

Proposta - PJI3

Projeto: Medidor de Validade de Filtro de Água

Guilherme Medeiros
Jhonatan Constantino

Motivação

Em locais com acesso de muito público é comum estarem dispostos filtros de água para o consumo, ficando dispostos pelo ambiente. Estes filtros possuem uma capacidade máxima de filtragem, um prazo de validade de garantia de água limpa, que é determinado pela quantidade de água que já foi filtrada.

Como os filtros não são inteligentes, normalmente se faz uma média do tempo que o filtro demoraria (considerando o consumo do local) para ter sua capacidade de filtrar expirada e se usa esse tempo para se fazer a troca ou manutenção do filtro. Entretanto, a média é um padrão muito estático e é necessário uma forma de fazer essa medida considerando a variável original: A quantidade de água. Este sistema tem o potencial de diminuir o desperdício e de evitar que pessoas tomem água de um filtro que já não limpa a água como o prometido.

Um sistema inteligente de medição de validade de filtros deve ser capaz de compreender quanta água já foi filtrada desde seu último *reset*, mostrar que a validade do filtro está perto do fim, além de ser capaz de ser configurável pela validade máxima do filtro (em quantidade de litros de água). Além disso, como o sistema é feito para um usuário final leigo, é necessário que possua uma interface simples e um manual explicativo.

Sobre o sistema:

O sistema deve ser capaz de, primeiramente, medir através de um sensor de vazão a quantidade de água que já passou pelo filtro.

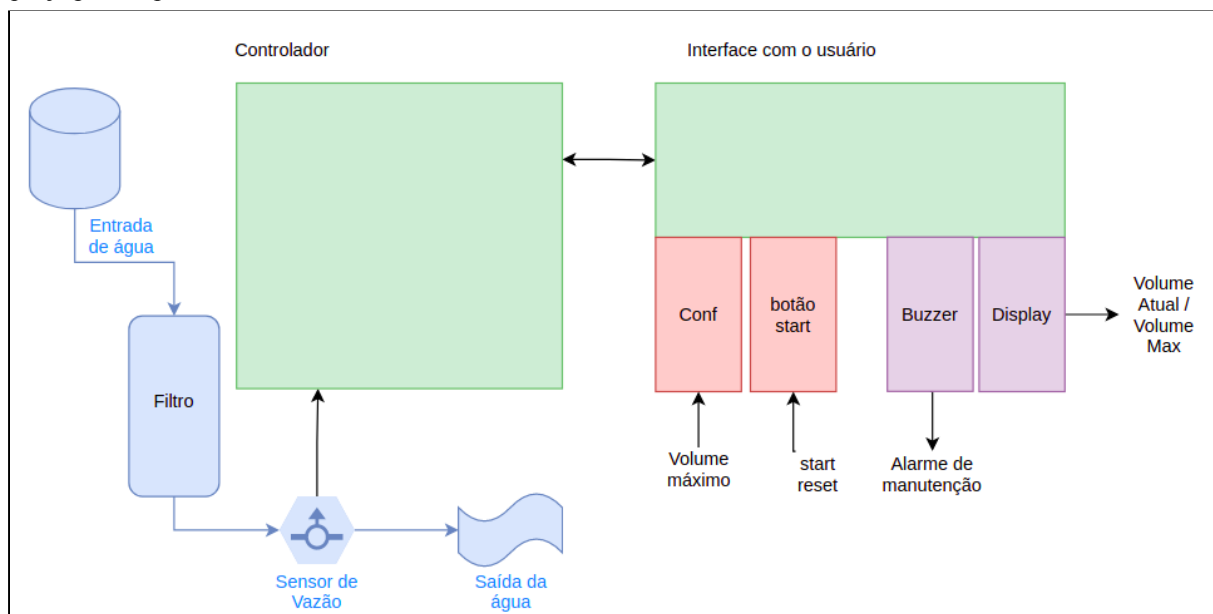


Figura 1: Diagrama de blocos do sistema

Com a medição em funcionamento, ele deve ser capaz de, em termos de interface de saída:

- Mostrar em um display a quantidade de litros que já passaram e comparar com a quantidade de litros máximo do filtro.
- Apresentar com um *buzzer* (componente sonoro) um alarme quando faltarem poucos litros para o fim da validade do filtro.

Já em especificações de entrada, o sistema deve ser facilmente configurável e não apresenta muitas interfaces de entrada, apenas:

- Botões de *start* e *reset*. O primeiro liga e desliga o sistema, que tem seu funcionamento totalmente interrompido e reiniciará suas variáveis quando religado, o segundo faz a ação de reinicializar sem necessidade de desligar o controlador.
- Uma interface de configuração, onde será permitido configurar a quantidade de litros máximos que o filtro é capaz de limpar. Após uma configuração efetuada, o sistema é reinicializado. Essa interface deve ser mediada via 3 botões, um para começar e terminar a configuração, um para aumentar o número máximo de litros e um para diminuir. A interface visual deve mostrar o andamento da configuração.

O sistema pode ser melhor compreendido com a figura 1, que mostra em um diagrama de blocos todo o sistema descrito.

Metodologia de implementação e atividades:

O projeto pode ser dividido em X partes quanto a seu desenvolvimento:

1. **Escolha do hardware e estudo de bibliotecas:** É necessário escolher os componentes que participarão do sistema, são eles:
 - a. Um microcontrolador.
 - b. Um sensor de vazão.
 - c. Uma tela (mostrador LCD ou displays de 7 segmentos)
 - d. Um buzzer.

Com os componentes escolhidos, já é possível começar a estudar as bibliotecas que serão utilizadas para programar o funcionamento do produto.

2. **Testes experimentais:** Testar os componentes fisicamente junto ao microcontrolador e entender em uma perspectiva prática como eles funcionam.
3. **Implementação:** Após a fase de estudos e testes iniciais, é iniciada a implementação do sistema, é necessário implementar, nessa ordem:
 - a. Base do controlador escolhido.
 - b. Entrada de dados do sensor de vazão.
 - c. Comparação dos dados de sensor de vazão com um dado limite colocado via código.
 - d. Implementação do *start* e do *reset*.
 - e. Desenvolvimento da entrada de dados de configuração (volume máximo).
 - f. Interfaces de saída.
4. **Testes:** É necessário realizar testes em cada uma das etapas da sessão 3 (a,b,c,d,e,f), para garantir o funcionamento das partes, entretanto o produto final deve ser devidamente testado utilizando uma situação real.
5. **Documentação:** Como é um produto utilizado por usuários finais, este projeto contará com uma documentação detalhada e simples sobre como usar e configurar o sistema. Além disso, uma segunda documentação será feita sobre a parte técnica de desenvolvimento, detalhando pinagem, funções de

código e componentes utilizados. Esta segunda documentação será de uso da empresa, caso o projeto tenha a possibilidade de ser ampliado.

Metodologia de desenvolvimento

Para este projeto já foi criado uma *board* no *trrello* com cada atividade descrita na seção anterior. Este *board* deverá ser seguido para garantir a implementação do projeto até o fim do prazo. Um link para a visualização dele pode ser encontrado [aqui](#). As tarefas foram divididas em pré desenvolvimento, desenvolvimento, homologação e documentação e estão todas descritas e datadas no board que é público e pode ser visualizado facilmente.

Orçamento

O orçamento em sua totalidade pode ser visto na tabela 1.

Quantidade	Nome	Valor	Valor total
1	Microcontrolador	150	150
1	Sensor de vazão	50	50
9	Botões	3	27
1	Display	30	30
1	Buzzer	10	10
52	Horas de dev	90	4680
Total			4947

Tabela 1: Orçamento

Documentação

Serão produzidas duas documentações para este projeto, são elas:

1. Documentação técnica:

Um documento para a empresa de desenvolvimento que será mantido em sigilo, este contará com aspectos técnicos relevantes para o desenvolvimento do produto. Aqui serão listadas as bibliotecas utilizadas, assim como os componentes físicos do produto. Além disso, estarão aqui diagramas de bloco detalhando o programa desenvolvido e a pinagem. O objetivo deste documento é facilitar o trabalho caso o produto seja expandido e/ou ajudar no *onboarding* de novos membros no time de desenvolvimento.

2. Documentação do usuário:

Esta documentação vai para o usuário que irá gerenciar o sistema no local onde será implantado, ele será de fácil compreensão e contará com detalhes sobre como fazer a instalação, como configurar o volume fácil, um detalhamento do funcionamento de cada botão e entrada de dados e deverá ser escrito de maneira clara visando a compreensão de cada parte do produto.

Cronograma de desenvolvimento

Este cronograma pode ser melhor compreendido acessando o *board* no [trelo](#) feito para o desenvolvimento do produto. Entretanto, abaixo segue um cronograma detalhado de cada atividade:

Atividade	Começo	Fim
Escolha do hardware e bibliotecas	10/05	14/05
Pré testes e experimentos	14/05	17/05
Dev: Controlador e vazão	17/05	23/05
Dev: Start e Reset	23/05	30/05
Dev: Entrada VolMax	30/05	13/06
Def: Interface Visual	13/06	04/07
Homologação final	04/07	12/07
Documentação e entrega	12/07	19/07

Tabela 2: Cronograma