

Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica - FEELT
Pró- Reitoria de Pesquisa e Pós-graduação - PROPP

Thiago Fernando Cuevas Mestanza

Relatório Final de Iniciação Científica
Desenvolvimento e Implementação de um Gateway Comunicador entre Protocolos
Modbus TCP/RTU

Uberlândia, MG

2022

Thiago Fernando Cuevas Mestanza

**Desenvolvimento e Implementação de um Gateway Comunicador entre Protocolos
Modbus TCP/RTU**

Relatório final do projeto de Iniciação Científica voltado para o Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Uberlândia. Edital DIRPE Nº 10/2021 - PIBIC/FAPEMIG/UFU.

Orientador: Prof. Dr. Renato F. Fernandes Jr.

Uberlândia, MG

2022

RESUMO

A constante busca pela otimização de processos tem sido um dos principais estímulos para os avanços técnico-científicos. O campo industrial é um dos setores que mais se nutre desses progressos, ficando evidente no âmbito das redes industriais, peça fundamental hoje para a construção da indústria moderna. Nesse sentido, a efervescente evolução desse setor faz com que as tecnologias desenvolvidas, rapidamente, substituam umas às outras, tornando algumas obsoletas ou incomunicáveis quando comparadas com os equipamentos mais modernos. Dessa forma, este trabalho teve como objetivo desenvolver um Gateway Modbus TCP/RTU capaz de servir como alternativa de baixo custo para integração de equipamentos mais antigos, que já estão disponíveis na indústria, com equipamentos mais modernos, no intuito de prolongar a vida útil desses equipamentos e dar a possibilidade de expandir os limites e conexões de uma rede industrial convencional. Como resultado, o dispositivo desenvolvido neste trabalho foi testado e aplicado com êxito na comunicação com equipamentos industriais.

1. INTRODUÇÃO

Atualmente as redes baseadas em ethernet tem dominado o setor industrial frente a outras redes com barramentos de campo já consolidados, e que tiveram sua concepção no início do século XXI. Esta transição para a ethernet industrial é impulsionada pela evolução dos equipamentos de rede ethernet, pela integração entre os diferentes níveis de automação dentro da indústria, e pelas tecnologias de IoT (*Internet of Things*) e Indústria 4.0 [1].

Dentro do conceito atual de Indústria 4.0, as tecnologias de ethernet industrial são responsáveis pela integração dos sistemas de tecnologia da informação (TI), com os sistemas de tecnologia operacional (TO). A TO se refere às redes industriais que fazem interface com os processos. Já, a TI engloba todos os componentes de software e hardware, que trabalham no nível de gestão da informação em uma rede ethernet industrial, indo desde o planejamento e execução até a verificação e alteração das várias etapas que compõe o processo industrial [1, 2].

Nesse sentido, as redes industriais são responsáveis por interligar as várias camadas do processo industrial, desde o nível mais baixo, chamado de nível de chão de fábrica, onde estão localizados os sensores e atuadores responsáveis pelo andamento do processo produtivo, até o nível mais alto de controle central. Através das sub redes, todos os equipamentos da rede podem comunicar-se, mesmo estando distantes a dezenas de quilômetros [3]. A informação que percorre o sistema de comunicação, desde os sensores até os controladores, é dada em protocolos. Entre os vários protocolos disponíveis no mercado, um dos mais usados é o Modbus, que, devido a sua facilidade de implementação e baixo custo, é um protocolo amplamente utilizado em diversos controladores e ferramentas para desenvolvimento de sistemas supervisórios [4].

O protocolo Modbus surgiu em 1979 pela empresa Modicon, atual Schneider Electric, e atualmente é controlado pela associação Modbus.org. O Modbus é um protocolo industrial de camada de aplicação e, por especificar funções somente de aplicação, pode ser utilizado juntamente com outros protocolos de camadas inferiores. Dentro do ambiente industrial existem diversos tipos de equipamentos que se comunicam pelo protocolo Modbus, como sensores, atuadores, controladores e até sistemas de supervisão. Existem vários padrões de comunicação Modbus, como RTU (Remote Terminal Unit), ASCII, TCP (*Transmission Control Protocol*), Plus, entre outros. Os protocolos mais utilizados são Modbus RTU para o padrão RS232, RS485 e RS422 e o Modbus TCP para o padrão Ethernet [5]. O Modbus é um

protocolo Mestre/Escravo, no qual o Mestre é responsável por fazer as requisições dentro da rede para os Escravos. Os Escravos somente respondem às perguntas feitas pelo Mestre [5].

A mensagem de dados Modbus RTU é dividida em quatro campos. O primeiro campo é o Endereço, responsável por identificar, através de um numero, o dispositivo Escravo. O segundo campo representa o código da função, que identifica qual função está sendo requisitada pelo Mestre. O terceiro campo é o de dados, e varia de acordo com a função. Por fim, o ultimo campo representa a checagem de erro do pacote (CRC). A figura 1 mostra um esquemático, bem como um exemplo de comunicação Modbus RTU [5].

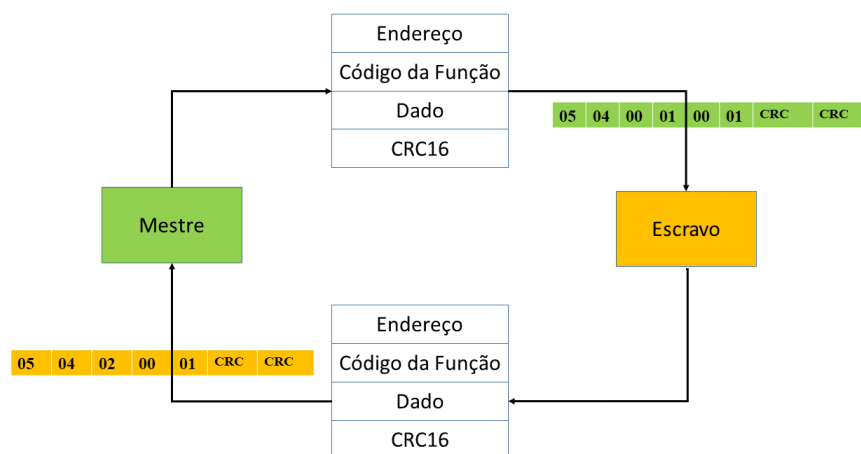


Figura 1: Diagrama de mensagem Modbus RTU.

Com o advento e popularização da rede Ethernet, e de sistemas de comunicação mais velozes, o protocolo Modbus foi atualizado para suportar o meio físico Ethernet, dando origem ao Modbus TCP. Atualmente, o Modbus TCP é a forma do Modbus de maior difusão no mercado [7]. Em redes Modbus TCP, o protocolo Modbus é aplicado com o padrão Ethernet sob a pilha de protocolos TCP/IP. A figura 2 mostra a pilha de camadas usada pelo Modbus e o pacote de dados Modbus incluído no pacote de dados TCP [5].

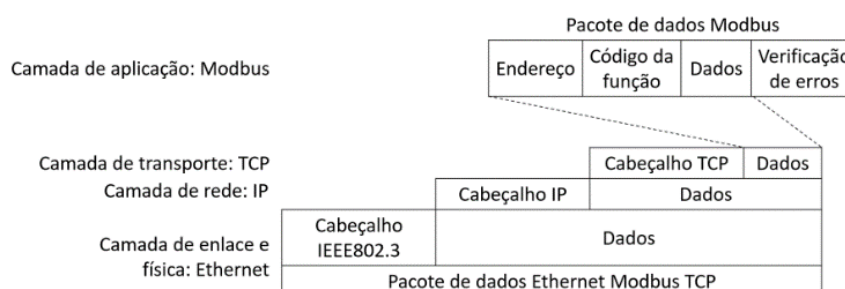


Figura 2: Representação do Modbus TCP. Fonte: [5]

Observa-se, pela figura 3, que o Modbus TCP tem similaridades com o Modbus RTU, sendo que o TCP apresenta a inclusão de um cabeçalho específico do protocolo, e a remoção do CRC presente no Modbus RTU.

O conceito de Gateway é um dos mais importantes dentro da lógica de IoT, já que consiste em um proxy, isto é, um intermediador, entre a rede de detecção e as camadas de aplicação, estabelecendo a comunicação entre sensores, e as camadas superiores da rede [6]. Com Gateways, é possível conectar equipamentos localizados em redes diferentes e que se comunicam em padrões ou protocolos distintos, expandindo assim a rede de conexão entre os diferentes dispositivos. No ambiente industrial, Gateways cumprem a função de intercessores de comunicação remota, entre sensores, atuadores, transmissores, com sistemas de controle e supervisão da planta de produção. Por essa razão, os Gateways precisam suportar uma ampla variedade de interfaces de entrada e saída, incluindo conexões com fio, sem fio, serial e Ethernet [7].

A figura 3 mostra o cenário de aplicação do Gateway dentro do Laboratório de Redes e Instrumentação Industrial da Universidade Federal de Uberlândia. Este laboratório conta com vários equipamentos Modbus RTU, como inversores, CLP's e Soft Starts que só podem se comunicar com equipamentos Modbus TCP através de um Gateway funcionando como Escravo TCP e Mestre RTU.

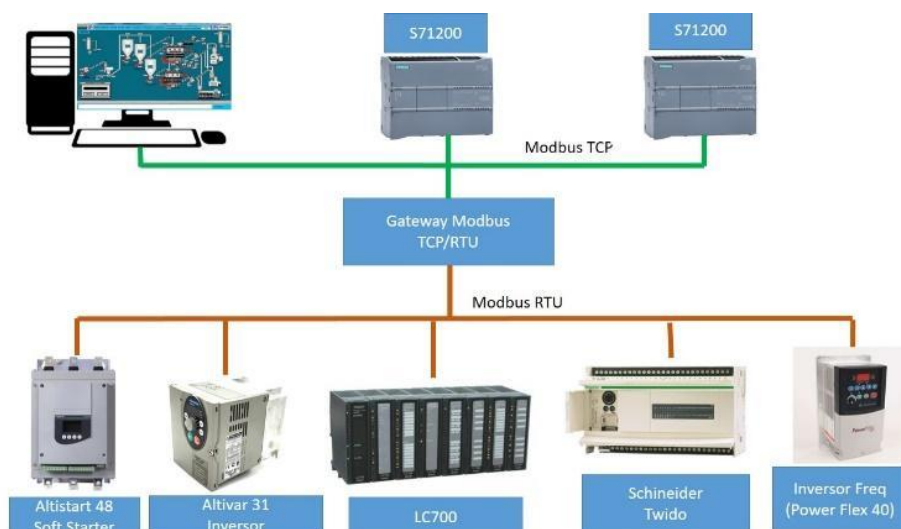


Figura 3: Esquemático de rede de equipamentos Modbus comunicados por um Gateway Modbus TCP/RTU.

2. OBJETIVOS

Esta pesquisa se propôs a desenvolver um *Gateway*, isto é, um intercessor de protocolos, capaz de mediar a troca de informações entre uma rede Modbus RTU, composta de equipamentos mais antigos, e uma rede Modbus TCP, composta de equipamentos mais modernos.

O desenvolvimento deste tipo de dispositivo atende uma demanda presente no setor industrial e acadêmico, que consiste na imprevista necessidade de aposentar prematuramente equipamentos com alto potencial de uso. Nesse sentido, o Gateway desenvolvido prolonga a usabilidade dos dispositivos disponíveis, reduz gastos, e no contexto acadêmico, permite o contato de tais equipamentos com mais membros da comunidade.

3. METODOLOGIA

O Gateway Modbus TCP/RTU foi desenvolvido utilizando a placa BeagleBone Black (BBB) que conta com um microprocessador ARM A8 denominado Sitara AM3358. A preferência pela BBB surge pela possibilidade de embarcar nela um sistema operacional Linux, sendo possível assim, trabalhar com aplicações mais complexas e sistemas multitarefas. Além disso, a BBB já conta com uma porta Ethernet embutida na placa, o que, a nível de hardware, facilita a montagem do Gateway.

Quanto ao desenvolvimento do Firmware (software embarcado na BBB), visando atingir os objetivos mencionados e tentando responder às questões surgidas durante o desenvolvimento deste projeto, foram realizadas várias etapas até atingir o objetivo final da pesquisa. A seguir serão descritas cada uma das etapas.

3.1 Desenvolvimento de um Escravo Modbus TCP

Nesta primeira etapa foi desenvolvido um Firmware, baseado na biblioteca de código aberto FreeModbus, capaz de receber e interpretar as requisições de um Mestre TCP. Várias bibliotecas de código aberto disponíveis online suportam a comunicação Modbus TCP como Escravo. Por essa razão, foi o ponto de partida do desenvolvimento do Gateway. A validação do funcionamento do Firmware foi feita através de testes utilizando o software simulador Mestre Modbus, ModScan, onde um ou mais valores, em um endereço específico, eram escritos no código Escravo e deveriam ser lidos pelos simuladores.

3.2 Desenvolvimento de um Escravo Modbus RTU

Em seguida, desenvolveu-se um Firmware capaz de receber e interpretar requisições do simulador Mestre Modbus, ModScan, no protocolo Modbus RTU. Por mais que a função Escravo Modbus RTU não seja acionada no estágio final do Gateway desenvolvido, este modo de operação foi implementado a fim de compreender mais sobre o protocolo. Os testes foram feitos com os meios físicos RS485 e RS232. Para trabalhar com o protocolo Modbus RTU, seja com os barramentos RS232 ou RS485, foram necessários dispositivos e módulos complementares. A figura 4 mostra os componentes que foram utilizados para efetivar a comunicação para ambos os barramentos.



Figura 4: Componentes utilizados e citados na seção de metodologia do trabalho. (a) Conversor de dados TTL para RS-485; (b) Conversor USB Serial FT232RL FTDI para RS-232; (c) Conversor de Nível Lógico RC;

Na validação do funcionamento do Firmware conectou-se a UART 4 da BBB hora com o circuito adaptador RS485 (figura 3.a), e hora com o circuito adaptador RS232 (figura 3.b). O uso do adaptador RS485, fez necessária a presença de outros componentes eletrônicos, como uma fonte de alimentação 5V e um conversor de tensão 5V-3V3 (figura 3.c) para converter o sinal que sai da BBB (3V3) para o sinal que chega ao modulo RS-485 (5V). Nos testes, seguindo a mesma lógica da etapa anterior, valores foram escritos no firmware Escravo Modbus RTU, e consequentemente deveriam ser lidos pelo simulador Mestre ModScan.

3.3 Desenvolvimento do Gateway funcionando como Escravo Modbus TCP e Mestre Modbus RTU

Na sequência, desenvolveu-se um Firmware capaz de fazer requisições para equipamentos Escravos da rede, no protocolo Modbus RTU e receber comandos de um Mestre no protocolo Modbus TCP, configurando assim, o Gateway Modbus TCP/RTU. A partir deste

momento, substituiu-se a linguagem de programação C pelo Python. Em relação aos meios físicos utilizados, usaram-se, para a comunicação Modbus RTU, os dois barramentos, RS485 e RS232. A primeira validação do funcionamento do Firmware, com os dois adaptadores, foi feita através do software simulador Escravo Modbus RTU, ModSim e Mestre Modbus, ModScan. A seguir, foram feitos testes com Controladores Lógico Programáveis (CLP's) de ambiente industrial. Estes equipamentos estão mostrados na figura 5 a 7.

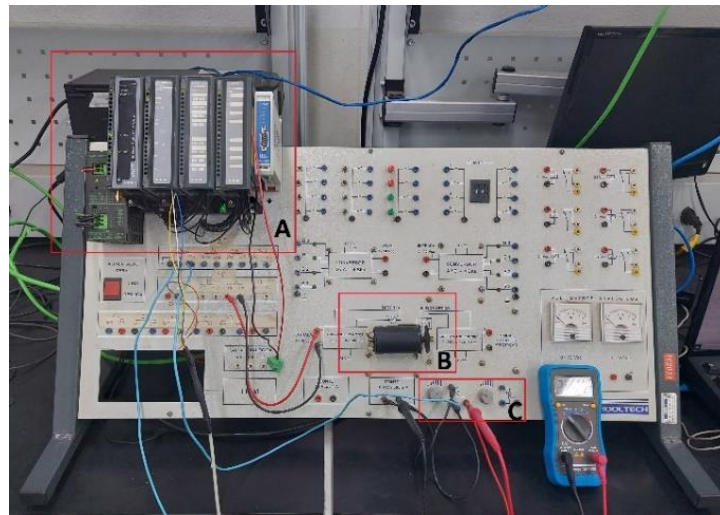


Figura 5: Painel com CLP LC700 (a) da Smar e os demais componentes já instalados - Motor CC (b) e Potenciômetros (c).

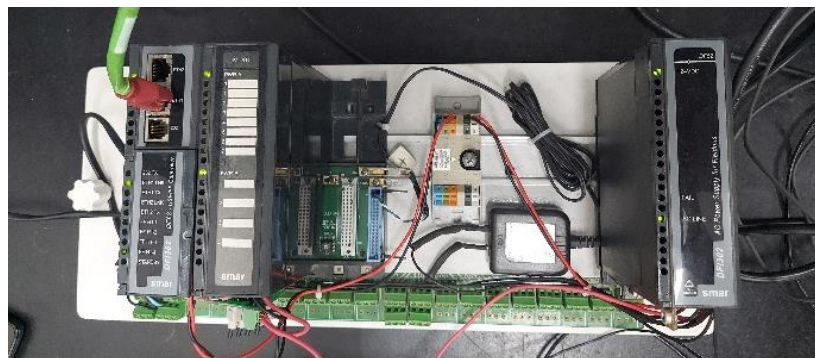


Figura 6: CLP DF63 da Smar.



Figura 7: CLP S71200 da Siemens.

No primeiro teste com CLP's testou-se o Gateway Modbus TCP/RTU sendo Escravo do simulador ModScan e Mestre do CLP LC700, que somente possui portas RS232 e RS485. Ele se encontra instalado em um painel com cartões de entrada e saída analógica ligados a um kit didático com potenciômetros (figura 5.c), motores (figura 5.b) e LED's. Os testes, nesta etapa, consistiam no simulador Mestre, ModScan, ser capaz de requisitar os dados de tensão em um potenciômetro e, ao mesmo tempo, escrever um valor de tensão a ser aplicado à um motor CC. O CLP Escravo recebia os comandos do Mestre e retornava/aplicava as tensões esperadas.

No teste seguinte, montou-se um experimento para a comunicação do Gateway em uma rede real de CLP's, sendo o LC700, Escravo Modbus RTU, e, separadamente, os CLP's S71200 da Siemens e DF63 da Smar como Mestres Modbus TCP. Neste teste, o CLP configurado como Mestre deveria ser capaz de requisitar os dados de tensão em um potenciômetro e, ao mesmo tempo, escrever um valor de tensão a ser aplicado à um motor CC. O CLP Escravo recebia os comandos do Mestre e retornava/aplicava as tensões esperadas.

4. RESULTADOS E DISCUSSÃO

Os primeiros Firmwares voltado ao protocolo Modbus foram programas que atuam na rede como Escravo, seja TCP ou RTU (tópicos A e B da metodologia). No caso da comunicação Modbus TCP, os testes com o simulador ModScan apresentaram resultados positivos quanto a comunicação e estabelecimento da rede composta por simulador e BBB. Na comunicação Modbus RTU a conexão também conseguiu ser estabelecida com ambos os barramentos, RS232 e RS485, demonstrando o funcionamento adequado dos adaptadores utilizados. A figura 8 mostra o protótipo do Gateway Modbus RTU com a interface RS-485.

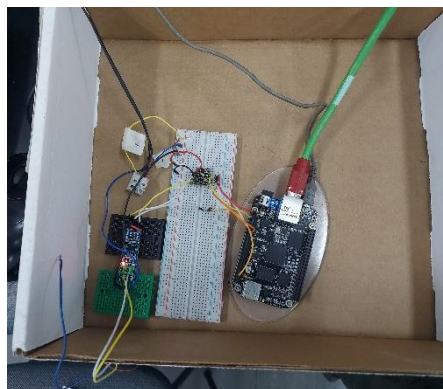
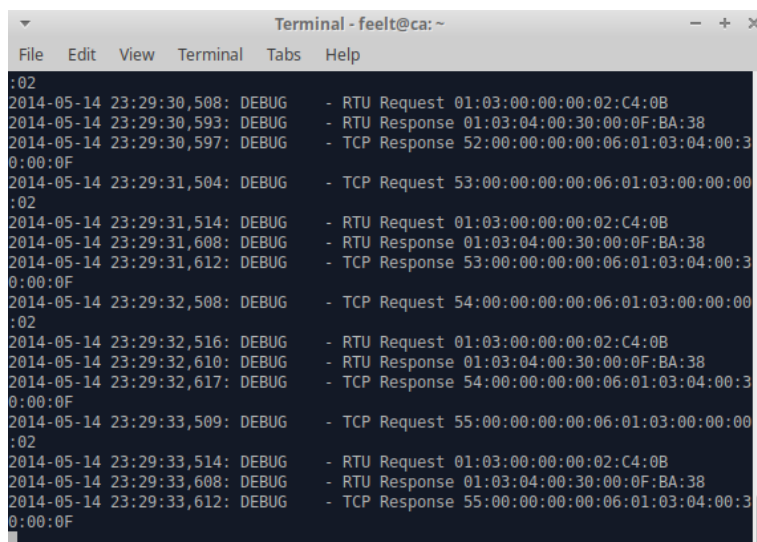


Figura 8: Circuito Gateway Modbus TCP/RTU com interface RS-485.

As telas de simulação dos testes desenvolvidos nesta etapa estão representadas nas figuras 9 a 10.



```
Terminal - feelt@ca: ~
File Edit View Terminal Tabs Help

:02
2014-05-14 23:29:30,508: DEBUG - RTU Request 01:03:00:00:00:02:C4:0B
2014-05-14 23:29:30,593: DEBUG - RTU Response 01:03:04:00:30:00:0F:BA:38
2014-05-14 23:29:30,597: DEBUG - TCP Response 52:00:00:00:00:06:01:03:04:00:3
0:00:0F
2014-05-14 23:29:31,504: DEBUG - TCP Request 53:00:00:00:00:06:01:03:00:00:00
:02
2014-05-14 23:29:31,514: DEBUG - RTU Request 01:03:00:00:00:02:C4:0B
2014-05-14 23:29:31,608: DEBUG - RTU Response 01:03:04:00:30:00:0F:BA:38
2014-05-14 23:29:31,612: DEBUG - TCP Response 53:00:00:00:00:06:01:03:04:00:3
0:00:0F
2014-05-14 23:29:32,508: DEBUG - TCP Request 54:00:00:00:00:06:01:03:00:00:00
:02
2014-05-14 23:29:32,516: DEBUG - RTU Request 01:03:00:00:00:02:C4:0B
2014-05-14 23:29:32,610: DEBUG - RTU Response 01:03:04:00:30:00:0F:BA:38
2014-05-14 23:29:32,617: DEBUG - TCP Response 54:00:00:00:00:06:01:03:04:00:3
0:00:0F
2014-05-14 23:29:33,509: DEBUG - TCP Request 55:00:00:00:00:06:01:03:00:00:00
:02
2014-05-14 23:29:33,514: DEBUG - RTU Request 01:03:00:00:00:02:C4:0B
2014-05-14 23:29:33,608: DEBUG - RTU Response 01:03:04:00:30:00:0F:BA:38
2014-05-14 23:29:33,612: DEBUG - TCP Response 55:00:00:00:00:06:01:03:04:00:3
0:00:0F
```

Figura 9: Tela de DEBUG do Firmware Gateway Modbus TCP/RTU.

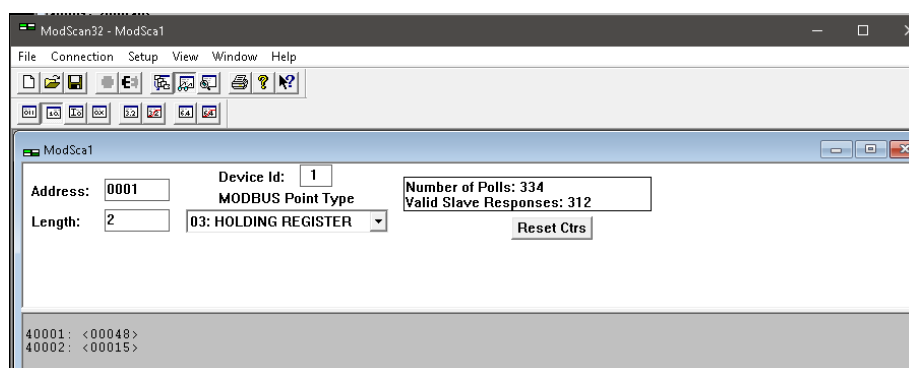


Figura 10: Tela do programa simulador Mestre Modbus, ModScan, em resposta às requisições feitas pelo Gateway.

Ao longo desta pesquisa, tentou-se desenvolver um Firmware Mestre Modbus RTU usando a biblioteca FreeModbus. As tentativas, no entanto, não tiveram êxito. Ao fazer um estudo de outras bibliotecas de código aberto (em linguagem C), verificou-se que elas, em sua maioria, não tinham suporte para operar como mestre RTU, apenas como Escravo. Uma possível explicação para isso está no fato do Linux embarcado na BBB não permitir uma forma simples de tratamento da comunicação serial a nível de interrupção para garantir os 3,5 caracteres de final de frame exigidos pelo Modbus RTU. A princípio, segundo os estudos realizados, a forma mais correta de se resolver este problema na BBB seria ou através de um Firmware sem sistema operacional Linux, ou utilizando o Linux em tempo real. Foram feitas

tentativas de adaptação do código, mas, devido à complexidade, nenhuma das tentativas teve êxito.

Diante disso, surge então a alternativa de desenvolver o Firmware Gateway Modbus na linguagem Python. Encontrou-se na plataforma GitHub, uma biblioteca em Python que se mostrou mais simples, mais concisa e até mais funcional que a bibliotecas antes utilizada. Dessa forma, começou-se a desenvolver o firmware Gateway Modbus baseado nessa biblioteca, que por sua vez já integrava a função Mestre Modbus RTU e Escravo Modbus TCP. Testes em uma rede de Mestre e Escravo simulados foram feitos com essa nova biblioteca e os resultados foram positivos.

Com a validação do funcionamento do sistema com Mestres e Escravos simulados, foram realizados os testes propostos no tópico C da metodologia. O Gateway foi colocado em um ambiente misto, sendo controlado por um Mestre TCP simulado, ModScan, e enviando comandos à um equipamento Escravo Modbus RTU, no caso o LC700 (figura 5.a). Esperava-se que a conexão entre os equipamentos fosse bem-sucedida, assim como nos simuladores, porém adaptações no Firmware tiveram de ser feitas. Observou-se que, por mais que os frames de mensagem Modbus de requisição dos valores (enviado pelo Mestre TCP ao Gateway) e de requisição traduzida à RTU (enviada pelo Gateway ao Escravo RTU) estivessem completos e chegassem devidamente ao Escravo, o LC700 era incapaz de responder corretamente à requisição feita, pois um byte ‘00’ era adicionado no começo de cada frame de resposta, causando assim, um atraso de um byte no frame de resposta RTU, conforme mostrado na figura 11.

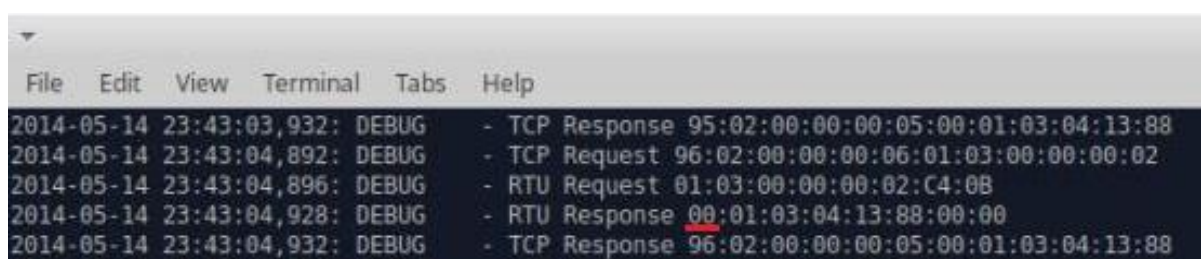


Figura 11: Tela de DEBUG do Firmware Gateway Modbus TCP/RTU com o byte ‘00’ de atraso sinalizado.

Este atraso era suficiente para gerar uma resposta errônea, um lixo de memória. Nesse contexto, tiveram de ser feitas manipulações no Firmware da BBB que corrigissem este problema na formação do frame de resposta. Com a correção deste atraso, o teste foi concluído com sucesso, a conexão entre os dois equipamentos foi bem-sucedida e os valores de rotação

do motor e potenciômetro foram devidamente implementados pelo sistema. Lembrando que novos testes nos simuladores ModScan e ModSim foram feitos e não apresentaram problemas com o novo código fonte.

Por fim, os últimos testes do sistema envolveram uma rede de comunicação de equipamentos reais. Primeiro, foram feitos testes com o CLP S71200, no modelo S71214C AC/DC/RLY (figura 7) como Mestre Modbus TCP e o LC700 como Escravo Modbus RTU. O S71200 é o equipamento com maior diversidade do laboratório, uma vez que pode comunicar protocolos Profinet e Modbus. Neste teste, o Gateway não teve êxito na comunicação. Observou-se que o S71200 apresentou uma peculiaridade na formação da requisição Modbus TCP. O primeiro byte da mensagem, que deveria ser o valor do endereço do Escravo a quem é feita a requisição, apresentava um valor fixo desconhecido de '0xFF' em hexadecimal, ou seja, o endereço Modbus do Escravo era desprezado. Dessa forma, a mensagem apresentou problemas já na requisição, o que fez o Escravo ser incapaz de responder. Este problema é mostrado nas figuras 12 e 13:

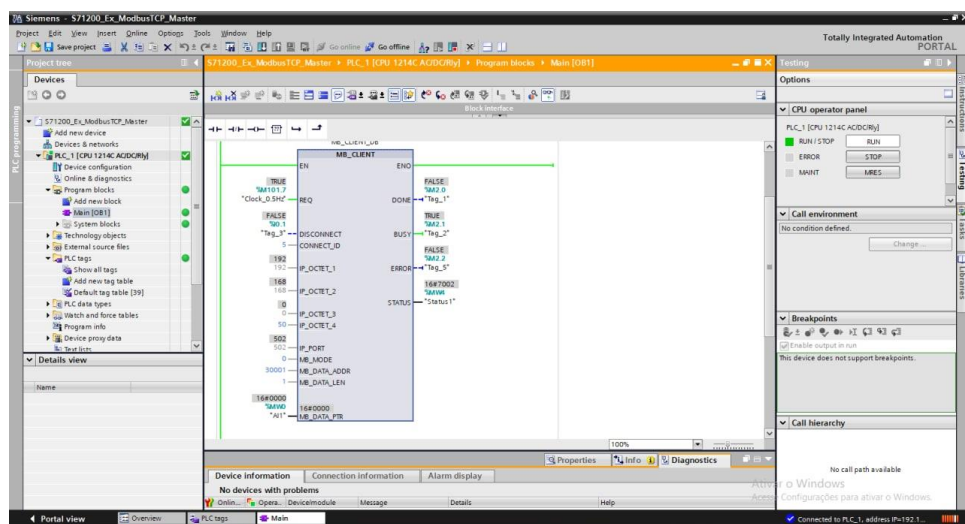


Figura 12: Tela do software da SIEMENS que programa o firmware do CLP S71200.

De acordo com a figura 12, o endereço configurado do Escravo a quem o Mestre está solicitando os dados tem o valor 5. Ou seja, o bloco Modbus do S71200 tem a informação do endereço do Escravo. No entanto, quando a mensagem é montada, o byte de endereçamento não é compatível, aparecendo '0xFF' ao invés de '0x05'. Este problema pôde ser comprovado na captura do software WireShark, mostrado na figura 13.

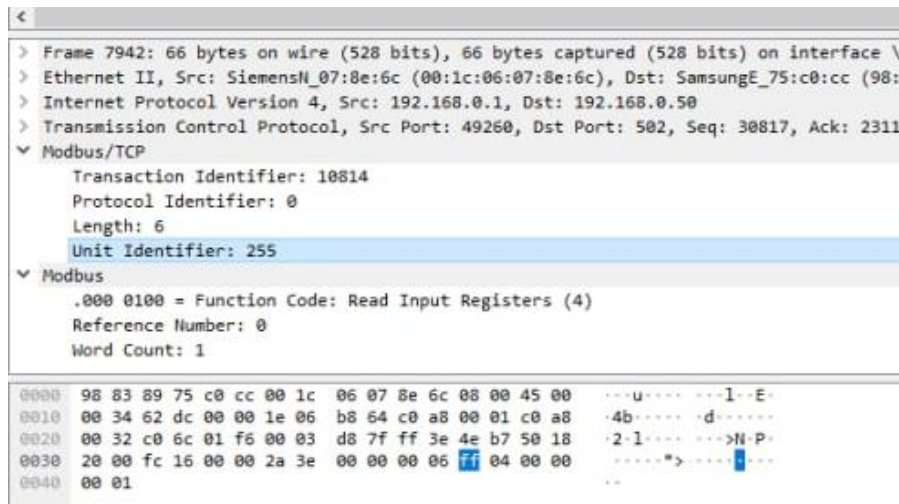


Figura 13: Tela do Software WireShark de verificação da comunicação entre a rede TCP dos dois CLP's.

O problema poderia ser facilmente contornado com uma manipulação no código, mas seria necessária uma solução específica para cada caso de endereço, o que é inviável por não ser uma solução genérica. Então, foram feitos testes com outro CLP do laboratório, o DF63 da Smar (figura 6) como Mestre TCP. Este CLP possui cartões de entrada e saída e pode comunicar nos protocolos FF, HSE e também Modbus TCP. A figura 6 mostra o DF63 utilizado.

Neste caso, obteve-se uma conexão bem-sucedida entre os dois controladores, como mostra a captura do WireShark na figura 14. Os valores de tensão no potenciômetro requisitados pelo Mestre do sistema, foram lidos e retornados pelo Escravo sem nenhum problema, da mesma forma que o valor de tensão inserido no motor pelo Mestre, foi aplicado ao motor pelo Escravo.

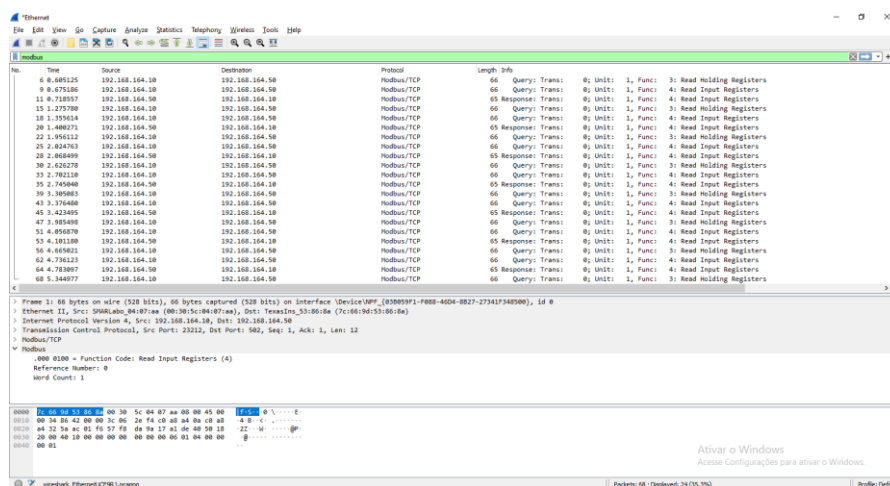


Figura 14 - Tela do Software WireShark de verificação da comunicação entre a rede TCP dos dois CLP's.

Para ilustrar o desempenho do sistema, foi feito um sistema supervisório simples no programa ProcessView da Smar, que demonstra, em curvas, a variação nos valores de cada parâmetro lido. O sistema supervisório está representado na figura 15.

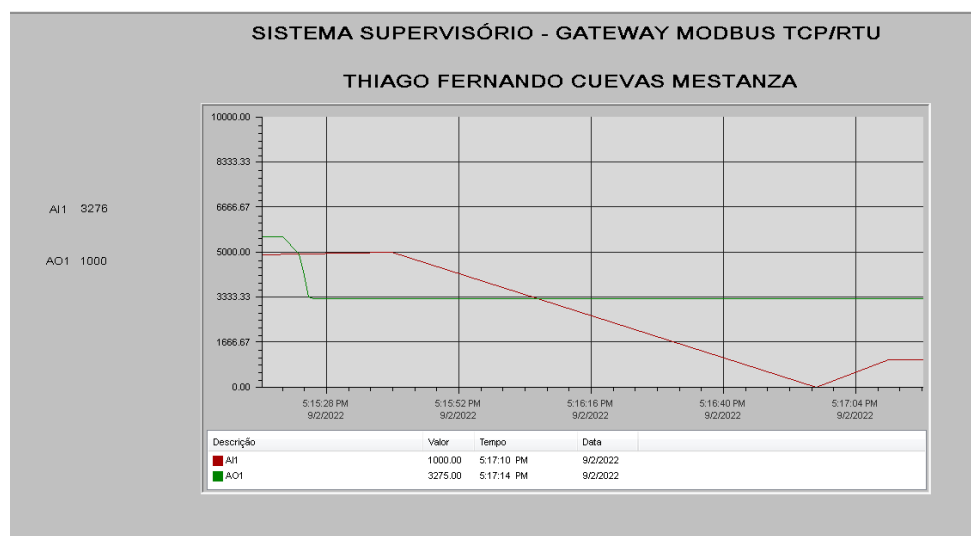


Figura 15: Tela do Sistema Supervisório desenvolvido, que acompanha de forma gráfica a variação nos valores de AI1 (Potenciômetro) em verde, e AO1 (Tensão de entrada no motor CC) em vermelho.

5. CONCLUSÕES

Este trabalho atingiu seu objetivo final, que era o desenvolvimento e implementação de um dispositivo capaz de integrar redes de protocolos distintos, estabelecendo assim a comunicação entre equipamentos que possuíam, inicialmente, diferentes contextos de atuação. A tecnologia utilizada neste projeto vem como uma forte alternativa para redução de custos e prolongamento da vida útil de dispositivos, seja no contexto industrial como no acadêmico. O sucesso no resultado do trabalho traz ao ambiente científico da comunidade uma expansão da noção de redes industriais, aumentando as possíveis ferramentas didáticas de ensino e demonstrando o potencial formativo, técnico e inovador da Faculdade de Engenharia Elétrica (FEELT). O desenvolvimento desta pesquisa deu origem a um artigo científico submetido a Conferência de Estudantes de Engenharia Elétrica (CEEL) da Universidade Federal de Uberlândia.

A bibliografia mostra que os Gateways são capazes de integrar não só redes industriais diferentes, mas também estabelecer uma conexão entre controladores e *data centers*, podendo cumprir as duas tarefas ao mesmo tempo. Nesse sentido, pode ser relevante para complementação deste trabalho, a integração de um sistema de nuvem que armazene os dados

trafegados no Gateway, de tal forma que ele seja capaz de conectar não só uma rede de equipamentos, mas também uma rede de dados.

Buscam-se agora, maneiras de aprimorar a estrutura física do Gateway, isto é, melhorar o aspecto construtivo da ferramenta, com o objetivo de torná-la mais prática, resistente e completamente compatível às exigências de funcionamento num ambiente industrial real.

6. REFERÊNCIAS

- [1] Industrial Ethernet powers ahead, while fieldbuses slide - Drives and Controls Magazine. Disponível em: <<https://drivesncontrols.com/news/fullstory.php/aid/6396/IndustrialEthernetpowersaheadwhilefieldbusesslide>>. Acesso em: 19 set. 2022.
- [2] I. Yaqoob et al., "Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges", *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10-16, 2017.
- [3] M. Tabaa, B. Chouri, S. Saadaoui and K. Alami, "Industrial Communication based on Modbus and Node-RED", *Procedia Computer Science*, vol. 130, pp. 583-588, 2018.
- [4] C. Seixas, M. Szuster, "*Programação concorrente em ambiente Windows - Uma visão de automação*" Editora da UFMG, Belo Horizonte, 2003.
- [5] F. D. Fagundes, *Segurança em Redes Industriais: Aplicação da técnica de autenticação HB-MP* em rede Modbus*, tese de doutorado, UFU, 2022.
- [6] A. Grygoruk, J. Legierski, "IoT gateway – Implementation proposal based on Arduino Board", *Federater Conference Computer Science and Information Systems (FedCSIS)*, 2016.
- [7] Sravani, B. *The Critical Role of Gateways in IoT*. Mouser Electronics. 2019. Nov. Disponível em: <<https://br.mouser.com/blog/blog/blog/critical-role-of-gateways-iot>>. Acesso em: 19 set. 2022.

*Link Video-Pitch: https://youtu.be/uTRXPBR_x7U