

## **AUTORES**

Tadeu Henrique Oliveira Ladeia Costa, Lytton Alves Pereira Souza.

## **Desenvolvimento de um Sistema Microprocessado para Controle Automático de Reservatórios de Água**

**Trabalho apresentado para avaliação na disciplina de Sistema Microprocessados do Curso de Engenharia da computação do Centro Universitário FipMoc - Afya de Montes Claros.**

**Prof. Alexandre Dantas Dias**

**MONTES CLAROS**

**2024**

***AGRADECIMENTOS***

*- A Deus pela vida e saúde.*

## SUMÁRIO

<b>INTRODUÇÃO</b>	4
<b>1 CONTEXTUALIZAÇÃO</b>	5
1.1 JUSTIFICATIVA	5
1.2 OBJETIVOS	5
<b>2 REVISÃO DE LITERATURA</b>	5
<b>3 METODOLOGIA</b>	6
3.1 ESTRUTURA SISTEMA	6
3.2 FLUXO DE OPERAÇÃO	6
3.3 CRITÉRIOS DE SUCESSO	6
<b>4 DESENVOLVIMENTO</b>	6
<b>5 RESULTADOS E DISCUSSÃO</b>	11
<b>6 CONCLUSÃO OU CONSIDERAÇÕES FINAIS REFERÊNCIAS</b>	16

## INTRODUÇÃO

Com o avanço das tecnologias de automação e microprocessamento, a implementação de sistemas inteligentes em processos de controle se tornou essencial em diversas áreas, como a gestão de recursos hídricos. Este artigo apresenta o desenvolvimento de um sistema microprocessado para o controle automatizado de níveis de água em reservatórios subterrâneos e superiores. A proposta abrange a integração de hardware e software para atender às demandas de eficiência e segurança, explorando o potencial de sensores, atuadores e microcontroladores para automação.

## **1. CONTEXTUALIZAÇÃO**

### **1.1 JUSTIFICATIVA**

O controle manual de sistemas de abastecimento de água frequentemente resulta em desperdício e ineficiência. A adoção de sistemas automatizados para monitorar e gerenciar os níveis de água contribui para um uso sustentável dos recursos, além de reduzir custos operacionais e evitar incidentes, como transbordamentos ou ociosidade de bombas.

### **1.2 OBJETIVOS**

O principal objetivo deste trabalho é desenvolver um sistema microprocessado que:

1. Monitore e controle os níveis de água nos reservatórios subterrâneo e superior.
2. Acione bombas automaticamente conforme limites pré-determinados.
3. Inclua mecanismos de alarme para situações críticas, como níveis baixos ou transbordamentos.

## **2. REVISÃO DE LITERATURA**

A automação de sistemas hidráulicos é amplamente abordada na literatura técnica, com ênfase no uso de sensores de nível ultrassônicos e bóias eletrônicas. Microcontroladores são frequentemente utilizados pela sua simplicidade e flexibilidade. Estudos mostram que a integração desses dispositivos com algoritmos de controle pode melhorar significativamente a eficiência operacional de sistemas de abastecimento.

### 3. METODOLOGIA

#### 3.1 ESTRUTURA DO SISTEMA

O sistema será composto pelos seguintes elementos:

- **Hardware:** 5 Mini Chave Gangorra - KCD11-101, microcontrolador 18F452, relé de 5V, 1 bomba de 5V DC, 1 LCD 16x2, 1 crystal de 8MHz, 1 Transistor NPN, 3 Resistores de 300Ω, 2 Resistores de 10KΩ, 3 Led, 1 Diodo, 2 capacitor de cerâmica 22pF.
- **Software:** Algoritmo de controle desenvolvido em C no aplicativo mikroC pro for Pic para gerenciar os níveis de água e acionar as bombas conforme necessário.

E a montagem esquemática do circuito para preparação da montagem e o teste do código em C foi feita no aplicativo Proteus 8 Professional.

#### 3.2 Fluxo de Operação

1. O sensor de nível monitora os volumes nos dois reservatórios (subterrâneo e superior).
2. Quando o nível do reservatório superior atingir 40%, a bomba é acionada para enchê-lo até 100%.
3. Caso o nível do reservatório subterrâneo atinja 30%, um alarme é ativado, e a bomba é desligada para evitar danos.
4. Um alarme sonoro adicional é acionado caso o sistema detecte transbordamento no reservatório superior.

#### 3.3 Critérios de Sucesso

O sistema será avaliado por sua capacidade de:

- Responder rapidamente a mudanças nos níveis de água.
- Operar de forma contínua e confiável por pelo menos 24 horas.
- Evitar transbordamentos ou vazios críticos nos reservatórios.

### 4. DESENVOLVIMENTO

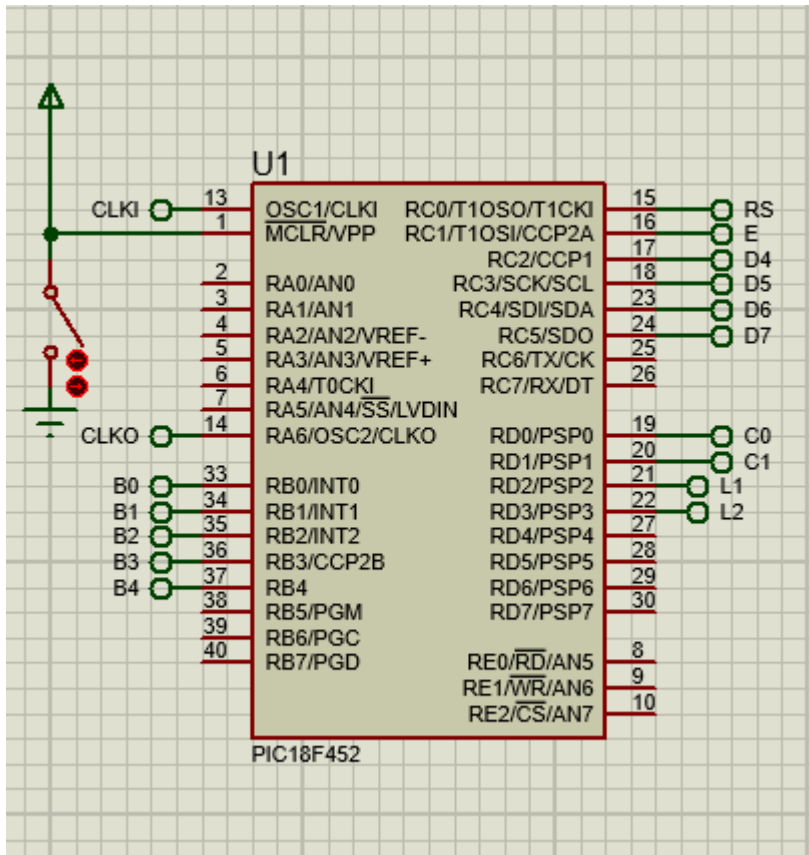
A construção do sistema foi dividida em etapas:

1. **Projeto do Circuito Elétrico:** Os sensores e atuadores foram conectados ao microcontrolador, garantindo segurança e eficiência.
2. **Desenvolvimento do Software:** Um algoritmo foi programado para monitorar os sensores em tempo real e acionar os atuadores.
3. **Testes e Ajustes:** O sistema foi testado em cenários simulados e ajustado para lidar com falhas potenciais.

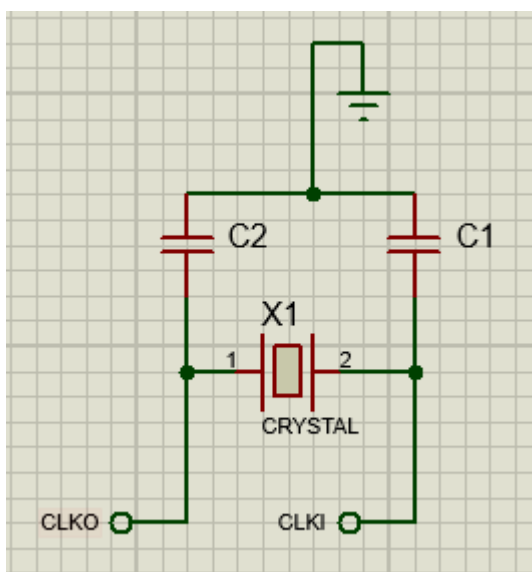
Montagem e visualização no Proteus:

Ligações do PIC18F452:

O componente Principal Pic18F452 para automação da caixa d'água e implementação do código.

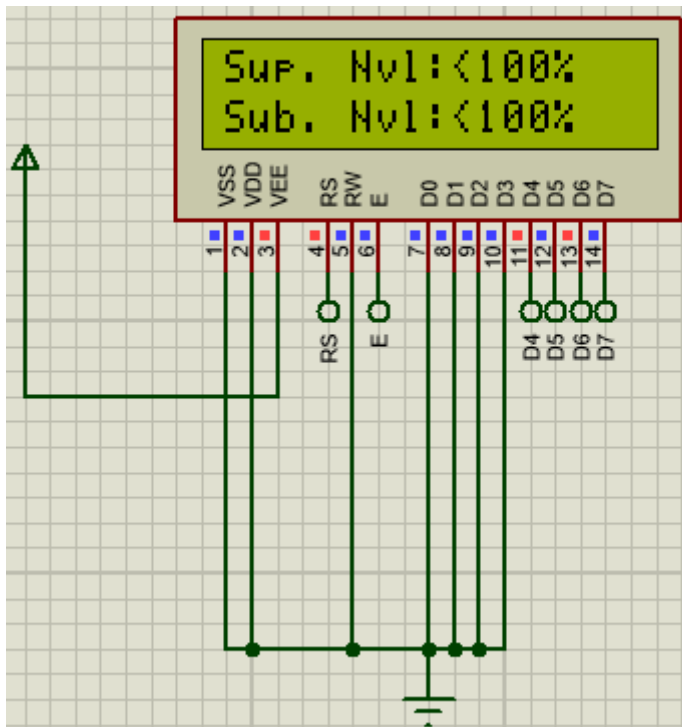


Cristal 8MHz em paralelo com dois capacitores de cerâmica de 22pF, para manter a oscilação e o controle do PIC:





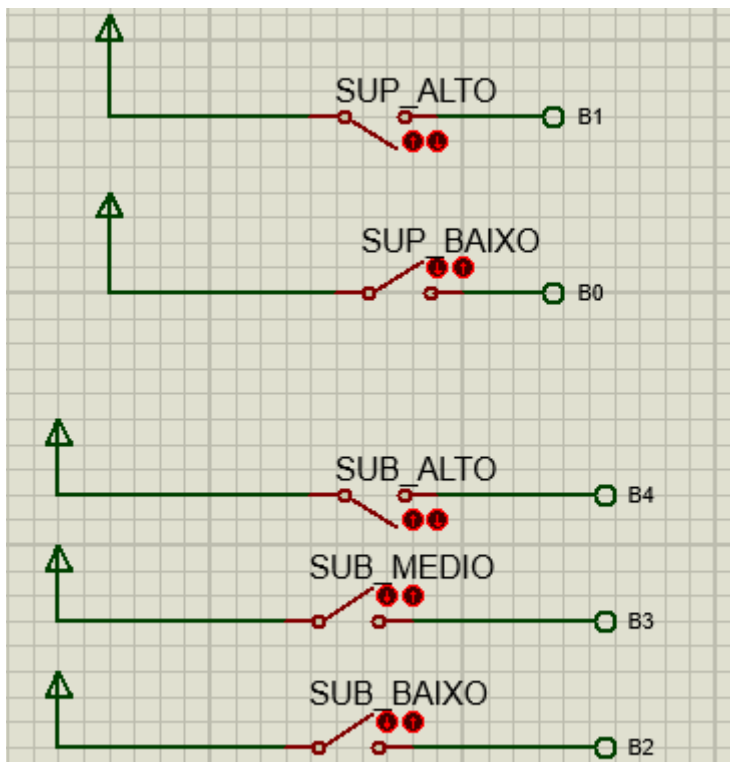
Ligação do LCD16x2 visualização da porcentagem das duas caixa água.



Montagem das Boias:

2 boias superior 1 no 40%(virada para cima) e a outra no 100%(virada para baixo).

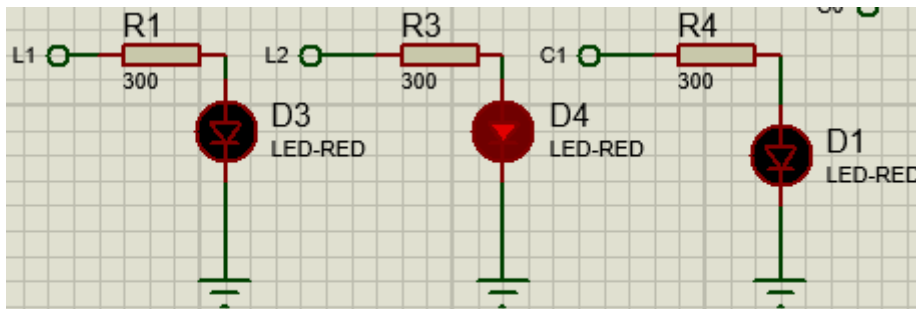
3 boias subterrânea 1 no 30%(virada para cima), 1 no 60%(virada para cima) e a outra no 100%(virada para baixo).





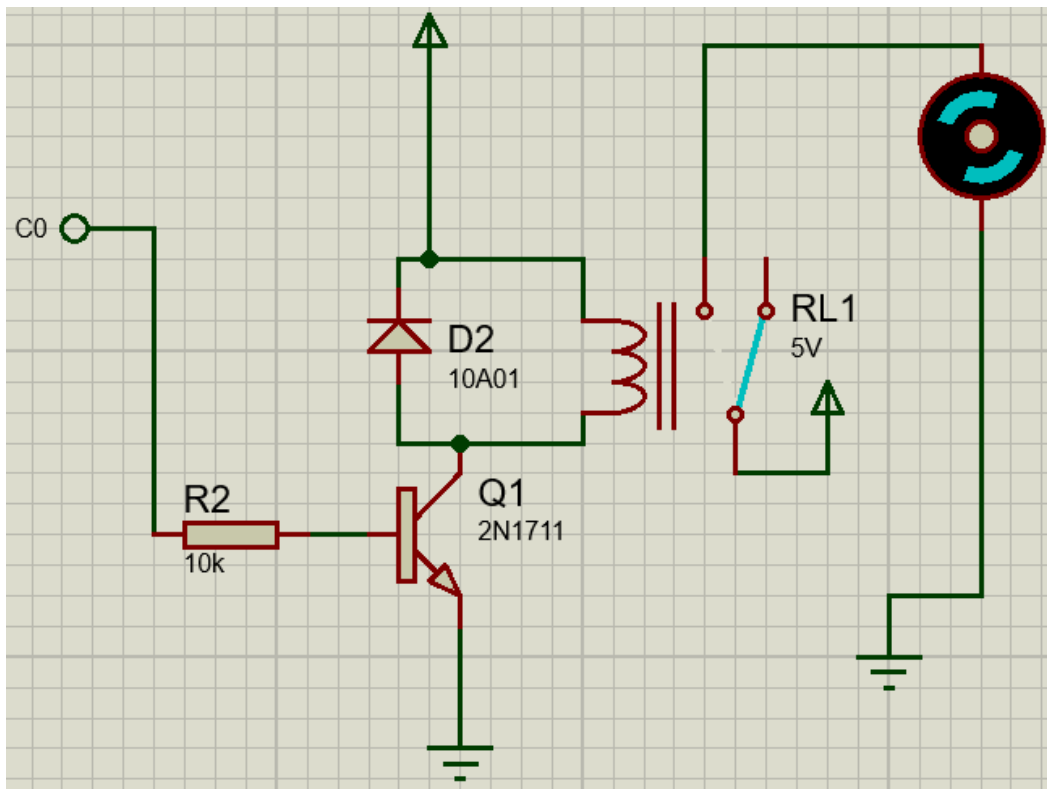
Ligação Dos Leds:

Visualização da bomba LED-D3 = bomba ligada, Led D4 = bomba desligada e Led-D1 =  
Alarme



Montagem do Relé e da bomba :

Um resistor de 10kΩ ligado na base do transistor NPN, o emissor mandando para GND e o coletor em paralelo com o relé e o diodo, motor ligado no relé no NO (normalmente aberto).



O código a seguir implementa o sistema descrito na metodologia:

```
// Configuração dos Fuses
#pragma config OSC = HS, WDT = OFF, LVP = OFF, BOR = ON
#define _XTAL_FREQ 8000000 // Cristal de 8 MHz

// Definição dos pinos
#define BOIA_RS_SUP_BAIXO PORTB.RB0 // Nível baixo do reservatório superior (40%)
#define BOIA_RS_SUP_ALTO PORTB.RB1 // Nível alto do reservatório superior (100%)
```

```

#define BOIA_RS_SUB_BAIXO    PORTB.RB2    // Nível crítico do reservatório
subterrâneo (30%)
#define BOIA_RS_SUB_ALTO    PORTB.RB3    // Nível seguro do reservatório
subterrâneo (60%)
#define BOIA_RS_SUB_Plato    PORTB.RB4

#define BOMBA                LATD.LATD0    // Controle da bomba
#define ALARME                LATD.LATD1    // Controle do alarme
#define LED_BOMBA_ATIVA      LATD.LATD2    // LED indicando bomba ativa
#define LED_BOMBA_INATIVA    LATD.LATD3    // LED indicando bomba inativa

// Pinos do LCD
sbit LCD_RS at LATC0_bit;
sbit LCD_EN at LATC1_bit;
sbit LCD_D4 at LATC2_bit;
sbit LCD_D5 at LATC3_bit;
sbit LCD_D6 at LATC4_bit;
sbit LCD_D7 at LATC5_bit;

sbit LCD_RS_Direction at TRISC0_bit;
sbit LCD_EN_Direction at TRISC1_bit;
sbit LCD_D4_Direction at TRISC2_bit;
sbit LCD_D5_Direction at TRISC3_bit;
sbit LCD_D6_Direction at TRISC4_bit;
sbit LCD_D7_Direction at TRISC5_bit;

// Variáveis globais
volatile unsigned long contadorTempo = 0; // Contador de tempo em segundos
unsigned long tempoParaAcionarBomba = 60; // 24 horas em segundos

// Função para inicializar o sistema
void init_system() {
    TRISB = 0b00011111; // Configura os pinos RB0 a RB4 como entradas
    TRISD = 0x00;        // Configura todos os pinos de PORTD como saídas
    LATD = 0;            // Inicializa os pinos de saída em nível baixo
    ADCON1 = 0x0F;       // Configura os pinos como digitais
}

// Função para inicializar o Timer1
void init_timer1() {
    T1CON = 0x31; // Timer1 habilitado, prescaler 1:8, clock interno
    TMR1H = 0x3C; // Valor inicial para gerar interrupção a cada 1 segundo (8
MHz e prescaler 1:8)
    TMR1L = 0xB0;
    PIE1.TMR1IE = 1; // Habilita a interrupção do Timer1
    INTCON.PEIE = 1; // Habilita interrupções de periféricos
    INTCON.GIE = 1;  // Habilita interrupções globais
}

// Interrupção do Timer1
void interrupt() {
    if (PIR1.TMR1IF) { // Verifica se a interrupção é do Timer1
        PIR1.TMR1IF = 0; // Limpa a flag de interrupção do Timer1
        TMR1H = 0x3C;     // Recarrega o Timer1 para gerar interrupção a
cada 1 segundo
        TMR1L = 0xB0;

        contadorTempo++; // Incrementa o contador de segundos
    }
}

// Verifica acionamento automático da bomba
void verifica_acionamento_diario() {
    if (contadorTempo >= tempoParaAcionarBomba) {
        BOMBA = 1; // Liga a bomba
        LED_BOMBA_ATIVA = 1;
        LED_BOMBA_INATIVA = 0;
        Delay_ms(3000); // Mantém a bomba ativa por 5 segundos
    }
}

```

```

        BOMBA = 0;                // Desliga a bomba
        LED_BOMBA_ATIVA = 0;
        LED_BOMBA_INATIVA = 1;
        contadorTempo = 0;        // Reseta o contador de tempo
    }
}

// Controle principal do sistema
void controle_sistema() {
    // Situação crítica no reservatório subterrâneo
    if (BOIA_RS_SUB_BAIXO == 1) {
        BOMBA = 0;                // Desliga a bomba
        ALARME = 1;                // Liga o alarme
        LED_BOMBA_ATIVA = 0;
        LED_BOMBA_INATIVA = 1;
    } else if (BOIA_RS_SUB_ALTO == 0) {
        ALARME = 0;                // Desliga o alarme quando o nível volta a
60%
    }

    // Controle da bomba para o reservatório superior
    if (BOIA_RS_SUP_BAIXO == 1 && BOIA_RS_SUB_BAIXO == 0 && BOIA_RS_SUB_ALTO
== 0) {
        BOMBA = 1;                // Liga a bomba se o reservatório superior
está baixo
        LED_BOMBA_ATIVA = 1;
        LED_BOMBA_INATIVA = 0;
    } else if (BOIA_RS_SUP_ALTO == 1) {
        BOMBA = 0;                // Desliga a bomba se o reservatório superior
está cheio
        LED_BOMBA_ATIVA = 0;
        LED_BOMBA_INATIVA = 1;
    }
}

// Função principal
void main() {
    init_system();                // Inicializa o sistema
    init_timer1();                // Inicializa o Timer1
    Lcd_Init();                   // Inicializa o LCD
    Lcd_Cmd(_LCD_CLEAR);         // Limpa o LCD
    Lcd_Cmd(_LCD_CURSOR_OFF);

    while (1) {
        // Atualiza o controle do sistema
        controle_sistema();

        // Verifica acionamento diário da bomba
        verifica_acionamento_diario();

        // Exibe status no LCD
        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1, 1, "Sup. Nvl:");
        Lcd_Out(1, 10, BOIA_RS_SUP_BAIXO ? "40%" : (BOIA_RS_SUP_ALTO ? "100%"
: "<100%"));
        Lcd_Out(2, 1, "Sub. Nvl:");
        Lcd_Out(2, 10, BOIA_RS_SUB_BAIXO ? "30%" : (BOIA_RS_SUB_ALTO ? "60%" :
(BOIA_RS_SUB_PLATO ? "100%" : "<100%")));

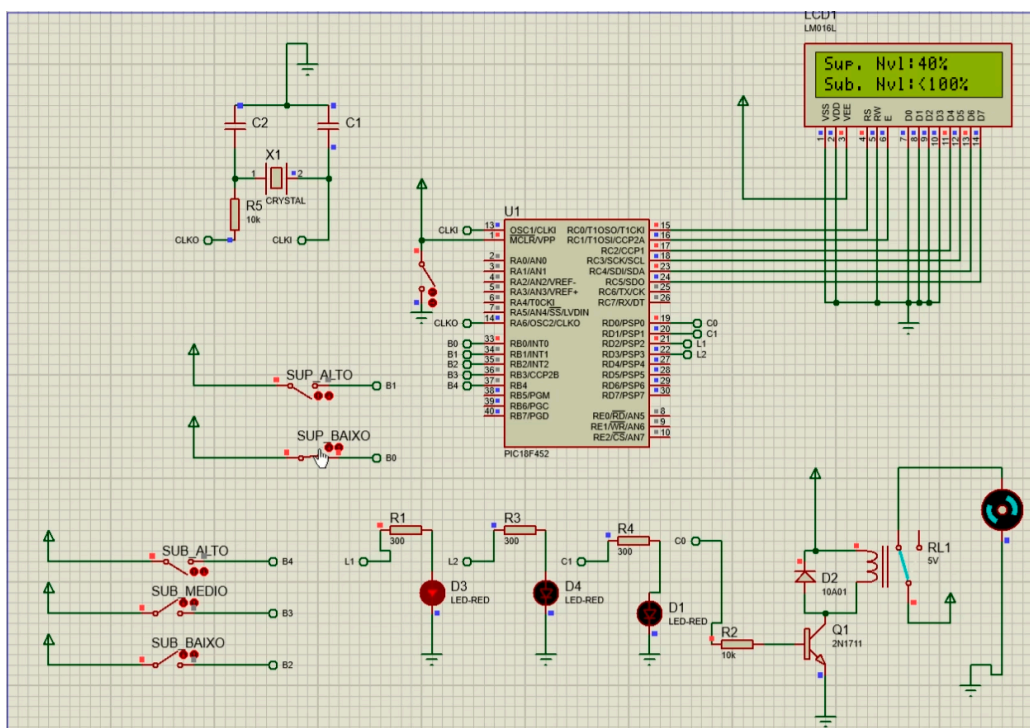
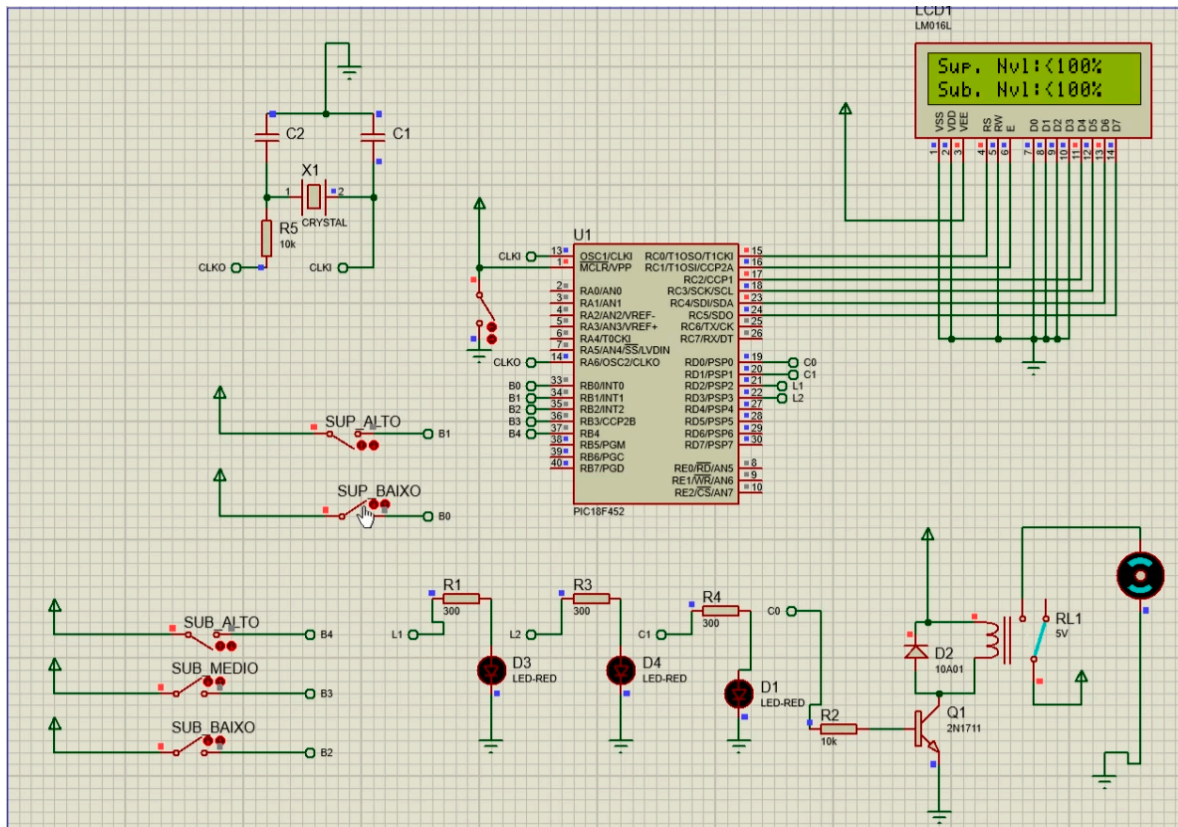
        Delay_ms(1000); // Atualiza a cada segundo
    }
}

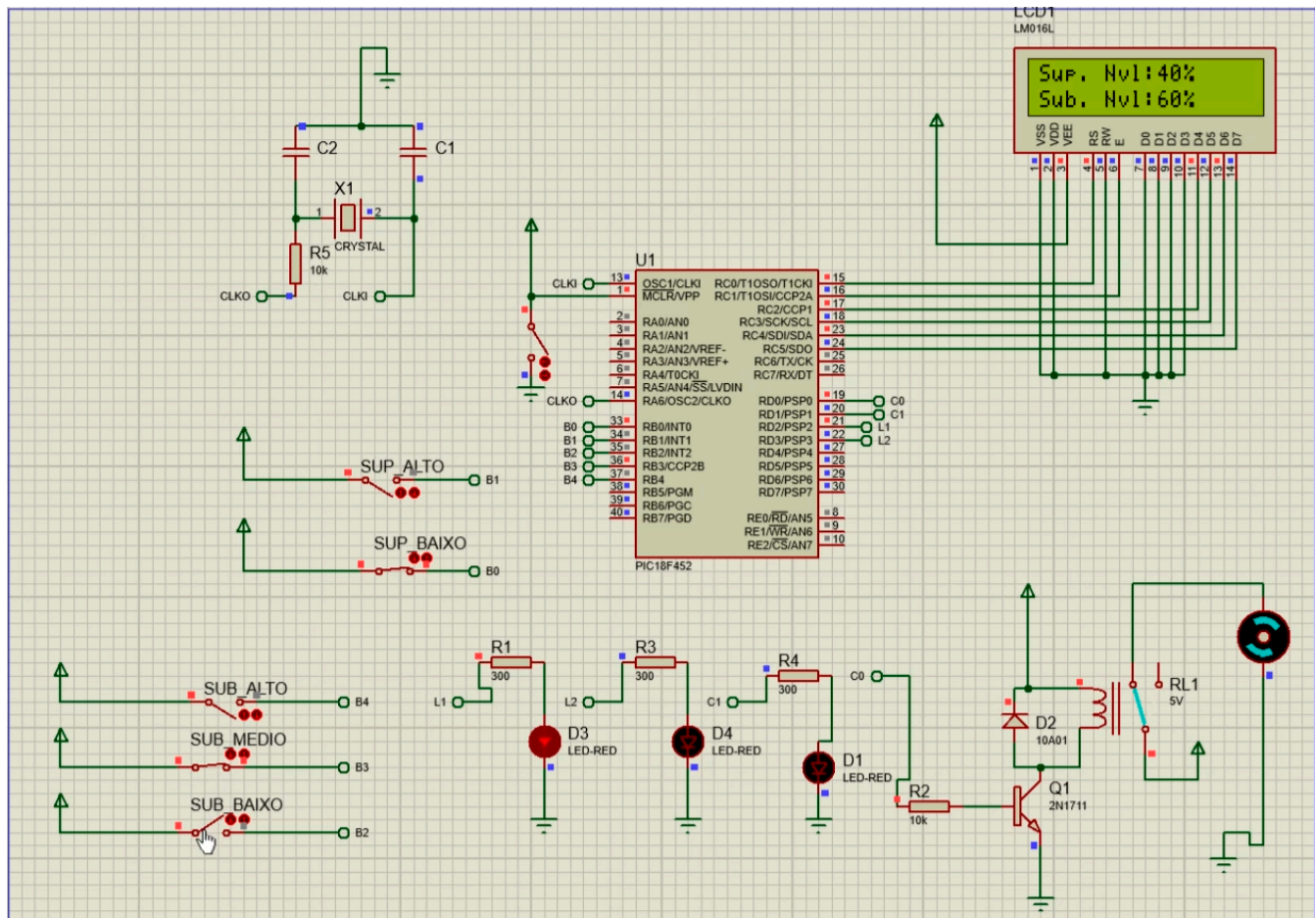
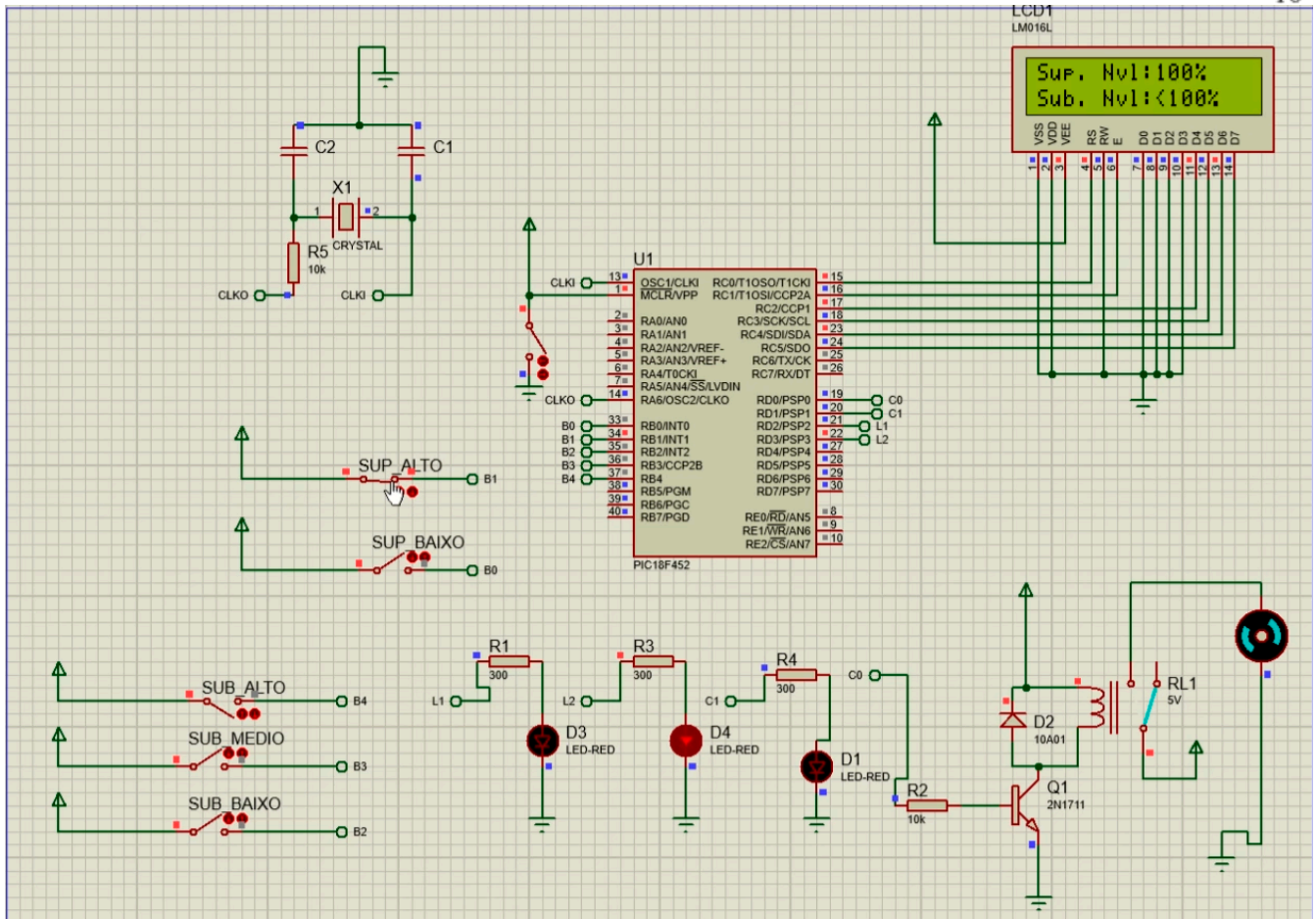
```

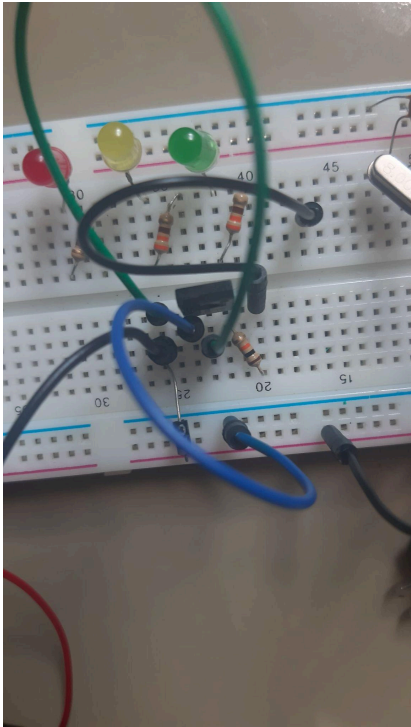
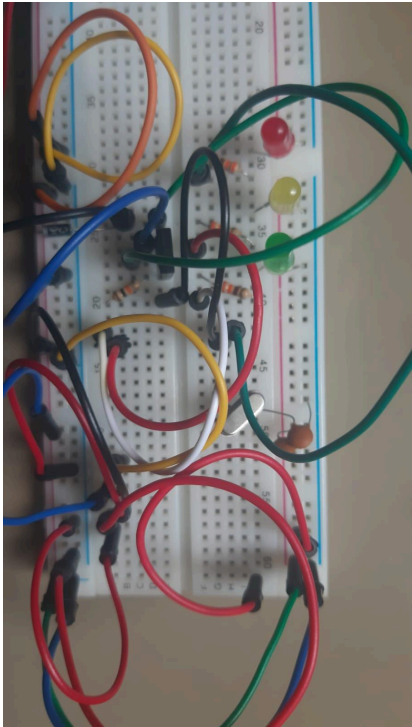
## 5. TESTE E TEORIAS

Os testes preliminares indicaram que o sistema consegue monitorar e controlar os níveis de água com precisão. No entanto, foram observados desafios, como a necessidade de calibração dos sensores para garantir leituras estáveis em ambientes com alta umidade. O uso de alarmes sonoros provou ser eficaz para alertar operadores em situações crítica

A seguir, são apresentados exemplos:







## CONCLUSÃO OU CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema microprocessado para o controle automático de reservatórios de água demonstrou ser uma solução promissora e eficiente para lidar com os desafios enfrentados no gerenciamento hídrico. A integração de hardware e software permitiu a criação de um sistema confiável, capaz de monitorar níveis de água em tempo real e acionar bombas de maneira precisa e automatizada, contribuindo para a redução de desperdícios, aumento da eficiência operacional e mitigação de falhas humanas.

Os resultados obtidos indicam que o sistema atende aos requisitos propostos, como a manutenção dos níveis adequados nos reservatórios e a ativação de alarmes em situações críticas, como níveis perigosamente baixos no reservatório subterrâneo ou transbordamento no reservatório superior. Esses recursos tornam o sistema ideal para aplicações em áreas residenciais, comerciais e industriais, onde o uso otimizado de recursos hídricos é essencial.

Durante o processo de desenvolvimento e testes, alguns desafios foram identificados, como a necessidade de calibrar os sensores em diferentes condições ambientais para garantir leituras estáveis e confiáveis. Além disso, o sistema se mostrou dependente de uma fonte de energia contínua, o que pode limitar sua aplicabilidade em locais sujeitos a interrupções frequentes de energia elétrica. Esses desafios representam oportunidades para melhorias futuras.

Como sugestões para evolução do projeto, destacam-se as seguintes possibilidades:

1. **Integração com Tecnologias de IoT (Internet das Coisas):** Incorporar conectividade via Wi-Fi ou LoRa para permitir o monitoramento remoto dos reservatórios por meio de aplicativos ou plataformas online.
2. **Desenvolvimento de Algoritmos Mais Sofisticados:** Utilizar algoritmos baseados em inteligência artificial ou lógica fuzzy para prever padrões de consumo e ajustar automaticamente os níveis de operação.
3. **Implementação de Sistemas de Energia Alternativa:** Adicionar painéis solares ou baterias de backup para garantir a continuidade do funcionamento em caso de falhas na rede elétrica.
4. **Expansão do Sistema de Monitoramento:** Acrescentar sensores adicionais para monitorar a qualidade da água, como níveis de pH e turbidez, aumentando o valor agregado do sistema.

Por fim, este trabalho contribui para o campo de sistemas microprocessados e demonstra como a automação pode ser aplicada de forma prática para resolver problemas do cotidiano. O

projeto servirá como base para futuras pesquisas e aplicações em áreas relacionadas, como a agricultura irrigada, o gerenciamento de redes de abastecimento urbano e sistemas industriais de grande escala. Com as melhorias propostas, o sistema poderá alcançar novos níveis de eficiência e acessibilidade, promovendo o uso sustentável dos recursos hídricos e beneficiando diretamente a sociedade e o meio ambiente.

## REFERÊNCIAS

- Datasheets de sensores ultrassônicos.
- Documentação oficial do microcontrolador 18F452
- Artigos sobre controle de sistemas hidráulicos com microcontroladores.