

# Sistema de Gerenciamento de Reservatório de Água

Lucas Oliveira da Silva - tic370100279

Este projeto consiste em um sistema embarcado para monitoramento e controle do nível de um reservatório de água, utilizando um Raspberry Pi Pico W, sensores e periféricos como display OLED, matriz de LEDs 5x5, joystick e botões para interação. Devido a limitações da placa, os sensores foram deixados para nível de simulação. Esse componente foi desenvolvido para operar seguindo a definição de um sistema embarcado, realizando somente uma temperatura, consumindo poucos recursos.

O sistema tem como funcionalidades a leitura do nível do reservatório, o controle da bomba, que pode ser de forma automática, operando com opções de encher, esvaziar ou parar. Já no controle manual, isso ocorre através do joystick no eixo y. Além disso, emite alertas em conjunto com as respectivas leituras, informando quando o reservatório está cheio ou vazio.

## **Título do Projeto**

Sistema de Monitoramento e Controle de Reservatório de Água

## **Objetivos do Projeto**

- Monitorar o nível de água em tempo real, e fornecer um feedback em tempo real por meio dos leds RGB.
- Exibir informações detalhadas no display OLED, e utilizar este como meio de navegação entre os menus, permitindo assim um controle visual do sistema.
- Representar visualmente o nível da água na matriz de LEDs, com a cor azul.
- Permitir ajustes manuais via joystick e botões, além da navegação e controle. Além disso, o joystick é usado para simular o sensor da bomba d'água.

## **Descrição do funcionamento**

O sistema utiliza sensores de nível para medir a altura da coluna de água dentro do reservatório, esses dados são fornecidos em porcentagem. Esses sensores convertem a variação de nível em um sinal elétrico analógico, que é lido por um conversor analógico-digital (ADC) presente no Raspberry Pi Pico W. O firmware do microcontrolador processa esses valores e os traduz em percentuais de capacidade do reservatório, permitindo que o usuário acompanhe o estado do sistema com precisão. Os dados processados são exibidos em um display OLED SSD1306, que utiliza

comunicação I2C para se conectar ao microcontrolador. A interface gráfica no display é projetada para apresentar o nível da água em tempo real, além de alertas visuais em caso de níveis críticos (muito baixo ou muito alto). Além da exibição no display OLED, o sistema conta com uma matriz de LEDs 5x5, que fornece uma representação gráfica simplificada do nível da água. Cada linha da matriz pode representar uma fração da capacidade total do reservatório, permitindo que um usuário consiga verificar rapidamente o status sem precisar ler valores numéricos. Essa funcionalidade é útil para aplicações onde uma visualização rápida e intuitiva é necessária, como em locais de difícil acesso. Para interação, o sistema possui um joystick analógico e botões físicos, permitindo ajustes manuais e navegação entre diferentes modos de exibição. O joystick, que opera em dois eixos (X e Y), possibilita alternar entre telas no display OLED, alterar configurações do sistema ou ativar/desativar funções específicas, como o acionamento manual da bomba d'água. Os botões físicos podem ser utilizados para ações rápidas, como confirmação de comandos ou ativação de modos de emergência. Com esse conjunto de sensores, atuadores e interface gráfica interativa, o sistema oferece uma solução robusta e acessível para o gerenciamento eficiente de reservatórios de água, garantindo confiabilidade, facilidade de operação e escalabilidade para futuras melhorias.

## **Justificativa**

A gestão eficiente do consumo de água é fundamental para evitar desperdícios e garantir um abastecimento adequado. A implementação de sistemas automatizados de monitoramento e controle de reservatórios de água desempenha um papel crucial nesse contexto. Esses sistemas permitem um controle preciso do nível de água, acionando bombas de abastecimento ou descarregamento conforme necessário, prevenindo transbordamentos e escassez. Além disso, regulam a vazão da água que entra e sai do reservatório, assegurando um abastecimento contínuo e evitando desperdícios. Outro benefício é o monitoramento da qualidade da água, permitindo a verificação de parâmetros como pH, turbidez e níveis de cloro. Isso possibilita a identificação de possíveis contaminações ou desvios nos padrões de qualidade, garantindo a segurança para consumo. A automação desses processos elimina a necessidade de monitoramento manual constante e reduz os riscos de falhas operacionais que poderiam resultar em transbordamentos, vazamentos ou danos aos equipamentos. O uso de sensores inteligentes e sistemas embarcados em reservatórios também possibilita a integração com redes de IoT, permitindo que gestores monitorem e controlem remotamente o sistema por meio de dispositivos conectados. Isso é particularmente útil em grandes edificações, condomínios e instalações industriais, onde a gestão da água deve ser eficiente e automatizada.

## **Originalidade**

Embora existam sistemas similares no mercado, este projeto se destaca por integrar uma solução interativa e de baixo custo, combinando visualização gráfica avançada, controle

manual intuitivo e integração eficiente dos sensores. Diferentemente de soluções tradicionais que utilizam apenas sensores para detecção do nível de água e acionamento de bombas, este projeto inova ao incluir um display OLED para exibição detalhada de informações e uma matriz de LEDs 5x5 para representações visuais simplificadas. Isso permite uma compreensão rápida do estado do reservatório sem a necessidade de consultar dados numéricos. Outro diferencial é o uso de um joystick e botões físicos para interação com o sistema. Enquanto muitas soluções similares dependem exclusivamente de aplicativos ou interfaces web para ajustes e monitoramento, este projeto oferece uma alternativa física, permitindo que ajustes sejam feitos localmente sem necessidade de dispositivos adicionais. Além disso, o uso do Raspberry Pi Pico W traz a possibilidade de conexão remota, facilitando futuras expansões do sistema, como o envio de notificações em tempo real para alertar sobre níveis críticos de água. A combinação desses elementos proporciona uma interface amigável e funcional, diferenciando-se de soluções convencionais que, muitas vezes, carecem de interatividade ou apresentam custos elevados.

### **Função de Cada Bloco**

**Raspberry Pi Pico W:** Atua como a unidade de processamento principal do sistema. Ele recebe os sinais do sensor de nível de água, processa os dados e controla a exibição das informações no display OLED e na matriz de LEDs 5x5. Além disso, interpreta os comandos do joystick e dos botões, permitindo ajustes no funcionamento do sistema. Sua conectividade Wi-Fi embutida pode ser utilizada para futuras expansões, como monitoramento remoto.

**Sensor de Nível de Água:** Mede a altura da água no reservatório e converte esse valor em um sinal analógico ou digital, dependendo do modelo do sensor utilizado. Esse dado é enviado ao Raspberry Pi Pico W, que processa a informação e determina o nível percentual do reservatório. O sistema pode usar sensores de boia, sensores ultrassônicos ou sensores resistivos, dependendo da precisão desejada.

**Display OLED SSD1306:** Exibe informações detalhadas sobre o nível de água do reservatório em tempo real. Esse display permite mostrar alertas, mensagens de status do sistema e até mesmo configurações ajustáveis via joystick e botões. A interface gráfica melhora a experiência do usuário, tornando o monitoramento mais intuitivo.

**Matriz de LEDs 5x5:** Proporciona uma representação visual simplificada do nível da água no reservatório. Cada linha da matriz pode corresponder a um percentual do reservatório, acendendo os LEDs conforme o nível sobe ou desce. Esse recurso possibilita uma rápida identificação do estado do reservatório sem a necessidade de interpretar números ou gráficos detalhados.

**Joystick e Botões:** Permitem a interação do usuário com o sistema, possibilitando ajustes de configurações e navegação pelas informações exibidas no display OLED. O joystick pode ser usado para selecionar opções, visualizar dados históricos ou ajustar limites de alerta, enquanto os botões podem confirmar ações ou redefinir parâmetros do sistema.

LEDs RGB: São utilizados para fornecer feedback visual rápido e intuitivo sobre o estado do reservatório. Diferentes cores podem indicar condições específicas, como azul para nível normal, amarelo para nível baixo e vermelho para alerta crítico. Além disso, podem ser usados para indicar status de conexão, erros no sistema ou funcionamento de bombas de água, caso haja controle ativo sobre elas.

### Configuração de Cada Bloco

- Display OLED SSD1306 → Comunicação via I2C (endereço 0x3C).
- Joystick → Utiliza ADCs para capturar movimento nos eixos X e Y.
- Sensor de Nível → Conectado a um ADC para converter a altura da água em um valor digital.
- Matriz de LEDs 5x5 → Controlada por GPIOs do Raspberry Pi Pico W.
- Leds RGB → Controlado por GPIOs permite um feedback visual acerca do nível do reservatório. Por exemplo, num nível 100% o led verde é ligado. Já no nível 0% o led vermelho é ligado.

### Comandos e Registros Utilizados

- Leitura do sensor de nível e joystick → Hardware/adc.h
- Atualização do display OLED → Hardware/i2c.h
- Controle dos LEDs → Pico/stdlib.h
- Matriz de leds → Hardware/pio.h

### Descrição da Pinagem Usada

Periférico	Pino
Botão A	05
Botão B	06
Joystick_eixo_x	27
Joystick_eixo_y	26
Joystick_botão	22
Led_vermelho	13
Led_azul	12
Led_verde	11

## Definição das Variáveis

- pino\_botao\_a → variável para armazenar o pino do botão A
- pino\_botao\_b → variável para armazenar o pino do botão B
- pino\_vrx → pino do eixo X do joystick
- Pino\_vry → pino do eixo Y do joystick
- Pino\_botao\_joystick → pino do botão do joystick
- Pino\_led\_vermelho → pino do led vermelho
- Pino\_led\_verde → pino do led verde
- Pino\_led\_azul → pino do led azul
- Ultimo\_tempo → variável para armazenar o último tempo de acionamento do botão
- Entrar\_menu → variável para controlar a entrada no menu
- Posicao\_joystick → estrutura para armazenar a posição do joystick
- Dados\_sistema → estrutura para armazenar os dados do sistema

## Inicialização

No início da função main, o programa começa inicializando a entrada e saída padrão com a função `stdio_init_all`. Isso configura a comunicação serial, essencial para depuração e outras operações de entrada e saída. Em seguida, o programa configura os pinos dos botões, preparando-os para detectar eventos de pressão. Isso é feito para os botões A e B, permitindo que o sistema responda a interações do usuário. Depois, o programa configura o joystick, definindo os pinos para ler as posições dos eixos X e Y, bem como o estado do botão do joystick. Isso permite que o sistema capture a posição do joystick e ações do botão, que podem ser usadas para navegação e controle. A matriz de LEDs é então inicializada, preparando-a para exibir informações visuais, como níveis de reservatório ou status do sistema. Isso é seguido pela configuração e inicialização do display OLED. O display é preparado para uso e a estrutura que armazena seu estado é inicializada, permitindo que o sistema exiba informações textuais e gráficas. Os LEDs são configurados como saídas, permitindo que o sistema controle os LEDs vermelho, azul e verde. Esses LEDs podem ser usados para indicar diferentes estados ou alertas. Os dados do sistema são então inicializados. O nível do reservatório é definido para 100%, o modo de operação é configurado para automático, e o controle da bomba é definido para parar. Isso estabelece o estado inicial do sistema. O programa configura três timers repetitivos que chamam funções de callback a cada 100 milissegundos. Esses timers são usados para gerenciar o menu, o joystick e o nível da matriz de LEDs. Finalmente, uma interrupção é configurada para o botão A, que será acionada na borda de descida (quando o botão é pressionado). A função de interrupção será chamada

quando a interrupção ocorrer. O display é então limpo, apagando todos os pixels e enviando os dados para o display OLED. Com todas essas configurações concluídas, o sistema entra em um loop principal, onde permanece em execução contínua, aguardando eventos e atualizando o estado conforme necessário.

## **Execução do Projeto**

Inicialmente, foi necessário desenvolver as funções para o display, tendo em vista que ele seria o responsável por comandar e exibir o sistema. O desenvolvimento dessas funções relacionadas ao display, foi possível com a biblioteca i2c do módulo de hardware da API do Raspberry Pi. A priori, foi adicionado ao sistema o menu principal, tendo sua primeira versão apenas a exibição das opções, sem uma navegação entre elas. A principal dificuldade nesse processo, foi a questão da fonte, e de qual metodologia utilizar para desenvolver esses menus, optei por utilizar as enumerações como arcabouço para o menu principal.

Em seguida, surgiu a necessidade de navegar entre as opções do menu, para isso acontecer, foi fundamental o desenvolvimento do módulo joystick. Utilizando como plano de fundo a biblioteca adc.h, também pertencente ao módulo hardware da API Raspberry. Para realizar as leituras periodicamente, foi implementado um timer para leitura constante dos valores do eixo x e eixo y. Como consequência, dessa leitura, o módulo do menu, periodicamente, analisa esses dados, e mapeia se é necessário ir para a esquerda, ou direita.

Após essas configurações básicas, foi necessário, desenvolver a lógica para as opções do menu, como o nível do reservatório, permitir a troca do estado manual para o automático, ajustar as opções de controle da bomba. Essa foi mais a parte das regras de negócio. Por fim, utilizei o módulo, pio.h, sendo este também originário da api hardware. Ele foi necessário para implementar o controle da matriz de leds, que trata de, basicamente, ser uma forma visual do nível do reservatório. Como adicional, foi implementado os leds RGB para emitir alertas visuais da situação do reservatório. Sendo a cor verde responsável por emitir um sinal informando quando o nível for acima de 98%. O sinal de nível intermediário representado pela cor Amarela, informa que está em um nível estável. Por fim, a cor vermelha informa que o sistema se encontra em um nível crítico.

## **Manual de uso**

Botão A → utilizado para a operação de confirmar, ou trocar de modo dentro das opções.

Botão B → utilizado para a operação de retornar para o menu anterior.

Joystick para esquerda → Ele permite navegar para as opções mais à esquerda do sistema, sendo que quando chega na primeira opção (Porcentagem Reservatório) ele não muda mais.

Joystick para direita → Ele permite navegar para as opções mais à direita do sistema, sendo que quando chega na última opção (Controle Bomba) ele não muda mais.

Menu principal do sistema:

Porcentagem Reservatório → Indica a porcentagem atual do reservatório, variando de 0% a 100%.

Modo operação → Permite a troca do modo de operação do sistema, caso seja manual, o usuário pode controlar a operação de encher e esvaziar o reservatório com o joystick no eixo y. Operando de forma automática, o usuário não se preocupa em encher ou esvaziar, já que o sistema fará de forma automática.

Controle Bomba → Pode ser considerada como uma função derivada do modo automático, isto é, essa opção só é acessível, caso o sistema esteja no modo automático. Isso ocorre porque, é nessa opção que o usuário define a operação que será executada automaticamente como encher, esvaziar, ou parar o sistema.

## **Possíveis melhorias**

Uma das principais melhorias que enxergo nesse sistema, é a implementação de uma forma de ajustes remoto, ou seja, o sistema fornece feedbacks, e o usuário a partir disso, escolhe em que mexer. Além disso, poderia ser adotado formas de enviar relatórios de consumo, como por exemplo a vazão, o consumo total, que poderia ser mostrado em um gráfico, ficando visualmente mais acessível. Além de implementar outras funcionalidades, como temperatura, dentre outros.

## **Referências**

Raspberry Pi Foundation. Pico-series Microcontrollers - Raspberry Pi Documentation. Disponível em: [https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html?utm\\_source=chatgpt.com](https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html?utm_source=chatgpt.com). Acesso em: 20 fev. 2025.

Raspberry Pi Foundation. Getting started with your Raspberry Pi Pico W. Disponível em: [https://projects.raspberrypi.org/en/projects/get-started-pico-w/1?utm\\_source=chatgpt.com](https://projects.raspberrypi.org/en/projects/get-started-pico-w/1?utm_source=chatgpt.com). Acesso em: 20 fev. 2025.

GUANAIS, Henrique. Tank Secure: Projeto de monitoramento de reservatório de água com autenticação RFID utilizando Raspberry Pi Pico W. Disponível em: <https://github.com/henriqueguanais/tank-secure>. Acesso em: 20 fev. 2025.

NEWBIELY. Raspberry Pi Pico - Water Sensor. Disponível em: [https://newbiely.com/tutorials/raspberry-pico/raspberry-pi-pico-water-sensor?utm\\_source=chatgpt.com](https://newbiely.com/tutorials/raspberry-pico/raspberry-pi-pico-water-sensor?utm_source=chatgpt.com). Acesso em: 20 fev. 2025.

HACKADAY.IO. Water Meter Monitor with Raspberry Pi Pico W. Disponível em: [https://hackaday.io/project/189706-water-meter-monitor-with-raspberry-pi-pico-w?utm\\_source=chatgpt.com](https://hackaday.io/project/189706-water-meter-monitor-with-raspberry-pi-pico-w?utm_source=chatgpt.com). Acesso em: 20 fev. 2025.

VOLT SOLUTION. Automação de Reservatórios: Como a Telemetria Está Modernizando o Controle de Água. Disponível em: <https://voltsolution.com.br/automacao-de-reservatorios-como-a-telemetria-esta-modernizando-o-controle-de-agua>. Acesso em: 20 fev. 2025.

ANAUGER. Importância do Controle de Nível de Água para Reservatórios, Cisternas e Caixas d'Água. Disponível em: <https://anauger.com.br/importancia-do-controle-de-nivel-de-agua-para-reservatorios-cisternas-e-caixas-dagua>. Acesso em: 20 fev. 2025.