



**DESENVOLVIMENTO DE UM SISTEMA IOT PARA CONTROLE E
MONITORAMENTO DE AMBIENTES USANDO ESP-NOW**

GUILHERME DE OLIVEIRA KFOURI

**TRABALHO DE CONCLUSÃO DE CURSO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**DEVELOPMENT OF AN IOT SYSTEM FOR CONTROL AND
MONITORING OF ENVIRONMENTS USING ESP-NOW**

**DESENVOLVIMENTO DE UM SISTEMA IOT PARA CONTROLE E
MONITORAMENTO DE AMBIENTES USANDO ESP-NOW**

GUILHERME DE OLIVEIRA KFOURI

ORIENTADOR: DANIEL CHAVES CAFÉ, DR.

**TRABALHO DE CONCLUSÃO DE CURSO EM
ENGENHARIA ELÉTRICA**

PUBLICAÇÃO: XXX

BRASÍLIA/DF: JULHO - 2023

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**DESENVOLVIMENTO DE UM SISTEMA IOT PARA CONTROLE E
MONITORAMENTO DE AMBIENTES USANDO ESP-NOW**

GUILHERME DE OLIVEIRA KFOURI

**TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA
ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL.**

APROVADA POR:

**Prof. Dr. Daniel Chaves Café – ENE/Universidade de Brasília
Orientador**

**Prof. Dr. José Edil Guimarães de Medeiros – ENE/Universidade de Brasília
Membro Interno**

**Prof. Dr. Renato Coral – ENE/Universidade de Brasília
Membro Interno**

BRASÍLIA, 21 DE JULHO DE 2023.

FICHA CATALOGRÁFICA

Desenvolvimento de um Sistema IoT para Controle e Monitoramento de Ambientes usando ESP-NOW [Distrito Federal] 2023.

ix, 50p., 210 x 297 mm (ENE/FT/UnB, Bacharel, Engenharia Elétrica, 2023).

Trabalho de Conclusão de Curso – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Sistema IoT

2. Monitoramento

3. ESP32

4. ESP-NOW

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

(2023). Desenvolvimento de um Sistema IoT para Controle e Monitoramento de Ambientes usando ESP-NOW . Trabalho de Conclusão de Curso em Engenharia Elétrica, Publicação XXX, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 50p.

CESSÃO DE DIREITOS

AUTOR: Guilherme de Oliveira Kfourri

TÍTULO: Desenvolvimento de um Sistema IoT para Controle e Monitoramento de Ambientes usando ESP-NOW .

GRAU: Bacharel ANO: 2023

É concedida à Universidade de Brasília permissão para reproduzir cópias desta trabalho de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa trabalho de conclusão de curso pode ser reproduzida sem autorização por escrito do autor.

Guilherme de Oliveira Kfourri

Departamento de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Dedico este trabalho à minha família e amigos, que sempre estiveram ao meu lado, compreendendo minha ausência em momentos importantes e apoiando meus sonhos e objetivos.

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos aos meus pais e minha família por terem proporcionado toda a estrutura e condições necessárias para a obtenção do grau de Bacharel em Engenharia Elétrica. À minha mãe, Viviane de Oliveira, pelo carinho e cuidado que sempre teve comigo. Ao meu pai, Washington Kfourri, por sempre me aconselhar e ser um exemplo para mim. Ao meu irmão, Tiago Kfourri, pelo companheirismo na vivência. Às minhas irmãs, Anna e Laís, e minha madrastra, Vanessa Dias, por me inspirarem a querer ser um melhor exemplo de pessoa.

À minha namorada, Luma Guadanhim, por todo amor, suporte e companheirismo durante a jornada. Sem seu apoio isso não seria possível.

Ao meu orientador, Dr. Daniel Chaves Café, que inicialmente me inspirou na matéria de Sistemas Microcontrolados e que tive o prazer de ser orientado para este trabalho. Levarei comigo sua paciência, dedicação e amor ao que faz.

Aos meus companheiros de curso, carinhosamente apelidados de Reino dos Magos, os quais passaram comigo os melhores e mais difíceis momentos da Graduação, em especial ao Alberto Junqueira, Fausto Brenner, João Pedro Silveira, Johannes Algera, Matheus Costa, Lallamand Canedo e Lucas Ferreira.

Aos meus amigos do ensino médio, carinhosamente apelidados de Carro do Leite, os quais sempre me inspiram na área acadêmica e a ser uma pessoa melhor e mais simples, em especial à Anne Caroline Pereira, Cecília Cipriano, Ciro Frazão, Gabriel Furlani, Giordano Suffert Monteiro, João Pedro Haltrat, Luís Clayton Mourão, Luísa Miranda, Luiz Eduardo Viana e Letícia de Amorim.

Aos meus amigos Fabrício Barcelos, João Francisco Londe, Leonardo Rocha, Marcos Blandim, Pedro Blandim e Samuel Figueiredo pela parceria e confiança de sempre.

À Universidade de Brasília, pela estrutura e possibilidade da graduação.

Serei eternamente grato à todos!

RESUMO

Título: Desenvolvimento de um Sistema IoT para Controle e Monitoramento de Ambientes usando ESP-NOW

Autor: Guilherme de Oliveira Kfourri

Orientador: Daniel Chaves Café, Dr.

Programa de Pós-Graduação em Engenharia Elétrica

Brasília, 21 de julho de 2023

Este trabalho apresenta o desenvolvimento e a implementação de um sistema de Internet das Coisas (Internet of Things (IoT)) para o monitoramento de portas e o controle remoto de um aparelho de ar-condicionado. O sistema foi projetado para proporcionar maior eficiência e comodidade em ambientes internos, permitindo aos usuários monitorar o estado de abertura e fechamento de portas e controlar o funcionamento do ar-condicionado através de uma interface web.

Na fase de implementação, foram utilizados dois tipos de sensores: o sensor de porta baseado no módulo Reed Switch e o sensor infravermelho para controlar o ar-condicionado. Ambos os sensores foram integrados ao microcontrolador ESP32 e configurados para se comunicarem com um gateway e um servidor web.

O teste de integração foi conduzido para verificar a correta comunicação entre os componentes do sistema IoT. Os resultados demonstraram a eficácia da arquitetura proposta, com o sensor de porta registrando de forma precisa o estado da porta no servidor web e o sensor infravermelho sendo capaz de ligar e desligar o ar-condicionado remotamente.

Apesar dos avanços alcançados, algumas limitações foram identificadas, como a incompatibilidade do dispositivo infravermelho com um modelo específico de ar-condicionado. Além disso, não foram implementadas funcionalidades de economia de bateria nos microcontroladores e o gerenciamento de computadores da Unidade de Laboratórios de Ensino de Graduação (ULEG/FT). Essas áreas são apontadas como possíveis direções para futuros trabalhos. No entanto, o sistema de IoT desenvolvido demonstrou-se funcional e adequado, abrindo caminho para aplicações práticas em ambientes internos para melhorar o monitoramento e o controle de dispositivos.

Palavras-chave: Sistema IoT, Monitoramento, ESP32, ESP-NOW.

ABSTRACT

Title: Development of an IoT System for Control and Monitoring of Environments using ESP-NOW

Author: Guilherme de Oliveira Kfourri

Supervisor: Daniel Chaves Café, Dr.

Graduate Program in Electrical Engineering

Brasília, July 21th, 2023

This work presents the development and implementation of an Internet of Things (IoT) system for door monitoring and remote control of an air conditioning unit. The system is designed to provide enhanced efficiency and convenience in indoor environments, allowing users to monitor the status of doors and control the operation of the air conditioning through a web interface.

During the implementation phase, two types of sensors were employed: a door sensor based on the Reed Switch module and an infrared sensor to control the air conditioning unit. Both sensors were integrated with the ESP32 microcontroller and configured to communicate with a gateway and a web server.

The integration test was conducted to verify the proper communication between the components of the IoT system. The results demonstrated the effectiveness of the proposed architecture, with the door sensor accurately registering the door status on the web server and the infrared sensor successfully turning the air conditioning unit on and off remotely.

Despite the achieved advancements, some limitations were identified, such as the incompatibility of the infrared sensor with a specific model of air conditioning unit. Additionally, battery-saving features were not implemented on the microcontrollers, and the management of ULEG/FT computers was not addressed. These areas are identified as potential directions for future work. Nevertheless, the developed IoT system proved to be functional and appropriate, paving the way for practical applications in indoor environments to enhance device monitoring and control.

Keywords: IoT System, Monitoring, ESP32, ESP-NOW.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CENÁRIO ATUAL	1
1.2	INTERNET OF THINGS	2
1.3	DESAFIOS DA GESTÃO UNIFICADA DE AMBIENTES LABORATORIAIS	3
1.4	GERENCIAMENTO, MONITORAMENTO E IMPLANTAÇÃO DE CONFIGURAÇÃO DE SOFTWARE	4
1.5	ARQUITETURA DA SOLUÇÃO	5
1.6	OBJETIVOS DO TRABALHO	6
1.7	ESTRUTURA DO TEXTO	7
2	REVISÃO BIBLIOGRÁFICA	8
2.1	CONTROLE DE PORTAS UTILIZANDO IOT	8
2.2	CONTROLE DE CONDICIONADORES DE AR UTILIZANDO IOT	10
3	PROJETO	11
3.1	ESCOLHA DO SENSOR DE MONITORAMENTO DA PORTA	11
3.2	COMPARAÇÃO DE MICROCONTROLADORES	12
3.2.1	PREÇO	13
3.2.2	TECNOLOGIA DE COMUNICAÇÃO	13
3.2.3	BATERIA E CONSUMO DE ENERGIA	14
3.2.4	HALL EFFECT SENSOR	16
3.3	GATEWAY	16
3.4	ESCOLHA DO MICROCONTROLADOR	17
3.5	COMPARAÇÃO DE SOFTWARES DE MONITORAMENTO/GERENCIAMENTO DE MÁQUINAS	18
4	METODOLOGIA	19
4.1	DESENVOLVIMENTO DE HARDWARE	19
4.1.1	DISPOSITIVO DE PORTA	19
4.1.2	DISPOSITIVO INFRAVERMELHO	23
4.1.3	GATEWAY	27
4.1.4	SERVIDOR WEB	28
4.2	DESENVOLVIMENTO DE FIRMWARE	28
4.2.1	DISPOSITIVO DE PORTA	28
4.2.2	DISPOSITIVO INFRAVERMELHO	29
4.2.3	GATEWAY	29

4.2.4	SERVIDOR WEB	31
4.2.5	REPOSITÓRIO DO FIRMWARE	34
4.3	TESTE DE INTEGRAÇÃO	34
4.3.1	COMPONENTES DO SISTEMA IoT	34
4.3.2	OBJETIVOS DO TESTE DE INTEGRAÇÃO	34
4.3.3	AÇÕES E EVENTOS SIMULADOS	34
4.3.4	DADOS UTILIZADOS NO TESTE	35
4.3.5	CONFIGURAÇÕES E CONEXÕES FÍSICAS	35
4.3.6	PROCEDIMENTOS ESPECÍFICOS SEGUIDOS.....	35
4.3.7	RESULTADOS ESPERADOS.....	36
4.3.8	MÉTRICAS UTILIZADAS PARA AVALIAR O SUCESSO.....	36
4.4	IMPLEMENTAÇÃO TESTE NA ULEG.....	36
4.4.1	CONFIGURAÇÃO DO AMBIENTE DE IMPLEMENTAÇÃO.....	37
4.4.2	INSTALAÇÃO DO DISPOSITIVO DE PORTA	37
4.4.3	INSTALAÇÃO DO DISPOSITIVO INFRAVERMELHO	37
4.4.4	CONFIGURAÇÃO DO GATEWAY	37
4.4.5	CONFIGURAÇÃO DO SERVIDOR WEB.....	38
5	RESULTADOS	39
5.1	RESULTADOS	39
5.1.1	RESULTADOS DO TESTE DE INTEGRAÇÃO.....	39
5.1.1.1	FLUXO DE COMUNICAÇÃO DO DISPOSITIVO DE PORTA	40
5.1.1.2	FLUXO DE COMUNICAÇÃO DO DISPOSITIVO INFRAVERMELHO .	40
5.1.2	RESULTADOS DA IMPLEMENTAÇÃO TESTE NA ULEG.....	41
5.1.2.1	FLUXO DE COMUNICAÇÃO DO DISPOSITIVO DE PORTA	41
5.1.2.2	FLUXO DE COMUNICAÇÃO DO DISPOSITIVO INFRAVERMELHO .	42
6	CONCLUSÃO.....	47
6.1	CONCLUSÃO	47
6.1.1	FUNCIONALIDADE LIMITADA DO DISPOSITIVO INFRAVERMELHO COM O AR-CONDICIONADO DA HITACHI	47
6.1.2	SUCESSO DO DISPOSITIVO DE PORTA.....	47
6.1.3	ARQUITETURA FUNCIONAL E ADEQUADA	47
6.1.4	CONCLUSÃO FINAL	47
	REFERENCES	48

1

INTRODUÇÃO

1.1 CENÁRIO ATUAL

Atualmente um dos grandes desafios de empresas e organizações é o controle e segurança patrimonial. Garantir a integridade de equipamentos, controlar entradas e saídas de pessoas, gerenciar o consumo energético de aparelhos, entre outros, são desafios que surgem com a necessidade de gerenciar um grande prédio.

A Universidade de Brasília (UnB) não foge dessa necessidade. Tendo em vista a grande extensão do Campus Darcy Ribeiro, a grande quantidade de prédios distintos dentro dele e a fácil transição e locomoção de pessoas por dentro da UnB e de seus prédios, fica fácil imaginar o grande desafio que a universidade tem de garantir a segurança patrimonial e os pontos acima citados.

Dentre os inúmeros prédios espalhados pela universidade o presente trabalho focará em um específico, a ULEG/FT, o mais novo prédio construído e pertencente à Faculdade de Tecnologia (FT).

Inaugurado no dia 05 de dezembro de 2018, a ULEG/FT tem como objetivo proporcionar aos alunos de graduação dos cursos de engenharia da UnB a possibilidade de estudar e aprender em laboratórios novos e equipados com bons aparelhos e computadores. A ULEG/FT serve todos os departamentos da FT.

Diante desse cenário, fica evidente a importância da segurança patrimonial do prédio novo. Além de possuir equipamentos novos e caros, a ULEG/FT possui fácil acesso, pois, como todos os prédios da UnB, não dispõe de muros, grades, portões ou qualquer tipo de dificultador para chegar ao prédio. Atualmente temos apenas as próprias portas do prédio e uma portaria interna como meio de controle de acesso interno.

Somado a isso, durante o período de aulas, a ULEG/FT deixa suas portas abertas ao público em geral que não precisa se identificar para adentrar ao prédio.

Além dessa questão da segurança, há uma preocupação também com o consumo de energia. Como diversas pessoas entre funcionários, professores e alunos utilizam as salas de aulas e banheiros, muitas vezes luzes e ar-condicionados são deixadas ligados sem ninguém estar presente ou utilizando.

Esse uso desnecessário pode acabar trazendo uma grande despesa, principalmente a questão do ar-condicionado, o grande vilão da conta de luz.

Os problemas citados não são únicos da ULEG/FT, são na verdade bem comuns, e é por

isso que o uso de IoT vêm ficando cada vez mais popular.

1.2 INTERNET OF THINGS

A Internet das Coisas é definida como um paradigma em que objetos equipados com sensores, atuadores e processadores comunicam entre si para servir um propósito [1].

Apesar da ideia de IoT não ser recente, foi o avanço de diversas tecnologias diferentes que permitiu que ela se tornasse viável e prática. O desenvolvimento na área de sensores foi o mais impactante. Os sensores viraram uma tecnologia acessível e confiável. Para aplicações corriqueiras, é possível encontrar um sensor com características específicas que permita aferir alguma variável do sistema que necessite de monitoramento. Isso só é possível graças à grande diversidade de sensores hoje disponíveis no mercado.

Outras tecnologias que tiveram avanços importantes e significativos para a viabilização do IoT foram a computação em nuvem, o aprendizado de máquina, a inteligência artificial e os protocolos de comunicação.

De acordo com P. Seth e S. R. Sarangi, a arquitetura de um sistema IoT é baseada em 3 camadas [1]. A primeira camada é a de coleta de dados, também chamada de camada de sensores ou camada física. Como o nome indica, ela consiste de sensores físicos, posicionados no ambiente ou no sistema que deseja-se monitorar, coletando dados e utilizando de algum protocolo de comunicação para repassar esses dados para a próxima camada.

A segunda camada IoT é a camada de processamento ou de middleware. Responsável por armazenar, analisar e processar os dados provenientes da primeira camada. Além disso é a responsável pela conexão entre dispositivos e sensores com dispositivos de rede e servidores.

A terceira e última camada é a de aplicação. Nela é onde se define o propósito do sistema, podendo ser uma casa ou cidade inteligente entre outras aplicações. Para isso ela é responsável por entregar serviços para o usuário.

A camada de processamento muitas vezes é destrinchada em outras três. Ao fazermos isso, entramos na arquitetura de 5 camadas, que substitui a camada de middleware por uma camada de transporte, uma de processamento e uma de regras de negócios.

Por último, a camada de aplicação é onde os dados coletados e processados são utilizados. Ela representa os serviços que o usuário utiliza. A Figura 1.1 ilustra a arquitetura de três e cinco camadas

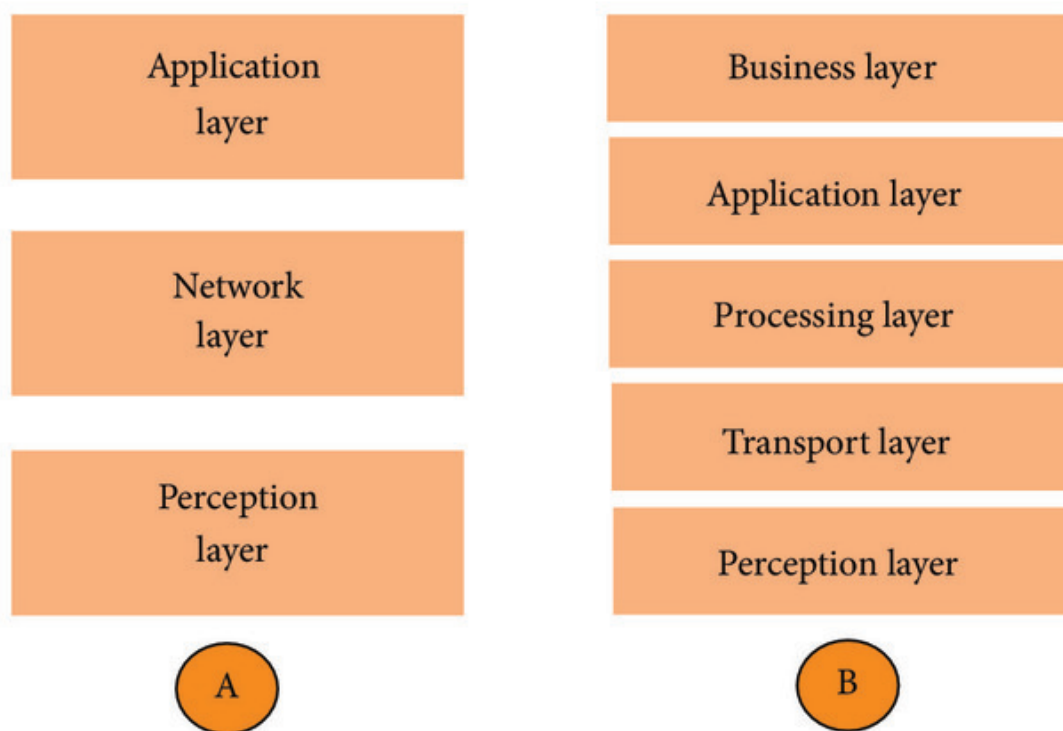


Figura 1.1 – Exemplo de arquitetura IoT (A: três camadas) (B: cinco camadas)

Começa a ficar evidente como é possível usar IoT para o gerenciamento e monitoramento dos ambientes laboratoriais do ULEG/FT.

1.3 DESAFIOS DA GESTÃO UNIFICADA DE AMBIENTES LABORATORIAIS

Para entendermos como o IoT é utilizado como solução precisamos primeiro compreender quais são os problemas e desafios de uma gestão unificada de ambientes laboratoriais, mais especificamente os que o ULEG/FT enfrenta.

Diante da era digital, a popularização do computador foi imensa. Somado a isso nos vimos no meio de uma pandemia, onde foi necessária a aquisição de meios para nos conectarmos de casa.

Com isso, o uso de computadores para simulações e estudos se tornou vital, inclusive em laboratórios. Nos dias atuais é comum encontrar laboratórios que possuem diversos computadores.

A ULEG/FT não é diferente disso, os alunos têm à disposição mais de 200 computadores espalhados por todo o prédio. Essa grande quantidade de aparelhos traz consigo um grande

receio de segurança, afinal não são máquinas baratas e seu uso deve ser controlado para o que se espera de um ambiente de aprendizado.

Além disso, cada máquina precisa estar configurada com os softwares e ferramentas corretos para que os alunos possam acompanhar as aulas. E assim nasce o problema de gerir computadores de diferentes ambientes laboratoriais. É facilmente compreensível o esforço que os técnicos da ULEG/FT têm para configurar cada máquina uma de cada vez.

Somado a isso, temos a possibilidade de computadores serem atualizados ou precisarem de novos programas frequentemente, seja para comportarem novas matérias, para acompanharem a evolução da tecnologia ou por questões de segurança.

Por isso é de interesse da ULEG/FT monitorar os computadores quanto ao fluxo de rede, ao estado (ligado ou desligado) e aos softwares disponíveis (sistema operacional, programas e versões). Também é de interesse ser capaz de instalar, desinstalar e atualizar softwares remotamente de acordo com seu sistema operacional.

1.4 GERENCIAMENTO, MONITORAMENTO E IMPLANTAÇÃO DE CONFIGURAÇÃO DE SOFTWARE

Um grande desafio de infraestrutura sempre foi gerenciar e configurar diversas máquinas de uma mesma arquitetura. Como reduzir o trabalho repetitivo de ter que configurar as máquinas com as mesmas configurações? Como realizar uma atualização geral? E claro, como monitorar se as máquinas estão fazendo o esperado?

Foi a partir dessas dificuldades que começaram a aparecer softwares capazes de automatizar e facilitar trabalhos repetitivos de configuração e manutenção, centralizando tudo em uma entidade com acesso à todas as máquinas e dispositivos.

A ideia de automatizar processos de entrega contínua utilizando de desenvolvimento e ferramentas de tecnologia é uma das vertentes do DevOps [2].

Esse conceito passou muito rápido para sistemas IoT. O que é compreensível, já que sistemas IoT são conhecidos por ter diversos dispositivos heterogêneos espalhados, onde todos devem enviar dados e serem monitorados e configurados.

Essa interface centralizada onde seja possível monitorar e atuar tanto nos sensores quanto nos computadores também é uma necessidade da ULEG/FT e será estudada neste trabalho.

1.5 ARQUITETURA DA SOLUÇÃO

Tendo em vista os requisitos apresentados, este trabalho propõe um sistema IoT com uma rede de sensores para monitorar os componentes físicos do ULEG/FT. O sistema precisa ter uma interface de gerenciamento e monitoramento que seja capaz de apresentar aos técnicos do ULEG/FT os estados dos sensores e realizar ações.

A arquitetura do sistema foi dividida em 4 camadas:

1. Camada física
2. Camada de gateway
3. Camada de monitoramento e atuação
4. Camada de armazenamento

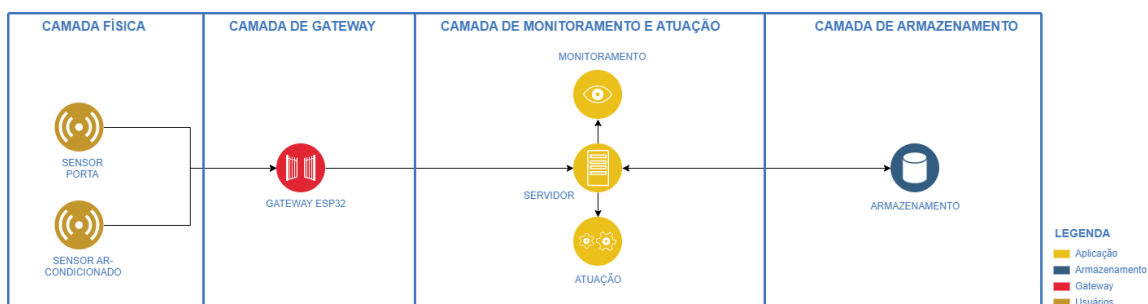


Figura 1.2 – Arquitetura da solução para a ULEG/FT

A primeira camada, camada física, compreenderá os dispositivos físicos instalados. Os dispositivos ficaram encarregados de coletar e repassar esses dados para a camada de gateway.

Já na camada de gateway teríamos o processamento desses dados que serão armazenados na camada de armazenamento. A camada de gateway também será usada para eventuais comandos enviados pela camada de monitoramento e atuação para os dispositivos, que deverão efetuar a ação.

Na terceira camada, temos a camada de monitoramento, que irá ler os dados atuais da camada de armazenamento e irá possuir uma interface de exibição para o usuário, que poderá através dela verificar os status dos dispositivos e emitir ações.

Por último, camada de armazenamento, utilizaremos o servidor presente no ULEG/FT para armazenar as informações coletadas pelos dispositivos. Essas informações devem ser fornecidas para a camada de monitoramento.

Vale ressaltar que nossa arquitetura possui uma camada a mais em relação à apresentada na seção 1.2. A camada de aplicação foi dividida nas camadas de armazenamento e monitoramento e atuação. Isso foi feito para facilitar o entendimento do nosso hardware e software, já que o armazenamento estará apartado do software de monitoramento e atuação.

1.6 OBJETIVOS DO TRABALHO

Diante do problema apresentado pela ULEG/FT e a necessidade dos técnicos de possuírem um controle melhor dos equipamentos e da segurança do prédio, esse trabalho tem como objetivo propor uma solução que seja suficiente para melhorar o dia a dia da gestão do prédio.

Para isso, o trabalho tem como objetivo implementar um sistema IoT capaz de ajudar o conteúdo da ULEG/FT. Será realizado um estudo para propor quais as melhores tecnologias, dispositivos e softwares a serem implementados na arquitetura proposta.

Tendo definido isso, vêm o desenvolvimento dos hardwares e softwares. Começando pelo hardware temos:

- Hardware do dispositivo responsável pelo controle de portas
- Hardware do dispositivo infravermelho para controlar ar-condicionado
- Hardware do gateway

Passando para o software temos:

- Software do dispositivo responsável pelo controle de portas
- Software do dispositivo infravermelho para controlar ar-condicionado
- Software do gateway
- Software do painel de controle
- Software de gerenciamento remoto de computadores

Os pontos listados formam o objetivo técnico que no final servirá de teste e validação da arquitetura proposta, já que o sistema desenvolvido será de menor escala. Espera-se que ele sirva de modelo para uma possível implementação na ULEG/FT.

1.7 ESTRUTURA DO TEXTO

Além desta introdução, este trabalho está organizado da seguinte forma: Capítulo 2 apresenta uma revisão da literatura sobre IoT, gerenciamento remoto e soluções para problemas similares. O capítulo 3 é dedicado à explicar a metodologia de desenvolvimento aplicada e de testes. O capítulo 4 descreve os respectivos resultados. Finalmente, o capítulo 5 apresenta as conclusões gerais e sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo tem como intuito apresentar trabalhos e soluções relacionadas com os problemas apresentados na introdução. Será feito um estudo buscando entender qual tecnologia melhor se aplica na arquitetura deste trabalho.

2.1 CONTROLE DE PORTAS UTILIZANDO IOT

Este estudo começa analisando o primeiro requisito da nossa arquitetura, monitorar remotamente quais portas do prédio da ULEG/FT estão abertas e quais estão fechadas. É um desafio comum em que é evidente a importância e necessidade de se ter um bom controle do fluxo de pessoas tanto dentro do prédio quanto em salas essenciais, onde ou o acesso é mais restrito ou possui itens de valor. Devido a essa importância, a discussão sobre o assunto é vasta no meio acadêmico. Aqui vamos apresentar trabalhos que serviram como base de estudo ou que tenham requisitos e soluções em comum com o presente trabalho.

Venkatraman, Varshaa e Vigneshwary[3] trazem uma proposta de arquitetura usando a tecnologia de comunicação LoRa para garantir a segurança de casas com um sistema de notificações de emergência. O artigo discute vantagens de utilizar a LoraWAN, uma rede de longa alcance, e elenca que o gateway LoRa consegue estar a uma distância de 100 km do microcontrolador e do sensor conectado à porta da casa. Esse longo alcance traz vantagens expressivas principalmente do ponto de vista comercial. É possível ter um mesmo gateway cobrindo um raio de cobertura de 100 km, o que permite a instalação do mesmo em um local seguro e apropriado, além de não necessitar da instalação de vários gateways espalhados por diversos lugares.

Além disso ele traz uma arquitetura parecida com a nossa. Possui um sensor conectado a um microcontrolador, que comunica com um gateway, que por sua vez comunica com um servidor que cuida do envio das mensagens de emergência.

Apesar da arquitetura ser semelhante, o artigo utiliza como microcontrolador um Arduino Uno. Por não ter em seu hardware um comunicador em LoRa, o Arduino Uno precisa ainda de um LoRa Shield, utilizado para realizar a comunicação em LoRa com o gateway. Uma vantagem da arquitetura proposta pelo presente trabalho é que o nosso microcontrolador, ESP32, já é capaz de comunicar diretamente com o gateway pelo protocolo de comunicação do ESP-NOW.

Já Luna, Sánchez-Rangel e Cosme-Aceves[4] trazem um estudo sobre um sistema monitoramento de três portas e duas janelas de uma sala de data center. Mudanças de estados são registrados no servidor e disponibilizados online por uma interface de usuário.

Foram utilizados dois sensores Hall effect, uma placa PyBoard e uma interface Wi-Fi em cada nó de monitoramento. Ao ocorrer uma mudança de estado, o nó comunica um nó central, uma espécie de gateway, que por sua vez transmite para o servidor através do Wi-Fi.

A escolha do sensor Hall effect é interessante e difere da escolha deste trabalho que foi por um sensor Reed Switch. Os dois sensores são os mais utilizados para a detecção de abertura de porta. Quando uma porta ou janela fica aberta por um tempo maior que o configurado, um SMS e uma mensagem via WhatsApp são enviados para o celular do administrador.

Outro diferencial do trabalho é que foi construído um servidor web utilizando a biblioteca open source uasyncio. Com ela foi possível criar uma interface simples para o usuário visualizar o estado dos sensores e que usa o mínimo de memória RAM.

Em comparação com o artigo anterior, o alcance da transmissão Wi-Fi só conseguiu chegar à uma distância de 47 metros em linha reta, porém, houve um foco maior na parte de interface do usuário.

Outro estudo, realizado por Adriano, Budi et al.[5], implementa o uso de um microcontrolador com interface Wi-Fi, um ESP8266, para detectar a presença de intrusos adentrando pela porta da casa monitorada. Ao perceber uma possível invasão, o ESP8266 envia uma notificação para o usuário responsável pela administração da casa por um aplicativo.

Fugindo um pouco do padrão de sensores Reed Switch e Hall effect para o monitoramento da porta, o estudo optou por utilizar um sensor PIR HC-SR501. É muito comum se deparar com esse tipo de sensor em corredores de prédio ou banheiros, ele é um sensor de movimento que nesses casos é utilizado para ligar as luzes ao perceber algum movimento. Essa característica dele pode tornar o sistema muito sensível, visto que qualquer movimento perto da porta será captado por ele.

Uma funcionalidade interessante do estudo é a implementação de um sistema de validação por tag e senha para a abertura da porta. Ele é descrito por um algoritmo que pode ser simplificado em primeiro verificar se a tag RFID é aceita e após isso ler de um teclado a senha. Se a senha estiver correta a porta abre, senão uma notificação é enviada via email.

Por último mas não menos importante, Aldawira et al.[6], traz um sistema de segurança de porta para monitoramento residencial utilizando o microcontrolador ESP32, o mesmo que este trabalho utilizará. O artigo propõe a utilização do protocolo MQTT com criptografia SSL para a comunicação no sistema de segurança. O microcontrolador ESP32 fica encarregado de ler as informações de um sensor PIR de movimento. Se detectado movimento, ele se comunica via Wi-Fi para transmitir a informação para o resto do sistema. Além disso, ele ainda recebe comandos de emergência e aciona um módulo de alarme de buzina.

O ESP foi compilado utilizando a plataforma Arduino, uma alternativa que traz grande

facilidade para o desenvolvimento do firmware. Além disso o sistema permite dar acesso à usuários confiáveis que conseguem controlar a porta, ver o histórico e o log do acesso dos usuários, utilizar um botão de reset e um alarme de buzina.

2.2 CONTROLE DE CONDICIONADORES DE AR UTILIZANDO IOT

O próximo tópico deste estudo também é um tema comum, porém, dessa vez se trata menos de segurança e mais de economia de energia elétrica. Não é nenhuma surpresa que o ar-condicionado é conhecido por ser o vilão da conta de luz, tanto em ambientes domésticos quanto em corporativos e estudantis.

Na ULEG/FT, a situação não é diferente. Com todas as salas contendo pelo menos um ar-condicionado pode-se imaginar o prejuízo de esquecer todos ligados durante apenas uma noite.

Como se trata de um problema comum, é de se esperar que haja várias soluções propostas na literatura. Santos[7] traz à tona as soluções e os desafios tecnológicos para controlar equipamentos de ar-condicionado utilizando IoT. O trabalho propõe uma central de processamento que conta com uma placa de Arduino que é equipada com sensor de umidade e temperatura e componentes de infravermelho que possibilitam o controle do aparelho de ar-condicionado de maneira automática pelo Arduino.

Para o controle remoto, a central tem anexado um módulo Wi-Fi ESP8266 Esp-01 responsável pela conexão wireless com um aplicativo de dispositivo móvel. Nele é possível configurar a central para agir analisando a temperatura e a umidade relativa do ambiente. Através do aplicativo também é possível desligar e ligar os equipamentos a qualquer momento.

O aplicativo foi desenvolvido especialmente para o projeto utilizando o Android Studio e as tecnologias Java/Kotlin para a codificação do aplicativo.

Cunha[8] traz outro contexto em que um sistema de controle e monitoramento de refrigeração pode ser usado. O artigo visa ambientes especiais, como unidades de saúde, mais precisamente farmácias ambulatoriais que necessitam de um controle de temperatura e umidade para prevenir danos e descartes indesejáveis dos insumos de grande importância para saúde como vacinas, medicamentos, sangue, amostras biológicas, entre outros.

Foi utilizado para interface com o usuário a plataforma Blynk, desenvolvida para iOS e Android para controlar Arduinos, Raspberry Pi e outros dispositivos. Conectando o arduino à um ESP8266, responsável pela comunicação Wi-Fi, o Blynk é capaz de receber e enviar formações para o Arduino, que por sua vez desliga e liga os equipamentos utilizando um LED de infravermelho.

3 PROJETO

Este capítulo tem o propósito de explicar as escolhas de hardware e firmware tomadas para a arquitetura, comparando tecnologias e modelos.

3.1 ESCOLHA DO SENSOR DE MONITORAMENTO DA PORTA

A primeira coisa que foi necessária decidir foi a escolha do sensor de monitoramento da porta. Como vimos no capítulo anterior, os sensores mais utilizados são o Reed Switch, o Hall effect e o sensor de movimento PIR.

A escolha do PIR foi inicialmente rejeitada. Ele é um sensor que quase todos já devem ter visto em corredores de prédios ou banheiros e é utilizado principalmente para detectar movimento. Nos casos mais comuns a detecção do movimento está ligada com o acionamento das luzes do local, garantindo a automação e economia de energia.

Porém, para este trabalho não é o desejado, queremos um sensor mais preciso, que consiga diferenciar uma pessoa passando na frente da porta da ação de realmente abrir a porta e por isso, o sensor PIR foi descartado.

O Reed Switch e o Hall effect são parecidos em alguns aspectos. Os dois são considerados sensores magnéticos, porém, apenas o Hall Effect é um transdutor, já que ele converte energia magnética em energia elétrica [9].

A grande diferença está na alimentação, enquanto o Reed Switch não consome energia e funciona apenas na presença de um campo magnético, o Hall effect precisa de circuito elétrico alimentando ele para produzir um sinal de saída.

Apesar de não ser considerado um transdutor, o Reed Switch funciona com um par de palhetas finas feitas de materiais ferromagnéticos protegidos por um invólucro. Ao estar suficiente perto, ou seja, quando há um campo magnético forte o suficiente para unir as palhetas, elas se tocam e fecham o circuito, permitindo a passagem de corrente elétrica.

Algumas das aplicações do Reed Switch são:

- Sensores de velocidades em rodas de bicicletas, engrenagens de carros e esteiras. É possível medir o tempo que um dos pares magnéticos demora para dar uma volta completa em uma roda e se encontrar novamente com o outro par. Sabendo o tamanho da roda fica fácil calcular a velocidade
- Sensores de proximidade em eletrônicos. Utilizado para detectar se dispositivos, como um laptop ou um celular flip, foi fechado

- Em ambientes com maiores imprevisibilidades de temperatura e fatores ambientais, devido a sua característica mecânica

Já o Hall effect é um transdutor semicondutor que na presença de um campo magnético gera uma diferença de potencial elétrica, o que leva à passagem de corrente elétrica.

Algumas das aplicações do Hall effect são:

- Medindo a velocidade de rotação de rodas e eixos
- Ignições temporizadas em motores de combustão interna
- Tacômetros e sistemas de freios antitravamento
- Interruptores de limite em atuadores para sistemas controlados automatizados.

Como o nosso sensor estará inevitavelmente ligado à um microcontrolador, a questão do Hall effect precisar ser alimentado é minimizada. Avaliando os dois sensores foi possível concluir que ambos funcionariam e supririam a necessidade de identificar a abertura de uma porta, o que fica evidente pelo equilíbrio na literatura do uso dos dois sensores.

A decisão de qual sensor utilizar ficou então dependente do microcontrolador. A maioria dos microcontroladores que possuem um dos sensores embutidos, utiliza o Hall effect, como vamos alguns exemplos, por isso a decisão ficou simplificada na disponibilidade do sensor que os microcontroladores já trazem.

3.2 COMPARAÇÃO DE MICROCONTROLADORES

Para a escolha do microcontrolador foram elencados quatro atributos para a comparação:

- Preço
- Tecnologia de comunicação
- Bateria e consumo de energia
- Possui Hall effect sensor

Após um estudo inicial dos artigos compartilhados no capítulo 2, de fóruns pela internet e alguns conhecimentos prévios, quatro microcontroladores foram selecionados como possíveis boas escolhas:

- ESP32

- ESP8266
- TIDA01066
- NRF51822

Eles foram submetidos aos quatro atributos e após a comparação feita abaixo, foi tomada a decisão de qual microcontrolador será utilizado.

3.2.1 Preço

É preciso manter em mente o objetivo inicial do trabalho. Desenvolver um sistema para uma entidade pública deve sempre levar em conta o preço final da solução. Lembrando também que o intuito é que o produto final possa ser replicado para o prédio da ULEG/FT todo.

Sabendo dessa importância, foi construída uma tabela com o preço médio dos microcontroladores a partir de pesquisas feitas pela internet.

Tabela 3.1 – Preços dos componentes

Componente	ESP32	ESP8266	TIDA0166	NRF51822
Preço (R\$)	34,99	19,90	100+	76,90

Com exceção do TIDA0166 todos os microcontroladores conseguem ser comprados no Brasil com certa facilidade em lojas de eletrônica.

O TIDA01066 é um design de referência da Texas Instruments [10]. Um design de referência é um projeto técnico de um sistema que se destina a ser copiado por outros. Normalmente, um circuito integrado ou outro dispositivo eletrônico que requer que outros componentes ou circuitos sejam usados em conjunto com seu produto. O que significa que ele não é um produto que é vendido pelo mercado, ele é uma série de especificações que tem o intuito de permitir que o usuário que queira um possa construí-lo ou encomenda-lo.

Seguindo o design utilizaríamos o módulo wireless CC1310 (R\$68,19 + taxas), uma bateria moeda CR2032 (R\$1,95), dois Hall effect sensor(U\$0.64 cada). O preço final seria de no mínimo 100 reais, já que ainda teríamos os pinos, os LEDs e a placa.

Comparando todos os microcontroladores, a família ESP levou grande vantagem em relação ao preço e disponibilidade.

3.2.2 Tecnologia de comunicação

Um aspecto muito importante que deve ser levado em consideração é a tecnologia de comunicação que o microcontrolador a ser selecionado utiliza. Para a nossa arquitetura é

necessário que a tecnologia seja sem fio, uma vez que a comunicação com o gateway ficaria muito limitada e pouco versátil ao ambiente se precisássemos de fios conectando diretamente os sensores ao gateway.

As tecnologias que os microcontroladores usam são:

- O ESP32 suporta Bluetooth 4.2, Bluetooth low energy, Wi-Fi e ESP-NOW.
- O ESP8266 suporta Wi-Fi e ESP-NOW.
- O TIDA01066 usa de uma comunicação Sub-1GHz wireless.
- O NRF51822 é feito para usar o Bluetooth® Low Energy e aplicações sem fio proprietárias de 2,4 GHz.

O protocolo de comunicação ESP-NOW é de baixa energia e proprietário da família dos ESPs. Ele possui um alcance de 220 metros em campo aberto [11] ou até de 480 metros, segundo Koyanagi[12]. A princípio o alcance é mais do que o suficiente para a arquitetura proposta.

Além de uma comunicação sem fio é preciso pensar no consumo de energia. Todos dispositivos possuem uma opção low energy, entre elas, o ESP-NOW é uma comunicação interessante que pode aumentar o tempo de vida da bateria do ESP32 em até 6 vezes [13], porém, ela pode ser utilizada apenas entre dispositivos ESP e com curtas mensagens, o que implicaria que o gateway da nossa arquitetura teria de utilizar também um ESP32 ou um ESP8266 para receber as mensagens.

Ainda pensando no consumo de energia, o ESP8266 só seria interessante se de fato utilizarmos o ESP-NOW, já que a comunicação Wi-Fi consumiria muita bateria e seria necessário uma troca muito constante.

Já o ESP32 possui a possibilidade também de utilizar o Bluetooth low energy, permitindo uma maior flexibilidade do dispositivo a ser utilizado no gateway.

O TIDA01066 utiliza de uma comunicação proprietária, o que implicaria novamente que nosso gateway tenha que se comunicar na tecnologia de Sub-1GHz e utilizar de um dispositivo da Texas.

3.2.3 Bateria e consumo de energia

A nossa solução foi pensada para possuímos diversos sensores espalhados por um grande prédio. Apesar de ser possível, a melhor abordagem seria que esses sensores não precisassem de cabos físicos conectando eles ao nosso gateway ou à uma alimentação.

A utilização de cabos exigiria um esforço de instalação bem maior, além de tornar alguns lugares longes de tomadas inoperantes. Por isso vamos levar em consideração na nossa comparação o consumo de energia e microcontroladores que tenham placas que já venham com bateria.

Mesmo que a presença de uma bateria não esteja presente na placa comercial do microcontrolador, iremos utilizar uma para alimentar o microcontrolador. Fica evidente então que a escolha do microcontrolador deve acarretar em uma solução que não demande as trocas de baterias muito constante, o que poderia tornar a solução ineficaz para o ambiente da ULEG/FT.

Dito isso foi construído a seguinte tabela:

Tabela 3.2 – Bateria e consumo de energia

Componente	ESP32	ESP8266	TIDA0166	NRF51822
Possui bateria	Não	Não	Sim	Sim

Os microcontroladores da família ESP não possuem bateria, porém, possuem diversos modos de economia de bateria, como Sono profundo ou Hibernação. Eles podem ser explorados para permitir que o uso de uma bateria externa não demande uma troca com frequência muito alta, já que ao estar "dormindo" o microcontrolador utilizará uma corrente elétrica muito menor.

No total o ESP32 possui 5 modos de energia, cada um possui recursos e capacidades de economia de energia diferentes. Eles são:

- Ativo
- Suspensão do modem
- Sono leve
- Sono profundo
- Hibernação

No modo ativo temos todos os recursos ativos e um consumo médio de 240 mA, sendo que este valor pode chegar a 790 mA ao utilizar o Wi-Fi junto ao Bluetooth.

Já no modo de suspensão do modem, o Wi-Fi, Bluetooth e rádio ficam desativados. Neste modo o chip consome de 3 a 20 mA.

O modo de sono leve é bem parecido com o anterior, porém, a CPU é pausada desabilitando seu pulso de clock, o que resulta em um consumo menor de energia em torno de 0,8 mA.

Passando para o modo de sono profundo, a CPU, a maioria da RAM e todos os periféricos digitais também ficam desabilitados. Apenas o co-processador ULP e o RTC permanecem ativos. Essa mudança permite que o chip consuma entre 150 μA (quando o ULP estiver ligado) e 10 μA .

Esse é um dos modos mais interessantes e utilizados do ESP32. Ao deixar apenas o ULP e o RTC habilitados, o microcontrolador consome uma quantidade baixíssima de energia, mas ainda consegue ser acordado através do ULP para responder a eventos. Fica evidente que é um modo que se encaixa muito bem na nossa proposta, já que podemos colocar nosso ESP para dormir e acordar apenas quando o estado da porta monitorada por ele for alterado, por exemplo.

O último modo, o modo de hibernação, é similar ao de sono profundo, porém, nele o ULP também é desabilitado, restando apenas o RTC. Isso resulta em um consumo de energia de 2.5 μA , mas sem o ULP não conseguimos acordar nosso microcontrolador em razão de um evento, apenas após um tempo pré estabelecido [14].

O grande problema do consumo de energia do ESP32 é a sua placa de desenvolvimento (ESP32 Devkit board), que consome quase 4.10 mA por utilizar o CP2102(Conversor USB-Serial) e um regulador linear [15].

Para contornar este problema existem outras placas de desenvolvimento de mais difícil acesso, como a ePulse [16]. Ela conta com uma corrente de 12 a 27 μA no modo de sono profundo e foi feita intencionalmente para casos como o nosso onde o ESP ficaria transitando entre modo ativo e de sono profundo, e uma economia de energia é necessária.

3.2.4 Hall effect sensor

Por último comparamos quais microcontroladores já vêm com um Hall effect sensor na sua placa. Dos componentes listados apenas o ESP32 e o TIDA0166 já vem com pelo menos um.

Tabela 3.3 – Possui Hall effect sensor

Componente	ESP32	ESP8266	TIDA0166	NRF51822
Possui Hall effect sensor	Sim	Não	Sim	Não

3.3 GATEWAY

De acordo com nossa arquitetura, o gateway será o responsável pelo processamento dos dados, pela comunicação com a camada de armazenamento e por enviar eventuais comandos da camada de monitoramento para a camada de sensores.

A escolha do nosso gateway é diretamente impactada pela escolha do microcontrolador a ser utilizado na camada física.

O gateway deve ser capaz de se comunicar com todos os sensores e com o servidor. Por isso, ele deve seguir os mesmos protocolos de comunicação dos microcontroladores utilizados nos sensores e possuir uma comunicação Wi-Fi com o servidor.

Lembrando que os microcontroladores da família ESP que estão sendo utilizados nas comparações já possuem um módulo Wi-Fi integrado em sua placa. O que nos inclina cada vez mais a escolhê-los.

3.4 ESCOLHA DO MICROCONTROLADOR

Tendo em vista todos os pontos apresentados durante este capítulo, optou-se por utilizar o ESP32 como nosso microcontrolador na camada de física e para uma compatibilidade fácil também usaremos ele como nosso gateway.

Em comparação com o ESP8266, a única desvantagem do ESP32 é o preço, porém, não é uma diferença tão expressiva que justifique a escolha do ESP mais antigo. O ESP32 é mais novo e possui interfaces e modos de economia de energia melhores, além de já vir com o Hall effect sensor.

Os microcontroladores serão configurados para utilizar o protocolo ESP-NOW para comunicarem entre si. O ESP-NOW permite uma comunicação sem fio e de baixo consumo de energia e será suficiente para a nossa solução no ULEG/FT.

Além disso, vamos explorar o modo de sono profundo do ESP32 para buscar a maior eficiência da bateria possível. O ESP posicionado na camada física ficará "dormindo" e quando o sensor registrar um valor superior ao limite, evidenciando uma mudança de estado da porta, o microcontrolador irá acordar e mandar uma mensagem para o gateway com o estado da porta.

Para o gateway, o ESP32 será configurado para conseguir se comunicar com todos os outros. Como teremos apenas um e seu posicionamento deve ser mais reservado, podemos alimentá-lo direto à uma fonte de energia, sendo assim ele não irá transitar entre modos de economia de energia.

3.5 COMPARAÇÃO DE SOFTWARES DE MONITORAMENTO/GERENCIAMENTO DE MÁQUINAS

Para a nossa arquitetura precisamos de um software capaz de gerenciar várias máquinas Windows e Linux que estão ligadas em uma mesma rede local. Eles devem ser monitorados quanto ao fluxo de rede, ao estado (ligado ou desligado) e aos softwares disponíveis (programas e versões).

Além disso, deve haver uma interface com o usuário que permita ele ter fácil visualização dos sensores como um tudo e realizar ações.

Os dois softwares mais famosos e utilizados para isso são o Ansible e o Puppet. Ambos possuem uma versão de código aberto gratuita [17].

O Ansible é conhecido por ser de fácil instalação e rápida implantação, já o Puppet é utilizado para instalações mais complexas.

A ideia principal do Ansible é se conectar em cada host, rodar todos os comandos e instruções previamente planejados, chamado de playbook, e após isso limpar tudo e desconectar do host. Essa abordagem fugiria um pouco da nossa ideia de monitoramento já que ele não oferece recursos de monitoramento remoto, busca de informações de localização e monitoramento de recursos em tempo real e registro de dados [18].

O Puppet possui uma GUI (Graphic User Interface) bem desenvolvida onde você pode gerenciar e monitorar suas máquinas. Já o Ansible possui uma GUI no Ansible Tower, sua versão comercial, porém, existe o AWX GUI, uma versão grátis do Ansible Tower [19].

Além disso, o Puppet possui o Puppet device [20], usado para gerenciar dispositivos que não conseguem ter um Puppet agent neles, como é o caso dos nossos sensores. Ele usa então um Puppet proxy agent, que seria nosso gateway, que ficaria encarregado da comunicação com os sensores, e então passaria para o Puppet master as informações, o sentido contrário também é válido.

4

METODOLOGIA

Este capítulo tem apresenta a metodologia desenvolvida e aplicada para a construção do sistema de gerenciamento e monitoramento para a ULEG/FT. Nele será debatido o desenvolvimento de hardware, desenvolvimento de firmware, teste de integração e a implementação e lançamento do sistema completo.

4.1 DESENVOLVIMENTO DE HARDWARE

Nesta seção será apresentada a metodologia para o desenvolvimento do hardware do sistema.

Foi desenvolvida uma prova de conceito do que seria aplicado para um prédio inteiro. Nela teremos apenas um dispositivo de cada tipo, ou seja, um dispositivo de porta e um dispositivo infravermelho para o ar-condicionado.

O microcontrolador a ser utilizado para os dispositivos é o ESP32. O dispositivo de porta é um dispositivo apenas de leitura, enquanto o dispositivo infravermelho é um dispositivo apenas de escrita e sendo assim, o microcontrolador deve ser configurado para ler do pino que o Reed Switch está conectado e escrever no pino conectado à um LED infravermelho.

Além dos dispositivos, teremos um gateway que também será um ESP32 e será responsável pela comunicação entre a camada física e a camada de monitoramento e atuação.

4.1.1 Dispositivo de porta

Para o dispositivo de porta, inicialmente tentou-se utilizar um sensor Hall effect, aproveitando que a nossa placa de desenvolvimento do ESP32 já possui um internamente. Ele mede a variação do campo magnético. Para sua efetividade, um ímã deve ser colocado preso à porta e de uma forma que ao abrir e fechar a porta, o ímã cause uma variação do campo magnético no sensor Hall effect.

Ao utilizá-lo, deve-se estabelecer limites inferior e superior, explicitando que tanto uma variação positiva quanto negativa signifique que a porta foi aberta ou fechada.

Ao implementar o dispositivo e começar sua leitura, percebeu-se que teríamos algumas dificuldades em sua utilização. Primeiramente, definir os limites inferior e superior dependeria muito da força do ímã e do posicionando tanto do ímã quanto do sensor Hall effect, condições que podem vir a ser difíceis de serem replicadas para as diversas salas da ULEG/FT, tendo em vista que há tipos diferentes de portas que podem dificultar principalmente o posi-

cionamento dos elementos.

Logo em seguida outro problema foi detectado, o tempo que seria considerado entre variações que significa-se uma mesma mudança de status da porta. Como o sensor de Hall effect não mede o status atual da porta, ou seja, se está fechada ou aberta, e sim suas variações, ações de abrir e fechar a porta, um erro na medida seria propagado causando diversas medidas errôneas.

Devido a esses problemas e outras dificuldades enfrentadas, decidiu-se utilizar o sensor Reed Switch ao invés do Hall Effect. Um efeito direto que a mudança causa é medir o status da porta e não sua variação.



Figura 4.1 – Sensor Reed Switch

O sensor Reed Switch funciona como uma chave. Ele é composto por 2 juncos ferrosos que na presença de um ímã com campo magnético forte o suficiente se tocam, fechando o circuito e permitindo a passagem de corrente elétrica.

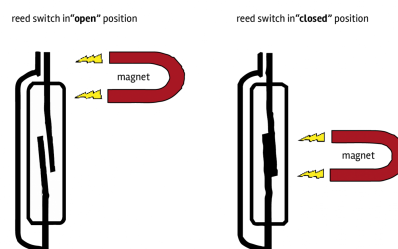


Figura 4.2 – Como o Reed Switch se comporta

Para a implementação do sensor Reed Switch no projeto, foi utilizado um módulo que pode ser visto na Figura 4.3.

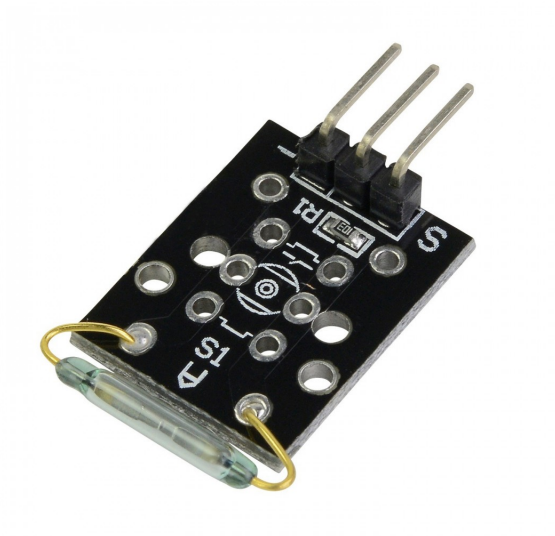


Figura 4.3 – Módulo Sensor Magnético

As Figuras 4.4 e 4.5 apresentam o hardware do dispositivo de porta utilizado, nelas é possível identificar o ESP32 ligado pelo pino 4 ao módulo que possui o Reed Switch, além disso ele está conectado também ao 3,3 V do ESP32 e ao terra.

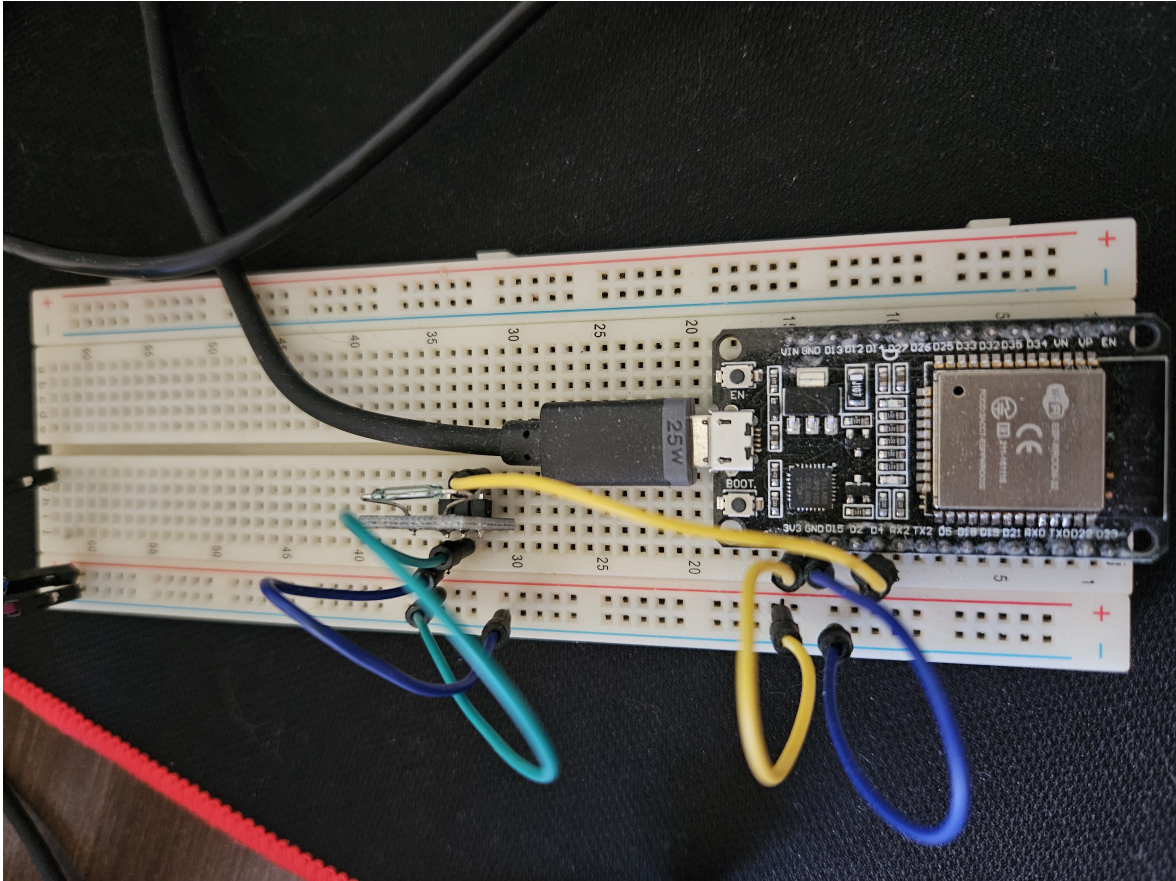


Figura 4.4 – Dispositivo de porta utilizado no protótipo

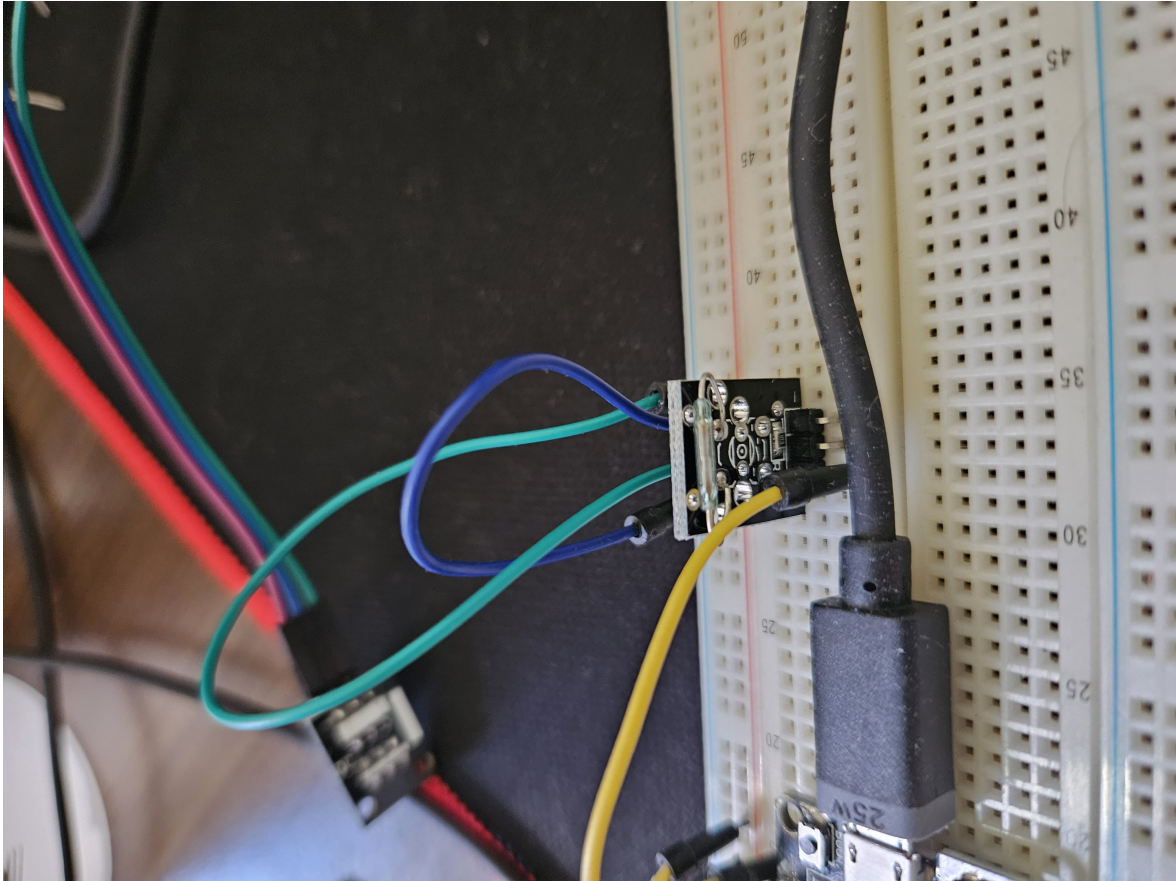


Figura 4.5 – Módulo magnético utilizado no protótipo

4.1.2 Dispositivo infravermelho

Para controlar um ar-condicionado é necessário enviar sinais infravermelho com comandos específicos. Iremos explorar apenas os comandos de ligar e desligar neste trabalho, o que é suficiente para o escopo da ULEG/FT.

Um sinal infravermelho pode ser enviado através de um LED infravermelho apontado diretamente para o aparelho de ar-condicionado, nesse quesito o hardware é simples, tendo em vista que o LED é um item barato e fácil de achar. Entretanto, o LED precisa ser controlado por um microcontrolador para que o sinal enviado esteja na frequência certa e enviando o comando certo.

Novamente foi utilizado o ESP32 como o microcontrolador, o que facilita a comunicação com o gateway. Ele será responsável por receber os comandos vindos do gateway e se encarregar de que o LED mande o comando correto para ligar ou desligar e na frequência correta.

Os protocolos de infravermelho utilizados pela fabricantes de ares-condicionados funcionam baseados no Protocolo de Transmissão por Infravermelho NEC. Este protocolo utiliza codificação de distância de pulso dos bits de mensagem. Para cada bit este tipo de codifica-

ção envia um sinal composto por duas partes. A primeira chamada de mark e a segunda de space. O que vai definir se o bit é '0' ou '1' é o tamanho do seu space.

O protocolo NEC funciona da seguinte maneira:

- '0' lógico - Um pulso de $562,5 \mu s$ seguido de um espaço de $562,5 \mu s$, totalizando um tempo de $1,125 \text{ ms}$
- '1' lógico - Um pulso de $562,5 \mu s$ seguido de um espaço de $1,6875 \text{ ms}$, totalizando um tempo de $2,25 \text{ ms}$

Uma mensagem consiste dos seguintes itens:

- um pulso inicial de 9 ms (16 vezes o comprimento da rajada de um pulso usado para um bit de dados lógicos)
- um espaço de $4,5 \text{ ms}$
- o endereço de 8 bits para o dispositivo receptor
- o inverso lógico de 8 bits do endereço
- o comando de 8 bits
- o inverso lógico de 8 bits do comando
- um pulso final de $562,5 \mu s$ para significar o fim da transmissão da mensagem.

Um exemplo de mensagem enviada utilizando o protocolo NEC pode ser vista na Figura 4.6.

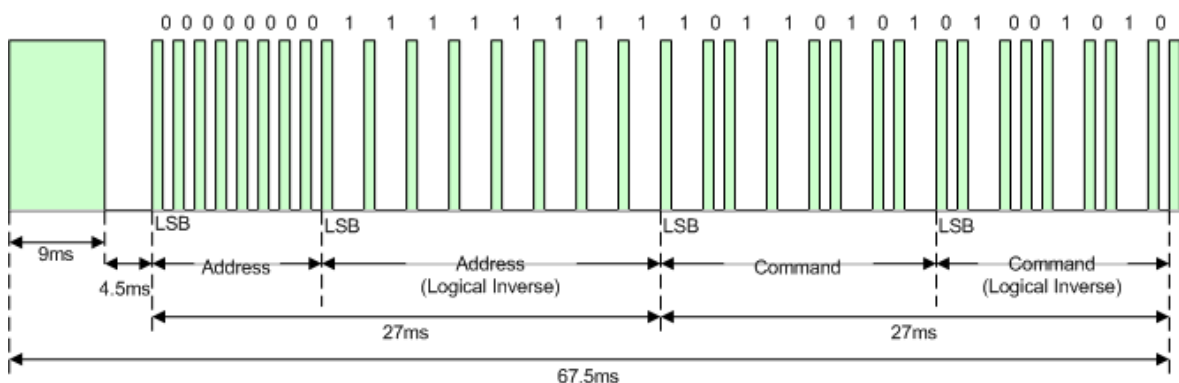


Figura 4.6 – Exemplo de mensagem utilizando o protocolo NEC IR

Nos protocolos proprietários das fabricantes, é esperado que sigam o mesmo modelo, porém, o tempo de mark, de space e dos pulsos iniciais e finais são outros.

Como podemos perceber, apesar do dispositivo ser simples, apenas um LED infravermelho conectado ao ESP32, ser capaz de reproduzir os comandos infravermelho produzidos por um controle de ar-condicionado pode ser uma tarefa árdua. É necessário descobrir como o protocolo funciona, quais são seus tempos de mark, space e pulsos iniciais e finais. Além disso, é necessário descobrir qual é o comando utilizado para ligar e desligar o aparelho.

Para descobrir esses pontos foi utilizado um decodificador de infravermelho, mostrado na Figura 4.7. Ele possui 3 pinos, um de Vcc, um de terra e um de sinal. Ao receber um sinal infravermelho seu LED interno brilha e ele transmite os bits codificados através do pino de sinal. É importante ressaltar que ele funciona baseado no protocolo NEC, ou seja, ele espera receber a portadora de 38 kHz. Assim, o sinal reportado pelo decodificador não apresenta a portadora.

Alinhado ao decodificador, foi utilizado um analisador lógico para facilitar a captura das medidas. Foi possível observar mensagens no mesmo formato da Figura 4.6.

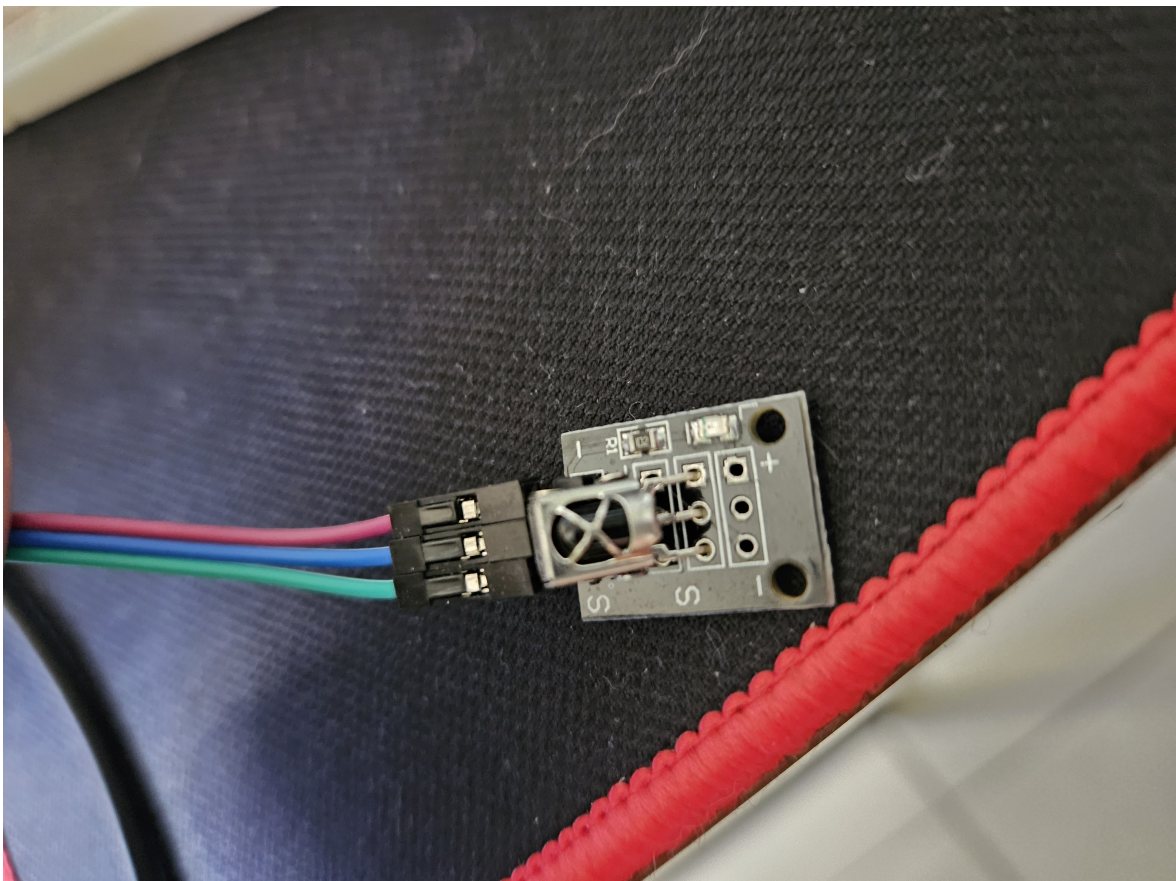


Figura 4.7 – Decodificador IR

Em cima disso, é necessário garantir também que a mensagem enviada utilize uma portadora de 38 kHz, ou seja, configurar o LED nessa frequência. A frequência da portadora também é outra variável que pode ser diferente para cada fabricante, porém, na maioria de suas vezes ela é mantida. Os protocolos estudados e analisados por este trabalho, das fabri-

cantes LG e Hitachi, confirmam isto utilizando a frequência de 38 kHz.

Todo este controle da mensagem enviada pelo LED infravermelho foi feito via software no programa do nosso microcontrolador ESP32, o qual irá receber os comandos via ESP-NOW do gateway.

As Figuras 4.8 e 4.9 apresentam o hardware do dispositivo infravermelho utilizado, nelas é possível identificar o ESP32 ligado pelo pino 4 à um resistor que por sua vez está conectado ao LED infravermelho, que fecha seu circuito conectando-se ao terra.

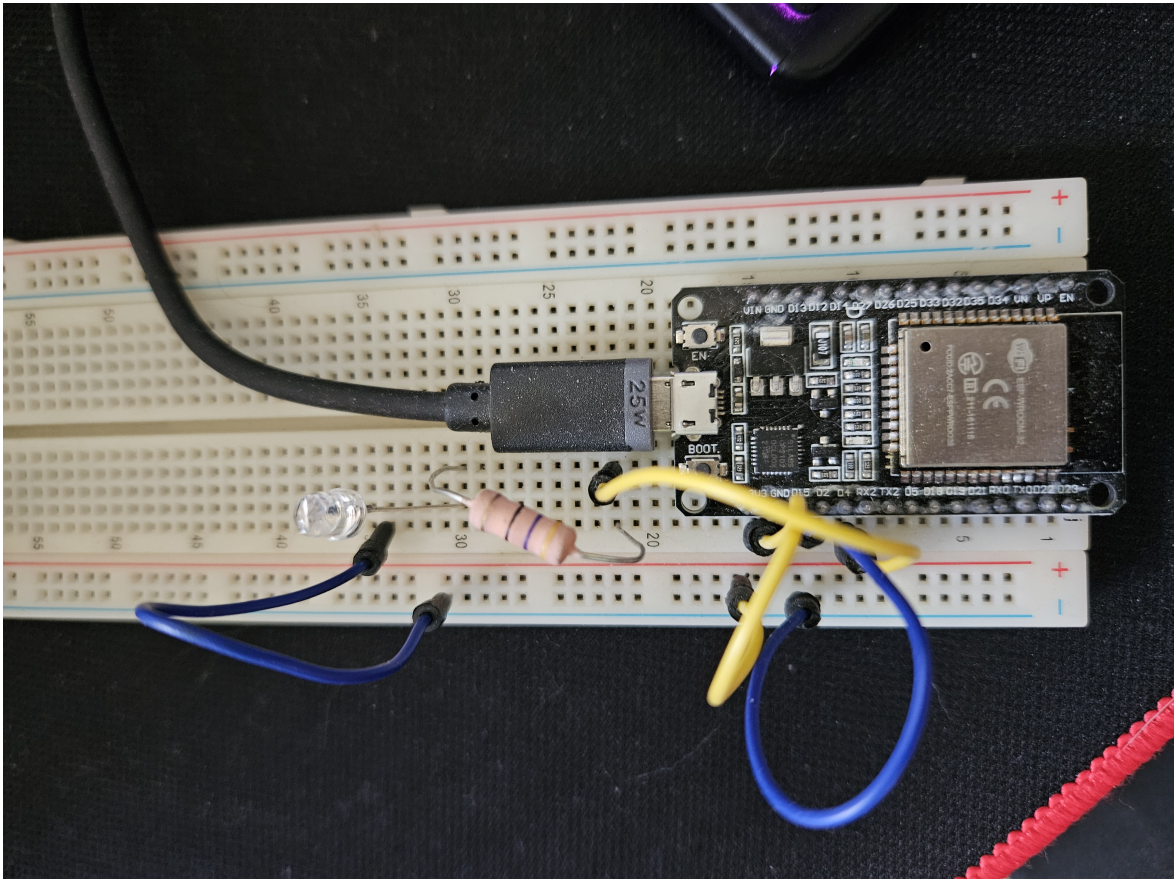


Figura 4.8 – Dispositivo infravermelho utilizado no protótipo

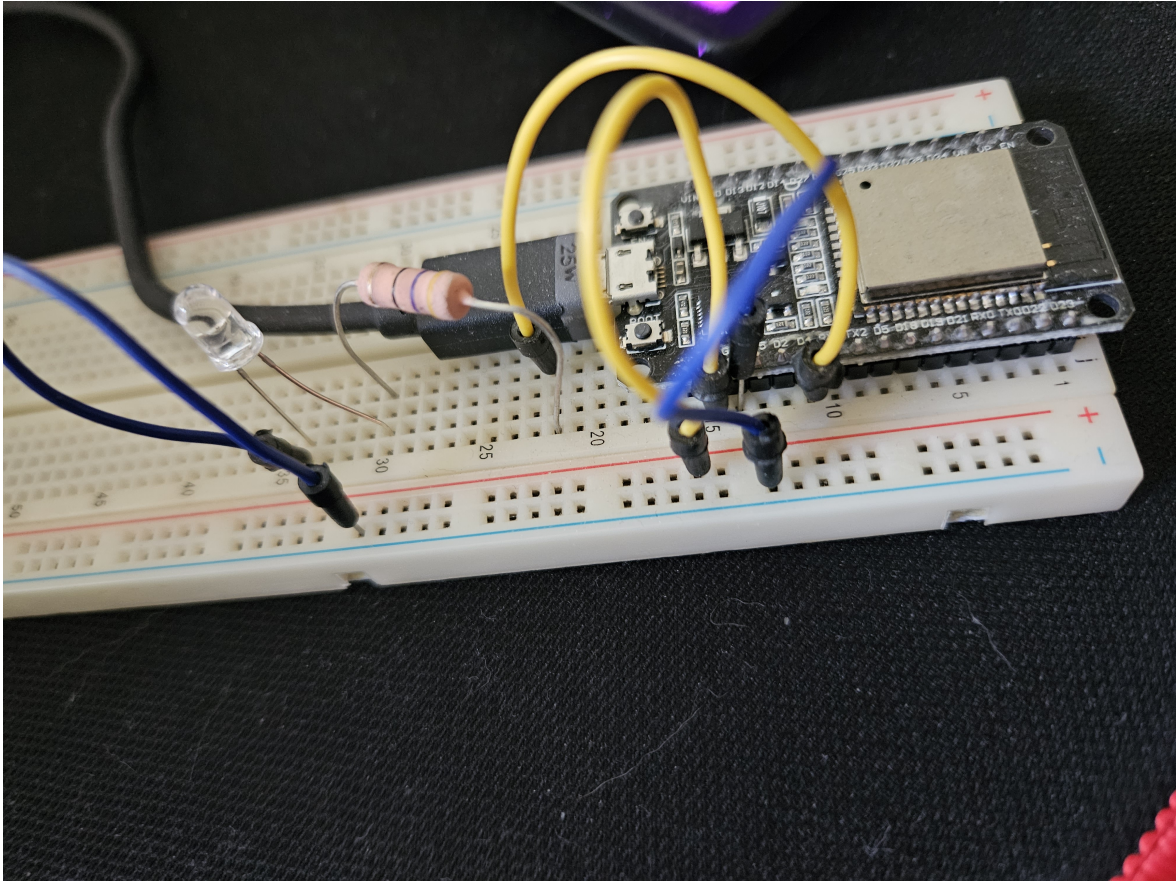


Figura 4.9 – Outra visão do dispositivo infravermelho utilizado no protótipo

4.1.3 Gateway

O Gateway é sempre uma peça chave de sistemas IoT, ele permite que os dispositivos necessitem de menos capacidade computacional, podendo inclusive permanecer em modos de economia de bateria.

Ele faz isso se posicionando entre a camada física e a camada do servidor, assim fica responsável por receber, transmitir e processar as mensagens trocadas entre as duas camadas. Dessa forma, é possível escolher uma comunicação mais simples e efetiva para os dispositivos. No nosso sistema, foi identificado que essa comunicação seria através do protocolo ESP-NOW, como foi explicado na seção 3.2.2.

Tendo definido o protocolo de comunicação do gateway, a escolha de qual dispositivo utilizar como gateway ficou limitada ao ESP32 e ao ESP8266.

Além da comunicação com dispositivos, o gateway também precisa se comunicar com o nosso servidor. Uma grande facilidade seria a possibilidade de se comunicar com o servidor via HTTP, tendo em vista que os servidores web atualmente seguem a arquitetura de APIs REST. O ESP32 já possui um módulo Wi-Fi integrado, permitindo que nosso gateway troque mensagens HTTP de GET e POST com nosso servidor web.

4.1.4 Servidor web

O servidor web ficará hospedado em um computador. Para o protótipo implementado foi utilizado um computador pessoal, não necessitando de um sistema operacional específico e salvando os dados em um banco SQLite. Ele é responsável por fornecer uma interface web onde é possível cadastrar e remover dispositivos e visualizar um painel com os dispositivos cadastrados.

Além disso, para os dispositivos de porta, é possível visualizar o status atual e um histórico com os horários em que a porta foi aberta ou fechada. Para os dispositivos infravermelho é possível enviar um comando de ligar ou desligar o aparelho. Para o protótipo, a interface foi disponibilizada no próprio localhost na porta 5000. Na ULEG/FT o servidor deve ficar hospedado em um dos computadores da ala técnica.

4.2 DESENVOLVIMENTO DE FIRMWARE

O firmware é o software incorporado em dispositivos eletrônicos que controla seu funcionamento. No caso de um sistema IoT, o firmware é responsável por controlar o dispositivo físico que coleta e envia dados para o sistema.

Como o firmware é de suma importância neste trabalho, ele será desenvolvido e testado separadamente

Cada funcionalidade do firmware foi desenvolvida isolada do resto primeiro e apenas após seu funcionamento completo a funcionalidade foi integrada ao resto do sistema.

4.2.1 Dispositivo de porta

Foi desenvolvido o código para ler o módulo do Reed Switch, separadamente da parte previamente desenvolvida do ESP-NOW para que cada parte pudesse ser testada individualmente.

A leitura funciona da seguinte maneira: ao ter um ímã por perto, o módulo do Reed Switch fecha o circuito e passa a enviar um '1' lógico. Ao afastar o ímã, o circuito se abre e passa a enviar um '0' lógico.

Esperava-se que essa fosse uma funcionalidade fácil de ser implementada, porém, ao receber as leituras do módulo percebeu-se que o estado aberto não estava representando sempre um '0' lógico e sim uma variação entre o '1' e '0' lógicos.

Foi implementada via software então um algoritmo de eliminação de rebotes que faz uma série de leitura durante 1s. Se durante esse tempo tiver ocorrido pelo menos 2 mudanças de

estado, é considerado que o sistema está oscilando e portanto está em aberto, ou seja, a porta está aberta. Se durante esse tempo o estado permanecer sempre como '1' lógico é considerado que a porta está fechada.

Ao ocorrer uma alteração do estado de aberto para fechado ou vice-versa é enviada uma mensagem para o gateway sinalizando que a porta foi aberta ou fechada.

Em nosso protótipo, a mensagem enviada para o gateway consiste apenas em um char '0' (aberto) ou '1' (fechado) representando o novo estado da porta.

4.2.2 Dispositivo infravermelho

Para o desenvolvimento do firmware do dispositivo infravermelho temos a lógica de mensageria invertida. Ao invés de apenas enviar mensagens para o gateway, o dispositivo irá apenas receber comandos do gateway, seja o comando de ligar ou desligar o aparelho.

Novamente foi desenvolvido separadamente a funcionalidade de receber mensagens do gateway por ESP-NOW da funcionalidade de executar o comando de ligar ou desligar.

O ESP-NOW é configurado de forma que o gateway precise saber previamente qual o endereço MAC do dispositivo, só assim ele é capaz de enviar mensagens para o dispositivo infravermelho. É cadastrada uma função de callback para ser executada sempre que uma mensagem nova for recebida.

Na função de callback recebe-se o endereço MAC do remetente, o dado enviado e seu tamanho. Assim é possível verificar se a mensagem veio realmente do gateway e se seu tamanho é o esperado.

Em nosso protótipo a mensagem recebida pelo gateway consiste apenas em um char '0' (desligar) ou '1' (ligar) indicando qual comando deve ser executado.

4.2.3 Gateway

Nosso gateway possui uma série de funcionalidades e diferentes tipos de comunicação, o que torna ele o firmware mais complexo de ser desenvolvido.

É necessário que o gateway esteja disponível para o protocolo ESP-NOW e para o protocolo HTTP via Wi-fi. Além disso, em ambos os protocolos ele deve ser capaz de enviar e receber mensagens.

Existem 2 fluxos de mensagens no sistema. O primeiro fluxo consiste em uma ação de ligar ou desligar iniciado pelo usuário ao interagir com a interface gráfica do sistema web. Ao requisitar a ação em um dos dispositivos infravermelho que estará apontado para um aparelho de ar-condicionado, o servidor web irá enviar uma requisição POST para o gateway.

Essa requisição deve ser encaminhada para o caminho `"/turnOn"` e deverá conter dois parâmetros. O primeiro parâmetro é o comando a ser executado, `'0'` para desligar e `'1'` para ligar o aparelho. O segundo parâmetro é o endereço MAC do ESP32 do dispositivo para qual o gateway deverá repassar o comando.

Ao receber um comando diferente de `'0'` ou `'1'` o gateway responde com um status de 404 e não executa o comando. Se o comando recebido foi uma das possibilidades esperadas, ele então utiliza o MAC para enviar por ESP-NOW o comando ao dispositivo infravermelho devido.

A mensagem enviada para o dispositivo contém apenas o comando de ligar ou desligar. Ao ter sucesso no envio da mensagem por ESP-NOW, o gateway adiciona ao log que o envio foi um sucesso. Isso é feito através de uma função de callback executada quando a mensagem é enviada.

O segundo fluxo consiste no caminho oposto. Uma variação de estado em uma das portas monitoradas pelo dispositivo irá disparar uma mensagem via ESP-NOW do dispositivo da porta para o gateway, que por sua vez deve encaminhar a mensagem para o servidor web.

Inicialmente, foi desenvolvido utilizando uma função de callback para o envio da mensagem para o servidor web que é chamada ao receber alguma mensagem via ESP-NOW. A função de callback então manda uma requisição POST para o servidor web, no caminho `"/setDoorStatus"` e como dado um JSON contendo o endereço MAC do ESP32 do dispositivo da porta que foi alterado o estado e para qual estado ele foi, aberto ou fechado.

Essa abordagem se mostrou falha, o ESP32 não foi capaz de executar o envio da mensagem HTTP durante o callback de recebimento do ESP-NOW, muito provavelmente devido ao fato de que o ESP-NOW é um protocolo baseado no Wi-Fi, o qual é utilizado para o HTTP, é poderia estar causando uma disputa de recursos.

Para contornar o problema, foi implementado via software uma fila. Ao invés de enviar a mensagem diretamente pelo callback do ESP-NOW, foi alterado para que o callback apenas adicione na fila os dados recebidos, endereço MAC e comando. Além disso, foi implementado que o gateway cheque se foi adicionado algum item na fila de um em um segundo. Ao perceber que um novo dado foi adicionado, significando a mudança de estado de uma porta, o gateway então envia o POST para o servidor web.

Também foi necessário garantir que tanto o gateway quanto os dispositivos estão no mesmo canal Wi-Fi. Assim é possível configurar o ESP32 para que funcione tanto como uma Station quanto uma Soft Access Point, simultaneamente. Isso é vital para que se consiga utilizar o Wi-Fi e o ESP-NOW juntos.

4.2.4 Servidor web

O servidor web foi desenvolvido utilizando o framework Python chamado Flask. Ele é um framework simples que nos permite criar e customizar rapidamente tanto a visualização quanto os endpoints necessários.

Foram utilizados três blueprints no servidor, um para o dispositivo de porta, um para o dispositivo infravermelho e um como principal. Uma blueprint no Flask nada mais é que separar sua aplicação em componentes para que seja melhor modularizado e organizado.

A blueprint principal contém 3 endpoints. O primeiro, no caminho "/", possui apenas a opção de GET e é o padrão de um servidor web para representar a página inicial. Ela irá conter um painel de cards com as informações mais importantes de cada dispositivo cadastrado. Para isso é necessário fazer uma query no banco de dados que retorne todos os elementos de dispositivos.

O segundo, no caminho "/addDevice", possui a opção de GET e POST e foi desenvolvido para conter as lógicas relacionadas à página de cadastro do dispositivo. Por último, no caminho "/deleteDevice", temos um método POST responsável pela remoção do dispositivo da base de dados.

A segunda blueprint desenvolvida para o dispositivo infravermelho, possui os endpoints responsáveis por enviar o comando de ligar ou desligar e a tela do dispositivo. A terceira blueprint implementa os endpoints de visualização de histórico e para receber uma atualização do status da porta.

Para a persistência dos dispositivos no banco de dados foram criados 2 entidades. A primeira entidade, chamada de Device, possui o campo id, name, type, macAddress, createdAt e readings. O campo de id é utilizado como chave primária e é auto incrementado. Os campos name, type e macAddress serão preenchidos no cadastro do dispositivo. O campo de createdAt é preenchido com a data do momento na hora da criação da nova entrada.

O campo de readings representa uma relação de One to Many com a entidade Read, que representa uma leitura do dispositivo. A entidade Read foi criada com o intuito de guardar e persistir as leituras dos dispositivos, criando um histórico. Ela possui os campos de id, value, createdAt e deviceId. Ela guarda então o valor de uma leitura de um dispositivo, e através de sua data de criação.

Em nosso protótipo foi utilizado o SQLite como banco de dados. É uma forma rápida e fácil de conseguirmos implementar a persistência do sistema, porém, não é uma opção segura e performática e não deve ser usada em ambientes produtivos.

A Figura 4.10 apresenta como ficou o visual da página inicial. Note que em cada card temos o nome do dispositivo, sua numeração, seu MAC e seu tipo (infravermelho ou de porta). Além disso, para o dispositivo de porta temos sua leitura e um botão para visualizar

seu histórico. Para o dispositivo infravermelho temos os botões de Ligar ou Desligar. E para ambos temos ainda o botão de remover dispositivo que conta com uma Modal para confirmar a remoção do dispositivo, como pode ser visto na Figura 4.11.



Figura 4.10 – Página inicial da interface web

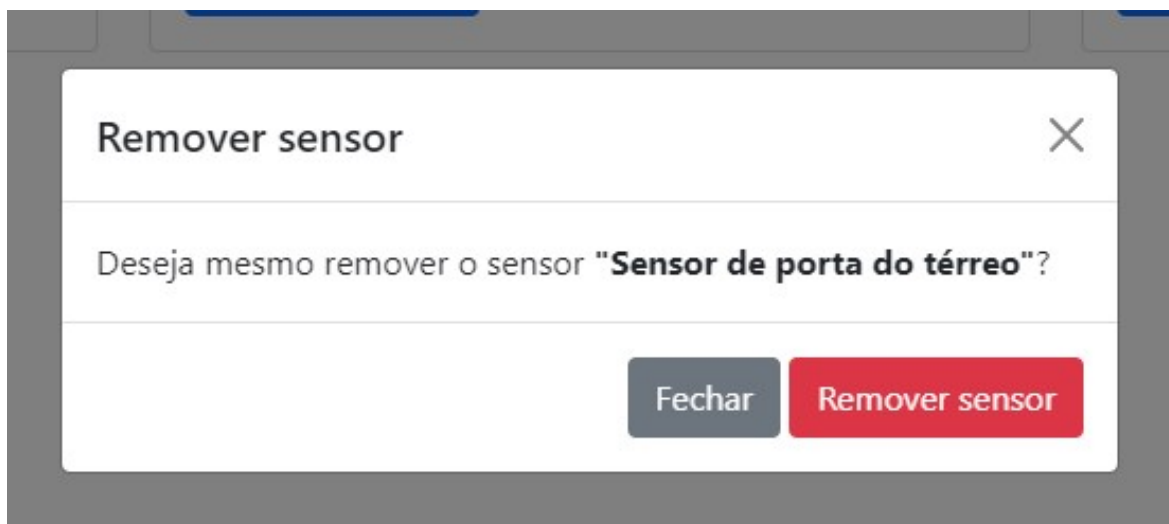


Figura 4.11 – Modal de confirmação para remover dispositivo

Para cadastramos um dispositivo, foi desenvolvida uma tela onde é necessário preencher o nome, tipo do dispositivo e endereço MAC. O nome do dispositivo é utilizado apenas para facilitar a identificação e organização dos dispositivos. Na Figura 4.12 é possível visualizar como ficou implementado.

Cadastre novos sensores

Preencha o formulário abaixo passando o tipo e endereço MAC corretos do sensor

Nome

Escolha um nome para identificar o sensor.

Tipo de sensor

Endereço MAC:

Formato: XX:XX:XX:XX:XX:XX com apenas números e letras maiúsculas

Figura 4.12 – Tela de cadastro do dispositivo

Ao clicar para acessar o histórico do dispositivo de porta, há um redirecionamento para a página específica daquele dispositivo de porta, como pode ser visto na Figura 4.14. Nessa página temos as identificações do dispositivo e o histórico de suas leituras, seu valor e sua Data e Hora. Isso é possível graças a relação que criamos das entidades Read e Device.

Histórico do sensor: Sensor de porta do térreo #8

Data e Hora	Status
2023-06-09 01:25:25	Fechada
2023-06-08 18:26:02	Aberta
2023-06-08 18:25:55	Fechada
2023-06-08 18:25:39	Aberta
2023-06-08 18:25:24	Fechada
2023-06-08 18:25:10	Aberta
2023-06-08 18:24:51	Fechada
2023-06-08 18:24:50	Aberta
2023-06-08 18:24:48	Fechada
2023-06-08 18:24:46	Aberta
2023-06-08 18:24:45	Fechada
2023-06-08 18:13:34	Aberta
2023-06-08 18:13:33	Fechada

Figura 4.13 – Tela de histórico do dispositivo de porta

Por fim, ao clicar nos botões de ligar ou desligar de um dispositivo infravermelho, a ação é executada e também há um redirecionamento para a página específica daquele dispositivo, onde será possível enviar novamente os comandos de Ligar ou Desligar.

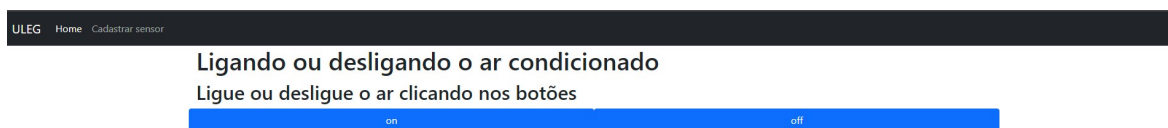


Figura 4.14 – Tela do dispositivo infravermelho

4.2.5 Repositório do firmware

O desenvolvimento do firmware foi feito utilizando a ferramenta de versionamento Git. Através do repositório público: Monitoramento ULEG. Nele é possível ter acesso aos códigos desenvolvidos.

4.3 TESTE DE INTEGRAÇÃO

Nesta seção, serão descritos os procedimentos e objetivos do teste de integração realizado no sistema IoT. O teste de integração tem como objetivo verificar o fluxo completo de comunicação entre os componentes do sistema, incluindo o dispositivo de porta, o dispositivo infravermelho, o gateway e o servidor web.

O teste foi realizado localmente utilizando um computador pessoal para hospedar o servidor web e alimentar os microcontroladores. O ar-condicionado presente no teste é da fabricante LG.

4.3.1 Componentes do sistema IoT

Os principais componentes que foram integrados durante o teste de integração incluem o dispositivo de porta, o dispositivo infravermelho, o gateway e o servidor web. Vale ressaltar que, devido à disponibilidade de apenas dois microcontroladores ESP32, cada dispositivo foi testado individualmente no sistema.

4.3.2 Objetivos do teste de integração

Os objetivos específicos do teste de integração foram:

- Verificar o fluxo completo de comunicação entre os componentes do sistema.
- Testar a funcionalidade de ligar e desligar o aparelho de ar-condicionado através da interação com a interface gráfica do sistema web.
- Validar o envio e recebimento de mensagens entre o dispositivo de porta e o gateway, assim como a transmissão dessas informações para o servidor web.

4.3.3 Ações e eventos simulados

Durante o teste de integração, foram simulados dois fluxos de ações:

1. Uma ação de ligar ou desligar o ar-condicionado foi iniciada pelo usuário através da interface gráfica do sistema web. O servidor web enviou uma requisição POST para o gateway, que, por sua vez, enviou o comando de ligar ou desligar para o dispositivo infravermelho via ESP-NOW.
2. Uma variação de estado em uma das portas monitoradas pelo dispositivo de porta disparou uma mensagem via ESP-NOW do dispositivo de porta para o gateway, que encaminhou essa mensagem para o servidor web.

4.3.4 Dados utilizados no teste

Durante o teste de integração, foram utilizados os dados cadastrados dos dois dispositivos, incluindo seus endereços MAC e tipos (infravermelho ou porta), armazenados no servidor web. Esses dados foram utilizados para configurar a comunicação entre os componentes do sistema.

4.3.5 Configurações e conexões físicas

Para realizar o teste de integração, foram necessárias as seguintes configurações e conexões físicas:

- O dispositivo infravermelho foi conectado ao LED infravermelho através de um resistor.
- O dispositivo de porta foi conectado ao módulo Reed Switch para detectar a abertura e fechamento da porta.

4.3.6 Procedimentos específicos seguidos

Os seguintes passos foram seguidos para executar o teste de integração:

1. Cadastramento do dispositivo infravermelho no servidor web
2. Teste da funcionalidade de ligar o ar-condicionado através da interface gráfica do sistema web
3. Teste da funcionalidade de desligar o ar-condicionado através da interface gráfica do sistema web
4. Cadastramento do dispositivo de porta no servidor web
5. Teste de abertura de uma porta simulada com um ímã próximo ao módulo Reed Switch

6. Teste de fechamento de uma porta simulada com um ímã próximo ao módulo Reed Switch
7. Avaliação da mudança no histórico de registros

4.3.7 Resultados esperados

Os resultados esperados do teste de integração são:

- Para o dispositivo infravermelho: verificar a capacidade de ligar e desligar o aparelho de ar-condicionado através da interação com a interface gráfica do sistema web
- Para o dispositivo de porta: verificar se o histórico de registros exhibe corretamente as variações de estado da porta, incluindo data e hora das mudanças.

4.3.8 Métricas utilizadas para avaliar o sucesso

As métricas utilizadas para avaliar o sucesso do teste de integração incluíram:

- Cadastro persistente dos dispositivos no servidor web atualizando o quadro de dispositivos
- Comunicação correta entre o servidor web e o gateway via requisições HTTP
- Recebimento correto das mensagens pelo gateway e transmissão para o dispositivo infravermelho via ESP-NOW
- Transmissão correta das mensagens pelo gateway para o servidor web após recebimento do dispositivo de porta via ESP-NOW
- Atualização adequada dos registros no servidor web, refletindo as ações e eventos simulados durante o teste.

4.4 IMPLEMENTAÇÃO TESTE NA ULEG

Nesta seção, será apresentada a implementação do sistema IoT no laboratório de técnicas digitais da ULEG/FT, utilizando um ar-condicionado da Hitachi. O objetivo da implementação foi testar o sistema em um ambiente controlado e verificar sua funcionalidade em uma situação real.

O experimento foi categorizado como uma implementação teste porque apesar de possuir os componentes finais e estar presente no ambiente o qual o sistema será implementado, não foi uma implementação completa.

Não foi simulado a alimentação dos microcontroladores por baterias ou tomadas do laboratório, nem a hospedagem do servidor web no próprio servidor da ULEG/FT, nem outras conexões e instalações que devem ser feitas em uma implementação real.

4.4.1 Configuração do ambiente de implementação

O sistema foi implementado no laboratório da ULEG/FT, onde foram disponibilizadas as condições necessárias para a instalação dos componentes do sistema IoT. O ar-condicionado utilizado foi da marca Hitachi, a qual é a fabricante do sistema instalado no prédio da ULEG/FT.

Foi feita a replicação do esquema do teste de integração: O esquema utilizado no teste de integração foi replicado no laboratório da ULEG/FT. Isso incluiu a instalação do dispositivo de porta, do dispositivo infravermelho e do gateway, bem como a configuração do servidor web para receber e exibir os dados.

4.4.2 Instalação do dispositivo de porta

O dispositivo de porta foi instalado em uma porta do laboratório, de forma a monitorar seu estado (aberta ou fechada). O dispositivo foi conectado ao microcontrolador ESP32 e configurado para enviar as informações de status para o gateway.

4.4.3 Instalação do dispositivo infravermelho

O dispositivo infravermelho foi instalado de maneira a apontar para o ar-condicionado da Hitachi. Ele foi conectado ao microcontrolador ESP32 e configurado para receber comandos de ligar e desligar do servidor web, por meio do gateway.

4.4.4 Configuração do gateway

O gateway, também baseado no microcontrolador ESP32, foi configurado para receber os dados dos dispositivos de porta e enviar as informações para o servidor web. Além disso, ele foi configurado para receber comandos do servidor web e repassá-los ao dispositivo infravermelho.

4.4.5 Configuração do servidor web

O servidor web foi configurado para receber os dados dos dispositivos de porta, exibir o histórico de registros e permitir ao usuário interagir com o sistema, enviando comandos de ligar e desligar o ar-condicionado.

5 RESULTADOS

5.1 RESULTADOS

O objetivo deste capítulo é fornecer uma análise objetiva e detalhada dos resultados obtidos dos testes e implementações descritos no capítulo de Metodologia, com o intuito de avaliar o desempenho, a funcionalidade e a eficácia do sistema IoT proposto. Por meio desses resultados, é possível verificar se o sistema atende aos requisitos estabelecidos, identificar possíveis limitações e apresentar conclusões significativas para o avanço do conhecimento na área.

Neste capítulo, serão apresentados e discutidos os resultados do teste de integração, que verificou a comunicação entre os componentes do sistema IoT, bem como a implementação em ambiente de teste na ULEG/FT, utilizando um ar-condicionado da Hitachi. A análise dos resultados permitirá uma compreensão mais aprofundada do sistema IoT proposto, suas capacidades, limitações e possíveis melhorias.

5.1.1 Resultados do teste de integração

Nesta seção, apresentamos os resultados obtidos durante o teste de integração do sistema IoT proposto. O objetivo desse teste foi avaliar a correta comunicação e interação entre os componentes do sistema, incluindo o dispositivo de porta, o dispositivo infravermelho, o gateway e o servidor web.

Durante o teste de integração, foram realizados diferentes cenários e fluxos de ações para verificar a funcionalidade e o desempenho do sistema como um todo. Essa etapa foi essencial para validar a integração e garantir que todos os componentes trabalhassem em conjunto de maneira eficiente.

Neste contexto, focamos em duas principais áreas de teste. Primeiramente, avaliamos o fluxo de comunicação do dispositivo de porta, verificando sua capacidade de registrar corretamente o estado de abertura e fechamento da porta no servidor web. Em seguida, testamos o fluxo de comunicação do dispositivo infravermelho, onde buscamos validar a funcionalidade de ligar e desligar o ar-condicionado por meio da interface web, enviando comandos do servidor para o dispositivo.

Por meio desses testes, foi possível obter dados e informações relevantes para avaliar a eficácia e a confiabilidade do sistema. Esses resultados são essenciais para compreender a viabilidade do sistema IoT proposto e verificar se ele atende às expectativas e requisitos estabelecidos.

A seguir, serão apresentados os resultados detalhados de cada fluxo de comunicação, discutindo as observações, os padrões identificados e as conclusões obtidas durante o teste de integração. Esses resultados fornecem uma visão aprofundada do desempenho e da funcionalidade do sistema IoT, além de servir como base para as conclusões gerais deste trabalho.

5.1.1.1 Fluxo de comunicação do dispositivo de porta

Para testar o fluxo de comunicação do dispositivo de porta, utilizou-se de um ímã simples com o módulo Reed Switch conectado à uma protoboard.

Ao aproximar o ímã do módulo, observou-se que o sistema reconheceu a mudança de estado e então enviou uma mensagem via ESP-NOW do dispositivo para o gateway. Analisando os logs do gateway foi possível reconhecer que ele recebeu com sucesso a mensagem, identificando o endereço MAC do microcontrolador ESP32 e o comando de porta fechada.

O gateway por sua vez adicionou em sua fila interna os parâmetros recebidos e rapidamente consumiu a fila para enviar um POST via HTTP para o servidor web, contendo o endereço MAC do microcontrolador e o comando de que a porta havia sido fechada.

Por último, o servidor web recebeu também com sucesso a mensagem do gateway, retornando um status de 200 OK. O servidor então criou uma nova leitura para aquele dispositivo, já que ele estava cadastrado no sistema, com o estado de porta fechada.

Ao acessar a página inicial pode-se notar que o status atual da porta era fechada e em seu histórico aparecia a data e hora corretos da variação executada.

O mesmo procedimento foi feito agora afastando o ímã do módulo do Reed Switch. O resultado também foi um sucesso. Alguns testes adicionais foram feitos, variando várias vezes a posição do ímã e percebeu-se que o sistema se comportou com perfeição. Seu histórico ficou correto e sempre um status de Aberto era seguido de um Fechado e vice-versa.

Concluiu-se então que o fluxo de comunicação do dispositivo de porta foi um sucesso.

5.1.1.2 Fluxo de comunicação do dispositivo infravermelho

Para testar o fluxo de comunicação do dispositivo de porta, utilizou-se de um LED infravermelho apontado para um aparelho de ar-condicionado da fabricante LG.

O teste realizado foi de enviar, a partir do servidor web, comandos de ligar e desligar para o aparelho. Ao clicar em um dos botões, o servidor web disparou um POST via HTTP para o gateway, contendo em seu corpo o comando de ligar ou desligar e o endereço MAC do dispositivo a executar o comando.

O gateway recebeu com sucesso a mensagem do servidor web, capturando corretamente

os dados do comando e do endereço. Em seguida, mandou uma mensagem via ESP-NOW para o dispositivo que possuía o MAC recebido com o comando.

O dispositivo infravermelho recebeu com sucesso o comando e então enviou para o LED o comando que deveria ser executado. O LED por sua vez enviou o comando nos tempos e frequência correta, resultando na mudança de estado do aparelho de ar-condicionado.

O teste foi feito algumas vezes desligando e ligando o aparelho e ao apontar o LED corretamente todas obtiveram êxito. Concluiu-se então que o fluxo de comunicação do dispositivo infravermelho foi um sucesso.

5.1.2 Resultados da implementação teste na ULEG

Após obter sucesso no teste de integração, o sistema estava pronto para ser testado no ambiente real, em um laboratório da ULEG/FT. A metodologia utilizada para testar o sistema, apresentada no Capítulo 4, define que os mesmos passos executados no teste de integração serão executados na implementação.

Isso significa principalmente que os 2 fluxos de comunicação serão testados individualmente. A grande diferença da implementação na ULEG/FT será a fabricante do ar-condicionado a ser testado. Enquanto tínhamos um ar-condicionado LG no teste de integração, uma fabricante com protocolos mais simples e conhecidos, agora teremos um ar-condicionado da Hitachi, uma fabricante conhecida por protocolos mais complexos e variados.

5.1.2.1 Fluxo de comunicação do dispositivo de porta

O fluxo de comunicação do dispositivo de porta foi o primeiro a ser testado e foi realizado em um ambiente mais realista para simular situações práticas. Para isso, o microcontrolador ESP32 com o sensor Reed Switch foi fixado na parede o mais perto da porta e um ímã colado na porta. Essa configuração permitiu que o dispositivo detectasse a abertura e o fechamento da porta com precisão.

Durante os testes, um desafio significativo foi encontrar a posição ideal para o ímã e o microcontrolador, garantindo que uma pequena abertura da porta fosse suficiente para que o sensor Reed Switch reconhecesse a variação. Ao mesmo tempo, era necessário garantir que, quando a porta estivesse completamente fechada, o ímã ainda mantivesse o impacto necessário para fechar o circuito do Reed Switch.

Após ajustes finos na posição do ímã e do microcontrolador, o sistema de detecção da porta funcionou conforme o esperado. O dispositivo foi capaz de atualizar corretamente o estado da porta, refletindo com precisão no servidor web. Além disso, o histórico de registros

no servidor web foi mantido de forma consistente, registrando todas as variações de abertura e fechamento da porta.

Esses resultados indicam que o dispositivo de porta atendeu aos requisitos e demonstrou ser uma solução confiável para monitorar o estado da porta. A integração bem-sucedida desse componente no sistema IoT destaca a eficácia da arquitetura proposta e sua capacidade de fornecer informações em tempo real sobre a porta monitorada.

5.1.2.2 Fluxo de comunicação do dispositivo infravermelho

Para o fluxo de comunicação do dispositivo infravermelho tivemos mais desafios devido à fabricante do aparelho de ar-condicionado. O primeiro teste que consistia em gravar e replicar o comando de ligar do aparelho utilizando o decodificador infravermelho foi bem sucedido.

Inicialmente foi gravado o comando apontando o controle do aparelho para o decodificador, que por sua vez estava conectado ao canal 0 do analisador lógico. Foi utilizado um gatilho para capturar o momento exato do início do comando. A Figura 5.1 representa um exemplo da gravação do sinal. No mesmo software de gravação é possível exportar os dados.

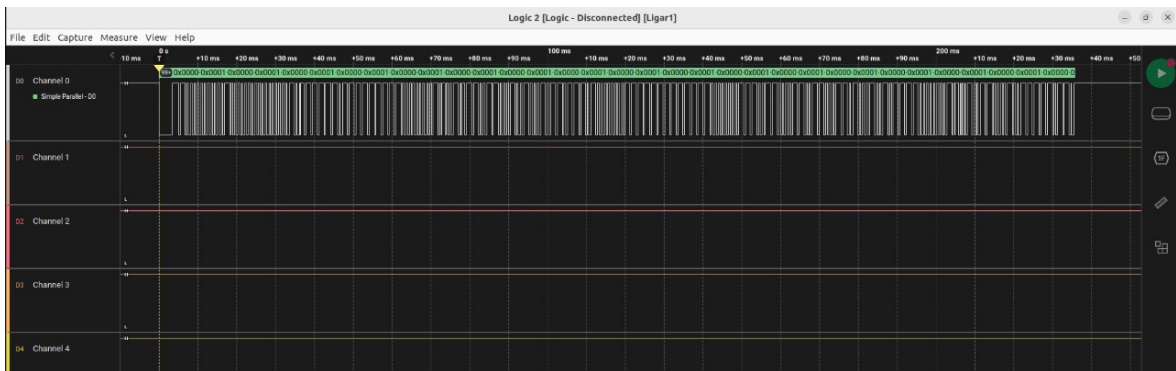


Figura 5.1 – Gravação através do analisador lógico de um comando de ligar do controle do ar-condicionado

Ao exportar os dados do analisador lógico e inseri-los em uma planilha, foi possível identificar uma padrão para as medidas de tempo. Nas Figuras 5.2 e 5.3 é possível ver os dados exportados e um padrão.

	A	B	C	D	E	F	G	H	I	J
1	name	type	start_time	duration	data	Bit	Byte	Error		Microseconds
2	Simple Parallel	data	-2.69118061e-16	0.003353375	0x0000000000000000		1		0	3353.375
3	Simple Parallel	data	0.0033535	0.001739125	0x0000000000000001		1		0	1739.125
4	Simple Parallel	data	0.00509275	0.000439375	0x0000000000000000		8		0	439.375
5	Simple Parallel	data	0.00553225	0.001267875	0x0000000000000001		1		0	1267.875
6	Simple Parallel	data	0.00680025	0.000437375	0x0000000000000000				0	437.375
7	Simple Parallel	data	0.00723775	0.000432375	0x0000000000000001		0		0	432.375
8	Simple Parallel	data	0.00767025	0.000462375	0x0000000000000000				0	462.375
9	Simple Parallel	data	0.00813275	0.000403625	0x0000000000000001		0		0	403.625
10	Simple Parallel	data	0.0085365	0.00046425	0x0000000000000000				0	464.25
11	Simple Parallel	data	0.00900875	0.00042725	0x0000000000000001		0		0	427.25
12	Simple Parallel	data	0.00942825	0.00041275	0x0000000000000000		0		0	412.75
13	Simple Parallel	data	0.009841125	0.00045425	0x0000000000000001		0		0	454.25
14	Simple Parallel	data	0.0102955	0.0004645	0x0000000000000000				0	464.5
15	Simple Parallel	data	0.010760125	0.0004035	0x0000000000000001		0		0	403.5
16	Simple Parallel	data	0.01116375	0.000462625	0x0000000000000000				0	462.625
17	Simple Parallel	data	0.0116265	0.0004055	0x0000000000000001		0		0	405.5
18	Simple Parallel	data	0.012032125	0.000488125	0x0000000000000000				0	488.125
19	Simple Parallel	data	0.012520375	0.000403375	0x0000000000000001		0		0	403.375
20	Simple Parallel	data	0.012923875	0.0004625	0x0000000000000000		0		0	462.5
21	Simple Parallel	data	0.0133865	0.000402375	0x0000000000000001		0		0	402.375
22	Simple Parallel	data	0.013789	0.00046475	0x0000000000000000				0	464.75
23	Simple Parallel	data	0.014253875	0.000428375	0x0000000000000001		0		0	428.375
24	Simple Parallel	data	0.014682375	0.0004115	0x0000000000000000				0	411.5
25	Simple Parallel	data	0.015094	0.000454875	0x0000000000000001		0		0	454.875
26	Simple Parallel	data	0.015549	0.00046475	0x0000000000000000				0	464.75
27	Simple Parallel	data	0.016013875	0.000402375	0x0000000000000001		0		0	402.375
28	Simple Parallel	data	0.016416375	0.000464	0x0000000000000000		8		0	464
29	Simple Parallel	data	0.0168805	0.001293375	0x0000000000000001		1		0	1293.375
30	Simple Parallel	data	0.018174	0.000387	0x0000000000000000				0	387
31	Simple Parallel	data	0.018561125	0.000455625	0x0000000000000001		0		0	455.625
32	Simple Parallel	data	0.019016875	0.000464	0x0000000000000000				0	464
33	Simple Parallel	data	0.019481	0.000403	0x0000000000000001		0		0	403
34	Simple Parallel	data	0.019884125	0.00043825	0x0000000000000000				0	438.25

Figura 5.2 – Planilha de dados exportados do comando de ligar

	A	B	C	D	E	F	G	H	I	J	K
1	name	type	start_time	duration	data	Bit	Byte	Error		Microseconds	
2	Simple Parallel	data	1.95E-16	0.003378	0x0000000000000000		1		0	3378	67.56
3	Simple Parallel	data	0.003378125	0.00171475	0x0000000000000001		1		0	1714.75	34.295
4	Simple Parallel	data	0.005093	0.00041575	0x0000000000000000		8		0	415.75	8.315
5	Simple Parallel	data	0.005508875	0.001291875	0x0000000000000001		1		0	1291.875	25.8375
6	Simple Parallel	data	0.006800875	0.0004405	0x0000000000000000				0	440.5	8.81
7	Simple Parallel	data	0.0072415	0.000429625	0x0000000000000001		0		0	429.625	8.5925
8	Simple Parallel	data	0.00767125	0.0004615	0x0000000000000000				0	461.5	9.23
9	Simple Parallel	data	0.008132875	0.00040325	0x0000000000000001		0		0	403.25	8.065
10	Simple Parallel	data	0.00853625	0.00046375	0x0000000000000000				0	463.75	9.275
11	Simple Parallel	data	0.009000125	0.00040275	0x0000000000000001		0		0	402.75	8.055
12	Simple Parallel	data	0.009403	0.000465	0x0000000000000000		0		0	465	9.3
13	Simple Parallel	data	0.009868125	0.00042875	0x0000000000000001		0		0	428.75	8.575
14	Simple Parallel	data	0.010297	0.000438375	0x0000000000000000				0	438.375	8.7675
15	Simple Parallel	data	0.0107355	0.000427875	0x0000000000000001		0		0	427.875	8.5575
16	Simple Parallel	data	0.0111635	0.000462875	0x0000000000000000				0	462.875	9.2575
17	Simple Parallel	data	0.0116265	0.00040375	0x0000000000000001		0		0	403.75	8.075
18	Simple Parallel	data	0.012030375	0.000437625	0x0000000000000000				0	437.625	8.7525
19	Simple Parallel	data	0.012468125	0.000455625	0x0000000000000001		0		0	455.625	9.1125
20	Simple Parallel	data	0.012923875	0.0004625	0x0000000000000000		0		0	462.5	9.25
21	Simple Parallel	data	0.0133865	0.0004025	0x0000000000000001		0		0	402.5	8.05
22	Simple Parallel	data	0.013789125	0.0004645	0x0000000000000000				0	464.5	9.29
23	Simple Parallel	data	0.01425375	0.000402875	0x0000000000000001		0		0	402.875	8.0575
24	Simple Parallel	data	0.01465675	0.0004645	0x0000000000000000				0	464.5	9.29
25	Simple Parallel	data	0.015121375	0.00043025	0x0000000000000001		0		0	430.25	8.605
26	Simple Parallel	data	0.01555175	0.00043575	0x0000000000000000				0	435.75	8.715
27	Simple Parallel	data	0.015987625	0.00042975	0x0000000000000001		0		0	429.75	8.595
28	Simple Parallel	data	0.0164175	0.000462625	0x0000000000000000		8		0	462.625	9.2525
29	Simple Parallel	data	0.01688025	0.00126725	0x0000000000000001		1		0	1267.25	25.345
30	Simple Parallel	data	0.018147625	0.000415875	0x0000000000000000				0	415.875	8.3175
31	Simple Parallel	data	0.018563625	0.000456625	0x0000000000000001		0		0	456.625	9.1325
32	Simple Parallel	data	0.019020375	0.000459875	0x0000000000000000				0	459.875	9.1975
33	Simple Parallel	data	0.019480375	0.000404625	0x0000000000000001		0		0	404.625	8.0925
34	Simple Parallel	data	0.019885125	0.000461875	0x0000000000000000				0	461.875	9.2375

Figura 5.3 – Planilha de dados exportados do comando de desligar

Após uma série de gravações de comandos tanto de ligar quanto de desligar, chegou-se ao seguinte padrão para as medidas de tempo do protocolo da Hitachi:

- Header mark - 3300 μs
- Header space - 1700 μs
- Bit mark - 400 μs
- One space - 1250 μs
- Zero space - 450 μs

O comando sempre se inicia com um mark de 3300 μs e um space de 1700 μs . Após isso, Um '0' lógico é representado por um bit mark de 400 μs seguido de um space de 450 μs , enquanto o '1' lógico é representado pelo mesmo bit mark de 400 μs seguido de um space de 450 μs .

Note que o protocolo da Hitachi segue o mesmo padrão do protocolo NEC explicado na Seção 4.1.2, mudando apenas os valores de tempo para cada item.

Ao exportar os dados do analisador lógico não obtemos diretamente o comando que ele representa. Como pode ser visto pelas Figuras 5.1 e 5.1, os dados exportados representam apenas as colunas A-E, as outras colunas foram criadas para conseguirmos identificar o comando.

O modo que os dados foram exportados traz uma sequência de marcações de tempo, que representam a duração de um mark ou space. Para tratar esses dados foi criada primeiro a coluna G, nela é avaliado se o tempo de space foi maior que 1000 μs , se sim caracteriza um One space e se não caracteriza um Zero space. Perceba que ela é executada de duas em duas linhas, isso se dá pois temos sempre dois tempos, o tempo de mark que é sempre 400 μs e o tempo de space, o qual define se o bit é um '0' ou '1'.

A coluna H analisa de 4 em 4 bits, colocando seus devidos pesos binários, para se obter o nibble correspondente.

A coluna I foi utilizada como uma validação de glitches, que são falhas ou interrupções temporárias do sistema. Estava sendo observado a presença de dados com duração inferior a 50 μs , eles foram considerados como falhas. Ao se deparar com um dado com duração inferior a 50 μs , a linha é marcada com um 1 na coluna de Error, facilitando assim a remoção dos glitches.

Por fim, na coluna J temos as durações em micro segundos. Para obter o comando foi feito uma concatenação de todos os bytes apresentados na coluna H da planilha. As 3 melhores gravações do comando de ligar resultaram nos seguintes comandos:

- 80080002FDFF006798916E807FFC03**F**8076897807F09F**67E**
- 80080002FDFF006798916E807FFC03**9**8676897807F09F**60**
- 80080002FDFF006798916E807FFC03**9**8676897807F09F**67F**

Enquanto as duas melhores gravações do comando de desligar resultaram nos seguintes comandos:

- 80080002FDFF006798916E807FFC03**9**8676897807F05FA**078**
- 80080002FDFF006798916E807FFC03**9**8676897807F05FA**78**

É possível notar a grande semelhança entre os comandos. Inicialmente, nota-se que no início de todo o comando há uma repetição. Os protocolos de infravermelho estudados possuem sempre uma espécie de endereço no início do sinal, por isso, é possível concluir que esse início do comando gerado é o endereço do protocolo da Hitachi.

Estes comandos foram então inseridos no código do dispositivo infravermelho para representar o comando de ligar e desligar que seriam enviados pelo nosso sistema para o aparelho da Hitachi.

Ainda sem testar diretamente no ar-condicionado, o LED infravermelho conectado ao microcontrolador foi apontado para o decodificador. Então foi feito o mesmo processo de captura, substituindo o controle do aparelho pelo LED, afim de verificar se o comando gravado e após o tratamento seria o mesmo.

A Figura 5.4 evidencia a captura do comando gerado através do LED infravermelho. É possível notar a grande similaridade com a Figura 5.1. Ao exportar os dados, inseri-los na planilha e ter o código foi possível comparar com o código criado pelo controle do aparelho. Em todas as vezes o resultado foi satisfatório, ou seja, o comando gravado estava igual ao comando gerado.

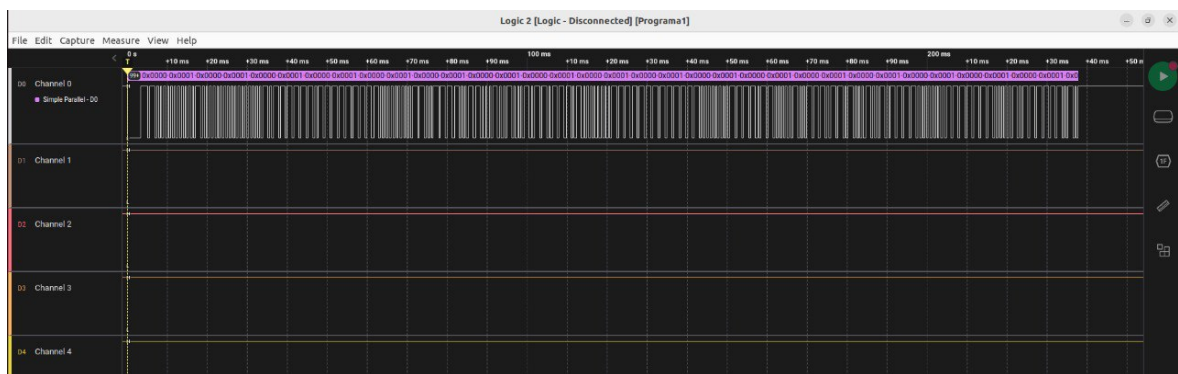


Figura 5.4 – Gravação através do analisador lógico de um comando de ligar do LED infravermelho

Agora o LED foi apontado diretamente para o ar-condicionado da Hitachi. Ao enviar o comando de ligar pelo servidor web, porém, não foi obtido êxito. Diante da falha de ligar o ar-condicionado diversas variações foram executadas, algumas foram:

- Testar todos os comandos que foram gravados de ligar
- Testar todos os comandos que foram gravados de desligar
- Aproximar mais o LED do aparelho
- Trocar o resistor por um de resistência menor
- Gravar um novo comando e imediatamente enviá-lo

Tentou-se também usar os tempos exatos que tinham sido gravados, ao invés dos tempos apresentados acima como o padrão que se observou, já que o software desenvolvido possibilitava essa opção.

Após diversas tentativas diversas, chegou-se à conclusão que não era possível obter êxito. Outro detalhe importante de se notar é que em nenhuma gravação foi possível identificar dois comandos iguais, o que leva a crer que o protocolo Hitachi possui mais funcionalidades de segurança que variam o comando, tornando sua replicação muito mais difícil.

6 CONCLUSÃO

6.1 CONCLUSÃO

Este capítulo apresenta as conclusões do trabalho, com base nos resultados obtidos e nas experiências durante a implementação e teste do sistema IoT proposto.

6.1.1 Funcionalidade limitada do dispositivo infravermelho com o ar-condicionado da Hitachi

Durante a implementação, foi constatado que o dispositivo infravermelho não conseguiu se comunicar adequadamente com o ar-condicionado da marca Hitachi. Isso limitou a capacidade do sistema de controlar remotamente o aparelho de ar-condicionado. Trata-se de uma área que requer mais estudos e ajustes específicos para compatibilidade com diferentes marcas e modelos de ar-condicionado.

6.1.2 Sucesso do dispositivo de porta

O dispositivo de porta apresentou resultados positivos durante o teste de integração e a implementação no ambiente da ULEG/FT. Foi possível monitorar com precisão o estado de abertura e fechamento da porta, registrando as informações corretamente no servidor web. Isso demonstra que o dispositivo de porta atendeu aos requisitos e foi eficaz em sua funcionalidade.

6.1.3 Arquitetura funcional e adequada

Apesar dos desafios e limitações encontrados, a arquitetura do sistema IoT proposto se mostrou funcional e adequada. A integração entre os componentes, como o dispositivo de porta, o dispositivo infravermelho, o gateway e o servidor web, foi efetiva na troca de dados e no controle de dispositivos. A arquitetura forneceu uma base sólida para o desenvolvimento do sistema IoT e evidenciou seu potencial em ambientes controlados.

6.1.4 Conclusão final

Em suma, este trabalho destacou os pontos fortes e as limitações do sistema IoT proposto. Embora algumas funcionalidades não tenham sido implementadas, como a compatibilidade com determinado ar-condicionado, os modos de economia de bateria nos microcontroladores e o gerenciamento dos computadores da ULEG/FT, os resultados alcançados com o dispo-

tivo de porta e a arquitetura geral são promissores. Esses pontos não abordados podem ser explorados em trabalhos futuros, visando aprimorar o sistema e estender sua funcionalidade em diferentes cenários. O trabalho atual forneceu uma base sólida e valiosas contribuições para a área de sistemas IoT, destacando a importância de considerar desafios e oportunidades para melhorias contínuas nesse campo em constante evolução.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 SETHI, P.; SARANGI, S. R. Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, Hindawi, v. 2017, 2017.
- 2 LEITE, L. et al. A survey of devops concepts and challenges. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 52, n. 6, p. 1–35, 2019.
- 3 VENKATRAMAN, S.; VARSHAA, R.; VIGNESHWARY, P. Iot based door open or close monitoring for home security with emergency notification system using lora technology. In: IEEE. *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*. [S.l.], 2021. v. 1, p. 173–178.
- 4 LUNA, J. I. V.; SÁNCHEZ-RANGEL, F. J.; COSME-ACEVES, J. F. Monitoring system for doors and windows of a data center with iot. *Ingenius*, Universidad Politécnica Salesiana del Ecuador, n. 22, p. 72, 2019.
- 5 ADRIANO, D. B.; BUDI, W. A. C. et al. Iot-based integrated home security and monitoring system. In: IOP PUBLISHING. *Journal of physics: conference series*. [S.l.], 2018. v. 1140, n. 1, p. 012006.
- 6 ALDAWIRA, C. R. et al. Door security system for home monitoring based on esp32. *Procedia Computer Science*, Elsevier, v. 157, p. 673–682, 2019.
- 7 SANTOS, M. D. d. A. Utilizando internet das coisas (iot) para controle de equipamentos de ar-condicionado: soluções e desafios tecnológicos. 2021.
- 8 CUNHA, J. R. L. D. *Monitoramento de ambientes especiais aliado ao conceito de internet das coisas (IoT)*. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2018.
- 9 VANDYKE, M. *Magnetic Switch Selection: Reed Switch vs. Hall Effect Switch*. 2019. <<https://www.magnelinkinc.com/blog/reed-vs-hall-effect-switch/>>. Accessed: 2023-02-02.
- 10 TIDA-01066 - Low-Power Door and Window Sensor With Sub-1GHz and 10-Year Coin Cell Battery Life Reference Design. <<https://www.ti.com/tool/TIDA-01066>>. Accessed: 2023-02-05.
- 11 SANTOS, R. *ESP-NOW Two-Way Communication Between ESP32 Boards*. 2020. <<https://randomnerdtutorials.com/esp-now-two-way-communication-esp32/>>. Accessed: 2023-03-06.
- 12 KOYANAGI, F. *ESP-NOW: WiFi With 3x More Range*. 2019. <<https://www.instructables.com/ESP-NOW-WiFi-With-3x-More-Range/>>. Accessed: 2023-03-06.
- 13 EICHHORN, D. *ESP32 – Ultra-Long Battery Life With ESP-NOW*. 2021. <<https://thingpulse.com/esp32-ultra-long-battery-life-with-espnow/>>. Accessed: 2023-02-05.
- 14 INSIGHT Into ESP32 Sleep Modes Their Power Consumption. <<https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>>. Accessed: 2023-03-06.

- 15 GUPTA, S. *ESP32 Active Mode and Deep Sleep Mode Power Consumption Comparison*. 2021. <<https://circuitdigest.com/microcontroller-projects/esp32-active-mode-and-deep-sleep-mode-power-consumption>>. Accessed: 2023-03-06.
- 16 EPULSE Feather – Low Power ESP32 development board. <<https://thingpulse.com/product/epulse-feather-low-power-esp32-development-board/>>. Accessed: 2023-03-06.
- 17 ARORA, S. *Ansible vs. Puppet: The Key Differences to Know*. 2023. <<https://www.simplilearn.com/ansible-vs-puppet-the-key-differences-to-know-article>>. Accessed: 2023-03-06.
- 18 ERANDA, K. *Ansible for edge remote Linux IoT devices*. 2021. <<https://jfrog.com/connect/post/ansible-for-edge-remote-linux-iot-devices/>>. Accessed: 2023-03-06.
- 19 SAMPATH, B. *Run Ansible Playbook Using AWX GUI*. 2021. <<https://medium.com/ankercloud-engineering/run-ansible-playbook-using-awx-gui-416a3696a891>>. Accessed: 2023-03-06.
- 20 PUPPET device. <https://www.puppet.com/docs/puppet/5.5/puppet_device.html>. Accessed: 2023-03-06.