

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA  
CURSO DE ENGENHARIA ELÉTRICA

GABRIELLE DAS VIRGENS LIMA  
GUSTAVO VINÍCIUS GARCIA DOS REIS  
STEPHEN DA SILVA MADEIRA

**PROJETO DE SISTEMA DE CONTROLE E MONITORAMENTO DE  
RESERVATÓRIOS DE ÁGUA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2020

GABRIELLE DAS VIRGENS LIMA  
GUSTAVO VINÍCIUS GARCIA DOS REIS  
STEPHEN DA SILVA MADEIRA

**PROJETO DE SISTEMA DE CONTROLE E MONITORAMENTO DE  
RESERVATÓRIOS DE ÁGUA**

Trabalho de Conclusão de Curso de Graduação em Engenharia Elétrica apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR) como requisito para obtenção do título de "Engenheiro(a) Eletricista".

Orientador: Prof. Dr. Glauber Gomes de Oliveira Brante

**CURITIBA**

**2020**

GABRIELLE DAS VIRGENS LIMA  
GUSTAVO VINÍCIUS GARCIA DOS REIS  
STEPHEN DA SILVA MADEIRA

## Projeto de Sistema de Controle e Monitoramento de Reservatórios de Água

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro Eletricista, do curso de Engenharia Elétrica do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 16 de novembro de 2020.

---

Prof. Paulo Sérgio Walenia, Dr.  
Coordenador de Curso  
Engenharia Elétrica

---

Profa. Annemarle Gehrke Castagna, Mestre  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia Elétrica do DAELT

### **ORIENTAÇÃO**

---

Glauber Gomes de Oliveira Brante, Dr.  
Universidade Tecnológica Federal do Paraná  
Orientador

### **BANCA EXAMINADORA**

---

Adriano Ruseler, Dr.  
Universidade Tecnológica Federal do Paraná

---

Glauber Gomes de Oliveira Brante, Dr.  
Universidade Tecnológica Federal do Paraná

---

Joaquim Eloir Rocha, Dr.  
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia Elétrica

Agradecemos em primeiro lugar a Deus que iluminou o nosso caminho durante esta caminhada.

Ao professor Glauber Gomes de Oliveira Brante pela paciência na orientação e incentivo que tornaram possível a conclusão deste trabalho.

## **AGRADECIMENTOS**

Primeiramente a Deus, que a todos nós até aqui sustentou, nos dando sabedoria e saúde, e nos protegendo da atual pandemia.

Eu, Stephen da Silva Madeira, deixo nesta página, mas não que nela se resuma, os agradecimentos devidos aos meus pais, maiores incentivadores ao meu ingresso neste curso, sustentando-me com o zelo que os pais podem dar e suporte financeiro necessário. E à minha noiva, Dr<sup>a</sup> Mayra Husein Leite, que além de ser refúgio nos tempos difíceis desta formação por meio do seu companheirismo, doçura, amor e paciência, agraciou-nos ainda com um osciloscópio digital, equipamento amplamente utilizado na realização deste trabalho, poupando-nos preciosas horas de trabalho e recursos financeiros. Deixo ainda aqui registrado meu muito obrigado a todos os professores e profissionais do Colégio Militar de Curitiba, cada docente que contribuiu de inúmeras maneiras na formação de quem sou hoje, e me deu muitas bases utilizadas ao longo da formação como Engenheiro.

Eu, Gustavo Vinícius Garcia dos Reis, honro o fechamento desse ciclo dedicando este trabalho aos meus pais, por todo o suporte que a mim dedicaram no decorrer da vida e viram a culminar nessa conquista. A todos os professores da UFPA e UTFPR que de alguma forma instigaram minha curiosidade e busca pelo conhecimento e verdade.

Eu, Gabrielle das Virgens Lima, dedico este trabalho aos meus pais, por sempre me incentivarem a conquistar meus objetivos e aos meus amigos que tornaram o período da faculdade em um dos melhores momentos da minha vida.

Por fim, nosso reconhecimento ao trabalho de todos os professores envolvidos em nossa formação; ao nosso orientador e à banca, responsáveis pelo progresso deste trabalho de conclusão e de curso; e aos servidores, funcionários terceirizados e todos os que demais que, dia após dia, mantém esta célebre instituição em condições de formar profissionais distintos em nossa amada nação.

”Não creio que haja uma emoção, mais intensa para um inventor do que ver suas criações funcionando. Essa emoção faz você esquecer de comer, de dormir, de tudo.”

- Nikola Tesla

## RESUMO

MADEIRA, Stephen da Silva; REIS, Gustavo Vinícius Garcia dos; LIMA, Gabrielle das Virgens. PROJETO DE SISTEMA DE CONTROLE E MONITORAMENTO DE RESERVATÓRIOS DE ÁGUA. 67 f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

O fornecimento hídrico pelas companhias de saneamento no Brasil não é ininterrupto e vem se tornando limitado. Ter conhecimento dessas interrupções, bem como uma estimativa do restante de água disponível pode auxiliar as empresas e a população na administração dos seus reservatórios. O mercado nacional, no entanto, carece de soluções para o monitoramento hidráulico. Este trabalho, portanto, consiste em validar a possibilidade de desenvolver um sistema autônomo e de baixo custo para o controle e o monitoramento de um reservatório de água, independentemente de sua geometria, bem como identificar interrupções no fornecimento. Para a sua realização foi desenvolvido um protótipo, constituído de dois módulos, sendo o primeiro responsável pela aquisição dos dados, o processamento das informações e o acionamento da válvula de controle. O segundo módulo, responsável pela interação com o usuário. Com isso, por meio dos resultados obtidos incentivar novos trabalhos voltados ao uso consciente de água.

**Palavras-chave:** monitoramento, reservatório, controle, água

## ABSTRACT

MADEIRA, Stephen da Silva; REIS, Gustavo Vinícius Garcia dos; LIMA, Gabrielle das Virgens. WATER RESERVOIR CONTROL AND MONITORING SYSTEM DESIGN. 67 f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

Water supply by sanitation companies in Brazil is not uninterrupted and has become limited. Being aware of these interruptions, as well as an estimate of the remaining available water can assist companies and the population in the administration of their reservoirs. The domestic market, however, lacks solutions for hydraulic monitoring. This paper consists of validating the possibility of developing a standalone and low-cost system for the control and monitoring of a water reservoir, regardless of its geometry, as well as identifying interruptions in supply. For its realization, a prototype was developed, consisting of two modules, the first responsible for data acquisition, processing of information, and the activation of the control valve. The second module, responsible for the interaction with the user. Thus, the results obtained encourage new studies aimed at the conscious use of water.

**Keywords:** monitoring, reservoir, control, water



## LISTA DE FIGURAS

FIGURA 1	– Composição geral do sistema .....	13
FIGURA 2	– Plataformas consideradas para o desenvolvimento do trabalho .....	14
FIGURA 3	– Sensores avaliados e tubo de Venturi; dispositivo que juntamente sensores de pressão, pode ser utilizado para medir fluxo de fluídos .....	15
FIGURA 4	– Módulos de comunicação <i>wireless</i> avaliados .....	15
FIGURA 5	– Módulo OLED monocromático de 128x64 <i>pixels</i> de resolução .....	17
FIGURA 6	– <i>Buzzer</i> eletromagnético de modelo similar ao utilizado no projeto .....	18
FIGURA 7	– LED de única cor em encapsulamento 1206 .....	19
FIGURA 8	– <i>Encoder</i> incremental com <i>push button</i> .....	19
FIGURA 9	– Conversor CC-CC encapsulado .....	21
FIGURA 10	– Tipos de sensores de fluxo .....	23
FIGURA 11	– Sensor de fluxo escolhido .....	24
FIGURA 12	– Válvula solenoide .....	26
FIGURA 13	– Sensor de nível de líquidos .....	26
FIGURA 14	– Terminação apropriada do RS-485 .....	30
FIGURA 15	– Cabo Automação padrão RS485 com blindagem .....	31
FIGURA 16	– Diagrama esquemático do MIU .....	33
FIGURA 17	– Diagrama esquemático do MAPA .....	34
FIGURA 18	– CAD do circuito impresso no <i>software</i> EAGLE .....	35
FIGURA 19	– Fotos do protótipo do MIU (Superior e Inferior) .....	36
FIGURA 20	– Fotos do protótipo do MAPA (Superior e Inferior) .....	36
FIGURA 21	– Elementos do protótipo .....	37
FIGURA 22	– Exemplo de mensagem exibida no <i>display</i> . Nota-se o LED de Alerta (D4) aceso .....	41
FIGURA 23	– Tela Principal .....	42
FIGURA 24	– Parâmetros indicados na tela .....	42
FIGURA 25	– Telas Ajuste_Volume e Ajuste_Tempo, respectivamente .....	43
FIGURA 26	– Telas de saída das telas de ajuste .....	44
FIGURA 27	– Sinal registrado no osciloscópio digital, mostrando a comunicação entre MIU e MAPA .....	53
FIGURA 28	– Montagem do sistema para simulação do funcionamento .....	54
FIGURA 29	– LED de “Reservatório esvaziando” e “Reservatório enchendo” acesos (em destaque), respectivamente .....	57

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>10</b>
1.1 TEMA	10
1.1.1 Delimitação do Tema	11
1.2 PROBLEMAS E PREMISSAS	11
1.3 OBJETIVOS	12
1.3.1 Objetivo Geral	12
1.3.2 Objetivos Específicos	12
1.4 JUSTIFICATIVA	12
1.5 VISÃO GERAL	13
<b>2 ESPECIFICAÇÃO DE REQUISITOS</b>	<b>16</b>
2.1 MIU (MÓDULO DE INTERAÇÃO COM O USUÁRIO)	16
2.1.1 <i>Display</i>	16
2.1.2 <i>Buzzer</i>	17
2.1.3 LEDs	18
2.1.4 <i>Encoder</i>	19
2.1.5 Alimentação	19
2.1.6 Microcontrolador	21
2.2 MAPA (MÓDULO DE AQUISIÇÃO, PROCESSAMENTO E ATUAÇÃO)	23
2.2.1 Sensor de fluxo	23
2.2.2 Eletroválvula	25
2.2.3 Sensor de nível	25
2.2.4 LEDs, Microcontrolador e Alimentação	27
2.3 COMUNICAÇÃO	28
2.3.1 Meio de Transmissão	28
2.3.2 Protocolo de Comunicação	29
<b>3 PROTÓTIPO</b>	<b>32</b>
3.1 PROTOTIPAGEM DOS MÓDULOS MIU E MAPA	32
3.2 DESAFIOS ENCONTRADOS DURANTE O DESENVOLVIMENTO	34
<b>4 EMULAÇÃO E RESULTADOS</b>	<b>38</b>
4.1 MIU	38
4.2 MAPA	46
4.3 EMULAÇÃO	53
<b>5 CONCLUSÃO</b>	<b>58</b>
<b>REFERÊNCIAS</b>	<b>60</b>
<b>Apêndice A – FIRMWARE</b>	<b>63</b>
A.1 INICIALIZAÇÃO	63
A.2 LÓGICA DO <i>ENCODER</i>	63
A.3 ESTADO DA CHAVE DO <i>ENCODER</i>	64
A.4 CONTROLE DO CICLO ATIVO DO PWM	66

# 1 INTRODUÇÃO

## 1.1 TEMA

O fornecimento de água potável pelas concessionárias locais no Brasil não é ininterrupto, seja por falhas, acidentes, condições pluviométricas desfavoráveis ou até mesmo manutenção programada. Baseado nisto, é comum nas edificações brasileiras abastecidas pelo sistema indireto (com ou sem bombeamento) a necessidade e, por consequência, a existência de, pelo menos, um reservatório de água que atenda à Norma Técnica ABNT NBR 5626 - Instalação Predial de Água Fria e às normas da companhia de saneamento local. Desta forma, garante-se a potabilidade e fornecimento regular de água, ainda que cesse temporariamente o suprimento desta pela concessionária (JÚNIOR, 2014).

De acordo com a Companhia de Saneamento do Paraná (SANEPAR) (2017, p. 9):

”O volume de reservação deverá ter capacidade de armazenamento de 1 a 2 dias da demanda de consumo prevista, para precaver-se das interrupções dos serviços devido a manutenção e/ou casos fortuitos, atendendo aos prazos prováveis necessários à normalização dos serviços.”

Saber quando o fornecimento hídrico for interrompido e ter uma estimativa do restante de água no reservatório pode auxiliar os usuários a melhor administrá-lo, garantindo que a água dure o previsto pelo projeto hidrossanitário.

Em 2018, durante estágio obrigatório, dois dos integrante deste trabalho, perceberam a necessidade de uma solução *stand-alone* para tal problema em um prédio de um centro educacional, que almejava certificação de construção sustentável, localizado no centro da cidade de Curitiba. Os responsáveis pelo projeto do prédio solicitaram um monitoramento pelo sistema de automação predial do consumo de água, e a empresa contratada para prestar os serviços de automação, onde os dois integrantes supracitados estagiaram, não encontrou no mercado uma solução pronta. Assim sendo, necessitou-se recorrer à programação de um terceiro equipamento para realizar esta tarefa.

### 1.1.1 DELIMITAÇÃO DO TEMA

Este trabalho busca, então, desenvolver um sistema autônomo, microprocessado, capaz de monitorar em tempo real o estado de um reservatório de água e o estado do fornecimento de água pela companhia de saneamento local. Esse sistema deve informar o usuário em caso de interrupção do fornecimento, bem como prover estatísticas de consumo e previsão da duração da água armazenada. Diferenciando-se assim, das soluções encontradas no mercado nacional, que provêm apenas informações discretas sobre o nível de água no sistema de armazenamento.

### 1.2 PROBLEMAS E PREMISSAS

Nos últimos anos registrou-se, no Brasil, um aumento no número de regiões adotando o racionamento no uso da água por conta de crises hídricas, como é mostrado no Balanço das Águas realizado pela Agência Nacional de Águas (ANA) entre os anos de 2015 e 2016 (ANA, 2015/2016). Em decorrência da estiagem, o nível de água armazenada nos reservatórios ao longo do país, principalmente no Nordeste, tem reduzido com o passar dos anos colocando a disponibilidade hídrica em uma péssima situação.

Como o índice de precipitação não é suficiente para manter o estoque de água nos reservatórios, a ANA se propõe intensificar estrategicamente o acompanhamento desses níveis e estabelecer regras que restringem o uso de água em determinadas regiões pelo país, para que o sistema de abastecimento não entre em colapso e deixe de atender a população de forma integral.

Porém a crise hídrica não está somente relacionada às estatísticas pluviométricas, mas também às perdas que ocorrem no sistema de abastecimento e que são consequência de uma rede de distribuição pouco eficiente. No Brasil, pouco mais de 38,5% da água tratada é perdida na etapa de distribuição, de acordo com uma pesquisa feita em 2018 avaliando as 100 maiores cidades do país em relação a quantidade de perdas, cidades estas que abrangem cerca de 40% da população brasileira (ASSOCIADOS, 2018).

Neste contexto, o controle e o gerenciamento de perdas através de ações de conservação de energia e água se tornam indispensáveis para as empresas e para a população em geral. Segundo Keeling (2012), o gerenciamento de perdas precisa primariamente de medidas que sejam a base para o gestor analisar os gastos e propor soluções (desligamento de equipamentos, substituição de equipamentos, alteração de processos produtivos, políticas de conscientização, entre outros).

Assim, o presente trabalho se faz necessário para um melhor entendimento dos aspectos práticos relacionados à implementação de um sistema de gerenciamento do reservatório de água em uma construção. Além disso, o usuário poderá ter acesso a informações mais específicas a respeito do seu consumo e quantidade de água disponível em caso de uma falha no abastecimento.

Tem-se como premissas a viabilidade de mensurar volume de água com um sistema eletrônico, e assumindo uma base de tempo constante, exibir dados baseados nestes de forma confiável e de que há interesse comercial em tal solução proposta.

### 1.3 OBJETIVOS

#### 1.3.1 OBJETIVO GERAL

O presente trabalho possui o objetivo central de desenvolver um aparato de instrumentação de baixo custo para o uso no controle e monitoramento do consumo de água de um reservatório a ser aplicado em uma instalação.

#### 1.3.2 OBJETIVOS ESPECÍFICOS

- Verificar, através de pesquisa bibliográfica, as tecnologias existentes que podem ser aplicadas para o desenvolvimento deste trabalho;
- Escolher, dentre as tecnologias pesquisadas, a mais viável técnico e financeiramente para implementação deste sistema;
- Buscar um método de quantificar o volume de água armazenada em um reservatório;
- Desenvolver *hardware* e *firmware* capazes de receber e processar os dados obtidos em medição;
- Implementar sistema de controle de abastecimento para o reservatório;
- Ensaiar e analisar o funcionamento do sistema.

### 1.4 JUSTIFICATIVA

Pretende-se desenvolver este equipamento devido à demanda de um grande cliente de um sistema, que não há pronto no mercado nacional para esta aplicação, de monitoramento

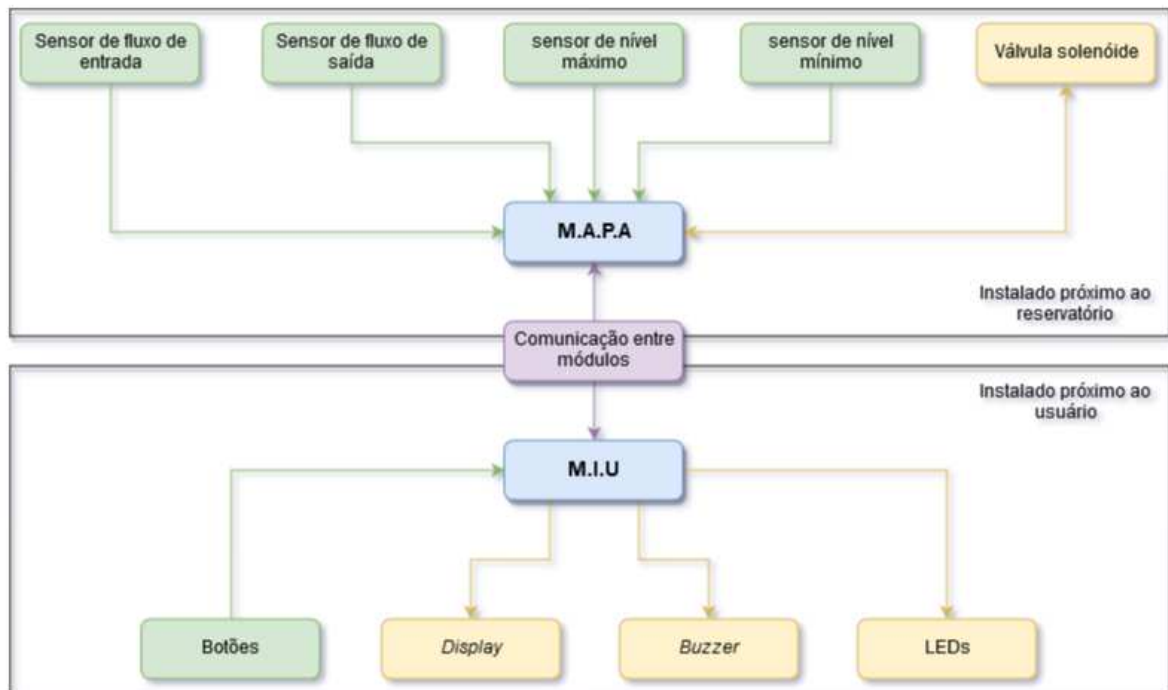
hidráulico para instalações hidrossanitárias prediais, e pela possibilidade de ofertar-se além do básico procurado.

Este trabalho abre, ainda, a possibilidade e necessidade de criação de sensores e/ou métodos capazes de monitorar grandes volumes e/ou fluxos de líquidos, podendo levar outros trabalhos a oferecer mais soluções ao mercado nacional e internacional, movimentando assim, o desenvolvimento tecnológico no Brasil. Por fim, a flexibilidade da ideia permite que o sistema seja empregado não apenas a reservatórios de água, mas também o seu emprego em qualquer reservatório de líquidos, como combustíveis, reagentes na indústria química, matéria-prima na indústria alimentícia, etc., o que nos motiva a desenvolver uma solução facilmente adaptável às necessidades que surgirem.

## 1.5 VISÃO GERAL

A Figura 1 apresenta uma visão geral e preliminar do sistema como um todo.

**Figura 1: Composição geral do sistema.**



**Fonte: Autores**

O sistema será dividido em dois módulos distintos, sendo um chamado de M.A.P.A. (Módulo de Aquisição, Processamento e Acionamento), responsável por fazer a aquisição dos dados dos sensores, enviá-los pré-processados ao outro módulo e realizar o acionamento da válvula solenóide. Este dispositivo deverá ser instalado próximo ao reservatório.

O outro módulo, denominado M.I.U. (Módulo de Interação com o Usuário) receberá os dados do M.A.P.A, interpretará e exibirá no *display*, acionará LEDs informando estados e falhas, também comunicadas ao usuário também por meio de um *buzzer* e, por fim, receberá entradas do usuário para configurações como o volume máximo do reservatório.

Para a elaboração dos módulos, foram considerados microcontroladores/placas de desenvolvimento disponíveis no mercado, atentando às arquiteturas, periféricos, custo e demais requisitos do projeto. Dentre as opções a serem avaliadas, estavam as plataformas *Raspberry Pi*, *Tiva™ C Series LaunchPad* e os microcontroladores PIC, respectivamente apresentados na Figura 2.

**Figura 2: Plataformas consideradas para o desenvolvimento do trabalho.**



**Fonte: (RASPBERRY, 2019; MOUSER, 2019; TME, 2019)**

Como sensores (Figura 3), foram estudados os de Efeito Hall, bem como outras alternativas: como sensores de distância, pressão, dentre outros. Dessa forma, foi decidida qual a melhor solução a ser empregada na validação da ideia do projeto.

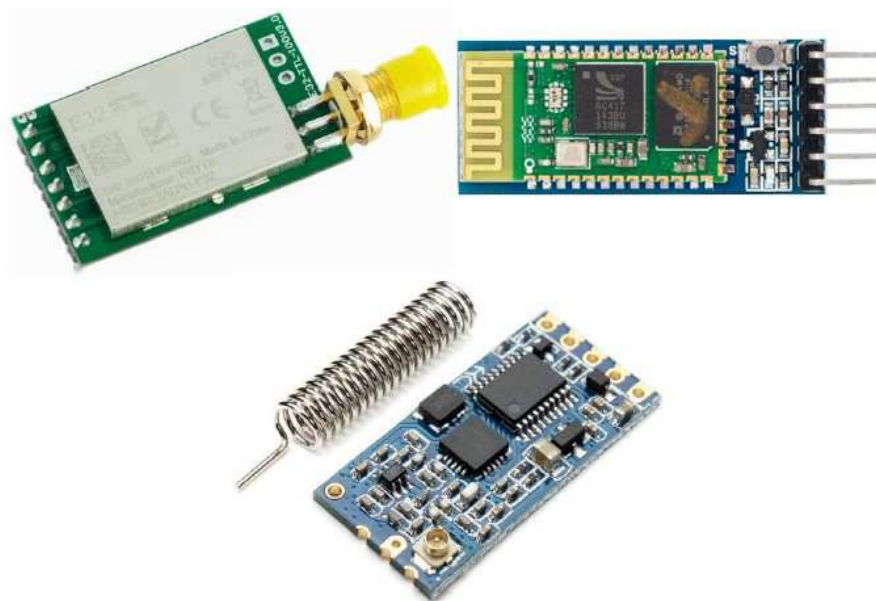
Por fim, foi avaliada qual seria a melhor maneira de realizar a comunicação entre os módulos, tendo como opções o par trançado, *bluetooth*, radiofrequência (RF) e padrão 802.11. Na Figura 4 são apresentados três módulos para conexão sem fio: módulo *wireless* LoRa, módulo *bluetooth* HC-05 e módulo RF HC-12, respectivamente. Não foram consideradas alternativas que envolvessem conexão com a *internet*, pois isto feriria a intenção de desenvolver um sistema capaz de operar por si só, sem depender de rede senão própria.

**Figura 3: Sensores avaliados e tubo de Venturi; dispositivo que juntamente sensores de pressão, pode ser utilizado para medir fluxo de fluídos.**



**Fonte: (MERCADO, 2017)(HUINFINITO, 2017)(INDIAMART, 2017)**

**Figura 4: Módulos de comunicação *wireless* avaliados.**



**Fonte: (FILIPEFLOP, 2017)**



## 2 ESPECIFICAÇÃO DE REQUISITOS

Este capítulo busca apresentar as especificações do projeto em nível de *hardware*, discutindo as decisões tomadas sobre os componentes, focando nas funcionalidades que devem ser implementadas no equipamento em um primeiro momento.

Para todas as partes componentes do sistema, as decisões estarão alicerçadas em sete princípios:

- Suprir as necessidades daquela parte;
- Estar comercialmente disponível ou poder ser facilmente adaptada;
- Ser de mais baixo custo possível;
- Possuir documentação clara e de fácil acesso;
- Consumir o mínimo de energia elétrica possível;
- Possuir qualidade para uma operação contínua e duradoura;
- Ser o mais compacta possível.

### 2.1 MIU (MÓDULO DE INTERAÇÃO COM O USUÁRIO)

A partir da Figura 1, apresentada no capítulo anterior e que ilustra uma visão generalista do sistema em blocos, abordaremos as necessidades, especificações e escolhas para cada parte do sistema, a começar pelo MIU.

#### 2.1.1 *DISPLAY*

O *display* é uma forma conveniente e compacta de apresentar informações a um usuário, quando este quiser visualizá-los. A existência de um módulo capaz de exibir informações dispensa o usuário de ter que acessar outras opções, como por exemplo, seu celular.

No mercado existem diversas tecnologias, como *displays* de segmentos de LEDs, LCD, *e-ink*, OLED, entre outras diversas tecnologias de LCD. Ao considerar o custo, versatilidade, disponibilidade no mercado e disponibilidade de informações dos fabricantes para o uso, os *displays* de LCD e OLED ganham destaque, porém, pelo tamanho, ângulo de visão, consumo de energia, brilho e contraste, a tecnologia OLED acaba sendo a escolhida para este projeto. Este tipo de *displays* é facilmente encontrado no mercado já montado em módulo contendo um controlador e todos os componentes necessários. Um exemplo de controlador largamente empregado é o SSD1306, capaz de controlar 128x64 *pixels* em *displays* monocromáticos, que será a escolha para este projeto. Um exemplo destes módulos é apresentado na Figura 5. Utilizar uma tela com cores não trará vantagens funcionais ao equipamento, elevando desnecessariamente os custos apenas para exibir as mesmas informações com cores.

**Figura 5: Módulo OLED monocromático de 128x64 *pixels* de resolução.**



Fonte: (WINSTAR, 2020)

### 2.1.2 BUZZER

Outra forma de alertar o estado do sistema é por meio de aviso sonoro. Este recurso também foi incorporado ao projeto, sendo sua principal função emitir os alertas propostos ao usuário, assim, o mesmo saberá, mesmo longe do MIU, que há uma informação relevante a ser consultada.

Existem diversas opções para comunicar informações entre um produto e o usuário. Uma das escolhas mais comuns para a comunicação por áudio é um *buzzer* (ROSE, 2018). Como não é pretendido reproduzir sons com alta fidelidade, mas sim gerar tons simples, o preço, tamanho e simplicidade do circuito associado ao uso de um *buzzer* o tornam a escolha ideal.

Existem duas tecnologias disponíveis: piezoelétricos e eletromagnéticos. Os que empregam a primeira tecnologia requerem tensões de 12 V a 220 V, correntes inferiores a 20 mA e são capazes de produzir maior nível de pressão sonora do que outros tipos, como o eletromagnético, sendo este acionado por corrente, tipicamente maior do que 20 mA, mas com tensões inferiores aos piezoelétricos, na faixa de 1,5 V a 12 V. Podem ser controlados por tensão, sendo que a impedância da bobina que o constitui junto à tensão aplicada determinará a corrente circulante por esta (ROSE, 2018). A tensão de acionamento e tamanho físico nos aponta o do tipo eletromagnético como sendo a escolha ideal para nosso cenário. O modelo de referência para o que será utilizado no projeto é mostrado na Figura 6.

**Figura 6:** *Buzzer* eletromagnético de modelo similar ao utilizado no projeto.



**Fonte:** (DIGIKEY, 2017)

### 2.1.3 LEDS

O último método de informação ao usuário também será visual, entretanto, representará estados discretos, assim, informações básicas podem ser rapidamente obtidas. Entretanto, diferente de alguns equipamentos que utilizam apenas um LED capaz de gerar diversas cores e estas informam os diferentes estados, utilizaremos um LED para cada estado. Esta decisão foi tomada considerando a acessibilidade para pessoas acometidas pelo daltonismo, sendo esta a situação de um dos integrantes deste trabalho.

Cada estado será representado por um LED, com cores diferentes, mas estas serão apenas para efeito estético, não sendo sua cor, mas sim sua posição que identificará o estado informado. O encapsulamento de LED pretendido para o projeto é o 1206, idêntico ao da Figura 7.

**Figura 7: LED de única cor em encapsulamento 1206.**



**Fonte: (OPTOELETRONICS, 2017)**

#### 2.1.4 ENCODER

O sistema deverá permitir alguns ajustes/comandos por parte do usuário, como o volume máximo da caixa de água, o tempo que deseja que a tela fique ligada, ligar a tela e cancelar os alarmes sonoros. Para isto, poderíamos empregar chaves tácteis, que são baratas e largamente utilizadas, a tecnologia por toque (sensoriamento *touch* capacitivo) dentre outras presentes no mercado. Escolhemos empregar um *encoder* incremental (Figura 8), similar ao encontrado em painéis de aeronaves e equipamentos de áudio. São componentes simples, versáteis, de fácil integração e boa durabilidade. Com ele fica possível navegar através de opções, seleccionar a desejada e silenciar sinais sonoros tudo por meio de um único dispositivo, de forma mais simples do que utilizar múltiplas ou até mesmo uma única chave táctil ou botão por toque.

**Figura 8: Encoder incremental com push button.**



**Fonte: (CUI, 2017)**

#### 2.1.5 ALIMENTAÇÃO

A alimentação elétrica deverá ser provida pelo MAPA, por meio do cabo de comunicação. Como os componentes trabalharão em níveis de tensão mais baixos, um

regulador de tensão será necessário. Para a escolha do melhor tipo a ser empregado, as seguintes considerações foram feitas:

1. A corrente total estimada para este projeto é de 250 mA. Este valor foi obtido da seguinte forma:
  - A corrente total do *display* com todos os *pixels* acesos, microcontrolador enviando dados e um LED aceso foi medida em 66 mA. Este valor foi obtido com um multímetro na escala de 200 mV e um resistor *shunt* de 0,1 Ohms, em série com a alimentação do circuito. A queda de tensão medida foi de 6,6 mV, o que pela lei de Ohm resulta em 66 mA de corrente;
  - A corrente máxima consumida pelo *buzzer* será de 45 mA, segundo o fabricante escolhido (CEM-1206S, 2006);
  - Considerando LEDs comuns, a corrente de cada um é assumida sendo menor ou igual a 20 mA, segundo o fabricante (APTD3216LZGCK, 2017). Como podemos ter até 4 LEDs acesos ao mesmo tempo além do acima mencionado, teremos um total de 80 mA;
  - As demais correntes encontram-se na escala de micro ou nano ampères, sendo, portanto, desconsideradas;
  - A soma das correntes acima mencionadas, resultante em 191 mA será arredondada para 250 mA por segurança.
2. Segundo a nota de aplicações AN761 da Microchip (PAGLIA, 2004), a dissipação de potência, em Watts, de um regulador linear pode ser calculada como:

$$P_D = (V_{Entrada} - V_{Saida}) \cdot I_{Carga} + V_{Entrada} \cdot I_{Quiescente} \quad (1)$$

Aplicando os valores do catálogo da ST Microelectronics para o regulador L7805 (L78, 2018) e os valores de tensão e corrente necessários para o projeto, obtemos o seguinte resultado:

$$P_D = (32 - 5) \cdot 0,25 + 32 \cdot 0,006 = 6,942W. \quad (2)$$

Ainda segundo a mesma nota, a temperatura esperada na junção do regulador, considerada a potência calculada, uma temperatura ambiente de 25 °C e resistência térmica junção-ambiente de 50 °C/W, dada pelo catálogo para o encapsulamento TO-220, será de

$$P_{Junção} = T_{Ambiente} + (\Theta_{JA} \cdot P_D) = 25^{\circ}C + 50^{\circ}C/W \cdot 6,942W = 372,1^{\circ}C, \quad (3)$$

o que claramente extrapola a máxima temperatura de junção recomendada de 125 °C. O emprego de dissipadores de calor é descartado por ocupar espaço físico, o que é contra intuitivo em um projeto onde se busca as menores dimensões.

Assumiu-se uma tensão de entrada de até 32 V para que o sistema esteja compatível com os níveis de tensão compatíveis no segmento de automação.

Baseado nestes dados, um conversor CC-CC chaveado do tipo Buck e não isolado é a escolha mais plausível, dado a simplicidade, custo e eficiência energética, que pode alcançar 94%, segundo a fabricante do modelo sugerido (VX78-500, 2017). Além disto, o conversor nos dá a flexibilidade de trabalhar com tensões entre 24 e 32 Volts, tensão geralmente já presente e utilizada em sistemas de automação em geral, sem nos preocupar com a elevada dissipação de potência em um regulador linear.

Optou-se por um conversor já fabricado em vez de se escolher um CI dedicado e projetar o próprio circuito por questões de custo, simplicidade, tempo de projeto e riscos, afinal, um conversor já montado é um item a menos do projeto para apresentar falhas por erros da equipe, desde que este seja usado dentro dos limites impostos pela fabricante. Como exemplo de conversor CC-CC pronto, temos o da Figura 9.

**Figura 9: Conversor CC-CC encapsulado.**



**Fonte: (VX78-500, 2017)**

### 2.1.6 MICROCONTROLADOR

Para integrar todas as partes do sistema, permitindo a flexibilidade de alterar o comportamento deste, adicionar funcionalidades conforme possível e corrigir erros que possam vir a surgir, uma unidade computacional é a escolha mais provável. Dentre todas as opções disponíveis no mercado, como microprocessadores (*Microprocessor Unit* ou MPU), microcontroladores (MCU, do inglês *Microcontroller Unit*) e pequenos computadores, como *Raspberry Pi* e *Beagle Bone*, os que mais se adequam ao projeto são os microcontroladores.

Estes possuem, via de regra, uma Central de Processamento de Dados (*Central Processing Unit* ou CPU), memórias de diversos tipos, como RAM, ROM, EEPROM e *Flash* e periféricos, como conversores A/D e D/A, geradores de *clock*, temporizadores, interfaces/protocolos de comunicações, E/S dentre outros recursos disponíveis dependendo do modelo e do fabricante (KEIM, 2019) tudo em um único encapsulamento, barateando e simplificando o projeto.

O microcontrolador escolhido será o PIC16F18857-I/SO, da Microchip. Este foi escolhido por possuir os recursos e periféricos necessários para o projeto a um baixo preço, cotado a U\$ 1,46 na distribuidora Mouser, no dia 02/10/2020 (PIC16F18857-I/SO, 2020).

A arquitetura de 8 bits se mostra suficiente, pois alguns periféricos utilizados operam independentes do processador, característica esta chamada pela fabricante de CIP (*Core Independent Peripherals*). Alguns dos recursos que operam desta maneira são a CRC, o NCO e a PWM. Destaque para o NCO, a ser utilizado para gerar os tons reproduzidos pelo *buzzer*. O fato deste periférico ser independente da CPU nos permite produzir alarmes sonoros e realizar outras tarefas simultaneamente, sem ter que alternar o processamento.

Outros fatores que levaram a escolha deste modelo é o fato dele ser relativamente novo, sendo seu catálogo datado de 2016; ser suportado pela interface de configuração gráfica MCC (*Microchip Code Configurator*), o que auxiliará na produção do código; compatibilidade com um gravador barato e facilmente encontrado à venda, o PicKit3; e por possuir um recurso denominado PPS (*Peripheral Pin Select*), que permite remapear a conexão de periféricos com os pinos físicos do microcontrolador, o que facilitará o desenvolvimento da placa de circuito impresso (PCI).

Como processaremos poucos dados e não está prevista a utilização de aritmética de ponto flutuante, utilizar um microcontrolador de 32 bits seria subutilizar sua capacidade além de encarecer o projeto. Tomando por exemplo o microcontrolador TM4C123GH6PMIR fabricado pela Texas Instruments, contendo uma CPU Cortex-M4F da Arm, sua cotação sob as mesmas condições do microcontrolador escolhido é de U\$9,88, cerca de sete vezes maior, oferecendo mais recursos e processamento que não seriam utilizados (TM4C123GH6PM, 2020).

A memória RAM mínima foi estimada com base em um programa exemplo criado para medir a corrente elétrica máxima circulando pelo *display* escolhido. Todo o código, que utiliza biblioteca gráfica proprietária, consumiu 1147 *bytes*. A escolha de um MCU contendo 4096 *bytes*, o que nos deixa com 2949 *bytes* para o restante do programa, foi feita para que futuras implementações de funcionalidades no código dificilmente sejam restringidas por falta de RAM e pelo fato de que a diferença de preço entre o microcontrolador escolhido e sua versão com 2048 *bytes* de RAM é de apenas 14 centavos de dólar americano (cotado sob as condições

já descritas).

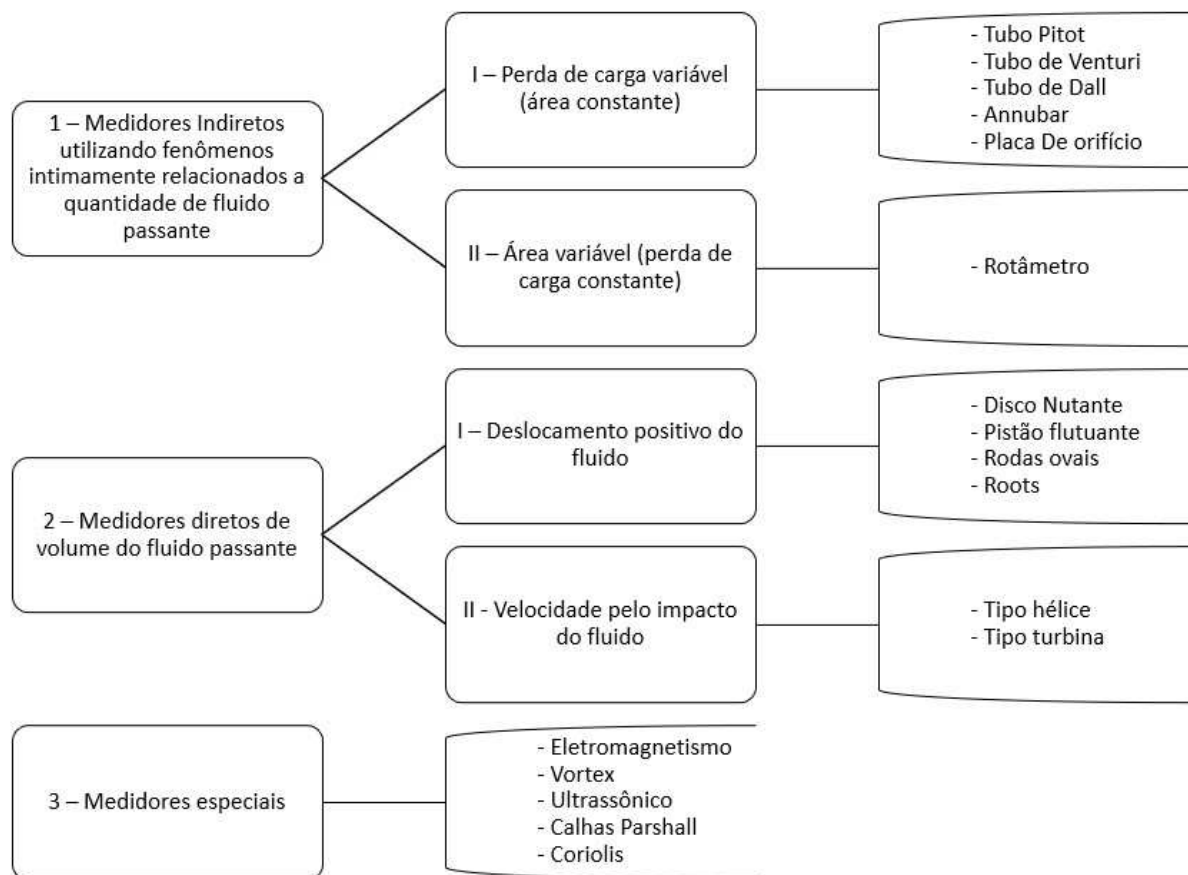
## 2.2 MAPA (MÓDULO DE AQUISIÇÃO, PROCESSAMENTO E ATUAÇÃO)

### 2.2.1 SENSOR DE FLUXO

Para monitorar as condições do reservatório, e conseqüentemente realizar um gerenciamento eficaz do fornecimento de água, o sistema deverá ser capaz de identificar a quantidade de água disponível. Portanto, serão utilizados sensores capazes de medir a taxa de fluxo de um líquido que passa através de uma tubulação.

Existem diversos tipos de sensores de fluxo (Figura 10) sendo utilizados nos mais variados setores da indústria, como o de petróleo, alimentos, produtos químicos e tratamento de resíduos, bem como na medicina. Para a escolha do sensor apropriado devem ser considerados fatores que incluem: o tipo de fluido a ser medido, a faixa de velocidade e a precisão do sensor, além do seu custo, facilidade de manutenção e instalação.

**Figura 10: Tipos de sensores de fluxo.**



Fonte: Adaptado (CASSIOLATO, s.d.)



Os sensores de fluxos, portanto, podem ser desde aqueles no qual o fluxo é determinado indiretamente com base na medição da pressão diferencial, como tubo de Venturi, assim como aqueles que medem diretamente o fluxo passante, como a engrenagem oval. Existem também os medidores com algumas características especiais, como os sensores ultrassônicos, que enviam um sinal a jusante e outro a montante e em seguida compara o tempo de viagem dos dois sinais para encontrar a velocidade do fluxo, e conseqüentemente permite calcular a vazão volumétrica.

Logo, devido à variedade de opções, foi estabelecido que o sensor ideal deveria atender a três requisitos fundamentais:

- Deveria ser adaptável aos diferentes formatos de reservatórios;
- Sua operação não dependeria do material no qual os reservatórios foram fabricados;
- E que fosse um produto comercial e de fácil substituição.

Portanto, foi escolhido o sensor de fluxo de água abaixo, modelo POW110D3B, da fabricante Seeed, apresentado na Figura 11.

**Figura 11: Sensor de fluxo escolhido.**



**Fonte: (POW110D3B, s.d.)**

Este sensor é um dos sensores de fluxo mais comuns, baratos, confiáveis e precisos, utilizados em muitas aplicações para medição de vazão de líquido. Seu funcionamento se baseia na medição da velocidade pelo impacto do fluido e consiste em um corpo de válvula de plástico, um rotor do tipo hélice e um sensor de efeito Hall.

Quando a água flui através do rotor, o rotor gira. Sua velocidade muda com diferentes taxas de fluxo. Um sensor de Efeito Hall integrado detecta o movimento do rotor e gera um sinal de frequência que é análogo à taxa de fluxo.

Tais sensores geralmente devem ser adaptados ao diâmetro do tubo através do qual o fluido é transportado, e devem estar localizados linearmente ao sistema, a fim de permitir uma maior rapidez e precisão na medição. No projeto serão utilizados dois sensores, no qual serão medidos tanto fluxo de saída, bem como o fluxo de entrada de água.

### 2.2.2 ELETROVÁLVULA

Uma válvula é um componente em um sistema de tubulação, normalmente utilizado para controlar o fluxo ou a direção de um líquido. Elas podem ser classificadas de diferentes maneiras, mas uma forma comumente utilizada é pela forma como elas são acionadas que pode ser por: acionamento hidráulico, manual, pneumático, motorizado ou elétrico.

Devido a sua simplicidade, além de eficiência e alta durabilidade, para o projeto será utilizado uma eletroválvula (ou válvula solenoide), que é uma válvula eletricamente ativada, ou seja, sem interferência manual. Ela vem sendo utilizada nos mais variados setores e equipamentos. Alguns exemplos do uso incluem sistemas de aquecimento, automação industrial, piscinas, máquinas de lavar, equipamentos odontológicos e sistemas de irrigação.

Existem um número diversificado de eletroválvulas, que podem ser normalmente fechadas e normalmente abertas, com ação direta, ou pilotadas internamente, ou até mesmo externamente, além de variar no número de portas, bem como no número de caminhos de fluxo.

Esta válvula opera utilizando uma bobina elétrica com um núcleo ferromagnético móvel no centro. Esse núcleo é chamado de êmbolo. É usada para alterar o estado de uma válvula de aberta para fechada ou vice-versa, ou seja, se a válvula solenoide estiver "normalmente fechada", quando a bobina for energizada, a válvula será aberta pela força eletromagnética produzida pela bobina, que devido a criação de um campo magnético, puxa o embolo, permitindo o fluxo de água. Quando desenergizado, o êmbolo retorna à sua posição original.

Portanto, foi escolhida para o controle da quantidade de água no reservatório, a válvula solenoide da Figura 12, que é normalmente fechada, com 12 V DC de alimentação.

### 2.2.3 SENSOR DE NÍVEL

O sensor de nível é um dispositivo utilizado para detectar níveis de líquido ou interfaces entre líquidos em sistemas abertos ou fechados.

Uma ampla variedade desses sensores é disponibilizada no mercado para atender as

**Figura 12: Válvula solenoide.**



**Fonte: (CANALAGRICOLA, s.d.)**

mais diversas aplicações na indústria, como na indústria petroquímica ou de tratamento de resíduos, bem como no setor de bens de consumo, em aplicações domésticas, estando presentes em máquinas de lavar roupa, espremedores de suco e ferros a vapor, por exemplo.

A Tabela 1 contém os principais sensores considerados para o projeto, apresentando suas vantagens e desvantagens. Para escolha do sensor apropriado foram utilizados os mesmos critérios considerados para o sensor de fluxo, tendo como pontos fundamentais a versatilidade, o baixo custo, a durabilidade e a confiabilidade. Portanto, foi escolhido utilizar um sensor tipo boia (Figura 13), feito de material plástico (Polipropileno), composto de uma haste na qual desliza um cilindro feito de material flutuante.

**Figura 13: Sensor de nível de líquidos.**



**Fonte: (9SS44, s.d.)**

Este tipo de sensor é um interruptor *ON/OFF* que abre ou fecha um circuito elétrico com base no movimento da boia. Ele é capaz de detectar o nível de líquidos na altura em que são instalados. Nesse projeto serão utilizados dois sensores para monitorar o reservatório e sinalizar se o líquido está abaixo do recomendável ou em condição de transbordamento.

**Tabela 1: Tipos de sensores.**

Sensor	Vantagens	Desvantagens
Fibra óptica	<ul style="list-style-type: none"> <li>Baixa interferência eletromagnética.</li> <li>Robustez em relação a ambientes hostis.</li> </ul>	<ul style="list-style-type: none"> <li>Não é capaz de medir líquidos turvos ou líquidos espessos.</li> <li>Alto custo</li> </ul>
Ultrassônico	<ul style="list-style-type: none"> <li>Estrutura simples</li> <li>Fácil instalação e manutenção</li> </ul>	<ul style="list-style-type: none"> <li>Susceptível a interferências</li> <li>Difícil de alcançar performance de segurança</li> <li>Perde precisão com a presença de sujeiras, espumas.</li> </ul>
Sensor de boia	<ul style="list-style-type: none"> <li>Estrutura simples</li> <li>Baixo custo</li> <li>Adaptável</li> </ul>	<ul style="list-style-type: none"> <li>Vulnerabilidade a danos mecânicos</li> </ul>
Capacitivos	<ul style="list-style-type: none"> <li>Baixo custo</li> <li>Alta linearidade</li> <li>Baixa dissipação de energia</li> </ul>	<ul style="list-style-type: none"> <li>Estrutura complexa</li> <li>Constante dielétrica variável</li> <li>Interferência em relação a camadas condutivas</li> </ul>
Pressão diferencial	<ul style="list-style-type: none"> <li>Performance estável</li> <li>Baixo custo</li> </ul>	<ul style="list-style-type: none"> <li>Fácil de ser obstruído</li> </ul>

Fonte: (JIN, 2015)

Além disso, estes sensores serão essenciais para a constante atualização do valor real de líquido no reservatório, uma vez que os sensores de fluxo não possuem alta classe de exatidão.

#### 2.2.4 LEDS, MICROCONTROLADOR E ALIMENTAÇÃO

Para o MAPA, as considerações acerca dos LEDs são as mesmas do MIU. Por conveniência e pela diferença pequena de preços para um modelo mais simples, utilizaremos o mesmo microcontrolador, facilitando assim a escrita do código e depuração posterior.

Sobre a alimentação, consideramos as mesmas ponderações, com a exceção de que este módulo deverá possuir, também, uma linha de 12 Volts para alimentar a eletroválvula.

## 2.3 COMUNICAÇÃO

A elaboração desse projeto determina a utilização de dois módulos, cada um com sua unidade de processamento particular. Também é previsto que eles estejam a uma distância de, pelo menos, 3 metros um do outro; visto que o MAPA será instalado mais próximo do reservatório e o MIU estará localizado em um local de fácil acesso para o usuário. Como em qualquer outro sistema de automação, também serão usados sensores e atuadores. E, para que haja a troca de informações dentro de todo esse processo, é necessário definir como esses dois módulos irão trocar informações.

### 2.3.1 MEIO DE TRANSMISSÃO

A cada dia que passa, as soluções de redes sem fio se tornam mais comuns nas mais diversas aplicações, inclusive na automação. Tem se destacado por sua facilidade de instalação, lançando mão da necessidade de estender cabos e providenciar infraestrutura adequada, principalmente em edificações que não são novas.

Porém, para que a comunicação através destas soluções seja eficaz, vários aspectos devem ser considerados, como cita Branquinho (2019). Segundo eles, embora as redes sem fio apresentem uma maior flexibilidade para o usuário, as redes cabeadas garantem maior confiabilidade. Portanto, a tecnologia sem fio deve ser aplicada, preferencialmente, nos casos em que não pode ser empregada uma rede cabeada, como aplicações em que o dispositivo conectado necessite de mobilidade.

A confiabilidade é essencial para a aplicação adequada desse projeto, entretanto Shinder (2014) exemplifica que, assim como podemos bloquear sinais de transmissões de rádio intencionalmente, esse bloqueio também pode acontecer de maneira não intencional, por conta dos materiais usados na construção de uma edificação. Ou talvez o sinal de transmissão não seja forte o suficiente para transpor as barreiras e alcançar toda a área da edificação.

Assim, para prevenir falhas de comunicação causadas por interferências ou baixa potência de sinal recebido, opta-se por adotar uma rede formada por condutores fixos para interligar os dispositivos. Também caracterizados por serem um meio de transmissão guiado, onde a informação trafega contida por limites físicos, o que evita a perda de dados.

Os cabos de par trançado, são os mais comuns e de mais fácil acesso no mercado, se comparados ao cabo coaxial e a fibra óptica. Ele utiliza um dos fios para transmitir sinais elétricos para o receptor e o outro apenas como uma terra de referência. O receptor utiliza a

diferença de potencial entre os dois fios para determinar a amplitude do sinal elétrico.

O principal cabo utilizado para redes de dados e voz é o cabo de quatro pares trançado não blindado, também conhecido como cabo UTP (*Unshielded Twisted Pair*– par trançado não blindado). Devido à sua constituição (trançamento dos pares) e à forma de transmissão utilizada (transmissão balanceada), este cabo fornece um bom grau de imunidade a interferências eletromagnéticas, principalmente considerando a aplicação em um ambiente residencial (MURATORI, 2011).

### 2.3.2 PROTOCOLO DE COMUNICAÇÃO

Geralmente sensores e atuadores utilizam uma topologia em barramento, por sua economia em uso de cabos comparada a outras topologias, boa confiabilidade e de fácil manutenção, pois os dispositivos podem ser removidos e adicionados a rede sem a necessidade de descontinuí-la. Sendo assim, o protocolo que mais se destaca, principalmente pela sua velocidade de transferência, fácil integração e por ser um protocolo aberto (que qualquer usuário pode implementar sem pagar taxas aos desenvolvedores) é o MODBUS.

Os meios de comunicação entre os equipamentos na rede industrial, são geralmente os meios seriais, ou seja, os padrões de RS (*Recommended Standard*).

O padrão serial RS atua sobre dois modelos *Single-Ended Data Transmission* e *Differential Data Transmission*. Esse segundo oferece uma alta taxa de transmissão e longas distâncias, chegando até 1200 metros. É usada linha diferenciada, o que significa dizer que o dado é representado através da corrente e não pela tensão como no modelo *Single-Ended*. A estrutura do *Differential* permite uma conexão multiponto, onde é possível a existência de um mestre e diversos escravos compartilhando mesmo meio. Essas implementações podem ser encontradas nos padrões RS422 e RS485 (CUNHA, 2000).

O RS485 é um dos padrões de comunicação serial e bidirecional utilizados em aplicações industriais, como: automação, equipamentos médicos, embarcações, laboratórios, robótica, dentre outras (FREITAS, 2017).

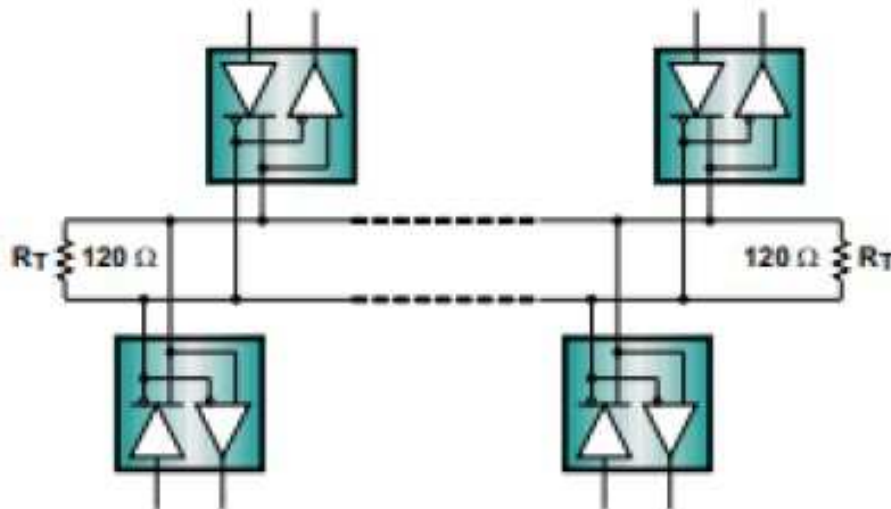
A RS485 diferencia-se dos demais modelos, por utilizar um ou dois pares de fios para transmissão de dados. O padrão pode atuar em modo *half-duplex* usando um par de cabos padrão ou em *full-duplex* usando dois pares de cabos, tal que em *half-duplex* somente é possível utilizar uma operação por vez no barramento, ou seja, as operações de emissão ou recebimento de informações não podem acontecer simultaneamente. No modo *full-duplex*, é possível enviar e receber dados simultaneamente, entretanto a utilização do modo *half-duplex*

predomina na maioria das aplicações industriais por conta do baixo custo de implantação e a excelente velocidade que o padrão já acomoda.

Tem também isolamento óptico, e pode atingir a taxas de transmissão de 10 Mbps e aproximadamente 1200 metros de distância, sem amplificação do sinal. Admite a conexão multiponto, suportando até 32 equipamentos na rede, ou seja, um mestre e 31 escravos. Utilizando o modelo *Differential* para transmissão.

De acordo com o padrão RS-485 o início e o fim do barramento devem ser ligados com terminadores de rede, que são resistores de 120 ohms ligados entre as duas linhas do barramento (Figura 14). Estes terminadores geralmente vêm montados dentro dos dispositivos, sendo acionados por uma pequena *dip-switch*. Apenas os dois dispositivos das extremidades da rede devem possuir o terminador, os demais não precisam do terminador. Isso acontece pois o sinal pode acabar sendo refletido pelas extremidades da rede, atrapalhando o sinal original de comunicação.

**Figura 14: Terminação apropriada do RS-485.**

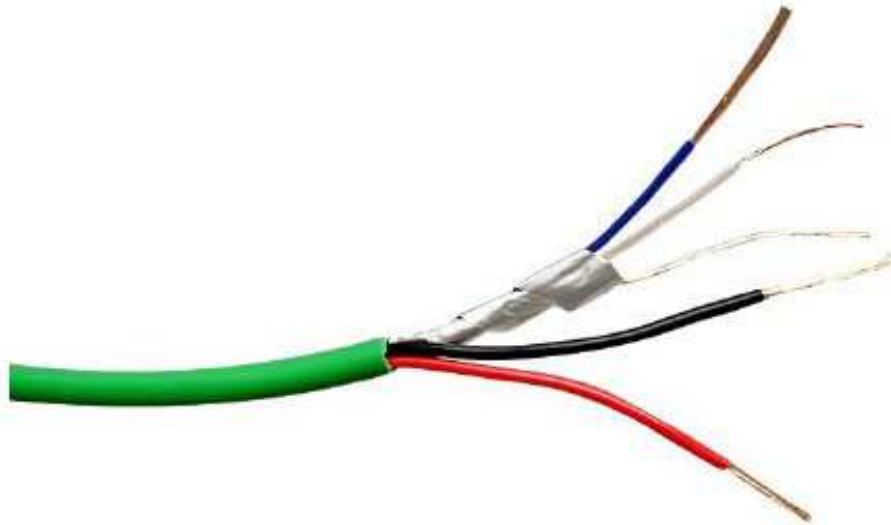


**Fonte: (TEXAS, 2016)**

O guia da Texas (2016) dá como válido para aplicações de baixo custo, cabos de pares trançados com bitola mínima de  $0,2 \text{ mm}^2$ , preferencialmente blindado, como o apresentado na Figura 15. O cabo também deve possuir uma capacitância de no máximo  $55 \text{ pF/m}$  e uma impedância maior que 100 ohms. Caso o condutor de terra não seja utilizado em comum com os dispositivos da rede a blindagem do cabo deve ser aterrada em apenas uma de suas extremidades.

Este protocolo foi criado baseado na arquitetura mestre/escravo oferecendo serviços de acordo com códigos de função. Um fator de grande importância para o contínuo uso do MODBUS se encontra no fato dele estar implementado na camada de aplicação do modelo OSI,

**Figura 15: Cabo Automação padrão RS485 com blindagem.**



**Fonte: (0825VD, 2020)**

o que torna possível sua integração com outros ambientes de rede. Para que haja integração entre tais tipos de rede, são necessários conversores de protocolo, também chamados de *gateways*.



### 3 PROTÓTIPO

#### 3.1 PROTOTIPAGEM DOS MÓDULOS MIU E MAPA

Todo o projeto eletrônico e da placa de circuito impresso (PCI) foi feito pela equipe, no software EAGLE, da empresa Autodesk. Os componentes foram adquiridos dos EUA, com a empresa Mouser Electronics, e montados nas PCIs pela equipe. Para a fabricação destas, foi escolhida a empresa JLC, que opera na China com preços bastante competitivos, principalmente para prototipagem.

Os esquemáticos dos dois protótipos, MIU e MAPA, são apresentados nas Figuras 16 e 17, respectivamente.

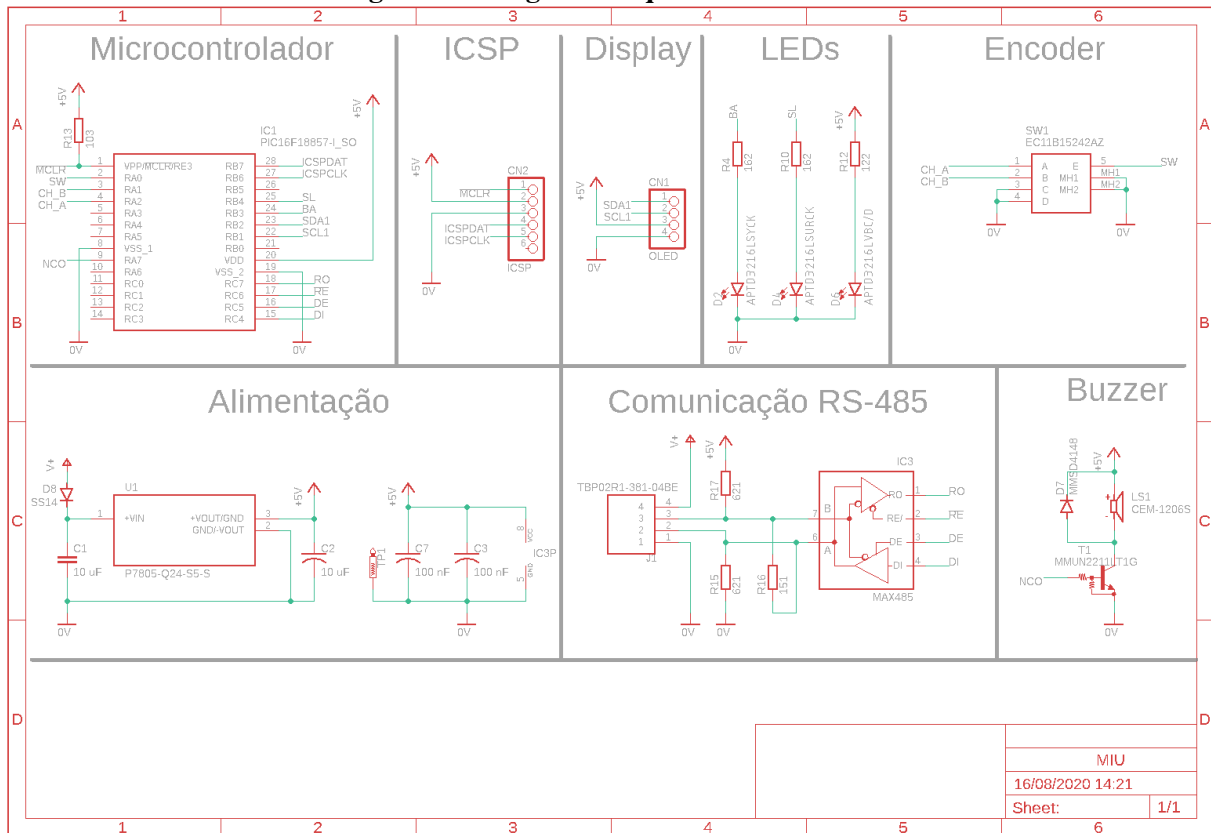
Os componentes no interior do retângulo de linha tracejada no esquemático do MAPA não foram instalados na versão final do protótipo. Como não eram componentes vitais ao projeto, sendo a função deles apenas medir a corrente elétrica na eletroválvula para fins de autodiagnóstico, estes foram omitidos inclusive de compra, uma vez que esta se deu no cenário de alta do dólar, o que já impactava no orçamento do projeto como um todo. A decisão da equipe foi, portanto, de não adquiri-los, sem nenhum prejuízo ao objetivo do projeto.

As placas de circuito impresso, dos dois protótipos, MIU e MAPA, são apresentados na Figura 18.

As trilhas em azul representam o lado inferior da placa, ao passo que os vermelhos, o lado superior. Vale destacar que, embora ambos os protótipos tenham sido projetados empregando o máximo de componentes de montagem superficial (SMD), a soldagem destes foi feita manualmente de forma convencional, sem empregar máscara de solda ou outros métodos, uma vez que isto elevaria ainda mais o custo de montagem. A utilização de componentes SMD, além de focar na redução do tamanho dos protótipos, buscava a abordagem mais profissional possível, e o método de soldagem acabou por pouco impactar este objetivo. O resultado final é apresentado nas fotos da Figura 19, para o MIU e na Figura 20, para o MAPA.

Pode-se observar em todas as imagens o cabo verde que interconecta as placas. Este

**Figura 16: Diagrama esquemático do MIU.**



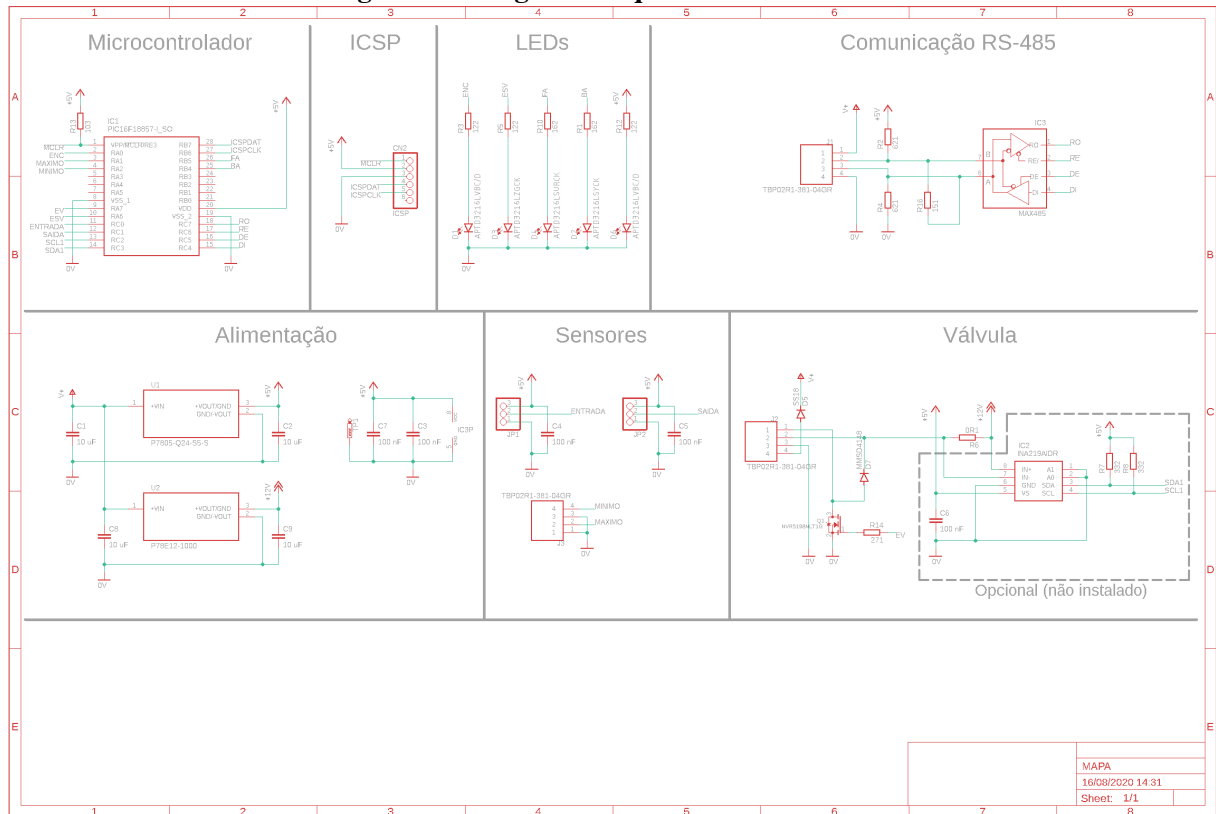
**Fonte: Autores**

cabo é próprio para a comunicação RS-485, possuindo blindagem e condutores trançados, necessários para preservar a integridade dos sinais.

Nota-se ainda, no MAPA, além dos componentes intencionalmente não instalados, que no lugar onde seria ligada a eletroválvula, um LED de 5 mm foi colocado no lugar. Isto foi feito pois, mesmo tendo o grupo adquirido e utilizado a eletroválvula, verificar o funcionamento do módulo ficaria mais fácil com uma indicação luminosa.

Outro motivo que levou à troca foi o fato de que, durante os ciclos de programação, por vezes o pino do microcontrolador que aciona o MOSFET, responsável por energizar a eletroválvula, acabava em estado flutuante, o que deixava o componente operando em sua região linear, o que elevava a resistência dreno-fonte (RDS) e, devido à corrente elétrica que circula pela eletroválvula, produzia muito calor no componente. Com o LED, esta dissipação se tornou insignificante, reduzindo o risco de danos ao componente. Todavia, durante a operação normal do módulo, mediu-se a temperatura no componente utilizando um termômetro infravermelho da FLUKE, modelo 62 MAX, e o mesmo nunca registrou mais do que 50 °C, estando dentro dos limites do componente e do esperado em projeto, levando em consideração a potência térmica

**Figura 17: Diagrama esquemático do MAPA.**



Fonte: Autores

dissipada em função da corrente de dreno  $I_D$  esperada, da resistência  $R_{DS(ON)}$  e dados de impedância térmica, fornecidos no catálogo do fabricante (NVR5198NL, 2019).

A programação de ambos os módulos se deu na linguagem C, por meio do compilador XC8 no ambiente integrado de desenvolvimento MPLABX, ambos da própria Microchip, fabricante do microcontrolador escolhido. Os anexos abordarão os aspectos de *firmware*.

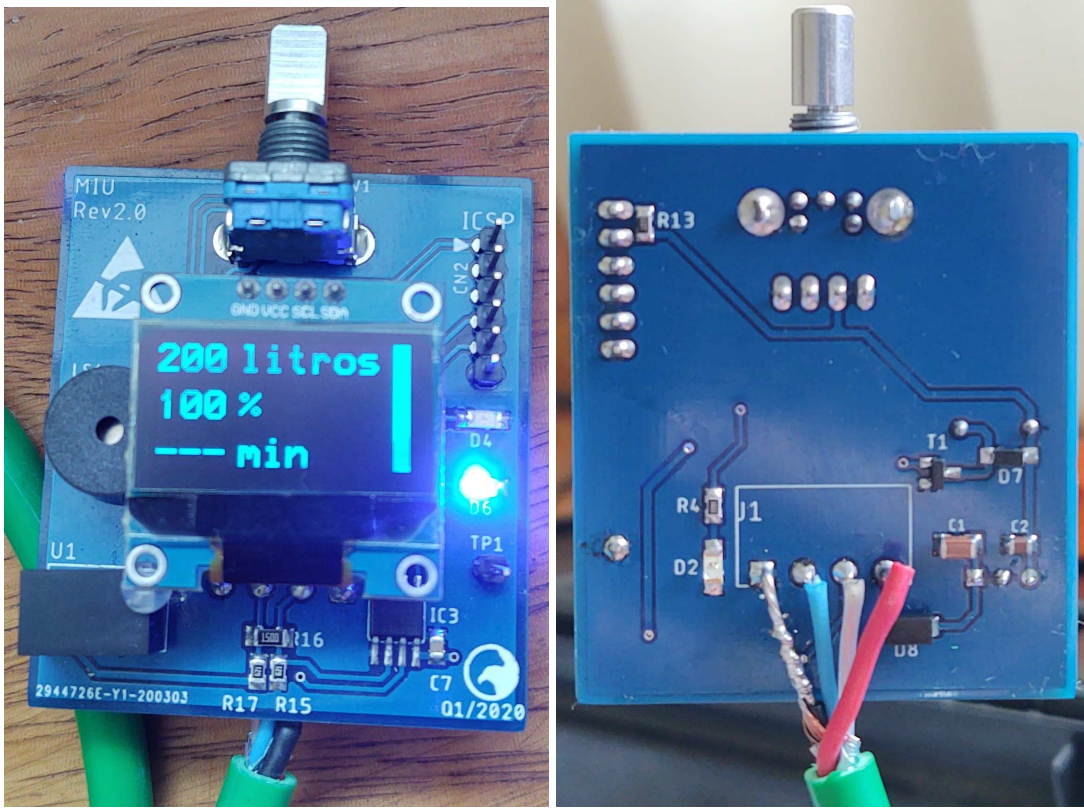
### 3.2 DESAFIOS ENCONTRADOS DURANTE O DESENVOLVIMENTO

Com a eclosão da pandemia do SARS-CoV-2, a equipe encontrou alguns obstáculos. O desenvolvimento e montagem dos protótipos se deu em meados de março, logo na época em que esta enfermidade tornou-se mundialmente conhecida. Com isto, houveram impactos econômicos, que acarretaram na alta do dólar, o que elevou os custos do projeto, em relação ao orçamento que possuíamos. Logo, alguns itens foram deixados de lado, como por exemplo, a medição da corrente na eletroválvula para fins de diagnóstico.

Outro impacto foi que a equipe se viu impedida de desenvolver o protótipo do reservatório, representado na Figura 21. O grupo concordou em não realizar reuniões

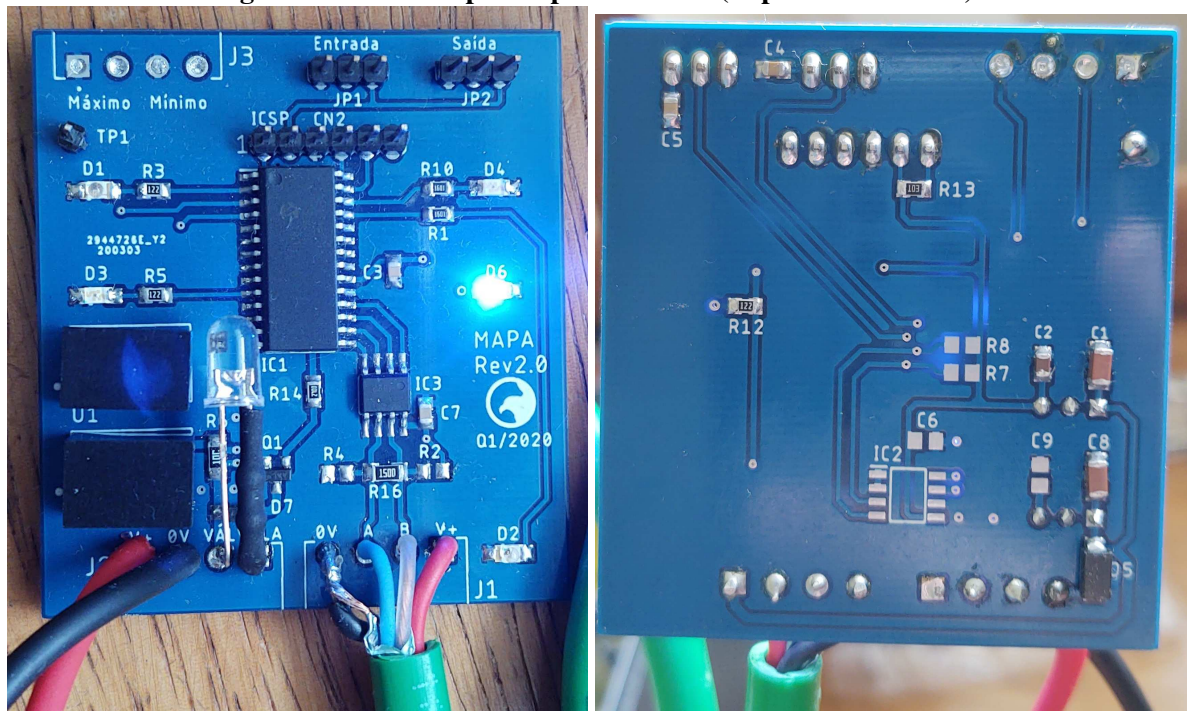


**Figura 19: Fotos do protótipo do MIU (Superior e Inferior).**



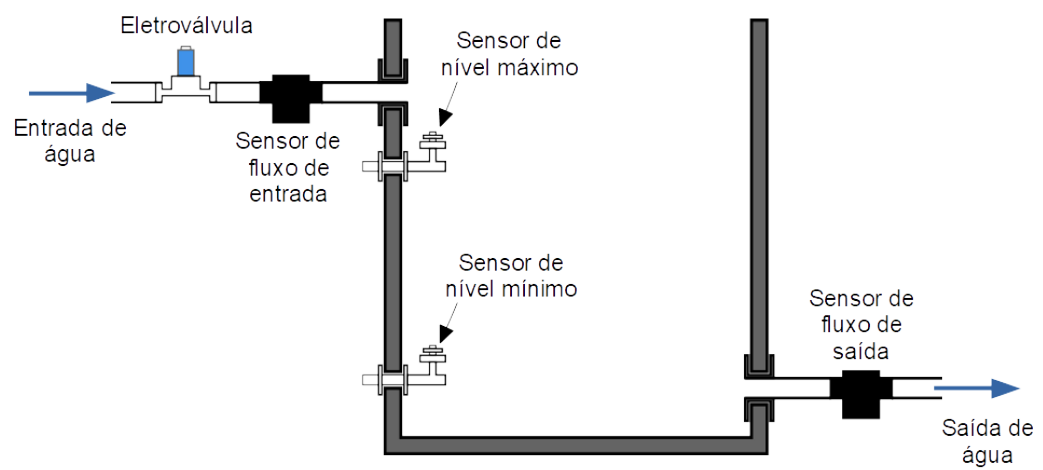
Fonte: Autores

**Figura 20: Fotos do protótipo do MAPA (Superior e Inferior).**



Fonte: Autores

**Figura 21: Elementos do protótipo.**



**Fonte: Autores**

## 4 EMULAÇÃO E RESULTADOS

De posse dos dados do *hardware* escolhido, podemos então começar o projeto do *firmware*. Este processo iniciará com a descrição de todas as funcionalidades de cada módulo, já estruturada nos moldes do fluxo de programa. Desta forma, será possível ter uma noção geral da estrutura, funções e comportamento do programa, facilitando e servindo como recurso auxiliar à escrita do *firmware* na linguagem de programação.

### 4.1 MIU

O módulo de interface com o usuário terá tarefas específicas no seu programa, sendo elas de forma geral:

- Inicializar o módulo como um todo;
- Manter a própria contagem do tempo para coordenar as operações do sistema;
- Controlar quando e o que será exibido no *display*;
- Acionar o LED e o *buzzer* conforme o estado do sistema;
- Receber os comandos do *encoder*;
- Receber e enviar dados para o MAPA.

Ao ser ligado, o programa executará três etapas principais: inicializar o sistema como um todo, apresentará a saudação (exibe mensagem no *display* e pulsa o LED de alerta), buscará configurações salvas e seguirá para o laço principal (Apêndice A.1).

Após a saudação, o programa verificará se existe uma configuração prévia do sistema gravada na memória EEPROM do microcontrolador. Esta configuração contém os dados do tempo de tela ligada em segundos. Caso os valores gravados não estejam coerentes como os valores pré-determinados, o programa salvará os valores pré-determinados no lugar.

Com isto, o sistema estará pronto para operar, entrando em um laço do qual ele não deverá sair (exceto nas interrupções ou no caso de falha na comunicação com o MAPA), cumprindo dentro deste as atividades responsáveis pelas funções do equipamento.

O primeiro passo é verificar se houve um incremento de um segundo na contagem de tempo. Caso afirmativo, o MIU solicita os valores do MAPA e exibe a mensagem vigente, se houver. Mensagem é o nome dado à informação do estado do reservatório interpretada pelo MAPA, podendo ser:

- Interrupção no fornecimento de água detectada (valor 1);
- Nível baixo no reservatório (valor 2);
- Nível crítico no reservatório (valor 4);
- Nenhuma mensagem (valor 0).

A mensagem com valor 0 tem o papel de desligar o LED de alerta do MIU e zerar as variáveis de controle do alerta sonoro, como mostrado no trecho do código a seguir:

```

229 //===== Mensagens geradas pelo MAPA =====//
230 void Exibe_Mensagem(void)
231 {
232     uint8_t n = 0;
233
234     if(Mensagem == 1 && Mensagem_Antiga != Mensagem)
235     {
236         Timeout_Tela = 60000;
237         SSD1306_Command(SSD1306_DISPLAYON);
238         Estado_LED = LED_Pulsando;
239         Clear_Buffer();
240         GD_Write_String("FALTA", 33, 14, _10x16);
241         GD_Write_String("AGUA", 39, 32, _10x16);
242         GD_Screen_Update();
243         Timeout_Alarme = 60000;
244         Silencia_Alarme = 0;
245     }
246     else if(Mensagem == 2 && Mensagem_Antiga != Mensagem)
247     {
248         Timeout_Tela = 60000;
249         SSD1306_Command(SSD1306_DISPLAYON);
250         Estado_LED = LED_Piscando;
251         Clear_Buffer();
252         GD_Write_String("NIVEL", 33, 14, _10x16);
253         GD_Write_String("BAIXO", 33, 32, _10x16);
254         GD_Screen_Update();
255         Timeout_Alarme = 60000;
256         Silencia_Alarme = 0;

```



```

257 }
258 else if(Mensagem == 4 && Mensagem_Antiga != Mensagem)
259 {
260 Timeout_Tela = 60000;
261 SSD1306_Command(SSD1306_DISPLAYON);
262 Estado_LED = LED_Piscando_Rapido;
263 Clear_Buffer();
264 GD_Write_String("NIVEL", 33, 14, _10x16);
265 GD_Write_String("CRITICO", 21, 32, _10x16);
266 GD_Screen_Update();
267 Timeout_Alarme = 60000;
268 Silencia_Alarme = 0;
269 }
270 else if(Mensagem == 0 && Mensagem_Antiga != Mensagem)
271 {
272 Estado_LED = LED_Apagado;
273 Timeout_Alarme = 0;
274 Silencia_Alarme = 0;
275 }
276
277 if(Silencia_Alarme == 0 && Mensagem != 0 && Timeout_Alarme != 0)
278 Reproduz_Som(2400, 500);
279
280 Mensagem_Antiga = Mensagem;
281 }

```

Os valores atribuídos aos estados do LED de alerta são definidos no arquivo *shared.c*, que possui as variáveis e definições compartilhadas entre os vários códigos. Os valores são:

```

14 #define LED_Apagado 0
15 #define LED_Aceso 1
16 #define LED_Piscando 2
17 #define LED_Pulsando 4
18 #define LED_Piscando_Rapido 8
19 #define LED_Manual 16

```

No *display*, as mensagens aparecem da seguinte forma, mudando apenas o texto apresentada na Figura 22.

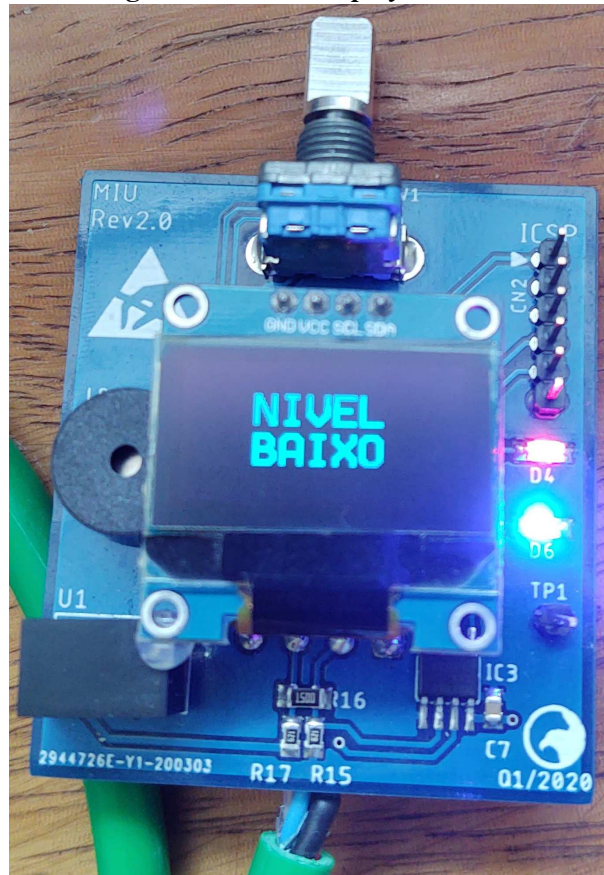
Em seguida, se a variável que controla o tempo que o *display* passa ligado for diferente de zero, a variável que comando a atualização do *display* é habilitada. Esta informação é utilizada já nas linhas seguintes, responsáveis por desenhar na tela aquilo que for competente à tela selecionada pelo usuário.

```

417 // Atualiza o display conforme necessário. Isto poupa energia e
418 // processamento, reduzindo a latência da parte gráfica do sistema
419 if(Atualiza_Display && Timeout_Alarme == 0)
420 {
421 // Troca de telas

```

**Figura 22: Exemplo de mensagem exibida no *display*. Nota-se o LED de Alerta (D4) aceso.**



**Fonte: Autores**

```

422 switch (Tela)
423 {
424 case Tela_Principal:
425 Tela_principal();
426 break;
427
428 case Tela_Ajuste_Volume:
429 Ajuste_Volume();
430 break;
431
432 case Tela_Ajuste_Tempo:
433 Ajuste_Tempo();
434 break;
435 }
436
437 Atualiza_Display = 0;
438 }

```

As informações são escritas no *display* de forma simples por meio da biblioteca gráfica desenvolvida por um dos integrantes da equipe. Por esta não ser o foco deste projeto e as sub rotinas responsáveis por desenhar cada uma destas telas ser meramente o emprego desta

biblioteca, este trecho do código será omitido neste documento. As telas existentes são:

- Tela\_principal: exibe os valores restantes estimados de volume, porcentagem e duração. Contém, ainda, uma barra gráfica para fornecer uma indicação visual do restante de água no reservatório (Figura 23);
- Ajuste\_Volume: utilizada para definir o volume máximo comportado pelo reservatório (Figura 25 - Esquerda);
- Ajuste\_Tempo: utilizada para definir por quanto tempo a tela permanece ligada (Figura 25 - Direita).

**Figura 23: Tela Principal.**



**Fonte: Autores**

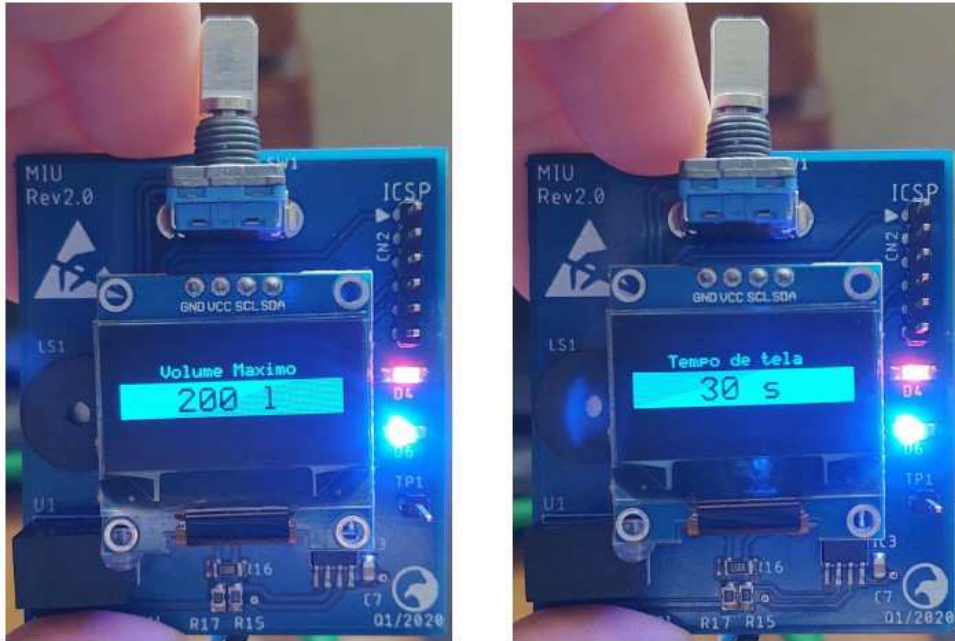
**Figura 24: Parâmetros indicados na tela. Todos os valores são os restantes estimados.**

Volume	<b>200 litros</b>	Indicação gráfica do volume restante
Porcentagem	<b>100 %</b>	
Duração	<b>--- min</b>	

**Fonte: Autores**

O próximo passo é analisar o estado do *encoder* rotativo e entender se ele foi girado no sentido horário ou anti-horário e ajustar o parâmetro que estiver na tela. O código gera diferentes tons para incremento ou decremento do valor, e um tom mais agudo ao atingir os limites dos parâmetros, limites estes impostos no código.

**Figura 25: Telas Ajuste\_Volume e Ajuste\_Tempo, respectivamente.**



**Fonte: Autores**

As variáveis utilizadas neste processo são:

- Estado\_ENC\_CH\_A e Estado\_ENC\_CH\_B: variáveis que armazenam o resultado do *debounce* via *software*, realizado na rotina de interrupção do TMR4;
- ChA\_Old e ChB\_Old: variáveis que armazenam o **último** resultado do *debounce* via *software*, servindo de base para determinar quando houve uma mudança no estado deste.

O *debounce* é feito deslocando todos os bits uma posição para a esquerda e realizando um OU lógico deste resultado com o valor atual do pino. Desta forma, temos as possíveis condições:

- Estado\_ENC\_CH\_x = 0xFF: significa que há uma leitura estável, onde o nível lógico da entrada permaneceu em 1;
- Estado\_ENC\_CH\_x = 0: significa que há uma leitura estável, onde o nível lógico da entrada permaneceu em 0;
- Estado\_ENC\_CH\_x diferente de 0 e 0xFF: significa que há uma variação na leitura (*bouncing*).

Vale lembrar que utilizamos a lógica invertida, ou seja, quando os contatos do *encoder* fecham, temos nível lógico zero (Apêndice A.2). Desta forma, além de utilizarmos

o recomendado pelo fabricante do *encoder*, utilizamos os resistores de *pull-up* internos do microcontrolador, poupando espaço na placa e custo total.

Por fim, será analisado o estado da chave do *encoder* (lembrando que este pode ser girado e pressionado). Esta chave é utilizada para confirmar alterações nos valores (pressionar rápido), cancelar as alterações (pressionar longo), ligar a tela e parar os alertas sonoros e na tela (a indicação por meio do LED de falha não pode ser suprimida). O funcionamento da leitura desta chave se dá de forma idêntica ao dos contatos do giro do *encoder* (Apêndice A.3).

**Figura 26: Telas de saída das telas de ajuste.**



**Fonte: Autores**

Além do código principal, outro arquivo-fonte que deve ser abordado aqui é o que trata do *Timer4* (doravante referido por TMR4). O código executado dentro da função de interrupção deste temporizador realiza diversas funções do módulo.

Logo no início da interrupção temos:

- A variável PWM1LDV recebe os valores dos registradores responsáveis pelo ciclo ativo do PWM que controla o LED de alerta;
- Controle da variável que armazena a contagem de interrupções (*Ticks*, que serve como base de tempo para o sistema. Cada interrupção ocorre a cada 1 ms;
- Controle da variável que armazena por quanto tempo a chave do *encoder* ficou pressionada. Com isto diferenciaremos um pressionar rápido de um longo.

```

191 void TMR4_DefaultInterruptHandler(void)
192 {
193 uint16_t PWM1LDV = (CCPR1H << 8) + CCPR1L;
194
195 WWDI_TimerClear();
196
197 // Contagem de Ticks do sistema
198 if(SysTicks != 10000)
199 SysTicks++;
200 else
201 SysTicks = 0;
202
203 // Contagem do tempo do botão pressionado
204 if(Tempo_Botao < 5000)
205 Tempo_Botao++;

```

O *debounce* dos contatos do *encoder* vem logo em seguida:

```

207 // Debounce via Software do Encoder
208 Estado_ENC_SW = (Estado_ENC_SW << 1) | ENC_SW;
209 Estado_ENC_CH_A = (Estado_ENC_CH_A << 1) | ENC_CH_A;
210 Estado_ENC_CH_B = (Estado_ENC_CH_B << 1) | ENC_CH_B;

```

Como utilizamos um *display* OLED, é interessante que este fique ligado apenas quando necessário, para prolongar sua vida útil. O controle de quando este deve ser ligado ou não é feito também dentro da interrupção:

```

212 // Wake-up e timeout da tela
213 if((Estado_ENC_SW == 0 || Estado_ENC_CH_A == 0) && Timeout_Tela == 0)
214 {
215 SSD1306_Command(SSD1306_DISPLAYON);
216 Timeout_Tela = Tempo_Tela;
217 Wake_Up = 1;
218 }
219 if((Estado_ENC_SW == 0 || Estado_ENC_CH_A == 0) && Timeout_Tela != 0)
220 Timeout_Tela = Tempo_Tela;
221 if(Estado_ENC_SW == 0xFF && Estado_ENC_CH_A == 0xFF && Timeout_Tela == 0)
222 SSD1306_Command(SSD1306_DISPLAYOFF);
223 if((Estado_ENC_SW & Estado_ENC_CH_A) == 0xFF && Timeout_Tela != 0)
224 Timeout_Tela--;

```

O controle do ciclo ativo do PWM que aciona o LED de alerta é feito no trecho seguinte da interrupção, garantindo a execução suave de todos os efeitos deste indicador, apresentado no Apêndice A.4.

Por fim, o código desabilita o módulo NCO ao fim do tempo determinado para o tom ser emitido, e força o pino que aciona o *buzzer* ao nível lógico 0. Isto é necessário pois o

*buzzer* não é fabricado para ser alimentado constantemente por uma tensão elétrica fixa, mas sim receber algum tipo de sinal, seja ele digital, como o PWM, ou analógico, que poderia ser gerado com um DAC (Conversor digital-analógico). Se uma tensão constante for deixada em sua bobina, esta aquecerá e poderá danificar o componente.

```

276 // Desabilita o NCO quando acabar o tempo de reprodução da nota
277 if(NCO1CONbits.N1EN == 1 && Tempo_Som != 0)
278 {
279     Tempo_Som--;
280
281     if(Tempo_Som == 0)
282     {
283         NCO1CONbits.N1EN = 0;
284         LATAbits.LATA7 = 0;
285     }
286 }
287 }

```

## 4.2 MAPA

O Módulo de Aquisição, Processamento e Atuação terá tarefas específicas no seu programa, sendo elas de forma geral:

- Manter a própria contagem do tempo para coordenar as operações do sistema;
- Guardar os valores de volume máximo do sistema;
- Ler os sinais dos sensores de fluxo;
- Ler o estado dos interruptores de nível de água;
- Atuar a eletroválvula de enchimento do reservatório;
- Acionar os LED conforme o estado do sistema;
- Enviar os dados para o MIU.

A primeira coisa que destacaremos do programa são seus *defines* e nossa única variável global definida no programa principal. Recomendamos que a existência destas informações sejam guardados na sua memória, para que o entendimento dos trechos de código exibidos adiante nesta seção sejam claros e compreensíveis.

```

49 #define MAXIMO PORTAbits.RA1
50 #define MINIMO PORTAbits.RA2
51
52 #define ENCHENDO LATAbits.LATA0
53 #define VALVULA LATAbits.LATA7
54 #define ESVAZIANDO LATAbits.LATA6
55
56 uint16_t Volume_Maximo;

```

Assim como no MIU, no MAPA o nosso código começa configurando o microcontrolador, definindo e inicializando as variáveis locais e já segue para o laço principal. Embora o MAPA seja o módulo mais importante deste trabalho inteiro, seu código é o mais enxuto.

O laço começa conferindo se a *flag* “Atualiza\_Info” é igual a 1. Isto significa que houve uma interrupção do Timer 0, que ocorre a cada 1 segundo. Dentro da interrupção, temos o seguinte código:

```

WWDT_TimerClear();
Pulsos_Entrada = TMR3_ReadTimer();
Pulsos_Saida = TMR5_ReadTimer();
TMR3_WriteTimer(0);
TMR5_WriteTimer(0);
// Contagem de Ticks do sistema
if(SysTicks != 99)
SysTicks++;
else
SysTicks = 0;
Atualiza_Info = 1;

```

Na interrupção, como podemos ver, transferimos os valores dos Timers 3 e 5 para as variáveis “Pulsos\_Entrada” e “Pulsos\_Saida”, que se referem aos pulsos gerados pelos sensores de fluxo na entrada e saída do reservatório. O registrador que armazena o valor de cada *timer* é zerado e, em seguida, é feito o controle da variável que serve para registrar a passagem do tempo (“SysTicks”). Por fim, a *flag* “Atualiza\_Info” recebe o valor 1. Com isto, a primeira parte do código do laço do programa principal entra em execução.

```

if(Atualiza_Info == 1)
{
Volume_Anterior = Volume_Atual;
// Considerou-se 1 pulso = 1 L para facilitar os testes.
if(Pulsos_Entrada > Pulsos_Saida)
{
Volume_Atual = Volume_Anterior + ((Pulsos_Entrada -
Pulsos_Saida) / 1);
if(Volume_Atual > Volume_Maximo )

```



```

Volume_Atual = Volume_Maximo;
}
else
{
if(Volume_Anterior > ((Pulsos_Saida - Pulsos_Entrada) / 1))
Volume_Atual = Volume_Anterior - ((Pulsos_Saida -
Pulsos_Entrada) / 1);
else
Volume_Atual = 0;
}
Porcentagem = Volume_Atual * 100 / Volume_Maximo;
Estima_Tempo++;
if(Estima_Tempo == 10)
{
if(Volume_Atual < Volume_Anterior_Tempo)
Tempo_Restante = Volume_Atual / ((Volume_Anterior_Tempo -
Volume_Atual) * 6);
else
Tempo_Restante = 255;
Volume_Anterior_Tempo = Volume_Atual;
Estima_Tempo = 0;
}
Atualiza_Info = 0;
}

```

A função do código no segundo *if* é calcular o volume de água no reservatório, baseado na seguinte relação:

$$V_{Reservatorio}(t) = V_{Inicial} + (Q_{Entrada} - Q_{Saida}) \cdot t. \quad (4)$$

Como a base de tempo é um segundo, a nossa equação fica semelhante ao código. A única diferença que nota-se é a existência de uma divisão por 1 no código. Isto deve-se ao fato do sensor pretendido para o projeto ter a característica de produzir aproximadamente 480 pulsos por litro, com uma variação em razão do fluxo. Todavia, como substituímos os sensores por um emulador, alteramos esta razão para 1 (ou seja, um pulso equivale a um litro) para simplificar o processo de avaliação.

Adiante, o código dentro do *if* trata de verificar se o volume anteriormente registrado é maior do que o estimado de saída. Caso o volume de água estimado que saiu seja maior do que o estimado existente previamente, o resultado disto seria um valor negativo, o que causaria um *underflow* na nossa variável de controle. Como não faz sentido físico falarmos em volume negativo, empregamos uma condicional para que os resultados negativos sejam desprezados e substituídos por zero. Adiante retomaremos este ponto.

Os passos seguintes deste trecho são calcular o valor restante de água no reservatório na forma percentual e estimar a duração desta quantia restante em função do consumo atual.

Para este cálculo, utilizamos uma base de tempo de dez segundos. Por fim, limpamos a *flag* que nos conduziu a este trecho.

Acima, quando comentamos sobre a possibilidade do equipamento calcular um volume restante negativo, entendemos que esta condição é perfeitamente aceitável, uma vez que temos pela folha de catálogo dos sensores de fluxo, que sua leitura não é exatamente linear, o que poderia implicar em um erro acumulado que levasse à leitura de um volume negativo. Para que o sistema se recupere deste erro acumulado, possuímos as chaves do tipo boia. O código responsável por corrigir isto é simples, como mostrado a seguir.

```
// Redefinição dos níveis com as chaves boia
if((MAXIMO == 0 && Volume_Atual != Volume_Maximo) &&
((VALVULA == 0 && Volume_Atual == Volume_Anterior) || Max_Old == 1))
{
Volume_Atual = Volume_Maximo;
Max_Old = 0;
}
if(MINIMO == 0 && Min_Old == 1 && Volume_Atual != 0)
{
Volume_Atual = 0;
Min_Old = 0;
}
if(MAXIMO == 1 && Max_Old == 0)
Max_Old = 1;
if(MINIMO == 1 && Min_Old == 0)
Min_Old = 1;
```

Max\_Old e Min\_Old são variáveis auxiliares utilizadas para detectar a borda de descida ou subida dos sensores. Este trabalho poderia ser realizado por meio de interrupções, uma vez que o microcontrolador empregado suporta interrupções de borda de subida ou descida nos pinos de entrada e saída, contudo, para esta aplicação não seria necessário empregar interrupção. O simples controle deste evento via software já basta. Lembramos que ambas as chaves utilizam lógica invertida, ou seja, nível lógico 0 implica que a chave está atuada, e vice-versa.

Na redefinição do volume atual para o volume máximo utilizamos a condição do volume atual ser igual ao anterior pois, no momento em que a eletroválvula atua, como a água ainda possui energia cinética, uma vez que a mesma estava em movimento, mesmo após o fechamento da válvula pode haver movimento de água pelo sensor de fluxo, o que provocaria uma leitura de um valor atual maior do que o máximo.

Seguindo na execução do código, baseado nos valores calculados anteriormente, acionar-se-á os LEDs indicadores do MAPA conforme o estado atual, que pode ser enchendo, esvaziando ou nenhuma das duas condições.

```

if(Pulsos_Entrada > Pulsos_Saida)
{
ENCHENDO = 1;
ESVAZIANDO = 0;
}
else if(Pulsos_Entrada < Pulsos_Saida)
{
ENCHENDO = 0;
ESVAZIANDO = 1;
}
else
{
ENCHENDO = 0;
ESVAZIANDO = 0;
}
}

```

O controle da eletroválvula é feito no trecho seguinte, utilizando uma lógica muito simples, como pode ser visto a seguir.

```

// Condições de controle da válvula de entrada
if(MAXIMO == 0 && VALVULA == 1)
VALVULA = 0;
else if(MAXIMO == 1 && Volume_Atual <= (Volume_Maximo * 95 / 100) &&
VALVULA == 0)
VALVULA = 1;

```

A válvula será fechada sempre que o sensor de nível máximo for atuado. Para a abertura desta, será necessário que o volume no interior do reservatório atinja 95% de sua capacidade total. Este mecanismo gera uma histerese similar ao que ocorre com o controle passivo feito por boias em caixas d'água de residências. Isto implica em um menor número de acionamentos da eletroválvula do que se a condição fosse o fim da atuação do sensor de nível máximo, gerando uma redução no consumo de energia, menos chaveamento e prolongamento da vida útil dos componentes.

Os trechos seguintes do código já dizem a respeito da comunicação. Como será notado, acabou que a versão final do protótipo não teve o protocolo MODBUS implementado. Embora toda a parte de hardware tenha sido implementada seguindo o documento “*MODBUS over Serial Line Specification and Implementation guide V1.0*”, da Modbus Organization, organização responsável por estabelecer o padrão MODBUS, a parte em *software* não foi implementada. Uma vez que o foco do projeto era conceber um sistema capaz de detectar a interrupção do fornecimento de água e mensurar o volume de água em um reservatório qualquer, optou-se por empregar esforços nestas soluções.

Antes de efetivamente iniciar a comunicação, o programa deve determinar o valor da mensagem, conforme explicado na seção referente ao MIU.

```

// Mensagem de nível crítico
if(MINIMO == 0)
mensagem = 0x04;
// Mensagem de nível baixo
else if(Porcentagem <= 10 && MINIMO == 1)
mensagem = 0x02;
// Mensagem Falta de água
else if(VALVULA == 1 && Volume_Atual <= Volume_Anterior)
mensagem = 0x01;
// Zera a mensagem caso o sistema esteja operando normalmente
else if((mensagem & 0x07) != 0)
mensagem = 0;

```

Além de servir para informar o MIU, a variável “mensagem” é utilizada dentro da rotina de interrupção do Timer 2 para controlar o LED de aviso do MAPA (LED vermelho, na porção superior direita). As interrupções acontecem a cada cem milissegundos, e a variável “TMR2Ticks” é utilizada como base para a frequência das piscadas do LED.

```

void TMR2_DefaultInterruptHandler(void)
{
if(mensagem == 4)
LATBbits.LATB5 ^= 1;
else if(mensagem == 2 && TMR2Ticks % 5 == 0)
LATBbits.LATB5 ^= 1;
else if(mensagem == 1 && LATBbits.LATB5 == 0)
LATBbits.LATB5 = 1;
else if(mensagem == 0 && LATBbits.LATB5 == 1)
LATBbits.LATB5 = 0;
if(TMR2Ticks < 100)
TMR2Ticks++;
else
TMR2Ticks = 0;
}

```

Desta forma temos o seguinte padrão:

- Se a mensagem for de nível crítico, o LED piscará a 5 Hz;
- Se a mensagem for de nível baixo, o LED piscará a 1 Hz;
- Se a mensagem for de falta de água (interrupção do fornecimento), o LED permanecerá aceso.

Uma vez determinada a mensagem, temos a comunicação, realizada pelo trecho de código a seguir:

```

// Se solicitado, envia as informações paa o MIU
if(EUSART_is_rx_ready())
{
RxData = EUSART_Read();
if(RxData == 'U')
{
__delay_ms(1);
RC1STAbits.CREN = 0;
LATCbits.LATC5 = 1;
__delay_ms(1);
EUSART_Write((uint8_t)Volume_Atual);
while(!EUSART_is_tx_done());
__delay_ms(1);
EUSART_Write((uint8_t)Porcentagem);
while(!EUSART_is_tx_done());
__delay_ms(1);
EUSART_Write((uint8_t)Tempo_Restante);
while(!EUSART_is_tx_done());
__delay_ms(1);
EUSART_Write(mensagem);
while(!EUSART_is_tx_done());
__delay_ms(1);
LATCbits.LATC5 = 0;
RC1STAbits.CREN = 1;
}
}

```

O funcionamento é bem simples e dá-se da seguinte forma:

1. O MIU envia o caracter “U” para o MAPA;
2. Ao receber este caracter, o módulo reconhece que deve enviar as informações e configura o CI responsável pelo padrão RS-485 para o modo “Driver”, ou seja, ele será o atuante na linha, enviando dados;
3. É enviado de forma sequencial o volume atual, a porcentagem, o tempo restante e a mensagem. Todas estas informações possuem um byte de tamanho, cada;
4. Terminado o envio, o CI responsável pelo padrão RS-485 retorna para o modo “Receiver”, ou seja, ele será o passivo da linha, recebendo dados.

Com isto, o programa retorna para o início do laço.

A comunicação pode ser observada na Figura 27, obtida com auxílio de um osciloscópio digital com analisador lógico (Hantek 6022BL).

Nela podemos observar os quatro bytes, referente às quatro informações enviadas. Pudemos observar que, durante a transmissão, no pino Rx do MAPA o analisador detectou

algumas mudanças de nível lógico. Como não é possível transmitir e receber dados ao mesmo tempo na configuração física do projeto, este não representou nenhum problema. Embora tenha-se empregado cabo blindado para a comunicação, como este efeito foi notado apenas na linha Rx do MAPA, temos evidências para afirmar que este problema é decorrente ou de falha no projeto da placa de circuito impresso, que levou a algum acoplamento entre as linhas ou características do próprio CI que lida com a conversão RS-485. Novamente, como a comunicação não foi afetada, não foi realizada a troca do componente para se determinar sua origem.

**Figura 27: Sinal registrado no osciloscópio digital, mostrando a comunicação entre MIU e MAPA.**



**Fonte: Autores**

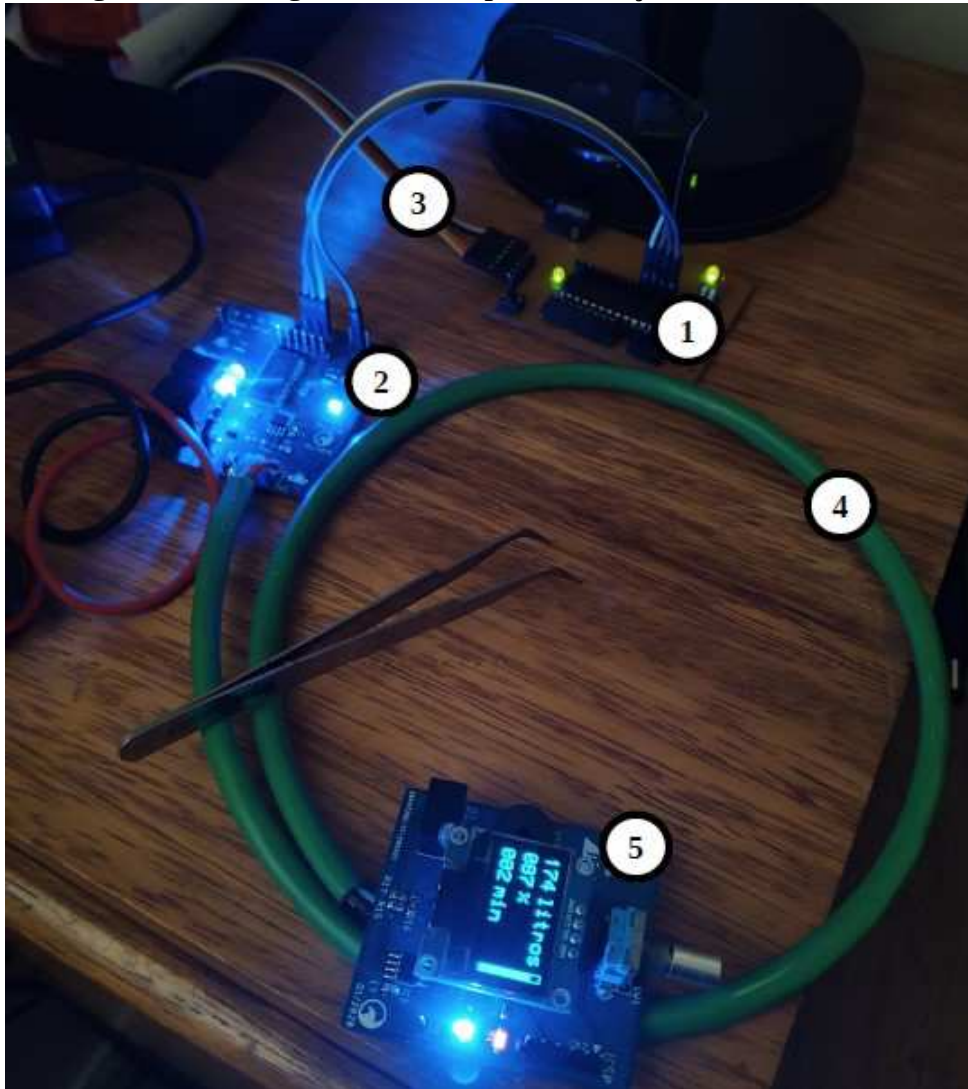
### 4.3 EMULAÇÃO

Para que os sensores fossem emulados, e assim todo o sistema como um todo testado, foi desenvolvido um simples gerador de ondas retangulares empregando um microcontrolador PIC18F2550. Este microcontrolador fornece sinais similares aos dos sensores, e estes sinais são ligados diretamente às entradas do MAPA. Os mesmos parâmetros elétricos foram empregados, diferindo apenas na quantidade de pulsos por litro, como explicado anteriormente.

O emulador (1) foi ligado diretamente ao MAPA (2), como mostrado na Figura 28, sendo inclusive alimentado pelo mesmo. A conexão (3) ao emulador vista na foto é nada além do programador PicKit3, utilizado para fins de depuração do emulador e alteração dos parâmetros utilizados nos testes. Na foto pode-se ver, ainda, o cabo (4) próprio para a comunicação diferencial RS-485 e o MIU (5), alimentado por meio do cabo, sendo a

alimentação compartilhada do MAPA, como explicado anteriormente.

**Figura 28: Montagem do sistema para simulação do funcionamento.**



**Fonte: Autores**

Para averiguar a estimativa de tempo do sistema e de volume, foi executado o seguinte teste:

- 20 amostras da simulação do sistema esvaziando, de forma que:
- 10 com sinal de 1 Hz, ou seja, 1 litro por segundo;
- 10 com sinal de 2 Hz, ou seja, 2 litros por segundo;
- Em cada grupo de 10, 5 amostras foram colhidas enviando 100 pulsos e as outras 5, 200;
- Entre cada amostra, fornecia-se 200 pulsos a 20 Hz na entrada do sensor de entrada, para conferir o funcionamento da mensuração do volume de água que adentra o reservatório.

Após isto, respeitava-se pelo menos dez segundos até o próximo teste, tempo suficiente para que o sistema zere as variáveis utilizadas para a estimativa do tempo de duração do reservatório.

As frequências foram adotadas como um balanço entre tempo e especificações do sistema. Uma vez que a quantidade de pulsos recebido por *Timer* é lida e zerada a cada 1 segundo, não há como receber pulsos com frequências menores que esta, mantendo uma leitura confiável. Da mesma forma, não se utilizou frequências maiores para que o tempo estimado não ficasse abaixo de 1 minuto, o que geraria uma leitura de zero minutos, não tendo nenhum significado para os testes. Os valores obtidos foram os da Tabela 2.

**Tabela 2: Valores obtidos durante emulação dos resultados.**

Amostra	Volume inicial (L)	Volume final (L)	Pulsos	Volume esperado (L)	Tempo estimado (minutos)	Tempo decorrido
1	200	0	200	0	3	00:03:21
2	200	0	200	0	4	00:03:22
3	200	0	200	0	3	00:03:20
4	200	0	200	0	3	00:03:20
5	200	0	200	0	4	00:03:21
6	200	0	200	0	1	00:01:40
7	200	0	200	0	1	00:01:40
8	200	0	200	0	1	00:01:40
9	200	0	200	0	1	00:01:40
10	200	0	200	0	1	00:01:40
11	200	100	100	100	3	00:01:40
12	200	100	100	100	3	00:01:40
13	200	100	100	100	2	00:01:39
14	200	100	100	100	3	00:01:40
15	200	100	100	100	3	00:01:39
16	200	100	100	100	1	00:00:49
17	200	100	100	100	2	00:00:50
18	200	100	100	100	1	00:00:50
19	200	100	100	100	1	00:00:49
20	200	100	100	100	1	00:00:50

**Fonte: Autores**

Como podemos observar na tabela, os valores ficaram dentro do esperado, não sendo observado em nenhuma das 20 amostras, ou seja, em 3000 pulsos nenhum fora perdido pelo MAPA, mesmo na sua resolução limite de 1 Hz. Nota-se, ainda, que respeitado o arredondamento realizado pelo código em execução no microcontrolador, os tempos ficaram também dentro do estimado, sendo que 4 em 20 amostras (20%) diferiram das suas semelhantes, mas nenhuma discrepância fora do esperado.



Ressalta-se, ainda, que da amostra 11 a amostra 15, embora o tempo estimado pareça incorreto, o mesmo encontra-se dentro do que se espera na teoria. Embora o tempo seja, em média, o dobro do decorrido, deve-se lembrar que o sistema estima o tempo para o completo esvaziamento do reservatório, e como utilizamos apenas metade da capacidade máxima configurada no sistema, a razão de dois apresentada é coerente.

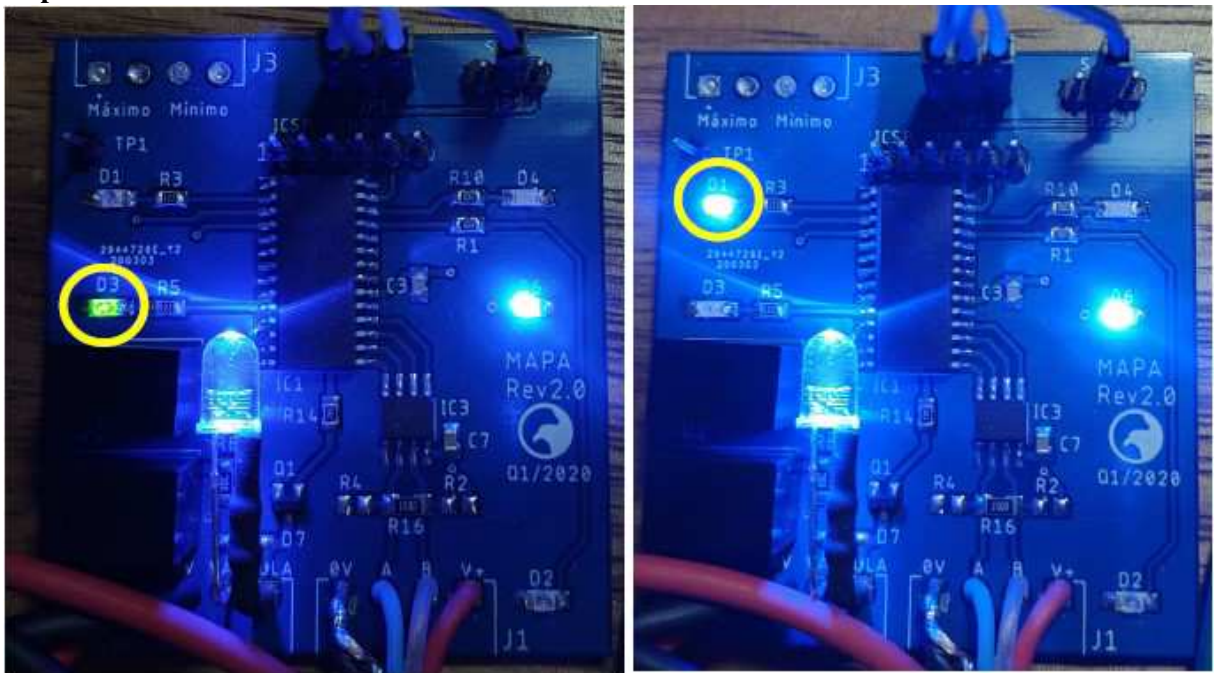
Outro recurso do MAPA testado foi o acionamento da eletroválvula que, como previamente comentado, fora substituída por um LED. Observou-se que, quando o volume indicado atingia 190 litros, o LED era aceso e no segundo seguinte a mensagem “FALTA AGUA” era apresentada no MIU. Aplicando pulsos na entrada do sensor de entrada do reservatório, a mensagem cessava. Ao cessar ambos os pulsos com a saída da eletroválvula acionada, a mensagem retornou. Recordando a lógica de identificação da interrupção do fornecimento de água:

- **SE** eletroválvula está aberta **E** Volume\_Atual é menor ou igual ao Volume\_Anterior, **ENTÃO** fornecimento interrompido.

Desta forma, averiguamos o correto funcionamento da detecção da interrupção do fornecimento de água utilizando uma eletroválvula e um sensor de fluxo, o primeiro objetivo deste trabalho. O segundo, consistindo de desenvolver um sistema capaz de mensurar o volume de água em um reservatório qualquer, foi provado com sucesso também durante os testes, através das amostras colhidas.

Por fim, mas não menos importante, observou-se também a correta detecção do estado do reservatório pelo MAPA, com os LEDs indicadores de reservatório enchendo e esvaziando sendo acionados logo que os respectivos pulsos eram fornecidos, como mostrado na Figura 29.

**Figura 29:** LED de “Reservatório esvaziando” e “Reservatório enchendo” acesos (em destaque), respectivamente.



Fonte: Autores

## 5 CONCLUSÃO

Pensado e elaborado a partir de uma necessidade de um cliente predial, o projeto aqui apresentado buscou atender aplicações comerciais e industriais, e de forma geral, cumpriu seus objetivos, que eram medir o volume de líquido em um reservatório qualquer e identificar a interrupção do fornecimento. Ao nos depararmos com a inexistência de um sistema pronto para esta tarefa, que atendesse a necessidade de um cliente em uma das empresas que dois integrantes deste grupo estagiaram, ficou claro que, em um cenário onde a eficiência energética e o consumo racional de recursos naturais têm sido pauta constante em projetos novos, e obter certificações como a AQUA eleva o valor agregado de uma edificação<sup>1</sup>, a dificuldade em monitorar este precioso recurso era inaceitável.

Contudo, ao pensar rapidamente na solução deste problema, o grupo percebeu outras dificuldades, como alguns sistemas existentes no mercado dependerem da geometria do reservatório de água empregado. Unindo o anseio por uma solução para este problema e aspirando detectar a interrupção do fornecimento de água para auxiliar o gerenciamento do reservatório, este trabalho surgiu, ainda em 2019. Passado um ano, o Estado do Paraná foi assolado por uma forte e duradoura estiagem<sup>2</sup>, e até o momento da redação deste capítulo, nada mudou. A população se viu tendo que, rapidamente, aprender a gerenciar o volume de água que possuía, necessidade reforçada a partir do momento que se deu início ao plano de racionamento do fornecimento de água, chegando a interrupções de 36 horas<sup>3</sup>. Mais do que nunca, o sistema idealizado por este grupo de estudantes mostrava-se útil e justificado.

Encontramos percalços, como a dificuldade em implementar o protocolo Modbus, chave para a integração do projeto em sistemas de automação residencial e predial já existentes, uma vez que, pela experiência do grupo ao atuar na área, era o protocolo mais amplamente utilizado. Contudo, a comunicação entre os módulos foi estabelecida assim mesmo, utilizando

---

<sup>1</sup><https://www.gbcbrasil.org.br/wp-content/uploads/2019/12/Ana%CC%81lise-do-valor-de-locac%CC%A7a%CC%83o-de-imo%CC%81vel-comercial-com-certificado-sustenta%CC%81vel-em-Joinville.pdf>

<sup>2</sup><http://www.aen.pr.gov.br/modules/noticias/article.php?storyid=108289&tit=Estiagem-no-Parana-pode-perdurar-atefevereiro-de-2021>

<sup>3</sup><http://www.aen.pr.gov.br/modules/noticias/article.php?storyid=108325>

protocolo desenvolvido pela própria equipe, para que os dados fossem trocados entre os módulos utilizando toda a camada física já elaborada. Desta forma, pudemos usufruir dos benefícios da comunicação serial RS-485 e dos demais requisitos do protocolo almejado, enquanto aplicamos uma solução mais simples para fins de teste, que satisfizesse as necessidades imediatas do projeto. De todo o projeto, esta foi certamente a parte mais difícil, pois estabelecer uma comunicação funcional apresentou desafios, como erros *Framing* e *Overrun*, lidar com perdas de bytes, mecanismos de correção e tratamento de erros e laços no programa que levavam ao travamento do mesmo quando as informações chegavam diferente do esperado. Embora tenhamos achado um código aberto do protocolo na linguagem de programação utilizada, tivemos dificuldade em implementá-lo no nosso programa, o que foi decisivo para buscarmos outra alternativa. Destacamos que este problema não interferiu de forma alguma em outro requisito para o atendimento do padrão de automação, que é a alimentação em 24 Volts em corrente contínua.

Um ponto detectado nas emulações que pode ser melhorado é a estimativa de duração do volume de água, que atualmente é calculada e exibida em minutos. Percebemos que, por conta dos arredondamentos necessários em função dos tamanhos das variáveis, os valores nem sempre são apresentados da melhor forma. Entendemos que mudar a base de tempo do cálculo para segundos e então exibir o resultado convertido para minutos poderia trazer resultados melhores. Atentamos apenas ao fato de que a base de tempo empregada tem total ligação com as dimensões do projeto. Para grandes volumes de água armazenados, como aplicações prediais, faria muito mais sentido apresentar o resultado em horas, o que implicaria mudanças na programação. O sensor empregado também teria influência nisto, uma vez que a resolução em pulsos/l implicaria na base de tempo utilizada para recuperar os dados do *Timer* que o lê. Quanto mais pulsos por litro tivéssemos, mais precisa e exata seria esta leitura.

Esperamos que a conclusão obtida, de que é possível sim identificar a interrupção do fornecimento de água e estimar com precisão o volume de água em um reservatório qualquer, independentemente de sua forma geométrica, utilizando três componentes principais, sirva de fomento a novos trabalhos que levem ao desenvolvimento de produtos que nos auxiliem na utilização consciente da água. Esperamos ainda, que motivadas por este trabalho, pesquisas sejam realizadas para aprimorar os tipos de sensores utilizados para a medição de fluxo de água, uma vez que poucas soluções comerciais prontas foram encontradas. Sendo este o componente chave para o correto funcionamento deste nosso protótipo, encontrar opções com preço justo e que apresentem boa resolução, exatidão e precisão faz-se imperativo.

## REFERÊNCIAS

- 0825VD. **Cabo De Automação Padrão RS-485 Rolo 100 Metros (blindado)**. Discabos, 2020. Disponível em: [https://www.loja.discabos.com.br/MLB-1584529496-cabo-de-automaco-padro-rs-485-rolo-100-metros-blindado-\\_JM?quantity=1](https://www.loja.discabos.com.br/MLB-1584529496-cabo-de-automaco-padro-rs-485-rolo-100-metros-blindado-_JM?quantity=1). Acesso em: 17 set. 2020.
- 9SS44. **Anúncio do Produto: Sensor de Nível de Água**. FilipeFlop, s.d. Disponível em: [https://www.filipeflop.com/produto/sensor-de-nivel-de-agua/?gclid=CjwKCAiAv4n9BRA9EiwA30WND3b-Z9ONccJnhYZ8wQKRC7BzGp82VW5vA0zDkU03HJeLX3JLPvt-DhoCISMQAvD\\_BwE](https://www.filipeflop.com/produto/sensor-de-nivel-de-agua/?gclid=CjwKCAiAv4n9BRA9EiwA30WND3b-Z9ONccJnhYZ8wQKRC7BzGp82VW5vA0zDkU03HJeLX3JLPvt-DhoCISMQAvD_BwE). Acesso em: 17 set. 2020.
- ANA. **Balanco das Águas**. [S.l.]: Agência Nacional de Águas, 2015/2016.
- APTD3216LZGCK. **Datasheet: 3.2 x 1.6 mm SMD Chip LED Lamp**. Kingbright, 2017. Disponível em: <https://br.mouser.com/datasheet/2/216/APTD3216LZGCK-1102215.pdf>. Acesso em: 17 set. 2020.
- ASSOCIADOS, G. **Perdas De Água 2018 (Snis 2016): Desafios Para Disponibilidade Hídrica E Avanço Da Eficiência Do Saneamento Básico**. Instituto Trata Brasil, São Paulo: [s.n.], 2018. Disponível em: <http://www.tratabrasil.org.br/images/estudos/itb/perdas-2018/estudo-completo.pdf>. Acesso em: 30 abr. 2019.
- BRANQUINHO, O. C. **REDES DE COMUNICAÇÃO DE DADOS SEM FIO: Uma análise de desempenho**. PUC - Campinas, Campinas: [s.n.], 2019. 16 p.
- CANALAGRICOLA. **Válvula Solenóide de Entrada de Água 180° Rosca 1/2"VA03 12V DC**. Canal Agrícola, s.d. Disponível em: <https://br.mouser.com/ProductDetail/Seed-Studio/POW110D3B?qs=Z7P4xsdcg2JwEFWu5erQfA%3D%3D>. Acesso em: 17 set. 2020.
- CASSIOLATO, C. **Medição de Vazão**. Smar Technology Company, s.d. Disponível em: <https://www.smar.com/brasil/artigo-tecnico/medicao-de-vazao>. Acesso em: 17 set. 2020.
- CEM-1206S. **Datasheet: Magnetic Buzzer**. CUI Devices, 2006. Disponível em: [https://br.mouser.com/datasheet/2/670/cs\\_3606-1778024.pdf](https://br.mouser.com/datasheet/2/670/cs_3606-1778024.pdf). Acesso em: 17 set. 2020.
- CUI, D. **Página do Fabricante na Internet**. 2017. Disponível em: <https://www.cuidevices.com/product/motion/rotary-encoders/incremental/panel/mechanical/acz11-series>. Acesso em: 30 nov. 2019.
- CUNHA, J. M. **Protótipo de rede industrial utilizando o padrão serial RS485 e protocolo ModBus**. Universidade Regional de Blumenau, Blumenau: [s.n.], 2000.
- DIGIKEY. **Página do Produto**. 2017. Disponível em: <https://www.digikey.com/product-detail/en/cui-inc/CEM-1206S/102-1155-ND/412414>. Acesso em: 30 nov. 2019.
- FILIFEFLOP. **Catálogo de módulos wireless da Filipeflop**. 2017. Disponível em: <https://www.filipeflop.com/categoria/wireless-e-iot/>. Acesso em: 01 dez. 2019.

FREITAS, C. M. **Redes de comunicação em RS-485**. 2017. Disponível em: <https://www.embarcados.com.br/redes-de-comunicacao-em-rs-485/>. Acesso em: 28 de agosto de 2019.

HUINFINITO. **Anúncio do Produto**. 2017. Disponível em: <https://www.huinfinito.com.br/sensores/924-sensor-de-fluxo-de-agua-3-4.html>. Acesso em: 01 dez. 2019.

INDIAMART. **Anúncio do Produto**. 2017. Disponível em: <https://www.indiamart.com/proddetail/ventury-tube-20154463197.html>. Acesso em: 01 dez. 2019.

JIN, B. *Design and Implementation of an Intrinsically Safe Liquid-Level Sensor Using Coaxial Cable*. Key Laboratory of Advanced Transducers and Intelligent Control Systems, Ministry of Education, Taiyuan University of Technology. Taiyuan. China: [s.n.], 2015. Disponível em: [https://www.researchgate.net/publication/277973474\\_Design\\_and\\_Implementation\\_of\\_an\\_Intrinsically\\_Safe\\_Liquid-Level\\_Sensor\\_Using\\_Coaxial\\_Cable](https://www.researchgate.net/publication/277973474_Design_and_Implementation_of_an_Intrinsically_Safe_Liquid-Level_Sensor_Using_Coaxial_Cable). Acesso em: 17 set. 2020.

JÚNIOR, R. d. C. **Instalações prediais hidráulico-sanitárias: princípios básicos para elaboração de projetos**. São Paulo: Blucher: [s.n.], 2014.

KEELING, R. **Gestão de projetos: uma abordagem global**. 2012.

L78. *Datasheet: Positive voltage regulator ICs*. Kingbright, 2018. Disponível em: <https://br.mouser.com/datasheet/2/389/l78-1849632.pdf>. Acesso em: 17 set. 2020.

MERCADO, L. **Anúncio do Produto**. 2017. Disponível em: <https://produto.mercadolivre.com.br/MLB-696767434-sensor-fluxo-hall-vaso-agua-12-automaco-queima-estoque-JM?quantity=1>. Acesso em: 01 dez. 2019.

MOUSER. **Website do Produto**. 2019. Disponível em: <https://www.mouser.com/new/Texas-Instruments/ti-tm4c123gxl-eval-board/>. Acesso em: 26 de novembro de 2019.

MURATORI, J. R. **AUTOMAÇÃO RESIDENCIAL**. 2011. Disponível em: <https://www.osestoreletrico.com.br/capitulo-ii-cabeamento-residencial-para-dados-voz-e-imagem/>. Acesso em: 27 jul. 2020.

NVR5198NL. *MOSFET – Power, SingleN-Channel, Logic Level, SOT-23*. On Semiconductor, 2019. Disponível em: <https://www.onsemi.com/pub/Collateral/NVR5198NL-D.pdf>. Acesso em: 17 set. 2020.

OPTOELETRONICS. **Página do Fabricante na Internet**. 2017. Disponível em: <https://optoelectronics.liteon.com/en-global/Led/LED-Component/Detail/444>. Acesso em: 30 nov. 2019.

PAGLIA, P. *LDO Thermal Considerations*. Microchip, 2004. Disponível em: <http://ww1.microchip.com/downloads/en/Appnotes/00761b.pdf>. Acesso em: 17 set. 2020.

PIC16F18857-I/SO. *Datasheet: Full-Featured 28/40/44-Pin Microcontrollers*. Mouser, 2020. Disponível em: [https://br.mouser.com/datasheet/2/268/PIC16LF1885777\\_Data\\_Sheet\\_40001825E-1903820.pdf](https://br.mouser.com/datasheet/2/268/PIC16LF1885777_Data_Sheet_40001825E-1903820.pdf). Acesso em: 17 set. 2020.

POW110D3B. **Seeed Studio.** Mouser, s.d. Disponível em: <https://br.mouser.com/ProductDetail/Seeed-Studio/POW110D3B?qs=Z7P4xscdg2JwEFWu5erQfA%3D%3D>. Acesso em: 17 set. 2020.

RASPBERRY. **Website do Produto.** 2019. Disponível em: <https://www.raspberrypi.org/products/>. Acesso em: 26 de novembro de 2019.

ROSE, B. *Buzzer Basics - Technologies, Tones, and Drive Circuits.* CUI Devices, 2018. Disponível em: <https://www.cuidevices.com/blog/buzzer-basics-technologies-tones-and-driving-circuits>. Acesso em: 17 set. 2020.

SHINDER, D. L. *It's 2017 - Why haven't all businesses gone wireless?* 2014. Disponível em: <https://techtalk.gfi.com/its-2014-why-havent-all-businesses-gone-wireless/>. Acesso em: 27 jul. 2020.

TEXAS, I. *The RS-485 Design Guide.* 2016. Disponível em: <http://www.ti.com/lit/an/slla272c/slla272c.pdf>. Acesso em: 27 jul. 2020.

TM4C123GH6PM. *Datasheet: Tiva™ TM4C123GH6PM Microcontroller.* Texas Instruments, 2020. Disponível em: <https://www.ti.com/product/TM4C123GH6PM>. Acesso em: 17 set. 2020.

TME. **Website do Produto.** 2019. Disponível em: [https://www.tme.eu/en/details/pic16f18857-i\\_sp/8-bit-pic-family/microchip-technology/](https://www.tme.eu/en/details/pic16f18857-i_sp/8-bit-pic-family/microchip-technology/). Acesso em: 26 de novembro de 2019.

VX78-500. *Datasheet: NON-ISOLATED DC SWITCHING REGULATOR.* CUI Devices, 2017. Disponível em: <https://www.cui.com/product/resource/pdf/vx78-500.pdf>. Acesso em: 17 set. 2020.

WINSTAR, O. *4 Pin OLED Display (COG+PCB) 0.96", 128x64.* 2020. Disponível em: <https://www.winstar.com.tw/products/oled-module/graphic-oled-display/4-pin-oled.html>. Acesso em: 27 jul. 2020.

## APÊNDICE A – *FIRMWARE*

### A.1 INICIALIZAÇÃO

```

375 //===== Main application =====//
376 void main(void)
377 {
378 uint8_t Atualiza_Display = 0, ChA_Old, SW_Old, n;
379
380 // initialize the device
381 SYSTEM_Initialize();
382 WWDT_TimerClear();
383 WWDT_SoftEnable();
384
385 // Habilita as interrupções
386 INTERRUPT_GlobalInterruptEnable();
387 INTERRUPT_PeripheralInterruptEnable();
388
389 // Inicializa o display
390 Initialize_SSD1306();
391
392 // Saudação do MIU
393 Saudacao();
394
395 // Recupera parâmetros ajustados previamente
396 Restaura_configuracoes_previas();
397
398 // Seleciona a Tela Principal
399 Tela = Tela_Principal;
400
401 ChA_Old = Estado_ENC_CH_A;
402 SW_Old = Estado_ENC_SW;

```

### A.2 LÓGICA DO *ENCODER*

```

440 // Analisa o comportamento do Encoder
441 if((Estado_ENC_CH_A == 0 && ChA_Old == 0xFF))
442 {
443 // Ajusta parâmetro de acordo com o movimento do encoder
444 if(Estado_ENC_CH_B == 0 && Timeout_Tela != 0 && Tela ==Tela_Ajuste_Volume)
446 {
447 if(Volume_Maximo < 250)

```



```

448 {
449 Volume_Maximo++;
450 Reproduz_Som(941, 50);
451 }
452 else
453 Reproduz_Som(2400, 200);
454 }
455 else if(Estado_ENC_CH_B == 0 && Timeout_Tela != 0 && Tela == Tela_Ajuste_Tempo)
456 {
457 if(Tempo_Tela_Ajuste < 60)
458 {
459 Temp_o_Tela_Ajuste += 5;
460 Reproduz_Som(1056, 100);
461 }
462 }
463 else
464 Reproduz_Som(2400, 200);
465 }
466 else if(Estado_ENC_CH_B == 0xFF && Timeout_Tela != 0 && Tela == Tela_Ajuste_Volume)
467 {
468 if(Volume_Maximo > 0)
469 {
470 Volume_Maximo--;
471 Reproduz_Som(888, 50);
472 }
473 }
474 else
475 Reproduz_Som(2400, 200);
476 }
477 else if(Estado_ENC_CH_B == 0xFF && Timeout_Tela != 0 && Tela == Tela_Ajuste_Tempo)
478 {
479 if(Tempo_Tela_Ajuste > 5)
480 {
481 Temp_o_Tela_Ajuste -= 5;
482 Reproduz_Som(997, 100);
483 }
484 }
485 else
486 Reproduz_Som(2400, 200);
487 }
488
489 ChA_Old = 0;
490 }
491 else if(ChA_Old == 0 && Estado_ENC_CH_A == 0xFF)
492 ChA_Old = 0xFF;

```

### A.3 ESTADO DA CHAVE DO *ENCODER*

```

494 if((Estado_ENC_SW == 0 && SW_Old == 0xFF))
495 {
496 Temp_o_Botao = 0;
497 SW_Old = 0;
498 Reproduz_Som(1185, 100);
499 Timeout_Alarme = 0;
500 Silencia_Alarme = 1;

```

```

501 }
502 else if(SW_Old == 0 && Estado_ENC_SW == 0xFF)
503 {
504 SW_Old = 0xFF;
505
506 if(Tempo_Botao >= 3000 && Tela == Tela_Principal && Wake_Up == 0)
507 Tela = Tela_Ajuste_Volume;
508 else if(Tempo_Botao >= 3000 && Tela != Tela_Principal && Wake_Up == 0)
509 {
510 Clear_Buffer();
511 GD_Write_String("Operacao", 15, 14, _10x16);
512 GD_Write_String("Cancelada!", 3, 32, _10x16);
513 GD_Screen_Update();
514 Reproduz_Som(880, 100);
515 while(Tempo_Som != 0);
516 Reproduz_Som(740, 100);
517 while(Tempo_Som != 0);
518 Reproduz_Som(660, 100);
519 while(Tempo_Som != 0);
520 __delay_ms(700);
521
522
523 Tela = Tela_Principal;
524 }
525 else if(Tempo_Botao < 3000 && Wake_Up == 0)
526 {
527 if(Tela == Tela_Ajuste_Volume)
528 Tela = Tela_Ajuste_Tempo;
529 else if(Tela == Tela_Ajuste_Tempo)
530 {
531 Tempo_Tela = (uint16_t)Tempo_Tela_Ajuste * 1000;
532 n = 0;
533
534 // Salva novo parâmetro de tempo na EEPROM e checa a
535 // consistência dos dados
536 do
537 {
538 DATAEE_WriteByte(0xF000, Tempo_Tela_Ajuste);
539 __delay_ms(100);
540 Tempo_Tela = 1000 * (uint16_t)DATAEE_ReadByte(0xF000);
541 n++;
542 }while(n != 3 && Tempo_Tela != Tempo_Tela_Ajuste * 1000);
543 if(n == 3)
544 Falha_EEPROM();
545
546 /*inserir aqui a gravação da configuração de
547 * volume máximo presente no MAPA*/
548 Clear_Buffer();
549 GD_Write_String("Parametros", 3, 14, _10x16);
550 GD_Write_String("Salvos!", 21, 32, _10x16);
551 GD_Screen_Update();
552 Reproduz_Som(660, 100);
553 while(Tempo_Som != 0);
554 Reproduz_Som(740, 100);
555 while(Tempo_Som != 0);

```

```

556 Reproduz_Som(880, 100);
557 while(Tempo_Som != 0);
558
559 Tela = Tela_Principal;
560 }
561 }
562
563 if(Wake_Up)
564 {
565 Wake_Up = 0;
566 Atualiza_Informacoes();
567 Exibe_Mensagem();
568 Atualiza_Display = 1;
569 }
570 }

```

#### A.4 CONTROLE DO CICLO ATIVO DO PWM

```

226 // Gerencia o PWM pra cada estado do LED
227 if(Estado_LED == 0 && PWM1LDV != 0)
228 PWM1_LoadDutyValue(0);
229 else if(Estado_LED == 1 && PWM1LDV != 1023)
230 PWM1_LoadDutyValue(1023);
231 else if(Estado_LED == 2)
232 {
233 if(SysTicks % 500 == 0)
234 {
235 if(PWM1LDV == 1023)
236 PWM1_LoadDutyValue(0);
237 else
238 PWM1_LoadDutyValue(1023);
239 }
240 }
241 else if(Estado_LED == 8)
242 {
243 if(SysTicks % 100 == 0)
244 {
245 if(PWM1LDV == 1023)
246 PWM1_LoadDutyValue(0);
247 else
248 PWM1_LoadDutyValue(1023);
249 }
250 }
251 else if(Estado_LED == 0x04 || Estado_LED == 0x14)
252 {
253 if( (Estado_LED & 0x10) == 0)
254 {
255 if(PWM1LDV != 1023)
256 PWM1_LoadDutyValue(PWM1LDV + 1);
257 else
258 {
259 PWM1_LoadDutyValue(PWM1LDV - 1);

```

```
260 Estado_LED = 0x14;
261 }
262 }
263
264 else
265 {
266 if(PWM1LDV != 0)
267 PWM1_LoadDutyValue(PWM1LDV - 1);
268 else
269 {
270 PWM1_LoadDutyValue(PWM1LDV + 1);
271 Estado_LED = 0x04;
272 }
273 }
274 }
```