Conversor embarcado de protocolos Modbus/MQTT para rede IoT utilizando Raspberry Pi

Embedded Modbus/MQTT protocol converter for IoT networks using Raspberry Pi

DOI:10.34117/bjdv9n6-042

Recebimento dos originais: 02/05/2023
Aceitação para publicação: 06/06/2023

Angilberto Muniz Sobrinho
Doutor em Engenharia Elétrica pela Universidade Federal de Campina Grande (UFCG)
Instituição: Universidade do Estado do Amazonas (UEA)
Endereço: Av Darcy Vargas, 1200, Manaus - AM
E-mail: asobrinho@uea.edu.br

Dyonathan Emanuel de Araujo Morais
Bacharel em Engenharia Elétrica pela Universidade do Estado do Amazonas - UEA
Instituição: Universidade do Estado do Amazonas (UEA)
Endereço: Av Darcy Vargas, 1200, Manaus - AM
E-mail: dyonathan.deam@gmail.com

Matheus Costa Soares de Azevedo
Graduando em Engenharia Elétrica pela Universidade do Estado do Amazonas (UEA)
Instituição: Universidade do Estado do Amazonas (UEA)
Endereço: Av Darcy Vargas, 1200, Manaus - AM
E-mail: matheus.costa.s.azevedo@gmail.com

Miguel Angel Orellana Postigo
Doutor em Engenharia de Controle e Automação pela Escola Politécnica da Universidade de São Paulo (USP)
Instituição: Universidade do Estado do Amazonas (UEA)
Endereço: Av Darcy Vargas, 1200, Manaus - AM
E-mail: mpostigo@uea.edu.br

Walter Prado de Souza Guimarães
Doutor em Engenharia Elétrica pela Universidade Federal de Pernambuco (UFPE)
Instituição: Universidade do Estado do Amazonas (UEA)
Endereço: Av Darcy Vargas, 1200, Manaus - AM
E-mail: wguimaraes@uea.edu.br

RESUMO
Neste artigo apresenta-se o uso de um microcontrolador Raspberry Pi como um conversor de protocolo Modbus para MQTT. Essa conversão permite que o Raspberry Pi se comunique com um Controlador Lógico Programável (CLP) utilizando o protocolo Modbus enquanto envia dados para um servidor MQTT. O objetivo desse sistema é permitir que uma rede industrial legada cujos dados estejam disponibilizados apenas via protocolo Modbus possa se integrar a uma rede mais moderna usando um protocolo aderente ao modelo recomendado pela Indústria 4.0 (MQTT). Este artigo discute, então,

o desenvolvimento e validação de um protótipo IoT-MQTT para interfaceamento entre um Simulador Físico controlado por um CLP legado baseado em protocolo Modbus e uma infraestrutura de comunicação aberta e aderente às recomendações da Indústra 4.0, e que pode ser utilizado em uma ampla gama de aplicações industriais.

Palavras-chave: MQTT, Iot, Indústria 4.0, Modbus, Conversor de Protocolos.

ABSTRACT
In this paper, a Raspberry Pi microcontroller based Modbus to MQTT protocol converter is presented. With this protocol conversion process, the microcontroller is able to receive data from a Modbus based CLP while relaying this data to MQTT based cloud server. The goal of this system is for a legacy industrial network whose data is only available via Modbus to be integrated into a modern network with a protocol adherent to the Industry 4.0 model. The development and validation of a prototype for IoT Modbus/MQTT gateway is presented. This gateway is used to interface a Modbus based legacy CLP controlled physical plant emulator to an open communication infrastructure adherent to the Industry 4.0 recommendation and can be used in a wide range of industrial application.

Keywords: MQTT, IoT, Industry 4.0, Modbus, Procol Converter.

1 INTRODUÇÃO

A Indústria 4.0 tem sido um tema cada vez mais presente na atualidade, buscando a integração dos sistemas e processos industriais de forma inteligente e eficiente. Nesse contexto, o protocolo MQTT (MQTT, 2023) tem se destacado como uma importante alternativa para a comunicação entre os dispositivos conectados em uma rede devido à sua simplicidade, baixa complexidade e alta interoperabilidade.

Ao mesmo tempo, os sistemas legados representam uma parte considerada de investimentos efetuados em infraestrutura de automação industrial que precisam ser adaptados para os modelos mais modernos aderentes às recomendações da Indústria 4.0. Uma boa parte desses sistemas tem por base protocolos de comunicação ainda que eficientes, incompatíveis com o novo paradigma, como, por exemplo, o protocolo Modbus (MODBUS, 2023).

Diversos são os esforços na direção de aproveitar a infraestrutura legada para sistemas mais eficientes e modernos e uma das opções é a utilização de conversores de protocolos, em geral fazendo a conversão de algum protocolo proprietário ou legado para OPC-UA (OPC-UA, 2023) ou MQTT.

Como exemplo dessas iniciativas pode-se citar (VIMOS et al, 2018), em que os autores discorrem sobre tendências e possibilidades de utilização de plataformas como Arduino e Raspberry Pi como elementos compatibilizados entre redes legadas e redes 4.0.

Trabalho similar é apresentado em (ANDRADA et al., 2020) com uma abordagem mais didática voltada para a formação e capacidade de pessoal.

Em (FERREIRA et al., 2019), os autores apresentam um conversor de protocolos sem fio embarcado no mesmo controlador que implementa as funções de CLP.

Nesse contexto, apresenta-se, neste trabalho, o desenvolvimento de um conversor de protocolo de comunicação industrial pela integração de hardware e software de código aberto (open source), para aplicações IIoT, especificamente conversão entre os protocolos Modbus e MQTT.

Este artigo está organizado da seguinte forma: A seção 2 apresenta brevemente o protocolo Modbus. O protocolo MQTT é discutido de forma resumida na seção 3, e o conversor de protocolos é apresentado na seção 4. Na seção 5 são apresentadas as conclusões preliminares.

2 PROTOCOLO MODBUS

É um protocolo de baixa complexidade com uma arquitetura mestre-escravo, baseado no paradigma Pergunta/Resposta, onde um mestre inicia as transações de comunicação, interrogando, sequencialmente, os elementos escravos que respondem de forma sincronizada. Ele permite a conexão de dispositivos eletrônicos, como sensores, atuadores, controladores e outros equipamentos em rede, possibilitando a troca de informações entre eles. Essa característica é fundamental para o controle e monitoramento de processos industriais, em que a comunicação entre dispositivos é essencial para o funcionamento adequado do sistema. Por ser um protocolo aberto e amplamente documentado, é facilmente implementável e está presente em uma ampla gama de dispositivos.

Esse protocolo pode ser usado tanto entre o controlador (CLP) e os elementos da planta industrial (sensores e atuadores) como também entre CLPs podendo ser adaptado para diversos meios físicos de comunicação (SANTOS & SOUZA, 2019).

2.1 FORMATO DO PACOTE

O protocolo Modbus é baseado em um formato de pacote simples que consiste em um cabeçalho seguido por dados. O cabeçalho contém informações sobre o endereço do

dispositivo de destino, o tipo de mensagem e o número de palavras (2 bytes) de dados que estão sendo transmitidos. O formato geral de um pacote Modbus é mostrado na figura 1:

Figura 1 – Estrutura do quadro/pacote Modbus

Endereço	Função	Dados	CRC

Fonte: Os autores

Endereço: O endereço é usado para identificar o dispositivo de destino. O endereço é um número entre 1 e 247 e é configurado no dispositivo durante a instalação.

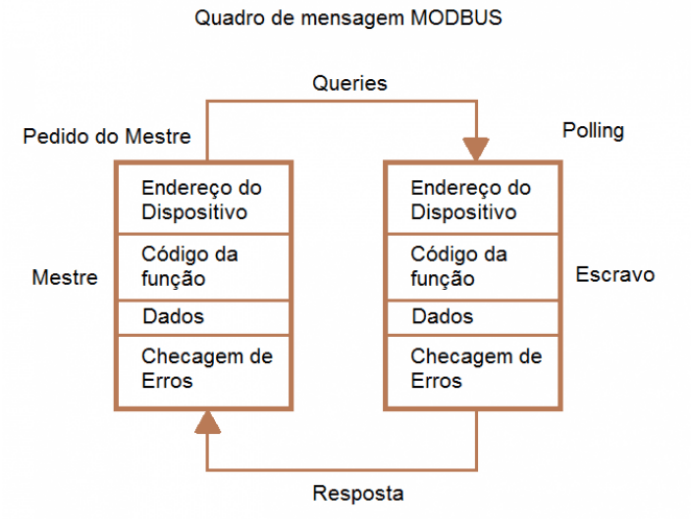
Função: A função é usada para especificar o tipo de mensagem que está sendo enviada. Existem várias funções diferentes definidas no protocolo Modbus, incluindo leitura de dados, escrita de dados e diagnóstico.

Dados: Os dados contêm as informações que estão sendo transmitidas. O número de palavras (2 bytes) de dados é especificado no cabeçalho do pacote.

CRC: O CRC (*cyclic redundancy check*) é um valor de verificação de erro que é calculado a partir dos bytes no pacote. Ele é usado para detectar erros de transmissão.

Esses quadros são transmitidos e recebidos ciclicamente conforme ilustrado na figura 2.

Figura 2 – Ciclos de Requisição/Resposta em um sistema Modbus



Fonte: (FREITAS, 2014)

2.2 MENSAGENS MODBUS

Dentre os vários tipos de mensagens definidas no protocolo Modbus, as mensagens usadas neste projeto foram:

Leitura de Dados (*Read Holding Registers*): Esta mensagem é usada para ler dados de um dispositivo. O dispositivo de destino retorna os dados solicitados no campo de dados do pacote.

Escrita de Dados (*Write Single Register*): Esta mensagem é usada para escrever um único valor em um registrador em um dispositivo de destino. O valor a ser escrito é especificado no campo de dados do pacote.

Leitura de Coil (*Read Coil*) **:** é um dos códigos de função disponíveis no protocolo Modbus e é usada para ler o estado dos coils (discretos) em um dispositivo escravo. Quando um dispositivo mestre envia uma solicitação "Read Coils" para um dispositivo escravo, o escravo responde com uma mensagem contendo um conjunto de valores booleanos que representam o estado dos coils solicitados

Escrita de Coil (*Write Single Coil*) : a função é usada para escrever um único coil (discreto) em um dispositivo escravo. Quando um dispositivo mestre envia uma solicitação "Write Single Coil" para um dispositivo escravo, ele especifica o endereço do coil a ser alterado e o valor booleano que deve ser escrito nele.

3 PROTOCOLO MQTT

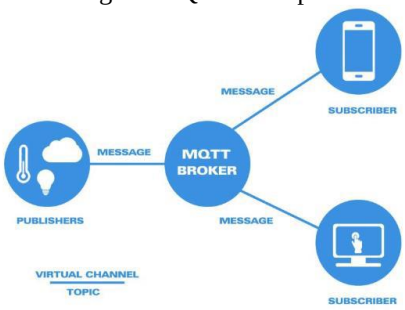
O protocolo MQTT *(Message Queuing Telemetry Transport*) é um protocolo de mensagens leves e assíncronas, que utiliza um modelo de publicação e subscrição de mensagens. Ele é amplamente utilizado em aplicações de Internet das Coisas (IoT) e Indústria 4.0, devido à sua eficiência, flexibilidade e escalabilidade.

3.1 MODELO DE COMUNICAÇÃO

O modelo de comunicação do MQTT se baseia na existência de três componentes básicos: o assinante (cliente ou consumidor da informação), o publicador (produtor da informação) e o broker (servidor ou intermediador) (AL-FUQAHA et al., 2015). Cada dispositivo interessado em um determinado tópico (informação), deve assiná-lo (informar ao broker sobre seu interesse nessa informação específica) para que o broker possa informar quando publicadores publicarem mensagens neste tópico. O broker também verifica a autorização dos publicadores e assinantes.

Este protocolo permite transmitir mensagens de um cliente (publicador) para um ou vários clientes (assinantes) através do broker que se encarrega de replicar essas mensagens para os assinantes (PERALTA et al., 2016). A figura 3 representa o modelo de comunicação utilizado pelo MQTT.

Figura 3 – Fluxo de mensagens MQTT entre produtor/broker/consumidor

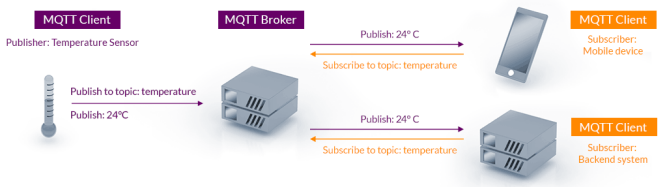


Fonte: (SILVA, 2019)

Uma das principais características do MQTT é a sua capacidade de suportar comunicação de baixa largura de banda e alta latência, tornando-o ideal para ambientes em que a comunicação é instável ou sujeita a interrupções. Além disso, o MQTT é projetado para ser simples e fácil de implementar, permitindo que os dispositivos se comuniquem de forma rápida e eficiente, sem exigir muitos recursos de processamento ou memória.

No contexto da Indústria 4.0, o MQTT é especialmente importante, pois permite a comunicação e o compartilhamento de dados em tempo real entre diferentes dispositivos e sistemas de automação industrial. E é frequentemente utilizado como protocolo de transporte para a transferência de dados entre dispositivos IoT e sistemas de controle de processos, permitindo que informações sejam trocadas de forma ágil e eficiente, aumentando a eficiência e a produtividade dos sistemas. Um exemplo de aplicação IoT com MQTT pode ser visto na figura 4.

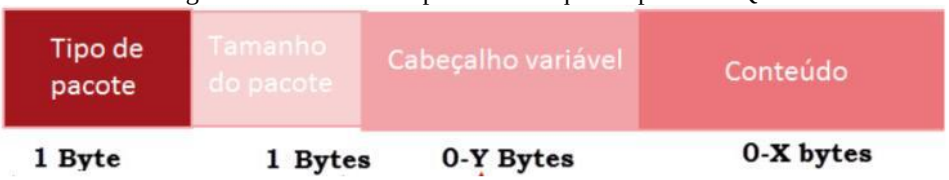
Figura 4 – Exemplo de Monitor de temperatura usando MQTT



Fonte: (MQTT, 2023)

O pacote MQTT é composto por um cabeçalho e um *payload (conteúdo)*. O cabeçalho inclui informações sobre o tipo de mensagem, o ID do pacote, o tópico e o QoS (Quality of Service) desejado. O QoS é um mecanismo para garantir que as mensagens sejam entregues com confiabilidade em ambientes de rede com perda de pacotes. A estrutura simplificada do quadro/pacote pode ser vista na figura 5.

Figura 5 – Estrutura simplificada do quadro/pacote MQTT

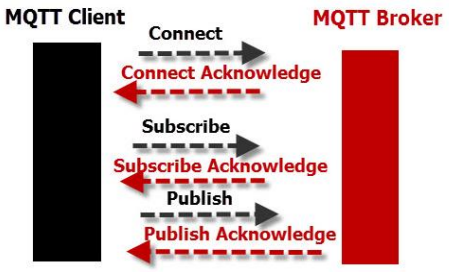


Fonte: Adaptado de (STEVES, 2023)

A Carga Útil (*Payload*) contém a Mensagem de Aplicação que está sendo publicada. O conteúdo e formato dos dados são específicos da aplicação. O comprimento da carga útil pode ser calculado subtraindo o comprimento do cabeçalho variável do campo "Comprimento Restante" (*Remaining Length*) que está no Cabeçalho Fixo. É válido que um Pacote PUBLISH contenha uma carga útil de comprimento zero.

Existem três tipos de mensagens MQTT: publish, subscribe e unsubscribe. A mensagem *publish* é usada para enviar uma mensagem para um tópico específico. A mensagem *subscribe* é usada para se inscrever em um ou mais tópicos. E a mensagem *unsubscribe* é usada para cancelar uma inscrição em um tópico. Na figura 6 pode-se observar a sequência de algumas dessas mensagens trocadas entre o cliente MQTT e o broker.

Figura 6 – Fluxo de mensagens entre um cliente e um intermediário MQTT



Fonte: (STEVES, 2023)

4 CONVERSOR DE PROTOCOLO

O sistema que motivou o desenvolvimento do conversor de protocolo é composto de um uma planta industrial didática que representa um sistema flexível de manufatura e emula, em pequena escala, um sistema produtivo dinâmico. Esse sistema contém sensores e atuadores diversos associados a esteiras, robôs, termômetros, etc... que são controlados por um CLP convencional cuja única forma de comunicação com o meio externo, um Sistema Supervisório, por exemplo, é através do protocolo Modbus/TCP. Esse sistema funciona em paralelo com um simulador de processos industriais. Sua estrutura está representada na figura 7.

Figura 7 – Sistema original

Fonte: Os autores

4.1 DEFINIÇÃO DO PROBLEMA

O sistema foi concebido com base em um sistema flexível de manufatura (físico) e sua representação virtual em um simulador, a partir do qual pode-se criar gêmeos digitais e disponibilizar informações atualizadas em uma base de dados para acompanhamento gerencial, conforme representado na figura 8.

Figura 8 – Sistema desejado

Fonte: Os autores

Entretanto, a disponibilização dos dados, no sistema de nuvem, através do simulador, impõe alta carga computacional além de que apenas as variáveis manipuladas pelos processos modelados seriam visualizadas. Como, no projeto, é desejado que diferentes níveis de monitoração seja realizado com, até mesmo, eventualmente estados de sensores e atuadores no “chão da fábrica”, foi necessário estender o barramento Modbus para além da rede local alcançando a nuvem. Mas a infraestrutura de nuvem utilizada, não tem suporte para o protocolo Modbus e tão somente para o protocolo MQTT, tornando-se necessário a criação de um dispositivo com a função de conversão e mapeamento entre as mensagens dos dois protocolos.

4.2 SISTEMA PROPOSTO

O sistema proposto modifica o sistema original pela inclusão de um microcontrolador Raspberry Pi com linguagem Python embarcada entre o barramento Modbus via Ethernet e o barramento IoT via Wifi com protocolo MQTT, ilustrado de forma esquemática na figura 9.

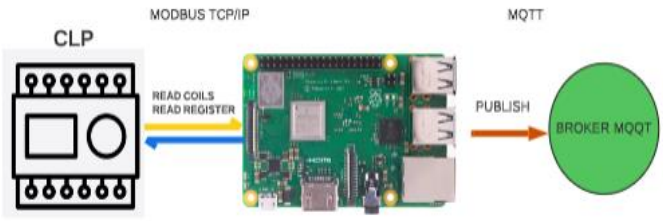
Figura 9 – Sistema proposto

Fonte: Os autores

4.3 CONVERSOR DE PROTOCOLOS MODBUS/MQTT

Um conversor de protocolos é uma combinação de hardware e software que permite a comunicação entre sistemas baseados em diferentes protocolos de comunicação. Neste projeto o conversor de protocolos trata especificamente dos protocolos Modbus e MQTT e, do ponto de vista de hardware, pode ser representado de forma simplificada como visto na figura 10.

Figura 10 – Conversor de protocolos Modbus/MQTT



Fonte: Os autores

O protocolo Modbus é amplamente utilizado em sistemas de controle industrial, enquanto o protocolo MQTT é frequentemente usado em aplicações de Internet das Coisas (IoT). O conversor de protocolos Modbus e MQTT traduz as mensagens enviadas entre os dois protocolos, permitindo que dispositivos que usam Modbus possam comunicar-se com dispositivos que usam MQTT. Isso torna a integração de diferentes

sistemas mais fácil e eficiente, permitindo uma maior interoperabilidade entre dispositivos e sistemas em ambientes de controle industrial e IoT.

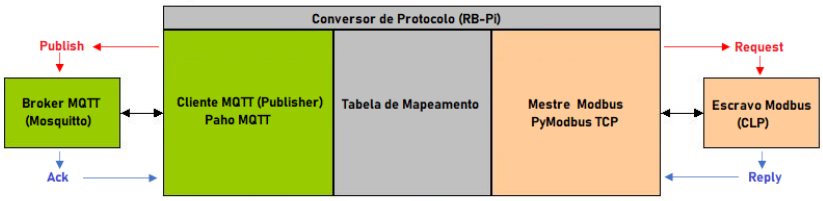
A aplicação de conversão de protocolo se caracteriza como um módulo de software escrito em Python que efetua a leitura de variáveis residentes no CLP e são acessadas via protocolo Modbus e as transfere para o sistema IoT via protocolo MQTT, permitindo a integração do sistema legado ao sistema 4.0.

Para o desenvolvimento do conversor utilizou-se a linguagem de programação Python embarcada em um microcontrolador Raspberry Pi. Para realizar a comunicação MQTT foi utilizado a biblioteca livre Paho MQTT (PAHO, 2021) e para o protocolo Modbus a biblioteca livre pyModbusTCP (PYMODBUS, 2022).

4.4 PROCESSO DE CONVERSÃO MODBUS/MQTT

O processo de conversão consiste, basicamente, de uma tabela de mapeamento entre as variáveis vindas do CLP na forma de registradores Modbus e suas estruturas de tópicos segundo a especificação MQTT, representadas na forma de *strings json*, cuja representação esquemática pode ser vista na figura 11.

Figura 11 – Conversor de protocolos Modbus/MQTT



Fonte: Os autores

5 CONCLUSÃO

O conversor de protocolos aqui apresentado, mostrou-se funcional permitindo que uma rede legada controlada por um CLP com suporte apenas ao protocolo Modbus pudesse compartilhar suas informações e participar de uma rede baseada no protocolo MQTT, mais moderno, e permitindo o compartilhamento de informações via internet.

A utilização da linguagem Python e elementos *opensource* como bibliotecas e hardware aberto, tornou possível sua construção com custo reduzido, mesmo que esse não tenha sido o objetivo principal do projeto.

REFERÊNCIAS

MQTT, MQTT: The Standard for IoT Messaging, 2023. Disponível em http://mqtt.org. Acessado em Abril de 2023.

MODBUS, Modbus Organization, 2023. Disponível em https://modbus.org/. Acessado em Abril de 2023.

OPC-UA, Unified Architecture, 2023. Disponível em https://opcfoundation.org/. Acessado em Abril de 2023.

SANTOS, C. H. A.; SOUZA, D. L. Integração dos Protocolos MMS (IEC 61850) Modbus TCP para aplicação em acionamento de equipamentos de média tensão. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) - Faculdade Doctum de João Monlevade. p. 92. 2019. Disponível em http://hdl.handle.net/123456789/2270 .

FREITAS, Carlos Márcio. Protocolo Modbus: Fundamentos e Aplicações. 2014. Disponível em https: //www.embarcados.com.br/protocolo-modbus/. Acessado em Abril de 2023.

PERALTA, Goiuri, et al. Fog Computing Based Efficient IoT Scheme for the Industry 4.0. Espanha: IK4-Ikerlan Technology Research Centre, Information and Communication Technologies Area, 2016.

AL-FUQAHA, Ala., et al. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 4, FOURTH QUARTER 2015.

SILVA, FABIO COELHO DE SOUZA, Novus, Artigo: Modelos Cliente-servidor do Modbus Tcp e Publish-subscribe do Mqtt, 2019. Disponível em: https://www.novus.com.br, Acesso em Abril de 2023.

STEVES, Understanding the MQTT Protocol Packet Structure, 2023. Disponível em http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/, Acesso em Abril de 2023.

PAHO, Eclipse Paho MQTT Python client library, 2021. Disponível em http://pypi.org/project/paho-mqtt/, Acesso em Outubro de 2022.

PYMODBUS, A simple Modbus/TCP client library for Python, 2022. Disponível em http://pypi.org/project/paho-mqtt/, Acesso em Outubro de 2022.

VIMOS, VICTOR; SACOTO-CABRERA, ERWIN; Results of the implementation of a sensor network based on Arduino devices and multiplatform applications using the standard OPC UA, IEEE LATIN AMERICA TRANSACTIONS, VOL. 16, NO. 9, SEP. 2018.

ANDRADA, A., et al, Didactic Prototype for Teaching the MQTT Protocol Based on Free Hardware Boards and Node-RED, IEEE LATIN AMERICA TRANSACTIONS, VOL. 18, NO. 2, FEB. 2020.

FERREIRA et al.: Development of a Wireless Gateway for Industrial Internet of Things Applications, IEEE LATIN AMERICA TRANSACTIONS, VOL. 17, NO. 10, OCT. 2019.