



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PAULO RICARDO MONETA NUNES

**PROPOSTA DE DISPOSITIVO DE COMUNICAÇÃO INDUSTRIAL
COMPATÍVEL COM A NORMA IEC 61850**

Recife

2024

PAULO RICARDO MONETA NUNES

**PROPOSTA DE DISPOSITIVO DE COMUNICAÇÃO INDUSTRIAL
COMPATÍVEL COM A NORMA IEC 61850**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Processamento de Energia

Orientador: Prof. DSc. Leonardo Rodrigues Limongi

Recife

2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Nunes, Paulo Ricardo Moneta.

Proposta de dispositivo de comunicação industrial compatível com a norma IEC 61850 / Paulo Ricardo Moneta Nunes. - Recife, 2025.

64f.: il.

Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Programa de Pós-Graduação em Engenharia Elétrica, 2025.

Orientação: Leonardo Rodrigues Limongi.

Inclui referências.

1. IEC 61850; 2. ESP32; 3. CLP; 4. IEC 61499. I. Limongi, Leonardo Rodrigues. II. Título.

UFPE-Biblioteca Central

PAULO RICARDO MONETA NUNES

**“PROPOSTA DE DISPOSITIVO DE COMUNICAÇÃO
INDUSTRIAL COMPATÍVEL COM A NORMA IEC 61850”**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica, na área de concentração em Processamento de Energia.

Aprovada em: 27/02/2025

BANCA EXAMINADORA

Prof.^o. Leonardo Rodrigues Limongi
(Orientador e Examinador Interno)
Universidade Federal de Pernambuco

Prof.^o. Dr. Márcio Evaristo da Cruz Brito
(Examinador Externo)
Universidade Federal de Pernambuco

Prof. Dr. Herbert Albérico de Sá Leitão
(Examinador Externo)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Agradeço à minha família, pelos ensinamentos de vida e pelo apoio ao longo do curso.

À Universidade Federal de Pernambuco, pelo ambiente de aprendizado e convivência que influenciou bastante a minha vida.

Aos amigos e amigas que tive a oportunidade de conhecer durante o mestrado e também na graduação.

Aos professores das Engenharias Elétrica, Eletrônica e Controle e Automação, responsáveis pelo aprendizado profissional e pessoal transmitido com muita dedicação, em particular ao meu orientador professor Limongi e professor Geraldo, sem o qual não seria possível a realização deste e de vários outros trabalhos.

Por fim, obrigado a todos que contribuíram para a realização deste trabalho.

RESUMO

Neste trabalho, é apresentada uma integração entre uma biblioteca de software do padrão de comunicação em subestações elétricas IEC 61850 para microcontroladores ESP32 através da criação de uma camada de abstração de hardware da biblioteca, configurando esses dispositivos como protótipos de controladores industriais que possuem algumas funções deste padrão. Essa configuração permite a comunicação padronizada adaptada e uma semântica bem definida dos dados trocados entre esses dispositivos, permitindo sua interação com dispositivos de outros fabricantes que também sejam compatíveis com a IEC 61850, o que não ocorre com dispositivos da indústria de processos. Os dispositivos foram utilizados em uma aplicação semelhante a uma planta industrial para verificar as vantagens e desvantagens de seu uso, que é equivalente a dispositivos tradicionais, como CLPs. Também foi testada a integração dessa norma com a IEC 61499, cuja combinação pode garantir não apenas a padronização da comunicação em rede, mas também do modo de programação dos dispositivos. Ao fim, verificou-se o funcionamento adequado dos microcontroladores integrados, ainda que algumas melhorias sejam necessárias.

Palavras-chave: IEC 61850, ESP32, CLP, IEC 61499.

ABSTRACT

In this work, an integration is presented between a software library of the IEC 61850 communication standard for electrical substations and ESP32 microcontrollers through the creation of a hardware abstraction layer for the library, configuring these devices as prototypes of industrial controllers that implement some functions of this standard. This configuration enables standardized communication with well-defined data semantics, allowing interaction with devices from other manufacturers that are also compatible with IEC 61850, which is not the case with devices from the process industry. The devices were used in an application similar to an industrial plant to evaluate the advantages and disadvantages of their use, which is comparable to traditional devices such as PLCs. Additionally, the integration of this standard with IEC 61499 was tested, as their combination can ensure not only standardized network communication but also standardized device programming. In the end, the integrated microcontrollers were found to function properly, although some improvements are still necessary.

Keywords: IEC 61850, ESP32, PLC, IEC 61499.

LISTA DE FIGURAS

Figura 1 – CLP Siemens SIMATIC S7-1200.	20
Figura 2 – Exemplo de diagrama na linguagem <i>ladder</i>	21
Figura 3 – Comunicação entre camadas no modelo OSI.	22
Figura 4 – Exemplo de funções em uma subestação.	26
Figura 5 – Exemplo de um IED modelado.	27
Figura 6 – Referência de instâncias.	27
Figura 7 – Mapeamento de serviços na norma 61850.	33
Figura 8 – Amostragem de sinal analógico.	34
Figura 9 – Composição de uma APDU.	35
Figura 10 – Transmissão de mensagens GOOSE.	36
Figura 11 – Arquitetura de componentes de servidor da biblioteca.	41
Figura 12 – Raspberry Pi atuando como conversor.	42
Figura 13 – Equipamentos no Laboratório de Armazenamento e Mobilidade (LAM).	43
Figura 14 – Execução de servidor MMS via terminal.	43
Figura 15 – Placa de desenvolvimento WT32-ETH01.	44
Figura 16 – Configurações GOOSE da biblioteca.	45
Figura 17 – Arquitetura de comunicação entre ESP32 e CLP.	46
Figura 18 – Placa de desenvolvimento T-ETH-Lite.	46
Figura 19 – Aeropêndulo em funcionamento.	47
Figura 20 – Malha de controle simplificada.	48
Figura 21 – Arquitetura de rede do sistema distribuído.	48
Figura 22 – Monitoramento do nó lógico TANG via cliente MMS.	51
Figura 23 – Tempo entre pacotes SV.	51
Figura 24 – Monitoramento e ajuste dos nós lógicos FSPT e FPID via cliente MMS.	52
Figura 25 – Monitoramento do nó lógico TPOS via cliente MMS.	52
Figura 26 – Modelo de sistema.	54
Figura 27 – Blocos de função distribuídos em equipamentos.	55
Figura 28 – Bloco de função de publicador via protocolo GOOSE.	56
Figura 29 – Blocos de função de assinatura via protocolo GOOSE.	56

LISTA DE TABELAS

Tabela 1 – Estrutura de Fieldbus 61158 para sistemas de controle industrial. . . .	14
Tabela 2 – Interoperabilidade nativa entre protocolos industriais.	15
Tabela 3 – Relação entre objetos MMS e IEC 61850.	37
Tabela 4 – Resumo de problemas de protocolos legados e como podem ser resolvidos com a IEC 61850.	38

LISTA DE SIGLAS

ACSI	Abstract Communication Service Interface
ADC	Analog-to-Digital Converter
CDC	Common Data Class
CENELEC	European Committee for Electrotechnical Standardization
CLP	Controlador Lógico Programável
DEE	Departamento de Engenharia Elétrica
DNP3	Distributed Network Protocol version 3
GOOSE	Generic Object Oriented Substation Event
HAL	Hardware Abstraction Layer
IEC	International Electrical Commission
IED	Intelligent Electronic Device
IHM	Interface Homem-Máquina
IP	Internet Protocol
ISA	Industrial, Systems and Automation Society
ISO	International Organization for Standardization
LAM	Laboratório de Armazenamento e Mobilidade
LLC	Logical Link Control
MAC	Medium Access Control
MMS	Manufacturing Message Specification
OPC UA	Open Platform Communications Unified Architecture
OSI	Open Systems Interconnection
PI	Proporcional-integral

RAM	Random Access Memory
SCADA	Supervisory Control and Data Acquisition
SCL	System Configuration description Language
SCSM	Specific Communication Service Mapping
SV	Sampled Values
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
XML	eXtensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO	18
1.2	ORGANIZAÇÃO DO TEXTO	18
2	REVISÃO DE CONCEITOS DE REDES INDUSTRIAIS	20
2.1	CLP (CONTROLADOR LÓGICO PROGRAMÁVEL)	20
2.2	MODELO OSI (OPEN SYSTEMS INTERCONNECTION)	21
3	IEC 61850	24
3.1	MODELAGEM	25
3.1.1	SERVIÇOS DE COMUNICAÇÃO ABSTRATA	28
3.2	LINGUAGEM SCL	29
3.2.1	SEÇÃO <i>IED</i> (INTELLIGENT ELECTRONIC DEVICE)	30
3.2.2	SEÇÃO <i>DATATYPETEMPLATES</i>	31
3.2.3	SEÇÃO <i>COMMUNICATION</i>	32
3.3	SERVIÇOS DE MAPEAMENTO E COMUNICAÇÃO	32
3.4	PROTOCOLOS UTILIZADOS DA IEC 61850	33
3.4.1	SAMPLED VALUES	34
3.4.2	GOOSE - GENERIC OBJECT ORIENTED SUBSTATION EVENT	35
3.4.3	MMS - MANUFACTURING MESSAGE SPECIFICATION	36
3.5	AVALIAÇÃO DE PROTOCOLOS INDUSTRIAIS EM RELAÇÃO À NORMA IEC 61850	37
3.6	CONSIDERAÇÕES SOBRE O USO DA IEC 61850 NO AMBIENTE INDUSTRIAL	39
4	ESCOLHA DE BIBLIOTECA DE SOFTWARE	41
4.1	PROVA DE CONCEITO	42
5	CRIAÇÃO DE CAMADA DE ABSTRAÇÃO DE HARDWARE PARA ESP32	44
6	DESENVOLVIMENTO DE PROTÓTIPO DE DISPOSITIVO INDUSTRIAL EM PLANTA DE AEROPÊNDULO	47
7	IEC 61499 E IEC 61850	54
8	CONCLUSÕES	57

9	TRABALHOS FUTUROS	59
	REFERÊNCIAS	61

1 INTRODUÇÃO

A comunicação de dados digitais envolve a transferência de informações de um ponto a outro por meio de estruturas conhecidas como protocolos, que permitem ao receptor da informação saber como ela deve ser decodificada (MACKAY *et al.*, 2004). A partir da ampla utilização de um determinado protocolo ou da criação de protocolos desenvolvidos por entidades representativas de indústrias específicas, são criados padrões de comunicação, simplificando para o consumidor a integração de produtos de diferentes fabricantes (MACKAY *et al.*, 2004).

Na década de 1980, diversas empresas identificaram a necessidade de criar um sistema de comunicação para conectar equipamentos industriais, como sensores, controladores e atuadores, de modo padronizado, pois havia uma crescente complexidade nos sistemas de automação da época (THOMESSE, 2005). Para esse fim, vários projetos de padronização foram iniciados na Europa, como CAN e PROFIBUS na Alemanha, FIP na França e P-NET na Dinamarca, criando protocolos para atender às necessidades de áreas específicas que facilitassem a comunicação industrial, posteriormente conhecidos como protocolos Fieldbus (THOMESSE, 2005). Durante esse período, organizações e empresas competiam para estabelecer sua própria solução Fieldbus como padrão dominante no mercado nacional, deixando os consumidores com diversas opções. Esse processo ficou conhecido como as guerras Fieldbus, e os principais candidatos a se tornarem padrões internacionais pela Comissão Eletrotécnica Internacional (IEC) foram FIP e PROFIBUS (FELSER; SAUTER, 2002). No entanto, ambos adotavam abordagens diferentes: PROFIBUS era baseado em sistemas distribuídos, enquanto FIP seguia um modelo centralizado, mas com um rigoroso esquema de controle em tempo real.

Em 1995, após anos de competição entre FIP e PROFIBUS, empresas norte-americanas entraram na disputa pela padronização internacional com uma nova especificação Fieldbus desenvolvida pela organização Foundation Fieldbus, através da *Industrial, Systems and Automation Society* (ISA), sem ligação com a IEC. Nesse momento, os padrões europeus em competição sob consideração para se tornarem padrão internacional foram unificados em um único padrão europeu pelo *Comité Européen de Normalização Electrotécnica* (CENELEC) (FELSER; SAUTER, 2002). Enquanto o CENELEC discutia a padronização europeia, a Foundation Fieldbus influenciou, em 1996, o processo em andamento na IEC,

resultando em um rascunho que combinava especificações do WorldFIP e da Foundation Fieldbus. No entanto, as contribuições do PROFIBUS foram excluídas, levando a organização PROFIBUS a confrontar a Foundation Fieldbus para impedir a adoção do padrão na IEC, causando uma estagnação no processo de padronização internacional que duraria até 1999. Nesse ano, os representantes das principais especificações assinaram um acordo que resultou em uma norma abrangente para acomodar os diferentes padrões Fieldbus: a IEC 61158, que encerrou as guerras Fieldbus (FELSER; SAUTER, 2002). Diferentemente do CENELEC, a IEC manteve a estrutura original em camadas no rascunho, dividindo-se em protocolos e serviços, conforme mostrado na Tabela 1. Também foram criados os perfis de padrões Fieldbus na IEC 61784, definindo como esses padrões são aplicados em sistemas funcionais.

Tabela 1 – Estrutura de Fieldbus 61158 para sistemas de controle industrial.

Standards part	Contents	Contents and meaning
IEC 61158-1	Introduction	Only Technical Report
IEC 61158-2	PhL: Physical Layer	8 Types of data transmission
IEC 61158-3	DLL: Data Link Layer Services	8 Types
IEC 61158-4	DLL: Data Link Layer Protocols	8 Types
IEC 61158-5	AL: Application Layer Services	10 Types
IEC 61158-6	AL: Application Layer Protocols	10 Types
IEC 61158-7	Network Management	Must be completely revised
IEC 61158-8	Conformance Testing	Work has been cancelled

Fonte: Adaptado de (FELSER; SAUTER, 2002).

Ao longo das últimas três décadas, os protocolos Fieldbus estabelecidos pela IEC 61158 tornaram-se populares e foram amplamente adotados em muitas indústrias, evidenciando a falta de interoperabilidade. Para permitir sua coexistência, foi necessário o desenvolvimento de equipamentos específicos, como conversores e gateways, que possibilitavam a comunicação entre eles (THOMESSE, 2005). Apesar de ainda serem amplamente utilizados, esses protocolos não atendem às novas demandas por flexibilidade e interoperabilidade nas indústrias e não têm perspectivas reais de atualização, perdendo gradativamente espaço para protocolos industriais baseados em Ethernet, que oferecem maior largura de banda e throughput, além de operarem de forma semelhante, utilizando atualizações cíclicas e acesso dividido em intervalos de tempo (SCANZIO; WISNIEWSKI; GAJ, 2021). No entanto, assim como no caso do Fieldbus, foram desenvolvidas várias redes desse tipo que também carecem de interoperabilidade e apresentam poucas diferenças entre si,

dificultando a identificação de grandes vantagens de um protocolo sobre outro entre os mais de 20 padrões existentes (SCANZIO; WISNIEWSKI; GAJ, 2021).

Nos sistemas de automação industrial, a falta de interoperabilidade entre diversos protocolos de rede resulta em custos mais elevados, necessidade de equipamentos adicionais que reduzem a confiabilidade e suscetibilidade a erros na interpretação de atributos, cuja nomeação ou endereçamento não segue uma regra geral. A Tabela 2 mostra as diferentes relações de interoperabilidade entre os principais protocolos de rede de diversos fabricantes da indústria, e também que não existe um protocolo comum a todos os fabricantes.

Tabela 2 – Interoperabilidade nativa entre protocolos industriais.

Protocolo	Interoperabilidade nativa	Fabricantes compatíveis (exemplos)
Modbus TCP	<ul style="list-style-type: none"> • Padrão aberto e amplamente suportado • Comunicação direta via Ethernet 	Schneider, Siemens, Rockwell, Omron, Mitsubishi e Yokogawa
OPC UA	<ul style="list-style-type: none"> • Independente de plataforma e fabricante • Padrão unificado para integração vertical/horizontal 	Siemens, Beckhoff, Honeywell, Rockwell, Microsoft, AWS
EtherNet/IP	<ul style="list-style-type: none"> • Padrão aberto (ODVA) • Certificação ODVA obrigatória 	Rockwell Automation, Omron, Schneider, Phoenix Contact, ABB
PROFINET	<ul style="list-style-type: none"> • Baseado em Ethernet padrão (IEC 61158) • Requer certificação PROFINET 	Siemens, Phoenix Contact, Festo, Bosch Rexroth, WAGO
MQTT	<ul style="list-style-type: none"> • Protocolo aberto e leve para IoT • Comunicação via publish/-subscribe 	IBM, Siemens, AWS, Microsoft, Huawei, Raspberry Pi

Continua na próxima página

Tabela 2 continuação da página anterior

Protocolo	Interoperabilidade nativa	Fabricantes compatíveis (exemplos)
BACnet	<ul style="list-style-type: none"> • Padrão aberto para automação predial (ISO 16484-5) • Certificação BACnet International 	Johnson Controls, Siemens, Schneider, Honeywell, Distech
EtherCAT	<ul style="list-style-type: none"> • Protocolo aberto (EtherCAT Technology Group) • Compatibilidade via ESI 	Beckhoff, Festo, Yaskawa, Omron, Panasonic
HART	<ul style="list-style-type: none"> • Padrão aberto (HART Communication Foundation) • Compatibilidade nativa entre dispositivos 	Emerson, Endress+Hauser, Yokogawa, ABB, Siemens
Protocolos com interoperabilidade limitada (requerem conversores)		
PROFIBUS	<ul style="list-style-type: none"> • Requer certificação PROFIBUS International • Comunicação via RS-485 	Siemens, ABB, Festo (com gateways para redes mistas)
CC-Link	<ul style="list-style-type: none"> • Dominado por fabricantes japoneses • Certificação CLPA necessária 	Mitsubishi, Hitachi, Danfoss (sem conversores apenas se certificados)
DeviceNet	<ul style="list-style-type: none"> • Padrão menos flexível que EtherNet/IP • Conformidade ODVA exigida 	Rockwell, Omron, dispositivos legados

Fonte: Autor.

Já na área dos sistemas elétricos de potência e de forma distinta dos sistemas de automação industrial, foi criada em 2003 a norma IEC 61850, que, além de padronizar a estrutura física e virtual da rede de comunicação, oferece interoperabilidade entre equipamentos compatíveis, agregando significado aos dados trafegados e facilitando a configuração ou reconfiguração desses equipamentos (MACKIEWICZ, 2006). Inicialmente desenvolvida para atuar em subestações, a norma atualmente abrange quaisquer recursos energéticos distribuídos, redes inteligentes e sistemas de recarga de veículos elétricos (AFTAB *et al.*, 2020).

Os elementos *Logical Nodes* (nós lógicos) definidos pela norma correspondem a funções específicas voltadas às necessidades dos sistemas de proteção, geração distribuída, qualidade de energia e automação de subestações, abrangendo desde a supervisão da planta elétrica no barramento de estação até a troca de mensagens com restrição de tempo entre dispositivos no barramento de processo. Esses elementos definem a semântica dos dados, como associar o nome de uma variável à medição de temperatura ou a funções de controle (IEC, 2013a). A linguagem de configuração da norma permite que o ajuste de qualquer IED seja facilmente realizado ou obtido por meio de um arquivo de configuração que contém todas as variáveis e informações de rede.

Um diferencial importante é o baixo custo associado à implementação do tráfego de dados via Ethernet, substituindo cabos de cobre utilizados em dispositivos Fieldbus. Por exemplo, em vez de usar múltiplos transdutores em equipamentos de campo, é possível concentrar os sinais em uma única unidade de interface (*Merging Unit*), reduzindo os custos com transdutores, cabeamento, calibração e manutenção (AFTAB *et al.*, 2020).

Por fim, a norma IEC 61850 é apresentada como uma possível solução para o problema de interoperabilidade da indústria, e a criação de um dispositivo protótipo de comunicação com algumas funcionalidades desta norma será apresentada ao longo do texto.

1.1 OBJETIVO

O objetivo principal deste trabalho é desenvolver um dispositivo de comunicação industrial de baixo custo para a indústria de processos e que seja compatível com a norma IEC 61850, visando garantir a interoperabilidade entre dispositivos em uma rede industrial e possibilitar a execução de aplicações práticas em rede padronizada. Esses dispositivos desempenham funções similares às de um CLP em uma planta industrial, como a coleta de dados de sensores e o acionamento de atuadores em diversas aplicações, que serão detalhadas ao longo do estudo. Além disso, são projetados para viabilizar a conexão de cada elemento da planta industrial à rede, permitindo a descentralização das operações de controle.

Como objetivo secundário, está o desenvolvimento de uma camada de abstração de hardware para microcontroladores com chips ESP32, através de funções específicas do ambiente de desenvolvimento (*ESP-IDF*) de microcontroladores da empresa *ESPRESSIF*, que os fabrica. Esta camada permite que todas as funções da biblioteca sejam reutilizadas em quaisquer microcontroladores ESP32.

1.2 ORGANIZAÇÃO DO TEXTO

- a) No capítulo 2, é feita uma revisão contendo alguns conceitos importantes relacionados às redes industriais.
- b) O capítulo 3 apresenta uma breve introdução da norma IEC 61850, com uma explicação dos principais aspectos e características essenciais que fundamentam a interoperabilidade e seu uso em áreas fora do escopo original da norma.
- c) O capítulo 4 aborda a escolha de uma biblioteca de software para desenvolver o dispositivo protótipo de comunicação, bem como uma prova de conceito.
- d) No capítulo 5 é mostrada a criação de uma camada de abstração de hardware da biblioteca de software para microcontrolador ESP32 juntamente com um teste de comunicação.
- e) No capítulo 6, a biblioteca é embarcada em outros microcontroladores ESP32, para trabalharem em uma aplicação de controle distribuído.

- f) O capítulo 7 trata de uma simples integração entre a biblioteca de software e blocos de função da IEC 61499.
- g) No capítulo 8, são apresentadas as conclusões e trabalhos futuros.

2 REVISÃO DE CONCEITOS DE REDES INDUSTRIAIS

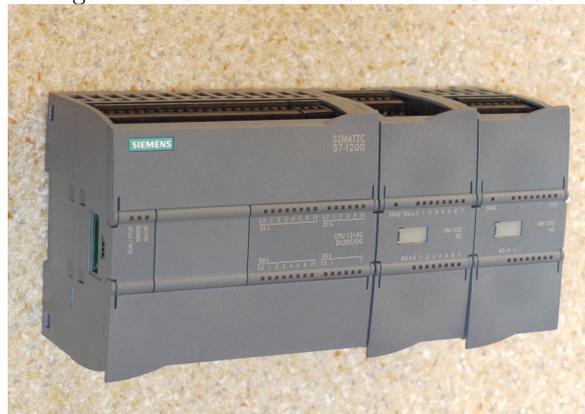
Este capítulo apresenta os conceitos Controlador Lógico Programável e modelo OSI, de forma resumida, que contribuem para o entendimento desta dissertação.

2.1 CLP (CONTROLADOR LÓGICO PROGRAMÁVEL)

Os CLPs são dispositivos eletrônicos semelhantes a computadores, utilizados na indústria para executar funções de controle. Eles representam um avanço em relação à tecnologia anterior baseada em relés eletromecânicos, oferecendo uma redução no número de fios necessários para o controle, maior facilidade de programação, comunicação em rede e alta velocidade na execução de tarefas (PETRUZELLA, 2014). São projetados para operar sob condições adversas, como temperaturas extremas, ruído elétrico, vibração e impacto.

Diversos tipos de variáveis podem ser armazenados na memória de um CLP, como por exemplo valores binários que indicam a ativação de botões, valores inteiros representando a quantidade de produtos fabricados ou valores reais que monitoram a temperatura de um forno industrial. O CLP pode ser configurado para utilizar módulos de entrada e saída, ampliando a quantidade de dispositivos ou variáveis que se comunicam diretamente com ele. A Figura 1 ilustra o modelo SIMATIC S7-1200 da Siemens, um dos CLPs mais comercializados atualmente, acompanhado de dois módulos de entrada e saída.

Figura 1 – CLP Siemens SIMATIC S7-1200.



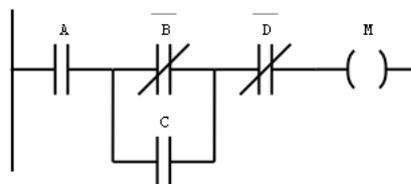
Fonte: (ULRICHAAB, 2024).

A programação de um CLP pode ser realizada por meio da linguagem *ladder*, uma das cinco linguagens normatizadas pela IEC 61131-3 (IEC, 2013c) e a mais popular.

Originalmente desenvolvida para relés, essa linguagem oferece uma interpretação visual simples de seu funcionamento.

Como mostrado na Figura 2, quatro variáveis de entrada (A, B, C e D) e uma bobina de saída (M) são analisadas. As variáveis A e C são representadas por contatos normalmente abertos (sem corrente elétrica de controle, os contatos permanecem abertos), enquanto B e D são representadas por contatos normalmente fechados (sem corrente elétrica, os contatos permanecem fechados). A sequência lógica define que, se a condição de A for verdadeira, seguida por B ou C, e então D, a saída M será ativada.

Figura 2 – Exemplo de diagrama na linguagem *ladder*.



Fonte: (GÁLVEZ, 2024).

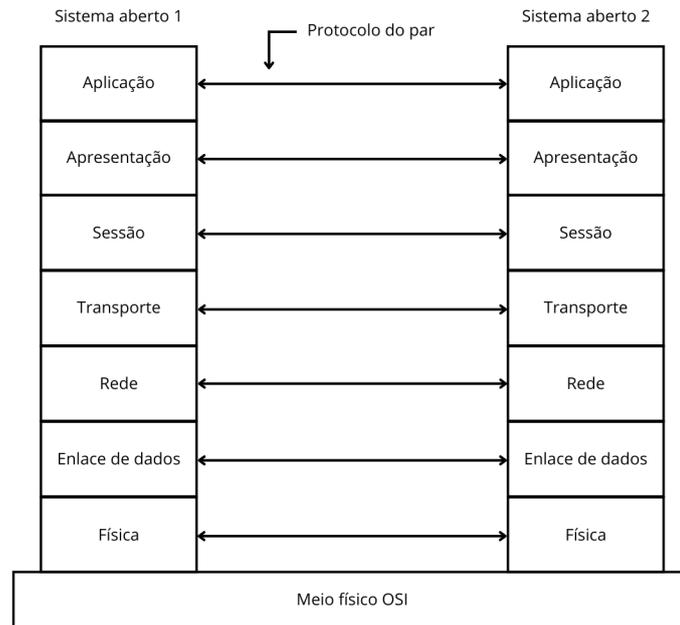
Com o desenvolvimento das redes de computadores, os CLPs e outros equipamentos da indústria foram adaptados com diversos protocolos de comunicação para garantir a entrega e o recebimento de dados sob condições específicas, atendendo a requisitos de alta velocidade e confiabilidade das informações trocadas. Essa comunicação em rede permite que os CLPs realizem controles em conjunto com outros CLPs para alcançarem objetivos comuns, algo frequente em indústrias de processos.

Os protocolos de redes industriais padronizados seguem uma estrutura comum baseada no modelo OSI (Open Systems Interconnection), desenvolvido para organizar o processamento de dados em camadas, conforme o nível de abstração da informação em redes de computadores.

2.2 MODELO OSI (OPEN SYSTEMS INTERCONNECTION)

O modelo OSI foi criado pela Organização Internacional para Padronização (ISO) para permitir a comunicação entre diferentes computadores em uma mesma rede (ISO/IEC, 1994). Ele é dividido em sete camadas de abstração, onde camadas de mesmo nível se comunicam em diferentes dispositivos. As informações das camadas superiores são repassadas para as inferiores, seguem para o meio físico e retornam à camada correspondente no dispositivo de destino (Figura 3).

Figura 3 – Comunicação entre camadas no modelo OSI.



Fonte: Autor.

Cada camada é independente das outras e responsável por funções específicas, conforme descrito a seguir (ZURAWSKI, 2005):

1. **Camada física:** Define os meios mecânicos, elétricos, funcionais e procedurais para ativar, manter e desativar conexões físicas. Transmite e recebe os bits de dados que são interpretados pelo computador e associados às propriedades do meio, como eletricidade ou luz.
2. **Camada de enlace de dados:** Responsável por detectar e corrigir erros da camada física, bem como pelo controle do fluxo de mensagens recebidas e enviadas. É subdividida em duas subcamadas: MAC (*Medium Access Control*), que faz a interface com a camada física, e LLC (*Logical Link Control*), que gerencia o fluxo de dados e realiza controle de erros.
3. **Camada de rede:** Estabelece, mantém e finaliza conexões de rede entre computadores e realiza a troca de pacotes de dados com a camada de transporte. O protocolo IP (*Internet Protocol*) opera nesta camada.
4. **Camada de transporte:** Otimiza o uso dos serviços de rede, ajustando o desempenho requerido por cada sessão a um custo mínimo. Protocolos como TLS (*Transport Layer Security*), TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*) operam nesta camada.

5. **Camada de sessão:** Estabelece, organiza e sincroniza conexões entre processos da camada de apresentação para prover uma troca ordenada de dados.
6. **Camada de apresentação:** Fornece a representação adequada dos dados usados na camada de aplicação, tornando-os independentes da sintaxe subjacente.
7. **Camada de aplicação:** Camada de nível mais alto, responsável por permitir o acesso de programas do usuário à rede através de diversos serviços.

As camadas 1 e 2 atuam diretamente no meio físico e suas funcionalidades são projetadas de acordo com as propriedades desse meio, como largura de banda e controle de erros. São as camadas mais importantes para as redes industriais, pois oferecem os requisitos para comunicação em tempo real e confiabilidade dos dados trafegados entre dispositivos vizinhos no espaço físico. As camadas restantes lidam com a comunicação fim a fim, entre dispositivos de diferentes redes (ZURAWSKI, 2005).

3 IEC 61850

A norma IEC 61850 foi inicialmente concebida para estabelecer uma infraestrutura padrão de comunicação entre dispositivos inteligentes em subestações. Posteriormente, seu escopo foi ampliado para incluir quaisquer recursos energéticos distribuídos. Desde sua concepção, foi considerada uma futura referência na área (MCDONALD, 2003), devido principalmente à modelagem dos dados e à estrutura dos protocolos de comunicação utilizados. Antes da norma, as integrações de dados nas subestações ocorriam por meio dos protocolos da IEC 60870 e DNP3 (Distributed Network Protocol version 3). O DNP3, assim como a IEC 61850, foi desenvolvido para subestações (GROUP, 2024) e apresenta categorias de variáveis e arquitetura mestre-escravo similares as de MODBUS, como entradas binárias, analógicas e de contador, além de saídas de controle e analógicas, além do acréscimo de estampas de tempo nos dados comunicados e tipos de dados mais comuns em subestações. Porém, DNP3 não possui as mesmas definições semânticas e requisitos de tempo de comunicação da IEC 61850.

Dado o caráter extenso e detalhado da norma, serão abordadas apenas as partes mais relevantes relacionadas às subestações e aos sistemas de potência. O foco recairá nos principais aspectos do escopo da norma, especialmente na integração de três conjuntos essenciais que asseguram a interoperabilidade entre dispositivos eletrônicos inteligentes (IEDs) (IEC, 2013a). Cada um desses conjuntos será detalhado em seções específicas ao longo deste trabalho.

- Modelagem dos dados, incluindo a nomenclatura padrão dos *Logical Nodes* (nós lógicos) e a padronização dos objetos de dados e seus atributos. Essa modelagem define a informação trocada entre dispositivos. Cada *Logical Node* possui um nome de quatro letras (geralmente as iniciais da sua função), representa um tipo de função e contém atributos específicos, detalhados adiante. Também serão descritos os serviços abstratos usados na norma, como GOOSE e SV.
- A linguagem padrão SCL (*System Configuration description Language*), usada para descrever a estrutura e a hierarquia dos elementos de dados, bem como a sintaxe para frases e textos em nível de máquina. Baseada no padrão XML, essa linguagem descreve as capacidades de cada IED, suas configurações e o ambiente completo de um sistema, abrangendo desde parâmetros de comunicação até a topologia elétrica.

- Um conjunto de serviços de comunicação para troca de informações em tempo real, projetado para ser adaptável sem a necessidade de modificar as especificações da norma e independente do meio ou protocolo de comunicação utilizado.

3.1 MODELAGEM

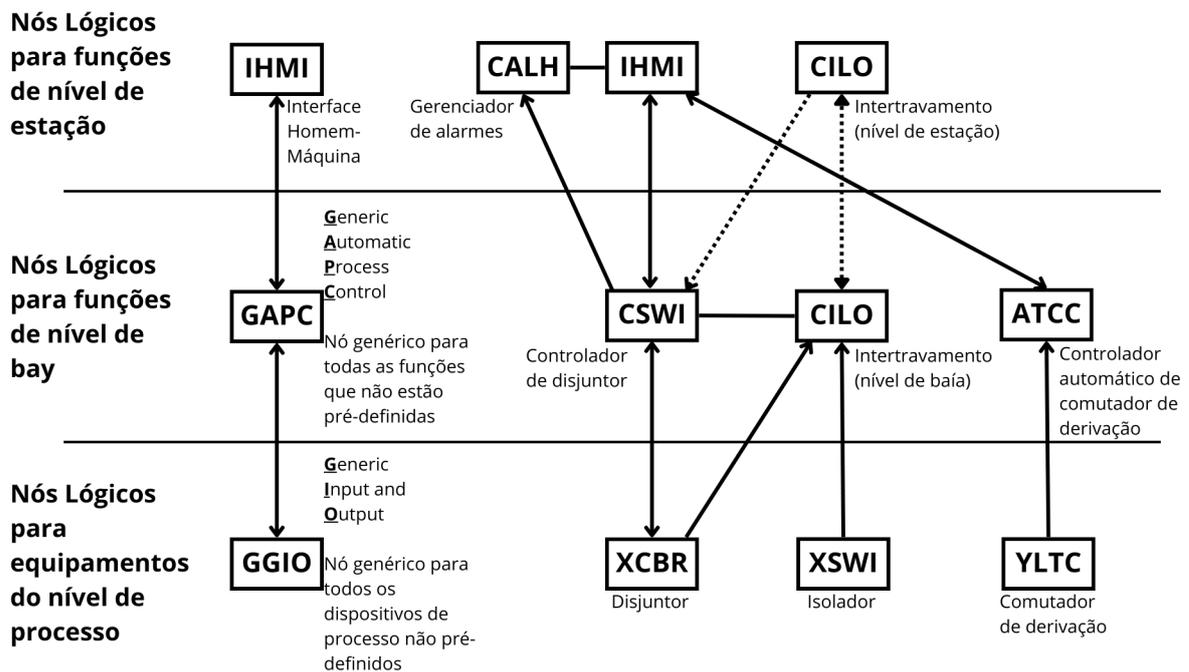
Criada para subestações e sistemas de potência, a norma define as funções mais adequadas para esses ambientes e estabelece requisitos rigorosos de tempo para garantir a confiabilidade das operações. As funções são divididas em seis categorias (IEC, 2013b):

1. **Funções de suporte do sistema:** Gerenciam a comunicação em rede, sincronização de tempo e autodiagnóstico de dispositivos físicos, garantindo que todos os sistemas funcionem corretamente.
2. **Funções de configuração do sistema e manutenção:** Usadas para modificar objetos no ambiente da norma, como adicionar, remover ou atualizar dados de dispositivos. Esta categoria não exige tempos de resposta rápidos, geralmente medindo-se em segundos, pois é executada fora da operação normal.
3. **Funções de controle e operações:** Necessárias para o funcionamento regular da subestação, incluindo dispositivos de interface homem-máquina (IHMs), que permitem aos operadores controlar processos e acessar parâmetros, alarmes de distúrbios ou faltas. O tempo de resposta dessas funções é similar ao da categoria anterior.
4. **Funções de automação de processos em bay:** *bay* refere-se a uma subestrutura local de uma subestação, geralmente onde se encontram transformadores de corrente ou tensão. As ações, como comandos, *trips* e *releases*, ocorrem dentro de um ambiente *bay*. Essas funções operam automaticamente, sem intervenção de um operador, e incluem proteções contra sobrecorrente, subtensão, intertravamentos e medições, com tempos de resposta na faixa de milissegundos.
5. **Funções de automação de processos distribuídos:** Garantem a segurança de pessoas e equipamentos. São associadas a ações que usam dados de uma bay, mas são executadas em várias bays ou distribuídas em mais de um IED, como proteção de barramento e controle de potências ativa e reativa. Essas funções são automatizadas e têm tempos de resposta rápidos.

6. **Funções genéricas:** Não pertencem a categorias específicas e podem ser aplicadas em outras áreas.

O espaço físico de uma subestação é dividido em três níveis: estação (onde estão os bancos de dados, interfaces de comunicação remota e terminais do operador), bay (monitoramento, proteção e controle) e processo (sensores e atuadores). A Figura 4 ilustra um exemplo de aplicação de nós lógicos em diferentes níveis.

Figura 4 – Exemplo de funções em uma subestação.

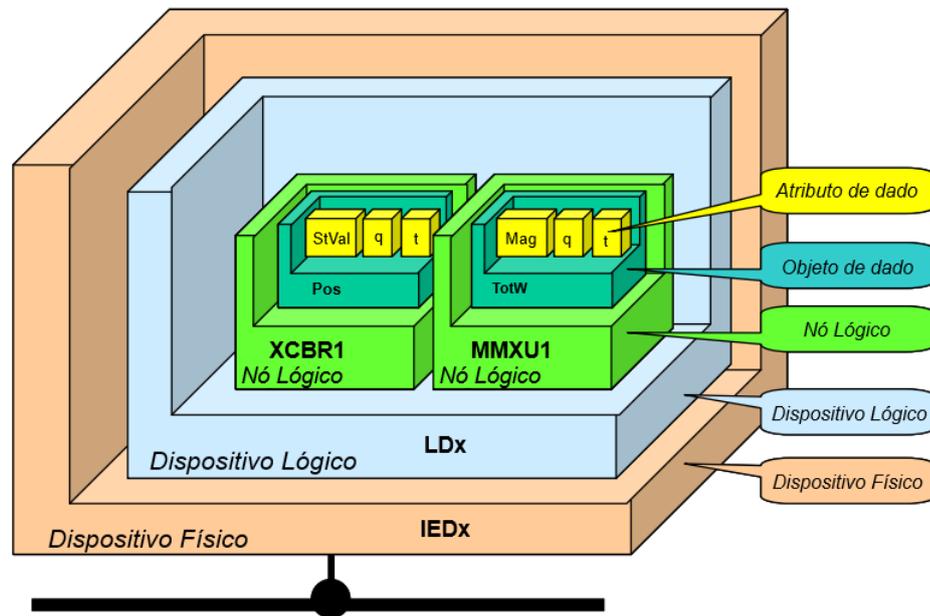


Fonte: Autor.

A modelagem de dados é orientada a objetos e descrita na parte 7 da norma (IEC 61850-7). Ela separa dispositivos físicos (IEDs) em dispositivos lógicos, que agrupam funções específicas e não podem ser distribuídos entre diferentes IEDs. Cada dispositivo lógico contém pontos de acesso de comunicação, serviços e informações sobre os dispositivos controlados ou o dispositivo físico que o hospeda, operando de forma independente dentro do mesmo IED (IEC, 2013a).

Os dispositivos lógicos são compostos por nós lógicos, fragmentando funções em partes menores para a troca de informações. Cada nó lógico está associado a classes de dados comuns (CDC) contendo atributos predefinidos. A Figura 5 mostra um exemplo de IED com dois nós lógicos, seus objetos de dados e atributos. Esses atributos têm restrições funcionais que indicam estados, valores medidos ou parâmetros de configuração.

Figura 5 – Exemplo de um IED modelado.

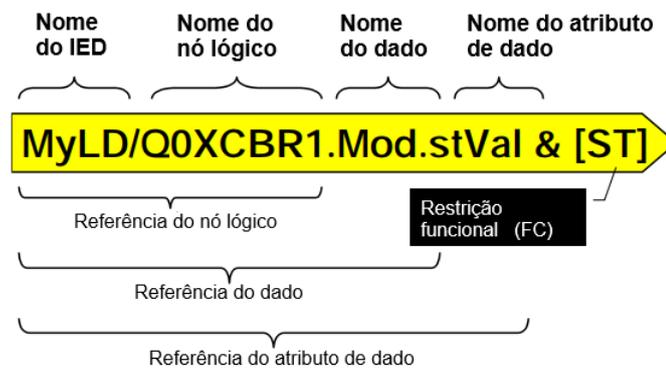


Fonte: Adaptado de (IEC, 2013a).

A estrutura predefinida desses dados estabelece uma semântica para a comunicação, garantindo que cada função tenha um único nó lógico correspondente. Isso assegura que a comunicação seja padronizada mesmo entre dispositivos de diferentes fabricantes, facilitando a identificação de funções na rede por referência textual a objetos específicos.

Nomes padronizados podem ser combinados com nomenclaturas definidas pelo fabricante, sem comprometer a semântica. A Figura 6 exemplifica a referência ao valor de estado do dado “Mod” de um nó lógico disjuntor (XCBR), onde o nome do IED (MyLD) e o prefixo do nó lógico (Q0) são definidos pelo fabricante, enquanto o sufixo (1) representa o número da instância. Mais de um nó pode ser criado, herdando os mesmos objetos de dados e atributos da sua classe.

Figura 6 – Referência de instâncias.



Fonte: Adaptado de (IEC, 2011a).

3.1.1 SERVIÇOS DE COMUNICAÇÃO ABSTRATA

Os modelos de serviços de comunicação entre IEDs são definidos pelas classes GSE (*Generic Substation Event*) e *Transmission of Sampled Value*, descritas na parte 7-2 da norma (IEC, 2010). Esta seção refere-se a esses modelos como Interface de Serviço de Comunicação Abstrata (ACSI), cujas classes são mapeadas para diferentes protocolos por meio do Mapeamento Específico de Serviços de Comunicação (SCSM), abordado na seção 3.3.

- **Generic Substation Event Class Model (GSE):** Fornece uma distribuição rápida e confiável dos dados de entrada e saída agrupados em conjuntos, também modelados nesta parte da norma. Este modelo descreve as mensagens GOOSE e GSSE (esta última descontinuada pela norma) e baseia-se no conceito de descentralização autônoma, permitindo que a mesma informação seja entregue a mais de um IED por meio de serviço multicast ou broadcast. A comunicação segue o mecanismo publicador/assinante, no qual o publicador armazena os dados em um buffer local para envio, e o serviço de mapeamento transmite essas informações aos assinantes, atualizando seus respectivos buffers de recepção. A mensagem GOOSE contém informações que permitem ao dispositivo assinante identificar a alteração de um dado e o momento da última modificação.
- **Transmission of Sampled Value Class Model:** A transmissão de valores amostrados é modelada com controle de tempo para minimizar o jitter (variação de atraso médio de pacotes na rede) durante o processo de amostragem e transmissão, garantindo uma sequência única e precisa de amostras. Assim como o modelo GSE, existem buffers locais no publicador e no assinante para armazenamento antes do envio e durante a recepção. Os dados são transmitidos em grupos, utilizando a classe de dados comum SAV (*Sampled Analogue Values*), específica para valores amostrados. Este modelo utiliza os modos de comunicação multicast e unicast.

Além dos padrões de dados e dispositivos já definidos pela norma, é possível criar novos nós, objetos de dados e atributos conforme necessário. Isso permite adicionar conteúdo padronizado mais adequado para o setor industrial caso a norma seja adotada nesse ambiente, já que seu conteúdo atual não abrange todos os tipos de equipamentos e dados exigidos em processos industriais.

3.2 LINGUAGEM SCL

O processo de engenharia de um sistema geralmente ocorre antes de sua disponibilidade física. Portanto, as capacidades dos dispositivos e suas configurações de projeto devem estar acessíveis de forma padronizada para permitir a configuração por ferramentas de diferentes fabricantes, garantindo interoperabilidade. A parte 6 da norma (IEC 61850-6) define a linguagem SCL (System Configuration description Language) para configurar e descrever as funcionalidades de um ambiente, as capacidades de um IED e a automação do sistema elétrico (IEC, 2013a).

A linguagem é baseada no padrão XML, e a norma especifica a estrutura de diversos arquivos desse tipo por meio de *schemas* XML com extensão .xsd. O arquivo que contém toda a configuração do sistema tem a extensão .scl, representado pelo SCL.xsd, composto pelo elemento de cabeçalho (*Header*) e por pelo menos um dos seguintes elementos: *Substation*, *Communication*, *IED* e *Data Type Templates*. A Listing 3.1 mostra o padrão de arquivo .scl definido pela norma.

```
<xs:element name="SCL">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="tBaseElement">
        <xs:sequence>
          <xs:element name="Header" type="tHeader">
            <xs:unique name="uniqueHitem">
              <xs:selector xpath="/scl:History/scl:Hitem"/>
              <xs:field xpath="@version"/>
              <xs:field xpath="@revision"/>
            </xs:unique>
          </xs:element>
          <xs:element ref="Substation" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Communication" minOccurs="0"/>
          <xs:element ref="IED" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="DataTypeTemplates" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="version" type="tScIVersion" use="required" fixed="2007"/>
        <xs:attribute name="revision" type="tScIRevision" use="required" fixed="A"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Listing 3.1 – Estrutura XML padrão de arquivo .scl. Fonte: adaptado de (IEC, 2009).

Nesta dissertação, será abordada apenas partes relacionadas ao dispositivo físico do ambiente da norma (IED) e sua programação, equivalente a um CLP. Por isso, serão apresentadas apenas a estrutura de uma seção IED (*IED*), a modelagem dos dados (*DataTypeTemplates*) e a comunicação (*Communication*), que podem ser representadas por um arquivo com extensão .icd para descrever as capacidades ou com extensão .iid para incluir configurações específicas de projeto, além das capacidades. As estruturas nas figuras Listing 3.2, Listing 3.3 e Listing 3.4 mostram arquivos XML relacionados, respectivamente, às seções *IED*, *DataTypeTemplates* e *Communication* da Listing 3.1, cujos elementos serão descritos a seguir.

3.2.1 SEÇÃO *IED* (INTELLIGENT ELECTRONIC DEVICE)

```
<xs:complexType name="tIED">
  <xs:complexContent>
    <xs:extension base="tNaming">
      <xs:sequence>
        <xs:element name="Services" type="tServices" minOccurs="0"/>
        <xs:element name="AccessPoint" type="tAccessPoint" maxOccurs="unbounded">
          <xs:unique name="uniqueLNInAccessPoint">
            <xs:selector xpath="/scl:LN"/>
            <xs:field xpath="@inst"/>
            <xs:field xpath="@lnClass"/>
            <xs:field xpath="@prefix"/>
          </xs:unique>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="type" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="manufacturer" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="configVersion" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="originalSclVersion" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="originalSclRevision" type="xs:normalizedString" use="optional"/>
      <xs:attribute name="engRight" type="tRightEnum" use="optional" default="full"/>
      <xs:attribute name="owner" type="xs:normalizedString" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Listing 3.2 – Estrutura XML padrão da seção *IED*. Fonte: adaptado de (IEC, 2009).

- **Services:** Define os tipos de serviços fornecidos pelo dispositivo, como leitura e escrita de variáveis ou conjuntos de dados, criação de relatórios e presença de protocolos de comunicação.

- **AccessPoint:** Cada ponto de acesso possui um elemento de servidor *Server*, que gerencia a autenticação necessária e contém todos os dispositivos lógicos, bem como seus nós lógicos e atributos acessíveis na rede. Esse elemento também descreve relatórios e seus conjuntos de dados.

3.2.2 SEÇÃO *DATATYPETEMPLATES*

```

<xs:complexType name="tDataTypeTemplates">
  <xs:sequence>
    <xs:element name="LNodeType" type="tLNodeType" maxOccurs="unbounded">
      <xs:unique name="uniqueDOInLNodeType">
        <xs:selector xpath="scl:DO"/>
        <xs:field xpath="@name"/>
      </xs:unique>
    </xs:element>
    <xs:element name="DOType" type="tDOType" maxOccurs="unbounded">
      <xs:unique name="uniqueDAorSDOInLDOType">
        <xs:selector xpath="./*/>
        <xs:field xpath="@name"/>
      </xs:unique>
    </xs:element>
    <xs:element name="DAType" type="tDAType" minOccurs="0" maxOccurs="unbounded">
      <xs:unique name="uniqueBDAInLDAType">
        <xs:selector xpath="scl:BDA"/>
        <xs:field xpath="@name"/>
      </xs:unique>
    </xs:element>
    <xs:element name="EnumType" type="tEnumType" minOccurs="0" maxOccurs="unbounded">
      <xs:unique name="uniqueOrdInEnumType">
        <xs:selector xpath="scl:EnumVal"/>
        <xs:field xpath="@ord"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Listing 3.3 – Estrutura XML padrão da seção *DataTypeTemplates*. Fonte: adaptado de (IEC, 2009).

- **LNodeType:** Contém os nomes e as referências dos objetos de dados associados a um nó lógico específico, além de sua classe.
- **DOType:** Identifica o conteúdo de cada objeto de dado, incluindo o nome, a classe de dados comum (CDC), a restrição funcional e as referências dos atributos de dado para o objeto especificado.

- **DAType:** Define o tipo básico de cada atributo de dado, como inteiro, ponto flutuante ou binário.
- **EnumType:** Representa dados definidos por estados discretos. Por exemplo, o objeto *Health* pode assumir os estados “Ok”, “Warning” e “Alarm”. As enumerações associam um número inteiro a cada estado textual.

3.2.3 SEÇÃO *COMMUNICATION*

```
<xs:complexType name="tCommunication">
  <xs:complexContent>
    <xs:extension base="tUnNaming">
      <xs:sequence>
        <xs:element name="SubNetwork" type="tSubNetwork" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Listing 3.4 – Estrutura XML padrão da seção *Communication*. Fonte: adaptado de (IEC, 2009).

- **Subnetwork:** Descreve um ou mais pontos de acesso conectados ao dispositivo e seus parâmetros de comunicação, como configuração de VLAN para protocolos GOOSE e SAMPLED VALUES ou definições de IP para o protocolo MMS.

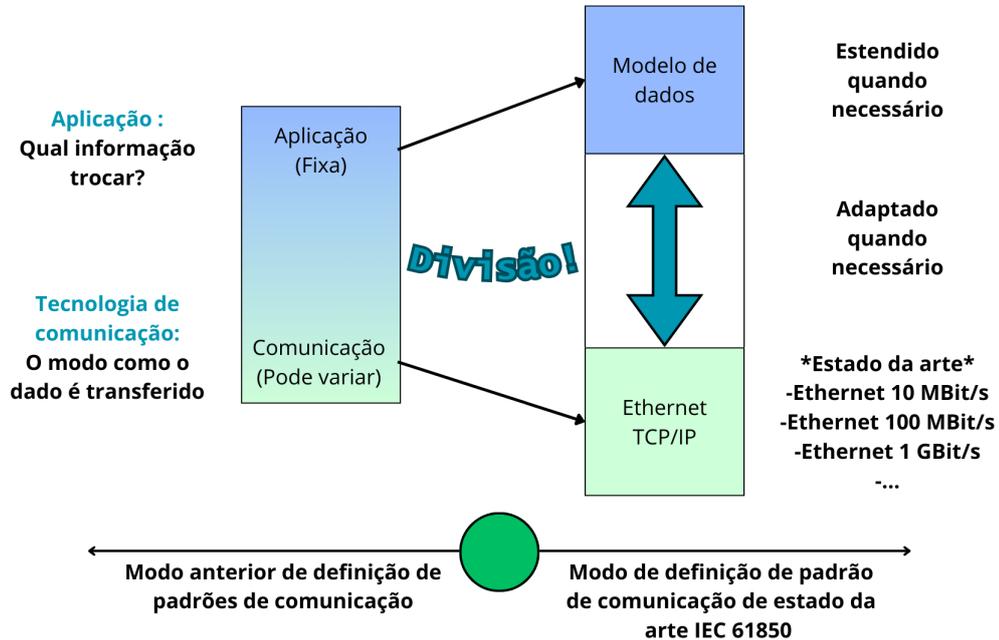
Com os elementos das seções apresentadas, é possível criar um arquivo XML que pode ser embarcado em qualquer IED, contendo todos os objetos, dados e configurações de rede. Também é possível obter o arquivo diretamente de um IED presente em uma rede, pois a norma garante a auto-descrição de cada dispositivo, permitindo acesso completo ao seu conteúdo.

3.3 SERVIÇOS DE MAPEAMENTO E COMUNICAÇÃO

Com o objetivo de ser independente das tecnologias utilizadas, a norma introduz uma separação entre aplicação e comunicação por meio de camadas de abstração (IEC, 2013a). Isso elimina a necessidade de reescrever a norma ao surgir um novo padrão ou meio de comunicação, bastando substituir a parte referente ao padrão anterior, conforme ilustrado na Figura 7. Os mapeamentos específicos de serviços de comunicação (SCSM)

fazem a ponte entre a Interface de Serviços de Comunicação Abstrata (ACSI) e o meio de comunicação atualmente utilizado.

Figura 7 – Mapeamento de serviços na norma 61850.



Fonte: Autor.

3.4 PROTOCOLOS UTILIZADOS DA IEC 61850

A norma especifica quatro protocolos de comunicação, cada um para um tipo de comunicação em subestações:

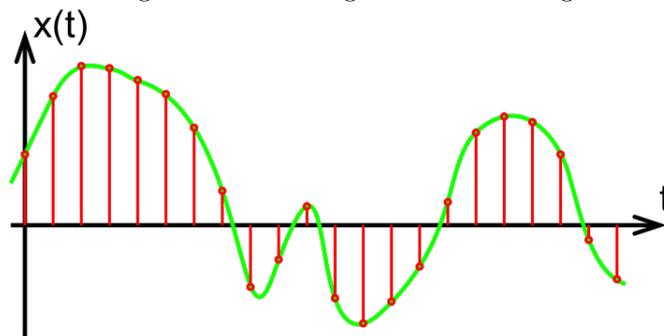
- **SAMPLED VALUES:** Transmissão contínua de dados amostrados de sensores, como transformadores de corrente ou tensão, para IEDs.
- **GOOSE:** Envio de eventos entre IEDs, como comandos de *trip* em disjuntores para isolar falhas.
- **SNTP:** Sincronização do relógio dos dispositivos na rede, baseado no protocolo UDP.
- **MMS:** Supervisão de dados e serviços de outros protocolos em IEDs. É usado principalmente em sistemas de supervisão em níveis superiores aos IEDs.

Dos quatro, esta dissertação não abordará o protocolo de sincronização de tempo, pois seu propósito é garantir precisão temporal para proteções ou medições síncronas, como as de fasores. O foco aqui é o funcionamento básico de um dispositivo conforme a norma em um ambiente de real de laboratório.

3.4.1 SAMPLED VALUES

As medições de grandezas analógicas são realizadas por conversores analógico-digitais, que amostram valores analógicos em intervalos específicos durante um período determinado. A Figura 8 ilustra um sinal analógico representado por uma linha verde, com pontos vermelhos indicando os valores amostrados em instantes específicos. Esses valores são então processados para calcular um valor médio amostrado, frequentemente obtido por meio da média quadrática.

Figura 8 – Amostragem de sinal analógico.

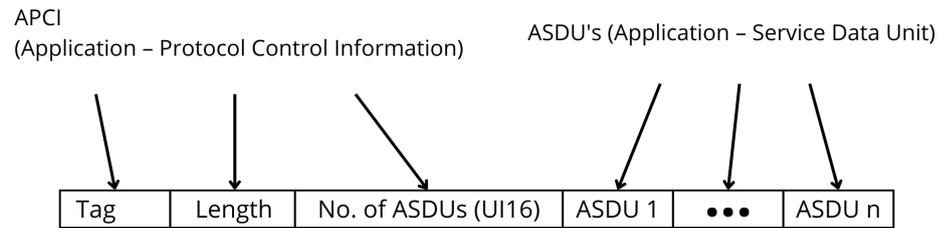


Fonte: (LUKMAK, 2024).

O mapeamento desse protocolo para Ethernet é definido na parte 9-2 da norma (IEC, 2011c). Cada valor amostrado é encaminhado para uma unidade de dados do serviço de aplicação (ASDU). A ASDU representa o valor amostrado juntamente com sua estampa de tempo. Um conjunto de ASDUs, combinado com o controle de informação do protocolo de aplicação (APCI), forma uma unidade de dados do protocolo de aplicação (APDU), que contém um ou mais valores amostrados a serem transmitidos em um frame Ethernet no modo multicast. A Figura 9 ilustra a composição de uma APDU, onde o campo *Tag* indica o tipo de dado, *Length* define o comprimento dos dados subsequentes e o número de ASDUs é representado por um inteiro sem sinal de 16 bits, seguido pelos valores de cada ASDU na ordem definida. O tempo de atraso máximo de uma APDU na rede varia geralmente entre 3 e 10 ms (IEC, 2013b).

Como as mensagens desse protocolo utilizam apenas a camada 2 do modelo OSI, a norma também especifica valores de atributos como faixa de endereços MAC multicast, ID e prioridade de VLAN, e ID de aplicação, usados para identificar o protocolo na rede.

Figura 9 – Composição de uma APDU.



Fonte: Autor.

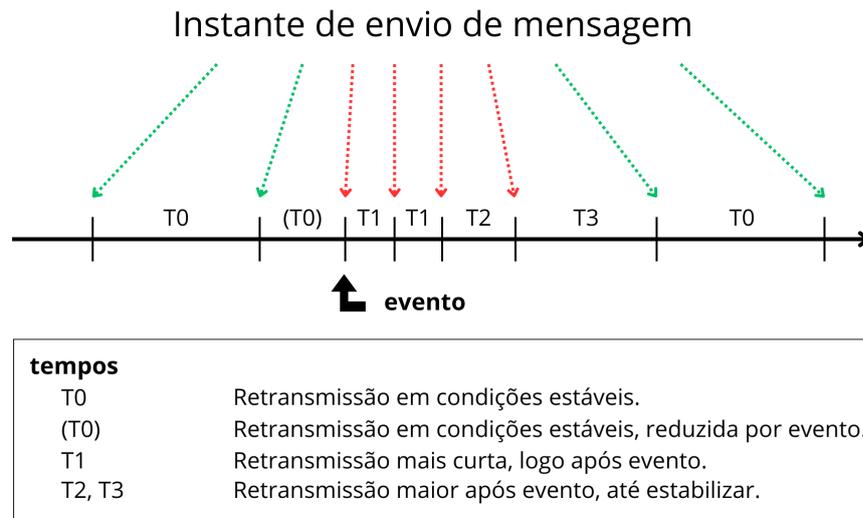
3.4.2 GOOSE - GENERIC OBJECT ORIENTED SUBSTATION EVENT

Em uma subestação, mensagens de eventos críticos, como *trip*, são utilizadas para executar funções de proteção que devem ocorrer em um intervalo de tempo muito curto após a detecção de uma falha, especialmente para isolar faltas. Para esse propósito, utilizam-se mensagens GOOSE, que contêm um ou mais dados provenientes de diferentes nós lógicos, organizados em um conjunto de dados específico do publicador e assinante, além de um número de sequência e uma estampa de tempo que ordenam a transmissão.

O mapeamento do protocolo GOOSE para Ethernet é especificado na parte 8-1 da norma (IEC, 2011b), e seu funcionamento é ilustrado na Figura 10. Durante a normalidade (quando não há variação nos dados do conjunto), são enviadas mensagens periódicas (setas verdes apresentadas na figura) para indicar ao assinante que o publicador está operando corretamente. Quando um dado é alterado, ocorre um evento (seta vermelha à esquerda) e a nova mensagem é enviada com os dados atualizados. Como não há uma forma "rápida" de garantir que o assinante recebeu a mensagem crítica (confirmar a recepção exigiria um protocolo de camada 4, como o TCP, que comprometeria a velocidade), a mensagem é retransmitida várias vezes em intervalos de tempo menores do que os das mensagens periódicas. Cada intervalo de retransmissão aumenta progressivamente para oferecer uma margem maior de recepção. Quando o evento termina, as mensagens periódicas são retomadas com o novo conjunto de dados.

Assim como no protocolo SAMPLED VALUES, as mensagens GOOSE operam exclusivamente na camada 2 do modelo OSI, utilizando tipos de atributos de configuração semelhantes, mas com diferentes valores padronizados.

Figura 10 – Transmissão de mensagens GOOSE.



Fonte: Autor.

3.4.3 MMS - MANUFACTURING MESSAGE SPECIFICATION

Para a troca de dados sem restrições de tempo crítico, a norma utiliza o protocolo MMS (Manufacturing Message Specification), desenvolvido pela General Motors para comunicação em ambientes de manufatura automatizada (SISCO, 1995) e posteriormente aprimorado pela Boeing. Esse protocolo permite a troca de mensagens entre sistemas de supervisão e IEDs, onde não é essencial que a informação seja transmitida com a mesma rapidez das comunicações entre IEDs ou entre IEDs e outros dispositivos de campo. O MMS também suporta a transferência de arquivos entre cliente e servidor, possibilitando a leitura e escrita de arquivos SCL.

O mapeamento do MMS é feito para TCP/IP, garantindo a manutenção de sessões cliente/servidor e a integridade das informações, conforme o controle de erros implementado pelo TCP.

A Tabela 3 apresenta os objetos e serviços MMS utilizados no mapeamento para os objetos definidos pela norma.

Tabela 3 – Relação entre objetos MMS e IEC 61850.

Objeto MMS	Objeto IEC 61850	Serviços MMS em uso
Application Process VMD	Server	Initiate Conclude Abort Reject Cancel Identify
Named Variable Objects	Logical Nodes and Data	Read Write InformationReport GetVariableAccessAttribute GetNameList
Named Variable List Objects	Data Sets	GetNamedVariableListAttributes GetNameList DefineNamedVariableList DeleteNamedVariableList Read Write InformationReport
Journal Objects	Logs	ReadJournal InitializeJournal GetNameList
Domain Objects	Logical Devices	GetNameList GetDomainAttributes StoreDomainContents
Files	Files	FileOpen FileRead ObtainFile FileClose FileDirectory FileDelete

Fonte: Adaptado de (IEC, 2011b).

3.5 AVALIAÇÃO DE PROTOCOLOS INDUSTRIAIS EM RELAÇÃO À NORMA IEC 61850

Em (KUMAR *et al.*, 2023), é feita uma revisão sobre os protocolos convencionais utilizados em subestações para avaliar mudanças decorrentes da adoção da norma IEC 61850. O artigo destaca problemas de expansão, atualização de versão e transferência de dados relacionados a mudanças no software, bem como à disponibilidade, acurácia e confiabilidade das informações ao usar os protocolos mencionados — dificuldades que não ocorrem com a IEC 61850. A Tabela 4 resume os principais problemas dos protocolos legados e como a norma os resolve.

Tabela 4 – Resumo de problemas de protocolos legados e como podem ser resolvidos com a IEC 61850.

Problemas de protocolos legados	IEC 61850
Todos os protocolos legados possuem software de comunicação proprietário que não interage bem com dispositivos similares de outros fabricantes, sendo necessária a realização de modificações ou <i>retrofitting</i> para interagir.	Além de possui ferramentas de softwares interoperáveis capazes de interagir com dispositivos semelhantes, a norma permite o uso de topologias estrela ou anel presentes em diversos protocolos.
A capacidade de manusear IEDs é limitada, pois muitos IEDs legados possuem funções particulares e não transferíveis, geralmente com interface serial.	A norma permite a comunicação com IEDs de diversos fabricantes e compatibilidade com dispositivos baseados em microprocessadores.
Os mais populares funcionam no modo mestre-escravo, enviando sinais apenas para um relé específico.	A norma transmite dados para IEDs conectados em rede Ethernet. Pacotes GOOSE e SV só passam por IEDs assinantes. A sincronização de tempo é um recurso fundamental.
Automação completa é difícil de alcançar com protocolos legados, que são baseados em comunicação serial ponto a ponto, e não em redes.	Todas as funções dos IEDs de uma subestação digital podem ser integradas e comunicadas em rede.
Não oferecem suporte adequado para identificar falhas complexas na rede.	A IEC 61850 permite coleta de dados eficiente e detecção rápida de falhas.

Fonte: Adaptado de (KUMAR *et al.*, 2023).

O estudo conclui que a adoção dessa norma traz vantagens como a redução de custos com a substituição de cabos de cobre por cabos de rede, menor esforço de engenharia e maior interoperabilidade. No entanto, apresenta também desvantagens, como o risco de ataques cibernéticos externos que podem comprometer a subestação, caso a rede IEC 61850 possa ser acessada remotamente.

Em (HORALEK; MATYSKA; SOBESLAV, 2013), são discutidas as vantagens e desvantagens dos protocolos mais utilizados em subestações, destacando-se a falta de interoperabilidade com novas tecnologias no IEC 60870-5-101 e as melhorias introduzidas pelo DNP3.

Além disso, o artigo apresenta a evolução da IEC 61850 em relação a esses protocolos, detalhando suas principais funcionalidades e apontando a necessidade de aprimorar a infraestrutura de rede para corrigir vulnerabilidades a ataques remotos e atender aos requisitos de velocidade e confiabilidade.

3.6 CONSIDERAÇÕES SOBRE O USO DA IEC 61850 NO AMBIENTE INDUSTRIAL

Após uma extensa busca sobre o uso da norma em ambientes industriais não relacionados a sistemas elétricos de potência, como subestações e recursos energéticos distribuídos, não foi encontrada literatura especializada abordando esse tema.

Para adaptar a norma IEC 61850 ao ambiente da indústria de processos gerais (fora do escopo de subestações, sistemas de potência e recursos energéticos distribuídos), seria necessária a criação de alguns nós lógicos específicos para representar funcionalidades presentes em setores industriais não contemplados pela norma. No entanto, a maioria dos nós lógicos já definidos atende partes gerais da indústria, como sensores de temperatura (TTMP), pressão (TPRS), posição (TPOS), ângulo (TANG), distância (TDST) e diversas funções, como contadores (FCNT), filtros (FFIL), vazão (TFLW), nível (TLVL), controle PID (FPID) e alarmes (CALH), entre outros.

Quanto aos protocolos de comunicação da norma, o GOOSE poderia ser reparametrizado para dispensar a função de rajada em transmissões periódicas de dados, como o envio de palavras de controle. Nessa configuração, frames adicionais seriam enviados periodicamente além do primeiro valor, o que certamente aumentaria a ocupação do meio de transmissão. Para transmissões não periódicas, o protocolo manteria seu modo de operação tradicional.

O protocolo SAMPLED VALUES seria adequado para a comunicação de sensores no chão de fábrica, uma vez que esses dispositivos realizam a amostragem de valores analógicos para processamento, geralmente por um CLP. Essa adaptação permitiria que tais dispositivos — muito mais numerosos do que os CLPs em uma fábrica — se tornassem "inteligentes" (cada um funcionando como um IED), sendo identificados na rede para trocar informações diretamente com outros equipamentos. Isso resultaria na substituição de cabos que transmitem tensão ou corrente elétrica por cabos de rede.

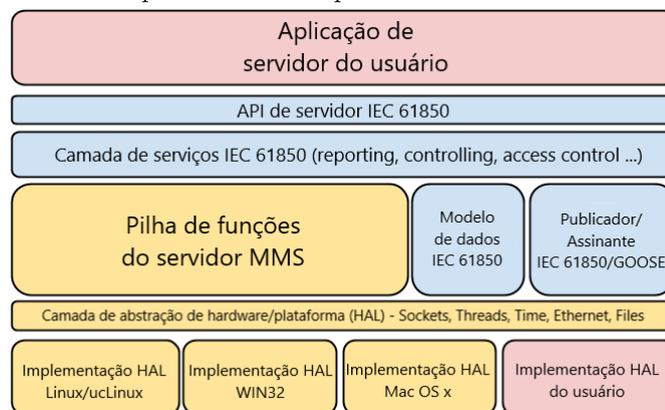
Existe, também, a possibilidade de mapear protocolos já existentes na indústria de processos como por exemplo PROFIBUS, MODBUS, CAN e PROFINET para serem integrados com a IEC 61850, criando mapeamentos específicos e serviços relacionados a estes protocolos, assim como ocorreu com MMS. Possivelmente, esta integração permitira a reutilização dos dispositivos de comunicação e CLPs presentes nas indústrias. Porém, esta solução demandaria um estudo extenso sobre estes protocolos e seu funcionamento em

conjunto com a IEC 61850. Logo, a solução mais simples encontrada para o uso adaptado da IEC 61850 na indústria de processo é a utilização de protocolos já definidos em norma, como será mostrado a seguir.

4 ESCOLHA DE BIBLIOTECA DE SOFTWARE

Para utilizar funções da IEC 61850 em sistemas programáveis, foram pesquisadas diversas bibliotecas que implementam a norma IEC 61850 em dispositivos embarcados, como computadores de placa única ou microcontroladores. Contudo, a maioria das bibliotecas é de código fechado, como (STMICROELECTRONICS, 2024) e (MICROWORKS, 2024), impossibilitando o desenvolvimento em outras plataformas. A biblioteca escolhida foi a *libIEC61850* (AUTOMATION, 2024b) por ser de código aberto, usar a linguagem de programação C, amplamente empregada em computadores e microcontroladores, e oferecer camadas de abstração de hardware (*Hardware Abstraction Layer* – HAL) para sistemas como *Windows*, *Linux* e *BSD (Unix)*, facilitando a migração para novos dispositivos. Além disso, a *libIEC61850* fornece exemplos prontos, incluindo a criação de IEDs cliente/servidor e publicadores/assinantes de mensagens GOOSE e SV. A Figura 11 mostra, em blocos amarelos, as partes de um servidor MMS configuradas pela biblioteca.

Figura 11 – Arquitetura de componentes de servidor da biblioteca.



Fonte: Adaptado de (AUTOMATION, 2024a).

A biblioteca oferece três modos de criar um servidor IED:

1. **Modo estático:** Utiliza arquivos SCL da norma contendo a descrição completa dos componentes e parâmetros de rede do IED, convertendo-os em arquivos de modelo, um de cabeçalho (*static_model.h*) e outro de código-fonte (*static_model.c*) em linguagem C, por meio de uma ferramenta JAVA (AUTOMATION, 2024a). Esses arquivos são referenciados na compilação do programa principal do usuário.
2. **Modo dinâmico:** Cria o modelo em tempo de execução (runtime) usando funções da própria biblioteca. Este modo será detalhado em seção posterior.

3. **Arquivo de configuração:** Similar ao modo estático, converte arquivos SCL em um arquivo de configuração (.cfg) por meio de ferramenta JAVA. No entanto, os atributos de dado do arquivo são acessados durante a execução do programa, e não na compilação.

4.1 PROVA DE CONCEITO

Para aprender a utilizar a biblioteca, foi desenvolvida uma prova de conceito através de um conversor entre os protocolos Modbus TCP e MMS da IEC 61850, embarcado em um computador Raspberry Pi, conforme ilustrado na Figura 12.

O protocolo Modbus TCP foi escolhido por ser um dos mais utilizados na indústria (MALDONADO *et al.*, 2021) e por já estar integrado nos controladores a serem empregados.

Figura 12 – Raspberry Pi atuando como conversor.



Fonte: Autor.

O conversor comunica dados de um cliente MMS com dispositivos Modbus da microrrede do Laboratório de Armazenamento e Mobilidade (LAM) do Departamento de Engenharia Elétrica (DEE) da UFPE, mostrado na Figura 13. Nesse laboratório, estão presentes equipamentos como inversores fotovoltaicos, bancadas de CLPs e o sistema BESS. Com o conversor, é possível acessar os dados dos dispositivos Modbus organizados como IEDs servidores via MMS, onde cada IED corresponde a um equipamento do laboratório.

Nesta prova de conceito escolheu-se o modo estático, pois os exemplos da biblioteca já incluem arquivos SCL que permitiram rápida criação e teste. Na aplicação do Capítulo 6, optou-se pelo modo dinâmico devido à sua facilidade de uso e por não exigir um arquivo SCL embarcado em microcontroladores. Como o terceiro modo é similar ao primeiro, apenas o modo estático foi mantido neste conversor por ser o padrão nos exemplos da biblioteca.

Após testes de comunicação com programas de exemplo compilados em diferentes máquinas virtuais com *Linux*, escolheu-se o exemplo *server_example_control* como base

Figura 13 – Equipamentos no Laboratório de Armazenamento e Mobilidade (LAM).



Fonte: (PERNAMBUCO, 2024).

para criar servidores, pois, além do servidor IED, ele implementa funções de controle para aceitar comandos de clientes MMS. A Figura 14 mostra o servidor em execução de terminal via Raspberry Pi.

Figura 14 – Execução de servidor MMS via terminal.

```

pi@raspberrypi: ~/libiec61850/examples/server_example_goose
Arquivo Editar Abas Ajuda
pi@raspberrypi:~/libiec61850/examples/server_example_goose $ ls
CMakeLists.txt      server_example_goose.c      static_model.h
Makefile            simpleIO_direct_control_goose.cid
server_example_goose static_model.c
pi@raspberrypi:~/libiec61850/examples/server_example_goose $ sudo ./server_example_goose
_goose
LogicalNode LLN0 has 4 fc components
  and 2 unbuffered RCBs
  and 2 GSE control blocks
LogicalNode LPHD1 has 2 fc components
LogicalNode GGIO1 has 5 fc components
IED_SERVER: create dataset: simpleIOGenericIO/LLN0$Events
IED_SERVER: create dataset: simpleIOGenericIO/LLN0$Events2
IED_SERVER: create dataset: simpleIOGenericIO/LLN0$Events3
IED_SERVER: create dataset: simpleIOGenericIO/LLN0$AnalogValues
IED_SERVER: Insert into cache simpleIOGenericIO - LLN0
IED_SERVER: Insert into cache simpleIOGenericIO - LLN0$ST
IED_SERVER: Insert into cache simpleIOGenericIO - LLN0$SCF
IED_SERVER: Insert into cache simpleIOGenericIO - LLN0$DC
IED_SERVER: Insert into cache simpleIOGenericIO - LLN0$EX
IED_SERVER: Insert into cache simpleIOGenericIO - LPHD1
IED_SERVER: Insert into cache simpleIOGenericIO - LPHD1$ST
IED_SERVER: Insert into cache simpleIOGenericIO - LPHD1$DC
IED_SERVER: Insert into cache simpleIOGenericIO - GGIO1

```

Fonte: Autor.

Este servidor exemplo utiliza *Logical Nodes* do tipo GGIO, que representam dados genéricos e vão na contramão do conceito de interoperabilidade semântica que a norma pretende alcançar. Porém, para realizar o teste da prova de conceito em conjunto com o protocolo MODBUS, o uso destes *Logical Nodes* é aceitável.

5 CRIAÇÃO DE CAMADA DE ABSTRAÇÃO DE HARDWARE PARA ESP32

Com o objetivo de desenvolver um dispositivo mais adequado à indústria de processos que permitisse o uso de recursos da IEC 61850, iniciou-se o processo de embarcar a biblioteca *libIEC61850* em um microcontrolador com sistema operacional de tempo real ideal para aplicações industriais, diferentemente dos sistemas operacionais de alto nível que já possuem camada de abstração de hardware presentes na biblioteca.

Mesmo com a utilização de um microcontrolador que possui esta biblioteca embarcada, não é possível definir este dispositivo como um IED, mas sim como um protótipo de CLP, pois não possui as mesmas características de robustez que garantem a integridade física e determinismo temporal de um IED em uma subestação elétrica e também não possui uma interface de programação e diversos outros requisitos presentes um CLP.

Para esse fim, foi escolhida a placa de desenvolvimento WT32-ETH01 (Figura 15), equipada com o microcontrolador ESP32, devido a suas características de boa capacidade de memória flash e RAM (4 MB e 512 KB, respectivamente), conjunto abrangente de funções (*ESP-IDF*) compatíveis com o padrão POSIX da HAL *Linux* da biblioteca, controlador MAC e porta RJ45, que atende ao requisito físico de comunicação via Ethernet da IEC 61850.

Figura 15 – Placa de desenvolvimento WT32-ETH01.



Fonte: (WIRELESS-TAG, 2024).

Para embarcar a biblioteca no ESP32, os módulos *file_provider*, *thread*, *socket*, *time* e *lib_memory* da HAL *Linux* foram reaproveitados com poucas ou nenhuma modificação, pois suas funções são semelhantes às definidas no framework ESP-IDF (ESPRESSIF, 2024a). No entanto, o módulo *ethernet* escolhido para realizar modificações foi o da HAL *BSD*. Apesar de usar funções POSIX de alto nível para a camada de enlace, incompatíveis com

o framework devido a funções específicas, essa abordagem facilitou a adaptação necessária para criar o módulo *ethernet* da HAL ESP32.

Após diversos testes simulados com a nova camada de abstração, verificou-se a necessidade de experimentação com CLPs para avaliar a comunicação do microcontrolador com equipamentos físicos, do mesmo modo que ocorreu com a prova de conceito via Raspberry Pi. Para isso, os parâmetros do protocolo GOOSE foram configurados conforme a Figura 16 e podem ser alterados via arquivo de cabeçalho antes de compilar a biblioteca. O intervalo de transmissão em condições estáveis foi mantido no padrão de 5 segundos, e o número de retransmissões durante eventos foi reduzido a zero, pois o alto volume de eventos torna desnecessária a redundância para garantir o recebimento de uma variável que está sendo modificada em intervalo de tempo constante.

Figura 16 – Configurações GOOSE da biblioteca.

```

/* The GOOSE retransmission interval in ms for the stable condition - i.e. no monitored value changed */
#define CONFIG_GOOSE_STABLE_STATE_TRANSMISSION_INTERVAL 5000

/* The GOOSE retransmission interval in ms in the case an event happens. */
#define CONFIG_GOOSE_EVENT_RETRANSMISSION_INTERVAL 100

/* The number of GOOSE retransmissions after an event */
#define CONFIG_GOOSE_EVENT_RETRANSMISSION_COUNT 0

/* The default value for the priority field of the 802.1Q header (allowed range 0-7) */
#define CONFIG_GOOSE_DEFAULT_PRIORITY 4

/* The default value for the VLAN ID field of the 802.1Q header - the allowed range is 2-4096 or 0 if VLAN/priority is not used */
#define CONFIG_GOOSE_DEFAULT_VLAN_ID 0

/* Configure the 16 bit APPID field in the GOOSE header */
#define CONFIG_GOOSE_DEFAULT_APPID 0x1000

/* Default destination MAC address for GOOSE */
#define CONFIG_GOOSE_DEFAULT_DST_ADDRESS {0x01, 0x0c, 0xcd, 0x01, 0x00, 0x01}

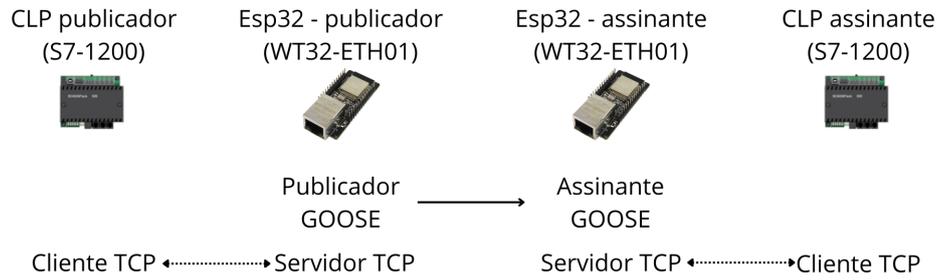
```

Fonte: Autor.

A aplicação consistiu em uma rede IEC 61850 com dois conversores, sendo um publicador e o outro assinante de mensagens GOOSE. Cada conversor se comunicava com um CLP S7-1200 da *Siemens* (Figura 1) por meio de socket TCP, representando os dados neste conversor. A Figura 17 ilustra a arquitetura lógica de comunicação entre os microcontroladores e seus respectivos CLPs. Assim como a prova de conceito via Raspberry Pi, foram utilizados *Logical Nodes* GGIO apenas para verificação do funcionamento da biblioteca.

O servidor TCP de cada microcontrolador baseou-se no modelo *TCP non-blocking server* do framework ESP-IDF, disponível em (ESPRESSIF, 2024b). Os projetos com os códigos-fonte desta aplicação e camada HAL para esp32 estão disponíveis em (NUNES, 2024). Como estes resultados foram semelhantes aos da prova de conceito via Raspberry

Figura 17 – Arquitetura de comunicação entre ESP32 e CLP.

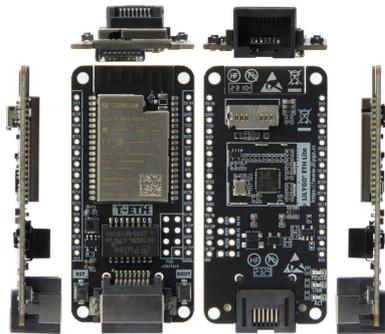


Fonte: Autor.

Pi, serão mostrados no Capítulo 6 resultados de um protótipo que se aproxima mais a um CLP em uma aplicação de controle distribuído, semelhante a uma planta industrial.

Após testes de integração da biblioteca com outras bibliotecas na placa de desenvolvimento utilizada, como por exemplo a biblioteca FORTE para programação distribuída do framework 4DIAC (ECLIPSE-4DIAC, 2024c), verificou-se que a memória RAM era insuficiente para incluir mais de uma biblioteca em aplicações futuras. Assim, tornou-se necessário escolher uma nova placa também baseada no ESP32, para que não fosse necessário reescrever a camada de abstração de hardware. A placa escolhida foi T-ETH-Lite (Figura 18), equipada com controlador MAC e porta RJ45, como a placa anterior, mas com 16 MB de memória flash e um adicional de 8 MB de memória PSRAM, sendo necessário habilitar esta memória extra no arquivo de projeto do microcontrolador durante a programação para que seja utilizada.

Figura 18 – Placa de desenvolvimento T-ETH-Lite.



Fonte: (LILYGO, 2024).

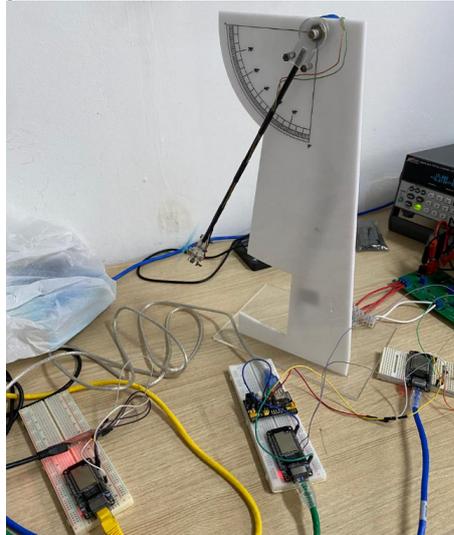
6 DESENVOLVIMENTO DE PROTÓTIPO DE DISPOSITIVO INDUSTRIAL EM PLANTA DE AEROPÊNDULO

Com a biblioteca embarcada na nova placa, foram programadas três placas para representar um sistema distribuído na planta de controle de um aeropêndulo, desenvolvido em (NETO *et al.*, 2023). A planta consiste em um pêndulo, um sensor angular resistivo, marcações de ângulo em acrílico para referência de altura, um motor DC conectado a uma hélice na extremidade do pêndulo e uma placa de circuito eletrônico para condicionamento de sinal e acionamento.

O sistema original do aeropêndulo funciona com o envio de um setpoint de ângulo, entre 0 e 90°, de um microcontrolador para a placa de circuito. Esta processa o sinal e aciona o motor DC para fornecer a potência necessária para manter o pêndulo na posição desejada. A leitura do sensor de posição é enviada ao microcontrolador por meio de um conversor analógico-digital (ADC), que compara a posição atual e a desejada para realizar o controle de altura usando um controlador proporcional-integral (PI).

Apesar das vantagens de memória e conectividade, o ESP32 possui um ADC limitado, exigindo calibração e filtros para reduzir ruídos. Mesmo assim, ocorre uma falta de linearidade na leitura analógica, especialmente entre 30° e 60°, com variações de até 6°. A Figura 19 ilustra o aeropêndulo em operação.

Figura 19 – Aeropêndulo em funcionamento.

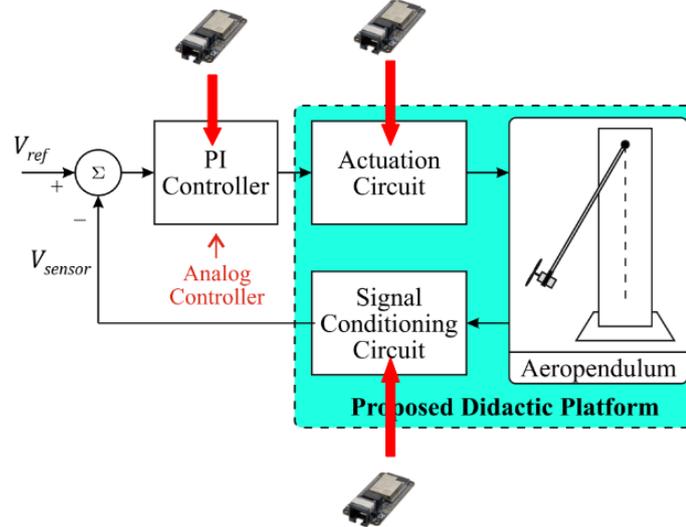


Fonte: Autor.

Para realizar controle distribuído, sensor, atuador e controlador foram implementados em três dispositivos protótipos, cada um representado por um microcontrolador (Figura 19) e ambos conectados em um switch de rede comum. A malha de controle

simplificada desta aplicação está mostrada na Figura 20, onde cada dispositivo protótipo corresponde a uma parte da malha.

Figura 20 – Malha de controle simplificada.



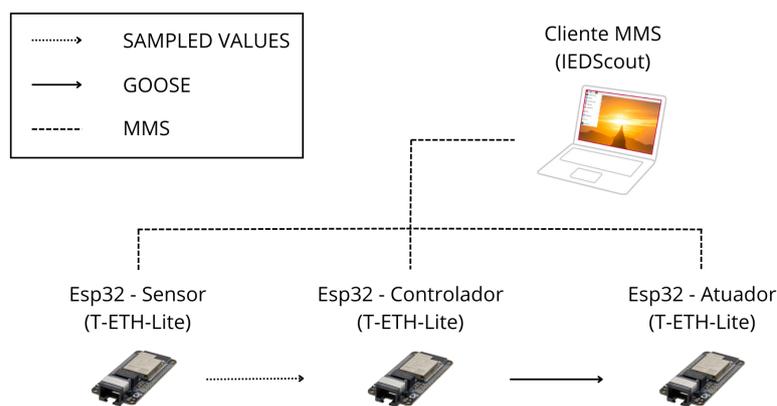
Fonte: Adaptado de (NETO *et al.*, 2023).

Para uma configuração mais próxima da IEC 61850, o ideal seria configurar variáveis de qualidade e a presença de uma VLAN em um switch ethernet compatível com esta norma, mas para esta aplicação, não é necessário um gerenciamento rigoroso das mensagens dos protocolos GOOSE e SAMPLED VALUES. Também foi utilizado um computador com cliente MMS *IEDScout* para monitorar os dispositivos e configurar o setpoint de ângulo e os ganhos do controlador PI no dispositivo associado ao controle.

Além destas configurações, os dispositivos não utilizaram a sincronização de tempo necessária de acordo com a norma para garantir a validade dos dados na rede, como por exemplo no caso dos dados amostrados via SAMPLED VALUES.

A arquitetura da rede está mostrada na Figura 21.

Figura 21 – Arquitetura de rede do sistema distribuído.



Fonte: Autor.

Diferentemente das abordagens descritas nas figuras Capítulo 4 e Capítulo 5, nesta etapa, os dispositivos foram programados através do modo dinâmico, com funções da biblioteca que definem nós lógicos e configurações de rede durante a inicialização do microcontrolador (AUTOMATION, 2024c), sem que seja necessária a utilização de um arquivo SCL.

A Listing 6.1 apresenta uma comparação (*if*) na função `create_LN` do código-fonte para criação dinâmica de nós lógicos, onde o nome “FPID” faz parte de um vetor contendo todos os nomes de nós lógicos a serem criados no dispositivo de controle.

```

if(strcmp("FPID",LN_array[i]) == 0){

    // Nó lógico FPID

    LogicalNode* logicn = LogicalNode_create(LN_array[i], lDevice);

    // Objetos de dado do nó lógico

    DataObject* logicn_out    = CDC_MV_create("Out", (ModelNode*) logicn, 0, false);
    DataObject* logicn_pact   = CDC_MV_create("Pact", (ModelNode*) logicn, 0, false);
    DataObject* logicn_iact   = CDC_MV_create("Iact", (ModelNode*) logicn, 0, false);
    DataObject* logicn_dact   = CDC_MV_create("Dact", (ModelNode*) logicn, 0, false);
    DataObject* logicn_errterm = CDC_MV_create("ErrTerm", (ModelNode*) logicn, 0, false);
    DataObject* logicn_pidalg  = CDC_ENG_create("PIDAlg", (ModelNode*) logicn, 0);
    DataObject* logicn_kp     = CDC_ASG_create("KP", (ModelNode*) logicn, 0, false);
    DataObject* logicn_ki     = CDC_ASG_create("KI", (ModelNode*) logicn, 0, false);
    DataObject* logicn_itmms   = CDC_ING_create("ITmms", (ModelNode*) logicn, 0);
    DataObject* logicn_kd     = CDC_ASG_create("KD", (ModelNode*) logicn, 0, false);
    DataObject* logicn_dtmms   = CDC_ING_create("DTmms", (ModelNode*) logicn, 0);
    DataObject* logicn_dfiltmms = CDC_ING_create("DFilTmms", (ModelNode*) logicn, 0);
    DataObject* logicn_bias   = CDC_ASG_create("Bias", (ModelNode*) logicn, 0, false);
    DataObject* logicn_ilmim  = CDC_ASG_create("ILim", (ModelNode*) logicn, 0, false);
    DataObject* logicn_drp    = CDC_ASG_create("Droop", (ModelNode*) logicn, 0, false);

    // Configuracoes GOOSE

    char *dataset_name = "OutVal";
    char *rptname = "events_fpid";

    DataSet* dataset_fpid = DataSet_create(dataset_name, logicn);

    DataSetEntry_create(dataset_fpid, "FPID$MX$Out$mag$f", -1, NULL);

    ReportControlBlock_create(rptname, logicn, rptname, false, NULL, 1, TRG_OPT_DATA_CHANGED,
    rptOptions, 50, 0);
    GSEControlBlock* gsecb_fpid = GSEControlBlock_create("gsecb_fpid", logicn, rptname, dataset_name, 1, false,
    -1, -1);

```

```
GSEControlBlock_addPhyComAddress(gsecb_fpid, phy_teste);

    continue;
}
```

Listing 6.1 – Criação dinâmica de nó lógico FPID. Fonte: Autor.

Durante a inicialização, o endereço MAC de publicação do dispositivo de controle é utilizado para criar um conjunto de dados da biblioteca (*PhyComAddress*) que contém os parâmetros de rede para a publicação de mensagens GOOSE. Esse conjunto é repassado como parâmetro para a função *create_LN* para configurar publicadores GOOSE relativos aos nós lógicos que publicam dados. A Listing 6.2 ilustra o exemplo de inicialização para deste dispositivo.

```
uint8_t mac_dst[6];
mac_dst[0] = 0x01;
mac_dst[1] = 0x0c;
mac_dst[2] = 0xcd;
mac_dst[3] = 0x01;
mac_dst[4] = 0x00;
mac_dst[5] = 0x01;

// Criacao de modelo dinâmico do IED servidor

PhyComAddress* phy_teste = PhyComAddress_create(4, 0, 0x1000, mac_dst);

model = IedModel_create("esp32cont");

LogicalDevice* lDevice = LogicalDevice_create("devc", model);

// Criar nós lógicos

create_LN(lDevice, phy_teste);

// Adicionar o modelo dinâmico ao IED

iedServer = IedServer_create(model);

// Funcao de callback para eventos de controle

IedServer_setControlHandler(iedServer, (DataObject*) IedModel_getModelNodeByShortObjectReference(model, "devc/
    FSPT.SptVal"), (ControlHandler) controlHandlerForFSPT, 0);

// Iniciar servidor

IedServer_start(iedServer, 102);
```

Listing 6.2 – Criação dinâmica de nó lógico FPID. Fonte: Autor.

Adicionalmente, não foram utilizadas estampas de tempo atualizadas, e os microcontroladores foram inicializados com o tempo inicial do padrão UNIX em 1º de janeiro de 1970, sujeito a variações da hora conforme a zona de tempo.

A seguir, serão descritas as funcionalidades de cada dispositivo utilizado:

- Dispositivo sensor

O dispositivo sensor foi responsável pela aquisição do ângulo atual e disponibilização via nó lógico TANG (Figura 22), utilizando o atributo *AngSv* da classe de dados de valores amostrados (SAV), com publicação na rede via protocolo SAMPLED VALUES. Na figura, observa-se um valor de ângulo medido em aproximadamente 45°.

Figura 22 – Monitoramento do nó lógico TANG via cliente MMS.

esp32 • Data Model • sensordevs • TANG			
LN TANG			
Name			Value
DO	AngSv		45.105
DA	instMag	[MX]	45.105
DA	q	[MX]	good
DA	t	[MX]	01/01/1970 00:16:24.186

Fonte: Autor.

A configuração de tempo foi realizada com um atraso no ciclo de aquisição de dados analógicos via conversor ADC, obtendo intervalos de aproximadamente 100 ms entre cada medida de ângulo publicada, conforme mostrado na segunda coluna mais à esquerda da Figura 23, obtida com o software *Wireshark*.

Figura 23 – Tempo entre pacotes SV.

```

140 24.926743  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
141 25.025905  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
144 25.125014  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
145 25.224658  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
147 25.323171  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
149 25.422638  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
150 25.521804  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
151 25.620863  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
152 25.720110  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
153 25.819098  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
155 25.917620  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
156 26.017501  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68
157 26.115698  Espressif_31:c7:af  IecTc57_01:00:01  IEC61850 Sampled Values  68

```

Fonte: Autor.

- Dispositivo de controle

Utiliza o nó lógico FSPT, com o atributo *SptVal* da classe de dados APC para entrada do setpoint de ângulo, e o nó lógico FPID para representar na rede o controlador

PI. O nó FPID contém atributos como *Out* (saída calculada), *Pact* (ação proporcional), *Iact* (ação integral) e *PIDAlg* (tipo de controlador, com valor 4 para PI), além de permitir ajustar os ganhos *KP* e *KI*. A Figura 24 exibe o monitoramento e ajustes via cliente MMS.

Figura 24 – Monitoramento e ajuste dos nós lógicos FSPT e FPID via cliente MMS.

The image shows two screenshots from an MMS client interface. The left screenshot displays the 'FPID' node with a table of parameters. The right screenshot displays the 'FSPT' node with a table of parameters, including sub-parameters under 'SptVal'.

esp32 • Data Model • contdevc • FPID	
LN FPID	
Name	Value
DO Out	59.774
DO Pact	0.07
DO Iact	59.841
DO Dact	0
DO ErrTerm	0
DO PIDAlg	4
DO KP	0.095
DO KI	0.01

esp32 • Data Model • contdevc • FSPT	
LN FSPT	
Name	Value
DO SptMem	0
DO ErrTerm	0
DO Out	0
DO SptChg	0
DO SptVal	45
DA mxVal	[MX] 45
DA f	[MX] 45
DA q	[MX] good
DA t	[MX] 31/12/1969 22:00:00.000
DA Oper	[CO]
DA ctlModel	[CF] direct-with-normal-security

Fonte: Autor.

Esse dispositivo assina mensagens do dispositivo sensor via protocolo SAMPLED VALUES para realizar os cálculos do controlador PI e publica sua saída na rede via protocolo GOOSE, sem atrasos adicionais, pois cada mensagem de ângulo é processada no instante de chegada. A saída do controlador também é publicada em cerca de 100 ms. No nó FPID, a ação integral excede a proporcional, pois o sistema já oscila próximo ao setpoint.

- Dispositivo atuador

O atuador assina mensagens GOOSE do controlador e representa a posição do pêndulo via nó lógico TPOS (Figura 25), alterando sua escala conforme a IEC 61850 (por exemplo, 50% de 90° equivale a 45°). Ele aplica o valor de saída do controlador ao motor DC via conversor digital-analógico.

Figura 25 – Monitoramento do nó lógico TPOS via cliente MMS.

The image shows a screenshot of the MMS client interface for the 'TPOS' node. It displays a table of parameters for the node.

esp32 • Data Model • contdevc • TPOS	
LN TPOS	
Name	Value
DO PosPctSv	50.714
DA instMag	[MX] 50.714
DA f	[MX] 50.71358
DA q	[MX] good
DA t	[MX] 31/12/1969 22:00:35.411

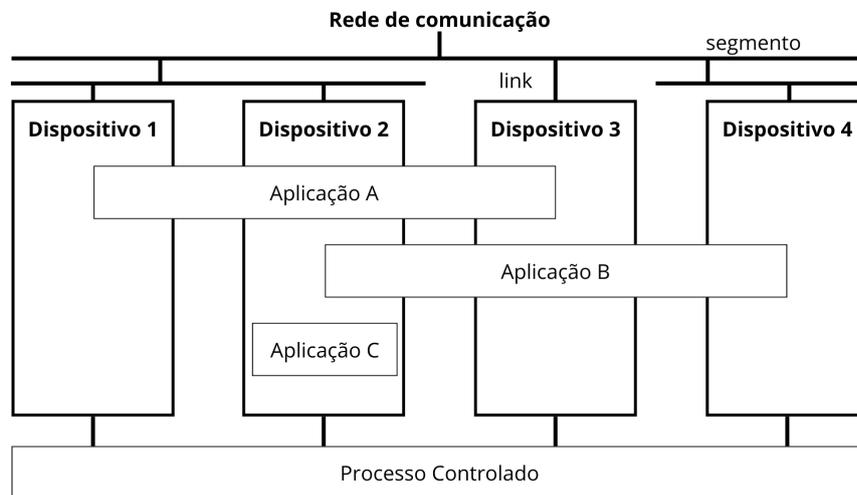
Fonte: Autor.

Após os testes, verificou-se o correto funcionamento de todos os dispositivos em rede e uma rápida dinâmica de controle com os ganhos da Figura 24, assim como uma diferença significativa entre o ângulo real e o amostrado em determinados intervalos, devido ao conversor ADC da ESP32.

7 IEC 61499 E IEC 61850

Além de utilizar a IEC 61850 para padronizar a comunicação entre controladores na indústria, é possível utilizar a norma IEC 61499 para padronizar o modo como os controladores são programados. A norma IEC 61499-1 especifica o uso de blocos de função em sistemas de medição e controle de processos industriais distribuídos, definindo modelos de referência, sintaxe textual e representações gráficas (IEC, 2012). A Figura 26 ilustra um modelo de sistema de medição e controle industrial, onde as aplicações A e B são distribuídas em múltiplos dispositivos para interagir com o processo controlado.

Figura 26 – Modelo de sistema.

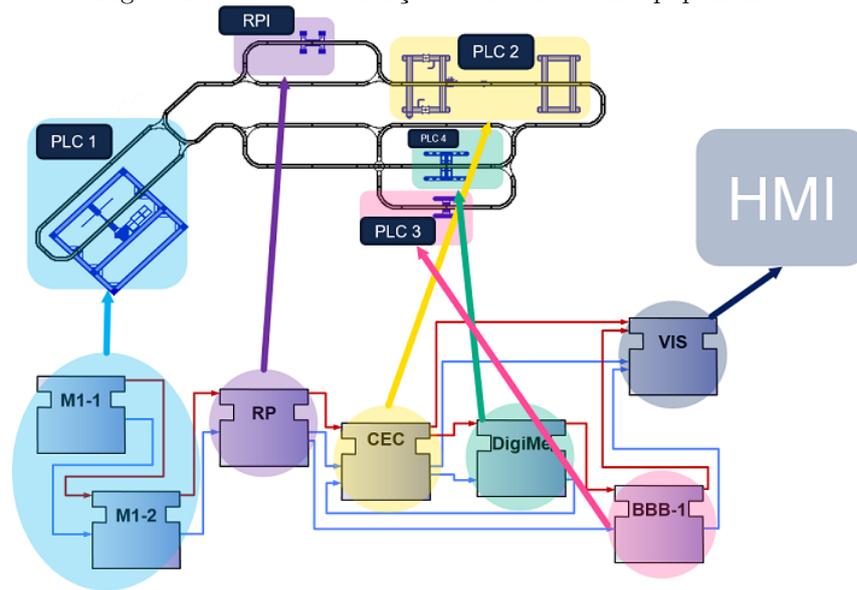


Fonte: Adaptado de (IEC, 2012).

A integração com a IEC 61850 possibilita padronizar tanto a programação quanto a comunicação em rede de dispositivos em sistemas industriais distribuídos. Para implementar essa integração, as bibliotecas *libIEC61850* e 4DIAC-FORTE foram embarcadas em um microcontrolador T-ETH-Lite seguindo as instruções de (HIKIKU, 2024), conforme mencionado no Capítulo 5. A biblioteca 4DIAC-FORTE, desenvolvida em linguagem C++, é um ambiente de execução de aplicações criadas no framework 4DIAC, atuando como uma camada de abstração de hardware para dispositivos como CLPs ou Raspberry Pi.

O framework 4DIAC oferece uma infraestrutura de código aberto para programar sistemas de controle industrial distribuídos, baseado na IEC 61499, utilizando blocos de função orientados a eventos. A Figura 27 exemplifica a distribuição de blocos de função em CLPs e um computador.

Figura 27 – Blocos de função distribuídos em equipamentos.



Fonte: (ECLIPSE-4DIAC, 2024b).

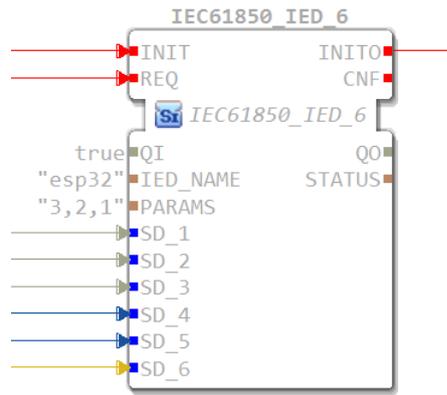
O modelo descrito em (ECLIPSE-4DIAC, 2024a) define a criação de blocos de função para interfaces de serviço que gerenciam eventos externos, como a comunicação via GOOSE. A Figura 28 mostra um bloco publicador que também cria um servidor MMS com nós lógicos GGIO, que já foram citados anteriormente como genéricos e servem aqui para fins de testes. As setas vermelhas indicam eventos que acionam o bloco, como *INIT* para inicialização e *INITO* para repassar eventos a blocos subsequentes. Eventos de requisição (*REQ*) são necessários para mudanças nas variáveis de entrada, mas eventos de confirmação (*CNF*) não foram utilizados.

O bloco segue o modelo dinâmico para publicação de variáveis, onde o sufixo numérico do nome (6 em IEC61850_IED_6) indica o número de variáveis de entrada publicadas. Os parâmetros *IED_NAME* e *PARAMS* definem o nome do servidor MMS e a configuração das variáveis, respectivamente.

Blocos assinantes foram criados dinamicamente para diferentes tipos de dados, como mostrado na Figura 29. Eles utilizam o endereço *MAC* de publicação e a referência do bloco de controle de dados GOOSE (*goCbRef*) como parâmetros de entrada, fornecendo as variáveis assinadas nas saídas *RD_1* a *RD_3*. Esses blocos não requerem eventos de requisição, pois a assinatura de mensagens GOOSE pode ser tratada como evento externo pela IEC 61499.

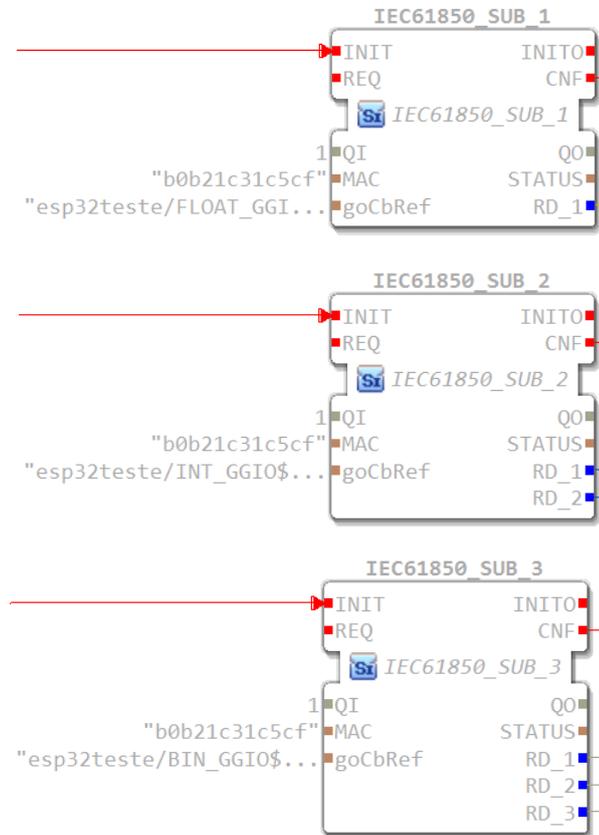
A integração da IEC 61499 com a IEC 61850 ainda exige pesquisas adicionais, e os blocos de função apresentados não abrangem a criação completa de todos os nós lógicos,

Figura 28 – Bloco de função de publicador via protocolo GOOSE.



Fonte: Autor.

Figura 29 – Blocos de função de assinatura via protocolo GOOSE.



Fonte: Autor.

protocolos especificados na norma e requisitos de comunicação para dispositivos da IEC 61850.

8 CONCLUSÕES

Conclui-se que o desenvolvimento de um dispositivo protótipo de comunicação industrial compatível com a norma IEC 61850 é uma tarefa complexa, que exige atenção não apenas aos aspectos de comunicação, mas também às características físicas do dispositivo, como memória e componentes de rede, os quais podem influenciar suas capacidades operacionais.

O dispositivo conversor via Raspberry Pi que serviu de base para entender o funcionamento da biblioteca também fornece a geração de código automático que está presente em algumas ferramentas de configuração da norma voltadas para o uso em subestações, ao criar um arquivo SCL da IEC 61850 seguido de duas compilações que geram um servidor MMS executável, apenas com o clicar de um botão.

A camada de abstração de hardware desenvolvida para esp32 permite que a biblioteca *libIEC61850* possa ser reutilizada em qualquer microcontrolador esp32 para o uso de algumas funcionalidades da IEC 61850 definidas na biblioteca. É necessário observar também que esta biblioteca foi criada para uso em sistemas operacionais generalistas e de alto nível como *Windows* e *Linux*. Portanto, é possível que ela ainda não esteja otimizada para o uso em microcontroladores, mesmo com a presença de uma camada de abstração de hardware para microcontroladores.

Os testes realizados em uma planta com controle distribuído adequado à indústria de processos e fora do escopo da IEC 61850 apresentaram resultados satisfatórios, e demonstraram que é possível trazer a interoperabilidade da norma IEC 61850 para os ambientes industriais. Seus protocolos são completamente aplicáveis e possuem requisitos de troca de mensagens e de tempo que superam os exigidos pela indústria de processos, geralmente na faixa de dezenas de milissegundos. Portanto, o dispositivo que foi apresentado no Capítulo 6 equivale a um protótipo de um CLP industrial, não possuindo a robustez e algumas funcionalidades necessárias para ser caracterizado como IED na IEC 61850.

O uso de protótipos de CLPs com as funções de comunicação padronizadas pelas IEC 61850 mostrou-se viável neste estudo, dada a implementação em ambiente controlado e o baixo custo de desenvolvimento. Foi constatada a possibilidade dos controladores serem configurados com um arquivo contendo uma semântica pré-definida, compreensível por um profissional especializado. Isso contrasta com os controladores que utilizam va-

riáveis configuradas de modo subjetivo, o que pode gerar dificuldades de entendimento quando alterações são realizadas por profissionais diferentes daqueles que efetuaram a configuração original.

9 TRABALHOS FUTUROS

Identificou-se a necessidade de aprimorar a integração da norma IEC 61850 com dispositivos embarcados por meio da criação de diferentes camadas de abstração de hardware, atendendo aos requisitos de interoperabilidade da norma e eliminando a dependência de dispositivos ou fabricantes específicos. Existe também a necessidade de verificar se as funções da biblioteca estão otimizadas para o uso em microcontroladores, já que ela foi criada para uso em sistemas operacionais de alto nível. Outro aspecto relevante para futuras pesquisas é a verificação do cumprimento dos requisitos temporais das diversas mensagens definidas pela norma, uma questão fora do escopo deste trabalho.

Além do uso em microcontroladores, um estudo sobre a aplicação da biblioteca *libIEC61850* em CLPs que executam sistemas operacionais de alto nível também é necessário, pois estes dispositivos já são certificados para operação nas indústrias e já possuem camada de abstração de hardware definida na biblioteca.

Evidencia-se também a possibilidade de integrar os protocolos de redes industriais que já estão em operação com a IEC 61850 para uso em indústrias, através de mapeamentos específicos destes protocolos e seus prováveis serviços para uma adaptação do ambiente da norma. Esta integração necessitaria de um estudo extenso sobre padronização de comunicação na indústria que permitira escolher entre criar uma norma semelhante à IEC 61850 ou expandir o escopo da IEC 61850.

Adicionalmente, torna-se essencial investigar o uso de nós lógicos em áreas não abrangidas pela norma. Estudos futuros devem explorar se os nós lógicos e os serviços de comunicação existentes, além dos mais comuns mencionados na seção 3.6, são suficientes ou se é necessária a criação de novos nós e serviços para atender a funcionalidades específicas conforme os requisitos de equipamentos industriais.

Outra necessidade importante é a maior integração entre as normas IEC 61499 e IEC 61850, o que pode trazer inovações significativas, ampliando as possibilidades de aplicação da IEC 61850 em ambientes industriais, tanto dentro quanto fora de seu escopo original, promovendo a interoperabilidade entre os ambientes de programação e de configuração da rede.

O processo de configuração de uma rede IEC 61850 composta por vários dispositivos industriais compatíveis com essas normas pode ser bastante simplificado quando comparado

à configuração de dispositivos tradicionais como CLPs. Com a IEC 61850, cada elemento de uma planta industrial conectado a um controlador é reconhecido como um dispositivo distribuído, trocando informações apenas via interface de rede, sem a necessidade de configurações específicas de cabos para entradas/saídas digitais ou analógicas.

A programação de dispositivos industriais através de ferramentas compatíveis com a IEC 61499 também seria consideravelmente mais simples quando comparada à programação em softwares tradicionais para controladores. A IEC 61499 permite a configuração de qualquer dispositivo compatível com a norma em qualquer ferramenta que também seja compatível, além de oferecer uma abordagem de programação orientada a eventos e de mais alto nível. Em contraste, controladores tradicionais permitem apenas a programação de dispositivos do mesmo fabricante, utilizando geralmente linguagens sequenciais.

Portanto, o uso conjunto das normas IEC 61499 e IEC 61850 pode transformar significativamente os modos de funcionamento e organização das plantas industriais, trazendo novas possibilidades para a automação e a padronização dos sistemas de controle.

REFERÊNCIAS

- AFTAB, M. A. *et al.* Iec 61850 based substation automation system: A survey. **International Journal of Electrical Power & Energy Systems**, v. 120, p. 106008, 2020. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061520304646>>.
- AUTOMATION, M. **Documentation | libIEC61850 / lib608704**. 2024. [Online; Acessado em 21-11-2024]. Disponível em: <<https://libiec61850.com/documentation/>>.
- AUTOMATION, M. **libIEC61850 / lib60870 | open source libraries for IEC 61850 and IEC 60870-5-101/104**. 2024. [Online; Acessado em 21-11-2024]. Disponível em: <<https://libiec61850.com/>>.
- AUTOMATION, M. **libiec61850: Helper functions to create common data classes (CDC) using the dynamic model API**. 2024. [Online; Acessado em 27-11-2024]. Disponível em: <https://support.mz-automation.de/doc/libiec61850/c/latest/group__COMMON__DATA__CLASSES.html>.
- ECLIPSE-4DIAC. **4diac-documentation/src/development/externalEvent_example.adoc at main · eclipse-4diac/4diac-documentation**. 2024. [Online; Acessado em 27-11-2024]. Disponível em: <https://github.com/eclipse-4diac/4diac-documentation/blob/main/src/development/externalEvent_example.adoc>.
- ECLIPSE-4DIAC. **4diac-documentation/src/index.adoc at main · eclipse-4diac/4diac-documentation**. 2024. [Online; Acessado em 27-11-2024]. Disponível em: <<https://github.com/eclipse-4diac/4diac-documentation/blob/main/src/index.adoc>>.
- ECLIPSE-4DIAC. **eclipse-4diac/4diac-forte**. 2024. [Online; Acessado em 25-11-2024]. Disponível em: <<https://github.com/eclipse-4diac/4diac-forte>>.
- ESPRESSIF. **ESP-IDF Programming Guide**. 2024. [Online; Acessado em 22-11-2024]. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/v5.3.1/esp32/index.html>>.
- ESPRESSIF. **esp-idf/examples/protocols/sockets/non_blocking/**. 2024. [Online; Acessado em 22-11-2024]. Disponível em: <https://github.com/espressif/esp-idf/tree/master/examples/protocols/sockets/non_blocking>.
- FELSER, M.; SAUTER, T. The fieldbus war: history or short break between battles? In: **4th IEEE International Workshop on Factory Communication Systems**. [S.l.: s.n.], 2002. p. 73–80.
- GROUP, D. U. **A DNP3 Protocol Primer**. 2024. [Online; Acessado em 05-09-2024]. Disponível em: <<https://www.dnp.org/Portals/0/AboutUs/DNP3%20Primer%20Rev%20A.pdf>>.
- GÁLVEZ, J. L. **File:Funcion ladder.PNG**. 2024. [Online; Acessado em 28-07-2024]. Disponível em: <https://commons.wikimedia.org/wiki/File:Funcion_ladder.PNG>.
- HIKIKU. **hikiku/esp-4diac-forte: ESP-4diac-FORTE is a standard ESP-**

IDF component. It describes how to run 4diac FORTE on ESP32 or ESP32S2 MCU. 2024. [Online; Acessado em 27-11-2024]. Disponível em: <<https://github.com/hikiku/esp-4diac-forte>>.

HORALEK, J.; MATYSKA, J.; SOBESLAV, V. Communication protocols in substation automation and iec 61850 based proposal. In: . [S.l.: s.n.], 2013. p. 321–326. ISBN 978-1-4799-0194-4.

IEC. **Communication networks and systems for power utility automation – Part 6: Configuration description language for communication in electrical substations related to IEDs.** Geneva, CH, 2009.

IEC. **Communication networks and systems for power utility automation – Part 7-2: Basic information and communication structure – Abstract communication service interface (ACSI).** Geneva, CH, 2010.

IEC. **Communication networks and systems for power utility automation – Part 7-1: Basic communication structure – Principles and models.** Geneva, CH, 2011.

IEC. **Communication networks and systems for power utility automation – Part 8-1: Specific communication service mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3.** Geneva, CH, 2011.

IEC. **Communication networks and systems for power utility automation – Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3.** Geneva, CH, 2011.

IEC. **Function blocks – Part 1: Architecture.** Geneva, CH, 2012.

IEC. **Communication networks and systems for power utility automation – Part 1: Introduction and overview.** Geneva, CH, 2013.

IEC. **Communication networks and systems for power utility automation – Part 5: Communication requirements for functions and device models.** Geneva, CH, 2013.

IEC. **Programmable controllers – Part 3: Programming languages.** Geneva, CH, 2013.

ISO/IEC. **Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model.** Geneva, CH, 1994.

KUMAR, S. *et al.* Review of the legacy and future of iec 61850 protocols encompassing substation automation system. **Electronics**, v. 12, n. 15, 2023. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/12/15/3345>>.

LILYGO. **T-ETH-Lite – LILYGO.** 2024. [Online; Acessado em 22-11-2024]. Disponível em: <<https://lilygo.cc/products/t-eth-lite>>.

LUKMAK. **File:Sampled signal.svg.** 2024. [Online; Acessado em 7-10-2024]. Disponível em: <https://commons.wikimedia.org/wiki/File:Sampled_signal.svg>.

MACKAY, S. *et al.* **Practical Industrial Data Networks: Design, Installation and**

Troubleshooting (IDC Technology). 1. ed. [S.l.]: Newnes, 2004. (IDC Technology). ISBN 075065807X; 9780750658072; 9781417537181; 1417537183.

MACKIEWICZ, R. Overview of iec 61850 and benefits. In: **2006 IEEE Power Engineering Society General Meeting**. [S.l.: s.n.], 2006. p. 8 pp.–.

MALDONADO, V. *et al.* Development of an industrial data server for modbus tcp protocol. In: _____. [S.l.: s.n.], 2021. p. 16–28. ISBN 978-3-030-63664-7.

MCDONALD, J. Substation automation. ied integration and availability of information. **IEEE Power and Energy Magazine**, v. 1, n. 2, p. 22–31, 2003.

MICROWORKS, T. **IEC 61850 Stack Overview**. 2024. [Online; Acessado em 16-12-2024]. Disponível em: <<https://trianglemicroworks.com/products/source-code-libraries/iec-61850-scl-pages/overview>>.

NETO, R. *et al.* An aeropendulum-based didactic platform for the learning of control engineering. **Journal of Control, Automation and Electrical Systems**, v. 34, p. 1–12, 02 2023.

NUNES, P. R. M. **p-r-m-n/esp32-61850-socket: TCP socket/IEC 61850 interface that turns PLC's into IEC 61850 IED's**. 2024. [Online; Acessado em 22-11-2024]. Disponível em: <<https://github.com/p-r-m-n/esp32-61850-socket>>.

PERNAMBUCO, R. J. **O Galpão da elétrica, UFPE tem agora seu próprio QG para energia limpa**. 2024. [Online; Acessado em 21-11-2024]. Disponível em: <<https://radiojornal.ne10.uol.com.br/noticia/tecnologia/2022/12/15149586-galpao-da-eletrica-o-qg-da-energia-limpa-da-ufpe.html>>.

PETRUZELLA, F. D. **Controladores logicos programaveis**. [S.l.: s.n.], 2014. ISBN 9788580552836; 8580552834; 9780073510880; 0073510882.

SCANZIO, S.; WISNIEWSKI, L.; GAJ, P. Heterogeneous and dependable networks in industry – a survey. **Computers in Industry**, v. 125, p. 103388, 2021. ISSN 0166-3615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0166361520306229>>.

SISCO. Overview and introduction to the manufacturing message specification (mms). 1995.

STMICROELECTRONICS. **IEC 61850 Library - Lightweight, highly configurable, platform agnostic, delivered as a compiled binary or as a source code IEC 61850 library for embedded devices - STMicroelectronics**. 2024. [Online; Acessado em 16-12-2024]. Disponível em: <<https://www.st.com/en/partner-products-and-services/iec-61850-library.html>>.

THOMESSE, J.-P. Fieldbus technology in industrial automation. **Proceedings of the IEEE**, v. 93, n. 6, p. 1073–1101, 2005.

ULRICHAAB. **File:Simatic S7-1200.JPG**. 2024. [Online; Acessado em 28-07-2024]. Disponível em: <https://commons.wikimedia.org/wiki/File:Simatic_S7-1200.JPG>.

WIRELESS-TAG. **WT32-ETH01_Shenzhen Qiming Cloud Technology**

Co., Ltd. 2024. [Online; Acessado em 22-11-2024]. Disponível em: <<https://en.wireless-tag.com/product-item-2.html>>.

ZURAWSKI, R. H. **The Industrial Communication Technology Handbook**(Industrial Information Technology Series). [S.l.: s.n.], 2005.