompts, sensor readings, and status messages. The PIC can effectively control the LCD through the use of libraries or custom code, which makes it an adaptable and user-friendly output device for a range of uses, such as educational projects, diagnostic displays, and user interfaces.

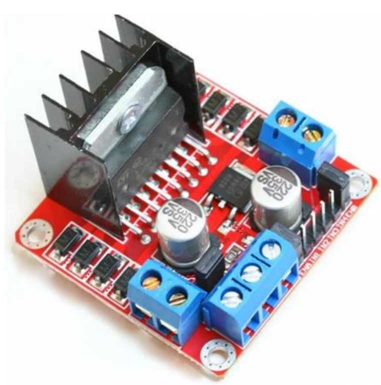


5. DC motor and water pump

An electric motor that operates on direct current (DC) power is known as a DC motor. By use of the interaction of magnetic fields, electrical energy is transformed into mechanical energy. The rotor, or revolving part, and the stator, or stationary part, make up the motor. The voltage applied to the motor can be changed to adjust its speed and torque. DC motors are widely employed in many different applications, including robotics, electric cars, and home appliances.
A mechanical tool used to transfer water from one place to another is called a water pump. Water is forced into pipes or hoses by means of a pressure differential that is created. Water pumps can be run manually, with the use of engines, electric motors, or both. They are extensively utilized in industrial processes, water delivery systems, and agricultural irrigation. The water pump can function effectively on a direct current power supply when paired with a DC motor, which makes it appropriate for off-grid or isolated areas.

6. H-bridge

An electronic circuit known as an H-bridge allows voltage to be delivered in either direction across a load, such a motor. This arrangement is called so because, when seen schematically, it resembles the letter "H". The four switches (transistors or relays) that control the load's current flow direction usually make up this configuration. The H-bridge is capable of causing the motor to spin clockwise, counterclockwise, or to halt by turning on particular pairs of switches. Because H-bridges allow a motor to be reversed, they are vital parts of electric vehicles, robots, and other automated systems.



7. Crystal oscillator 8MHZ

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal, typically quartz, to create an electrical signal with a precise frequency. An 8MHz crystal oscillator generates a steady 8 million cycles per second signal, which is used to keep time in digital watches, provide a clock signal for microcontrollers, and stabilize frequencies for radio transmitters and receivers. The crystal ensures high stability and accuracy, making it ideal for applications requiring precise timing. The 8MHz crystal oscillator is a common component in various electronic devices and systems due to its reliability and precision.

8. potentiometer 10k ohm

A potentiometer is a three-terminal resistor that can be used to create an adjustable voltage divider. It has a sliding or revolving contact. The overall resistance of a 10k ohm potentiometer is 10,000 ohms. You may precisely control the voltage and current in a circuit by adjusting the resistance between the wiper, which is the movable middle terminal, and the two fixed terminals by rotating the knob or sliding the contact. Potentiometers are therefore perfect for regulating electrical devices, such microcontroller projects, circuit tuning, and modifying the brightness or contrast of an LCD panel. Its maximum resistance is indicated by the 10k ohm rating, which makes it adaptable to a wide range of electrical applications.

9. LEDs

A semiconductor device known as a Light Emitting Diode (LED) releases light when an electric current flows through it. LEDs are recognized for their superior performance over conventional incandescent lamps, extended longevity, and low power consumption. They are widely utilized in applications ranging from large-scale displays and lighting solutions to indicator lights on electronic gadgets. They are available in a variety of colors and sizes. LEDs function by allowing electrons to recombine with holes inside the device, which releases photons, which are units of energy. They are widely used in contemporary electronics and lighting systems due to their robustness and adaptability.

The design of our Smart water tank is based on two parts:

- Software design
- Harware design

The MPLAB X IDE, a complete development environment from Microchip for PIC microcontrollers, was used to complete the software design. The XC8 compiler, which provides a wealth of tools for effective microcontroller programming, was used to program the C code. The development process is aided by the comprehensive debugging tools and intuitive interface of the MPLAB X IDE. Built-in libraries for peripheral device control, such as those for LCD screens, sensors, and communication interfaces, were used in our project. In particular, we used a number of utility libraries for data manipulation and sensor integration, including the LCD library to manage the 16x2 LCD screen. The primary function ensures that the system works for as long as it is powered by running a continuous loop.
We also used the Proteus Design Suite for testing, simulation, and circuit design. We were able to digitally build and test our microcontroller systems and electronic circuits with Proteus, which gave us a useful platform to confirm the design before putting it on actual hardware. Our project's dependability and productivity were increased by the MPLAB X IDE and Proteus Design Suite working together to guarantee a seamless software development process from software design to virtual testing.

MikroC code:

```
sbit LCD_RS at RC2_bit;
sbit LCD_EN at RC3_bit;
sbit LCD_D4 at RC4_bit;
sbit LCD_D5 at RC5_bit;
sbit LCD_D6 at RC6_bit;
sbit LCD_D7 at RC7_bit;
sbit LCD_RS_Direction at TRISC2_bit;
sbit LCD_EN_Direction at TRISC3_bit;
sbit LCD_D4_Direction at TRISC4_bit;
sbit LCD_D5_Direction at TRISC5_bit;
sbit LCD_D6_Direction at TRISC6_bit;
sbit LCD_D7_Direction at TRISC7_bit;
char txts1[5];
char txts2[5];
int sen1;
int sen2;
float v;
float v2;
int p1;
int pp1;
int p2;
int pp2;
char asc[5];
```

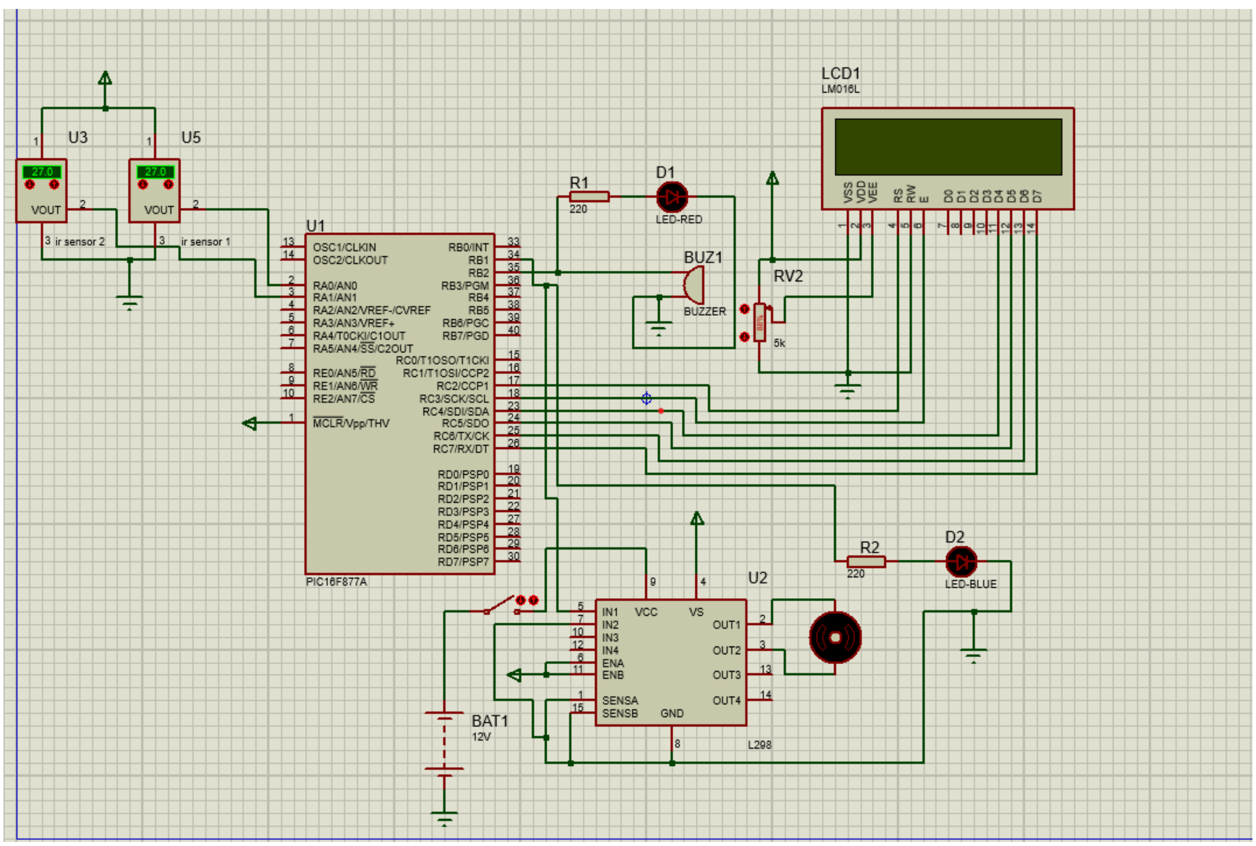
```
char asc2[5];
int fir;
int sec;
int th;
void msDelay (unsigned int mscnt);


void main()
{
 Lcd_Init(); // Initialize LCD
 Lcd_Cmd(_LCD_CLEAR); // Clear display
 Lcd_Cmd(_LCD_CURSOR_OFF);


 TRISA.F0=1;

 TRISB = 0;
 ADCON1 = 0xC0;

 //msDelay(5000);
 while(1){


 ADCON0 = 0x81;       ///ch0 ,stop adc,module on
 msDelay(50);
 ADCON0 = 0x85;
 while(ADCON0.F2);
 sen1=((ADRESH<<8)|(ADRESL));
 msDelay(50);
 ADCON0 = 0x81; /////same same just stop adc
 v = sen1 * 0.48;

 if(v<=120){
  PORTB.F1 = 1;

 }
if(v>=121)
{PORTB.F1 = 0;}
  pp1 = (int)v;
 p1 = (pp1 - 80)*(100 - 0)/(230-80) + 0;
 if(v>=230){
 p1 = 100;

    }

  if(v<=80){
```

```
 p1 = 0;
 }

asc[2] = (p1 % 10 ) + 48 ;    ///////////////////third digit
asc[1] = (p1 % 100) / 10 + 48 ;
asc[0] = (p1 % 1000) / 100 + 48;
if(asc[0] == 48){asc[0] = 32;};
Lcd_Out(1,1,"T1= ");
Lcd_Out(1,4,asc);
Lcd_Out(1,7,"%");
 if(v<=120){
 Lcd_Out(1,8,"LOW LEVEL");
 }

 if(v>=121){  Lcd_Out(1,8,"NORMAL    ");}
  msDelay(500);
///////////////////////////////////////////////////////////////////////second sensor

ADCON0 = 0x89;
 msDelay(50);
 ADCON0 = 0x8D;

 while(ADCON0.F2);
 sen2=((ADRESH<<8)|(ADRESL));
 msDelay(50);
 ADCON0 = 0x89; /////same same just stop adc
 v2 = sen2 * 0.48;

 if(v2<=105){
  PORTB.F2 = 1;
 }
if(v2>=106)
{PORTB.F2 = 0;}
  pp2 = (int)v2;
 p2 = (pp2 - 80)*(100 - 0)/(230-80) + 0;
 if(v2>=230){
 p2 = 100;

    }

  if(v2<=80){
 p2 = 0;
 }

asc2[2] = (p2 % 10 ) + 48 ;    ///////////////////third digit
asc2[1] = (p2 % 100) / 10 + 48 ;
```

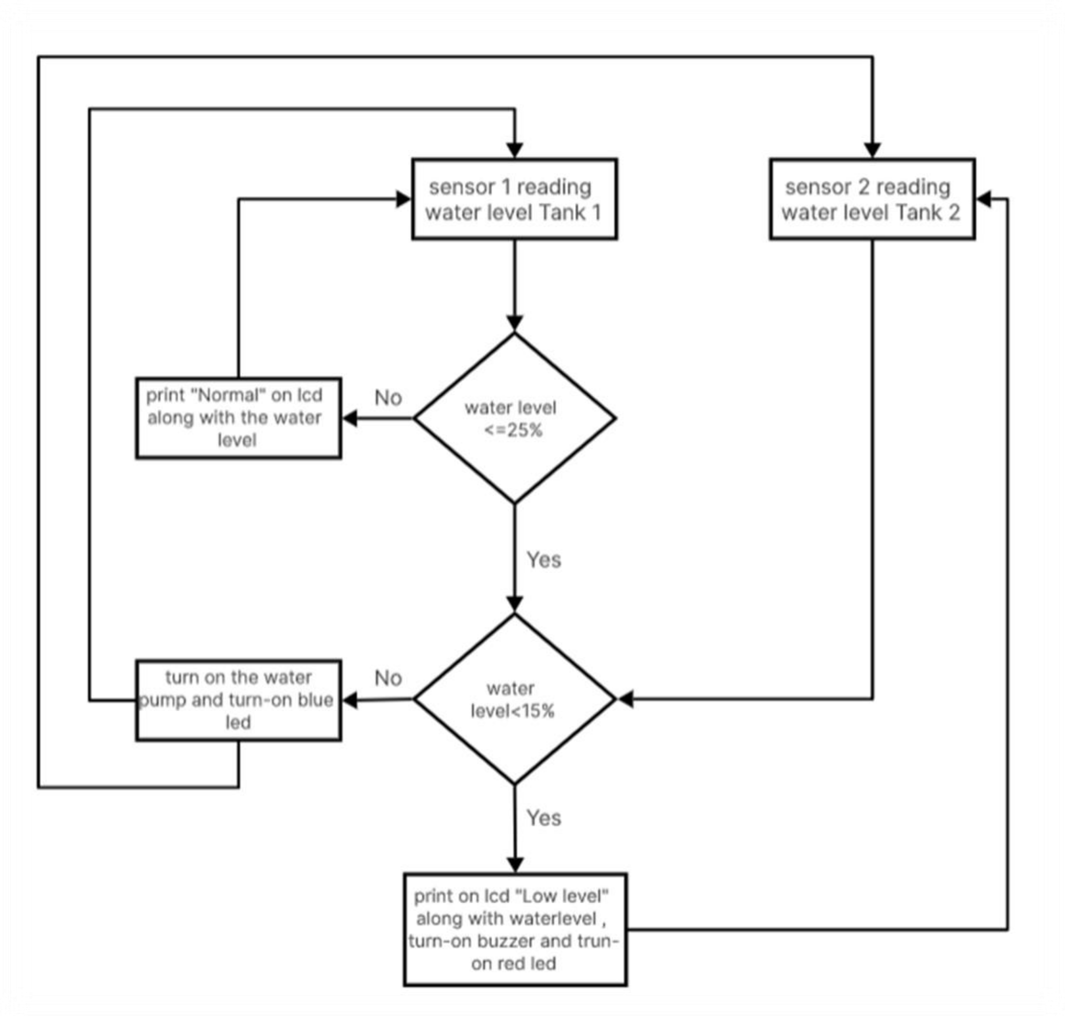
```
asc2[0] = (p2 % 1000) / 100 + 48;
if(asc2[0] == 48){asc2[0] = 32;};
Lcd_Out(2,1,"T2= ");
Lcd_Out(2,4,asc2);
Lcd_Out(2,7,"%");
 if(v2<=105){
 Lcd_Out(2,8,"LOW LEVEL");
 }

 if(v2>=106){  Lcd_Out(2,8,"NORMAL    ");}
  msDelay(500);







 //txts2 = p1 + "49";
 //intToStr(p1,txts1);
// Lcd_Out(1,1,"T1=");
// Lcd_Out(1,4,txts1);
msDelay(100);

 }

}

void msDelay(unsigned int mscnt) {
        unsigned int ms;
        unsigned int cnt;
        for (ms = 0; ms < mscnt; ms++) {
                for (cnt = 0; cnt < 155; cnt++);//1ms
        }
}
```

Hardware design on proteus:

In order to implement our recommended design, which determines whether water should be pumped based on the parameters set using the microcontroller, the hardware design compromises IR distance sensors to measure the water level. Based on the sensor's measurement if the water level in tank 1 is less than or equal to 25% aside with the water level in tank 2 is normal the water pump will start working along with blue light turned on and tank 1 will be filled until it reaches 100% or until tank 2 reach less than or equal to 15%. If tank two at anytime reached below 15% the buzzer and red light will be turned on. To provide power to the pic and the other components we used a 12v battery to mainly supply our dc motor and water pump and connected it to a buck converter (power regulator) to supply the microcontroller with 5v only as it can't receive high voltages, or it will burn. The main idea of this project is to ease the process of constantly checking the water level on our tanks and to minimize any danger a human could face through that process, so everything is displayed through a 16×2 LCD screen implemented at home or any desired place along with a buzzer and red light to alert the costumer that the main water tank is empty so they can call for a refill instantly, and then when refilling while the pump is on a blue light will be on along with the LCD displaying the water level on both tanks.

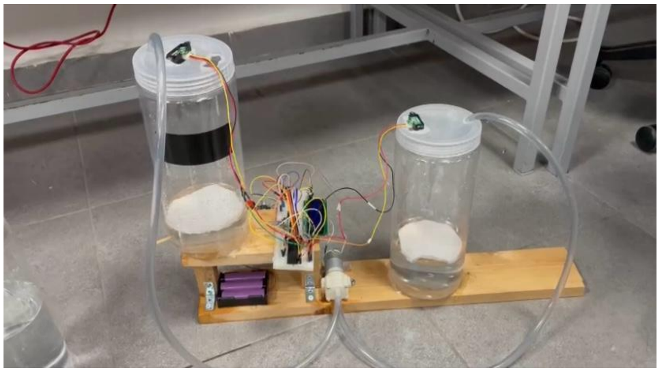


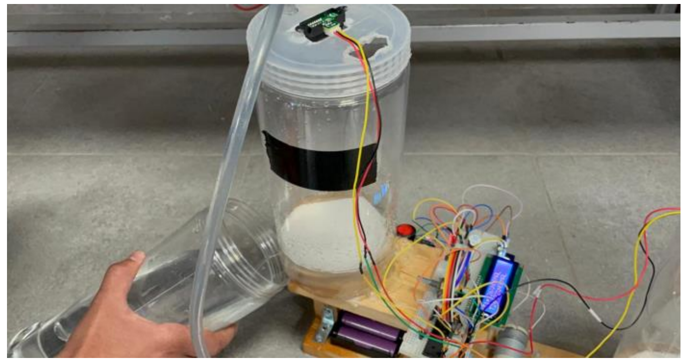
Flow chart

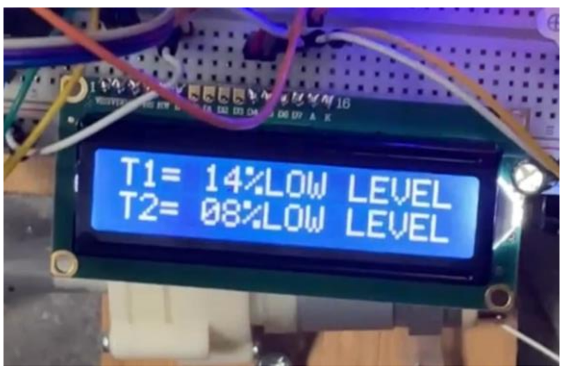


sensor 1 reading water level Tank 1
sensor 2 reading water level Tank 2
print "Normal" on lcd along with the water level
No
water level <=25%
Yes
turn on the water pump and turn-on blue led
No
water level<15%
Yes
print on lcd "Low level" along with waterlevel , turn-on buzzer and trun-on red led

Prototype






T1= 14%LOW LEVEL
T2= 08%LOW LEVEL


T1= 11%LOW LEVEL
T2= 18%NORMAL


T1= 34%NORMAL
T2= 32%NORMAL

Problems and recommendations

During the tank project, we encountered several challenges that required creative solutions and persistence. Initially, we faced difficulties in creating the delay function, but by following the instructor's guidelines, we managed to resolve this issue. Another challenge arose when displaying results on the LCD; converting integers to strings successfully addressed this problem. We also experienced issues with wire connections, which were traced back to a faulty breadboard. To unload tank 1, we innovatively created a small hole, which proved effective. A significant setback occurred when we incorrectly connected the wires to the PIC, resulting in a burnt PIC. However, our most persistent problem was with the LCD, which we attempted to fix over 20 times, only to discover that the LCD itself was broken. Based on these experiences, we recommend thorough testing of components before use, careful adherence to wiring diagrams, and having spare parts on hand to mitigate similar issues in future projects.

Conclusion

Our tank project was a journey of learning and problem-solving. We faced several challenges, from creating delay functions and displaying results on the LCD to dealing with faulty breadboards and burnt PICs. Despite these hurdles, we managed to build a functional system that monitors and controls water levels in two tanks.

Using the PIC 16F877A microcontroller, we read sensor values, converted them to meaningful data, and displayed the information on an LCD. The system effectively controls water levels, ensuring they stay within desired limits. This project highlighted the importance of thorough testing, careful assembly, and having backup plans.

In the end, we not only created a useful water management system but also gained valuable insights into the practical applications of embedded systems. This experience has equipped us with the knowledge to tackle similar projects in the future, making technology work for us in everyday life.