



Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica



Trabalho de Conclusão de Curso

Estudo de técnicas para monitoramento e controle de instrumentos a partir de uma rede CAN

Rafael Pires Mota

João Monlevade, MG
2024

Rafael Pires Mota

Estudo de técnicas para monitoramento e controle de instrumentos a partir de uma rede CAN

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.

Orientador: Prof. Marcelo Moreira Tiago

Universidade Federal de Ouro Preto
João Monlevade
2024

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

M917e Mota, Rafael Pires.

Estudo de técnicas para monitoramento e controle de instrumentos a partir de uma rede CAN. [manuscrito] / Rafael Pires Mota. - 2024.
69 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Marcelo Moreira Tiago.

Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia Elétrica .

1. Controle automático. 2. Microcontroladores. 3. Processamento de sinais. 4. Protocolo de aplicação sem fio (Protocolo de rede de computadores). 5. Rede CAN (Rede de computador). 6. Redes de sensores sem fio. I. Tiago, Marcelo Moreira. II. Universidade Federal de Ouro Preto. III. Título.

CDU 681.5

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



FOLHA DE APROVAÇÃO

Rafael Pires Mota

Estudo de técnicas para monitoramento e controle de dispositivos a partir de uma rede CAN

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia Elétrica.

Aprovada em 05 de abril de 2024.

Membros da banca

Dr. Marcelo Moreira Tiago — Orientador — Universidade Federal de Ouro Preto
Dr. Igor Dias Neto de Souza — Universidade Federal de Ouro Preto
Dr. Márcio Feliciano Braga — Universidade Federal de Ouro Preto

Marcelo Moreira Tiago, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 10/06/2024.



Documento assinado eletronicamente por **Marcelo Moreira Tiago, PROFESSOR DE MAGISTERIO SUPERIOR**, em 10/06/2024, às 14:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0719941** e o código CRC **F83B4F18**.

Dedico este trabalho aos meus amados pais, cujo apoio foi a bússola que guiou cada passo desta jornada. Suas palavras de encorajamento e sua crença constante em mim foram os pilares que sustentaram minha caminhada até este momento. A vocês, que sempre estiveram ao meu lado nos momentos de dúvida e nas vitórias mais significativas, expresso minha mais profunda gratidão. Seus ensinamentos, valores e exemplo inspirador moldaram o ser humano que almejo ser. Que esta conquista seja também um reflexo do amor e dedicação que vocês depositaram em mim desde o primeiro dia. Este trabalho é dedicado a vocês com todo o meu amor e eterna gratidão.

Agradecimentos

Gostaria de agradecer a minha família, que sempre me apoiou e me deu força para perseguir meus sonhos. Sem o amor, encorajamento e suporte deles, eu não teria conseguido chegar até aqui.

Também quero expressar minha gratidão ao meu orientador, Marcelo Moreira Tiago, cuja orientação e conhecimento foram fundamentais para o sucesso deste trabalho.

Por fim, gostaria de agradecer a todos os amigos e as pessoas que me apoiaram ao longo do caminho. Suas palavras de incentivo, encorajamento e amizade foram muito importantes para mim.

"Aprender é a única coisa que a mente nunca se cansa, nunca tem medo e nunca se arrepende."

– Leonardo da Vinci

Resumo

Este trabalho detalha o projeto e a implementação de uma rede de sensores microcontrolados interligados através de uma rede CAN, do inglês *Controller Area Network* (CAN). Uma planta piloto, composta por um sistema de tanques interconectados, sensores e atuadores, foi desenvolvida para avaliar o desempenho da rede numa aplicação envolvendo monitoramento e controle de instrumentos. Microcontroladores foram utilizados para adquirir informações dos sensores e aplicar sinais de controle nos atuadores, e *transceivers* foram empregados para implementar a rede CAN utilizada para integrar todos os dispositivos. Os ensaios envolveram o uso de mensagens com identificadores de 11 bits, transmitindo 2 *bytes* de dados por mensagem a uma taxa de 125 kbps. A integridade dos sinais elétricos transmitidos foi verificada usando um analisador lógico, evidenciando a capacidade da rede CAN de recuperar mensagens corretamente, em situações em que ocorrem erros de transmissão. Ao final, optou-se por conectar um dos microcontroladores a um servidor OPC, do inglês *OLE for Process Control* (OPC), sistema frequentemente utilizado em aplicações industriais para simplificar o processo de troca de informações entre dispositivos produzidos por diferentes fabricantes. Dados recebidos foram armazenados e visualizados graficamente em uma interface de usuário desenvolvida no MATLAB, atualizada em tempo real, ilustrando a eficácia do sistema para aplicações de monitoramento.

Palavras-chave: Rede CAN, Servidor OPC, Microcontrolador, Instrumentação Industrial.

Abstract

This work details the design and implementation of a network of microcontrolled sensors interconnected through a CAN network. A pilot plant, composed of a system of interconnected tanks, sensors, and actuators, was developed to evaluate the performance of the network in an application involving monitoring and control of instruments. Microcontrollers were used to acquire information from the sensors and apply control signals to the actuators, and transceivers were employed to implement the CAN network used to integrate all the devices. The tests involved the use of messages with 11-bit identifiers, transmitting 2 bytes of data per message at a rate of 125 kbps. The integrity of the transmitted electrical signals was verified using a logic analyzer, demonstrating the CAN network's ability to correctly recover messages in situations where transmission errors occur. In the end, it was decided to connect one of the microcontrollers to an OPC server, a system frequently used in industrial applications to simplify the process of exchanging information between devices produced by different manufacturers. Received data was stored and graphically displayed in a user interface developed in MATLAB, updated in real-time, illustrating the system's effectiveness for monitoring applications.

Keywords: CAN Network, OPC Server, Microcontroller, Industrial Instrumentation.

Lista de ilustrações

Figura 1 – A arquitetura padronizada em camadas ISO 11898.	9
Figura 2 – Topologia e componentes das redes CAN.	10
Figura 3 – Lógica invertida de um barramento CAN.	11
Figura 4 – Arbitragem em um barramento CAN.	12
Figura 5 – Formato de quadro de mensagem	14
Figura 6 – Redes CAN reduzem significativamente a fiação.	17
Figura 7 – Demografia de membros da fundação OPC.	19
Figura 8 – Caso típico de uso de clientes e servidores OPC.	20
Figura 9 – Arquitetura em camadas do OPC UA.	22
Figura 10 – Diagrama de blocos.	25
Figura 11 – Exemplo da implementação do sistema.	25
Figura 12 – Módulo CAN MCP2515 - TJA1050.	26
Figura 13 – Circuito utilizado ao longo dos ensaios propostos neste trabalho.	27
Figura 14 – Diagrama de blocos da configuração do servidor OPC.	28
Figura 15 – Diagrama de instrumentação e tubulação implementado.	29
Figura 16 – Componentes da válvula motorizada.	31
Figura 17 – Válvula solenoide.	32
Figura 18 – Sensor de Vazão YF-S201.	33
Figura 19 – Sensor VL53L0X.	33
Figura 20 – Bomba de água 12V.	34
Figura 21 – Analisador lógico.	36
Figura 22 – Mensagens recebidas pelo módulo receptor.	37
Figura 23 – Dados coletados pelo <i>software</i> da Salae.	38
Figura 24 – Valores digitais recebidos pelo módulo receptor. Os dados foram adquiridos pelo módulo transmissor e enviados pela rede CAN.	38
Figura 25 – Dados coletados pelo <i>software</i> da Salae.	39
Figura 26 – Tela de configuração do servidor OPC.	39
Figura 27 – Tela principal exibida pelo <i>software</i> OPC client dOPC.	40
Figura 28 – Conexão do cliente OPC no servidor OPC.l.	41
Figura 29 – Status da conexão estabelecida entre servidor OPC e Matlab OPC <i>toolbox</i>	41
Figura 30 – Gráfico em tempo real dos valores do potenciômetro no MATLAB.	42
Figura 31 – Planta composta por quatro tanques interconectados.	43
Figura 32 – Dados coletados pelo software da Salae.	44
Figura 33 – Interface do Cliente OPC Elipse E3 Studio.	45
Figura 34 – Interface Gráfica Desenvolvida.	46

Lista de tabelas

Tabela 1 – Conexões entre microcontrolador Atmega 328p e módulo MCP2515. . .	26
Tabela 2 – Tabela de Identificadores (TAGs, do inglês <i>Tags</i>) (TAGs) de equipamentos conforme Norma ISA	30
Tabela 3 – Especificações das válvulas esfera motorizadas	32
Tabela 4 – Especificações das válvulas solenoides on/Off	32
Tabela 5 – Especificações dos sensores de Vazão	33
Tabela 6 – Especificações dos Sensores de Nível VL53L0X	34
Tabela 7 – Especificações da bomba de água 12V	34
Tabela 8 – Detalhes das mensagens CAN	44

Lista de siglas e abreviaturas

A&E	Alarme e Evento (A&E, do inglês <i>Alarm and Event</i>)
ACK	Reconhecimento (ACK, do inglês <i>Acknowledgment</i>)
CAN	CAN, do inglês <i>Controller Area Network</i>
CiA	CiA, do inglês <i>CAN in Automation</i>
COM	COM, do inglês <i>Component Object Model</i>
CRC	Verificação de Redundância Cíclica (CRC, do inglês <i>Cyclic Redundancy Check</i>)
CSMA	Acesso Múltiplo por Senso de Portadora (CSMA, do inglês <i>Carrier-Sense Multiple Access</i>)
DA	Acesso a Dados (DA, do inglês <i>Data Access</i>)
DCOM	Modelo de Objeto Componente Distribuído (DCOM, do inglês <i>Distributed Component Object Model</i>)
EOF	Fim do Quadro (EOF, do inglês <i>End of Frame</i>)
FIT	Transmissor Indicador de Vazão (FIT, do inglês <i>Flow Indicating Transmitter</i>)
FY	Atuador de Controle de Vazão (FY, do inglês <i>Flow Relay</i>)
HDA	Acesso a Dados Históricos (HDA, do inglês <i>Historical Data Access</i>)
IDE	Extensão de Identificador (IDE, do inglês <i>Identifier Extension</i>)
IHM	Interface Homem-Máquina (IHM, do inglês <i>Human-Machine Interface</i>)
ISA	Sociedade Internacional de Automação (ISA, do inglês <i>International Society of Automation</i>)
ISO	Organização Internacional para Padronização (ISO, do inglês <i>International Organization for Standardization</i>)
LT	Transmissor de Nível (LT, do inglês <i>Level Transmitter</i>)
NRZ	Não-retorno-a-zero (NRZ, do inglês <i>Non-return-to-zero</i>)
OPC	OPC, do inglês <i>OLE for Process Control</i>
OPC UA	OLE para Controle de Processo Arquitetura Unificada (OPC UA, do inglês <i>OLE for Process Control Unified Architecture</i>)
PAM	Modulação por Amplitude de Pulso (PAM, do inglês <i>Pulse Amplitude Modulation</i>)
RTR	Solicitação de Transmissão Remota (RTR, do inglês <i>Remote Transmission Request</i>)

SCADA	Controle de Supervisão e Aquisição de Dados (SCADA, do inglês <i>Supervisory Control And Data Acquisition</i>)
SOF	Início do Quadro (SOF, do inglês <i>Start of Frame</i>)
SPI	Interface de Periféricos Seriais (SPI, do inglês <i>Serial Peripheral Interface</i>)
SRR	Solicitação de Substituição Remota (SRR, do inglês <i>Substitute Remote Request</i>)
TAGs	Identificadores (TAGs, do inglês <i>Tags</i>)
UART	Receptor/Transmissor Assíncrono Universal (UART, do inglês <i>Universal Asynchronous Receiver/Transmitter</i>)

Sumário

1	INTRODUÇÃO	1
1.1	Justificativa	3
1.2	Objetivos	3
1.3	Revisão bibliográfica	4
1.4	Estrutura do trabalho	5
2	REVISÃO TEÓRICA	7
2.1	Redes CAN	7
2.1.1	Camadas de rede	8
2.1.2	Topologia de rede CAN	9
2.1.3	Codificação binária	11
2.1.4	Formato do quadro de mensagens	12
2.1.5	Mecanismos de controle de acesso ao meio	13
2.1.6	Gerenciamento de erros	14
2.1.7	Arquitetura de controladores CAN	15
2.1.8	Desempenho	16
2.1.9	Normas	17
2.2	Plataforma de comunicação aberta OPC	18
2.2.1	Configuração clássica	19
2.2.2	Arquitetura unificada	20
2.3	Considerações parciais	22
3	METODOLOGIA	24
3.1	Rede CAN	24
3.1.1	Conexão elétrica entre os módulos	26
3.2	Servidor OPC	28
3.3	Diagrama de instrumentação e tubulação	28
3.3.1	Identificação de equipamentos segundo a norma ISA	30
3.3.2	Composição da planta	31
3.4	Integração do sistema supervisório com o servidor OPC	35
3.5	Considerações parciais	35
4	RESULTADOS E DISCUSSÕES	36
4.1	Análise dos sinais transmitidos pela rede	36
4.2	Servidor OPC	39
4.3	Projeto de uma planta piloto	42

4.3.1	Análise do sistema	42
4.3.1.1	Cliente OPC	45
4.3.1.2	Supervisório	45
4.4	Considerações Parciais	46
5	CONSIDERAÇÕES FINAIS	48
5.1	Sugestões de trabalhos futuros	49
	REFERÊNCIAS	50
	 APÊNDICES	 52
	Apêndice A – DIAGRAMA P&I	53
A.1	Diagrama de instrumentação e tubulação implementado	53
A.2	Tabela com a simbologia	54
	Apêndice B – FORMAS DE ONDA	55
B.1	Sensores de fluxo FIT-11 e FIT-21.	55
B.2	Sensores de nível LT-15, LT-16.	56
B.3	Sensores de nível LT-22 e LT-23.	57

1 Introdução

Desde o começo da primeira revolução industrial, por volta de 1760, a indústria tem passado por transformações e evoluções. Esse processo começou com a introdução da máquina a vapor e da produção em massa. Posteriormente, avançou com a popularização da eletricidade e o início da era da automação. Finalmente, chegamos à fase da revolução digital na década de 1950, marcada pelo uso generalizado de computadores e pela digitalização de informações em aplicações industriais (SACOMANO et al., 2018).

A redução dos custos de sistemas microprocessados e microcontrolados fez com que aplicações industriais passassem a utilizar cada vez mais aplicações envolvendo algoritmos para processamento digital de sinais, aprendizado de máquina, estimação de parâmetros e, atualmente, sistemas de inteligência artificial. Esses avanços, somados à integração de sistemas de monitoramento e controle de dispositivos por meio de redes de comunicação cabeadas ou sem fio, constituem um cenário definido atualmente como quarta revolução industrial.

Segundo Sacomano et al. (2018) a indústria 4.0 tem como característica o desenvolvimento de aplicações integradas à internet, permitindo que os dispositivos possam compartilhar informações por meio de uma nuvem. Nesse sentido, o avanço de sistema de comunicação *wireless* e 5G vem contribuindo de maneira significativo para o desenvolvimento dos mais diversos tipos de aplicação.

Tendo em vista esse crescente avanço tecnológico, percebe-se que a conectividade nas indústrias tem o propósito de viabilizar uma maior facilidade de acesso aos dados das empresas, seja para uma análise mais criteriosa da produção, a identificação de falhas ou mesmo para garantir uma comunicação entre setores e máquinas. Com o aumento da utilização de sistemas embarcados em ambientes industriais, tornou-se necessário empregar protocolos de comunicação preestabelecidos para ser possível conectar equipamentos desenvolvidos por diferentes fabricantes a uma mesma rede. Esses sistemas embarcados definem um padrão para comunicação e trocas de mensagens entre diferentes sistemas utilizados para monitoramento e controle e também entre sistemas computacionais responsáveis por centralizar as informações e, em alguns casos, gerenciar os dados armazenados em nuvem (MAHNKE; LEITNER; DAMM, 2009).

Nesse sentido, surge a rede CAN, do inglês *Controller Area Network* (CAN). Trata-se de um protocolo de comunicação baseado em *multicast*, que possibilita um controle distribuído de equipamentos em tempo real, com elevado nível de segurança. Esse protocolo foi desenvolvido em 1986 pela empresa Bosh, sendo projetado inicialmente para aplicações automotivas. Nesse período, a indústria automobilística passava por um processo de evolução, resultado da adição de diversos sistemas eletrônicos embarcados aos veículos que, até então, eram compostos por dispositivos puramente mecânicos. Esse processo per-

mitiu que diversas aplicações comuns atualmente, como controle eletrônico de aparelhos de ar-condicionado, controle de tração e estabilidade do veículo, além de sistemas para monitoramento de parâmetros relacionados ao consumo do veículo, controle do motor e câmbio, entre outros, pudessem ser implementados (KIM; TRAN-DANG, 2019; NATALE et al., 2012).

Ao longo dos anos, a utilização da rede CAN se expandiu para outros setores industriais, tais como controle de processos, aquisição de dados e automação de fábricas, resultando em uma expansão do seu uso em áreas como o setor aeroespacial, em aplicações agrícolas e na área médica.

Por outro lado, a rede CAN permitiu que equipamentos desenvolvidos por diferentes fabricantes pudessem operar numa mesma rede física, mas os protocolos de transmissão de dados utilizados por cada fabricante eram diferentes, inviabilizando a troca direta de informações. Essa característica dificultava o gerenciamento dos dispositivos a partir de um mesmo ambiente, haja vista que os protocolos de rede eram os mesmos, mas os protocolos de transmissão de dados eram diferentes.

Para ser possível centralizar todas as informações num mesmo servidor, uma plataforma de comunicação aberta, *OLE for Process Control*(OPC), foi desenvolvida. Essa plataforma é gerenciada pela Fundação OPC, uma organização sem fins lucrativos, que disponibiliza as informações livremente para serem utilizadas por diferentes fabricantes (MAHNKE; LEITNER; DAMM, 2009).

Segundo Mahnke, Leitner e Damm (2009), diversos fabricantes, que desenvolvem sistemas para automação industrial, tornaram-se membros da fundação e passaram a fornecer soluções compatíveis com essa plataforma, o que permitiu que soluções desenvolvidas por fornecedores diferentes pudessem ser integradas num mesmo *software* de gerenciamento. Os principais exemplos de aplicação para essa plataforma estão relacionados ao desenvolvimento de interfaces para aplicações de automação industrial, como interfaces homem-máquina, IHM e sistemas de controle supervisão e aquisição de dados, SCADA, que permitem monitorar os dados atuais compartilhados por dispositivos, além de fornecer informações relacionadas a eventos específicos e históricos para aplicativos de gerenciamento.

Dessa forma, este trabalho propõe-se a enfrentar o desafio de implementar, monitorar e controlar um sistema de tanques que simula um ambiente de processo industrial. O desenvolvimento concentra-se em um sistema de aquisição de dados capaz de capturar leituras precisas de sensores de nível e vazão. Esses dados serão transmitidos a um microcontrolador através da rede CAN, e, por meio do protocolo OPC, será possível gerenciar de maneira eficaz válvulas motorizadas, solenóides e bombas. O sistema integrado não apenas facilitará a gestão operacional, mas também refletirá os princípios da Indústria 4.0, promovendo a eficiência e o controle em tempo real.

1.1 Justificativa

A automação industrial desempenha um papel fundamental na evolução dos processos de produção, sendo essencial para a otimização de recursos, o aumento da eficiência e a garantia da segurança operacional. Em um cenário no qual a Indústria 4.0 dita o ritmo de transformações tecnológicas, o monitoramento e controle de sistemas de automação tornam-se elementos estratégicos para a sustentabilidade e competitividade das empresas. Neste contexto, de acordo com Kim e Tran-Dang (2019), a rede CAN, surge como uma solução robusta, oferecendo uma comunicação confiável entre diversos instrumentos em ambientes industriais complexos.

Este trabalho justifica-se pela necessidade de compreender e aperfeiçoar as técnicas de monitoramento e controle em sistemas automatizados, tendo em vista a sua aplicabilidade em uma ampla gama de setores industriais. A escolha de um sistema de tanques, simulando uma planta industrial como objeto de estudo, se reflete em processos que exigem gestão de fluidos e controle de níveis, essenciais em diversas aplicações, desde a petroquímica até o tratamento de águas residuais.

A rede CAN, consolidada e amplamente utilizada no setor automotivo, tem seu potencial ainda subutilizado em outros ramos da automação. Portanto, a investigação de seu uso em outras aplicações industriais, como no caso deste estudo, contribui para o fechamento de uma lacuna técnica importante, permitindo a transferência de conhecimento e a adaptação de técnicas entre setores.

A relevância deste trabalho se estende também ao campo acadêmico, no qual os resultados podem servir como base para futuras pesquisas e desenvolvimentos no campo da automação e controle de processos. De maneira prática, a aplicação das técnicas estudadas pode resultar em melhorias diretas na eficácia operacional, na redução de custos e na sustentabilidade ambiental das plantas industriais.

1.2 Objetivos

O objetivo geral deste trabalho é implementar e avaliar uma rede de comunicação CAN entre dispositivos microcontrolados em uma planta industrial simulada, especificamente por um conjunto de tanques interconectados, empregando o protocolo OPC para a sincronização de dados, com a expectativa de alcançar uma comunicação eficiente e a criação de uma Interface Homem-Máquina para monitoramento remoto e controle preciso de medidores e atuadores. Além disso, tem-se como objetivos específicos:

- Compreender os princípios de funcionamento e normas para utilização de uma rede CAN;
- Utilizar o protocolo aberto OPC para implementar um servidor responsável por sincronizar o processo de transferência de dados entre os dispositivos e um computador;

- Projetar uma planta piloto;
- Desenvolver soluções de *hardware* para implementar o sistema proposto;
- Realizar testes de comunicação entre instrumentos e computador;
- Projetar uma IHM para monitorar as informações e controlar os atuadores remotamente.

1.3 Revisão bibliográfica

Nesta seção serão apresentados os principais artigos utilizados como referência ao longo do desenvolvimento deste trabalho.

O trabalho apresentado por Ma (2021) descreve um estudo do uso da rede de comunicação CAN para o monitoramento dos parâmetros da rede elétrica de minas de carvão. O sistema desenvolvido visa melhorar a segurança da rede elétrica, buscando reduzir a ocorrência de acidentes ao longo do processo de produção e aumentando o volume de material produzido na mina.

Elert (2022) apresenta a modelagem de um sistema de comunicação envolvendo microcontroladores F28379D da Texas Instruments e sensores responsáveis por medir parâmetros de voo de mísseis através da rede CAN. A implementação do modelo foi desenvolvida usando o *software* MATLAB Simulink, e a mesma ferramenta foi empregada para gerar os algoritmos usados. Esses algoritmos foram implementados nos microcontroladores por meio de uma interface gráfica do próprio MATLAB Simulink, facilitando a programação e os testes dos dispositivos.

Cao, Ye e Li (2015) desenvolveram um sistema de controle aplicado a um forno industrial que utilizava um servidor implementado via plataforma OPC para controlar o processo de transferência de informações entre sistema embarcado e computador. O MATLAB foi utilizado para implementar uma IHM, e um controlador *fuzzy* proporcional-integral-derivativo foi utilizado para ajustar a potência entre os resistores do forno industrial.

No trabalho apresentado por Pawar, Bhirud e Kolhe (2021), os autores apresentam um estudo em que uma rede CAN é utilizada em uma aplicação automotiva. No estudo, os autores utilizam como plataforma de processamento microcontroladores de 8 bits (Atmega, modelo 328P). Uma vez que os microcontroladores escolhidos para o projeto não possuem um periférico para comunicação por protocolo CAN embarcado, os autores propuseram a utilização de um módulo conversor entre os protocolos SPI e CAN, baseado num circuito integrado (Texas Instruments, modelo MCP2515).

Dhivya, Devi e Suguna (2020) apresentam o projeto de um sistema de controle para um farol. O sistema utiliza um barramento CAN para integrar diversos sensores. O modelo proposto é composto por uma unidade de controle de direção, que determina a

direção do farol conforme as mudanças de posição do volante do veículo, permitindo um ajuste numa faixa de $\pm 16^\circ$. O trabalho apresentado por Madhavan et al. (2022) apresenta uma temática semelhante, no qual se realizou o projeto de um sistema para controle automático de faróis de neblina e de um limpador de para-brisas para uso automotivo.

O estudo das redes CAN e suas características motivou o desenvolvimento dos trabalhos apresentados por Bozdal et al. (2020) e Gergeleit e Streich (1994) que avaliaram questões relacionadas à segurança, taxas de transferência de dados e sincronismo de dispositivos conectados. Lokman, Othman e Abu-Bakar (2019) avaliam as vulnerabilidades de uma rede CAN numa aplicação veicular, explorando alguns tipos de ataques que, em tese, poderiam ser objeto de interesse de pessoas mal-intencionadas. Além disso, os autores apresentam algumas possíveis soluções de segurança que poderiam ser aplicadas para minimizar esses efeitos. Ainda nesse sentido Amato et al. (2021), propõe um método baseado em aprendizado profundo (do inglês, *deep learning*), que visa descobrir ataques ao barramento CAN.

Song e Li (2013) comparam diferentes métodos para comunicação de sistema de monitoramento, destacando as vantagens da utilização de um barramento de comunicação CAN para esse tipo de aplicação. No trabalho, uma placa conectada a um computador por meio de um barramento PCI (ZHOU Li-gong, modelo PCI-5110) foi empregada para conectar um computador à rede, permitindo que fosse possível enviar e receber informações para os instrumentos. Um servidor operando com protocolo OPC foi implementado no computador, sendo usado para controlar o processo de monitoramento em tempo real e de envio de informações para uma interface gráfica.

Oksanen, Linkolehto e Seilonen (2016) apresentam um protocolo de automação industrial para monitoramento remoto de máquinas agrícolas móveis a partir de uma rede CAN. O sistema desenvolvido foi aplicado a uma colhedeira, que possuía recursos IoT (do inglês, *Internet of Things*). Os autores apresentam o sistema para o servidor, embarcado na colhedeira, e para o cliente, utilizado para monitoramento remoto das informações. Os resultados incluem métricas relacionadas ao volume de dados transmitido e latências medidas a partir da utilização do sistema.

Os trabalhos analisados mostram que a conexão entre rede CAN e servidor OPC vem sendo utilizada para simplificar o processo de troca de informação entre dispositivos desenvolvidos por diferentes fabricantes tanto em aplicações industriais quanto em aplicações automotivas. Essas informações motivaram o desenvolvimento deste trabalho, cujo foco é desenvolver um sistema voltado para aplicações relacionadas à instrumentação e controle de equipamentos industriais.

1.4 Estrutura do trabalho

Este trabalho foi dividido em cinco capítulos.

No Capítulo 1, apresenta-se inicialmente uma introdução ao tema de estudo deste trabalho, destacando aspectos relacionados ao processo de automação industrial. Na sequência, uma revisão bibliográfica é apresentada, destacando trabalhos focados no estudo de redes CAN e projeto de servidores de protocolo aberto OPC, usados como referência ao longo do desenvolvimento deste projeto.

No Capítulo 2, apresenta-se uma revisão teórica, focada nos conceitos necessários para compreensão do funcionamento e das normas que envolvem as redes CAN, seus protocolos e modos de transmissão, além de uma descrição detalhada do processo de funcionamento dos servidores OPC.

No Capítulo 3, apresenta-se a metodologia empregada neste estudo. São detalhados os procedimentos adotados para a configuração da rede CAN implementada e do servidor OPC, responsável pela sincronização dos dados. Além de descrever a configuração e o diagrama da planta piloto desenvolvida, este capítulo também aborda os instrumentos, sensores e atuadores utilizados para garantir o funcionamento eficaz do sistema.

O Capítulo 4 apresenta os resultados obtidos, incluindo detalhes do sistema desenvolvido, da rede CAN implementada, da operação do servidor OPC, imagens da bancada experimental e análises de formas de onda capturadas por um analisador lógico.

Finalmente, no Capítulo 5 são apresentadas as considerações finais, considerando os resultados obtidos, destacando as propostas para trabalhos futuros.

2 Revisão teórica

Este capítulo aborda as principais características e funcionalidades das redes CAN e do protocolo de comunicação OPC. Inicialmente, alguns conceitos relacionados a organização de redes de comunicação são apresentados. Na sequência, são descritas as características da arquitetura e topologia de rede, bem como as principais normas relacionadas a esse tipo de rede. Finalmente, apresenta-se a plataforma de comunicação aberta OPC, destacando-se as principais características de sua arquitetura e organização.

2.1 Redes CAN

O protocolo utilizado para implementação das redes CAN foi originalmente desenvolvido em fevereiro de 1986 pela companhia alemã Bosh, sendo inicialmente definido como *Automotive Serial Controller Area Network*. Trata-se de um protocolo de rede multi-mestre, baseado em um mecanismo de arbitragem não destrutivo que concede acesso ao barramento à mensagem sem causar atrasos.

O protocolo surgiu com o intuito de reduzir o número de cabos elétricos utilizados para interligar os diversos módulos veiculares que estavam sendo desenvolvidos na época, permitindo que um sistema de controle distribuído pudesse ser implementado (KIM; TRAN-DANG, 2019).

A primeira versão do *hardware* desenvolvido para implementar uma rede CAN foi produzida pela companhia Intel, em meados de 1987. Desde então, diversos fabricantes passaram a desenvolver transceptores (do inglês, *transceivers*) com essa mesma finalidade. Como exemplo, pode-se citar a Phillips Semiconductors e a Infineon Technologies, que desenvolveram versões de circuitos integrados que podem ser utilizados juntamente com os protocolos *Basic CAN* e *Full CAN*. No início da década de 1990, o protocolo CAN começou a ser adotado em diferentes cenários, devido a sua versatilidade e robustez (NATALE et al., 2012).

O protocolo CAN passou por diversas modificações ao longo do tempo e posteriormente incluiu diferentes versões, cada uma com características distintas, que atendem às necessidades de aplicações específicas. Dessa maneira, torna-se interessante destacar algumas versões mais comuns do protocolo CAN.

O protocolo *Classical CAN* representa a versão original do protocolo, que suporta taxas de transferência, dados de até 1 Mbps e tem um formato de quadro de mensagem composto por identificador, dados e bits de controle. A rede *Classical CAN* é amplamente utilizada em diversas aplicações industriais, que vão desde redes veiculares até o controle de dispositivos médicos e automação industrial (LAUFENBERG; KROPF; BRINGMANN, 2023).

Em 2011, a Bosch iniciou o desenvolvimento da versão CAN FD (do inglês, *Flexible Data Rate*) para atender à crescente demanda por taxas de transferência de dados mais altas em aplicações automotivas e industriais. O protocolo CAN FD suporta taxas de transferência de dados de até 8 Mbps e permite a transferência de uma carga de dados maior do que a versão clássica (LAUFENBERG; KROPF; BRINGMANN, 2023).

A versão CAN FD *Light* é uma versão simplificada do protocolo CAN FD. A CAN FD *Light* foi projetada para aplicações que exigem soluções de baixo custo e baixa complexidade, com taxas de transferência de dados de até 1 Mbps e formato de quadro de mensagem semelhante ao da Classical CAN, mas com requisitos de tempo de bit mais curtos (LAUFENBERG; KROPF; BRINGMANN, 2023).

Segundo Laufenberg, Kropf e Bringmann (2023), a versão CAN XL (do inglês, *Extra large*) é a atualização mais recente do protocolo CAN. A CAN XL oferece taxas de transferência de dados ainda mais elevadas do que a CAN FD (até 10 Mbps), e o formato do quadro de mensagem pode acomodar até 2048 B de dados.

Em suma, as diferentes versões do protocolo CAN oferecem recursos e desempenho diferentes, permitindo que os desenvolvedores escolham soluções diferentes em função de suas necessidades específicas para cada aplicação.

2.1.1 Camadas de rede

A camada física do protocolo CAN é responsável pela transmissão e recepção de bits brutos sobre o meio físico de comunicação. Ela define as características elétricas e mecânicas da interface de comunicação, bem como as técnicas de codificação e modulação do sinal para permitir a transmissão em alta velocidade e com baixo nível de erro (NATALE et al., 2012).

Segundo Natale et al. (2012), o meio físico do protocolo CAN é tipicamente um par trançado de fios de cobre, usado para transmitir o sinal diferencial do protocolo. A transmissão diferencial é uma técnica de transmissão que usa dois fios com polaridades opostas para transmitir um sinal. Isso ajuda a reduzir o ruído eletromagnético e a interferência externa no sinal, melhorando a qualidade da comunicação.

Para realizar a transmissão de dados, a camada física do protocolo CAN utiliza a técnica de modulação por amplitude de pulso, PAM, sendo uma técnica de modulação que varia a amplitude do sinal para representar os dados a serem transmitidos. O protocolo CAN utiliza uma técnica de modulação específica NRZ, sendo uma técnica de codificação que mantém o nível de tensão constante durante a transmissão de um bit 1 e muda o nível de tensão durante a transmissão de um bit 0.

Outro aspecto importante do protocolo CAN é a detecção de erros. O protocolo CAN utiliza um mecanismo de detecção de erros por meio da contagem do número de erros detectados na transmissão. Se um número predeterminado de erros for detectado, o protocolo entra em um estado de erro e suspende a transmissão.

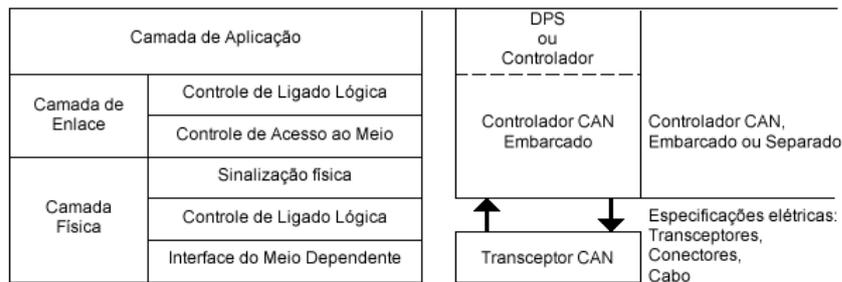
De acordo com Kim e Tran-Dang (2019), as características da camada física do protocolo CAN são descritas pela norma ISO 11898-1. Essa norma estabelece as especificações elétricas, mecânicas e funcionais para a camada física da rede, incluindo a codificação de sinalização, as características do cabo de transmissão, a taxa de transmissão de dados e a configuração dos nós na rede.

A camada física do protocolo CAN é caracterizada por uma topologia em barramento, em que vários nós se conectam a um único cabo de transmissão. Cada nó presente na rede é equipado com um transceptor CAN, que converte os sinais digitais de um dispositivo em sinais diferenciais de tensão para transmissão no barramento, e vice-versa. Além da norma ISO 11898-1, as normas ISO 11898-2 e ISO 11898-3 também apresentam conjuntos de especificações para esse tipo de rede (ISO/TC 22/SC 31, 2015-2016).

A norma ISO 11898: 2003, que descreve como a informação é transmitida entre dispositivos em uma rede, está em conformidade com o modelo de Interconexão de Sistemas Abertos (OSI, do inglês *Open Systems Interconnection*) definido em termos de camadas (INSTRUMENTS, 2016). A comunicação real entre dispositivos conectados pelo meio físico é definida pela camada física do modelo.

A arquitetura ISO 11898 define as duas camadas mais baixas das sete camadas do modelo OSI/ISO como camadas de *link* de dados e camada física, conforme ilustrado na Figura 1.

Figura 1 – A arquitetura padronizada em camadas ISO 11898.



Fonte: Adaptado de Texas Instruments (2016).

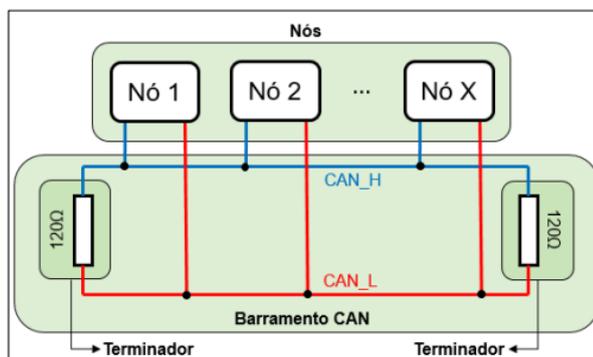
Na camada de aplicação, define-se uma conexão a partir de um protocolo específico para aplicações de nível superior como, por exemplo, o protocolo CANopen™ independentemente do fornecedor. Esse protocolo é suportado pelo grupo internacional de usuários e fabricantes CiA.

2.1.2 Topologia de rede CAN

De acordo com Kim e Tran-Dang (2019), as redes CAN são baseadas em topologia de barramento compartilhado. Os barramentos devem ser terminados em cada extremidade com resistores, com impedância nominal recomendada é de 120 Ω, para suprimir

reflexões de sinal. A Figura 2 apresenta a topologia de um barramento compartilhado utilizado por uma rede CAN.

Figura 2 – Topologia e componentes das redes CAN.



Fonte: Retirado de Farinelli (2016).

Por esse mesmo motivo, os documentos padrão afirmam que a topologia de uma rede CAN deve ser o mais próxima possível de uma linha única. Derivações são permitidas para conectar dispositivos ao barramento, mas seu comprimento deve ser o mais curto possível. Como exemplo, para transmissões de 1 Mbps, o comprimento de uma derivação deve ser inferior a 30 cm (GUIMARÃES; SARAIVA, 2002).

Várias taxas de transferência de dados diferentes podem ser utilizadas, porém, as mais adotadas estão na faixa compreendida entre 50 Kbps a 1 Mbps (esse último valor representa a taxa máxima permitida conforme as especificações da CAN) .

Segundo Guimarães e SARAIVA (2002), a extensão máxima de uma rede CAN depende diretamente da taxa de transferência de dados. A relação exata entre essas duas grandezas envolve parâmetros como os atrasos introduzidos por transceptores e optoacopladores. Em geral, o produto matemático entre o comprimento do barramento e a taxa de transferência de bits por segundo deve ser aproximadamente constante. Como exemplo, a extensão máxima permitida para uma rede de 500 Kbps é de cerca de 100 m, e aumenta para cerca de 500 m quando uma taxa de 125 Kbps é considerada.

Repetidores de sinal podem ser usados para aumentar a extensão da rede, especialmente quando grandes plantas precisam ser cobertas e a taxa de transferência de bits por segundo é baixa. No entanto, eles introduzem atrasos adicionais nos caminhos de comunicação, de modo que a distância máxima entre quaisquer dois nós é efetivamente encurtada quando se utilizam taxas de transferência de bits mais altas. O uso de repetidores também possibilita a utilização de topologias diferentes de barramento, como árvores ou pentes, por exemplo. Nesse caso, um bom projeto poderia aumentar a área efetiva coberta pela rede (KIM; TRAN-DANG, 2019).

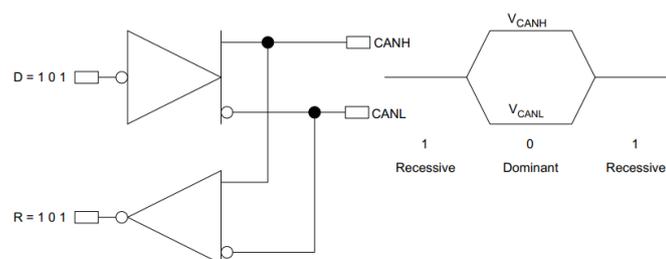
2.1.3 Codificação binária

A interface elétrica entre um nó e o barramento em uma rede CAN é baseada em um esquema semelhante a um coletor aberto (*open-collector-like scheme*). Como consequência, o nível no barramento pode assumir dois valores complementares, que são simbolicamente denotados como dominante e recessivo (KIM; TRAN-DANG, 2019). Geralmente, o nível dominante corresponde ao valor lógico 0, enquanto o nível recessivo coincide com o valor lógico 1.

A rede CAN utiliza a codificação de bits não retorno a zero (NRZ, do inglês *Non-return-to-zero*), que apresenta alta eficiência, uma vez que a informação de sincronização não é codificada separadamente dos dados. A sincronização de bits em cada nó é alcançada por meio de um laço de fase digitalmente controlado (DPLL, do inglês *digital phase-locked loop*), que extrai a informação de temporização diretamente do fluxo de bits recebido do barramento. Em particular, as bordas do sinal são utilizadas para sincronizar os sinais de *clock* locais, de modo a compensar tolerâncias e desvios dos osciladores (KIM; TRAN-DANG, 2019).

Segundo Texas Instruments (2016), uma característica fundamental da rede CAN, apresentada na Figura 3, é a oposição do estado lógico do barramento no que diz respeito a entrada do *driver* e a saída do receptor. Normalmente, um estado lógico alto é associado a um bit 1, e um estado lógico baixo é associado a um bit 0, mas o barramento CAN opera de forma oposta.

Figura 3 – Lógica invertida de um barramento CAN.



Fonte: Retirado de Texas Instruments (2016).

Segundo Texas Instruments (2016), o acesso ao barramento CAN é orientado a eventos e ocorre aleatoriamente. Portanto, no caso de dois nós tentarem ocupar o barramento ao mesmo tempo, a arbitragem não destrutiva bit a bit é empregada para o acesso, ou seja, o nó vencedor no processo de arbitragem será capaz de transmitir a mensagem sem destruí-la ou corrompê-la.

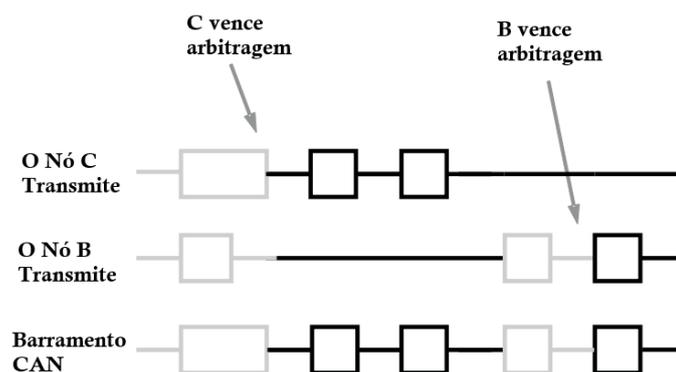
Uma das características que torna o CAN interessante para ser utilizado em um ambiente de controle em tempo real é a atribuição de prioridade às mensagens com base em seu identificador. Sendo que, menor o número do identificador da mensagem em binário, maior sua prioridade. Uma mensagem com um identificador formado inteiramente por zeros é considerada a de maior prioridade em uma rede, pois mantém o barramento

dominante por mais tempo. Portanto, caso dois nós comecem a transmitir simultaneamente, o nó que enviar o último bit de seu identificador como zero (dominante), enquanto os outros nós enviarem um (recessivo), manterá o controle do barramento CAN e continuará completando sua mensagem. Em um barramento CAN, um bit dominante sempre substitui um bit recessivo, como explicado pela Texas Instruments (2016).

Na configuração do transceptor ilustrada na Figura 3, cada bit transmitido por um nó é monitorado constantemente pelo próprio nó transmissor. Os pinos de saída CANH e CANL do driver são conectados internamente à entrada do receptor, como explicado pela Texas Instruments (2016). O atraso de propagação de um sinal no laço interno, que vai da entrada do driver até a saída do receptor, é frequentemente utilizado como uma medida de qualidade para um transceptor CAN. Esse tempo de atraso é comumente chamado de tempo de laço, embora possa ter nomes diferentes conforme o fabricante.

A Figura 4 mostra como um controlador CAN lida automaticamente com o processo de arbitragem CAN. Cada nó monitora constantemente suas próprias transmissões. Quando um bit dominante de prioridade mais alta é transmitido por um nó C e sobrecreve o bit recessivo enviado por um nó B, o nó B detecta que o estado do barramento não corresponde ao que ele transmitiu. Em resposta, o nó B interrompe a transmissão, permitindo que o nó C continue com sua mensagem. Assim que o barramento é liberado pelo nó C, o nó B tentará transmitir novamente a mensagem. Esse recurso faz parte da camada física de sinalização ISO 11898 e é completamente transparente para um usuário CAN, como explicado pela Texas Instruments (2016).

Figura 4 – Arbitragem em um barramento CAN.



Fonte: Adaptado de Texas Instruments (2016).

2.1.4 Formato do quadro de mensagens

Segundo Kim e Tran-Dang (2019), o protocolo CAN suporta dois formatos de quadro de mensagem, sendo a principal diferença entre eles o tamanho do campo de identificação e alguns outros bits no campo de arbitragem. Em particular, o formato de quadro

padrão (do inglês, *standard frame*), também conhecido como formato CAN 2.0A, define um campo de identificação de 11 bits, o que significa que até 2048 identificadores diferentes estão disponíveis para as aplicações executando na mesma rede (muitos controladores CAN mais antigos só suportam identificadores na faixa de 0-2031).

O formato de quadro estendido (do inglês, *Extended frame*), identificado como CAN 2.0B, atribui 29 bits ao identificador, de forma que até meio bilhão de objetos diferentes podem existir (em teoria) na mesma rede. Trata-se de um valor bastante alto, sendo virtualmente suficiente para qualquer tipo de aplicação. Esse segundo formato é opcional, dado que as mensagens com identificadores de 11 bits continuarão a ser mensagens padrão.

O campo de identificação do quadro é usado para identificar a mensagem, e é de importância crítica no processo de arbitragem, que determina qual nó da rede tem prioridade para transmitir a mensagem. No formato de quadro padrão, o campo de identificação é composto por 11 bits, enquanto no formato estendido ele é composto por 29 bits. Além disso, o formato de quadro estendido permite a inclusão de bits de controle adicionais, que podem ser usados para identificar informações importantes, como a prioridade da mensagem.

A diferença entre os dois formatos é determinada pelo bit IDE, definido como dominante para *frames* no formato padrão e recessivo para *frames* no formato estendido. Em vez do bit RTR, o bit SRR é utilizado, com um valor sempre recessivo, para garantir que *frames* padrão tenham prioridade sobre *frames* estendidos com o mesmo identificador base em situações de conflito.

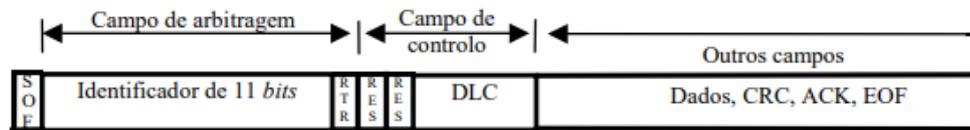
Segundo Texas Instruments (2016), os três bits de controle do protocolo CAN possuem significados distintos. O bit RTR é utilizado para distinguir *frames* de dados e remotas, sendo dominante em *frames* de dados e recessivo em *frames* remotas. O bit SRR é recessivo, sendo utilizado em *frames* estendidas na mesma posição que o bit RTR ocupa em *frames* padrão. Por fim, o bit IDE é utilizado para distinguir *frames* padrão de estendidos, sendo dominante em *frames* padrão e recessivo em *frames* estendidos.

Na Figura 5, são apresentadas diferentes formas de formato que podem ser utilizadas. É possível observar que todos os formatos possuem o bit de início do SOF seguido pelos bits do identificador. Além disso, há um ou três bits de controle (RTR, SRR, e IDE) que determinam se a mensagem é um *frame* padrão ou estendido.

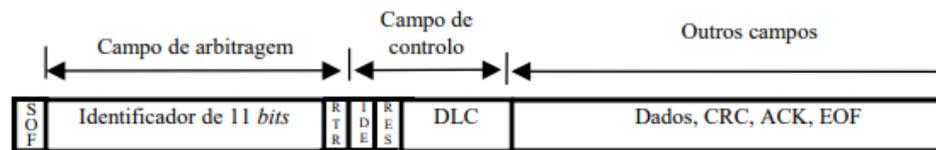
2.1.5 Mecanismos de controle de acesso ao meio

Segundo Kim e Tran-Dang (2019), a técnica de acesso ao meio de transmissão da rede CAN utiliza o método CSMA. Quando nenhum quadro está sendo transmitido, a rede encontra-se ociosa e o nível no barramento é recessivo. Antes de transmitir um quadro, os nós devem observar o estado da rede. Se a rede estiver ociosa, a transmissão do quadro é iniciada imediatamente; caso contrário, o nó deve aguardar a transmissão do quadro atual

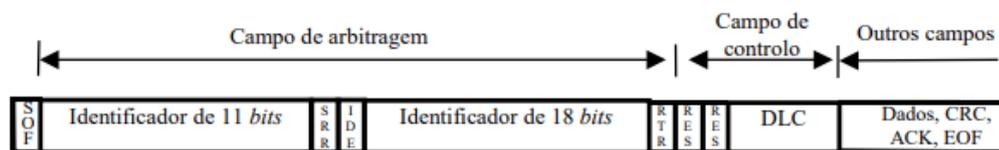
Figura 5 – Formato de quadro de mensagem



Formato do quadro de dados seguindo as versões 1.0, 1.1, 1.2 e 2.0 A.



Formato do quadro de dados seguindo a versão 2.0B (Formato padrão).



Formato do quadro de dados seguindo a versão 2.0B (Formato estendido).

Fonte: Autor.

terminar. Cada quadro começa com o bit SOF em nível dominante, informando a todos os outros nós que a rede mudou para o estado ocupado.

Embora improvável, é possível que dois ou mais nós comecem a enviar seus quadros exatamente simultaneamente. Isso ocorre porque os atrasos de propagação no barramento, embora muito pequenos, permitem que um nó inicie sua transmissão enquanto o bit SOF de outro quadro já está viajando no barramento. Nesse caso, ocorre uma colisão.

Em redes CSMA que se baseiam na detecção de colisão, isso leva inevitavelmente à corrupção de todos os quadros envolvidos, o que significa que eles devem ser retransmitidos. Como consequência, há uma perda de tempo e uma diminuição da largura de banda disponível. Em condições de alta carga, esse efeito pode ser crítico quando o número de colisões é tão alto que a taxa de chegada na rede cai abaixo da taxa de transferência, tornando a rede estagnada (KIM; TRAN-DANG, 2019).

2.1.6 Gerenciamento de erros

Segundo Kim e Tran-Dang (2019), um dos principais requisitos para definição do protocolo CAN é a necessidade de ter um sistema de comunicação caracterizado por alta robustez, ou seja, um sistema capaz de detectar a maioria dos erros de transmissão. Por isso, foi dada uma atenção especial à definição do gerenciamento de erros.

Na verificação de redundância cíclica, CRC, ao transmitir um quadro, o nó de origem adiciona um CRC de 15 bits ao final do próprio quadro. Os nós receptores reavaliam

o CRC para verificar se ele coincide com o transmitido. Em geral, o CRC usado no protocolo CAN pode ser usado para identificar até 5 bits errados distribuídos arbitrariamente no quadro ou erros em rajadas que incluam até 15 bits.

No processo de verificação do quadro, os campos de formato fixo nos quadros recebidos podem ser facilmente testados em relação aos seus valores esperados. Por exemplo, o CRC e os delimitadores ACK, bem como o campo EOF, devem estar no nível recessivo. Se um ou mais bits ilegais forem detectados, um erro de forma é gerado.

Na etapa de verificação de reconhecimento, o nó transmissor verifica se o bit ACK foi definido como o valor dominante no quadro recebido. Caso contrário, é emitido um erro de reconhecimento.

No processo de monitoramento de bit, cada nó transmissor compara o nível no barramento em relação ao valor do bit que está sendo escrito. Quando ocorre uma incompatibilidade, é gerado um erro. Isso não se aplica ao campo de arbitragem ou ao slot de reconhecimento. Essa verificação de erro é muito eficaz para detectar erros locais que podem ocorrer nos nós transmissores.

Finalmente, no processo de inserção de bits (do inglês, *stuffing*), cada nó verifica se as regras de *stuffing* de bits foram violadas na porção dos quadros do bit SOF até a sequência CRC. No caso de ler seis bits de valor idêntico no barramento, é gerado um erro.

Kim e Tran-Dang (2019) destacam que o uso desses mecanismos de detecção de erros contribui significativamente para a alta confiabilidade e robustez do protocolo CAN. Quando um erro é detectado, o protocolo prevê um procedimento de recuperação de erro.

A correção de erros no protocolo CAN é realizada pela retransmissão de quadros de dados ou pela retransmissão de quadros de erros ativos. A recuperação de erros também pode ser feita por meio da exclusão de nós defeituosos da rede ou pela reconfiguração da rede em caso de falhas críticas.

2.1.7 Arquitetura de controladores CAN

Segundo Kim e Tran-Dang (2019), é possível classificar os controladores CAN em duas categorias: Basic CAN e Full CAN. Os controladores Basic CAN possuem um *buffer* de transmissão e um de recepção, semelhante aos UART convencionais.

A função de filtragem de quadros, nesse caso, é geralmente deixada para os programas de aplicação, ou seja, está sob controle do controlador *host*, embora algum tipo de filtragem possa ser feito pelo controlador.

Para evitar condições de sobrescrita, um esquema de duplo *buffering*, baseado em *buffers* de recepção sombreados, geralmente está disponível, o que permite que um novo quadro seja recebido do barramento enquanto o anterior está sendo lido pelo controlador *host*.

Por outro lado, os controladores Full CAN preveem vários *buffers* internos que podem ser configurados para receber ou transmitir mensagens específicas. Nesse caso, a função de filtragem é implementada diretamente no controlador CAN.

Quando um novo quadro de interesse para o nó é recebido pela rede, ele é armazenado no *buffer* relacionado, onde pode ser lido pelo controlador *host*. Em geral, os novos valores simplesmente sobrescrevem os anteriores.

2.1.8 Desempenho

O desempenho da rede CAN está intimamente relacionado com a escolha da taxa de transmissão de bits por segundo e com o comprimento máximo permitido para o barramento (KIM; TRAN-DANG, 2019). Isso ocorre porque, como a posição de amostragem é localizada aproximadamente após a metade de cada bit, o atraso de propagação de ponta a ponta deve ser menor que cerca de um quarto do tempo do bit.

Além disso, a velocidade de propagação dos sinais é fixa, o que implica que o comprimento máximo permitido para o barramento é necessariamente limitado e depende diretamente da taxa de bits escolhida para a rede.

De acordo com Kim e Tran-Dang (2019), o protocolo CAN é muito mais simples e robusta do que os esquemas de acesso baseados em *token*, como o PROFIBUS (do inglês, *Process Field Bus*), por exemplo, quando usados em configurações *multi-master*. Não há necessidade de construir ou manter o anel lógico, ou gerenciar a circulação do *token* em torno das estações mestre.

Além disso, esse tipo de rede é mais flexível do que as soluções baseadas em acesso múltiplo por divisão de tempo (TDMA) ou mensagens combinadas, duas técnicas adotadas por SERCOS (do inglês, *Serial Real-time Communication System*) e INTERBUS (do inglês, *Interconnected Binary Universal System*), respectivamente. Isso ocorre porque as trocas de mensagens não precisam ser conhecidas com antecedência.

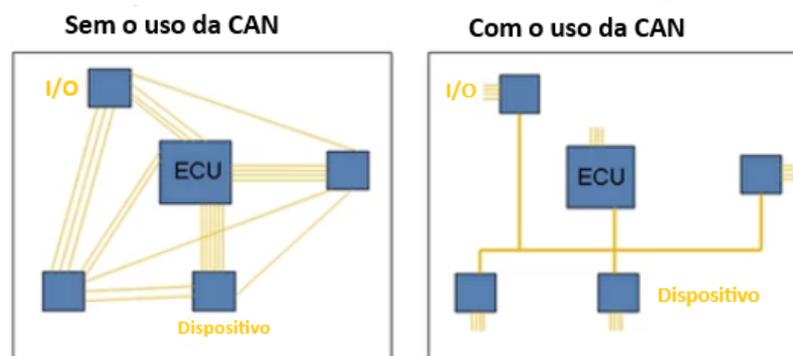
Quando comparada aos esquemas baseados em *polling* centralizado, como o FIP (do inglês, *Factory Instrumentation Protocol*), não é necessário ter um nó na rede que atue como árbitro do barramento, o que pode se tornar um ponto de falha para todo o sistema. Uma vez que todos os nós são mestres na CAN (pelo menos do ponto de vista do mecanismo MAC, do inglês, "*Media Access Control*"), notificar eventos assíncronos torna-se uma tarefa mais simples. Em todos os casos em que esse aspecto é importante, o protocolo CAN é claramente superior às soluções acima citadas.

Com o esquema de arbitragem, é certo que nenhuma mensagem será atrasada por trocas de menor prioridade. Como o protocolo CAN não é preemptivo (como quase todos os protocolos existentes), uma mensagem ainda pode ser atrasada por uma de menor prioridade cuja transmissão já tenha começado, algo inevitável em qualquer sistema não preemptivo.

No entanto, como o tamanho do quadro no protocolo CAN é muito pequeno (os quadros padrão têm no máximo 135 bits de comprimento), o tempo de bloqueio experimentado pelas mensagens urgentes é geralmente bastante baixo. Isso faz da CAN uma rede de resposta rápida, o que explica por que ela é usada em muitas aplicações de controle em tempo real, apesar de sua largura de banda relativamente baixa.

A Figura 6 apresenta um exemplo de como a utilização de redes CAN pode reduzir significativamente a quantidade de fios necessários em um sistema de controle. Na imagem, apresenta-se um sistema de controle com vários componentes interconectados por fios individuais, resultando em um excesso de fios. Em contrapartida, o sistema de controle que utiliza redes CAN apresenta uma quantidade muito menor de fios, já que os componentes estão conectados a uma rede comum usando apenas alguns fios de comunicação. Isso simplifica significativamente a instalação e manutenção do sistema, além de reduzir custos de materiais e mão de obra.

Figura 6 – Redes CAN reduzem significativamente a fiação.



Fonte: Adaptado de Texas Instruments (2016).

2.1.9 Normas

Para assegurar que as comunicações na rede CAN sejam interoperáveis e confiáveis, é necessário seguir normas e especificações técnicas. Organizações como a Organização Internacional de Padrões (ISO) e a CiA (do inglês, *CAN in Automation*) são responsáveis por definir as principais normas relacionadas à rede CAN.

A norma ISO 11898-1 estabelece as características físicas e elétricas da rede, incluindo as especificações para cabos, conectores, resistores de terminação e transceptores. A norma ISO 11898-2 define o formato dos quadros de dados transmitidos na rede CAN, incluindo a mensagem, o endereço de identificação e o comprimento do quadro. Já a norma ISO 11898-3 define os requisitos de temporização da rede, incluindo os parâmetros de tempo para a transmissão e recepção de mensagens (ISO/TC 22/SC 31, 2015-2016).

Além dessas normas, a ISO 11898-4 define as especificações para a transmissão de dados sem fio na rede CAN. A organização CiA também desenvolveu diversas especificações técnicas para atender a necessidades específicas de aplicações na rede CAN. Por exemplo, a especificação CiA 301 define as funções básicas de comunicação e protocolos para aplicações da rede CAN, enquanto a CiA 402 é específica para o controle de motores e define protocolos adicionais para essa aplicação.

Outra norma relevante é a SAE J1939, desenvolvida pela *Society of Automotive Engineers* (SAE) para aplicações da rede CAN em veículos comerciais e de serviço pesado. Essa norma define mensagens e protocolos específicos para aplicações veiculares, incluindo ônibus, caminhões, tratores e outros veículos pesados. Além disso, a indústria padronizou outros protocolos baseados em CAN, como a DeviceNet™ da Rockwell Automation e o CAN Kingdom da KVASER.

2.2 Plataforma de comunicação aberta OPC

A Arquitetura Unificada OPC foi desenvolvida por mais de 30 empresas ao longo de aproximadamente 5 anos. Implementações de referência completas foram desenvolvidas para validar as especificações e provar a viabilidade técnica.

O objetivo da Arquitetura Unificada OPC é garantir um padrão de interoperabilidade independente de plataforma para mover dados/informações entre o chão de fábrica e a empresa. A plataforma OPC foi expandida para áreas da automação predial, segurança, automação residencial, geração de energia e indústria petroquímica. Por se tratar de uma arquitetura altamente escalável, essa plataforma também vem sendo utilizada em aplicações embarcadas e desenvolvimento de sistemas inteligentes.

A plataforma OPC UA é resultado de um esforço colaborativo entre diversas organizações interessadas na padronização de um sistema de gerenciamento de dados, e foi construído sob a premissa de não reinventar tecnologias já existentes (MAHNKE; LEITNER; DAMM, 2009).

Assim como a questão do acesso aos dados de impressoras em sistemas operacionais antigos, no qual cada fabricante deve escrever seus próprios *drivers* para cada sistema operacional disponível no mercado, havia um problema para acesso a informações de instrumentos de medição e atuadores conectados a uma rede.

A solução encontrada para o problema de *drivers* de impressoras foi incorporar ao sistema operacional *drivers* desenvolvidos por diferentes fabricantes, tornando o processo de instalação desses dispositivos uma tarefa mais simples.

Uma iniciativa semelhante foi tomada pelas empresas Fisher-Rosemount, Rockwell Software, Opto 22, Intellution e Intuitive Technology em 1995, que formaram uma força-tarefa para definir um padrão Plug&Play para *drivers* de dispositivos utilizados na automação de sistemas (MAHNKE; LEITNER; DAMM, 2009). O resultado do trabalho

dessa força-tarefa resultou na especificação dos padrões da plataforma OPC Data Access, lançada em agosto de 1996.

A OPC Foundation, uma organização sem fins lucrativos, foi criada para manter esse padrão. Grande parte dos fornecedores de sistemas para automação industrial tornaram-se membros desta fundação.

A Figura 7 representa a demografia dos membros da OPC Foundation, classificados por classe de associação e região. No gráfico à esquerda, as classes de associação são representadas em um gráfico de setores que evidencia a proporção de membros com vendas anuais inferiores a 2 milhões de dólares, entre 2 e 20 milhões, entre 20 e 100 milhões e acima de 100 milhões. O gráfico à direita mostra a distribuição geográfica dos membros, com uma presença significativa na Europa (43%), seguida pela América do Norte (39%) e Ásia (13%), com os 5% restantes espalhados por outras regiões. É notável também a inclusão de usuários finais e membros não votantes na composição da fundação. Além disso, a governança da OPC Foundation é assegurada por um conselho de diretores eleito pelos membros, que tem entre suas funções nomear o restante do corpo técnico e gestor (MAHNKE; LEITNER; DAMM, 2009).

Figura 7 – Demografia de membros da fundação OPC.



Fonte: Adaptado de Mahnke, Leitner e Damm (2009).

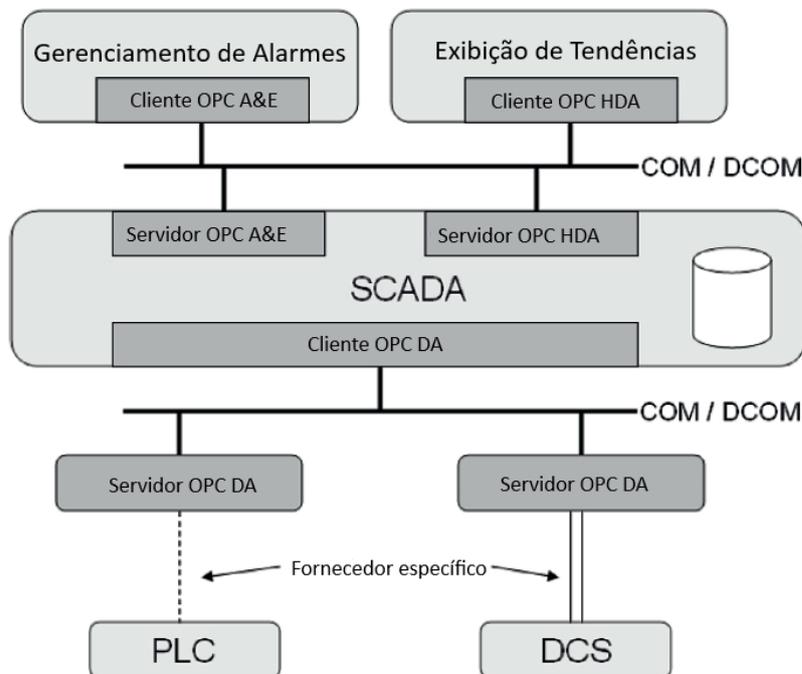
2.2.1 Configuração clássica

Segundo Mahnke, Leitner e Damm (2009), três principais especificações foram definidas para a plataforma OPC: acesso a dados, DA, alarme e eventos, A&E, e acesso a dados históricos, HDA. O acesso aos dados de processo atuais é descrito na especificação DA, A&E descreve uma interface para informações baseadas em eventos, incluindo reconhecimento de alarmes de processo, e HDA descreve funções para acessar dados arquivados. Todas as interfaces oferecem uma maneira de navegar pelo espaço de endereço e fornecer informações sobre os dados disponíveis.

A plataforma OPC usa uma abordagem cliente-servidor para a troca de informações. Um servidor OPC encapsula a fonte de informações do processo, como um dispositivo, e disponibiliza as informações por meio de sua interface. Um cliente OPC se conecta ao servidor OPC, podendo acessar e consumir os dados oferecidos. As aplicações que consomem e fornecem dados podem ser geradas tanto por cliente quanto por servidor.

A Figura 8 apresenta um caso típico de uso de clientes e servidores OPC (MAHNKE; LEITNER; DAMM, 2009).

Figura 8 – Caso típico de uso de clientes e servidores OPC.



Fonte: Adaptado de Mahnke, Leitner e Damm (2009).

As interfaces Classic OPC são baseadas na tecnologia COM e DCOM da Microsoft. A vantagem dessa abordagem foi a redução do trabalho de especificação para a definição de diferentes APIs para necessidades especializadas diferentes, sem a necessidade de definir um protocolo de rede ou um mecanismo de comunicação inter-processo.

As tecnologias COM e DCOM fornecem um mecanismo transparente para um cliente poder chamar métodos em um objeto COM usando um servidor em execução no mesmo processo, em outro processo ou em outro nó de rede. O uso dessa tecnologia, disponível em todos os sistemas operacionais Windows baseados em PC, reduziu o tempo de desenvolvimento das especificações e produtos e o tempo de colocação no mercado para OPC. Esse fator foi crucial para o sucesso da plataforma OPC.

As duas principais desvantagens dessa plataforma são a dependência da plataforma Windows e os problemas do DCOM ao usar a comunicação remota com OPC. Ressalta-se que a ferramenta DCOM apresenta tempo limite elevado e não configurável, e não pode ser usada para comunicação pela internet.

2.2.2 Arquitetura unificada

Segundo Mahnke, Leitner e Damm (2009), a arquitetura unificada OPC UA surgiu visando encontrar um substituto para todas as especificações baseadas em COM já

existentes, sem perder recursos ou desempenho. Ademais, a nova plataforma tem como objetivo atender a todos os requisitos para interfaces de sistema independentes de plataforma, oferecendo recursos de modelagem ricos e extensíveis que possam descrever sistemas complexos. O padrão aberto OPC UA suporta mecanismos estendidos de segurança e pode ser combinado de forma otimizada com redes como PROFINET ou Ethernet/IP em uma rede Ethernet Industrial compartilhada.

A Arquitetura Unificada OPC UA é um padrão aberto para comunicações horizontais entre máquinas (M2M) e para comunicações verticais de máquinas para a nuvem, que surgiu para substituir todas as especificações baseadas em COM existentes sem perder recursos ou desempenho, cobrindo todos os requisitos para interfaces de sistema independentes de plataforma com recursos de modelagem extensíveis capazes de descrever também sistemas complexos (MAHNKE; LEITNER; DAMM, 2009).

Os componentes fundamentais da plataforma OPC UA são os mecanismos de transporte e a modelagem de dados. O transporte define diferentes mecanismos otimizados para diferentes casos de uso. A primeira versão do OPC UA define um protocolo binário TCP otimizado para comunicação de alto desempenho em intranets, bem como um mapeamento para padrões da Internet, como serviços Web, XML e HTTP, para comunicação amigável ao *firewall* da Internet. Ambos os mecanismos de transporte usam o mesmo modelo de segurança baseado em mensagens conhecido nos *Web Services* (MAHNKE; LEITNER; DAMM, 2009).

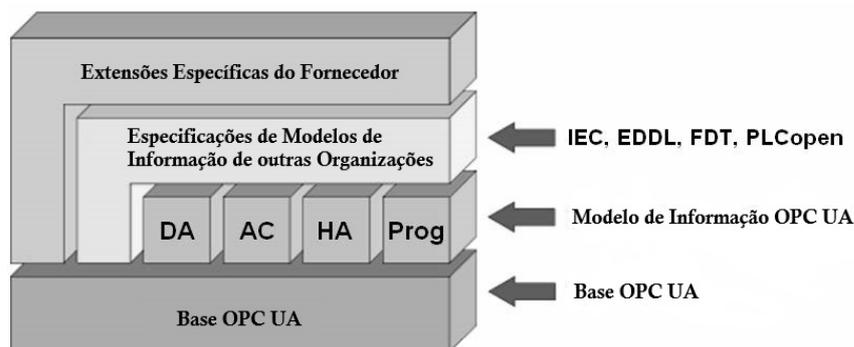
Segundo Mahnke, Leitner e Damm (2009), a modelagem de dados define as regras e blocos de construção básicos necessários para expor um modelo de informações com OPC UA. Também define os pontos de entrada no espaço de endereços e os tipos básicos usados para construir uma hierarquia de tipos. Essa base pode ser estendida por modelos de informações construídos em cima dos conceitos abstratos de modelagem. Além disso, define alguns conceitos aprimorados, como descrever máquinas de estado usadas em diferentes modelos de informações.

Os serviços UA são a interface entre servidores como fornecedores de um modelo de informações e clientes como consumidores desse modelo de informações. Os serviços são definidos de maneira abstrata e usam mecanismos de transporte para trocar dados entre cliente e servidor (MAHNKE; LEITNER; DAMM, 2009).

Segundo Mahnke, Leitner e Damm (2009), esse conceito básico de OPC UA permite que um cliente OPC UA acesse as menores peças de dados sem a necessidade de entender todo o modelo exposto por sistemas complexos. Por outro lado, os clientes OPC UA podem usar recursos mais avançados definidos para domínios e casos de uso especiais. A Figura 9 mostra as diferentes camadas de modelos de informação definidos pela fundação OPC, por outras organizações ou por fornecedores.

Para cobrir todas as características bem-sucedidas conhecidas do OPC Clássico, modelos de informações para o domínio de informações de processo são definidos pela

Figura 9 – Arquitetura em camadas do OPC UA.



Fonte: Adaptado de Mahnke, Leitner e Damm (2009).

OPC UA em cima das especificações básicas.

O termo DA define extensões específicas de automação de dados, como a modelagem de dados analógicos ou discretos e como expor a qualidade de serviço. Todas as outras características do DA já são cobertas pela base. *Alarm & Conditions* (AC) especifica um modelo avançado para gerenciamento de alarmes de processo e monitoramento de condições. *Historical Access* (HA) define os mecanismos para acessar dados históricos e eventos históricos. *Programs* (Prog) especifica um mecanismo para iniciar, manipular e monitorar a execução de programas.

Outras organizações podem construir seus modelos em cima da base UA ou do modelo de informações OPC, expondo suas informações específicas via OPC UA. Exemplos de padrões que já trabalham em mapeamentos para OPC UA incluem a integração de dispositivos de campo (FDI), que combina a Linguagem de Descrição de Dispositivos Eletrônicos (EDDL) e a Ferramenta de Dispositivos de Campo (FDT), ambas usadas para descrever, configurar e monitorar dispositivos, além do padrão PLC open, para linguagens de programação de CLP. Ademais, Modelos de informação específicos do fornecedor adicionais serão definidos usando diretamente a base UA, os modelos OPC ou outros modelos de informação baseados em OPC UA.

2.3 Considerações parciais

Este capítulo apresentou uma revisão teórica envolvendo os principais conceitos necessários para o desenvolvimento da metodologia proposta neste trabalho. Inicialmente, foram apresentados os fundamentos da Rede CAN, sendo uma das tecnologias mais utilizadas em sistemas de controle e automação industrial. Destacam-se como principais características da rede CAN a alta velocidade de comunicação, robustez, imunidade a ruídos e facilidade de implementação. Além disso, foram apresentadas as diferenças entre as arquiteturas Basic CAN e Full CAN, suas vantagens e especificações técnicas.

Em seguida, foram descritos os padrões definidos pela plataforma OPC UA, utili-

zada para garantir o processo de comunicação entre sistemas heterogêneos. Foram descritos os princípios de funcionamento e principais características da plataforma, e apresentados alguns exemplos de aplicações.

3 Metodologia

Neste capítulo, apresenta-se a metodologia utilizada para configurar uma rede CAN para conectar microcontroladores Atmega 328p, com auxílio de um módulo conversor composto pelos CIs MCP2515 e TJA1050. São apresentadas as bibliotecas empregadas para implementar a rede de comunicação, bem como um diagrama esquemático apresentando as conexões físicas que devem ser realizadas antes do início dos ensaios, com o intuito de estabelecer a comunicação, que serão aplicadas ao sistema de tanques interconectados.

Na sequência, são descritas as etapas de configuração do servidor OPC utilizado neste trabalho, bem como os métodos empregados para enviar e receber informações aos módulos embarcados por meio do servidor, usando como IHM o *software* Matlab.

3.1 Rede CAN

Existem diversos dispositivos microcontrolados que podem ser utilizados para a construção de uma rede CAN, mas neste projeto, optou-se por utilizar o microcontrolador Atmega 328p. Essa escolha foi motivada exclusivamente devido à facilidade de programação, haja vista que esse dispositivo pode ser programado por meio da plataforma Arduino IDE, e pelo custo.

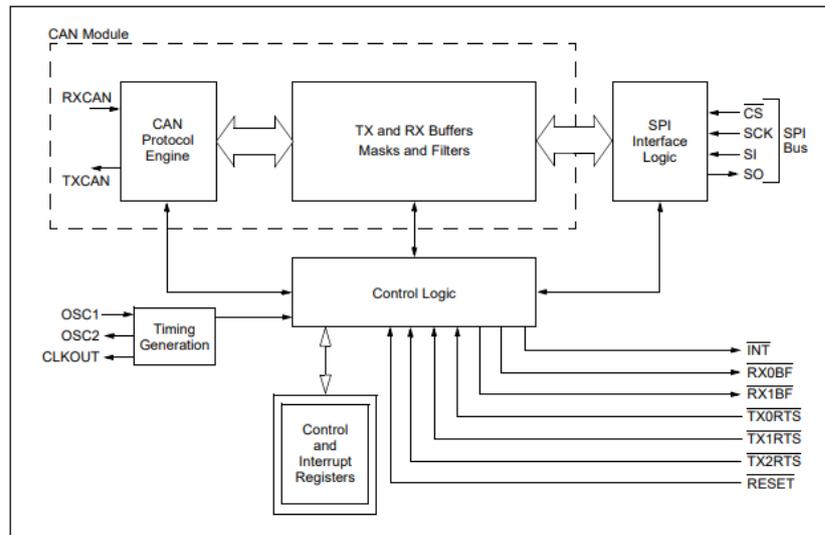
Por ser um dispositivo programável a partir da plataforma Arduino, que é uma plataforma de código aberto, há uma abundância de recursos disponíveis, como bibliotecas e exemplos de projetos, disponíveis gratuitamente para utilização. No entanto, há outras opções de microcontroladores, como o ESP32 da fabricante chinesa Espressif ou o SMT32F1x da fabricante ST Electronics, que apresentam maior desempenho e podem ser considerados, dependendo das especificações do projeto.

O Atmega 328p não possui um módulo CAN encapsulado dentro de seu circuito integrado. Dessa forma, foi necessário utilizar um módulo complementar (*shield*) para converter sinais transmitidos e recebidos através do protocolo SPI para a rede CAN. O módulo utilizado neste trabalho utiliza um CI para realizar a conversão entre os protocolos SPI e CAN (Microchip, modelo MCP2515) e um *transceiver* para ajustar os níveis de tensão de saída conforme o padrão CAN (Nxp Semiconductors, modelo TJA1050).

O módulo MCP2515 é um controlador responsável por gerenciar o processo de troca de informações entre os dispositivos conectados à rede, fornecendo um meio de transmissão confiável e seguro.

De acordo com Technology (2012), o MCP2515 é um controlador CAN independente, desenvolvido para simplificar aplicações que exijam uma interface entre um módulo de comunicação SPI e um barramento CAN. Um diagrama de blocos simplificado do MCP2515 é apresentado na Figura 10.

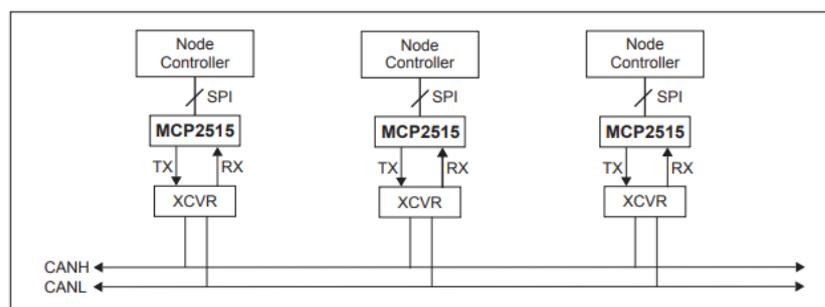
Figura 10 – Diagrama de blocos.



Fonte: Retirado de Technology (2012).

O dispositivo é composto por um bloco de comunicação SPI, um circuito lógico para controle, cujo funcionamento deve ser configurado a partir da definição de parâmetros ajustados por meio de registradores de uso específico, e um módulo CAN, responsável por implementar a rede de comunicação CAN básico, incluindo filtros e *buffers* de transmissão e recepção. Um exemplo de implementação do sistema usando este dispositivo é apresentado na Figura 11.

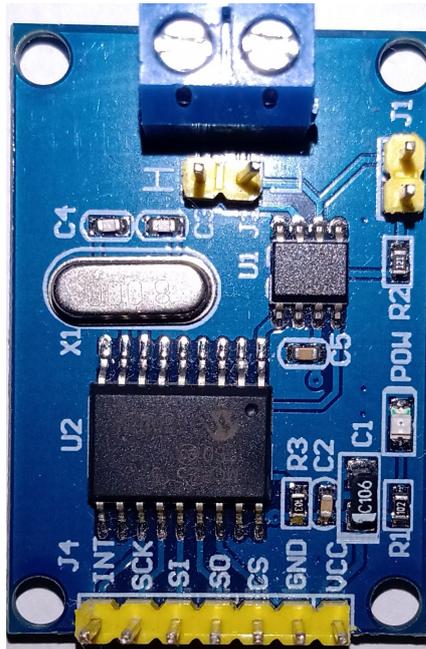
Figura 11 – Exemplo da implementação do sistema.



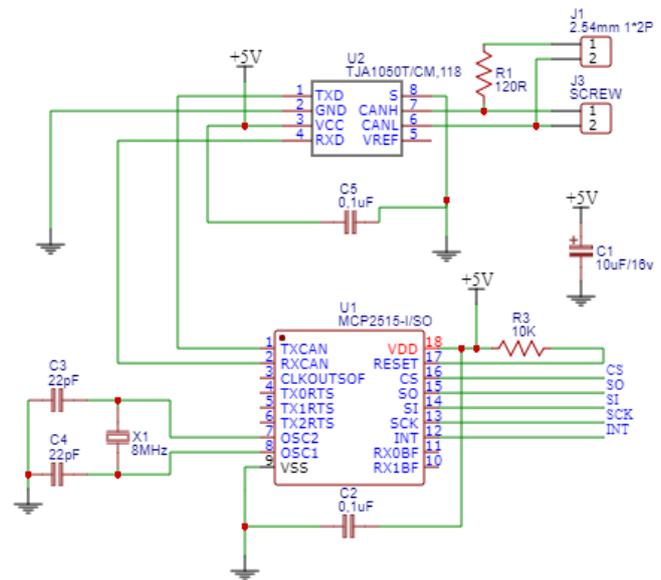
Fonte: Retirado de Technology (2012).

No módulo de comunicação CAN aplicado neste trabalho, o CI MCP2515 opera em conjunto com o *transceiver* TJA1050, um transceptor de alta velocidade. O módulo suporta velocidades de operação de até 1 Mbps, e pode ser configurado para operar em diferentes modos de operação, como normal, *loopback* e escuta. De acordo com Semiconductors (2007), O CI TJA1050 pode operar com taxas de transmissão de 60 kbps até 1 Mbps, com capacidade de transmissão e recepção diferencial. Além disso, esse CI é compatível com o padrão ISO-11898. A Figura 12 apresenta uma foto do módulo usado e o diagrama esquemático do circuito da placa.

Figura 12 – Módulo CAN MCP2515 - TJA1050.



(a) Módulo MCP2515 TJA1050.



(b) Diagrama esquemático do módulo.

Fonte: Autor.

3.1.1 Conexão elétrica entre os módulos

Após a seleção dos componentes a serem utilizados, montou-se uma rede CAN considerando inicialmente apenas dois nós. Nesse momento, realizou-se um teste inicial para verificar se os microcontroladores eram capazes de trocar informações entre si. A Tabela 1 apresenta uma lista com a relação dos pinos conectados para realizar este ensaio. Adicionalmente, um diagrama esquemático e uma foto do circuito empregado nos ensaios pode ser vistos na Figura 13.

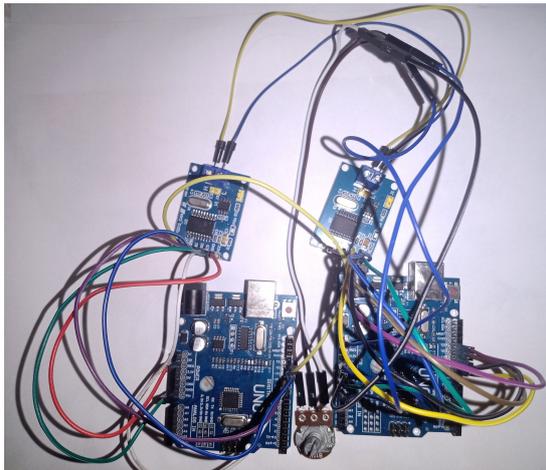
Tabela 1 – Conexões entre microcontrolador Atmega 328p e módulo MCP2515.

MCP2515	Atmega 328p
VCC	5 V
GND	GND
CS	D10
SO	D12
SI	D11
SCK	D13
INT	D2

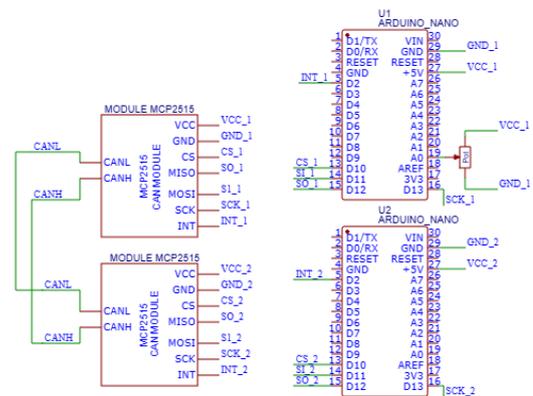
Fonte: Autor.

Um potenciômetro foi conectado a uma entrada analógica (A0) de um dos microcontroladores. Um sinal de tensão de 5 V foi utilizado para alimentar o potenciômetro, e os valores de tensão analógicos foram convertidos em valores digitais, com resolução

Figura 13 – Circuito utilizado ao longo dos ensaios propostos neste trabalho.



(a) Protótipo do circuito para teste.



(b) Diagrama esquemático do circuito.

Fonte: Autor.

de 10 bits. Finalmente, o sinal digital amostrado foi transmitido pela rede CAN para o segundo microcontrolador.

A conexão entre os dois módulos MCP2515 foi feita utilizando um par de condutores trançados. Resistores de terminação de 120Ω foram habilitados para garantir um casamento de impedâncias adequado.

A biblioteca Autowp, desenvolvida para controlar o módulo MCP2515 a partir da IDE Arduino, foi utilizada neste trabalho (AUTOWP, 2023). Essa biblioteca foi adicionada à IDE por meio do gerenciador de bibliotecas. Como já mencionado, à vantagem de se utilizar essa biblioteca está relacionada a sua facilidade de implementação e utilização, reduzindo significativamente os tempos de desenvolvimento de *software*.

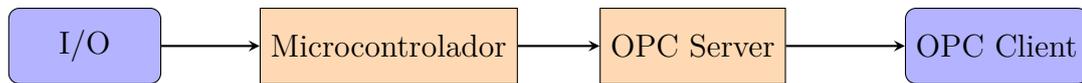
Nos testes realizados, um dos microcontroladores foi configurado para operar como emissor de dados, e o outro dispositivo atuou como receptor. O emissor foi programado para transmitir os valores digitais de tensão adquiridos ao monitorar o terminal central do potenciômetro, e o receptor foi ajustado para enviar os dados recebidos pela rede CAN para um computador por meio do protocolo de comunicação RS-232. Um *software* monitor serial foi empregado para exibir os resultados de forma gráfica na tela do computador.

Dessa forma, o experimento envolve a utilização de dois microcontroladores, sendo um emissor de dados e o outro receptor, para realizar a transmissão de valores de tensão obtidos por meio de um potenciômetro. Para isso, foi escolhido o ID CAN 0x037 e configurada uma taxa de transmissão de dados em 125 kbps, utilizando 11 bits para o ID e 2 bytes para enviar a palavra de 10 bits adquirida pelo ADC do microcontrolador. Além disso, as mensagens foram atualizadas a cada 100ms.

3.2 Servidor OPC

A Figura 14 apresenta um diagrama de blocos que ilustra a maneira como o servidor OPC utilizado neste trabalho foi estruturado.

Figura 14 – Diagrama de blocos da configuração do servidor OPC.



Fonte: Autor.

Inicialmente, deve-se configurar o servidor OPC para que ele possa ser utilizado para controlar a troca de informações entre módulos microcontrolados e computador.

Para que o servidor OPC funcione em conjunto com a plataforma Arduino, é necessário instalar um *software* adicional no computador. Além disso, é necessário importar a biblioteca OPC na Arduino IDE e carregar um algoritmo no microcontrolador que será usado como servidor. O *software* instalado no computador é responsável por gerenciar a comunicação entre os clientes OPC e microcontrolador, e a biblioteca OPC para o Arduino permite que o Atmega 328p seja utilizado em conjunto com um servidor OPC.

Existem diversas opções de *softwares* para implementação de servidores OPC disponíveis no mercado, mas, neste trabalho, optou-se por utilizar o Arduino OPC Server (MAKERS, 2023). Trata-se de uma solução simples, capaz de integrar microcontroladores a um servidor OPC instalado num computador com sistema operacional Windows.

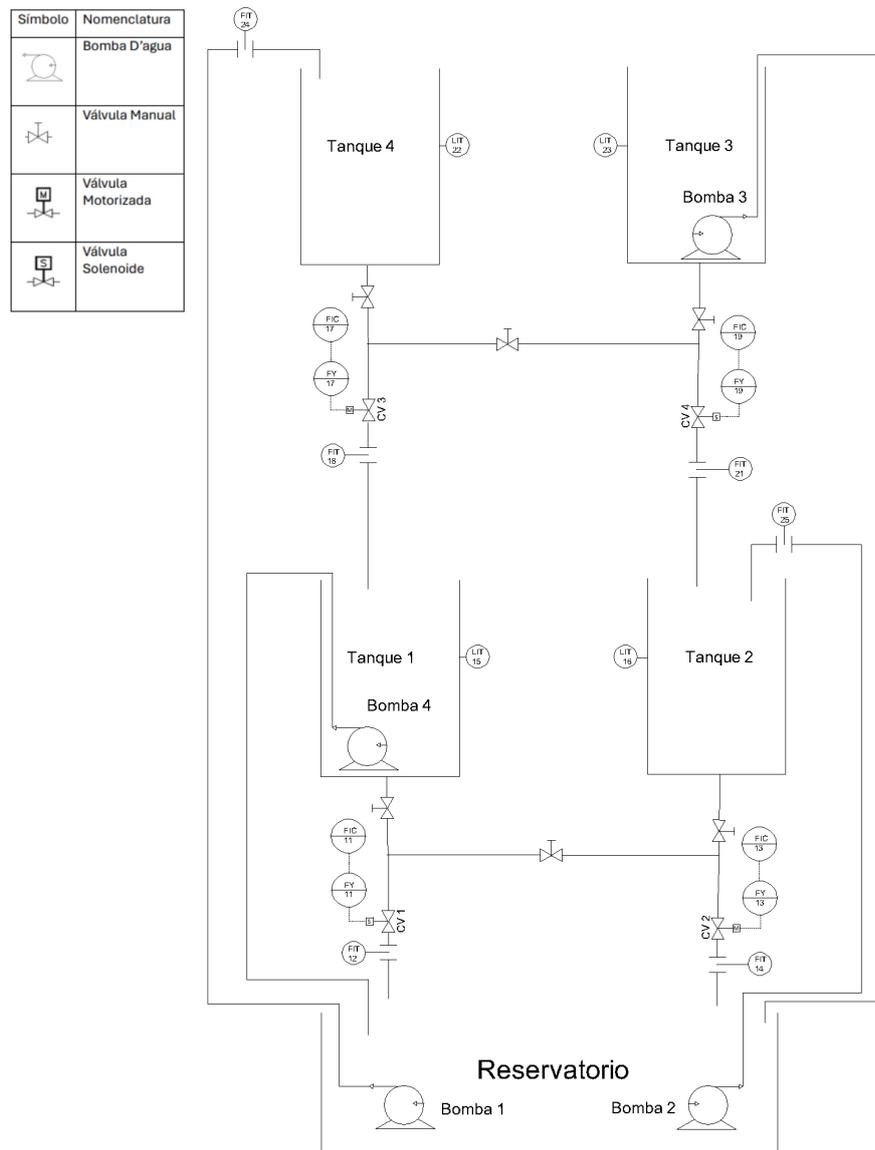
O servidor OPC instalado no computador pode ser acessado por outros *softwares* clientes de diversas formas. Neste trabalho, esse acesso aos dados foi realizado a partir do *software* MATLAB, por meio do OPC *Toolbox*, o qual oferece um conjunto de ferramentas para interagir com servidor e clientes OPC, permitindo a leitura e escrita de dados em tempo real.

Através da conexão com o servidor OPC, pode-se acessar os dados armazenados no servidor OPC, gerenciado por um dos microcontroladores Atmega 328p, e, com isso, enviar e receber informações aos demais nós por meio da rede CAN. Com isso, torna-se possível realizar o monitoramento em tempo real das variáveis de interesse, além de controlar o funcionamento do sistema por meio do envio de comandos específicos de controle.

3.3 Diagrama de instrumentação e tubulação

Após a implementação e verificação da rede CAN, o próximo passo foi integrar o sistema de controle com o arranjo físico dos tanques interconectados, conforme ilustrado na Figura 15 e no Apêndice A, em maior escala. Este arranjo consiste em quatro tanques, interligados por tubulações e válvulas controladas, visando simular um ambiente de processo industrial dinâmico.

Figura 15 – Diagrama de instrumentação e tubulação implementado.



Fonte: Autor.

Os tanques possuem dimensões de 10 cm x 15 cm x 20 cm cada, complementados por um reservatório maior com dimensões de 40 cm x 20 cm x 30 cm. Cada tanque é equipado com sensores de nível LT, sensores de vazão FIT, e atuadores (válvulas motorizadas controladas por sinais de FY), gerenciados por microcontroladores Atmega 328p. A comunicação entre os microcontroladores e o sistema central é realizada via rede CAN, permitindo a transmissão em tempo real dos dados de sensoriamento e o recebimento de comandos de controle.

Os sensores de vazão e os transmissores de nível LT são utilizados para monitorar constantemente as condições dos tanques. Os dados são enviados aos microcontroladores, que, por sua vez, ajustam as válvulas motorizadas para manter a vazão e o nível desejados dentro dos parâmetros estabelecidos.

Além disso, as válvulas manuais são posicionadas estrategicamente ao longo das tubulações para permitir ajustes físicos na vazão, introduzindo um aspecto de controle manual no sistema predominantemente automatizado, de forma que, essas válvulas ofereçam meios de modificar a configuração do sistema.

A integração com a rede CAN é essencial para a orquestração do sistema. As informações de nível e vazão são transmitidas pela rede, utilizando as funcionalidades do módulo MCP2515, e recebidas pelo servidor OPC para monitoramento e registro. A comunicação possibilita que ajustes de controle sejam feitos a partir do servidor OPC, que são então enviados de volta para os atuadores nos tanques.

Para validar a configuração do sistema, serão realizados ensaios simulando diversas condições operacionais. O objetivo era verificar a resposta do sistema de tanques ao controle automático, bem como a eficácia da rede CAN e do servidor OPC na execução de comandos em tempo real.

3.3.1 Identificação de equipamentos segundo a norma ISA

Para assegurar a organização e interconexão eficiente dos instrumentos na planta, cada dispositivo é atribuído uma tag única. Essa medida é crucial para garantir que os dados transmitidos sejam recebidos e processados pelo nó correto. A Tabela 2 apresenta uma lista das TAGs designadas a cada instrumento presente no sistema.

Tabela 2 – Tabela de TAGs de equipamentos conforme Norma ISA

TAG	Descrição	Localização
FIT-11	Controlador Indicador de Vazão	Tanque 1
FIT-12	Transmissor Indicador de Vazão	Tanque 1
FIC-13	Controlador Indicador de Vazão	Tanque 2
FIT-14	Transmissor Indicador de Vazão	Tanque 2
LT-15	Transmissor Indicador de Nível	Tanque 1
LT-16	Transmissor Indicador de Nível	Tanque 2
FIC-17	Controlador Indicador de Vazão	Tanque 4
FIT-18	Transmissor Indicador de Vazão	Tanque 4
FIC-19	Controlador Indicador de Vazão	Tanque 3
FIT-21	Transmissor Indicador de Vazão	Tanque 3
LT-22	Transmissor Indicador de Nível	Tanque 4
LT-23	Transmissor Indicador de Nível	Tanque 3
FIT-24	Transmissor Indicador de Vazão	Saída Reservatório
FIT-25	Transmissor Indicador de Vazão	Saída Reservatório
P-01	Bomba 1	Reservatório
P-02	Bomba 2	Reservatório
P-03	Bomba 3	Tanque 3
P-04	Bomba 4	Tanque 1

Fonte: Autor.

3.3.2 Composição da planta

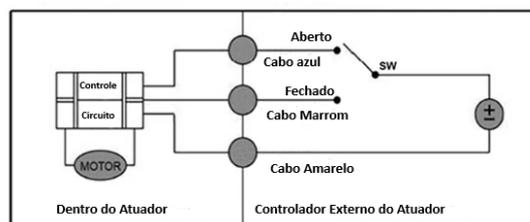
A planta foi projetada pensando no custo-benefício para as aplicações de automação e controle do sistema proposto. Dessa maneira, entre os instrumentos têm-se sensores de nível e de vazão, além de alguns atuadores, como detalhado a seguir. No entanto, devido à aquisição dos equipamentos de fabricantes genéricos visando a redução de custos, não foi possível obter os *datasheets* dos mesmos. Os dados apresentados nas tabelas são fornecidos pelos vendedores dos equipamentos.

Conforme ilustrado na Figura 16 e detalhado na Tabela 3, as válvulas esferas motorizadas são dispositivos utilizados para controlar a vazão de líquidos, projetados com motores síncronos, além de possuírem contatos de limite integrados, permitindo que a válvula fique sem energia quando atinge as posições totalmente aberta ou fechada, economizando energia e reduzindo o desgaste.

Figura 16 – Componentes da válvula motorizada.



(a) Válvula esfera motorizada.



(b) Esquema de ligação do atuador da válvula.

Fonte: Autor.

As válvulas solenoides On/Off, como pode ser visto na Figura 17 e na Tabela 4, são normalmente fechadas (N/F). A válvula permanece fechada até que uma corrente elétrica seja aplicada ao solenoide, criando um campo magnético que levanta o atuador da válvula, permitindo a vazão. Quando a corrente é interrompida, uma mola retorna o atuador à sua posição original, fechando a válvula e cessando a vazão.

Tabela 3 – Especificações das válvulas esfera motorizadas

Componente	Especificação
Material	Latão
Modo de Fiação	Três-fios dois-controle; Três-fios um-controle
Tensão	12VDC
Consumo	6W (somente na abertura e fechamento)
Temperatura	0-95°C
Ação Tempo	5-15 s
Pressão Nominal	1,6Mpa

Fonte: Autor.

Figura 17 – Válvula solenoide.



Fonte: Autor.

Tabela 4 – Especificações das válvulas solenoides on/Off

Componente	Especificação
Tensão	DC 12V
Potência	5W
Operação	Normalmente fechado (N/C)
Pressão	0,02-0,8Mpa
Temperatura	0-100°C

Fonte: Autor.

O sensor de vazão modelo YF-S201, como pode ser visto na Figura 18 e na Tabela 5, é um dispositivo utilizado para monitorar e medir a taxa de vazão de um líquido, composto por uma válvula de plástico, um rotor e um sensor de efeito Hall. Quando a água passa pelo rotor, ele gira e sua velocidade varia conforme a taxa de Vazão. Logo, o sensor de efeito Hall emite um sinal de pulso correspondente.

Figura 18 – Sensor de Vazão YF-S201.



Fonte: Autor.

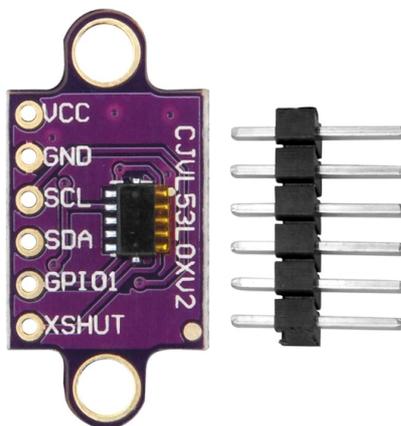
Tabela 5 – Especificações dos sensores de Vazão

Componente	Especificação
Modelo	YF-S201
Tensão de Trabalho	DC 4.5V - 24V
Faixa de Medição	1 - 30L/min
Temperatura de Operação	$\leq 80^{\circ}\text{C}$
Pressão Permitida	$\leq 1,75\text{MPa}$

Fonte: Autor.

O sensor VL53L0X, como pode ser visto na Figura 19 e na Tabela 6, foi escolhido como sensor de nível devido ao seu baixo custo, embora existam sensores mais específicos para esta finalidade. Ele mede com precisão a distância até a superfície da água, emitindo um laser que reflete no líquido e retorna ao sensor. Isso permite a determinação do nível da água em tempo real. Contudo, para assegurar a precisão das medições, é necessário o uso de um anteparo de cor opaca para otimizar a detecção do laser pelo sensor, fazendo esse sensor uma escolha apropriada para o projeto proposto.

Figura 19 – Sensor VL53L0X.



Fonte: Autor.

Tabela 6 – Especificações dos Sensores de Nível VL53L0X

Componente	Especificação
Aplicação	Medição de distância, controle de acesso, robótica
Tecnologia	<i>Time-of-Flight (ToF)</i> , <i>SPAD array</i>
Faixa de Medição	Até 2m
Temperatura de Operação	-25 a 80°C

Fonte: Autor.

Conforme ilustrado na Figura 20 e detalhado na Tabela 7, As bombas de água 12V utilizadas neste projeto são equipadas com motores sem escova, com baixo consumo de energia, com uma vazão de 800 litros por hora, podendo trabalhar fora ou dentro da água.

Figura 20 – Bomba de água 12V.



Fonte: Autor.

Tabela 7 – Especificações da bomba de água 12V

Característica	Especificação
Material	ABS
Entrada	DC12V
Potência	19W
Corrente Nominal Máxima	1000MA
Caudal	800L/h
<i>Max Water Lift</i>	5 metros
Temperatura do ambiente de trabalho	1-60 °C
Temperatura Máxima da água circulante	100 °C
Ruído	< 40 db
Modo de Condução	Eletromagnético
Posição do Eixo da Bomba	Horizontal
Estrutura do Impulsor	Impulsor aberto
Linha do Cabo	500mm
Tamanho do Item	77 × 65 × 42mm
Peso do Pacote	220g
Tamanho do Pacote	80 × 70 × 50mm

Fonte: Autor.

3.4 Integração do sistema supervisório com o servidor OPC

O sistema supervisório, em contextos de automação industrial, é uma plataforma de *software* que permite a operadores humanos monitorar e intervir em processos de controle automatizados. Neste projeto, essa plataforma é essencial para a interface de usuário, facilitando a visualização de dados, o controle de atuadores e a interação com o sistema de controle distribuído da planta didática. Com a implementação no MATLAB usando o App Designer, o sistema supervisório serve como um painel centralizado onde informações de múltiplos sensores e atuadores são exibidas, e comandos podem ser enviados de forma segura e eficiente.

A comunicação entre o sistema supervisório e os microcontroladores Atmega 328p é mediada pelo servidor OPC, cuja configuração é ilustrada na Figura 14. O servidor OPC, estabelecido no computador, serve como um intermediário para enviar comandos e receber dados dos microcontrolados. Por fim, o sistema supervisório atuará como cliente OPC, acessando os dados processados pelo servidor OPC, de forma, garantir que as funcionalidades de monitoramento e controle possam ser testadas e validadas dentro do contexto funcionamento estabelecida para a planta.

3.5 Considerações parciais

Neste capítulo, foram apresentados os dispositivos eletrônicos que constituem a base do sistema proposto neste trabalho. Foram demonstrados, por diagramas esquemáticos, o esquema elétrico desenvolvido durante os ensaios e discutido as ferramentas utilizadas para gerenciar a planta didática.

Além disso, foi detalhado as etapas de configuração e uso de um servidor OPC. Este servidor desempenha um papel crucial ao facilitar a comunicação entre os dispositivos conectados à rede CAN e um computador. Este último, por sua vez, utiliza o *software* MATLAB e o OPC Toolbox para monitorar e interagir com os dados do sistema em tempo real. Por outro lado, é importante destacar o desenvolvimento de um sistema supervisório, criando uma interface gráfica que permite aos usuários visualizar o estado atual da planta didática. Este sistema supervisório é integrado ao restante do projeto através do servidor OPC, garantindo uma comunicação eficaz entre a interface de usuário e os dispositivos de campo.

4 Resultados e discussões

Neste Capítulo, são apresentados os principais resultados obtidos ao longo do desenvolvimento deste trabalho. Inicialmente, a série de ensaios foi realizada com dois microcontroladores interligados por uma rede CAN. Os dados coletados por esses ensaios foram transmitidos para um servidor OPC e, posteriormente, acessados e analisados utilizando o *software* MATLAB por meio do OPC *toolbox*. Essa configuração preliminar serviu como base para a construção e teste do sistema final desenvolvido, no qual ocorre a coleta contínua de dados dos sensores de nível e de vazão. Esses dados são transmitidos através da rede CAN desenvolvida e acessados via cliente OPC e essa integração permite o monitoramento em tempo real do sistema.

4.1 Análise dos sinais transmitidos pela rede

Para a análise dos resultados transmitidos pela rede CAN, utilizou-se um analisador lógico de 8 canais, com frequência de amostragem de 24 MHz, impedância de entrada de 100 k Ω e suporte a tensões de operação de até 5,5 V. Esse dispositivo foi controlado a partir do *software* Logic 2, desenvolvido pela empresa Salae. A Figura 21 apresenta uma foto do analisador lógico utilizado.

Figura 21 – Analisador lógico.



Fonte: Autor.

Por meio dessa ferramenta, pode-se adquirir sinais digitais a partir de um computador, usando para isso um *buffer* de memória que pode ser ajustado com valores entre 0,5 GB e 3 GB.

O *software* utilizado em conjunto com o analisador lógico possui um conjunto de ferramentas adicionais que podem ser utilizadas para decodificar as informações recebidas. Essas ferramentas, além de úteis, simplificam o processo de validação dos resultados.

Por se tratar de um dispositivo de 8 canais, pode-se monitorar de forma simultânea 8 sinais diferentes. Entretanto, é importante destacar que a frequência de amostragem

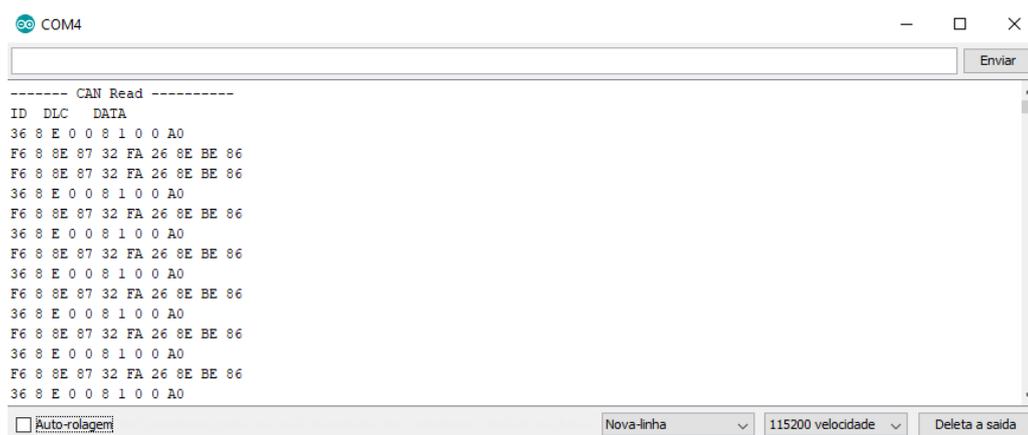
especificada pelo fabricante será dividida pelo número de canais utilizados.

No primeiro ensaio realizado, foram enviadas duas mensagens, cada uma com um ID de mensagem diferente e dados definidos de forma arbitrária utilizando valores hexadecimais, a saber:

- A primeira mensagem enviada tinha o ID 0xF6 e os dados: 0x8E, 0x87, 0x32, 0xFA, 0x26, 0x8E, 0xBE e 0x86. O ID dessa mensagem tem 8 bits de comprimento.
- A segunda mensagem enviada tinha o ID 0x37 e os dados: 0x0E, 0x00, 0x00, 0x08, 0x01, 0x00, 0x00 e 0xA0. O ID dessa mensagem também possui 8 bits de comprimento, representado por um único byte.

Como comentado, um dos módulos foi utilizado como emissor, e outro foi configurado como receptor. Os dados recebidos pelo módulo receptor foram enviados a um computador por meio de comunicação serial RS-232. O resultado obtido pode ser visualizado na Figura 22.

Figura 22 – Mensagens recebidas pelo módulo receptor.

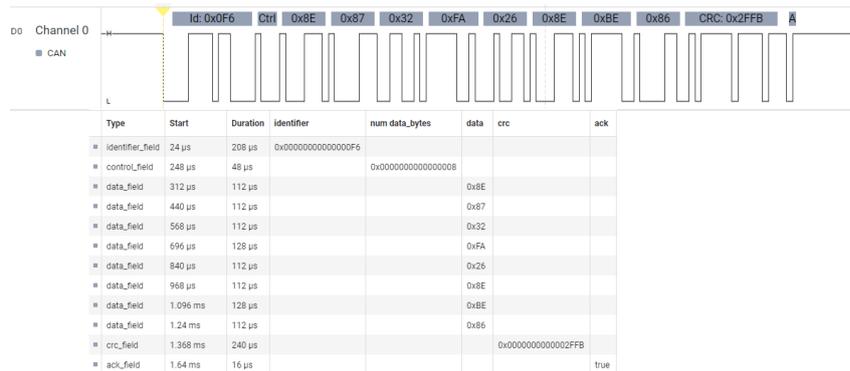


Fonte: Autor.

Em seguida, conectou-se o canal 1 do analisador lógico ao terminal CANH do módulo MCP 2515. Essa conexão foi possível pois o terminal CANH apresenta valores de tensão de, aproximadamente, 5 V, sendo compatível com os limites de tensão especificados pelo fabricante do analisador lógico. A Figura 23 apresenta a forma de onda obtida a partir do *software* Logic durante esta análise.

No teste seguinte, foram transmitidos pela rede CAN valores de tensão adquiridos a partir da tensão no terminal central de um potenciômetro alimentado com uma tensão de 5 V. Uma vez que o conversor analógico-digital do microcontrolador Atmega 328p apresenta uma resolução de 10 bits, foi necessário dividir a palavra digital adquirida em 2 bytes, sendo o primeiro responsável por armazenar os 2 bits mais significativos, e o segundo responsável por armazenar os 8 bits menos significativos.

Figura 23 – Dados coletados pelo *software* da Salae.



(a) Primeira mensagem enviada.

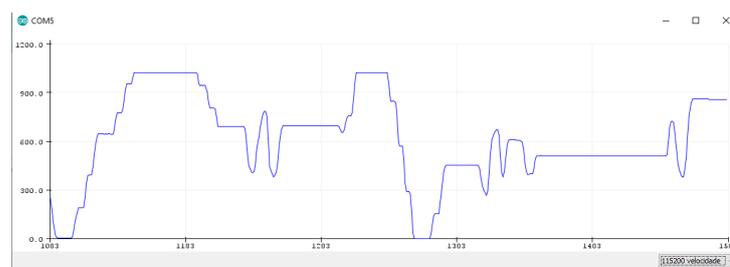


(b) Segunda Mensagem Enviada.

Fonte: Autor.

Assim como no caso anterior, os dados enviados pelo módulo transmissor foram recebidos pelo módulo receptor, que transferiu os sinais via protocolo serial RS-232 para um computador. Esses dados foram exibidos em forma de gráfico, como pode ser visto na Figura 24.

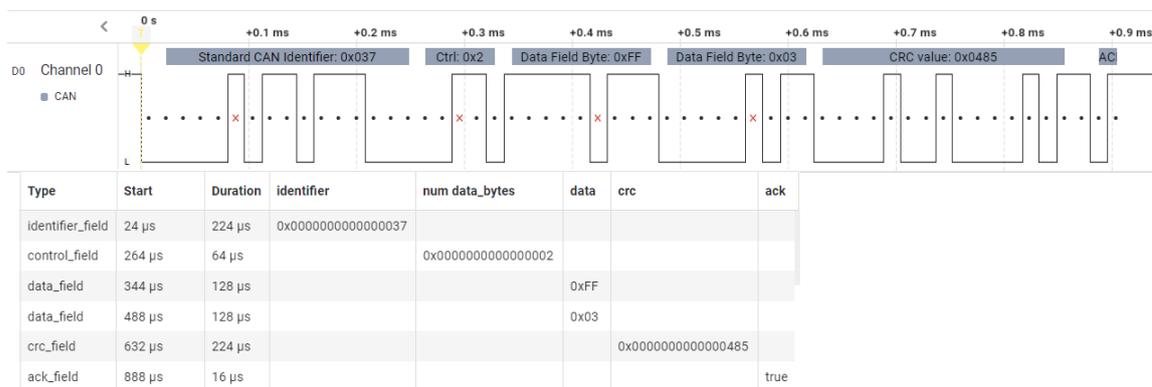
Figura 24 – Valores digitais recebidos pelo módulo receptor. Os dados foram adquiridos pelo módulo transmissor e enviados pela rede CAN.



Fonte: Autor.

A Figura 25 apresenta a forma de onda obtida a partir do *software* Logic quando o terminal CANH foi monitorado.

Figura 25 – Dados coletados pelo *software* da Salae.



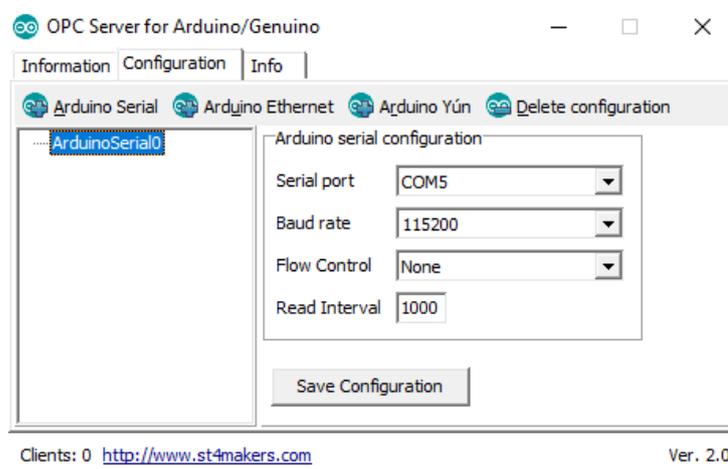
Fonte: Autor.

Com base nos testes realizados, pode-se verificar que os dados transmitidos foram recebidos corretamente pelo módulo receptor. A utilização do analisador lógico mostrou-se bastante útil para avaliar e diagnosticar problemas nesse tipo de rede, permitindo a identificação rápida e precisa de falhas no sistema. A partir dos resultados desses testes, é possível fazer ajustes e melhorias na rede para garantir seu bom funcionamento e alto desempenho, o que é fundamental para garantir a confiabilidade e segurança das aplicações que utilizam a rede CAN.

4.2 Servidor OPC

A Figura 26 apresenta uma imagem com a tela de configuração inicial do *software* utilizado para implementar o servidor OPC empregado neste trabalho. O processo de instalação envolve extrair o conteúdo do arquivo zip e executar o arquivo correspondente. O *software* pode ser baixado a partir do site oficial da *st4makers*.

Figura 26 – Tela de configuração do servidor OPC.

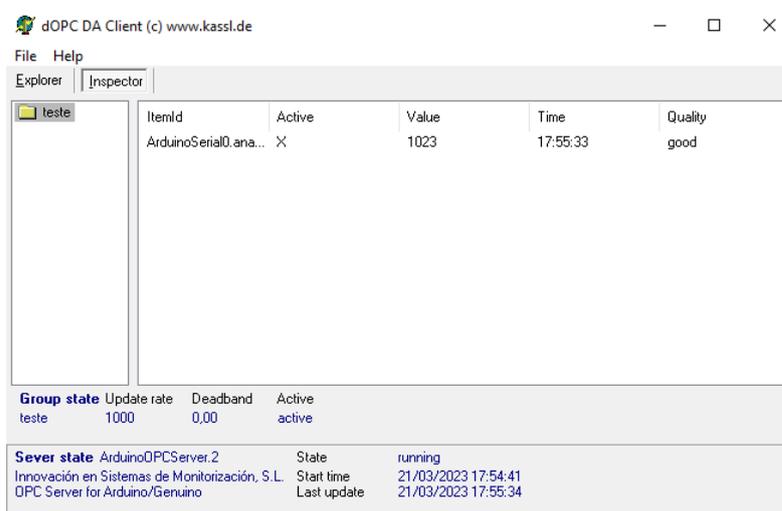


Fonte: Autor.

Na etapa de configuração inicial, deve-se definir os parâmetros necessários para que seja possível estabelecer uma comunicação entre computador e microcontrolador que irá executar as aplicações OPC por meio de interface serial RS-232. Neste trabalho, definiu-se um *baud rate* de 115200 bps sem controle de fluxo como padrão para a comunicação serial, e uma taxa de atualização de 1 Hz para o servidor OPC. Ressalta-se que esses valores poderiam ser ajustados de forma diferente, dependendo da aplicação a ser desenvolvida.

Para verificar se o servidor OPC funcionava de forma adequada, utilizou-se o *software dOPC DA Client*, da fabricante Kassl. Com esta ferramenta, é possível verificar o nome do *ItemID* criado, o valor analógico associado e a qualidade da transmissão. A Figura 27 ilustra a tela principal do *software*, na qual essas informações podem ser visualizadas de forma clara e intuitiva.

Figura 27 – Tela principal exibida pelo *software* OPC client dOPC.



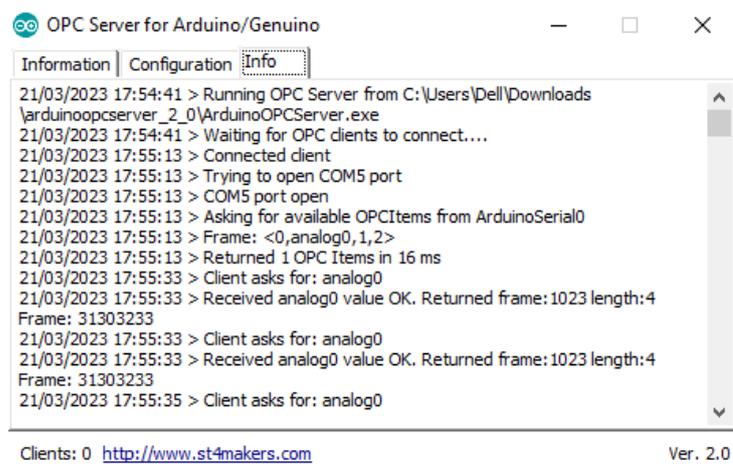
Fonte: Autor.

A Figura 28 apresenta uma imagem da tela de informações presente no *software* OPC Server instalado no computador. Como pode ser visto, a nova conexão é reconhecida pelo *software*, que passa a gerenciar o processo de comunicação. Além disso, sempre que uma nova solicitação é feita, essa tela exibe uma mensagem indicando quais informações foram solicitadas e quais requisições foram atendidas.

Por meio dessa ferramenta, estabeleceu-se uma primeira conexão entre cliente e servidor OPC. Como pode ser visto na imagem, o *software* exibe um conjunto de *tags*, definidos pelo microcontrolador usado no processo de construção do servidor. Além disso, pode-se verificar a qualidade do sinal e o instante de tempo correspondente ao último acesso.

Na sequência, substituiu-se o *software* cliente pelo MATLAB. Nessa etapa, a conexão foi realizada por meio do OPC *toolbox*, que foi configurada para acessar o servidor previamente definido. Assim que uma nova conexão é estabelecida, informações relacio-

Figura 28 – Conexão do cliente OPC no servidor OPC.I.



```

OPC Server for Arduino/Genuino
Information | Configuration | Info
21/03/2023 17:54:41 > Running OPC Server from C:\Users\Dell\Downloads
\arduinoopcserver_2_0\ArduinoOPCServer.exe
21/03/2023 17:54:41 > Waiting for OPC clients to connect...
21/03/2023 17:55:13 > Connected client
21/03/2023 17:55:13 > Trying to open COM5 port
21/03/2023 17:55:13 > COM5 port open
21/03/2023 17:55:13 > Asking for available OPCItems from ArduinoSerial0
21/03/2023 17:55:13 > Frame: <0,analog0,1,2>
21/03/2023 17:55:13 > Returned 1 OPC Items in 16 ms
21/03/2023 17:55:33 > Client asks for: analog0
21/03/2023 17:55:33 > Received analog0 value OK. Returned frame: 1023 length:4
Frame: 31303233
21/03/2023 17:55:33 > Client asks for: analog0
21/03/2023 17:55:33 > Received analog0 value OK. Returned frame: 1023 length:4
Frame: 31303233
21/03/2023 17:55:35 > Client asks for: analog0
Clients: 0 http://www.st4makers.com Ver. 2.0

```

Fonte: Autor.

nadas ao servidor, grupo de mensagens, status e qualidade da transmissão são exibidas no terminal, como o exemplo apresentado na Figura 29.

Figura 29 – Status da conexão estabelecida entre servidor OPC e Matlab OPC *toolbox*.

```

Server Parameters
Host: localhost
ServerID: ArduinoOPCServer.2
Status: connected
Timeout: 10 seconds
Object Parameters
Group: 0-by-1 dagroup object
Event Log: 0 of 1000 events
ItemID: 'Arduinoseria10.analog0'
Value 50
Quality: 'good: Non- specific'
TimeStamp: [2023 3 15 3 14 31.31.9360]
Error: "

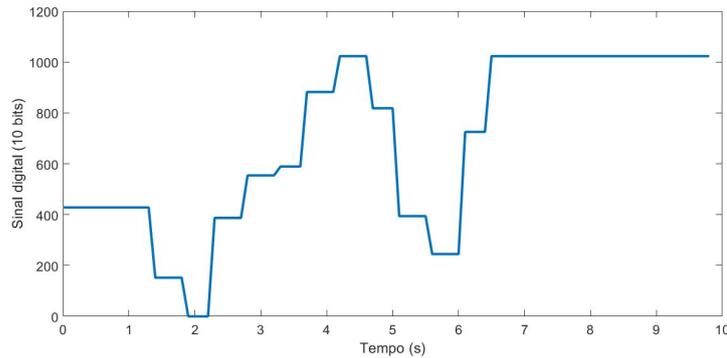
```

Fonte: do autor.

Na sequência, os valores de tensão medidos pelo microcontrolador transmissor e enviadas para o servidor OPC foram adquiridas pelo MATLAB e exibidas por meio de um gráfico atualizado em tempo real. Uma imagem estática desse sistema pode ser visualizada na Figura 30.

Com os dados disponíveis no ambiente MATLAB, pode-se manipular os dados usando funções predefinidas. Além disso, é possível realizar testes com algoritmos de controle ou processamento digital de sinais nessa plataforma, sem que seja necessário reprogramar os microcontroladores conectados à rede.

Figura 30 – Gráfico em tempo real dos valores do potenciômetro no MATLAB.



Fonte: Autor.

4.3 Projeto de uma planta piloto

Esta seção descreve a fase final do projeto, na qual uma planta piloto, composta por quatro tanques interconectados, foi implementada. É importante destacar que, para os propósitos deste estudo, os sensores e atuadores utilizados não foram calibrados, focando primariamente na avaliação da comunicação entre os dispositivos e o servidor OPC através da rede CAN.

Utilizando a rede CAN para a comunicação entre os microcontroladores e empregando um servidor OPC para a interface com o MATLAB, se estabelece um sistema capaz de monitorar e ajustar o sistema em tempo real. A Figura 31 apresenta uma foto do protótipo utilizado neste trabalho.

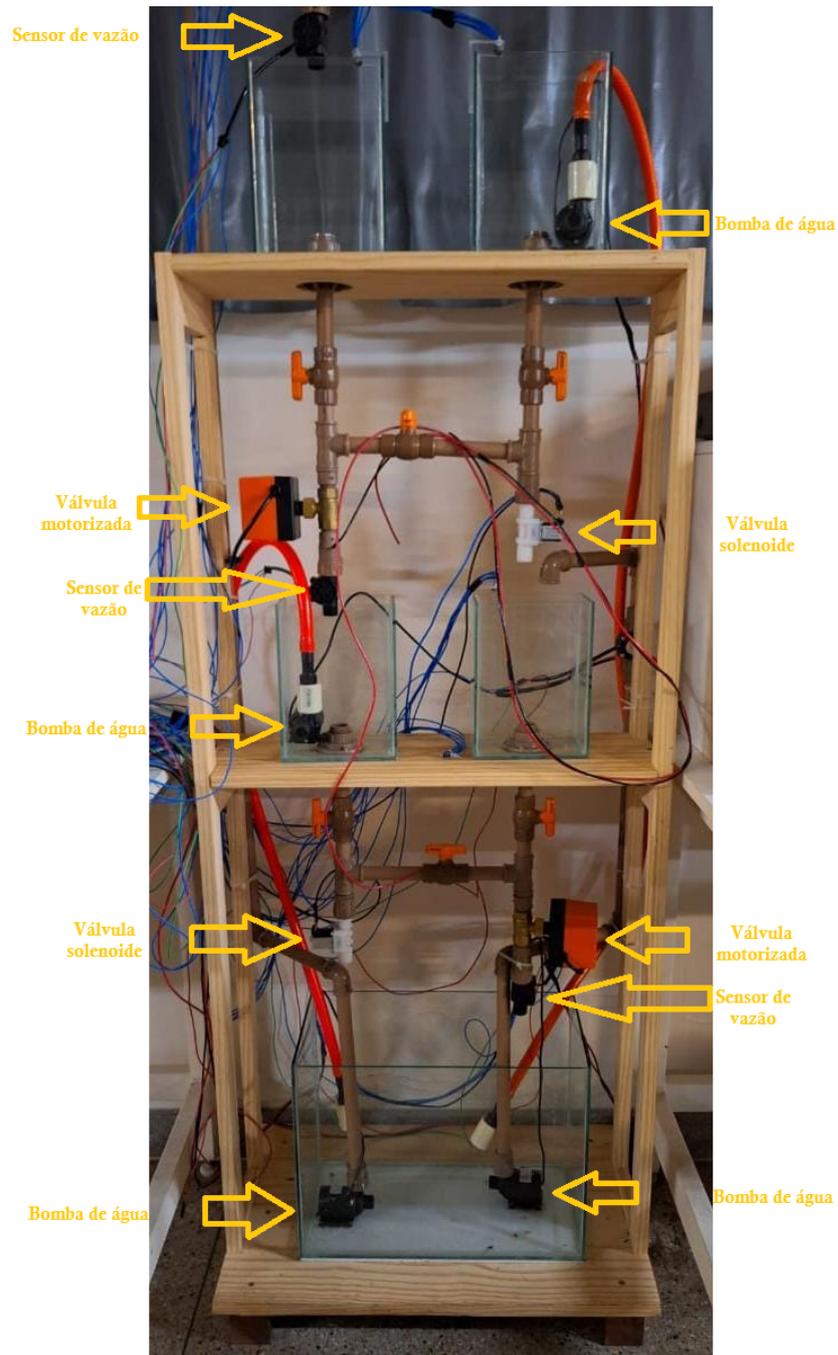
Esta estrutura visa não apenas apresentar o processo de configuração, mas também fornecer uma análise detalhada dos resultados alcançados, levando em consideração a comunicação entre os dispositivos e o servidor OPC como o foco principal dos testes realizados.

4.3.1 Análise do sistema

Nesta etapa, realizaremos uma análise detalhada da rede projetada, utilizando o analisador lógico, o *software* Logic 2, da Salae, para extrair os dados da rede e analisar esses dados, visando compreender o comportamento e a eficiência da rede.

A rede foi configurada com um oscilador de 8 MHz e com uma taxa de transmissão de 125 kbps, uma configuração adequada para o ambiente de testes proposto. Essa escolha considerou a taxa de transmissão de bits, diretamente relacionada ao tempo necessário para transmitir e receber as mensagens, e o baixo índice de erros de transmissão ao longo do envio e recebimento de mensagens. Posteriormente, após ser configurada a rede, foram iniciados os testes e coletado os dados utilizando o analisador lógico, e o configurando para interceptar os pacotes transmitidos na rede CAN, conforme as imagens da Figura 32.

Figura 31 – Planta composta por quatro tanques interconectados.

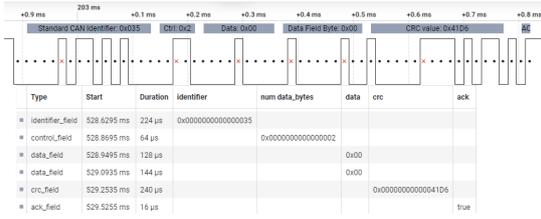


Fonte: Autor.

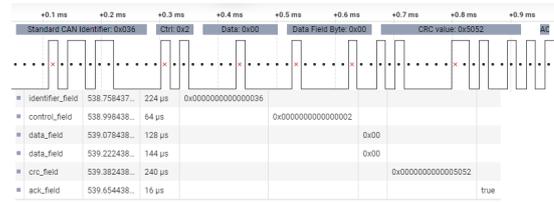
As imagens da Figura 32 também podem ser vistas no Apêndice B, em que cada imagem mostra uma mensagem CAN individual, identificada pelo seu campo de identificação único, como pode ser visto na Tabela 8.

Um aspecto a ser observado na Tabela 8 é o intervalo entre os IDs das mensagens CAN, que representa o atraso entre as transmissões na rede. O maior intervalo é de 99.394 ms, correspondente ao ID 0x035, que é o tempo que leva desde a última mensagem enviada pelo ID 0x043 até o início de um novo ciclo de envio das mensagens.

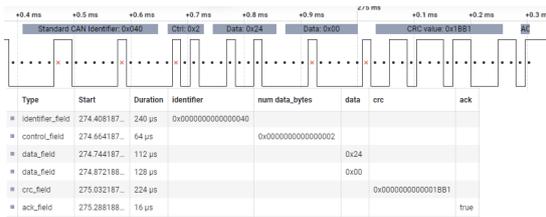
Figura 32 – Dados coletados pelo software da Salae.



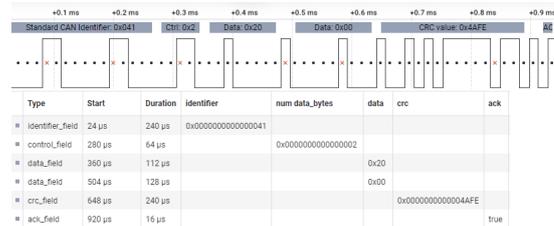
(a) Dados coletados do sensor FIT-11.



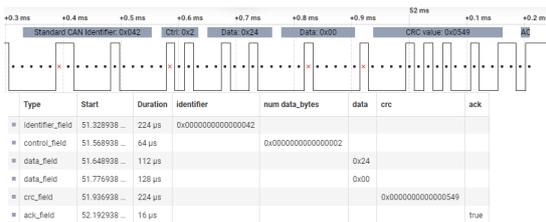
(b) Dados coletados do sensor FIT-21.



(c) Dados Coletados do sensor LT-15.



(d) Dados Coletados do sensor LT-16.



(e) Dados Coletados do sensor LT-22.



(f) Dados Coletados do sensor LT-23.

Fonte: Autor.

Tabela 8 – Detalhes das mensagens CAN

ID	Intervalo (ms)	Tamanho (bytes)	Duração (us)
0x035	99,394	2	816
0x036	9,217	2	816
0x040	60,442	2	784
0x041	50,393	2	800
0x042	50,376	2	768
0x043	50,392	2	784

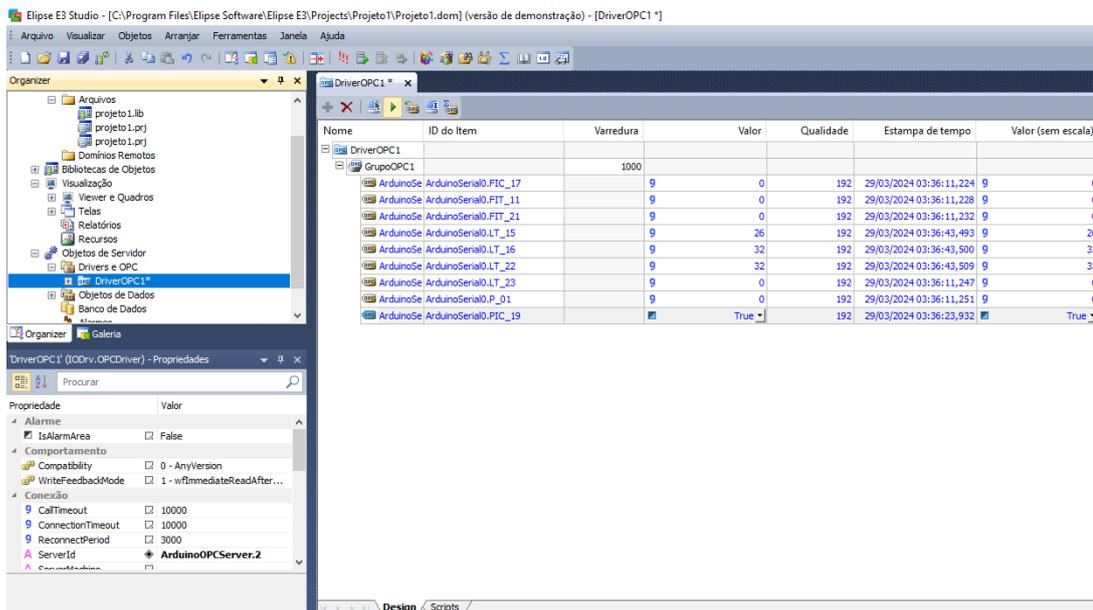
Além disso, o uso do analisador lógico possibilitou a observação direta da transmissão de mensagens na rede CAN, incluindo a verificação dos campos de identificação, controle, dados, CRC e ACK. Dessa maneira, se acabou garantindo a integridade e a autenticidade das mensagens transmitidas.

Em suma, a configuração da rede CAN projetada demonstrou ser bem-sucedida em termos de confiabilidade e eficiência, adequando-se perfeitamente às demandas do sistema proposto.

4.3.1.1 Cliente OPC

Posteriormente, para averiguar a operacionalidade do sistema desenvolvido, inicialmente, empregou-se o software Eclipse E3 Studio como cliente OPC alternativo, conforme ilustrado na Figura 33. A exibição dessa interface no Eclipse E3 serviu como uma verificação preliminar da funcionalidade do sistema, qualidade de transmissão e o *ItemID* criado de cada TAG gerada. Tal etapa foi crucial para assegurar que o servidor OPC estivesse efetivamente operacional antes da condução dos testes finais no MATLAB

Figura 33 – Interface do Cliente OPC Eclipse E3 Studio.



Fonte: Autor.

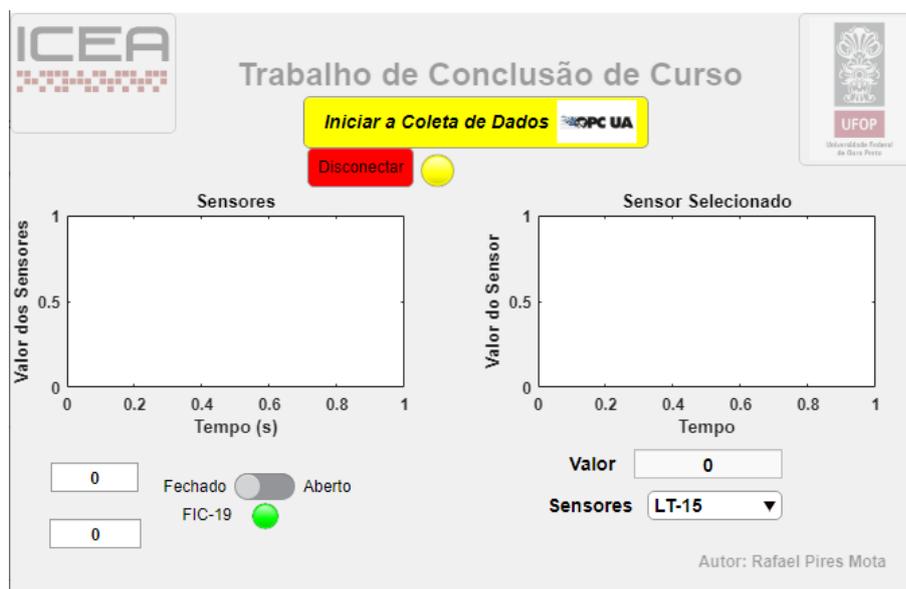
4.3.1.2 Supervisório

Nesta etapa, o desenvolvimento do sistema supervisorio é detalhado, evidenciando o uso combinado do App Designer do MATLAB e do OPC Toolbox para a criação de uma interface gráfica do usuário (GUI) interativa que se comunica em tempo real com dispositivos. A GUI finalizada é apresentada na Figura 34.

Antes de finalizar a IHM, foram realizados testes de conexão usando o OPC Toolbox. Esses testes consistiram em checar a coerência na transferência de dados e a eficiência do sistema. A integração do OPC Toolbox ao App Designer, permitiu a incorporação direta de funcionalidades de comunicação OPC dentro da lógica da IHM desenvolvida. Contudo, esse processo teve algumas dificuldades. Uma das vantagens foi a possibilidade de acompanhar em tempo real pela interface de usuário. Mas também houve pontos negativos, como a configuração complicada no começo da comunicação.

Na tela principal, dois gráficos são exibidos: o gráfico à esquerda apresenta dados em tempo real de múltiplos sensores, enquanto o gráfico à direita destaca os dados de

Figura 34 – Interface Gráfica Desenvolvida.



Fonte: Autor.

um único sensor selecionado pelo usuário. Abaixo dos gráficos, tem-se indicadores que mostram o estado atual dos sensores e dos atuadores. O botão “Iniciar a Coleta de Dados” foi projetado para iniciar a comunicação OPC. A caixa de seleção rotulada “Sensores” oferece ao usuário a possibilidade de escolher qual sensor monitorar individualmente, refletindo os dados selecionados no gráfico à direita e no indicador de valor adjacente.

4.4 Considerações Parciais

Neste capítulo, foi detalhado os resultados alcançados ao longo deste projeto. Inicialmente, por meio de ensaios focados, foi comprovado a eficácia da rede CAN implementada, que garantiu uma transmissão de dados precisa e confiável. O primeiro ensaio validou a rede com a transmissão bem-sucedida de mensagens, enquanto o segundo ampliou a aplicabilidade do sistema para a instrumentação, enviando informações digitais, resultantes de um processo de aquisição de sinais, de um módulo transmissor para um receptor.

A utilização de um analisador lógico foi essencial para monitorar e decodificar as formas de onda transmitidas via rede CAN, oferecendo uma visão detalhada do comportamento dos sinais no pino CANH., Além disso, a implementação do servidor OPC se mostrou uma solução simples e prática, facilitando a comunicação e a troca de informações.

Além disso, neste trabalho foram realizados ensaios com o sistema supervisorio desenvolvido utilizando o OPC *toolbox* e o *App Designer* do MATLAB. Esta abordagem permitiu o processamento flexível dos dados coletados, permitindo que dados adquiridos a

partir do servidor possam ser processados usando diferentes abordagens sem a necessidade de se reprogramar os microcontroladores.

A implementação do sistema final e do supervisor ilustrou o potencial da integração de tecnologias de comunicação avançadas para o desenvolvimento de soluções de automação e controle industriais eficazes. Esta etapa destacou a versatilidade e a capacidade do nosso sistema em lidar com aplicações complexas, abrindo caminho para futuras inovações no campo da automação industrial

5 Considerações finais

Este trabalho apresentou as etapas de projeto e validação de sistemas de controle e monitoramento utilizando microcontroladores de baixo custo e a rede CAN para automação industrial. O sistema foi desenvolvido utilizando microcontroladores Atmega 328p e módulos de comunicação MCP 2515. Essa escolha considerou a facilidade de acesso a esses dispositivos no mercado local, além da possibilidade de programação por meio da IDE Arduino, que apresenta diversas bibliotecas com licença livre que podem ser utilizadas ao longo do desenvolvimento de aplicações.

Os ensaios realizados mostraram que, ao conectar dois microcontroladores numa mesma rede CAN, foi possível transferir dados a uma taxa de 125 kpbs, com uma taxa de atualização de 10 Hz. Logo, trata-se de uma configuração adequada para as aplicações de monitoramento e controle propostas por este trabalho. Além disso, a utilização de um servidor OPC se mostrou uma estratégia eficiente para controlar o processo de troca de informações entre dispositivos diferentes conectados em rede.

A opção pela utilização do MATLAB, juntamente com o OPC *toolbox*, como cliente simplificou o processo de desenvolvimento de aplicações, permitindo que diversas funções associadas a outros *toolboxes* pudessem ser utilizadas para implementar estratégias de controle ou algoritmos de processamento digital de sinais, sem a necessidade de reprogramação dos módulos microcontrolados. Por outro lado, o desenvolvimento do sistema supervisorio utilizando o MATLAB, o OPC Toolbox e o App Designer proporcionou uma interface interativa, garantindo um monitoramento efetivo e em tempo real.

Por fim, a integração do servidor OPC estabeleceu uma base sólida para o intercâmbio de dados entre dispositivos distintos na rede. A escolha do MATLAB como cliente OPC simplificou significativamente o processo de desenvolvimento de aplicações, por permitir o aproveitamento das funções disponíveis em outros *toolboxes* do MATLAB. Esta integração provou ser uma solução eficiente para ambientes em que os tempos de processo são relativamente longos, considerando que a latência da rede e o tempo de acesso do MATLAB ao servidor OPC podem impor limitações à frequência de atualização de mensagens.

Em suma, os resultados alcançados neste trabalho não só reiteram a viabilidade de microcontroladores de baixo custo e rede CAN em aplicações de automação, mas também abrem caminho para a exploração de técnicas avançadas de controle e estratégias de processamento de sinais que podem ser incorporadas sem a necessidade de alterações complexas na programação dos dispositivos de campo. O sistema proposto, portanto, representa um passo significativo rumo à inovação e otimização em automação industrial, com potencial para expansão e adaptação em aplicações futuras que demandem maior complexidade e integração.

5.1 Sugestões de trabalhos futuros

A partir, dos resultados obtidos neste projeto, foram definidas algumas possibilidades para avanços e melhorias futuras, tais como:

- **Substituição do microcontrolador:** Atualização do microcontrolador atual para o ESP32, devido as suas capacidades, como maior poder de processamento e conectividade integrada. Essa fase incluirá a adaptação e otimização do código existente, além de testes para assegurar a funcionalidade com o novo *hardware*.
- **Utilização do sistema de RTOS:** Integração de um Sistema Operacional de Tempo Real para uma gestão mais eficiente das tarefas. Um RTOS facilitará o manejo de operações concorrentes e melhorará a precisão temporal das atividades.
- **Sistema de Controle Integrado:** Desenvolvimento e validação de um sistema de controle que permita a gestão conjunta de nível e vazão na planta. Serão projetados algoritmos de controle, com foco na melhoria do desempenho operacional e na redução de custos.
- **Exploração de Novas Aplicações:** Após a implementação bem-sucedida dos passos anteriores, podem-se explorar novos domínios de aplicação para o sistema. O intuito é investigar a viabilidade do uso dessa tecnologia em diferentes contextos, desde aplicações domésticas até complexos sistemas industriais.

Referências

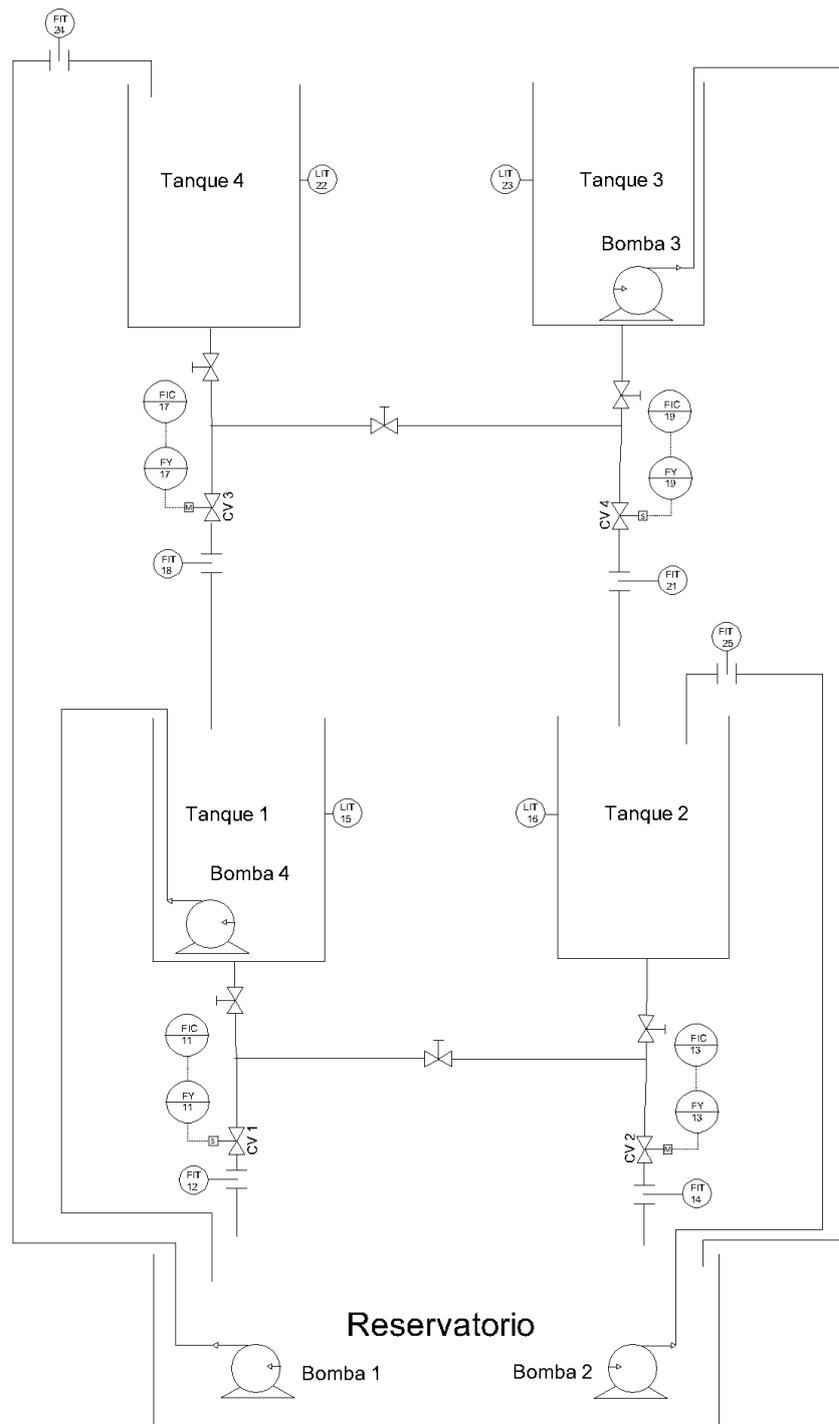
- AMATO, F. et al. Can-bus attack detection with deep learning. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 22, n. 8, p. 5081–5090, 2021.
- AUTOWP. *arduino-mcp2515*. 2023. Disponível em: <<https://github.com/autowp/arduino-mcp2515/>>.
- BOZDAL, M. et al. Evaluation of can bus security challenges. *Sensors*, MDPI, v. 20, n. 8, p. 2364, 2020.
- CAO, J.; YE, Q.; LI, P. Resistance furnace temperature control system based on opc and matlab. *Measurement and Control*, SAGE Publications Sage UK: London, England, v. 48, n. 2, p. 60–64, 2015.
- DHIVYA, M.; DEVI, K. G.; SUGUNA, S. K. Design and implementation of steering based headlight control system using can bus. In: SPRINGER. *Second International Conference on Computer Networks and Communication Technologies: ICCNCT 2019*. Cham: Springer International Publishing, 2020. p. 373–383.
- ELERT, S. Development and implementation of a model in matlab simulink responsible for the analysis of sensor data via the can bus. *Journal of Physics: Conference Series*, IOP Publishing, v. 2224, 2022.
- FARINELLI, F. A. *Desenvolvimento de um sistema embarcado de bordo para monitoramento de parâmetros via diagnóstico de rede CAN*. 2016. Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) - Universidade Tecnológica Federal do Paraná (UTFPR), Ponta Grossa, Brasil.
- GERGELEIT, M.; STREICH, H. Implementing a distributed high-resolution real-time clock using the can-bus. In: CAN IN AUTOMATION (CIA). *Proceedings of the 1st International CAN Conference*. Mainz, Germany, 1994. v. 94. Sponsored by Allen Bradley, National Semiconductor, Philips Semiconductors.
- GUIMARÃES, A.; SARAIVA, A. M. O protocolo can: entendendo e implementando uma rede de comunicação serial de dados baseada no barramento “controller area network”. *Universidade de São Paulo. São Paulo, SP-Brasil*, p. 10, 2002.
- INSTRUMENTS, T. *Introduction to the Controller Area Network*. 2016. Application Note. Acessado em 12 de março de 2023.
- ISO/TC 22/SC 31. *ISO 11898: Road vehicles – Controller area network (CAN)*. 2015–2016. International Standard. Comprising multiple parts including but not limited to Part 1: Data link layer and physical signalling (2015), Part 2: High-speed medium access unit (2016), Part 3: Low-speed, fault-tolerant, medium-dependent interface (2016).
- KIM, D.-S.; TRAN-DANG, H. Industrial sensors and controls in communication networks. In: *Computer Communications and Networks*. Cham: Springer, 2019.
- LAUFENBERG, J.; KROPF, T.; BRINGMANN, O. Can simulation framework-from classic can to can xl. In: IEEE. *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. Bilbao, Spain, 2023. p. 3343–3348.

- LOKMAN, S.-F.; OTHMAN, A. T.; ABU-BAKAR, M.-H. Intrusion detection system for automotive controller area network (can) bus system: a review. *EURASIP Journal on Wireless Communications and Networking*, Springer, v. 2019, p. 1–17, 2019.
- MA, Y. Design of coal mine power grid parameter monitoring system based on can bus. In: *IOP Conference Series: Earth and Environmental Science*. Shenyang: IOP Publishing, 2021. v. 651, n. 3.
- MADHAVAN, S. et al. Controller area network (can)-based automatic fog light and wiper controller prototype for automobiles. In: *Emerging Research in Computing, Information, Communication and Applications: Proceedings of ERCICA 2022*. Singapore: Springer, 2022. p. 1003–1011.
- MAHNKE, W.; LEITNER, S.-H.; DAMM, M. *OPC unified architecture*. Berlin, Germany: Springer Science & Business Media, 2009.
- MAKERS, S. *OPC Server for Arduino*. 2023. [Online; acesso em 13-Março-2023]. Disponível em: <<https://www.st4makers.com/opc-server-for-arduino/>>.
- NATALE, M. D. et al. *Understanding and using the controller area network communication protocol: theory and practice*. New York: Springer Science & Business Media, 2012.
- OKSANEN, T.; LINKOLEHTO, R.; SEILONEN, I. Adapting an industrial automation protocol to remote monitoring of mobile agricultural machinery: a combine harvester with iot. *IFAC-PapersOnLine*, Elsevier, v. 49, n. 16, p. 127–131, 2016.
- PAWAR, U.; BHIRUD, S.; KOLHE, S. R. Study and prototype implementation of basic non-can and can based communication in context with automotive application (s). In: *Proceedings of the 3rd International Conference on Communication & Information Processing (ICCIP)*. Jalgaon, India: SSRN, 2021. Disponível em: <https://ssrn.com/abstract=3917797> or <http://dx.doi.org/10.2139/ssrn.3917797>.
- SACOMANO, J. B. et al. *Indústria 4.0*. São Paulo: Editora Blucher, 2018.
- SEMICONDUCTORS, N. *TJA1050 High Speed CAN Transceiver Datasheet*. 2007. <<https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf>>. Accessed: 2023-03-08.
- SONG, Y. Q.; LI, C. L. Monitoring system of can bus based on opc technology. In: *Advanced Materials Research*. Switzerland: Trans Tech Publications, 2013. v. 662, p. 713–716.
- TECHNOLOGY, M. *MCP2515 Stand-Alone CAN Controller with SPI Interface*. 2012. <<https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>>. Accessed: 2023-03-03.

Apêndices

Apêndice A – Diagrama P&I

A.1 Diagrama de instrumentação e tubulação implementado



Fonte: Autor.

A.2 Tabela com a simbologia

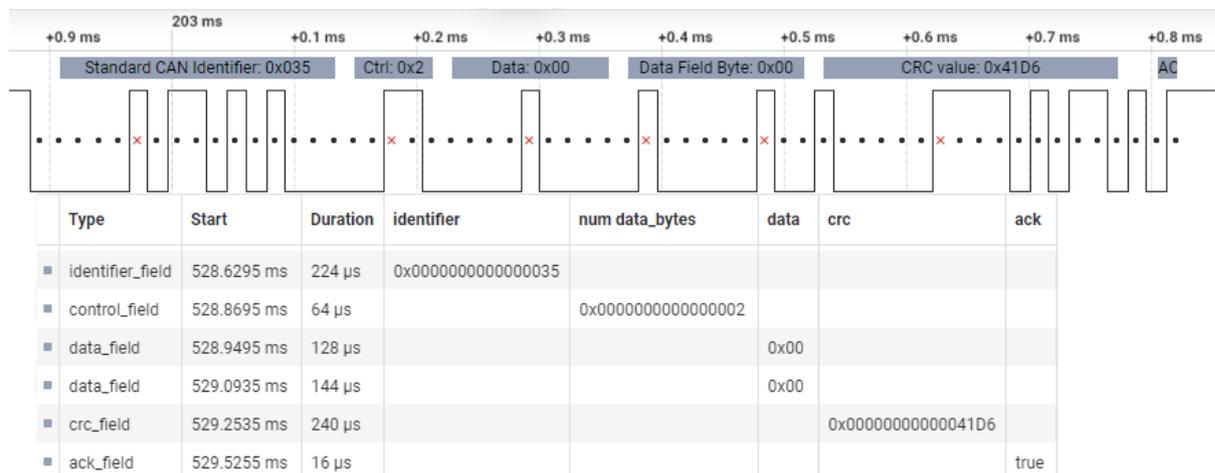
Símbolo	Nomenclatura
	Bomba D'agua
	Válvula Manual
	Válvula Motorizada
	Válvula Solenoide

Fonte: Autor.

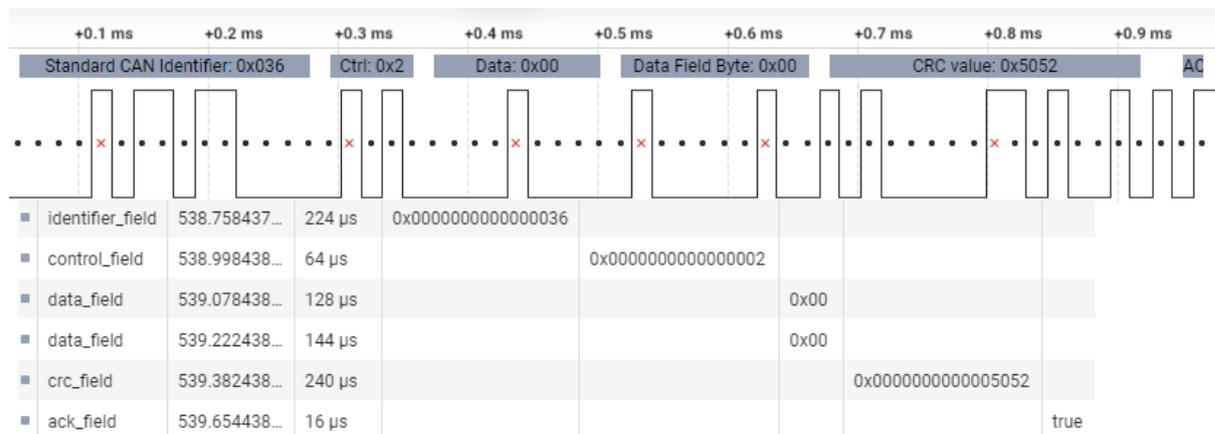
Apêndice B – Formas de onda

As figuras a seguir apresentam as formas de onda adquiridas pelos sensores de fluxo e de nível. Os testes foram realizados para verificar a comunicação dos sensores FIT-11, FIT-21, LT-15 e LT-16. Em cada figura, são mostrados os campos padrão da rede CAN, incluindo o identificador, campo de controle, campo de dados, campo CRC e campo de confirmação (ack) e o "x" indica bits de erro detectados durante a transmissão dos dados.

B.1 Sensores de fluxo FIT-11 e FIT-21.

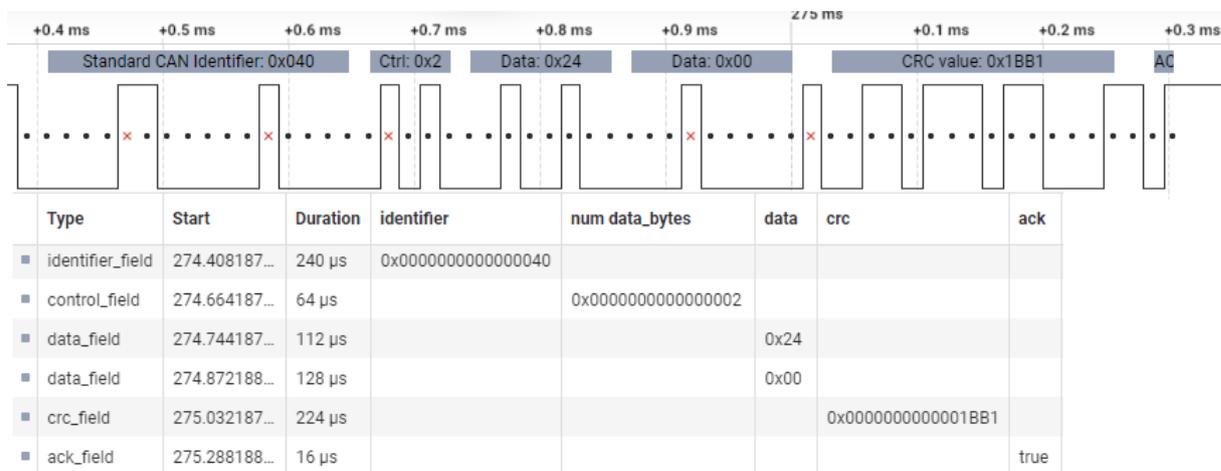


Fonte: Autor.

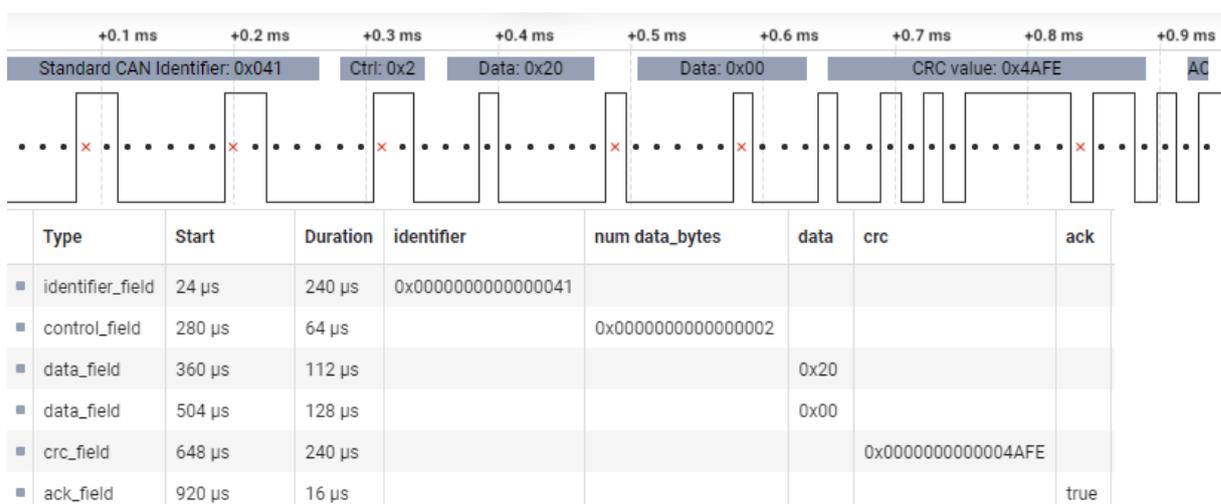


Fonte: Autor.

B.2 Sensores de nível LT-15, LT-16.

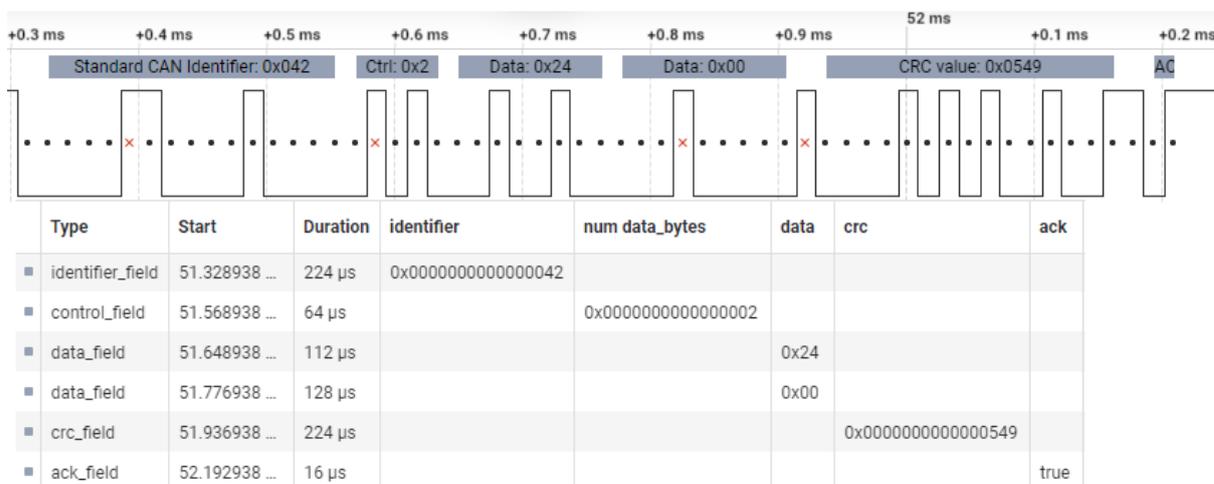


Fonte: Autor.

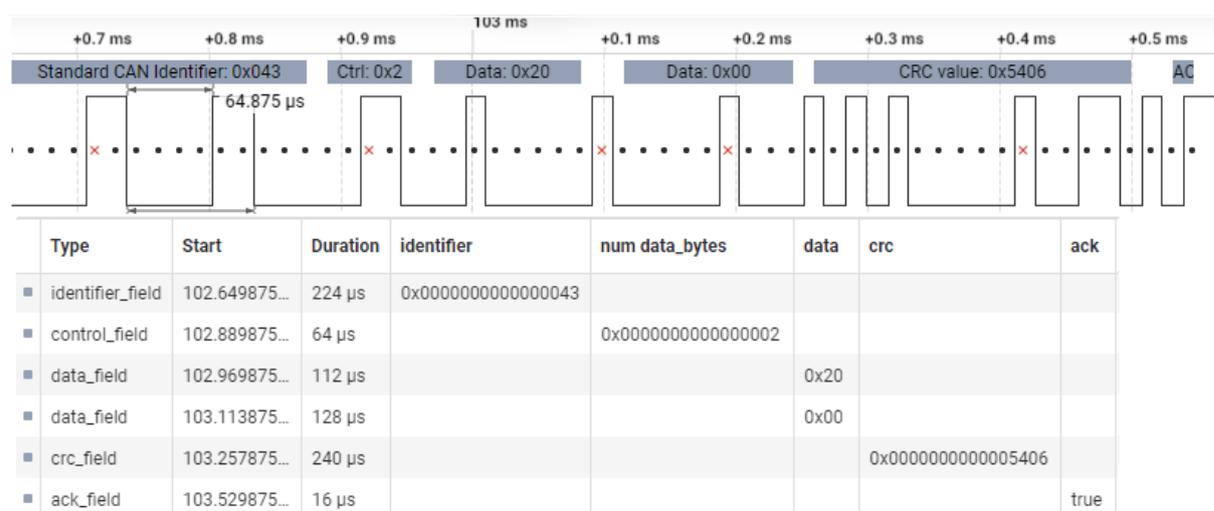


Fonte: Autor.

B.3 Sensores de nível LT-22 e LT-23.



Fonte: Autor.



Fonte: Autor.



MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Colegiado do Curso de Engenharia Elétrica



TERMO DE RESPONSABILIDADE

O texto do trabalho de conclusão de curso intitulado Estudo de técnicas para monitoramento e controle de instrumentos a partir de uma rede CAN é de minha inteira responsabilidade. Declaro que não há utilização indevida de texto, material fotográfico ou qualquer outro material pertencente a terceiros sem a devida citação ou consentimento dos referidos autores.

João Monlevade, 11 de junho de 2024.

Rafael Pires Mota