

Smart Tank

GitHub do projeto: <https://github.com/klebersm/Embarcatech-U7T1>

Vídeo de demonstração:

<https://drive.google.com/file/d/1UaTDoonUdeKe41Mfk1ap8MS4KUDjbxsi/view>

Escopo do Projeto

A nova organização econômica mundial exige das empresas cada vez mais eficiência na gestão de recursos e redução de custos. Neste cenário, a logística enfrenta grandes desafios para melhorar a cada dia as cadeias de suprimentos.

Neste universo, tanques e silos são estruturas fundamentais para manutenção de estoques dos mais variados tipos de produtos. Monitorar essas estruturas e a evolução dos itens armazenados pode trazer grandes benefícios para os seus usuários.

Pensando nisto, este projeto visa propor um sistema de monitoramento de nível dessas estruturas baseado em sinais captados a partir de células de carga, capazes de indicar a quantidade de produto disponível em tempo real, com o intuito de auxiliar na gestão do suprimento monitorado.

O sistema consiste na leitura de duas informações de carga, permitindo o controle de dois tanques/silos em paralelo, disponibilizando essas informações para o usuário e possibilitando a criação de alarmes para nível crítico, com o objetivo de gerar eventos que podem desencadear pedidos de reposição, por exemplo.

Ao ser inicializado, o sistema considera como nível mínimo, pra cada tanque, o valor correspondente a 30% da sua capacidade, permitindo ao usuário a alteração desta configuração para melhor controle.

Além das funcionalidades já disponíveis, em versões futuras o sistema pode se aproveitar da conectividade do Pi Pico W para descarregar os dados monitorados em um servidor de dados, agregando a possibilidade de criação de dashboards e históricos que ajudem a entender melhor o comportamento de uso/reposição de cada produto ao longo do tempo.

É possível encontrar outros produtos parecidos no mercado, como o petrolíder (petrolider.com.br), para monitoramento de combustíveis, ou o incontrol (incontrol.ind.br) para monitoramento de água. A desvantagem desses produtos é que eles aplicam sondas específicas para cada aplicação, fazendo com que o uso de cada um deles atenda a um público muito limitado, além da necessidade de introdução das sondas nos tanques, que, a depender do ambiente, pode se tornar algo muito desafiador. Um exemplo disso é a abertura de um furo em um tanque cheio de combustível para a inserção do sensor. Em áreas classificadas, este simples procedimento pode se tornar completamente inviável ou, no mínimo, bastante custoso, pelo alto risco de explosão.

A execução do projeto foi realizada seguindo uma abordagem top down, partindo do comportamento esperado para a aplicação e quebrando-a em partes menores, a saber:

- Leitura dos sensores
- Conversão dos valores lidos para porcentagem
- Comparação dos valores de porcentagem com os setpoints de alarmes
- Geração de informações visuais para o usuário
- Geração de sinais de alarme
- Leitura de botões para possibilita reconhecimento de alarmes ou atualização de setpoints.

Como requisitos funcionais foram levantadas as seguintes características:

- Leitura de um ou mais sinais analógicos, para captura dos níveis de carga
- Exibição dos dados em tempo real para que o usuário consiga ler com facilidade o status atual dos parâmetros lidos.
- Indicadores de alarme para chamar atenção do usuário sempre que um dos níveis estiver abaixo do valor desejado.
- Possibilidade de interação do usuário para silenciar alarmes e configurar o nível de alarme de acordo com sua preferência.

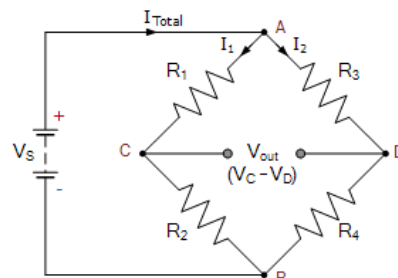
Como requisitos não funcionais, foram mapeados:

- Baixo custo de implementação
- Possibilidade de conexão com a internet para compartilhamento dos dados.
- Possibilidade de integração de novos hardwares enriquecendo a interação com o usuário, como teclados alfanuméricos, possibilitando a entrada de dados como senhas, ou valores de configurações desejados.

Especificação do Hardware

O projeto foi desenvolvido com base na placa BitDogLab, baseada no microcontrolador Raspberry RP2040, na versão Pi Pico W. Quase todos os itens disponíveis na placa foram utilizados. O display oLED apresenta os valores atuais do sistema, a matriz de leds, assim como os sinais sonoros, apresentam informações de alarmes

A função do Joystick merece destaque, já que ele opera simulando a principal entrada de dados do sistema, responsável pela leitura das informações de campo. A ideia neste ponto é simular duas células de cargas, fazendo o papel de módulos HX711, capazes de amplificar o sinal de uma ponte de Wheatstone, fazendo a conversão das resistências presentes em uma célula de carga para valores digitais.

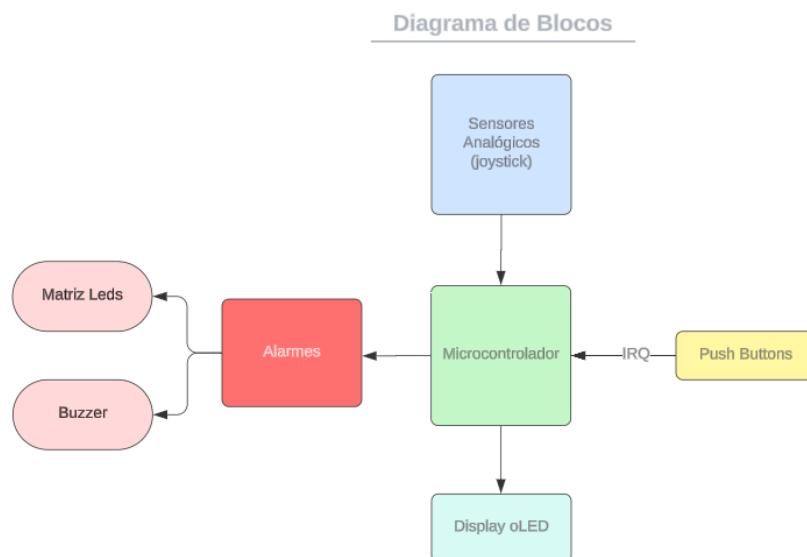


Ponte de Wheatstone
Fonte: [Electronics Tutorials](#)

O módulo HX711 recebe os dados de uma ou mais células de carga, sempre respeitando as características da ponte de Wheatstone, faz a amplificação do sinal e a conversão analógico digital com uma resolução de 24 bits e envia essa informação para o microcontrolador usando um protocolo SPI simplificado, que usa só os pinos SCK(para sincronização de clock) e DT para envio de dados do módulo para o controlador.

Para este projeto, o controlador atuará fazendo a leitura diretamente dos potenciômetros do joystick, recebendo o valor analógico convertido para valor digital com resolução de 12 bits, ou seja, podendo variar de 0 até 4096.

O diagrama de blocos abaixo demonstra como se dá o funcionamento do hardware, que tem o microcontrolador como elemento central. Os sensores são lidos de forma contínua e, a partir dessas leituras, os valores são convertidos para gerar dados visuais no display e alarmes, dependendo das comparações realizadas com os valores de setpoint. Além disso, o controlador está sujeito a interrupções geradas pelos botões A e B, capazes de realizar o reconhecimento de um alarme, para cessar o sinal sonoro, ou de configurar o valor de setpoint para futuras comparações e geração de alarmes.



Os blocos de hardware foram configurados da seguinte forma:

- Bloco de sensores: As duas entradas analógicas ligadas aos potenciômetros do Joystick foram configuradas para leituras simulando o conversor A/D das células de carga, de forma que o eixo y representa o nível atual no tanque A e o eixo x representa o nível no tanque B.
- Bloco de botões: Configurados como entradas digitais com PULL UPS, foram habilitados em formato de interrupção na borda de descida, fazendo com que o controlador seja interrompido sempre que um dos botões for pressionado. Nestas interrupções, o controlador checa se existe alarme acionado para o tanque respectivo. Se existe alarme, o clique funciona inibindo o sinal sonoro, como uma forma de indicar que o alarme já foi reconhecido. Se o tanque não está em situação de alarme, o controlador carrega o valor atual do nível para o setpoint de alarme,

permitindo ao usuário indicar qual a melhor posição para o disparo dos próximos alarmes.

- Bloco de Display: Foi configurado usando a biblioteca SSD 1306, que utiliza interface i2c para exibição de dados no display. Exibe as informações atuais relativas aos dois tanques lidos, indicando em formato de barras, para facilitar a leitura do usuário, além de exibir o valor do nível em porcentagem e indicar status de alarme para cada tanque. Os status exibidos podem variar entre: NRML, para nível normal; ALM, ocorrência de alarme no respectivo tanque; ou ~~ALM~~ quando o tanque está em nível crítico, mas o usuário já indicou reconhecimento do alarme através dos botões.
- Bloco de alarmes: Se divide em dois blocos:
 - Buzzer: Foi configurado a partir de um PWM para controle de sinais sonoros. É acionado de forma intermitente sempre que um alarme acontece, tem o objetivo de chamar atenção do usuário para uma situação de alarme. Pode ser desativado manualmente pelo usuário, através dos botões disponíveis, ou deixa de bipar se as condições de alarme deixarem de existir, quando os níveis voltam ao normal.
 - Matriz de led: Foi inicializada usando o PIO para controle dos leds WS2812 de forma independente do loop principal da aplicação. Atua como sinal auxiliar, indicando pro usuário qual dos tanques está em situação de alarme, exibindo a letra A ou B, de acordo com o tanque que gerou o alerta, ou intercalando as duas letras caso os dois tanques apresentem nível baixo. As letras são exibidas normalmente na cor vermelha. Mas mudam para laranja quando o alarme é reconhecido pelo usuário, indicando que o tanque ainda continua em nível baixo, mas que o usuário já atestou ciência da situação. A matriz deixa de exibir informações quando os tanques voltam aos seus níveis normais.

Os pinos utilizados foram os seguintes:

DESCRIÇÃO	PINO
<i>Joystick X</i>	26
<i>Joystick Y</i>	27
<i>Botão A</i>	5
<i>Botão B</i>	6
<i>Buzzer</i>	21

Além disso, um debounce foi configurado para 75ms, funcionando com o auxílio de alarmes de tempo. Sempre que um botão é pressionado, um alarme interno do microcontrolador é habilitado pra acontecer 75 milissegundos depois desse acionamento. Se qualquer outra coisa acontecer com o botão, este alarme é anulado, indicando que alguma oscilação aconteceu. O alarme só é realmente disparado, entrando na função handler do botão, se o mesmo foi pressionado e passou 75ms estabilizado, sem apresentar nenhuma alteração de comportamento que invalide o evento criado.

Especificação do Firmware

O firmware inicia configurando todos os componentes necessários para a execução das tarefas, incluindo o Display oLED, a matriz de led, os botões A e B e os sinais analógicos do joystick.

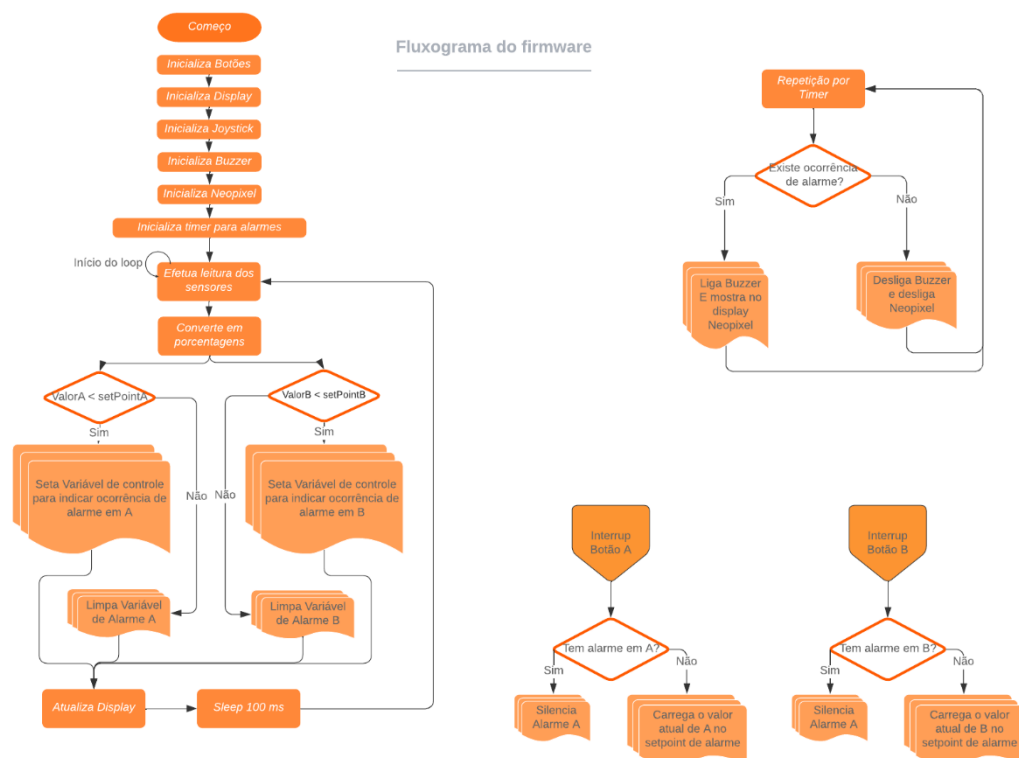
Além disso, são inicializadas as rotinas de debounce, que utilizam alarmes para aguardar a estabilização dos sinais de clique dos botões.

Já no loop principal, o firmware atua fazendo a leitura dos dois potenciômetros do joystick, que geram sinais simulando os níveis em cada tanque. Depois da leitura, os valores são convertidos em porcentagens para geração dos gráficos e demais comportamentos do sistema, como indicadores de alarmes. A partir dessas porcentagens e demais cálculos, o firmware procede com a atualização dos dados no display.

Fora do loop principal, controlado por interrupções, os botões A e B possuem comportamentos relativos aos alarmes de nível baixo dos dois tanques. Se o alarme está disparado, o acionamento do botão indica o reconhecimento do alarme, fazendo pausar o sinal sonoro referente ao respectivo tanque. Se não existe alarme disparado ou se o alarme atual encontra-se em estado de “CLEARED”, ou seja, alarme reconhecido pelo usuário, o acionamento do botão faz com que o setpoint de alarme seja configurado para o nível atual do tanque referente ao botão clicado: Botão A para o tanque A e Botão B para o tanque B.

Em um loop secundário, executado a partir de um alarme de timer, a matriz de led e os sinais sonoros são atualizados, sempre baseados nos valores atuais calculados durante o loop principal da aplicação.

O Diagrama de fluxo abaixo demonstra o funcionamento do Firmware



A lógica do sistema usa 3 variáveis principais para cada tanque:

- `pct_x`: Depois de lidos os valores analógicos, os mesmos são convertidos em porcentagem e carregados nas variáveis `pct_a` e `pct_b`. Essas leituras são realizadas dentro do loop principal e ocorrem a cada 100ms
- `pct_alm_x`: Pra cada tanque, um setpoint de alarme é definido. Essas variáveis são carregadas inicialmente com o valor de 30%, gerando alarmes sempre que a carga for inferior a esse valor. O usuário tem a possibilidade de alterar esse valor usando os botões A e B.
- `alm_x`: Esta variável é usada como controle. Após converter os valores e fazer as comparações, ela é utilizada para armazenar o status de cada alarme, recebendo o valor `NORMAL`, sempre que o valor lido estiver acima do setpoint e `ALARM` para indicar que o valor lido está abaixo do esperado, neste caso, os alarmes sonoro e visual são disparados. Durante a ocorrência de alarme, se o usuário pressionar o botão referente ao tanque em alarme, esta variável recebe o valor `CLEARED`, indicando que o usuário já está ciente da ocorrência. Este estado tem o objetivo de silenciar o alarme sonoro, mesmo que o valor de leitura ainda não tenha voltado ao patamar normal. Os valores `NORMAL`, `ALARM` e `CLEARED` foram organizados em um enum, para tornar o código mais limpo e eficiente.

Execução do Projeto

A definição do projeto teve o objetivo de atender dois aspectos principais: atender uma demanda real de mercado; e ser um sistema que consiga utilizar a maior quantidade de recursos possível do BitDogLab, envolvendo PWM, sensores analógicos, display, etc.

O Hardware escolhido, girando em torno das capacidades da placa disponível, também tinha foco em apresentar bom custo-benefício, com o intuito principal de aumentar a viabilidade do projeto. Por isso, como sensor de nível foi estabelecido o uso de células de carga, que apresentam custo baixíssimo com relativa facilidade de aplicação.

Após o desenvolvimento, testes de bancada foram realizados usando o joystick para simular as leituras dos sensores em campo. O projeto demonstrou excelente comportamento dentro da expectativa, atuando com os alarmes nos momentos devidos e proporcionando fácil leitura dos dados para acompanhamento do usuário.

Para uma próxima etapa, seria de grande valia a implementação de métodos de conexão, como o WiFi, já disponível na placa, para envio de dados a um servidor de aplicação que seria capaz de armazenar os dados gerados, cruzando com diversas outras informações e dando ao sistema a capacidade de, além da geração de históricos, facilitar tomadas de decisões mais inteligentes, baseadas num conhecimento mais profundo do comportamento dos materiais estocados ao longo do tempo.

O projeto mostrou resultados bastante positivos, possibilitando ao gestor desse tipo de estoque o acompanhamento em tempo real da situação dos produtos armazenados. Com uma interface de fácil interpretação e com os alarmes para destacar situações críticas, mostrou excelente confiabilidade, focando nas informações mais importantes para esse tipo de operação.

Referências

DE CASTRO, Giovanni. **Criando uma balança com Arduino**. Robocore. Disponível em <https://www.robocore.net/tutoriais/celula-de-carga-hx711-com-arduino?gad_source=1&gclid=Cj0KCQiAq-u9BhCjARIsANLj-s3p509c5OJUHJ7Rbfhx7PKnzqJxNW4NKUuH3pdTWlyj2lurzxH7WHwaAj0iEALw_wcB> Acessado em 20/02/2025

CUGNASCA, Carlos Eduardo. **Projetos de Sistemas Embarcados**. Disponível em <<https://integra.univesp.br/courses/2710/pages/texto-base-projetos-de-sistemas-embarcados-%7C-carlos-eduardo-cugnasca>> Acessado em 22/02/2025

Monitoramento de Nível do Tanque: 3 vantagens de uma solução sem fio. Disponível em <https://www.bannerengineering.com/br/pt/company/expert-insights/What-Are-the-Advantages-of-a-Wireless-Tank-Monitoring-System.html?srsltid=AfmBOorSFh_vZ5l0qgfn3xu99de0u37ae9mlgDdFTAKQYEGhu6p8Zmc5> Acessado em 23/02/2025

BOJORGE, Ninoska. Simbologia e Nomenclatura de Instrumentação e Controle. Disponível em https://www.professores.uff.br/ninoska/wp-content/uploads/sites/57/2017/08/Aula02_Instrumen_Nomenclat_2sem2014.pdf Acessado em 23/02/2025

RP2040 Datasheet. Disponível em <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf> Acessado em 23/02/2025