Aplicação de sistemas computacionais de baixo custo para emular IED com protocolo MMS

Marissa Wong Na * **Rainer Zanghi** **

* *Graduação em Engenharia Elétrica, Universidade Federal Fluminense, RJ, (e-mail: marissawong@id.uff.br).*
** *Departamento de Engenharia Elétrica, Universidade Federal Fluminense, RJ, (e-mail: rzanghi@id.uff.br).*

Abstract: The communication among Intelligent Electronic Devices (IEDs) in the automation of power electrical systems is paramount for their real-time operation. The IEC 61850 standard was conceived to standardize models and ensure interoperability among IEDs manufactured by different entities. Nevertheless, owing to its intricacy, the execution of tests and simulations becomes imperative to validate its functionalities and the expected behavior of its constituents. Thus, for research and development endeavors, the utilization of open-source platforms and software for the verification of specific functionalities is enticing, given the potential impracticality of procuring equipment for such purposes due to their exorbitant costs. This article delineates the development of an IED emulation embedding the MMS protocol within a low-cost computational system. Tests are delineated to ascertain if the emulated IED can replicate functions and the data structure of an authentic device employing the MMS protocol.

Resumo: A comunicação entre Dispositivos Eletrônicos Inteligentes (IEDs) na automação de sistemas elétricos de potência é crucial para seu funcionamento e sua operação em tempo real. A norma IEC 61850 foi desenvolvida com o objetivo de padronizar os modelos e garantir a interoperabilidade entre IEDs de diferentes fabricantes. No entanto, devido a sua complexidade, faz-se necessária a realização de testes e simulações para validação de suas funções e do comportamento esperado de seus componentes. Portanto, para fins de pesquisa e desenvolvimento, é atraente a utilização de plataformas e softwares de código aberto para verificação de funcionalidades específicas, uma vez que pode ser inviável a aquisição de equipamentos para essas finalidades devido aos seus altos custos. Nesse artigo, é relatado o desenvolvimento de uma emulação de IED embarcando o protocolo MMS em um sistema computacional de baixo custo. São apresentados testes para verificar se o IED emulado é capaz de reproduzir funções e a estrutura de dados de um dispositivo real, utilizando o protocolo MMS.

Keywords: Intelligent Electronic Device, IEC 61850, MMS, Computational Systems.
Palavras-chaves: Dispositivos Eletrônicos Inteligentes, IEC 61850, MMS, Sistemas Computacionais.

1. INTRODUÇÃO

À medida em que são dados passos acelerados em direção ao domínio digital, seja na esfera industrial, comercial ou no cotidiano (Uzair, 2020), o avanço da tecnologia e o ambiente industrial moderno encontram-se cada vez mais integrados (Souza, 2019). No que diz respeito aos sistemas elétricos, a modernização de subestações de energia elétrica, impulsionada pelo aumento da demanda, exige investimentos em projetos de automação (Dutra, 2012) e uma comunicação segura, eficiente e confiável entre os dispositivos para a integração de sistemas e processos (Souza, 2019).

A comunicação sempre representou um papel crítico na operação em tempo real do sistema de potência (Adamiak et al., 2009). Enquanto no mercado encontram-se diversos tipos de protocolos, suas diferenças e variedades têm como consequência a dificuldade de integração dos componentes, sendo necessário o uso de conversores para estabelecer a comunicação (Souza, 2019). Dessa forma, viu-se necessário um padrão para a transmissão de informações pelos equipamentos, o que foi atendido pela norma IEC 61850, a qual estabeleceu meios para garantir a interoperabilidade de comunicação entre dispositivos, trazendo também vantagens econômicas e de segurança, e tornando os sistemas "a prova de futuro"(Almeida, 2011).

Do ponto de vista hierárquico, as subsestações possuem três níveis: processo, *bay* e estação. Em Sistemas de Automação de Subestações (SAS), os Intelligent Electronic Devices (IEDs) são os principais dispositivos físicos microcontrolados capazes de proteger, controlar e monitorar o sistema, e estão localizados no *bay*. Estando no centro da estrutura hierárquica do SAS, qualquer falha acarreta efeitos operacionais e em performance mais severos em comparação aos outros níveis. Além disso, a utilização de novas tecnologias deixam o sistema mais suscetível a falhas e vulnerabilidades (Uzair, 2020).

Portanto, para garantir que o sistema elétrico opere de forma esperada, torna-se necessária a realização de testes para garantir seu funcionamento e de seus componentes. Para validação de instalações, é comum a utilização de equipamentos iguais aos que serão utilizados de fato. No entanto, essa alternativa pode acarretar em altos custos. Para testes utilizando relés de proteção de transformadores, por exemplo, em maio de 2022, a fabricante Schweitzer Engineering Laboratories (SEL) ofereceu opções a partir de U$3.200,00, enquanto um dispositivo para proteção, automação e controle do *bay* pode chegar a U$4.550,00 (SEL, 2022).

Embora possa fazer sentido em instalações de grande porte, esses investimentos são, muitas vezes, inviáveis para fins de pesquisa e desenvolvimento. Dessa forma, torna-se atraente a utilização de plataformas e softwares de código aberto que permitam validar a integridade dos processos para fins de verificação das funcionalidades. Há poucos trabalhos na literatura que especificam plataformas que podem ser replicadas e utilizadas como protótipos de desenvolvimento e pesquisa.

Visando detectar possíveis problemas, foram criadas ferramentas que possibilitassem a realização de testes de sistemas de automação com a utilização da norma IEC 61850, como simulações em tempo real utilizando Giga-Transceiver Network Communication Card (GTNET) em (Kuffel et al., 2010), e utilizando o framework OMNeT++/INET (León et al., 2016). Ainda, a incorporação de testes hardware-in-the-loop (HIL) permite avaliar situações cujos testes de campo seriam custosos, arriscados ou demandariam muito tempo. Um exemplo consiste na simulação de subsistemas que não podem ser incluídos fisicamente nos testes (Morais, 2017), podendo ser combinado à simulação em tempo real da norma IEC 61850 para o desenvolvimento e teste de sistemas embarcados, como em (Delavari et al., 2020).

Neste artigo, são avaliados os aspectos teóricos da emulação de um IED, embarcando o protocolo Manufacturing Message Specification (MMS), sugerido pela IEC 61850, em um sistema computacional de baixo custo. O objetivo do trabalho é implementar a integração de uma biblioteca com protocolo MMS, emulando um IED em um dispositivo embarcado de baixo custo. Para validar esta implementação, é apresentado um caso de emulação de estrutura de dados de um IED real e são feitos testes qualitativos de comunicação entre o IED simulado e um aplicativo que emula um cliente MMS. Este artigo representa uma pesquisa inicial sobre a análise de requisitos mínimos para implementar uma plataforma de baixo custo para teste de funcionalidades de IEDs. Busca-se especificar uma plataforma de *hardware* e software que permita emular funcionalidades de um IED real, criando um protótipo para fins didáticos, de pesquisa e de desenvolvimento.

2. ASPECTOS TEÓRICOS

2.1 Norma IEC 61850

A norma IEC 61850 foi desenvolvida pela *International Electronical Commission* (IEC) visando garantir a interoperabilidade entre dispositivos de fabricantes distintos, de modo a reduzir custos de engenharia e implementação (Dutra, 2012). Recomenda o uso da interface Ethernet e de três protocolos de comunicação: Manufacturing Message Specification (MMS), Generic Object Oriented Substation Event (GOOSE) e Sampled Value (SV), responsáveis pelo controle, proteção e monitoramento do sistema elétrico (Souza, 2019). A norma especifica os requisitos de qualidade dos sistemas de comunicação de um SAS, bem como requisitos de rede, condições ambientais, serviços auxiliares, requisitos de engenharia, tempo para mensagens, entre outros.

A IEC 61850 também especifica uma linguagem para descrição da configuração da subestação. Esta linguagem é baseada na Extensible Markup Language (XML), é denominada Substation Configuration Description Language (SCL) e define a padronização dos arquivos gerados pelas diversas ferramentas de desenvolvimento.

2.2 IED

Um IED é definido como um equipamento que possui um ou múltiplos microprocessadores integrados, com o objetivo principal de transmitir e receber dados ou sinais de controle de outro dispositivo externo, como transdutores, relés, entre outros (Coach, 2022).

Além de suas funcionalidades relacionadas à comunicação, os IED executam algoritmos de supervisão e controle. Também mantém registros de configuração e de funcionamento, tendo interface com sensores e atuadores, conforme é possível visualizar na Figura 1.

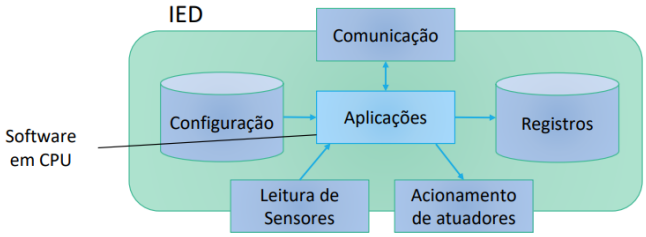

Figura 1. Abstração funcional de IED

Para se integrar ao sistema de automação, os IED possuem interfaces de comunicação e protocolos para comunicação com sistemas SCADA em uma comunicação vertical e com outros IEDs em uma comunicação horizontal. O termo IED é utilizado comumente para se referir a relés de proteção modernos, os quais utilizam microprocessadores para suas funcionalidades relacionadas à proteção e controle. Além disso, estes IED devem seguir o padrão determinado pela norma IEC 61850 para automação de subestações (iGrid T&D, 2022).

2.3 Protocolo MMS

O protocolo MMS é um padrão internacional que lida com sistemas de mensagens para transferência de dados em tempo real entre dispositivos. Trata-se de protocolo de camada de aplicação, possibilitando o controle e comunicação de IEDs (Souza, 2019).

A norma IEC 61850 define o protocolo MMS como comunicação do tipo servidor/cliente, sendo usado para troca

de informações entre IEDs e dispositivos de alto nível (como softwares supervisórios), através de interface Ethernet (HIL, 2022). Devido a esta característica de aplicação, é considerado um protocolo vertical. Com a configuração adequada, pode ser integrado a qualquer modelo de simulação, assim como outros protocolos definidos na IEC 61850 (Opal-RT, 2022).

2.4 Interface Ethernet

Ethernet é uma interface hardware que permite que computadores se comuniquem através de redes conectadas utilizando dispositivos como roteadores, switches e hubs para direcionar o tráfego.

A simplicidade da interface Ethernet, junto a sua habilidade de suportar velocidades maiores e ainda manter a compatibilidade com aplicações anteriores permitiu que a Ethernet crescesse junto aos avanços tecnológicos (Breeden, 2022).

Atualmente, a Ethernet é a tecnologia Local Area Network (LAN) mais utilizada no mundo. Isso se deve a sua facilidade de entendimento, manutenção e por permitir implementações de rede a baixo custo, além de fornecer flexibilidade em termos de topologias permitidas (GeeksforGeeks, 2022).

2.5 Sistemas Embarcados

Sistemas embarcados são desenvolvidos com a integração de hardware e firmware para realização de tarefas específicas (CPQD, 2022), possuindo capacidade de processamento de dados adequada à aplicação. Estes sistemas computacionais fazem parte do projeto de sistemas eletrônicos de dispositivos industriais e produtos de consumo.

Podem ser gerenciados por microcontroladores ou processadores de sinais digitais, circuitos integrados para realização de atividades específicas, entre outros. Seus sistemas de processamento são integrados com componentes dedicados, cuja estrutura básica inclui sensores, conversores A/D e D/A e atuadores (Heavy.AI, 2022).

As linguagens C e C++ são comumente utilizadas para a programação de sistemas embarcados. O programa é desenvolvido utilizando um ambiente de desenvolvimento integrado (IDE), compilado por um software compilador específico para o sistema embarcado, o qual pode estar integrado à IDE. O programa compilado é enviado para armazenamento na memória interna do sistema embarcada e é executado quando a fonte de alimentação do sistema está ativa. O Arduino, por exemplo, é programado utilizando uma variação da linguagem C++ com adições de métodos e funções especiais e utilizando o software Arduino Integrated Development Environment (Arduino IDE), o qual é um editor com suporte a bibliotecas integradas e permite compilar e carregar os programas de forma fácil na placa Arduino conectada ao computador (Copes, 2022).

Uma vez que sistemas embarcados devem responder a mudanças no ambiente em que estão, esses devem processar os dados de entrada e computar resultados em tempo real buscando minimizar seu tempo de resposta e evitar atrasos.

Tabela 1. Especificações técnicas do ESP32

CPU	Processador Xtensa® Dual-Core 32-bit LX6 com *clock* de 80 a 240MHz
Memória FLASH	4MB
Memória RAM	520KB
Memória ROM	448KB
Wi-Fi	2,4 GHz 802.11 b/g/n/e/i (802.11n até 150 Mbps)
Bluetooth	Bluetooth Low Energy v4.2 (BLE)
Fonte de alimentação	3,0V 3,6V (80mA a 500mA) DC Entrada com interface micro USB
Periféricos de baixo nível	34 GPIOS de 3.3V e 12mA
Sistemas Operacionais	*Bare metal*

2.6 ESP32

O ESP32, desenvolvido e comercializado pela empresa *Espressif Systems*, é um *System on Chip*(SOC) com 1 ou 2 processadores Xtensa LX6 de 32 bits, memória Flash e SRAM e funcionalidades para comunicação sem fio via *Wi-Fi* e *Bluetooth*. São comercializados módulos com variações e funcionalidades diversificadas, como conectores para antenas e capacidades de memória *flash* para atender necessidades de aplicações distintas.

O ESP32 é um módulo *bare metal*, ou seja, consiste em um servidor sem um sistema operacional base. A utilização de sistemas operacionais visa flexibilidade e eficiência, porém, para isso, são inseridas camadas de abstração adicionais, resultando em redução de performance devido à virtualização dos recursos. Portanto, a utilização de módulos *bare metal* se torna atrativa pois a ausência de diversas camadas de *software* pode proporcionar melhor desempenho.

Projetos desenvolvidos para sistemas operacionais específicos ou que dependam de conjuntos de ferramentas, denominadas *toolchains*, devem ser adaptados para possibilitar a compilação e execução na placa.

A página oficial da fabricante fornece uma documentação completa para a implementação do sistema de compilação e conceitos necessários para o desenvolvimento. Além disso, a empresa fornece o *framework* ESP-IDF que permite o desenvolvimento de aplicações para sua linha de dispositivos.

A placa ESP32-DevKitM-1 é um dos modelos disponibilizados e comercializados pela *Espressif*. Esse fabricante denomina de *DevKits* placas de circuito impresso contendo o módulo SOC da fabricante e circuitos periféricos. As placas possuem conexão micro-USB, facilitando o desenvolvimento e prototipagem, e ainda disponibilizam botões de *boot* (permitindo alternar entre modo de *download* e *upload*) e *reset*, além de indicadores de alimentação e terminais para conexão com outros dispositivos. Suas especificações técnicas estão descritas na Tabela 1.

2.7 Orange Pi Zero2

A empresa *Orange Pi* fornece soluções de código aberto de hardware e software para desenvolvimento (Pi, 2022). Com a proposta de fornecer produtos de qualidade e com baixo custo, suas placas possuem módulo *Wi-Fi* e possibilidade de executar diversos sistemas operacionais, permitindo a utilização em diferentes projetos.

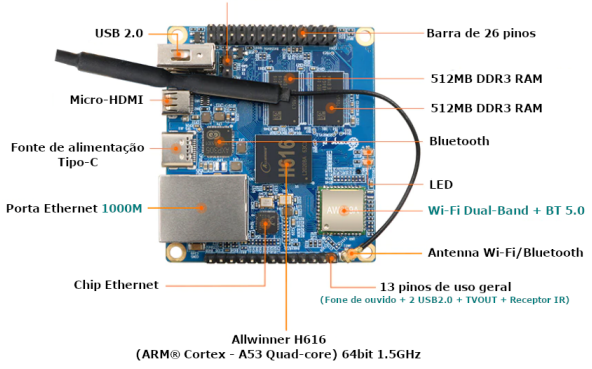

Figura 2. Orange Pi Zero2

Tabela 2. Especificações técnicas do Orange Pi Zero 2

CPU	Processador Allwinner H616 64-bit Quad-core Cortex-A53 de alta performance
GPU	Mali G31 MP2
Memória (SDRAM)	1GB DDR3 (Compartilhada c/ GPU)
Armazenamento	Slot para cartão microSD Memória SPI-Flash de 2MB
Rede	Suporta Ethernet TX 1000-100-10Mb/s
Wi-Fi + Bluetooth	Chipset AW859A Suporta IEEE 802.11 a/b/g/n/ac e BT5.0
Saída de vídeo	Micro HDMI 2.0a até 4K@60fps
Saída de áudio	Micro HDMI ou Porta de áudio de 3,5mm
Fonte de alimentação	Entrada 5V/3A com interface USB C
Portas USB 2.0	3 × USB 2.0 HOST
Periféricos de baixo nível	26 pinos com portas I2C, SPI, UART e múltiplas portas GPIO
Porta serial para debug	UART-TX, UART-RX e GND
LED	Led de alimentação e led de status
Receptor Infravermelho	Suporta controle remoto infravermelho
Sistemas Operacionais Suportados	Android10, Ubuntu, Debian, Armbian

A Figura 2 mostra o modelo Orange Pi Zero2. De acordo com a fabricante, o modelo suporta utilização dos sistemas Android 10, Ubuntu, Debian e Armbian. Além da conexão *Wi-Fi*, a placa também possui porta Ethernet para conexão de rede, além de portas USB 2.0, e é alimentada através de uma interface USB tipo C, comum em *smartphones*. Ainda, a solução possui entradas e saídas de uso geral (GPIO) e possibilidade de comunicação direta com sensores e atuadores, através de interfaces SPI e I2C. Suas especificações técnicas estão descritas na Tabela 2.

2.8 Revisão Bibliográfica

A seleção das referências para a revisão bibliográfica foi realizada buscando as palavras-chave "*IED Emulation*", "*Power Hardware in the Loop*"no Portal de Periódicos da CAPES. Foram priorizados artigos publicados nos últimos cinco anos.

A comunicação se tornou indispensável para a operação segura de sistemas de potência, e uma forma de obter uma rede completamente conectada é através da padronização dos modelos de troca de mensagens. Dessa forma, a IEC 61850 é, atualmente, o padrão de comunicação mais promissor, introduzido inicialmente para padronizar a automação de subestações.

Devido à complexidade da norma IEC 61850, seus testes podem requerer muito tempo, compra de bibliotecas de softwares e hardwares compatíveis com custos elevados, ou modificação de IEDs já existentes (Blair et al., 2013). Existem diferentes ferramentas de teste da norma disponíveis, variando de código-aberto a softwares licenciados (Ustun, 2021).

Em (Blair et al., 2013), é proposta uma solução de código-aberto que possibilita a prototipagem rápida de esquemas de comunicação. A plataforma gera modelos de dados automaticamente e os códigos de comunicação requeridos para um ou mais IEDs. É implementada a comunicação utilizando os protocolos GOOSE e SV, de modo a garantir a interoperabilidade com outros dispositivos que implementam a IEC 61850. A plataforma foi a primeira implementação de código-aberto da norma, sendo adequada para aplicações em tempo real, como esquemas de proteção, e está disponível para fins de pesquisa e educacionais.

O projeto libIEC61850 fornece uma biblioteca de código aberto para os protocolos de comunicação MMS, GOOSE e SV. O objetivo do projeto é fornecer uma implementação da IEC 61850 que possa ser executada em sistemas embarcados, garantindo a portabilidade entre sistemas computacionais (Zillgith, 2022). Essa solução possibilita a construção de softwares cliente-servidores compatíveis com a IEC 61850 (libIEC61850, 2022).

Existem ainda ferramentas pagas de emulação e modelagem da norma. Alguns exemplos são analisados em (Ustun, 2021), como o software *Infotech*, o qual fornece diversas ferramentas de teste da norma IEC 61850 com uma interface simples que permite enviar e receber mensagens GOOSE e SV com diferentes variáveis e configurações.

3. METODOLOGIA PROPOSTA

Neste trabalho, foram elaboradas três propostas de solução para embarcar a biblioteca libIEC61850 em um sistema computacional de baixo custo. Esta biblioteca fornece a implementação da norma IEC 61850 com o procotolo MMS, podendo ser executada em sistemas embarcados (Zillgith, 2022). Todas as abordagens e resultados aqui relatados estão detalhados em Na (2022).

A primeira alternativa com a placa ESP32 utiliza o software *Visual Studio Code* e o *framework* ESP-IDF para desenvolver, compilar e realizar o *upload* da aplicação na placa sem sistema operacional, ou seja, *bare metal*. A documentação do *framework* ESP-IDF (Espressif, 2022) fornece a opção de utilizar projetos terceiros como componentes. Como o CMake é utilizado em diversos projetos de código aberto nas linguagens C e C++, é possível importá-los para que suas funcionalidades possam ser utilizadas na forma de componentes. Além da possibilidade de utilizar bibliotecas de terceiros, a documentação (Espressif, 2022) fornece a possibilidade de utilizar bibliotecas pré-compiladas estáticas com formato `.a` através da função `add_prebuilt_library`. O comando permite especificar o nome que será utilizado como referência para a bilioteca, o diretório do arquivo pré-compilado, além de possibilitar especificar dependências da biblioteca, a qual deve ser compilada previamente para o mesmo *target* do projeto que a consumirá, ou seja, utilizando a *toolchain* do ESP-IDF.

A segunda alternativa utiliza o sistema operacional para microcontroladores NuttX (NuttX, 2022), baseado no ker-

nel Linux, como base para receber a aplicação contendo a libIEC61850. Para isso, deve-se utilizar uma biblioteca pré-compilada, sendo o processo similar ao realizado no sistema Linux. Para utilizar o NuttX, deve-se criar um ambiente para o processo de *build* e *flash*, além de programar a placa com as ferramentas da fabricante utilizando linguagem Python. A *toolchain* da Espressif deve ser instalada utilizando *cross compiling*, isto é, deve-se compilar o código para um sistema ou plataforma diferente do qual o compilador está sendo executado.

A terceira alternativa utiliza a placa Orange Pi, a qual embarca um sistema operacional Armbian, baseado em Linux. Com isso, seria possível compilar a biblioteca libIEC61850 diretamente sem alterações para adaptá-la ao seu sistema. O site oficial do Orange Pi (Pi, 2022) disponibiliza as imagens dos sistemas operacionais suportados, além de instruções para o *flash* do sistema no cartão microSD. Para isso, a imagem do sistema desejado pode ser baixada na página da placa. Para o objetivo proposto, o sistema Linux deverá ser utilizado, uma vez que é suportado pela biblioteca libIEC61850.

4. RESULTADOS

A partir da análise das metodologias apresentadas, foram feitos testes para estudar a viabilidade de cada uma das opções apresentadas, as quais estão detalhadas nas subseções a seguir.

4.1 Utilizando ESP32

A primeira abordagem adotou a utilização de bibliotecas de terceiros como componentes ou bibliotecas estáticas pré-compiladas. No entanto, ao compilar o código com o componente gerado, erros são gerados devido à existência de referências à funções exclusivas de distribuições Linux com alvos em outros sistemas computacionais como ARM e x86. Como o ESP32 não possui um sistema operacional, mesmo que a biblioteca seja compilada para o sistema Linux, por exemplo, o processo de compilação resulta em erro, pois os caminhos para as funções necessárias não são encontrados. Se torna necessário adaptar a biblioteca para incluir referências à camada de abstração de hardware (do inglês HAL) do ESP32, que trata chamadas aos recursos de rede de forma distinta de outras plataformas de hardware como ARM ou x86. Dado o esforço demandado para esta tarefa e o escopo do trabalho, decidiu-se tentar outras abordagens. Na segunda abordagem, utilizando o Nuttx, é introduzido um sistema operacional ao ESP32. No entanto, o sistema NuttX não é suportado pela biblioteca libIEC61850, o que resulta nas mesmas complicações apresentadas para as abordagens anteriores. Dessa forma, para as duas metodologias apresentadas utilizando a placa ESP32, seria necessário adaptar o código disponibilizado da biblioteca libIEC61850 para não depender do sistema operacional - como é recomendado na documentação do CMake - ou estar habilitado a utilizar a *toolchain* da fabricante Espressif ou do sistema NuttX. Considerando o escopo do trabalho, esta dependência da biblioteca de um sistema operacional alvo compatível tornou estas abordagens inviáveis.

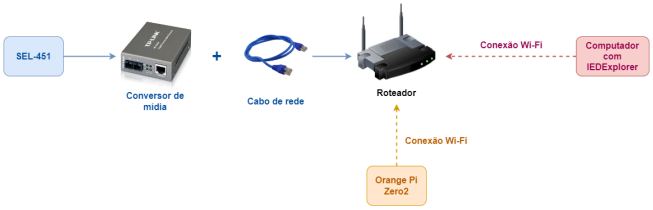

Figura 3. Configuração utilizada para os testes

4.2 Utilizando Orange Pi Zero2

Com a placa configurada para a distribuição disponível no site do fabricante, foi possível baixar a biblioteca libIEC61850 e compilar um dos exemplos disponibilizados utilizando o comando `make` no terminal de comandos. Como o sistema operacional utilizado foi Linux, a biblioteca será compilada com esse sistema como *target*, a menos que outro seja especificado. Esta abordagem obteve sucesso e foi adotada nos testes de validação. Vale notar que o uso de distribuições do tipo CLI ou SERVER permite disponibilizar mais recursos computacionais para o IED emulado, permitindo um desempenho melhor da plataforma.

4.3 Testes de Validação

Para a validação do projeto desenvolvido, foi extraído o arquivo de configuração `.cid` do IED de modelo SEL-451 (fabricante SEL). A partir do arquivo obtido, foram gerados os arquivos de configuração `static_model.c` e `static_model.h` correspondentes através da ferramenta *Model Generator* da biblioteca libIEC61850. Os testes realizados utilizaram como base o exemplo *server example basic io* fornecido pela biblioteca. A partir deste, foram realizadas adaptações de forma a configurá-lo de acordo com a estrutura do IED utilizado para validação. Para garantir que o funcionamento do dispositivo emulado está dentro do esperado, foram utilizados softwares que permitissem testar as funcionalidades esperadas. O software IEDExplorer fornece uma ferramenta gratuita e de código aberto para fins de teste e educacionais, possibilitando a conexão a dispositivos dentro dos padrões na norma IEC 61850 através do protocolo de comunicação MMS, atuando como cliente. O software Wireshark é um analisador de protocolos e foi utilizado por sua capacidade de identificar mensagens do protocolo MMS, permitindo validar a implementação. A Figura 3 mostra os dispositivos e interfaces utilizados. O IED SEL-451 estava conectado ao roteador por um *switch* via cabo, enquanto o computador que rodava os softwares para validação e o Orange Pi Zero2 estavam conectados à mesma rede via *Wi-Fi*.

Foi utilizado o software IEDExplorer para validar a comunicação via protocolo MMS uma vez que o servidor estava em execução no IED emulado e disponível para conexão. Para isso, foi utilizado o endereço IP da placa Orange pi Zero 2. Uma vez conectado, o software mostra a árvore com os objetos e nós do IED emulado. Assim como foi possível comunicar o IED emulado com o software IEDExplorer, também é possível realizar tal comunicação com o IED real utilizando seu endereço IP. O software possibilita o mapeamento da estrutura de ambos os dispositivos, permitindo que seja feita a comparação de seus componentes.

Foi possível verificar que o IED emulado a partir de um arquivo de configuração de um dispositivo real possui a mesma estrutura para uso do protocolo MMS. Toda a estrutura foi emulada com sucesso, sem perda de dados em todas as tentativas de comunicação.

Utilizando também o software Wireshark, foi possível monitorar o tráfego da rede ao realizar comandos através do software IEDExplorer, tanto no IED real como no IED emulado. Através da análise dos pacotes enviados, é possível observar que o protocolo de comunicação utilizado foi o MMS, mostrando que a utilização da biblioteca libIEC61850 permite a emulação de um IED embarcando o protocolo, com o dispositivo atuando como servidor, possibilitando a conexão com clientes e o processamento de comandos. Todos os pacotes do IED emulado estavam íntegros, funcionais, sem erros estruturais e compatíveis com os pacotes enviados pelo IED real.

5. CONCLUSÃO

A utilização de dispositivos de baixo custo com alta capacidade computacional é atraente para cenários de emulação de IEDs, considerando o alto custo dos dispositivos reais. Neste trabalho, foram apresentadas implementações de uma biblioteca de código aberto que implementa o protocolo MMS, sugerido pela norma IEC 61850 para a comunicação entre IEDs e sistemas SCADA. A dependência de um sistema operacional alvo, tornou a utilização da plataforma ESP32 inviável, devido à necessidade de customizações nas dependências da biblioteca. A plataforma Orange Pi, baseada em processador ARM e com suporte a sistema operacional Linux, permitiu a emulação da estrutura de dados de um IED real. Testes comprovaram seu funcionamento na leitura e no envio de comandos, comparando os pacotes recebidos e enviados entre um software cliente MMS, um IED real e o IED emulado.

REFERÊNCIAS

Adamiak, M., Baigent, D., and Mackiewicz, R. (2009). Iec 61850 communication networks and systems in substations: An overview for users. URL `https://www.gevernova.com/grid-solutions/multilin/journals/issues/spring09/iec61850.pdf`.

Almeida, E.M.d. (2011). Norma iec 61850 - novo padrão em automação de subestações.

Blair, S.M., Coffele, F., Booth, C.D., and Burt, G.M. (2013). An open platform for rapid-prototyping protection and control schemes with iec 61850. *IEEE transactions on power delivery*, 28(2), 1103–1110.

Breeden, J. (2022). What is ethernet? URL `https://www.networkworld.com/article/3657734/what-is-ethernet.html`.

Coach, E. (2022). Intelligent electronic devices (ied) in scada. URL `https://electronicscoach.com/intelligent-electronic-devices.html`.

Copes, F. (2022). Introduction to the arduino programming language. URL `https://flaviocopes.com/arduino-programming-language/`.

CPQD (2022). Sistemas eletrônicos embarcados. URL `https://www.cpqd.com.br/inovacao/sistemas-eletronicos-embarcados/`.

Delavari, A., Brunelle, P., and Mugombozi, C.F. (2020). Real-time modeling and testing of distance protection relay based on iec 61850 protocol. *Canadian Journal of Electrical and Computer Engineering*, 43(3), 157–162. doi:10.1109/CJECE.2020.2968404.

Dutra, R.d.S. (2012). Iec 61850 - comunicação e automação de subestações de energia elétrica.

Espressif (2022). Build system. URL `https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/build-system.html`.

GeeksforGeeks (2022). Lan technologies | ethernet. URL `https://www.geeksforgeeks.org/local-area-network-lan-technologies/`.

Heavy.AI (2022). Embedded systems. URL `https://www.heavy.ai/technical-glossary/embedded-systems`.

HIL, T. (2022). Iec 61850 mms protocol. URL `https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iec_61850_mms_protocol.html`.

iGrid T&D (2022). What is ied? URL `https://www.igrid-td.com/smartguide/gridandsubstationautomation/ied-intelligent-electronic-device/`.

Kuffel, R., Ouellette, D., and Forsyth, P. (2010). Real time simulation and testing using iec 61850. In *2010 Modern Electric Power Systems*, 1–8.

León, H., Montez, C., Stemmer, M., and Vasques, F. (2016). Simulation models for iec 61850 communication in electrical substations using goose and smv time-critical messages. In *2016 IEEE World Conference on Factory Communication Systems (WFCS)*, 1–8. doi: 10.1109/WFCS.2016.7496500.

libIEC61850 (2022). libiec61850 on github. URL `https://github.com/mz-automation/libiec61850`.

Morais, D.P.d. (2017). Infraestrutura de hardware in the loop para testes funcionais de sistemas de monitoramento remoto.

Na, M.W. (2022). Emulação de ied embarcando o protocolo mms em um sistema computacional de baixo custo. Trabalho de conclusão de curso, Universidade Federal Fluminense, Niterói, RJ.

NuttX (2022). Nuttx documentation. URL `https://nuttx.apache.org/docs/latest/index.html`.

Opal-RT (2022). Iec 61850 mms. URL `https://www.opal-rt.com/software-communication-protocols/iec-61850-mms/`.

Pi, O. (2022). Orange pi. URL `http://www.orangepi.org/`.

SEL (2022). Schweitzer engineering laboratories. URL `https://selinc.com/pt/solutions/control-hil/`.

Souza, D.L. (2019). Integração dos protocolos mms (iec 61850) e modbus tcp para aplicação em acionamentos de equipamentos de média tensão.

Ustun, T.S. (2021). A critical review of iec 61850 testing tools. *Sustainability (Basel, Switzerland)*, 13(11), 6213.

Uzair, D.M. (2020). An overview of the implementation issues of iec 61850 based substation automation system (sas). *The Journal of Engineering, Science and Computing.*

Zillgith, M. (2022). libiec61850 website. URL `https://libiec61850.com/`.