



Desenvolvimento de um Sistema de Aquisição de Dados Modular e Escalável com Comunicação sem Fio para Aplicações Industriais

Isaac Vinícius Fontes Sampaio

Dissertação apresentada à Escola Superior de Tecnologia e Gestão de Bragança para a obtenção do grau de Mestre em Engenharia Industrial, no âmbito do programa de dupla diplomação com a Universidade Tecnológica Federal- Paraná.

Trabalho orientado por:

Professor PhD José Augusto

Professor PhD Paulo Leitão

Professor PhD Joaquim Mira

Bragança

2022-2023



Desenvolvimento de um Sistema de Aquisição de Dados Modular e Escalável com Comunicação sem Fio para Aplicações Industriais

Isaac Vinícius Fontes Sampaio

Dissertação apresentada à Escola Superior de Tecnologia e Gestão de Bragança para a obtenção do grau de Mestre em Engenharia Industrial, no âmbito do programa de dupla diplomação com a Universidade Tecnológica Federal - Paraná.

Trabalho orientado por:

Professor PhD José Augusto

Professor PhD Paulo Leitão

Professor PhD Joaquim Mira

Bragança

2022-2023

Dedicação

Dedico este trabalho primeiramente à minha mãe, cuja luta incansável foi fundamental para que eu pudesse atingir meu objetivo. Seu apoio incondicional, amor e sacrifícios são uma fonte de inspiração para mim. Agradeço de coração por todo o suporte e dedicação ao longo dessa jornada.

Dedico também este trabalho à minha avó Joana, a quem tenho imensa gratidão e eterna admiração. Seu exemplo de força, sabedoria e perseverança tem sido uma inspiração constante em minha vida. Sou profundamente grato por todo o carinho e encorajamento que ela sempre me ofereceu.

De forma especial, dedico este trabalho aos meus familiares e amigos. Seu apoio, compreensão e incentivo ao longo dessa jornada acadêmica foram inestimáveis. Agradeço por estarem ao meu lado, compartilhando alegrias, desafios e vitórias.

Não posso deixar de dedicar este trabalho aos meus orientadores, José Augusto e Paulo Leitão. Sua orientação especializada, paciência e dedicação foram essenciais para o meu crescimento acadêmico e o sucesso deste trabalho. Agradeço por compartilharem seu conhecimento, por me desafiarem a ir além e por acreditarem em meu potencial.

Esta dedicação é uma pequena expressão da minha gratidão a todas essas pessoas incríveis que contribuíram de maneira significativa para minha jornada. Seu apoio e presença foram fundamentais para eu alcançar este objetivo.

Com profundo apreço,

Isaac Vinícius Fontes Sampaio

Agradecimento

Gostaria de começar expressando minha profunda gratidão à minha mãe, cujo amor, apoio e incentivo constantes foram fundamentais para a minha jornada acadêmica. Sua presença e confiança em mim me deram força e determinação para enfrentar os desafios ao longo desse processo.

Gostaria de agradecer de forma especial à Luíza por sempre acreditar em mim e me incentivar a superar meus limites. Sua confiança inabalável foi uma fonte constante de motivação e inspiração. Também quero expressar minha gratidão ao Gustavo e à Vitória pelo apoio incondicional e atenção dedicada a este processo. Suas contribuições e encorajamento foram inestimáveis para o sucesso desta pesquisa.

Agradeço ao projeto ON-SURF pela oportunidade de participar e contribuir para essa iniciativa inovadora. Sua visão e trabalho árduo na área de aquisição de dados em ambientes industriais são inspiradores.

Aos meus orientadores, José Augusto, Paulo Leitão e Joaquim Mira, sou imensamente grato pela orientação especializada, paciência e apoio ao longo desta jornada acadêmica. Suas orientações valiosas e conhecimentos compartilhados foram fundamentais para o desenvolvimento e aprimoramento deste trabalho.

Gostaria de expressar minha gratidão às instituições IPB e UTFPR por fornecerem o ambiente acadêmico propício e os recursos necessários para a realização desta pesquisa.

Por fim, gostaria de agradecer a todos os indivíduos e organizações que, de alguma forma, contribuíram para o sucesso deste trabalho, mesmo que não tenham sido mencionados individualmente. Sua contribuição e apoio foram inestimáveis.

Muito obrigado!

Abstract

The main objective of this work is the development of a modular and scalable data acquisition system with wireless communication, applicable in various industries and capable of handling different magnitudes. To achieve this objective, three specific goals were established: to develop a Lite version of the existing acquisition system, to create a wireless scalable and modular version, and to elaborate a protocol that ensures scalability.

Throughout the work, communication tests using the UDP protocol were performed, as well as scalability and modularity tests. Additionally, an innovative protocol was developed to guarantee the system's scalability without the need for complex configurations in the WiFi network. The test results were conclusive, demonstrating the efficiency and functionality of the proposed system.

This dissertation contributes to technological advancement and industrial process efficiency, driving improvements and innovations in the field of data acquisition. Based on the obtained results, it can be concluded that the proposed objectives were satisfactorily achieved, highlighting the data acquisition system as a promising solution for monitoring and controlling magnitudes in the industry.

Keywords: Scalability protocol, Wireless communication, Modular and scalable systems.

Resumo

Este trabalho visa o desenvolvimento de um sistema de aquisição de dados modular e escalável com comunicação sem fio, aplicável em diversas indústrias e capaz de lidar com diferentes grandezas. Para alcançar esse objetivo, foram estabelecidos três objetivos específicos: desenvolver uma versão Lite do sistema de aquisição existente, criar uma versão escalável e modular sem fio, e elaborar um protocolo que garantisse a escalabilidade.

Ao longo do trabalho, foram realizados testes de comunicação utilizando o protocolo UDP, assim como testes de escalabilidade e modularidade. Além disso, foi desenvolvido um protocolo que segura a escalabilidade do sistema sem a necessidade de configurações na rede WiFi. Os resultados dos testes foram conclusivos, demonstrando a eficiência e funcionalidade do sistema proposto.

Esta dissertação contribui para o avanço tecnológico e a eficiência dos processos industriais, impulsionando melhorias e inovações na área de aquisição de dados. Com base nos resultados obtidos, conclui-se que os objetivos propostos foram alcançados satisfatoriamente, destacando o sistema de aquisição de dados como uma solução promissora para o monitoramento e controle de grandezas na indústria.

Palavras-chave: Protocolo de escalabilidade, Comunicação sem fio, Sistemas Modular e Escalável.

Conteúdo

Agradecimento	vii
Abstract	ix
Resumo	xi
1 Introdução	1
1.1 Enquadramento	2
1.2 Objetivos	3
1.3 Estrutura do Documento	3
2 Estado da Arte	5
2.1 Projeto On-Surf	5
2.2 Trabalhos envolvidos	6
2.2.1 Condicionamento de sinal para sistema de aquisição de dados em superfícies metálicas finas	7
2.2.2 Sistema de aquisição de dados para monitoramento de processo de superfície fina	8
2.2.3 Desenvolvimento de um concentrador de dados sensoriais para transmissão em rede	9
2.2.4 Plataforma de análise e visualização de dados para sistemas IoT industriais baseada em métodos de Big Data	9
2.3 Sistema de aquisição de dados	9

2.4	Sistema modular	10
2.5	Sistema escalável	11
2.6	Protocolos de comunicação em rede	11
2.6.1	TCP	12
2.6.2	UDP	12
2.7	Rede de sensores sem fio	13
2.7.1	Na indústria	13
2.8	Funcionamento da Atual Arquitetura	14
3	Proposta de Solução	17
3.1	Proposta de Solução	17
4	Desenvolvimento/Implementação	21
4.1	Comunicação	22
4.2	Comunicação serial	23
4.3	Comunicação MQTT	24
4.4	Comunicação UDP	25
4.5	Sistema com comunicação sem fio	26
4.6	Protocolo Não Orientado À Conexão Para Rede De Sensores Distribuídos .	28
4.7	Análise da estrutura do protocolo	28
4.8	Interpretação do protocolo	30
5	Análise e Discussão de Resultados	31
5.1	Teste de comunicação UDP	31
5.1.1	Taxa de perda de pacote	32
5.2	Modularidade e escalabilidade	33
5.2.1	Com 4 publicadores e 1 leitor	35
5.2.2	Com 4 leitores e 1 publicador	36
5.3	Robustez	37
6	Conclusões	39

A	Protocolo Não Orientado À Conexão Para Rede De Sensores Distribu-	
	dos	44
B	Códigos desenvolvidos	52
B.1	Publicação em C realizada pelo ESP32	52
B.2	Leitura em C realizada pelo ESP32	59
B.3	Leitor em python realizado pelo computador	65

Lista de Figuras

2.1	Sistema de aquisição de dados do projeto ON-SURF	6
2.2	condicionamento de sinal da placa ON-SURF	7
2.3	Placa ON-SURF	8
2.4	Arquitetura do sistema de aquisição de dados e condicionamento de sinal - ON-SURF	14
3.1	Arquitetura	19
4.1	1ª versão <i>LITE do sistema de aquisição de dados</i>	24
4.2	2ª versão <i>LITE do sistema de aquisição de dados</i>	25
4.3	versão final <i>LITE do sistema de aquisição de dados</i>	26
4.4	Sistema de aquisição com comunicação sem fio	27
4.5	Protocolo Não Orientado À Conexão Para Rede De Sensores Distribuídos .	29
5.1	Taxa de sucesso na entrega	32
5.2	3 Publicadores e 2 Leitores	34
5.3	implementação	34
5.4	4 publicadores e 1 leitor	35
5.5	4 leitores e 1 publicador	36
5.6	Funcionamento com 5 ESP	38
5.7	Funcionamento com 2 ESP	38

Capítulo 1

Introdução

Com o crescimento da indústria 4.0 cada vez mais os dados de um sistema são cruciais, com isso os avanços em busca por aquisições, transmissões, armazenamento e análise de dados vem ser tornando mais necessário para o desenvolvimento de tarefas como monitorar, diagnosticar, prevenir e otimizar [1].

Um exemplo disso é um operador acompanhando um processo de produção em tempo real com base nos dados mostrados ele pode interpretar o funcionamento da máquina e diagnostica a qualidade daquela produção e caso haja alguma anomalia poderá intervir imediatamente, assim contribuindo para o aumento de vida útil do equipamento, com o apoio de algoritmos pode-se ainda prever e planejar futuras manutenções, reduzindo a inatividade e custos operacionais.

Os sistemas de aquisição de dados são desenvolvidos em diversas áreas, principalmente na indústria, a busca por sistemas de permitam uma escalabilidade e reconfiguração sem necessidade de grandes investimentos estão cada vez mais se tornando necessário. Esses avanços implicam diretamente na questão de qualidade e controle de equipamentos e máquinas.

Cada sistema de aquisição de dados possui em sua estrutura: sensores, amplificadores de sinal, conversores de AC para DC, processador de dados e um modulo de comunicação[2], essa definição pode mudar de algo com a característica e aplicabilidade do sistema, em algumas aplicações devem ser projetadas com alimentações por baterias [3], o que

levar a pensam em sua durabilidade e eficiência a longos dos anos, todo esse conjunto denominamos de nó de comunicação.

O modulo de comunicação de um nó em sua maioria é o principal responsável pelo consumo de energia, isso se deve ao fato da transmissão e recepção de dados [3], esse consumo ainda depende da tecnologia que será implementada, bluetooth, infravermelho, radio frequência entre outras.

Em ambientes industriais em algumas aplicações podemos desconsiderar a questão de durabilidade da bateria, pois dependo da localização da máquina no layout da indústria e do seu tipo podemos realizar uma alimentação local, assim evitando mal funcionamento após certos períodos.

Entre as soluções para a escalabilidade de um sistema de aquisição de dados está a sua forma de comunicação, onde dependente de sua organização podemos estar configurando-a para mais nós interligados e com isso podemos estar reconfigurando as disposições dos nós e suas aplicações [4].

1.1 Enquadramento

O presente trabalho é desenvolvido em contexto com o projeto ON-SURF - Plataforma tecnológica de Engenharia de Superfícies, que conta com a participação de diversas empresas nacionais e internacionais, como diversos centros de investigação nacionais. O projeto pretende desenvolver e aplicar, tecnologias e ferramentas capazes de prolongar a vida útil de ferramentas e produtos de diversos sectores como: automóvel, Aeronáutica, Ferramentas e Moldes, Saúde e Eletrônica [5].

O projeto ON-SURF em parceria com o CeDri - IPB (Centro de Investigação em Digitalização e Robótica Inteligente) visa a aquisição de dados de maquinas e ferramentas elétricas, utilizadas na industria, nesse caso em especial a industria automobilística que utiliza de prensas para estampagem de peças metálicas, como fuselagem e peças internas.

A motivação para este trabalho é desenvolver uma versão LITE do sistema de aquisição de dados existente, onde o sistema possa ser escalável e reconfigurável, utilizando comunicação sem fio.

O presente trabalho foi desenvolvido na parte final do projeto e concluído após o encerramento do mesmo, onde já possuíamos um sistema completo de aquisição de dados e uma plataforma de análise de dados.

1.2 Objetivos

Este projeto tem como objetivo desenvolvimento de um sistema de aquisição de dados que seja reconfigurável e escalável com comunicação sem fio, que possa ser aplicado na indústria como um todo, para diversos tipos de grandezas , principalmente tensão superficial, temperatura, pressão e vibração.

Os objetivos específicos são:

- Desenvolvimento de uma versão Lite do sistema de aquisição;
- Desenvolvimento de uma versão escalável e modular, sem fio;
- Criação de um protocolo que garanta a escalabilidade.

1.3 Estrutura do Documento

O presente documento está distribuído em 6 capítulos. Este capítulo trata de uma breve introdução contendo histórico, contexto do projeto e objetivos. O Capítulo 2, o estado da arte é uma coleção de fontes importantes sobre a obra, literatura e técnicas existentes para aquisição de dados escaláveis e distribuídos. O Capítulo 3 apresenta a descrição do problema a metodologia usada para solução. O Capítulo 4 explica como o trabalho foi desenvolvido, as fases da obra, acertos e erros. O Capítulo 5 apresenta a avaliação dos resultados, se o trabalho correu bem e como corrigir os erros. Por fim, no Capítulo 6, são

apresentadas as conclusões, onde são avaliados e destacados os principais objetivos. O Capítulo 6 também aponta alguns trabalhos futuros.

Capítulo 2

Estado da Arte

Neste capítulo será apresentado o estado da arte do presente documento. Atual estado de desenvolvimento do projeto, conceitos, tecnologias e revisão da literatura serão abordados afim de contextualizar o leitor com o tema proposto. Antes precisamos entender o objetivo do projeto em que o presente documento se encontra inserido, após isso nos aprofundaremos em no tema de aquisição de dados, modelo de sistemas, protocolos e rede de sensores em fio.

2.1 Projeto On-Surf

O projeto ON-SURF visa mobilizar e envolver transversalmente empresas nacionais de diferentes setores e atividades e também outras entidades, em torno do eixo prioritário na agenda de inovação internacional, a engenharias de superfícies. O projeto envolve empresas posicionadas em toda a cadeia de valor, as modificadoras de superfície, as aplicadoras das soluções e os utilizadores finais [5] . Pretende-se desenvolver e aplicar processos de modificação de superfície que promovam soluções avançadas e inovadoras em diferentes setores de atividades, tais como automóvel, aeronáutica, moldes e ferramentas, saúde e eletrônica; alinhados com os objetivos estratégicos dos clusters nacionais: health cluster Portugal, energyIN, engineering e tooling, mobinov, produtech, tice.pt [5].

O projeto On-Surf em parceria com o CeDri pretende desenvolver propostas de soluções para aquisição de dados e monitoramento de superfícies, sendo que os sensores produzidos pelo projeto On-Surf serão sensores finos, sensores este semelhantes a extensômetro, também conhecidos como extensômetro. Estes sensores são capazes de medir a tensão, força superficial de máquinas, em especial as prensas. O CeDri é responsável por desenvolver um sistema de aquisição de dados modular, escalável e robusto. Tal sistema deve ser capaz de adquirir dados não somente de sensores de tensão, mas também de outras variáveis de importância para indústria como temperatura.

2.2 Trabalhos envolvidos

O sistema de aquisição de dados atual, ilustrado na figura 2.1, é resultado de um processo de desenvolvimento anterior que foi dividido em quatro trabalhos distintos. Esses trabalhos foram realizados com o objetivo de fornecer uma base sólida e uma compreensão aprofundada do estado da arte atual. A seguir, apresentamos uma breve visão geral dessas quatro teses de mestrado, que contribuiram para a evolução e explicação detalhada do sistema em questão.

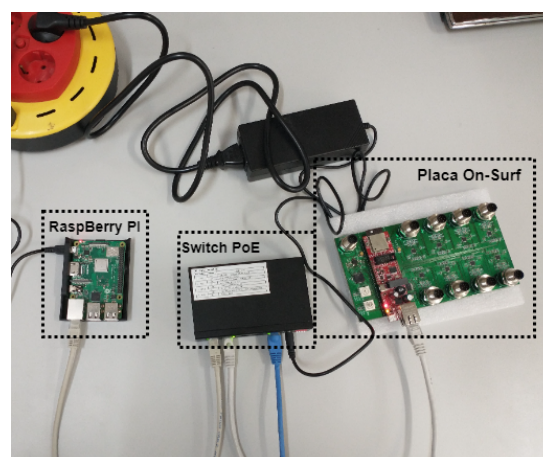


Figura 2.1: Sistema de aquisição de dados do projeto ON-SURF

Fonte: autoria própria

2.2.1 Condicionamento de sinal para sistema de aquisição de dados em superfícies metálicas finas

Trabalho este desenvolvido no [6], onde desenvolve o condicionamento de sinal do atual projeto, o condicionamento do sinal é de suma importância pois é a primeira etapa de um sistema de aquisição de dados, ou seja, está em contato diretamente com as variáveis e fenômenos físicos, isso implica que este trabalho seja o mais importante. Possuindo etapas cruciais entre elas, aquisição do sinal analógico, onde é filtrado para evitar os ruídos, passa pela amplificação do sinal, e depois a conversão do sinal de analógico para digital. Para compreender todos os padrões exigidos pelo projeto e requisitos mínimos para que possa ser implementado dentro da indústria, foi utilizado o circuito integrado ZSSC 4151, que em suas especificações possui internamente amplificação com fator de correção não linear e ajuste conforme a temperatura e conversor de analógico/digital internamente[6].



Figura 2.2: condicionamento de sinal da placa ON-SURF

Fonte: autoria própria

2.2.2 Sistema de aquisição de dados para monitoramento de processo de superfície fina

Este trabalho desenvolvido no [7], onde desenvolve a parte de transmissão do sistema, isso envolve o módulo de comunicação e os sistemas embebidos que estão envolvidos.

Para garantir todos os requisitos industriais e objetivos do projeto, tornando o sistema modular e escalável, utilizando o trabalho desenvolvido anteriormente [6], neste trabalho foi inserido um I/O Expander PCF8574 que permite a expansão sistema de aquisição anterior, fazendo com que ele possa coletar dados de 8 canais simultaneamente, além disso este dispositivo funciona como um switch fornecendo 7 bits por meio da interface I2C, este numero é convertido e adionado ao IP do ESP32. É utilizado o ESP32-PoE-ISO como sistema embebido responsável pelo processamento e transmissão do dados, O ESP32 é um excelente microcontrolador para aplicação em internet das coisas com um processador de 240MHz, 34 GPIOs programáveis, SAR-ADCs de 12 bits com 18 canais e suporta I2C, I2S, SPI e outros periféricos [7]. Foi utilizada a porta de comunicação Ethernet do ESP32-PoE-ISO para comunicação por cabo para um concentrador, enviando os dados via UDP.

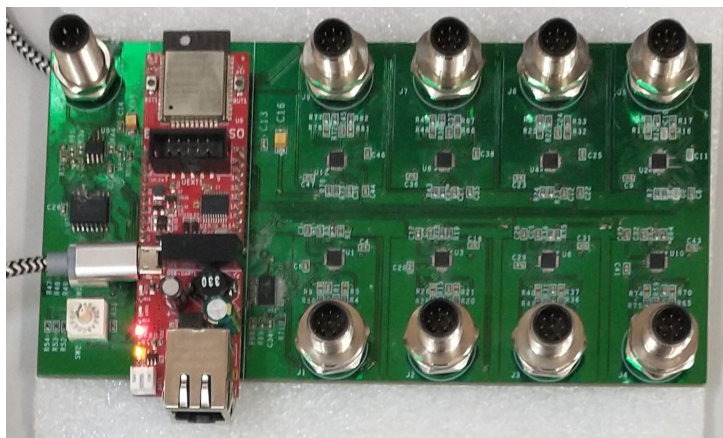


Figura 2.3: Placa ON-SURF

Fonte: autoria própria

2.2.3 Desenvolvimento de um concentrador de dados sensoriais para transmissão em rede

Este trabalho desenvolvido no [8], onde desenvolve a parte de transmissão do sistema, isso envolvendo o concentrador de dados e qual sistema embebidos que estão envolvidos.

Mantendo os requisitos anteriores de escalabilidade e modulabilidade, foi utilizado o Raspberry Pi 3B+ como concentrador de dados, o raspberry Pi 3B+ possui uma melhor funcionalidade em relação às versões anteriores, como Ethernet mais rápida e suporte Power-over-Ethernet. Ele também possui um processador quad-core de 1,4 GHz e 64 bits, a memória é o cartão SD externo de 16GB. Dentro do soquete de comunicação ethernet do raspberry Pi foi configurado para comunicação multicast, o mesmo raspberry era responsável por enviar os dados por um base de dados na nuvem (MongoDB)[8].

2.2.4 Plataforma de análise e visualização de dados para sistemas IoT industriais baseada em métodos de Big Data

Este trabalho desenvolvido no [9], onde desenvolve a parte de visualização do sistema, isso envolvendo o concentrador de dados, a base de dados a visualização.

Com o objetivo de possuir duas formas de visualização uma operacional e outra remota, foi adicionado um raspberry a ao sistema, assim um para visualização local e outro para visualização remota, alimentando a base de dados, dentro da base de dados foi desenvolvidos os níveis de acesso e um dashboard para melhor interpretação do sistema[9].

2.3 Sistema de aquisição de dados

Os sistemas de aquisição de dados, ou como comumente abreviado para DAS ou DAQ forma mais utilizada, são sistemas projetados para adquirir informações sobre determinada variação de um sistema físico[10]. Os sistemas de aquisição de dados são amplamente utilizados em diversos campos, incluindo de produção industrial a experimentos científicos, sendo que cada aplicação requer um tipo específico de sistema [10].

Em termos gerais, os DAS podem ser divididos em três componentes principais: sensores, responsáveis pela coleta de dados dos sistemas físicos; circuitos, utilizados para transmitir esses dados a um computador; e sistemas computacionais, responsáveis pela visualização e análise dos dados coletados [10][11][12].

Os sinais analógicos são frequentemente gerados através da conversão de parâmetros físicos, como temperatura, pressão, umidade e tensão, em sinais elétricos correspondentes, utilizando sensores e transdutores especializados [11]. Em comparação com os sistemas tradicionais de medição (sistemas analógicos), os sistemas de aquisição de dados (DAQ) baseados em computador ou sistemas embebidos exploram as vantagens do processamento digital, da visualização gráfica final e dos recursos de conectividade [10][11].

Com essa abordagem, é possível obter uma solução de medição mais poderosa, flexível e com melhor custo-benefício, além de manter a precisão e a integridade do sinal de medição. Como resultado, os sistemas de aquisição de dados se tornam mais confiáveis e mantêm a qualidade do sistema em níveis elevados [13][12].

2.4 Sistema modular

A modularidade é a habilidade de um sistema em separar seus componentes e a forma como podem ser recombinaados, levando em consideração tanto o acoplamento entre componentes quanto as restrições impostas pela arquitetura do sistema em relação à mistura e combinação de componentes[14]. A maioria dos sistemas pode ser considerada modular até certo ponto, já que poucos possuem uma arquitetura fixa em que os componentes são inseparáveis e não podem ser rearranjados.

Diversos sistemas têm a tendência de migrar para uma modularidade maior. Sistemas que antes eram estritamente agregados podem ser separados, tendo seus componentes acoplados e misturados, o que permite uma flexibilidade muito maior. De acordo com a autora em [14], a modularidade auxilia no aumento da flexibilidade do sistema, proporcionando um aumento exponencial nas configurações possíveis para um determinado conjunto de entradas.

A arquitetura modular possibilitam a construção de sistemas personalizados pela recombinação de seus módulos. Utilizar sistemas modulares em unidades personalizadas apresenta diversas vantagens, como desenvolvimento mais rápido e econômico, confiabilidade e um caminho robusto para atualização e modificação [15].

2.5 Sistema escalável

Depois de modulabilidade o atributo mais desejável dentro de um sistema e sua capacidade de escalabilidade. Este conceito diz respeito à capacidade de um sistema se adaptar a um número crescente de elementos, objetos ou dados, informações ainda sendo suscetível à expansão[16].

Uma aplicação muito importante para os sistemas escaláveis utilizados neste projeto é a capacidade de um sistema de aquisição de dados processar e analisar a quantidade de dados fornecidos, aumentando ou diminuindo de acordo com o tempo. Combinado com sistemas modulares e escaláveis[15][16].

Para dimensionar um sistema ou um componente do sistema, é essencial conhecer a arquitetura do sistema, os algoritmos utilizados e como os componentes se comunicam. Os sistemas escaláveis podem funcionar bem com cargas baixas, mas conforme a carga aumenta, eles se tendem a degradar conforme a quantidade de dados processados aumente significativamente[17].

2.6 Protocolos de comunicação em rede

Protocolos de comunicação em rede são conjuntos de regras, procedimentos e formatos de dados que definem como os dispositivos em uma rede se comunicam entre si. Esses protocolos estabelecem padrões para a transmissão de dados, incluindo como os dados são estruturados, como são enviados e recebidos.

Os protocolos de comunicação em rede são essenciais para garantir que os dispositivos em uma rede possam se comunicar entre si de forma eficiente e segura. Eles permitem que

diferentes tipos de dispositivos em uma rede possam se comunicar, independentemente do hardware ou software que estão usando, e garantem que os dados sejam transmitidos com precisão e segurança[18].

2.6.1 TCP

O TCP (Transmission Control Protocol) é um protocolo de comunicação utilizado em redes de computadores para fornecer uma comunicação confiável e orientada a conexão entre dispositivos. O TCP divide os dados em pacotes antes de enviá-los, para que possam ser transmitidos de maneira eficiente e, em seguida, os reagrupa corretamente no destino. Cada pacote é numerado sequencialmente, o que facilita o controle e a reconstrução precisa dos dados na ponta receptora. Além disso, o TCP possui mecanismos para detectar e corrigir erros. Ele usa somas de verificação para verificar se os pacotes não foram corrompidos durante a transmissão, descartando aqueles que apresentam problemas. Se algum pacote for perdido durante a transmissão, o TCP se encarrega de retransmiti-lo para garantir que os dados sejam entregues de forma confiável. Em resumo, o TCP é um protocolo fundamental para a comunicação em redes de computadores. Ele garante uma transmissão confiável, orientada a conexão e com controle de congestionamento, sendo amplamente utilizado em aplicações que requerem uma comunicação segura e eficiente, acesso à web, e-mail e muitos outros.

2.6.2 UDP

O UDP (User Datagram Protocol) é um protocolo de transporte que foi projetado para oferecer uma maneira rápida e eficiente de transmitir dados de um host para outro. Ao contrário do TCP (Transmission Control Protocol), o UDP não fornece controle de congestionamento, controle de fluxo ou mecanismos de retransmissão. Em vez disso, o UDP simplesmente envia datagramas de um host para outro, sem garantia de entrega. A comunicação UDP é baseada em datagramas, que são pacotes de dados que são enviados de um host para outro. Cada datagrama UDP contém um cabeçalho e uma carga útil. O

cabeçalho do datagrama contém informações sobre a origem e o destino do pacote, bem como o número da porta usada para a comunicação. A carga útil do datagrama contém os dados que estão sendo transmitidos[19]. O UDP é frequentemente usado em aplicativos que exigem transmissão de dados em tempo real, como jogos online ou streaming de vídeo. Isso ocorre porque o UDP é rápido e eficiente, o que é importante para garantir uma boa experiência do usuário.

2.7 Rede de sensores sem fio

Rede de sensores sem fio ou em inglês Wireless sensor networks (WSN) nomenclatura mais utilizada no meio acadêmico. As redes de sensores sem fio são umas das tecnologias cruciais para o século XXI. através dos avanços na microeletrônica e tecnologias de comunicação sem fio, sensores inteligentes, baratos e minúsculos podem ser implantados em áreas físicas e conectados através de diversas tecnologias existentes, proporcionando oportunidades sem precedentes para várias aplicações civis e militares, como monitoramento ambiental, vigilância de campo de batalha e controle de processos industriais[20].

Diferente das redes de comunicação sem fio tradicionais, como sistemas celulares e redes ad hoc móveis, as WSNs possuem características únicas, incluindo implantações mais densas de nós, maior falta de confiabilidade dos nós de sensores e severas restrições de energia, computação e armazenamento, o que traz muitos desafios no desenvolvimento e aplicação dessas redes[21].

2.7.1 Na indústria

Na indústria, as Rede de sensores sem fio podem ser utilizadas para monitorar processos de fabricação ou as condições dos equipamentos e máquinas utilizadas. Por exemplo, sensores sem fio podem ser instalados em linhas de produção e montagem para monitorar e controlar o processo de fabricação. Plantas químicas ou refinarias de petróleo podem utilizar sensores para acompanhar a condição de seus extensos oleodutos. Sensores minúsculos podem ser inseridos em áreas de uma máquina que são inacessíveis aos seres

humanos para monitorar o estado da máquina e alertar sobre qualquer falha. Com as redes de sensores, a manutenção pode ser realizada com base no estado do equipamento, reduzindo significativamente o custo de manutenção, aumentando a vida útil da máquina e até mesmo salvando vidas[21].

2.8 Funcionamento da Atual Arquitetura

Na figura 2.4 podemos ver a arquitetura do sistema de aquisição de dados do projeto ON-SURF, onde possui comunicação e alimentação via cabos, tal como possui Raspberry PI para a visualização dos dados e comunicação com a base de dados.

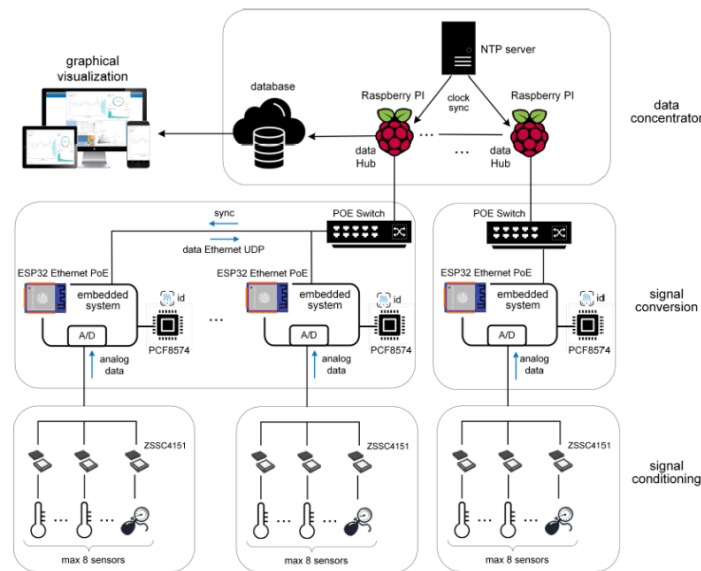


Figura 2.4: Arquitetura do sistema de aquisição de dados e condicionamento de sinal - ON-SURF

Fonte:[22]

A arquitetura do sistema possui quatro módulos, divididos em: Condicionamento de sinal, Conversão de sinal, Concentrador de dados e a Visualização. Cada módulo possui suas características e funcionalidades que empenham papéis cruciais na aquisição de dados, os módulos devem agir em conjunto cada módulo com sua etapa para o bom funcionamento do sistema[22].

O primeiro módulo "Condicionamento de Sinal", desempenhar o papel mais fundamental de todo o sistema, este módulo está em contato direto com os sensores, estes que atua diretamente com os fenômenos físicos, o condicionamento de sinal do projeto ON-SURF possui uma elevada limitação, isso se deve ao fato que os sensores de filme fino possuem características diversas comparando com os sensores comerciais. sendo assim tal processo de condicionamento de sinal deve possuir amplificação altamente precisa e correção específica de sinais de sensores em ponte, compensação digital do deslocamento do sensor, sensibilidade, desvio de temperatura e não linearidade [22] [6].

O segundo módulo "Conversão de sinal", tendo em vista os requisitos de modularidade e escalabilidade, os módulos de conversão de sinal foram desenvolvidos para combinar sinais de diferentes sensores e adicionar facilmente mais módulos, para isso foi utilizado o ESP32-Ethernet-PoE, que suporta até no máximo 8 parâmetros diferentes[22] [7], para possuir a característica de modularidade, foi inserido em cada ESP32 um expensor de I/Os remoto de 8bits no barramento I2C com o objetivo de identificar cada endereço de microcontrolador, como esse expensor é também definir o último octeto do endereço IP da interface [22], a conversão de analógico para digital é realizada com aproximação sucessiva de 12 bits do ESP32 para uma tensão de referência de 3,3V [7]. Os dados convertidos são enviados ao switch via comunicação Ethernet UDP utilizando o esquema unicast.

O terceiro e quarto módulo "concentrador de dados e visualização", o concentrador de dados é baseado em um RaspberryPi que trabalha como intermediário, isso é entre a camada inferior que são o condicionamento de sinal e conversão de sinal é a camada superior que é visualização na Cloud [22] [23], foi desenvolvido dessa maneira para que cada módulo seja responsável por uma determinada função sem sobre carregar os demais módulos, garantindo assim uma eficiência e sincronismo.

Capítulo 3

Proposta de Solução

O sistema de aquisição de dados é um equipamento de extrema importância para a indústria e sociedade, visto quão utilitário é para o controle, automação e inspeção de qualidade, sendo assim cada vez mais sendo desenvolvidos em diversas áreas, principalmente na indústria, a busca por sistemas de permitam uma escalabilidade e reconfiguração sem necessidade de grandes investimentos.

Este projeto visa desenvolver uma versão mais "LITE" do sistema de aquisição de dados do projeto ON-SURF onde o sistema possua comunicação sem fio que possa ser escalável e modular, e um protocolo que garanta esta escalabilidade de forma ampla e organizada.

3.1 Proposta de Solução

O desenvolvimento desses sistemas de aquisição de dados com uma arquitetura adaptável e modular é extremamente relevante, pois eles podem ser aplicados em uma variedade de máquinas industriais, como prensas, prensas contínuas, esteiras e muitas outras. Esses sistemas são capazes de lidar com diferentes variáveis, incluindo pressão, temperatura e força, e são projetados para aceitar qualquer tipo de sensor. A arquitetura já desenvolvida anteriormente oferece garantias para todos esses aspectos. No entanto, ela também apresenta algumas limitações, especialmente em relação à comunicação por fio.

A comunicação por fio impõe restrições significativas à reconfiguração dos layouts nas

indústrias, pois requer a instalação de equipamentos como switches, cabeamentos e a contratação de mão de obra para essas instalações. Além disso, há a necessidade de um broker para a concentração de dados e a utilização de dispositivos como o Raspberry Pi, o que acaba resultando em uma estrutura inflexível.

Essas limitações levam à proposta de um sistema mais leve, capaz de superar os desafios mencionados. A ideia é eliminar a necessidade de comunicação cabeada, substituindo-a por uma comunicação sem fio. Dessa forma, será possível reduzir a dependência de equipamentos como switches e cabeamentos, eliminando a rigidez da estrutura atual. Além disso, não será mais necessário utilizar um broker para a concentração de dados, e a utilização do Raspberry Pi poderá ser dispensada.

A proposta para um sistema mais "LITE" é apresentada na figura 3.1, utilizando comunicação sem fio para garantir os aspectos anteriores de escalabilidade, adaptabilidade, reconfigurabilidade e robustez. Essa solução consiste no desenvolvimento de um módulo de comunicação que utilize o mínimo de equipamentos necessário. Esse módulo já está integrado aos ESP32, que foram utilizados em trabalhos anteriores, e requer a criação de um protocolo de comunicação específico.

Essa abordagem busca simplificar a estrutura do sistema, eliminando a dependência de uma comunicação cabeada e aproveitando os benefícios da comunicação sem fio. Ao trabalhar com menos equipamentos, o sistema se torna mais leve e flexível, permitindo uma maior facilidade na reconfiguração dos layouts industriais. Além disso, o desenvolvimento de um protocolo particular de comunicação contribui para a otimização do sistema, garantindo uma transmissão eficiente e confiável dos dados adquiridos.

Com essa proposta, busca-se alcançar uma solução que atenda às necessidades das diversas máquinas industriais, aceitando diferentes variáveis de interesse, como pressão, temperatura e força. O objetivo é garantir uma arquitetura adaptável e modular, capaz de se ajustar às demandas específicas de cada aplicação. Através do uso de uma comunicação sem fio e da redução da complexidade dos equipamentos, o sistema se torna mais eficiente, econômico e flexível, proporcionando uma solução "LITE" de aquisição de dados para a indústria.

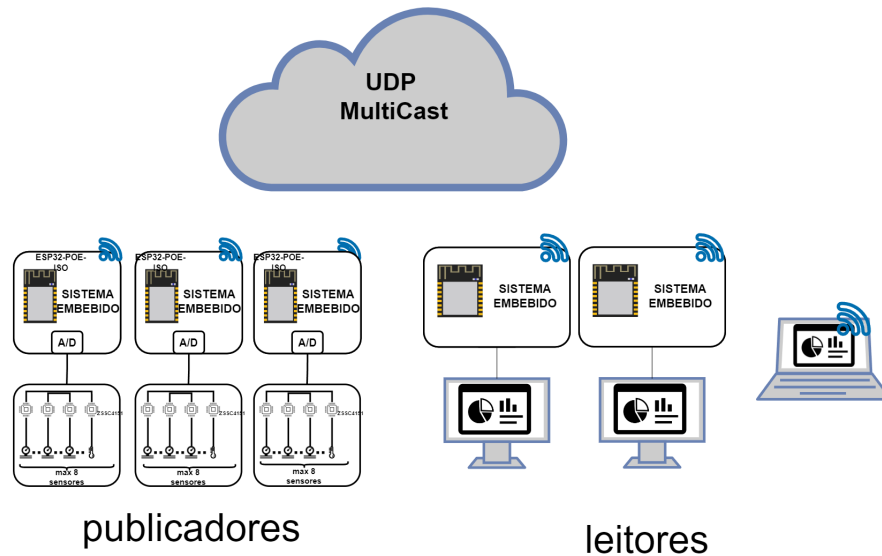


Figura 3.1: Arquitetura

Fonte:autoria própria

Como proposta para essa arquitetura, optaremos por utilizar a comunicação UDP (User Datagram Protocol), pois esse protocolo oferece uma comunicação leve e robusta. Além disso, é de fácil acesso e possui um overhead (custo adicional) inferior a outros protocolos. Para garantir uma organização e clareza nas transmissões de mensagens, será adotado o Protocolo Não Orientado à Conexão para Rede de Sensores Distribuídos.

A escolha pelo UDP se deve às suas características favoráveis para essa aplicação. Ele é conhecido por sua simplicidade e eficiência na transmissão de dados, especialmente em ambientes com restrições de recursos. A natureza não orientada à conexão do UDP permite uma comunicação direta entre os dispositivos, sem a necessidade de estabelecer uma conexão prévia.

A utilização do UDP também oferece uma maior flexibilidade no desenvolvimento da arquitetura, uma vez que ele não impõe restrições de controle de fluxo ou confirmação de recebimento de dados. Isso permite que o sistema seja adaptável a diferentes necessidades e requisitos específicos de cada aplicação.

Para garantir a organização e clareza na transmissão das mensagens, será adotado o Protocolo Não Orientado à Conexão para Rede de Sensores Distribuídos, este apresentado

no capítulo 4. Esse protocolo estabelece um conjunto de regras e convenções para a transmissão de dados em redes de sensores distribuídos, garantindo uma comunicação eficiente e confiável. Ele define a estrutura das mensagens, a forma como são transmitidas e como são interpretadas pelos dispositivos receptores.

Dessa forma, ao utilizar a comunicação UDP em conjunto com o Protocolo Não Orientado à Conexão para Rede de Sensores Distribuídos, a arquitetura proposta se beneficia de uma comunicação leve, eficiente e confiável, capaz de atender às demandas de transmissão de dados em um sistema de aquisição de dados distribuído.

Capítulo 4

Desenvolvimento/Implementação

Quando falamos de sistemas de aquisição de dados, referindo a um conjunto de dispositivos e softwares que permitem a coleta, processamento e armazenamento de informações em tempo real. Esses sistemas são amplamente utilizados em áreas como engenharia, ciência e tecnologia, onde a precisão e a confiabilidade dos dados são fundamentais.

Para garantir que esses sistemas funcionem corretamente, é importante entender quais são seus componentes essenciais e em quais deles deve-se concentrar o desenvolvimento. Alguns dos componentes mais importantes incluem sensores, amplificadores, conversores analógico-digital, processadores e software de controle.

Quando se trata de criar uma versão "lite" de um sistema de aquisição de sinal, é possível focar em alguns desses componentes, principalmente no condicionamento de sinal. O condicionamento de sinal refere-se à manipulação dos sinais brutos coletados pelos sensores, com o objetivo de melhorar sua qualidade e torná-los mais adequados para a análise.

No entanto, se quisermos melhorar a escalabilidade e modularidade de um sistema de aquisição de dados, é importante considerar outros componentes ou conjunto de componentes. Isso pode envolver a otimização do hardware e do software, a escolha de tecnologias de comunicação eficientes e a implementação de técnicas de gerenciamento de dados para garantir que os dados sejam armazenados e acessados de maneira rápida e confiável.

Para utilizar o máximo dos trabalhos anteriormente desenvolvidos, optamos por desenvolver uma versão "lite" alterando a forma de comunicação do atual projeto, assim focando em reduzir toda a estrutura desenvolvida para a comunicação com fios.

4.1 Comunicação

A comunicação é uma parte importante do sistema de aquisição de dados e pode ser realizada utilizando diversas tecnologias disponíveis no mercado. Algumas dessas tecnologias são a radiofrequência, wireless, Bluetooth, infravermelho e Ethernet. Cada uma dessas tecnologias possui suas próprias características, limitações e vantagens.

A escolha da tecnologia mais adequada para a comunicação no sistema de aquisição de dados depende muito da aplicação do projeto e de como ele se desenvolverá ao longo do tempo. Por exemplo, a radiofrequência é uma boa opção para comunicação de dados em longas distâncias, enquanto o Bluetooth é mais indicado para comunicação em curta distância.

A tecnologia infravermelho é útil para comunicação em ambientes com interferência eletromagnética, mas tem a limitação de que o receptor deve estar na linha de visão do transmissor. A Ethernet é uma opção mais robusta para comunicação de dados em ambientes de rede e é amplamente utilizada em sistemas de automação industrial.

A comunicação via wireless é uma das tecnologias de comunicação mais populares atualmente. Ela permite que dispositivos se comuniquem sem fios, o que possibilita maior mobilidade e flexibilidade no sistema. Essa tecnologia tem se expandido significativamente nos últimos anos, sendo adotada em diversos setores, como na indústria, no comércio, na saúde e na educação.

Uma das vantagens da comunicação wireless é a possibilidade de modularidade, ou seja, a facilidade de adicionar e remover dispositivos da rede sem a necessidade de cabos ou fios adicionais. Além disso, a tecnologia também amplia a capacidade de escalabilidade do sistema de aquisição de dados em rede, permitindo que a rede cresça em termos de dispositivos conectados e cobertura de área de maneira mais flexível e econômica.

Tendo como informação da capacidade de comunicação do ESP32-PoE-ISO, onde o mesmo possui comunicação via Ethernet e modulo Wifi, além da comunicação serial. Desenvolvemos a primeira versão "LITE" do sistema com comunicação serial.

4.2 Comunicação serial

A comunicação serial é um método de transferência de dados que transmite bits de dados sequencialmente, um após o outro, por meio de uma única linha de comunicação. Esse tipo de comunicação é amplamente utilizado em sistemas embarcados, sistemas de aquisição de dados, redes industriais e comunicações entre dispositivos eletrônicos.

Existem vários protocolos de comunicação serial, incluindo SPI (Serial Peripheral Interface) e I2C (Inter-Integrated Circuit). Cada protocolo tem suas próprias especificações, limitações e requisitos de hardware.

A comunicação serial tem a vantagem de ser simples e econômica, com requisitos de hardware mínimos. Ela também pode ser facilmente implementada em microcontroladores e outros dispositivos eletrônicos, permitindo a transferência de dados em tempo real.

Tendo como objetivo a redução do sistema, tornando-o mais "LITE", inicialmente, removemos toda a estrutura de comunicação do sistema anterior que utilizava cabos Ethernet para alimentar os sistemas de aquisição de dados. Esses sistemas eram conectados a um switch, que tinha um concentrador de dados ligado a um Raspberry Pi que enviava as informações para a nuvem. Após isso, faríamos a visualização dos dados.

Em seguida, desenvolvemos um sistema simples com comunicação serial para software Arduíno, como podemos ver na arquitetura da figura 4.1.

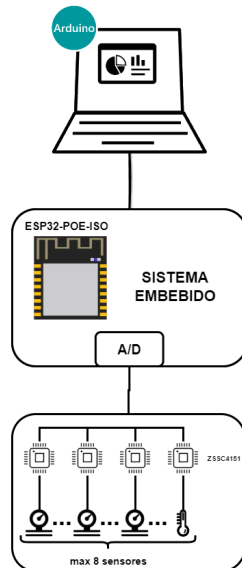


Figura 4.1: 1ª versão *LITE* do sistema de aquisição de dados

Fonte: autoria própria

4.3 Comunicação MQTT

O protocolo MQTT (Message Queue Telemetry Transport) é uma tecnologia de comunicação leve, aberta e altamente escalável, que é utilizada em ambientes de rede com baixa largura de banda e dispositivos com limitações de processamento e energia. Esse protocolo é comumente utilizado em sistemas de aquisição de dados para conectar dispositivos IoT (Internet of Things) à nuvem.

A comunicação MQTT é baseada em um modelo de publicação e assinatura *publish/subscribe*, onde os dispositivos publicam mensagens em tópicos específicos e os aplicativos ou sistemas de nuvem que desejam receber esses dados se inscrevem nesses tópicos. Isso permite uma transmissão eficiente e seletiva de dados, reduzindo a largura de banda necessária e o tempo de processamento.

Na aquisição de dados, o protocolo MQTT é utilizado para coletar informações de sensores ou dispositivos IoT e enviá-los para uma nuvem ou servidor, onde podem ser armazenados e processados. Ele também é utilizado para enviar comandos ou configurações para os dispositivos, permitindo que sejam controlados remotamente.

A principal vantagem da comunicação MQTT é sua eficiência em termos de largura de banda e processamento, o que a torna uma opção viável para ambientes com recursos limitados. Além disso, o protocolo possui uma grande comunidade de desenvolvedores e é suportado por muitas plataformas de nuvem, tornando sua integração em sistemas existentes relativamente fácil.

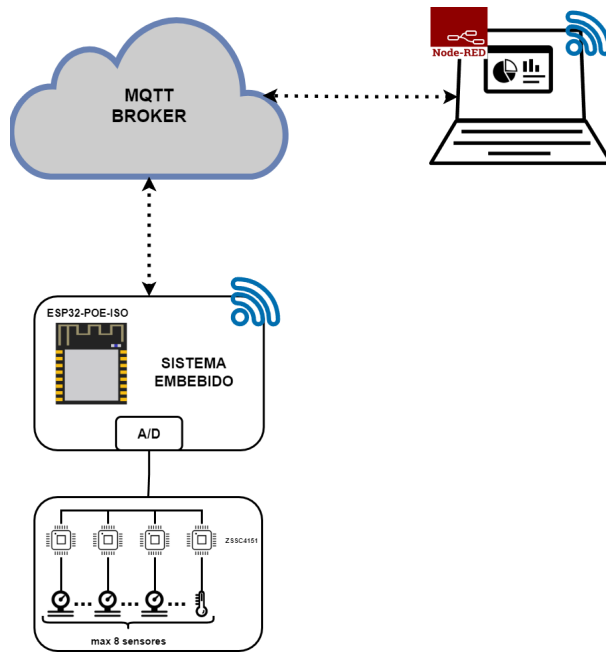


Figura 4.2: 2ª versão *LITE* do sistema de aquisição de dados

Fonte:autoria própria

4.4 Comunicação UDP

A comunicação UDP multicast é uma técnica utilizada para enviar dados de um transmissor para múltiplos receptores simultaneamente. Nesse tipo de comunicação, o transmissor envia pacotes UDP para um endereço IP multicast específico, que é um endereço especial reservado para esse propósito. Os receptores interessados em receber os dados se juntam ao grupo multicast usando o mesmo endereço IP.

A comunicação UDP multicast é particularmente útil em redes em que é necessário transmitir informações para vários dispositivos ao mesmo tempo, como em aplicações de

streaming de vídeo ou em jogos online. É uma forma mais eficiente em termos de largura de banda do que a comunicação unicast (que envia dados para um único receptor), já que os dados são enviados uma única vez e todos os dispositivos interessados recebem as informações.

Essa técnica também é conhecida por ser confiável e escalável, já que ela permite que os dados sejam transmitidos para um grande número de dispositivos, sem afetar significativamente o desempenho da rede. Além disso, a comunicação UDP multicast é suportada por muitos sistemas operacionais e protocolos de rede, o que a torna amplamente utilizada em diferentes aplicações.

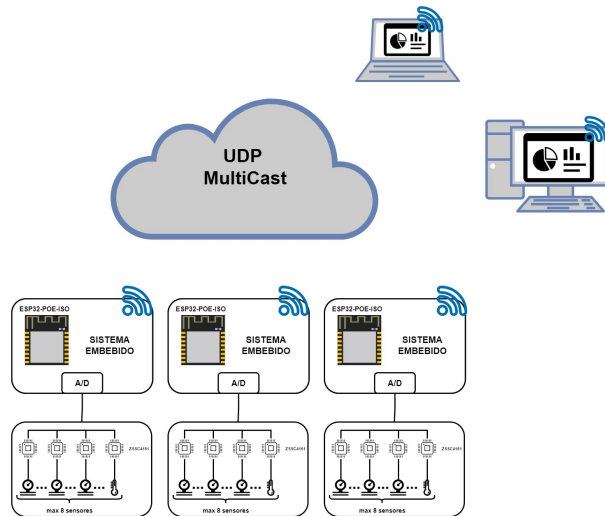


Figura 4.3: versão final *LITE* do sistema de aquisição de dados

Fonte: autoria própria

4.5 Sistema com comunicação sem fio

O sistema atualmente desenvolvido adota uma abordagem que elimina por completo a necessidade de comunicação por fio, resultando em uma redução significativa na quantidade de hardware envolvido. Essa mudança tem como consequência um aumento considerável na modulabilidade do sistema, permitindo maior flexibilidade e adaptabilidade em relação aos nós presentes no sistema.

Ao remover a comunicação por fio, o sistema se beneficia de uma série de vantagens. Em primeiro lugar, há uma diminuição no número de dispositivos físicos necessários para estabelecer as conexões cabeadas, como cabos, switches e outros equipamentos relacionados. Isso resulta em uma simplificação da infraestrutura, reduzindo custos e tornando o sistema mais eficiente.

Além disso, a ausência de fios permite uma maior liberdade na configuração e reconfiguração do sistema. Sem a limitação física dos cabos, é possível adicionar ou remover nós de forma mais ágil e sem grandes obstáculos. Isso proporciona uma maior modularidade, permitindo a expansão ou adaptação do sistema de acordo com as necessidades específicas.

Essa abordagem também tem o potencial de facilitar a manutenção do sistema. A ausência de fios reduz a necessidade de lidar com problemas relacionados ao cabeamento, como possíveis falhas, desconexões ou problemas de conexão. Isso simplifica a identificação e solução de problemas, otimizando o tempo e os recursos dedicados à manutenção do sistema.

Em resumo, a eliminação da comunicação por fio no sistema atual resulta em benefícios significativos, como a redução de hardware, maior modularidade e simplificação da infraestrutura. Essa abordagem promove uma maior flexibilidade, adaptabilidade e facilidade na configuração e manutenção do sistema, contribuindo para um ambiente mais eficiente e eficaz no contexto da aquisição de dados

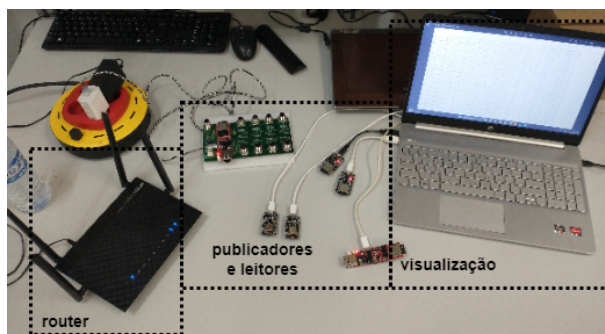


Figura 4.4: Sistema de aquisição com comunicação sem fio

Fonte: autoria própria

4.6 Protocolo Não Orientado À Conexão Para Rede De Sensores Distribuídos

Protocolo Não Orientado À Conexão (Non-Connection Oriented Protocol, NCP) é um protocolo de rede que se destaca pela sua simplicidade e eficiência em ambientes com recursos limitados, como é o caso de redes de sensores distribuídos.

Redes de sensores distribuídos consistem em um conjunto de dispositivos pequenos e autônomos que coletam dados do ambiente e os transmitem para um dispositivo central, como um computador ou um servidor. Esses dispositivos são geralmente limitados em termos de capacidade de processamento, memória e energia, o que significa que a comunicação entre eles deve ser eficiente e consumir o mínimo de recursos possível.

O NCP é um protocolo de camada de rede que permite que os dispositivos em uma rede de sensores distribuídos se comuniquem sem a necessidade de estabelecer uma conexão prévia entre eles. Isso significa que os dispositivos podem enviar pacotes de dados independentemente, sem precisar esperar por uma resposta do dispositivo receptor. O protocolo é baseado em datagramas, que são unidades independentes de dados que contêm informações de cabeçalho.

Uma das principais vantagens do NCP é a sua eficiência em termos de consumo de recursos. Por não requerer a estabelecimento de uma conexão prévia, o protocolo economiza recursos de rede e processamento. Além disso, o protocolo é capaz de lidar com perda de pacotes e erros de transmissão de forma eficiente, o que o torna adequado para ambientes com alta taxa de erro.

4.7 Análise da estrutura do protocolo

A estrutura do protocolo desenvolvido tem como objetivo garantir a escalabilidade e modularidade do sistema, utilizando um formato de mensagem padronizado. Essa mensagem é composta por seis partes essenciais: Tipo de Pedido, ID do Equipamento, ID do Sensor, Variável, Número de Bytes a serem transmitidos e os Dados em si.

Essa abordagem permite a flexibilidade de adicionar ou remover módulos no sistema sem a necessidade de reconfigurar a rede Wi-Fi. Antes de adicionar os módulos ao sistema, é necessário realizar sua configuração, definindo quais dados eles estão interpretando e enviando para os leitores.

Dessa forma, o protocolo oferece uma solução que simplifica o processo de expansão do sistema, ao mesmo tempo em que permite uma fácil integração dos módulos adicionados. Essa característica contribui para a modularidade e escalabilidade do sistema como um todo, facilitando a sua manutenção e adaptação às necessidades específicas de cada contexto de aplicação.

O presente protocolo se encontra em anex A.

TIPO DE PEDIDO	EQUIPAMENTO ID	SENSOR ID	VARIAVEL	Nº DE BITS QUE SEGUE	DADOS
QQ	YYYY	ZZZZ	WW	XX	????
1 Byte	2 Bytes	2 Bytes	1 Byte	1 Bytes	Dados

Figura 4.5: Protocolo Não Orientado À Conexão Para Rede De Sensores Distribuídos

Fonte:autoria própria

4.8 Interpretação do protocolo

A interpretação do protocolo pode ser realizada de diversas maneiras, dependendo das necessidades do projeto. Neste trabalho, desenvolvemos um código em linguagem Python que tem a função de processar os bytes recebidos pelos módulos leitores e convertê-los em uma mensagem de texto (string) legível e manipulável. Isso abre possibilidades para realizar diferentes operações com esses dados. Por exemplo, podemos realizar manipulações na string, como transformá-la em formato JSON, que facilita o armazenamento e a análise dos dados, por outras plataformas.

Além disso, o código também oferece a opção de criar arquivos nos formatos CSV ou XLS, permitindo a exportação dos dados para planilhas ou outros sistemas de análise externos. Essa flexibilidade na manipulação dos dados coletados proporciona uma maior versatilidade no tratamento e na utilização dessas informações, de acordo com as necessidades específicas do projeto em questão.

Capítulo 5

Análise e Discussão de Resultados

Neste capítulo, são apresentados os testes realizados para verificar se o projeto desenvolvido cumpre os objetivos assumidos e resolve, de fato, o problema descrito.

No entanto, devido à impossibilidade de se contar com uma grande rede de sensores para validar alguns dos testes, optou-se por simular a aplicabilidade com o uso de dados aleatórios.

5.1 Teste de comunicação UDP

Inicialmente, para realizar os testes de comunicação utilizando o protocolo UDP, foram utilizados dois dispositivos ESP32. Um deles foi configurado como ponto de acesso (AP) e leitor, enquanto o outro foi configurado como publicador. No entanto, durante a simulação, foi observada uma grande dificuldade, além do fato de que o ESP32 configurado como AP estava sobrecarregado. Essa situação levou a equipe a buscar alternativas para contornar o problema.

A solução encontrada foi utilizar um roteador genérico de uso residencial como ponto de acesso, aliviando assim o ESP32 dessa função e aumentando a sua capacidade de conexões e distância. Isso se mostrou crucial, visto que o ESP32 saturava com mais de cinco conexões simultâneas.

Dessa forma, a estrutura inicial foi finalizada com dois dispositivos ESP32, um deles

operando como publicador na rede, enquanto o outro operava como leitor na rede criada pelo roteador.

5.1.1 Taxa de perda de pacote

Com o objetivo de verificar a confiabilidade da comunicação em UDP, que não oferece garantia de entrega dos pacotes, foram realizados testes para calcular a taxa de perda de pacotes em diferentes intervalos de tempo de publicação. Essa análise permitiu simular a situação real e verificar a efetividade da comunicação em condições adversas. Ademais, além de avaliar a taxa de perda de pacotes, os testes também permitiram avaliar a capacidade do leitor em receber e capturar as mensagens transmitidas dentro do intervalo de tempo, a fim de garantir que todas as informações sejam devidamente processadas e armazenadas.

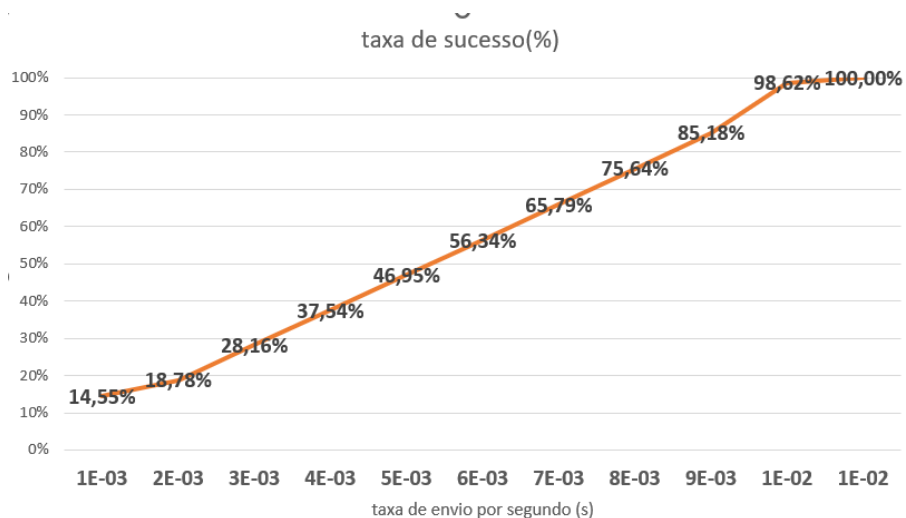


Figura 5.1: Taxa de sucesso na entrega

Fonte: autoria própria

Neste gráfico, podemos analisar a taxa de sucesso, que representa a capacidade do publicador enviar dados e o leitor recebê-los com êxito. É possível perceber que, com uma taxa de envio de 1 milissegundo, a taxa de sucesso é de apenas 14,55%, enquanto que, conforme diminuimos a taxa de envio, há um crescimento praticamente linear da

taxa de sucesso, alcançando 98,62% com 1 centissegundo e 100% com 1,1 centissegundos.

Durante a discussão dos resultados, foi constatado que a perda de algum pacote durante a transmissão não afetará o desempenho, uma vez que a frequência de operação das máquinas é inferior a essa taxa. Além disso, é possível armazenar os pacotes no ESP32 e enviá-los com uma taxa segura de transmissão, garantindo a entrega das informações sem prejuízo para o desempenho.

5.2 Modularidade e escalabilidade

Os testes de modularidade e escalabilidade são importantes para avaliar o desempenho e a eficiência do sistema em relação à sua capacidade de lidar com a complexidade e o aumento da rede sensores.

O teste de modularidade é realizado para avaliar a capacidade do sistema de ser dividido em módulos independentes, que possam ser desenvolvidos, testados e mantidos separadamente. O objetivo é garantir que as alterações em um módulo não afetem negativamente outros módulos, facilitando assim a manutenção e evolução do sistema.

Já o teste de escalabilidade é realizado para avaliar a capacidade do sistema de lidar com um aumento da carga de trabalho, ou seja, da demanda por recursos computacionais. Esse tipo de teste é essencial para garantir que o sistema possa suportar o aumento do número de usuários ou transações sem comprometer o desempenho ou a qualidade do serviço prestado.

Pensando em realizar os testes de modularidade e escalabilidade simultaneamente, considerando as limitações de hardware que tínhamos no momento, decidimos desenvolver uma arquitetura com os recursos disponíveis: 5 ESP32, um roteador e um computador. Para isso, foram criados três cenários diferentes, com quatro publicadores e um leitor, quatro leitores e um publicador e, por último, três publicadores e dois leitores, o computador foi utilizado para a verificação do fluxo de mensagens dentro da rede ele também pode ser implementado como leitor.

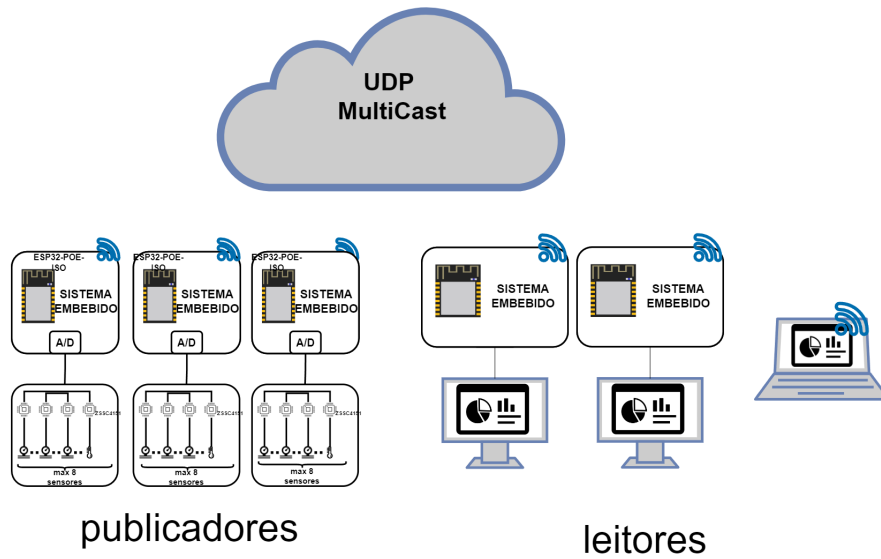


Figura 5.2: 3 Publicadores e 2 Leitores

Fonte:autoria própria



Figura 5.3: implementação

Fonte:autoria própria

5.2.1 Com 4 publicadores e 1 leitor

O teste realizado com o maior número de publicadores tem como objetivo primordial testar e analisar a capacidade de modulabilidade do sistema, bem como sua reconfiguração. Isso é especialmente relevante, considerando que os sensores podem estar distribuídos em múltiplas posições distintas, com uma infinidade de variações possíveis dentro do layout da fábrica. Dessa forma, é fundamental garantir que o sistema seja capaz de lidar eficientemente com essa distribuição espacial dos sensores, permitindo a coleta e o processamento de dados de maneira eficaz e confiável. A realização desse teste permitirá avaliar a capacidade do sistema em lidar com um ambiente complexo, no qual os sensores estão localizados em diferentes pontos, proporcionando "insights" valiosos sobre a escalabilidade e a modularidade do sistema em um contexto real.

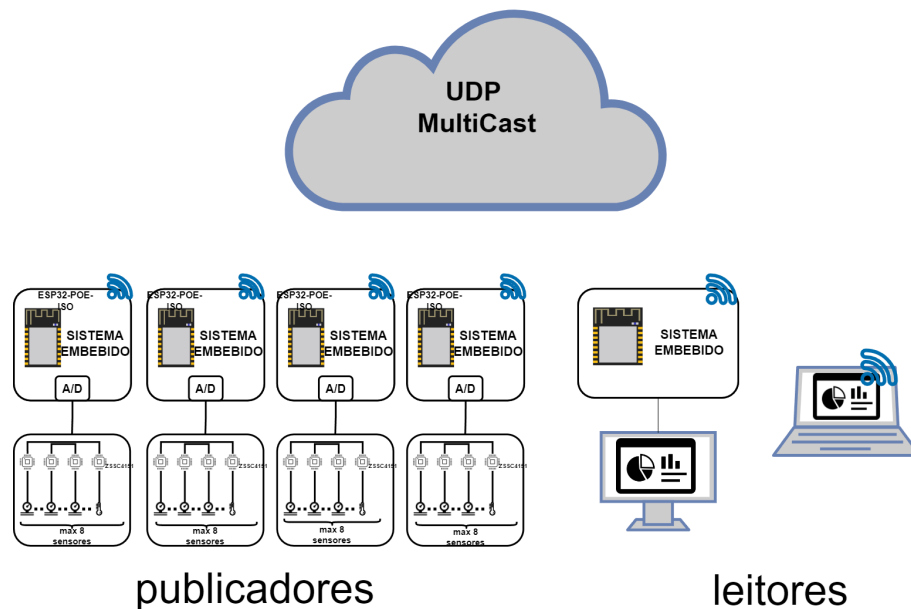


Figura 5.4: 4 publicadores e 1 leitor

Fonte: autoria própria

5.2.2 Com 4 leitores e 1 publicador

Assim como no teste anterior, a modularidade deve ser avaliada tanto para os leitores quanto para os publicadores, garantindo a capacidade de expansão em ambos os lados da troca de informações. Além disso, durante esse teste, também serão avaliadas a capacidade de escalabilidade do sistema desenvolvido neste trabalho, considerando a possibilidade de adicionar e remover módulos conforme necessário. Essa análise permitirá verificar a capacidade do sistema em se adaptar a mudanças e incorporar novos componentes de forma eficiente, fornecendo uma base sólida para futuras expansões e aprimoramentos do sistema.

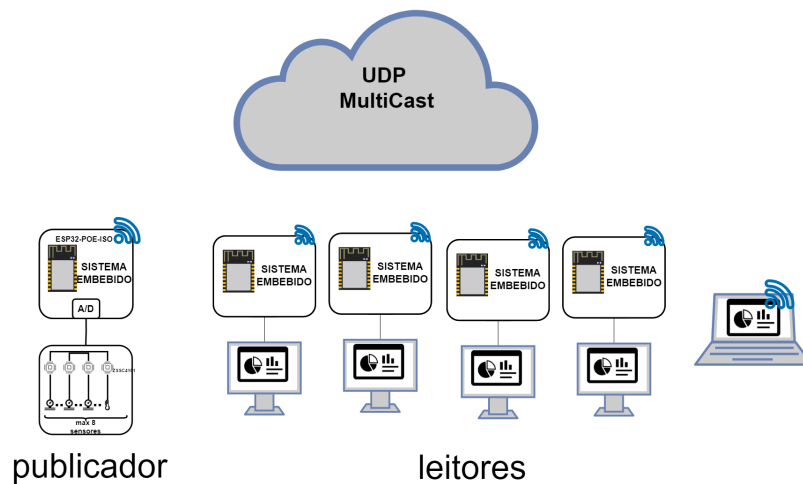


Figura 5.5: 4 leitores e 1 publicador

Fonte: autoria própria

5.3 Robustez

Para avaliar a robustez do sistema, conduzimos um teste abrangente, no qual procedemos à remoção e adição de nós de comunicação enquanto o sistema estava em pleno funcionamento. O objetivo era verificar se o sistema seria capaz de lidar com essas mudanças sem interrupções ou alterações indesejadas.

O teste foi conduzido de forma meticulosa, removendo cuidadosamente os nós de comunicação e adicionando-os novamente ao sistema, um por um. Durante todo o processo, monitoramos atentamente o desempenho do sistema, observando se ocorriam falhas ou instabilidades.

Felizmente, o resultado do teste foi um sucesso. O sistema se mostrou altamente robusto, sendo capaz de lidar com as alterações nos nós de comunicação sem afetar a funcionalidade geral. Não houve interrupções na aquisição e transmissão de dados, e o sistema manteve sua integridade e desempenho esperados.

Esse resultado é um indicativo importante da confiabilidade e capacidade de adaptação do sistema. A capacidade de realizar alterações nos nós de comunicação sem prejudicar o funcionamento geral do sistema é um atributo valioso, permitindo uma maior flexibilidade e escalabilidade em ambientes industriais.

Com base nos resultados positivos desse teste de robustez, podemos afirmar que o sistema demonstrou sua capacidade de enfrentar mudanças nas configurações de comunicação sem comprometer sua funcionalidade. Isso oferece uma maior tranquilidade e confiança na utilização do sistema em ambientes industriais dinâmicos e em constante evolução.

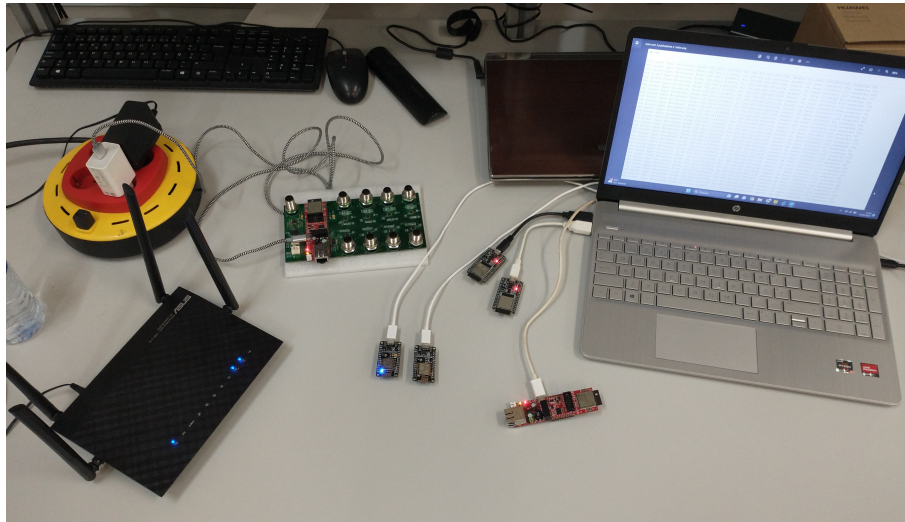


Figura 5.6: Funcionamento com 5 ESP

Fonte: autoria própria

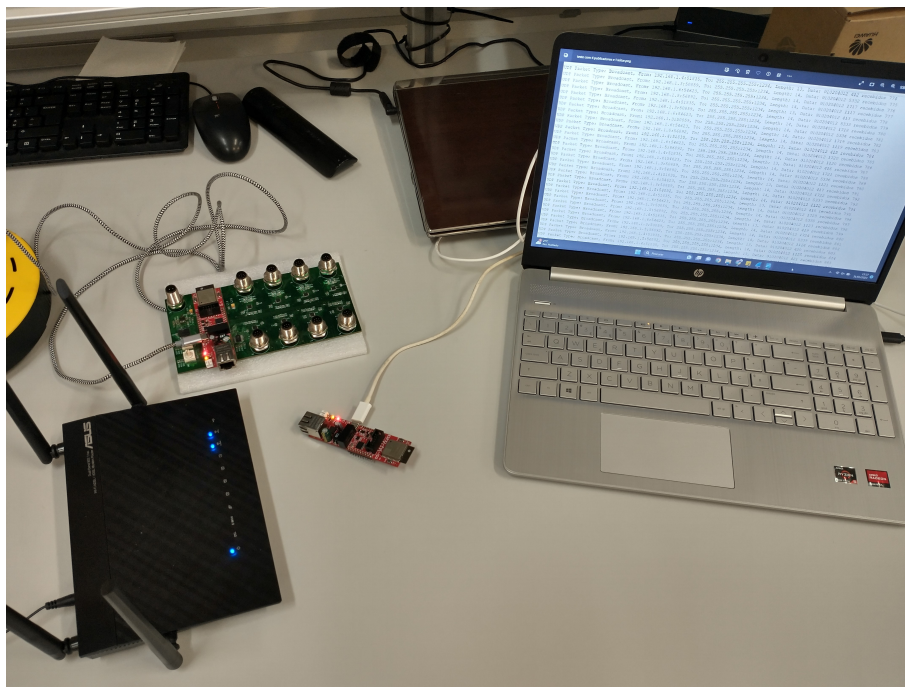


Figura 5.7: Funcionamento com 2 ESP

Fonte: autoria própria

Capítulo 6

Conclusões

Com a finalização deste trabalho, que teve como objetivo o desenvolvimento de um sistema de aquisição de dados modular e escalável com comunicação sem fio, aplicável em diversas indústrias e capaz de lidar com diferentes grandezas, como tensão superficial, temperatura, pressão e vibração, alcançamos os seguintes objetivos específicos: a criação de uma versão Lite do sistema de aquisição, o desenvolvimento de uma versão escalável e modular sem fio e a elaboração de um protocolo que garanta a escalabilidade.

Durante o desenvolvimento deste trabalho, conduzimos diversos testes para avaliar a eficiência e a funcionalidade do sistema proposto. Realizamos testes de comunicação utilizando o protocolo UDP, os quais demonstraram a capacidade de transmissão de dados de forma eficiente. Além disso, conduzimos testes de escalabilidade, modularidade e robustez, que comprovaram a adaptabilidade do sistema a diferentes configurações e necessidades.

Uma contribuição significativa deste trabalho é a criação de um protocolo que garante a escalabilidade do sistema sem a necessidade de configurações complexas na rede WiFi. Esse protocolo possibilitou a expansão simplificada do sistema de aquisição de dados, oferecendo flexibilidade e facilidade de implementação em diferentes contextos industriais.

Com base nos resultados obtidos nos testes realizados, concluímos que os objetivos propostos foram alcançados de forma satisfatória. O sistema de aquisição de dados desenvolvido revelou-se modular, escalável e capaz de lidar com diversas grandezas, atendendo,

assim, às necessidades da indústria como um todo.

Recomenda-se a realização de futuros estudos para explorar ainda mais as aplicações desse sistema em diferentes setores industriais. Esses estudos podem se concentrar em analisar os benefícios do sistema em contextos específicos, como a indústria automotiva, de alimentos ou farmacêutica, por exemplo. Além disso, é importante considerar aprimoramentos adicionais nas funcionalidades do sistema, como a incorporação de recursos avançados de análise de dados e a implementação de algoritmos de aprendizado de máquina para obter insights valiosos.

Outra sugestão relevante é o desenvolvimento de uma interface de visualização por meio de um dashboard. Essa visualização gráfica e intuitiva permitiria aos usuários monitorar e analisar facilmente os dados adquiridos pelo sistema em tempo real, fornecendo informações valiosas para a tomada de decisões estratégicas. Através do dashboard, os usuários poderiam ter uma visão abrangente do desempenho do sistema, identificar tendências, detectar anomalias e ajustar os parâmetros de operação conforme necessário.

Além disso, é recomendável considerar a implementação de um sistema de armazenamento na nuvem para os dados adquiridos. Essa abordagem oferece uma série de benefícios, como a redução da carga de armazenamento local, a escalabilidade para acomodar grandes volumes de dados e a acessibilidade remota aos dados de qualquer lugar e a qualquer momento. O armazenamento na nuvem também pode permitir a integração com outras ferramentas e sistemas, facilitando a análise avançada e o compartilhamento de dados com partes interessadas relevantes.

A partir deste trabalho, espera-se que sejam impulsionadas importantes melhorias e avanços na área de aquisição de dados, resultando em benefícios significativos para o setor industrial. Essas melhorias têm o potencial de impulsionar a eficiência dos processos industriais, otimizando a produção, reduzindo custos e minimizando desperdícios.

Bibliografia

- [1] C. Winkelmann e W. Lang, “Sensorial surfaces-embedding sensor structures into the surface of materials,” *IEEE Sensors Journal*, vol. 14, pp. 2078–2083, 7 2014, ISSN: 1530437X. DOI: 10.1109/JSEN.2013.2290134.
- [2] N. Rodrigues, J. Lima, P. J. Rodrigues e J. A. Carvalho, “Modular Data Acquisition Architecture for Thin-Film Sensors Surfaces,” *IEEE*, 2020.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam e E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, n.º 4, pp. 393–422, 2002, ISSN: 1389-1286. DOI: [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4). URL: <https://www.sciencedirect.com/science/article/pii/S1389128601003024>.
- [4] A. C. Djedouboum, A. A. A. Ari, A. M. Gueroui, A. Mohamadou e Z. Aliouat, “Big data collection in large-scale wireless sensor networks,” *Sensors (Switzerland)*, vol. 18, 12 dez. de 2018, ISSN: 14248220. DOI: 10.3390/s18124474.
- [5] On-surf, *Projeto ON-SURF*, <http://onsurf.teandm.pt/>, Fevereiro de 2023.
- [6] M. A. Bussmann, “Condicionamento de sinal para sistema de aquisição de dados em superfícies metálicas finas,” 2020.
- [7] N. Nonato, J. Luís, S. De et al., “Data acquisition system for fine surface process monitoring,” 2020.
- [8] J. V. G. Neto e P. J. R. N. R. F. L. B. Bragança, “Development of a sensory data concentrator for network broadcasting,” 2020.

- [9] G. Felipe, N. Ferreira, P. Alexandre e V. Alves, “Plataforma de análise e visualização de dados para sistemas IoT industriais baseada em métodos de Big Data,” 2020.
- [10] D. D. A. Blog, *Types of data acquisition systems*, <https://daqifi.com/types-of-data-acquisition-systems/>, 2023.
- [11] D. A. Systems, *A reliable source for Data Acquisition System information and leading Data Acquisition System Companies Manufacturers*. <https://www.dataacquisitionssystems.com/>, 2023.
- [12] L. M. V. Serrano, C. J. O. P. J. Alcobia, M. L. O. S. Mateus e M. C. G. Silva, “Sistemas de Aquisição, Processamento e Armazenamento de Dados,”
- [13] M. Abdallah e O. Elkeelany, “A Survey on Data Acquisition systems DAQ,” 2009.
- [14] M. Schilling, “Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity,” *Academy of Management Review*, vol. 25, abr. de 2000. DOI: 10.5465/AMR.2000.3312918.
- [15] N. Rodrigues, E. Oliveira e P. Leitão, “Decentralized and on-the-fly agent-based service reconfiguration in manufacturing systems,” *Computers in Industry*, vol. 101, pp. 81–90, out. de 2018, ISSN: 01663615. DOI: 10.1016/j.compind.2018.06.003.
- [16] A. B. Bondi, “Characteristics of Scalability and Their Impact on Performance,” 2000. URL: <http://www.whatis.com/scalabil.htm..>
- [17] A. J. N. Jr, P. F. Rosa, I. O. Porto e M. dos Santos Soares, “Padrões e diretrizes arquiteturais para escalabilidade,” *X Encontro Anual de Computação - EnAComp*, 2013.
- [18] P. C. TEBALDI, *Conheça os principais protocolos de rede e seus usos*, <https://www.opservices.com.br/protocolos-de-rede/>, jan. de 2019.
- [19] J. Postel, *User Datagram Protocol*, <https://www.rfc-editor.org/rfc/rfc768>, Fevereiro de 2023.

- [20] C. Y. Chong e S. P. Kumar, “Sensor networks: Evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, pp. 1247–1256, 8 ago. de 2003, ISSN: 00189219. DOI: 10.1109/JPROC.2003.814918.
- [21] J. Zheng e A. Jamalipour, *Wireless sensor networks : a networking perspective*. IEEE, 2009, p. 489, ISBN: 9780470167632.
- [22] On-surf, *Projeto ON-SURF Mobilizar Competências Tecnológicas em Engenharia de Superfícies E2-2-1*, jun. de 2020.
- [23] On-surf, *Projeto ON-SURF Mobilizar Competências Tecnológicas em Engenharia de Superfícies E2-2-2*, setembro de 2020.

Apêndice A

Protocolo Não Orientado À Conexão Para Rede De Sensores Distribuídos

O protocolo apresentado foi elaborado durante o desenvolvimento da dissertação de mestrado, que tem como propósito criar sistemas de aquisição de dados configuráveis e com capacidade de expansão. O objetivo principal é possibilitar a escalabilidade de um grande número de sensores sem fio em uma rede que seja modular e adaptável.

O protocolo desenvolvido proporciona uma estrutura organizada e padronizada para os dados, o que resulta em maior escalabilidade e redução das limitações do sistema. Ele foi concebido para lidar com a complexidade do aumento do número de sensores e garantir que o sistema possa ser facilmente adaptado e expandido conforme necessário.

Portanto, o protocolo desempenha um papel crucial na construção de um sistema de aquisição de dados que seja altamente escalável, configurável e capaz de atender às demandas crescentes de um ambiente em constante evolução.

Protocolo Não Orientado À Conexão Para Rede De Sensores Distribuídos

Autores: Isaac Vinicius Fontes Sampaio & José Augusto Almeida Pinheiro De Carvalho

Conteúdo

Introdução	2
Abreviações	2
Contexto	2
Funções	2
Estrutura.....	3
Tipo de pedido.....	3
Equipamento ID.....	3
Sensor ID	4
Variável	5
Nº De Bytes Que Segue	5
Dados.....	6
Utilização	6
Adaptações.....	7

Introdução

O presente protocolo de mensagem foi especificado para correr sobre o protocolo de comunicação UDP. Está orientada para a interligação de uma rede de sensores de um sistema de aquisição de dados para monitorização e apoio a controlo de qualidade em sistema de produção industrial. O desenvolvimento desde protocolo tem como objetivo garantir a escalabilidade da rede de sensores, fazendo com que a rede se torne configurável e adaptável para todos os setores da indústria.

Abreviações

TCP - Transport Control Protocol

UDP - User Datagram Protocol

MQTT - Message Queuing Telemetry Transport

IP - Internet Protocol

Contexto

O presente trabalho foi escrito juntamente com o desenvolvimento da tese de mestrado, que visa desenvolver sistemas de aquisição de dados escaláveis e configuráveis, onde temos o objetivo de escalabilidade do maior número de sensores sem fio em uma rede que possa ser modular e adaptável, o protocolo garante essa organização e padronização dos dados aumentando sua escalabilidade e reduzindo suas limitações.

Funções

Herda do protocolo UDP Multicast a importante característica de não ser orientada à conexão e permitir a comunicação em um grupo de módulos de rede sem a necessidade de conhecimento dos constituintes da rede. Recorre ao modelo interação *Publisher Subscriber* sem necessidade de broker como o utilizado em outros protocolos como por exemplo de MQTT. Desta forma é reduzida a estrutura física da rede ao mesmo tempo que é facilitada a escalabilidade da rede.

O protocolo é um conjunto de regras de sintaxe e semântica com o objetivo de permitir a comunicação entre dispositivos, o atual protocolo visa identificar a origem do dado para garantir uma escalabilidade e adaptação do sistema para diversas aplicações para isso dividimos o cabeçalho da mensagem em 6 partes com tamanho diferentes e adaptáveis de acordo com utilização, partes essas descritas a seguir.

Estrutura

O presente protocolo possui dois tipos de estrutura, a primeira estrutura é mais completa e a estrutura de publicação, a segunda estrutura é a estrutura de solicitação que possui menos dados dentro do seu conjunto. Dentro das duas estruturas possuímos alguns campos em comum como: Tipo de pedido, Equipamento ID, Sensor ID e Variável. Para além destes dados a primeira estrutura, a de publicação, possui mais dois campos são eles o Nº De Bytes Que Segue e Dados. Segue o exemplo a estrutura completa na Tabela 1 - estrutura genérica. A seguir os detalhes de campo.

TIPO DE PEDIDO	EQUIPAMENTO ID	SENSOR ID	VARIAVEL	Nº DE BITS QUE SEGUE	DADOS
QQ	YYYY	ZZZZ	WW	XX	????
1 Byte	2 Bytes	2 Bytes	1 Byte	1 Bytes	Dados

Tabela 1 - estrutura genérica.

Tipo de pedido

Campo inicial de tamanho de um byte identificando o tipo de mensagem. O tipo de pedido é dividido em dois estados, onde um informará que é uma publicação periódica do nó de comunicações do sensor onde dependerá do ciclo pré-programado pela montagem do circuito e logica de funcionamento daquele sensor.

O segundo tipo de pedido é uma solicitação de publicação de variáveis de um determinado sensor. Este processo permite comunicação com o nó dentro da rede, e sincronizar com o estado das variáveis que subscreve na rede. Por exemplo: isto é utilizado quando o ciclo de publicação de determinado sensor tem um período muito longo, assim evitando que o usuário fique sem resposta por muito tempo.

01	Publicação padrão do sensor
02	Solicitação de publicação de sensor específico

Tabela 2 - tipo do pedido

Equipamento ID

O Campo Equipamento ID é composto por dois bytes, o primeiro identifica o tipo de equipamento. No segundo byte identifica o número do equipamento. Esta divisão facilita a diferenciação de equipamentos por tipo (prensas, tornos, robôs, tapetes de transferência etc.). O que permite uma fácil filtragem das mensagens publicadas. O segundo campo permite a distinção de equipamento (ex. se existir mais que uma prensa), ele filtrar a mensagem para um equipamento específico.

Separámos algumas das mais utilizadas para predefinição do protocolo, e caso alguma não esteja listada nesse protocolo será possível a introdução dela posteriormente.

01	Prensa de estampagem única
02	Prensa de estampagem continua
03	Serra
04	Motores elétricos
05	Motores de combustão
06	Ferramenta de torneamento
07	Esteiras
08	Estufas
09	Fornos
10	Tubulações
NN	Adição de novos equipamentos

Tabela 3 – Equipamento ID

Sensor ID

O Campo sensor ID e composto por dois bytes, o primeiro identifica o tipo de sensor. No segundo byte identifica a posição no equipamento. Esta divisão facilita a diferenciação de sensores por tipo (indutivos, capacitivos, fotoelétricos, fibras ópticas, lasers, ultrassônicos, magnéticos, transdutores lineares, pressão etc.) Exemplo Tabela 4 - Tipo de Sensor. O que permite uma fácil filtragem das mensagens publicadas. O segundo campo permite a distinção da posição do sensor no equipamento (ex. se existir mais que um sensor no mesmo equipamento), ele filtrar a mensagem para um sensor específico.

01	Indutivo
02	Capacitivo
03	Fotoelétrico
04	Magnéticos
NN	N Tipos

Tabela 4 - Tipo de Sensor

Exemplo: Uma prensa que possui 6 sensores do mesmo tipo em diferentes pontos como podemos ver na Figura 1 - prensa com 6 sensores distribuídos, os sensores são representados pelos S1, S2, S3, S4, S5 e S6, cada sensor deste deverá ter sua própria definição para ser enviada no protocolo, como mostrado na Tabela 5 - Exemplo Lista Sensor ID.

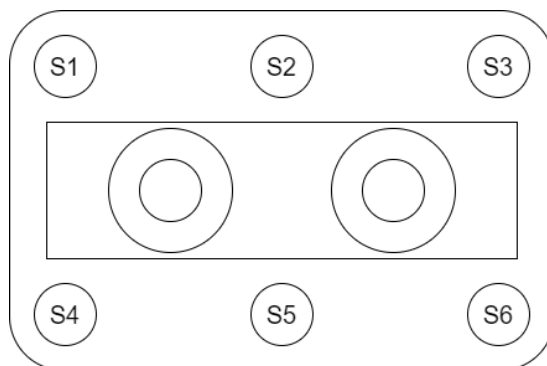


Figura 1 - prensa com 6 sensores distribuídos

01	Sensor 1 (S1)
02	Sensor 2 (S2)
03	Sensor 3 (S3)
04	Sensor 4 (S4)
05	Sensor 5 (S5)
06	Sensor 6 (S6)

Tabela 5 - Exemplo Lista Sensor ID

Variável

O Campo Variável é composto por um byte, que identifica o tipo de variável a ser analisada. Isto facilita a identificação da variável, tais como pressão, torque, temperatura, vibração, ruído entre outros fenômenos físicos que podemos analisar dentro do ambiente industrial e cotidiano. Para a indústria possuímos N tipos de variáveis, vendo isso separamos algumas das mais utilizadas para predefinição no protocolo, e caso alguma não esteja listada nesse protocolo será possível a introdução dela posteriormente, como podemos ver na Tabela 6 - Lista de Variáveis.

01	Medição de deformação, tensão, força e movimento
02	Temperatura
03	Pressão atmosférica
04	Ruído
05	Luminosidade
06	Umidade
07	Vibração
08	Vazão
09	Densidade
10	Nível

Tabela 6 - Lista de Variáveis

Nº De Bytes Que Segue

O Campo Nº de bytes que segue é composto por 1 byte, que informa a quantidade de bytes a serem tratados e analisados após o mesmo, são os números de bytes que depois deste ponto devem ser analisados pelo leitor, pois conforme fazemos a leitura poderá estar vindo alguma informação não interessante fazendo com que os dados percam sentido, sendo assim saber a quantidade de dados a seguir que devem ser tratados e coletados é crucial para obtermos resultados conclusivos.

Exemplo: Enviando um dado de temperatura com uma escala de precisão de 2 dígitos depois do 0, para isso são necessários 2 bytes, logo na mensagem terá a descrição de 0002.

Exemplo: Enviando um dado de pressão, somente com dois dígitos decimais, logo para isso só é necessário 1 bytes, sendo assim na mensagem terá a descrição de 0001

01	1 Bytes
02	2 Bytes
03	3 Bytes
04	4 Bytes

Tabela 7 - Nº de Bytes

Caso haja necessidade de mais byte, seguir a lógica apresentada na Tabela 7 - Nº de Bytes, o número de bytes corresponde a quantidade número decimal convertido em hexadecimal.

Dados

O campo dado é composto pelos números de bytes definido no Nº De Bytes Que Segue, isto porque a quantidade de dados é definida pelo utilizador. Cada nó terá uma estrutura de cabeçalho predefinida pelo programador, essa estrutura não será alterada a menos que haja uma nova formatação e reconfiguração, o único dado que é variável dentro da mensagem são os dados, que poderá variar a cada publicação.

Utilização

Após definida todas as partes da mensagem deste protocolo, devesse unir todas as partes da mensagem em uma única, assim facilitando a transmissão, necessitando de apenas um envio para a transmissão de todo o protocolo.

Como podemos ver na estrutura genérica de publicação deste protocolo na Tabela 8 e na estrutura genérica de solicitação de publicação na Tabela 10

TIPO DE PEDIDO	EQUIPAMENTO ID	SENSOR ID	VARIABEL	Nº DE BITS QUE SEGUE	DADOS
01	YYYY	ZZZZ	WW	XX	????
1 Byte	2 Bytes	2 Bytes	1 Byte	1 Bytes	Dados

Tabela 8 - estrutura genérica do protocolo de publicação

Exemplo: Para publicação (01) de uma prensa de estampagem continua (02) (prensa número 03) do sensor indutivo (01) posicionado no S4 (04) para transmissão da variável de deformação (01) com 1 números de bytes que segue (01), seja na Tabela 9 .

Solução:

TIPO DE PEDIDO	EQUIPAMENTO ID	SENSOR ID	VARIABEL	Nº DE BITS QUE SEGUE	DADOS
01	0203	0104	01	01	A2

1 Byte

2 Bytes

2 Bytes

1 Byte

1 Byte

1 Byte

Tabela 9 - solução para publicação

Como será enviado pelo nó: 01020301040101A2

TIPO DE PEDIDO	EQUIPAMENTO ID	SENSOR ID	VARIAVEL
02	YYYY	ZZ	WW
1 Byte	2 Bytes	2 Bytes	1 Byte

Tabela 10 - Estrutura genérica do protocolo solicitação de publicação

Exemplo: Para solicitação de publicação (02) de um forno (09) número 12 (12) do sensor indutivo (01) posicionado no S2 (02) para transmissão da variável de Temperatura (02), veja na Tabela 11

Solução:

TIPO DE PEDIDO	EQUIPAMENTO ID	SENSOR ID	VARIAVEL
02	0912	0102	02
1 Byte	2 Bytes	2 Bytes	1 Byte

Tabela 11 - solução para solicitação de publicação

Como será enviado pelo nó: 020912010202

Adaptações

As adaptações devem ser feitas de forma lógica e escalável para cada particularidade encontrada sem alterar as funções pré-existentes, sendo assim elas devem ser adicionadas ao bloco de dados que deve ser desenvolvido pelo usuário.

Apêndice B

Códigos desenvolvidos

B.1 Publicação em C realizada pelo ESP32

O primeiro código fornecido é referente ao programa de publicação em UDP que utiliza o protocolo desenvolvido. Esse código é responsável por enviar os dados coletados para outros dispositivos por meio do protocolo UDP. O programa de publicação em UDP desempenha um papel fundamental na transmissão eficiente e confiável dos dados, permitindo que sejam recebidos e processados por outros sistemas ou dispositivos conectados na rede. Através desse código, é possível estabelecer a comunicação e compartilhar as informações coletadas pelo sistema de aquisição de dados, possibilitando sua utilização em diversas aplicações e contribuindo para o monitoramento e controle das grandezas em tempo real.

```
/*
 * testar a função do que trabalha com o tempo
 * testar as portas de UDP
 */
#include "WiFi.h"
#include "AsyncUDP.h"
//#include <stdio.h>
#include <WiFiUdp.h>

const char * ssid = "ISAAC";
const char * password = "12345678";

unsigned int multicastPort = 1234; // local
port to listen on
//IPAddress multicastIP(255,255,255,255);

IPAddress udpAddress(255,255,255,255);

const int udpPort = 1234;

int X = 0;
//AsyncUDP udp;
WiFiUDP udp;
WiFiUDP Multicast;

unsigned long previousMillis = 0;
const long interval = 100; // interval at which
to blink (milliseconds)
```

```

boolean connected = false;

void setup() {
  Serial.begin(115200);

  // delete old config
  WiFi.disconnect(true);
  delay(1000);

  /*
   * SYSTEM_EVENT_STA_CONNECTED: o ESP32 está
conectado em modo estação a um ponto de
acesso/hotspot (seu roteador);
   * SYSTEM_EVENT_STA_DISCONNECTED: a estação
ESP32 desconectada do ponto de acesso.
   */
  WiFi.onEvent(WiFiStationConnected,
SYSTEM_EVENT_STA_CONNECTED);
  WiFi.onEvent(WiFiGotIP,
SYSTEM_EVENT_STA_GOT_IP);
  WiFi.onEvent(WiFiStationDisconnected,
SYSTEM_EVENT_STA_DISCONNECTED);

  /* Remove WiFi event
Serial.print("WiFi Event ID: ");
Serial.println(eventID);
WiFi.removeEvent(eventID);*/

  WiFi.begin(ssid, password);

  Serial.println();

```

```

    Serial.println();
    Serial.println("Aguarde pelo WiFi... ");
}

void WiFiStationConnected(WiFiEvent_t event,
WiFiEventInfo_t info) {
    Serial.println("Conectado ao AP com sucesso!");
}

/*
 * Essa função simplesmente imprime o endereço
IP no Serial Monitor.
 */
void WiFiGotIP(WiFiEvent_t event,
WiFiEventInfo_t info) {
    Serial.println("WiFi conectado");
    Serial.print("IP : ");
    Serial.println(WiFi.localIP());
    connected = true;
}

/*
 * Essa função imprime uma mensagem indicando
que a conexão foi perdida e tenta reconectar:
 */
void WiFiStationDisconnected(WiFiEvent_t event,
WiFiEventInfo_t info) {
    Serial.println("Desconectado do ponto de
acesso WiFi");
    Serial.print("WiFi perdeu a conexão. Motivo:

```

```
");  
    Serial.println(info.disconnected.reason);  
    connected = false;  
    Serial.println("Tentando reconectar");  
    WiFi.begin(ssid, password);  
}
```

```
void loop()  
{  
    //delay(1000);  
    //Send broadcast on port 1234  
    unsigned long currentMillis = millis();  
    if(connected){  
        if (currentMillis - previousMillis >=  
interval) {  
            previousMillis = currentMillis;  
            //Send a packet  
  
            String protocol = "010204012";  
            String hexData = String(X, DEC);  
            //String hexData = String(X, HEX);  
            protocol = String(protocol + " ");  
            protocol = String(protocol + hexData);  
            Serial.println(protocol);  
            udp.beginPacket(udpAddress, udpPort);  
            udp.print(protocol);  
            //Serial.println(" ");  
            //Serial.printf("enviados %d", X);
```

```
        udp.endPacket();
        //udp.broadcastTo("alguem ai? ",1234);
//enviando o dado para o Broadcast via UDP
        //Serial.println("  alguem ai?");

        X = analogRead("PORTA do ESP32");

    }
}

}
```

```
/*
WIFI_REASON_UNSPECIFIED           = 1,
WIFI_REASON_AUTH_EXPIRE          = 2,
WIFI_REASON_AUTH_LEAVE           = 3,
WIFI_REASON_ASSOC_EXPIRE         = 4,
WIFI_REASON_ASSOC_TOOMANY        = 5,
WIFI_REASON_NOT_AUTHED           = 6,
WIFI_REASON_NOT_ASSOCED          = 7,
WIFI_REASON_ASSOC_LEAVE          = 8,
WIFI_REASON_ASSOC_NOT_AUTHED     = 9,
WIFI_REASON_DISASSOC_PWRCAP_BAD  = 10,
WIFI_REASON_DISASSOC_SUPCHAN_BAD = 11,
WIFI_REASON_IE_INVALID           = 13,
WIFI_REASON_MIC_FAILURE          = 14,
WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT = 15,
WIFI_REASON_GROUP_KEY_UPDATE_TIMEOUT = 16,
WIFI_REASON_IE_IN_4WAY_DIFFERS  = 17,
WIFI_REASON_GROUP_CIPHER_INVALID = 18,
```

WIFI_REASON_PAIRWISE_CIPHER_INVALID	= 19,
WIFI_REASON_AKMP_INVALID	= 20,
WIFI_REASON_UNSUPP_RSN_IE_VERSION	= 21,
WIFI_REASON_INVALID_RSN_IE_CAP	= 22,
WIFI_REASON_802_1X_AUTH_FAILED	= 23,
WIFI_REASON_CIPHER_SUITE_REJECTED	= 24,
WIFI_REASON_BEACON_TIMEOUT	= 200,
WIFI_REASON_NO_AP_FOUND	= 201,
WIFI_REASON_AUTH_FAIL	= 202,
WIFI_REASON_ASSOC_FAIL	= 203,
WIFI_REASON_HANDSHAKE_TIMEOUT	= 204,

* /

B.2 Leitura em C realizada pelo ESP32

O segundo código fornecido refere-se ao programa de leitura desenvolvido em linguagem C, projetado especificamente para o microcontrolador ESP32. Este código tem como finalidade principal a interpretação e processamento dos dados coletados pelo sistema de aquisição de dados, possibilitando a sua utilização em diversas aplicações. A implementação do código em linguagem C permite a otimização do desempenho e a maximização da eficiência do ESP32 como leitor de dados. Dessa forma, o programa de leitura em C é uma peça fundamental para o funcionamento adequado e eficaz do sistema de aquisição de dados, garantindo uma coleta precisa e confiável das grandezas monitoradas.

```
#include "WiFi.h"
#include "AsyncUDP.h"
#include <WiFiUdp.h>

const char * ssid = "ISAAC";
const char * password = "12345678";
int X = 1;

AsyncUDP udp;

unsigned long previousMillis = 0;
const long interval = 1000; // interval at
which to blink (milliseconds)
const long intervalcon = 1000;
unsigned long currentMillis = millis();

boolean connected = false;

void setup()
{
  Serial.begin(115200);

  // delete old config
  WiFi.disconnect(true);
  delay(1000);

  /*
   * SYSTEM_EVENT_STA_CONNECTED: o ESP32 está
conectado em modo estação a um ponto de
acesso/hotspot (seu roteador);
   * SYSTEM_EVENT_STA_DISCONNECTED: a estação
```

ESP32 desconectada do ponto de acesso.

*/

```
WiFi.onEvent(WiFiStationConnected,  
SYSTEM_EVENT_STA_CONNECTED);  
WiFi.onEvent(WiFiGotIP,  
SYSTEM_EVENT_STA_GOT_IP);  
WiFi.onEvent(WiFiStationDisconnected,  
SYSTEM_EVENT_STA_DISCONNECTED);
```

```
/* Remove WiFi event
```

```
Serial.print("WiFi Event ID: ");
```

```
Serial.println(eventID);
```

```
WiFi.removeEvent(eventID);*/
```

```
WiFi.begin(ssid, password);
```

```
Serial.println();
```

```
Serial.println();
```

```
Serial.println("Aguarde pelo WiFi... ");
```

```
if(udp.listen(1234)) {
```

```
    Serial.print("UDP Listening on IP: ");
```

```
    Serial.println(WiFi.localIP());
```

```
    udp.onPacket([](AsyncUDPPacket packet)
```

```
{
```

```
        Serial.print("UDP Packet Type: ");
```

```
        Serial.print(packet.isBroadcast()?
```

```
"Broadcast":packet.isMulticast()?
```

```
"Multicast":"Unicast");
```

```
        Serial.print(", From: ");
```

```
        Serial.print(packet.remoteIP());
```

```

        Serial.print(":");
        Serial.print(packet.remotePort());
        Serial.print(", To: ");
        Serial.print(packet.localIP());
        Serial.print(":");
        Serial.print(packet.localPort());
        Serial.print(", Length: ");
        Serial.print(packet.length());
        Serial.print(", Data: ");
        Serial.write(packet.data(), packet.
length());

        Serial.print(" recebidos ");
        Serial.println();
    });
}

```

```

}

```

```

void WiFiStationConnected(WiFiEvent_t event,
WiFiEventInfo_t info){
    Serial.println("Conectado ao AP com sucesso!");
}

```

```

/*

```

```

 * Essa função simplesmente imprime o endereço
IP no Serial Monitor.

```

```

*/

```

```

void WiFiGotIP(WiFiEvent_t event,
WiFiEventInfo_t info){
    Serial.println("WiFi conectado");
    Serial.print("IP : ");
}

```

```

Serial.println(WiFi.localIP());
connected = true;
}

/*
 * Essa função imprime uma mensagem indicando
 que a conexão foi perdida e tenta reconectar:
 */
void WiFiStationDisconnected(WiFiEvent_t event,
WiFiEventInfo_t info) {
    Serial.println("Desconectado do ponto de
acesso WiFi");
    Serial.print("WiFi perdeu a conexão. Motivo:
");
    Serial.println(info.disconnected.reason);
    connected = false;
    Serial.println("Tentando reconectar");
    WiFi.begin(ssid, password);
}

void loop()
{
    delay(1000);
    //Send multicast
    udp.print(X);
    udp.println("  ");
}

```

/*

WIFI_REASON_UNSPECIFIED	= 1,
WIFI_REASON_AUTH_EXPIRE	= 2,
WIFI_REASON_AUTH_LEAVE	= 3,
WIFI_REASON_ASSOC_EXPIRE	= 4,
WIFI_REASON_ASSOC_TOOMANY	= 5,
WIFI_REASON_NOT_AUTHED	= 6,
WIFI_REASON_NOT_ASSOCED	= 7,
WIFI_REASON_ASSOC_LEAVE	= 8,
WIFI_REASON_ASSOC_NOT_AUTHED	= 9,
WIFI_REASON_DISASSOC_PWRCAP_BAD	= 10,
WIFI_REASON_DISASSOC_SUPCHAN_BAD	= 11,
WIFI_REASON_IE_INVALID	= 13,
WIFI_REASON_MIC_FAILURE	= 14,
WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT	= 15,
WIFI_REASON_GROUP_KEY_UPDATE_TIMEOUT	= 16,
WIFI_REASON_IE_IN_4WAY_DIFFERS	= 17,
WIFI_REASON_GROUP_CIPHER_INVALID	= 18,
WIFI_REASON_PAIRWISE_CIPHER_INVALID	= 19,
WIFI_REASON_AKMP_INVALID	= 20,
WIFI_REASON_UNSUPP_RSN_IE_VERSION	= 21,
WIFI_REASON_INVALID_RSN_IE_CAP	= 22,
WIFI_REASON_802_1X_AUTH_FAILED	= 23,
WIFI_REASON_CIPHER_SUITE_REJECTED	= 24,
WIFI_REASON_BEACON_TIMEOUT	= 200,
WIFI_REASON_NO_AP_FOUND	= 201,
WIFI_REASON_AUTH_FAIL	= 202,
WIFI_REASON_ASSOC_FAIL	= 203,
WIFI_REASON_HANDSHAKE_TIMEOUT	= 204,

*/

B.3 Leitor em python realizado pelo computador

O terceiro código é composto por um leitor desenvolvido em Python que desempenha um papel crucial na interpretação do protocolo desenvolvido. Além disso, esse código é responsável por enviar os dados coletados para um banco de dados armazenado na nuvem. Através desse leitor em Python, os dados são processados e formatados de acordo com as especificações do protocolo, permitindo a sua correta interpretação e armazenamento na nuvem. Essa integração com o banco de dados na nuvem possibilita o acesso e a análise dos dados coletados de forma remota, oferecendo uma solução flexível e escalável para a gestão dos dados provenientes do sistema de aquisição. Com o auxílio desse código, é possível armazenar, consultar e compartilhar os dados de forma eficiente, facilitando a tomada de decisões e o monitoramento das grandezas em tempo real.

```

import socket

# Configurações do soquete
ip = "0.0.0.0"      # Endereço IP para escutar todas as interfaces
porta = 1234        # Porta para escutar

# Cria o soquete UDP
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Vincula o soquete ao endereço e porta de escuta
sock.bind((ip, porta))

# Recebe o pacote UDP
data, endereco = sock.recvfrom(1024)

# simulação do protocolo com variaveis criadas
#for i in range(10, 99):

    # data valor enviado via multicast
#data = "010301020301"+str(i)

#frAGMENTAÇÃO da string

tipo_de Equipamento      = data[:2]
N_do Equipamento = data[2:4]
tipo_de sensor = data[4:6]
posicao = data[6:8]
variavel = data[8:10]

## identificação do bit que segue
bit = data[10:12]
## converte hex para dec
bit_int = int(bit,16)
#print(bit_int)
segue = (bit_int*2) + 12
#print(segue)
dado = data[12:segue]
## converte hex para dec

dadod = int(dado,16)
#print(dadod)

#funções case tipo de equipamento
def func_case(tipo_de Equipamento):
    if tipo_de Equipamento == '01':
        texto = "Prensa de estampagem única; "
    elif tipo_de Equipamento == '02':
        texto = "Prensa de estampagem continua; "
    elif tipo_de Equipamento == '03':
        texto = "N tipos; "
    else:
        print("mensagem fora de padrão")
    return texto

```

```
func_case(tipo_de Equipamento)
descricao1 = func_case(tipo_de Equipamento)
```

```
# função case N equipamento
```

```
def func_case(N_do Equipamento):
    if N_do Equipamento == '01':
        texto1 = "EQ 1; "
    elif N_do Equipamento == '02':
        texto1 = "EQ 2; "
    elif N_do Equipamento == '03':
        texto1 = "EQ 3; "
    else:
        print("mensagem fora de padrão")
    return texto1
```

```
func_case(N_do Equipamento)
descricao2 = func_case(N_do Equipamento)
```

```
#case tipo de sensor
```

```
def func_case(tipo_de sensor):
    if tipo_de sensor == '01':
        texto2 = "indutivo; "
    elif tipo_de sensor == '02':
        texto2 = "capacitivo; "
    elif tipo_de sensor == '03':
        texto2 = "N tipos; "
    else:
        print("mensagem fora de padrão")
    return texto2
```

```
func_case(tipo_de Equipamento)
descricao3 = func_case(tipo_de Equipamento)
```

```
# case posição
```

```
def func_case(posicao):
    if posicao == '01':
        texto3 = "S1; "
    elif posicao == '02':
        texto3 = "S2; "
    elif posicao == '03':
        texto3 = "S*N; "
    else:
        print("mensagem fora de padrão")
    return texto3
```

```
func_case(posicao)
```

```
descricao4 = func_case(posicao)
```

```
# case variavel
```

```
def func_case(variavel):
```

```

    if variavel== '01':
        texto4 = "deformação, tensão, força; "
    elif variavel == '02':
        texto4 = "temperatura; "
    elif variavel == '03':
        texto4 = "N variaveis; "
    else:
        print("mensagem fora de padrão")
    return texto4

func_case(variavel)
descricao5 = func_case(variavel)

textoc = descricao1+descricao2+descricao3+descricao4+descricao5 +
str(dadod)+";"+ "!!"
# print(textoc)

#bibliotecas MQTT
import time
import paho.mqtt.client as paho
from paho import mqtt

#comunicação MQTT

def on_connect(client, userdata, flags, rc, properties=None):
    print("CONNACK received with code %s." % rc)

def on_publish(client, userdata, mid, properties=None):
    print("mid: " + str(mid))

def on_subscribe(client, userdata, mid, granted_qos, properties=None):
    print("Subscribed: " + str(mid) + " " + str(granted_qos))

def on_message(client, userdata, msg):
    print(msg.topic + " " + str(msg.qos) + " " + str(msg.payload))

client = paho.Client(client_id="", userdata=None, protocol=paho.MQTTv5)
client.on_connect = on_connect
client.tls_set(tls_version=mqtt.client.ssl.PROTOCOL_TLS)
client.username_pw_set("ISAAC", "ISAACTESE2023")
client.connect("db0f5149b32442f59b7ee9687309f054.s2.eu.hivemq.cloud", 8883)

client.subscribe("encyclopedia/#", qos=1)
client.publish("encyclopedia/data", payload= textoc, qos=1)

client.loop_forever()

```