

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
ENGENHARIA DE COMPUTAÇÃO

GUSTAVO BERTOLDI DE MELLO

**CONTROLE REMOTO INFRAVERMELHO UNIVERSAL WIRELESS UTILIZANDO
O ESP8266**

Porto Alegre

2024

GUSTAVO BERTOLDI DE MELLO

**CONTROLE REMOTO INFRAVERMELHO UNIVERSAL WIRELESS UTILIZANDO
O ESP8266**

Trabalho de conclusão de curso de graduação apresentado na Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador: Prof. Júlio César Marques de Lima

**Porto Alegre
2024**

AGRADECIMENTOS

Agradeço aos meus pais, Maurício Bisso de Mello e Maria do Carmo Bertoldi de Mello, por todas as oportunidades que me trouxeram até aqui e por sempre torcerem pelo meu sucesso e me darem todo o apoio necessário.

Agradeço à minha irmã Carolina Bertoldi de Mello que apesar de morar longe nunca mediu esforços para se manter presente e colaborar com o meu crescimento.

Agradeço à minha namorada Natália Pritsch Steigleder que é acima de tudo uma grande amiga e que me apoiou incansavelmente em todas as fases da graduação e deste projeto.

Agradeço a todos os meus amigos, que tornaram a experiência mais leve.

Sou grato pela confiança depositada na minha proposta de projeto pelo meu professor, Prof. Júlio César Marques de Lima, orientador do meu trabalho, Agradeço profundamente pelas contribuições dadas durante todo o processo.

Por fim, agradeço à *PUCRS* e ao seu corpo docente que sempre demonstrou estar comprometido com a qualidade e excelência do ensino.

RESUMO

Com o rápido avanço da tecnologia, uma ampla variedade de dispositivos voltados à automação residencial tem sido lançada no mercado. Esses aparelhos permitem controle e monitoramento remoto por meio de computadores, dispositivos móveis e até comandos de voz. Contudo, a grande oferta de novos produtos muitas vezes acelera a obsolescência de outros, que, apesar de serem de excelente qualidade, acabam sendo deixados de lado. Este trabalho tem como objetivo desenvolver um controle remoto universal sem fio, capaz de dar uma nova utilidade a esses dispositivos, prolongando sua vida útil e evitando o seu descarte precoce. É apresentada inicialmente a história da radiação infravermelha, abordando seu descobrimento, o cientista responsável, o ano em que ocorreu, suas aplicações naquela época e como ela é utilizada atualmente para a transferência de dados nos dispositivos residenciais. Para atingir o objetivo, foi desenvolvido um projeto que integra *firmware* e *hardware*, capaz de receber sinais infravermelhos modulados, armazená-los na memória do SoC (System-on-Chip) *ESP8266* e emití-los posteriormente. Todo o controle é realizado por meio de uma página *WEB* com design responsivo, permitindo o uso tanto em dispositivos móveis quanto em *desktops*. Os detalhes da implementação são descritos no trabalho, através de fluxogramas e esquemáticos. Para uma apresentação mais prática e organizada, todo o circuito necessário foi instalado em uma caixa preta, alimentada via porta USB-C. Por fim, são apresentados os resultados, incluindo a distância de operação e o consumo energético, que atenderam plenamente os objetivos propostos.

Palavras-chave: *ESP8266*; infravermelho; controle remoto; *IoT*.

ABSTRACT

With the rapid advancement of technology, a wide variety of devices aimed at home automation have been launched on the market. These devices allow remote control and monitoring through computers, mobile devices and even voice commands. However, the large supply of new products often accelerates the obsolescence of others, which, despite being of excellent quality, end up being left aside. This work aims to develop a universal wireless remote control, capable of giving a new use to these devices, extending their useful life and avoiding their premature disposal. Initially, the history of infrared radiation is presented, addressing its discovery, the scientist responsible, the year in which it occurred, its applications at that time and how it is currently used for data transfer in residential devices. To achieve the objective, a project was developed that integrates firmware and hardware, capable of receiving modulated infrared signals, storing them in the memory of the *ESP8266 SoC (System-on-Chip)* and emitting them later. All control is performed through a *WEB* page with responsive design, allowing use on both mobile devices and desktops. The implementation details are described in the paper, through flowcharts and schematics. For a more practical and organized presentation, the entire necessary circuit was installed in a black box, powered via USB-C port. Finally, the results are presented, including the operating distance and energy consumption, which fully met the proposed objectives.

Keywords: *ESP8266*; infrared; remote control; IoT.

SUMÁRIO

1. INTRODUÇÃO.....	7
1.2 Motivação.....	7
1.3 Objetivo.....	8
2. DESENVOLVIMENTO.....	9
2.1 Descrição dos itens utilizados.....	9
2.1.1 Placa de Desenvolvimento NodeMCU ESP-12E ESP8266 Amica.....	9
2.1.2 Receptor IR Vishay TSOP4838.....	10
2.1.3 LED Infravermelho 5mm.....	11
2.1.4 Circuito Integrado NE555P.....	12
2.1.5 Transistor PN2222A.....	12
2.1.6 Protoboard.....	13
2.1.7 Placa de CI Universal.....	14
2.1.8 Conector USB-C Fêmea.....	15
2.1.9 Caixa Plástica.....	15
2.2 Infravermelho.....	16
2.2.1 História.....	16
2.2.2 CIR: Como funciona?.....	18
2.2.3 Protocolos.....	19
2.2.3.1 Protocolo NEC.....	19
2.2.3.2 Protocolo Philips RC-5.....	21
2.2.3.3 Protocolo Sony SIRC.....	23
2.2.4 Abordagem Utilizada.....	24
2.3 Arquitetura do Sistema.....	25
2.3.1 Módulo de recepção de sinal infravermelho.....	26
2.3.2 Módulo de emissão de sinal infravermelho.....	26
2.3.3 Módulo de front-end voltado ao usuário.....	27
2.4 Projeto do Firmware.....	28
2.4.1 Ambiente de Desenvolvimento.....	28
2.4.2 Organização dos Arquivos.....	29
2.4.3 Recepção de sinal IR.....	30
2.4.4 Emissão de sinal IR.....	32
2.4.5 Inicialização.....	33
2.4.6 Armazenamento de sinal IR.....	34
2.4.7 Interface WEB.....	35
2.5 Projeto do Hardware.....	38
2.5.1 Esquemático.....	39
2.5.1.1 Botão reset.....	39
2.5.1.2 Filtro de entrada.....	39
2.5.1.3 Microcontrolador.....	40
2.5.1.4 Receptor e demodulador IR.....	41
2.5.1.5 Gerador de sinal modulado.....	41
2.5.1.6 Emissor IR 360 graus.....	42

2.5.2 Processo de montagem.....	43
3. RESULTADOS.....	48
3.1 Armazenamento de Sinal IR.....	48
3.2 Recepção de Sinal IR.....	49
3.3 Emissão de Sinal IR.....	49
3.4 Consumo de Energia.....	50
4. CONCLUSÃO.....	51
REFERÊNCIAS.....	53
APÊNDICE A. Esquemático da Placa de Circuito Impresso Universal.....	54

1. INTRODUÇÃO

A evolução tecnológica nas últimas décadas é notável, transformando diversos aspectos do cotidiano. Hoje se vive em um mundo onde a tecnologia não só facilita tarefas diárias, mas também possibilita novos níveis de automação e monitoramento em diversos ambientes.

IoT (*Internet das Coisas*) é um termo muito utilizado hoje, que engloba qualquer dispositivo físico conectado a uma rede, normalmente sem fio, cujo objetivo é coletar informações, normalmente através de sensores, e realizar a troca destas com outros dispositivos. Um mercado IoT não muito conhecido e que vem crescendo ultimamente é o automotivo, hoje já existem carros populares que contam com um aplicativo móvel com recursos, como: autonomia, local exato através de GPS (*sistema de posicionamento global*), possibilidade de configurar ar-condicionado, entre outras funções.

A automação residencial é uma das maiores e mais consolidadas aplicações de IoT, movimentando dezenas de bilhões de dólares anualmente. Ela possui uma gama variada de dispositivos, que vão de robôs de serviços e lâmpadas inteligentes incluindo sistemas de segurança, como câmeras de vigilância e sistemas de alarme.

No quesito conectividade, há diversos protocolos utilizados para a comunicação de baixo alcance, entre os mais comuns a WLAN/Wi-Fi (*wireless local area network ou rede sem fio de área local*) e o Bluetooth, tecnologias já presentes no cotidiano, como também Zigbee e o Z-Wave, estes criados com o foco no uso em dispositivos IoT.

1.2 Motivação

Com o crescente interesse por produtos voltados à automação residencial, as empresas começaram a lançar versões atualizadas de dispositivos já existentes, como ar-condicionados, agora equipados com conectividade sem fio. Esse movimento tem acelerado a obsolescência de produtos que, apesar de funcionais e de boa qualidade, não possuem essa funcionalidade integrada.

Atualmente, o foco das empresas está em desenvolver novos produtos com recursos de IoT, muitas vezes sem considerar o impacto ambiental e o aumento do lixo eletrônico que isso pode gerar. Assim, este trabalho de conclusão de curso propõe uma solução que prolonga a vida útil de produtos antigos, proporcionando-lhes uma nova funcionalidade e contribuindo para uma abordagem mais sustentável no prolongamento via controle e conectividade da automação residencial.

1.3 Objetivo

O objetivo deste trabalho é combinar o melhor dos dois mundos, proporcionando uma solução que permite transformar produtos antigos, não originalmente “inteligentes”, em dispositivos controláveis por meios modernos. Isso pode ser alcançado através da integração entre o *IR (infrared ou infravermelho)*, amplamente utilizado hoje para controle de dispositivos, como televisores e ar-condicionados, e a conectividade *Wi-Fi*.

Para isso, este projeto contará com o desenvolvimento de um "Controle Remoto Universal Wireless" que permite ao usuário clonar e armazenar sinais IR de controles tradicionais e emitir estes comandos através de uma interface *WEB*, acessível tanto através de dispositivos móveis quanto *desktops*.

2. DESENVOLVIMENTO

Este capítulo descreve o desenvolvimento do projeto de conclusão de curso. O projeto trata-se de um controle remoto infravermelho universal, com acionamento através de *smartphone* e computadores.

2.1 Descrição dos itens utilizados

Aqui são descritos todos os itens utilizados para a elaboração do projeto, assim como o motivo, características e funcionalidades.

2.1.1 Placa de Desenvolvimento NodeMCU ESP-12E ESP8266 Amica

Esta placa (Figura 2.1.1.1) é a base de processamento do projeto e foi, em 2020, adquirida e utilizada pelo autor durante a pandemia na disciplina de “Programação de Periféricos”.

No decorrer do desenvolvimento observou-se que a capacidade do SoC presente nela, *ESP8266*, não foi suficiente para que o processo de modulação do sinal se tornasse estável, porém, seguindo a ideia do reuso e sustentabilidade, optou-se por manter a sua utilização.

Uma dúvida frequente é o sobre o próprio nome da mesma. “*ESP8266*” seria o SoC (*System-on-Chip*), “*ESP-12E*” é uma placa que conta com o SoC descrito, alguns componentes e antena *Wi-Fi* e, por fim “*NodeMCU*” se trata da placa de desenvolvimento, com tudo dito anteriormente mais *USB-UART*, regulador de tensão, botões, pinos de *I/O*, entre outros recursos.

Abaixo seguem as especificações técnicas relevantes:

- Módulo: *ESP-12E Amica*;
- Padrão Wi-Fi: *802.11 b/g/n*;
- Portas GPIO: 11;
- Tensão de operação: 4,5v ~ 9,0v;
- Frequência de operação: 80MHz;
- Memória *Flash*: 4MB;

- Memória *RAM*: 80KB.

Figura 2.1.1.1 - Placa de Desenvolvimento NodeMCU ESP-12E ESP8266 Amica.



Fonte: Modificado de Eletrogate - "Módulo WiFi ESP8266 NodeMcu ESP-12E Amica" (2024).

2.1.2 Receptor *IR Vishay TSOP4838*

O *TSOP4838* (Figura 2.1.2.1) é um receptor *IR* produzido pela *Vishay* muito utilizado nas aplicações de controle remoto infravermelho e essencial para o projeto. Como funcionalidades relevantes temos a imunidade contra ruídos e a amplificação do sinal. Detalhes podem ser vistos no diagrama de blocos (Figura 2.1.2.2). Os últimos dois algarismos de seu nome referenciam a frequência da portadora, neste caso estamos usando 38KHz, que atende a vasta maioria de dispositivos encontrados no mercado. Seguem abaixo as especificações técnicas:

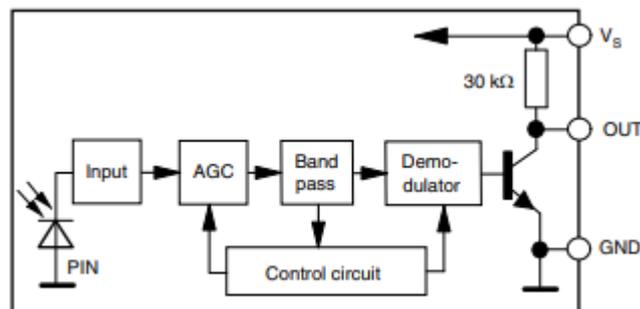
- Frequência da portadora: 38KHz;
- Tensão de operação: 2,0V ~ 5,5V;
- Corrente: 5mA;
- Consumo de energia: 10mW.

Figura 2.1.2.1 - Módulo Receptor IR Vishay TSOP4838.



Fonte: Modificado de Vishay - "IR Receiver Modules for Remote Control Systems - TSOP22..., TSOP24..., TSOP48..., TSOP44" (2024).

Figura 2.1.2.2 - Diagrama de Blocos do Módulo Receptor IR Vishay TSOP4838.



Fonte: Modificado de Vishay - "IR Receiver Modules for Remote Control Systems - TSOP22..., TSOP24..., TSOP48..., TSOP44" (2024).

2.1.3 LED Infravermelho 5mm

Os *LEDs* (Figura 2.1.3.1) foram escolhidos já se baseando no *design* final do projeto. A ideia é que o controle remoto universal consiga emitir sinais para qualquer dispositivo ao seu redor, portanto a emissão deve ocorrer num ângulo de 360°.

Este requisito atingido através do posicionamento perfeito de 6 *LEDs*, onde cada um conta com um ângulo de emissão de 60°. Isto somado a uma alta potência garante o funcionamento perfeito. Detalhes técnicos sobre o *LED* abaixo:

- Tensão de operação: 1,5V
- Corrente: 50mA
- Ângulo de emissão: 60 graus
- Comprimento de onda: 940nm

Figura 2.1.3.1 - LED Infravermelho 5mm.



Fonte: Modificado de Tiggercomp - "Led Infravermelho 5mm - 1.5V 100mW" (2024).

2.1.4 Circuito Integrado *NE555P*

O *NE555P* (Figura 2.1.4.1) é um dispositivo de temporização capaz de fornecer oscilações em alta precisão. Este item passou a ser utilizado ao ser validado que o SoC *ESP8266* não teria capacidade de processamento suficiente para a modulação do sinal. Como este *CI* trabalha na modulação do sinal, ele trabalha numa oscilação de 38KHz. Mais detalhes técnicos abaixo:

- Tensão de operação: 4,5V ~ 16V;
- Corrente: ~10mA;
- Corrente de saída: ~200mA.

Figura 2.1.4.1 - Circuito Integrado *NE555P*.



Fonte: Modificado de Ponto da Eletrônica: "Circuito Integrado *NE555P*" (2024).

2.1.5 *Transistor PN2222A*

O *transistor PN2222A* (Figura 2.1.5.1) é necessário no projeto para o chaveamento da matriz de *LEDs*. Como se trata de um *TBJ* (Transistor Bipolar de

Junção), o seu acionamento é realizado por corrente na base, ao contrário dos *FETs*, que são acionados por tensão.

Como a alimentação dos componentes é feita através dos 5V do *USB*, uma tensão relativamente baixa, a utilização de *FETs* disponíveis no laboratório se provou inviável, pois o *FET* precisa de uma tensão de disparo mais elevada, afetando a funcionalidade do controle remoto *wireless*. Especificações técnicas do *transistor PN2222A* utilizado abaixo:

- Corrente máxima no Coletor I_c : 1,0A
- Tensão máxima Coletor-Emissor: 40V
- Tensão máxima Coletor-Base: 75V
- Tensão máxima Emissor-Base: 6V
- Ganho *HFE* (nas características do projeto): entre 50 e 100

Figura 2.1.5.1 - Transistor *TBJ* (*Transistor Bipolar de Junção*) *PN2222A*.

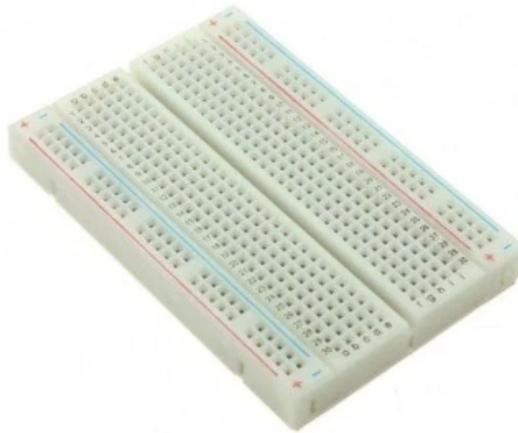


Fonte: Modificado de Eletrodex - "PN2222A PTH - Transistor NPN" (2024).

2.1.6 *Protoboard*

A utilização de *protoboard* (Figura 2.1.6.1) é indispensável para a prototipação do projeto, e foi utilizada em toda a fase de testes do trabalho. A imagem é apenas ilustrativa.

Figura 2.1.6.1 - *Protoboard*.



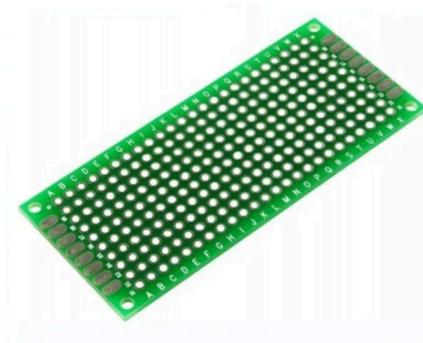
Fonte: Modificado de Eletrogate: “Protoboard 400 Pontos” (2024).

2.1.7 Placa de CI Universal

A placa de circuito impresso (Figura 2.1.7.1) é utilizada na etapa final do projeto, onde os componentes montados inicialmente na protoboard são transferidos para uma placa universal. Esse processo permite acomodar o projeto de forma organizada dentro de uma caixa plástica.

O CI universal selecionado possui dimensões de 3 cm x 7 cm, e a utilização de três unidades é suficiente para atender às necessidades do projeto.

Figura 2.1.7.1 - Placa de CI (*circuito impresso*) Universal.



Fonte: Achei Componentes: “Placa de Circuito Impresso Face Dupla 03cm x 07cm Universal” (2024).

2.1.8 Conector *USB-C* Fêmea

A alimentação do projeto, conforme mencionado anteriormente, é feita através de uma fonte USB de 5V. O conector escolhido foi o *USB-C*, pois garante alta compatibilidade, já que esse padrão é amplamente adotado nos dispositivos.

Figura 2.1.8.1 - Conector *USB-C* Fêmea.



Fonte: Descomplica Solucoes Industriais: “Conector Jack De Embutir Usb Tipo C Fêmea Carregamento Cabo” (2024).

2.1.9 Caixa Plástica

Utilizada para acomodar todos os componentes do projeto após as validações na *protoboard*. Possui as dimensões de 32 x 72 x 85 mm, tamanho ideal para os requisitos.

Figura 2.1.9.1 - Caixa Plástica.



Fonte: Mult Comercial: “Caixa Plástica PB-085/2 - Patola - Caixa 70” (2024).

2.2 Infravermelho

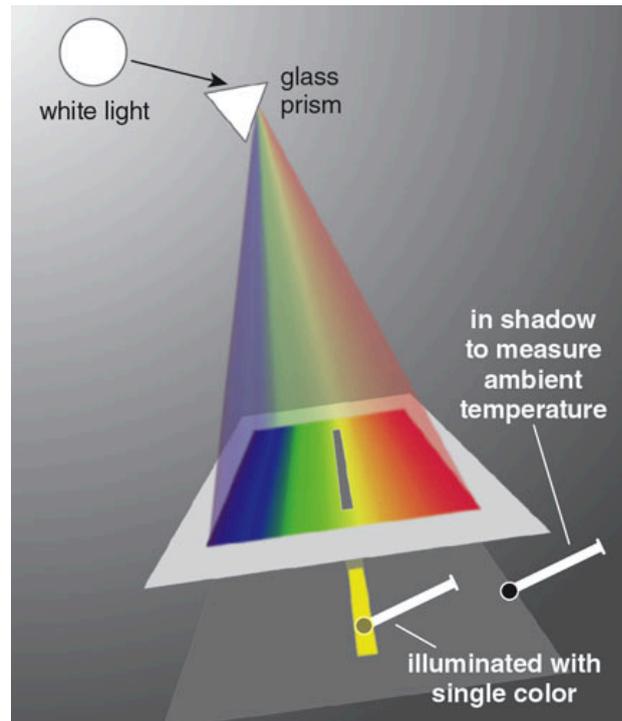
Esta seção explora em detalhes a radiação infravermelha, incluindo sua história, o processo de descoberta, o início de sua aplicação na transmissão de dados e a abordagem adotada neste projeto.

2.2.1 História

A descoberta da radiação infravermelha foi feita por Friedrich William Herschel (1738-1822). Friedrich nasceu em Hanôver, na Alemanha, porém foi naturalizado inglês. Ele foi muito conhecido tanto no ramo da música, quanto por suas pesquisas sobre o sistema solar, onde, neste último, ficou famoso pela descoberta do planeta Urano.

Herschel, no ano de 1800, através de um simples experimento (Figura 2.2.1.1) onde, com um prisma, tentava descobrir quanto calor passava através de filtros de diferentes cores, acabou descobrindo a luz infravermelha. Friedrich colocou termômetros de mercúrio em cada faixa de cor para medir a temperatura. Ele observou que, ao medir uma região além do vermelho, onde a luz visível já não era mais perceptível, a temperatura continuava a aumentar, indicando a presença de uma forma invisível de radiação, que, naquele momento, nomeou de de “Raios Caloríficos”.

Figura 2.2.1.1 - Ilustração do experimento realizado por Friedrich William Herschel.



Fonte: Jack R. White - "Herschel and the Puzzle of Infrared".

O inglês seguiu com os experimentos e, posteriormente, descobriu que estes raios eram refletidos, refratados, absorvidos e transmitidos de maneira semelhante à luz visível. Com essas informações notou que o que foi descoberto era uma forma de luz (ou radiação) além da luz vermelha, e é daí que vem o nome que conhecemos hoje, infravermelho, onde "infra" significa "abaixo" em latim.

Apesar dessa descoberta ter ocorrido tão cedo, somente na primeira metade do século XX começaram a haver avanços significativos, que trariam aplicações práticas a radiação infravermelha. No ano de 1929 o físico Húngaro Kálmán Tihanyi desenvolveu a primeira câmera de visão noturna sensível a luz infravermelha, que foi patenteada para controle de aeronaves e veículos militares.

Em 1980, a empresa canadense Viewstar lançou um *set-top box* (receptor-decodificador de televisão a cabo) acompanhado do primeiro controle remoto infravermelho. O produto foi um sucesso imediato, com 1,6 milhão de unidades vendidas até 1989. A partir de então, controles remotos com essa tecnologia se tornaram amplamente utilizados, permanecendo como padrão até hoje, e é sobre ela que trataremos nas próximas seções.

2.2.2 CIR: Como funciona?

CIR (consumer infrared) é uma classe de dispositivos eletrônicos que utilizam os raios infravermelhos como método de comunicação sem fio, sendo os mais comuns os controles remotos para operar televisores e aparelhos de ar-condicionado a distância.

Outra aplicação, que atualmente não é mais utilizada, foi a transmissão de dados entre dispositivos. No início do século XXI, era comum que aparelhos móveis e notebooks fossem equipados com essa tecnologia, permitindo a transferência de arquivos binários ao simplesmente apontar o emissor e o receptor de cada dispositivo um para o outro.

A troca de informações funciona por meio de pulsos de luz infravermelha, emitida por um *LED* (diodo emissor de luz) presente no transmissor, como em um controle remoto, e é captada por um receptor infravermelho no dispositivo que recebe o sinal. Essa luz infravermelha transporta informações na forma de pulsos que representam estados binários, usando combinações de “0s” e “1s” para enviar comandos simples.

A transmissão de dados infravermelhos segue uma sequência codificada, na qual cada comando, como “ligar”, “desligar” ou “aumentar volume”, é convertido em uma sequência específica de pulsos de luz. Quando o botão de um controle remoto é pressionado, o circuito do transmissor gera uma série de *bits* e estes são modulados na frequência da portadora, normalmente entre 30 e 60 kHz, sendo 38 kHz a frequência mais comum, que representam o comando desejado. Esse sinal infravermelho viaja até o receptor, que realiza o processo de demodulação, avalia os *bits* e converte-os em um comando que o dispositivo executa.

Uma característica importante da transmissão de dados via infravermelho é que ela requer um campo de visão desobstruído entre o transmissor e o receptor. Isso ocorre porque a radiação infravermelha, assim como a luz visível, não atravessa obstáculos opacos como paredes e móveis. Embora essa limitação restrinja a área de operação, ela também torna o infravermelho uma opção segura para controle de curto alcance, pois elimina interferências com dispositivos em ambientes próximos.

2.2.3 Protocolos

Na transmissão de dados via radiação infravermelha, diversos protocolos padronizam a forma como eles são codificados e modulados. Esses protocolos são conjuntos de regras que definem a duração e o espaçamento dos pulsos de luz infravermelha para representar dados binários, garantindo que o receptor entenda corretamente o comando enviado pelo transmissor. Os principais protocolos usados em sistemas de controle remoto *IR* são *NEC*, *Philips RC-5* e *Sony SIRC*, cada um com suas particularidades, que serão vistas nas seções a seguir.

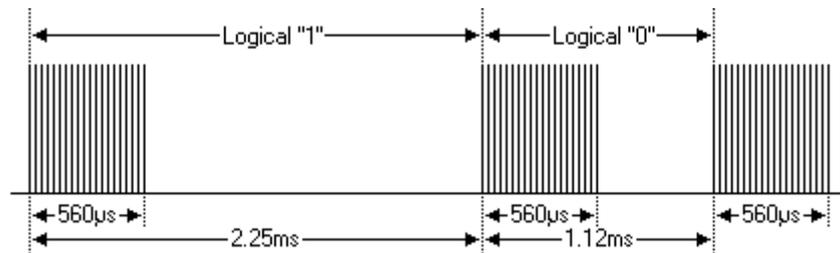
2.2.3.1 Protocolo NEC

O protocolo *NEC*, criado pela empresa japonesa *Nippon Electric Company*, é um dos mais utilizados em controles remotos de *TVs*, sistemas de som e outros eletrônicos. Suas principais características estão descritas abaixo:

- Frequência da portadora: 38 kHz;
- Usa modulação por distância de pulso;
- Tempo do *bit* '0': 1.125ms;
- Tempo do *bit* '1': 2.25ms;
- Espaço para endereçamento de 8 *bits*;
- Espaço para comando de 8 *bits*;
- Endereço e comando são enviados duas vezes para redundância.

O protocolo utiliza modulação por distância de pulso, sendo cada pulso de 560 μ s (em torno de 21 ciclos da portadora). Como podemos ver abaixo (Figura 2.2.3.1.1), o *bit* '1' leva 2.25ms para ser transmitido, enquanto o *bit* '0' leva a metade disso, 1.125ms.

Figura 2.2.3.1.1 - Modulação do Protocolo NEC.

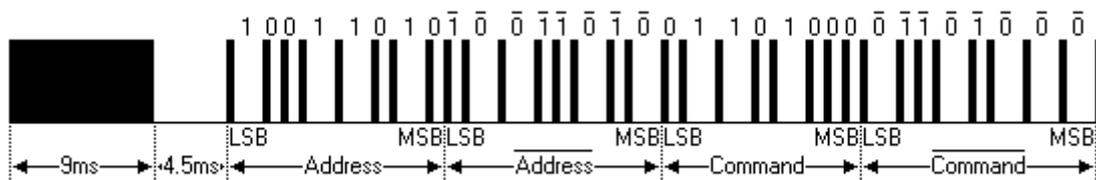


Fonte: SB-Projects - "NEC Protocol - Modulation" (2024).

Abaixo (Figura 2.2.3.1.2), é possível visualizar um trem de pulsos típico deste protocolo, onde o valor do endereço escolhido para exemplo foi 59 (hexadecimal) e o de comando dezesseis (hexadecimal). A mensagem sempre começa com uma série de *bits* '1', totalizando 9ms, cuja tarefa é definir o ganho em receptores *IR* mais antigos, seguido por *bits* '0' (4.5ms). Após isso tanto o endereço quanto o comando são enviados duas vezes (uma delas invertida) para garantir a redundância e também deixar o tamanho da mensagem constante.

É possível remover a redundância descrita acima, com o intuito de aumentar o espaçamento destinado a endereço e comando, ao realizar este procedimento ambos ficam com 16 *bits* de largura.

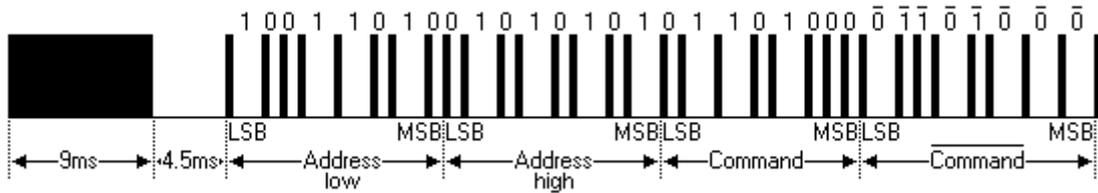
Figura 2.2.3.1.2 - Pulsos do Protocolo NEC.



Fonte: SB-Projects - "NEC Protocol - Protocol" (2024).

O Protocolo NEC é utilizado em tão larga escala que muitas vezes os vemos na forma estendida (Figura 2.2.3.1.3), onde a quantidade de *bits* destinada para endereçamento é aumentada de oito para dezesseis, perdendo a redundância e também a característica de tempo de transmissão constante.

Figura 2.2.3.1.3 - Pulsos do Protocolo NEC Estendido.



Fonte: SB-Projects - "NEC Protocol - Extended NEC protocol" (2024).

O comando em si é enviado apenas uma única vez, mesmo que o botão seja pressionado. Ao invés de realizar o reenvio completo, o emissor transmite uma mensagem de repetição (Figura 2.2.3.1.4) a cada 110ms.

Figura 2.2.3.1.4 - Pulsos do sinal de repetição do Protocolo NEC.



Fonte: SB-Projects - "NEC Protocol - NEC protocol" (2024).

2.2.3.2 Protocolo Philips RC-5

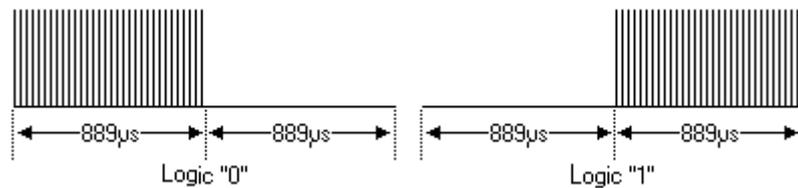
O RC-5, desenvolvido pela Philips no início da década de 80, é amplamente utilizado em equipamentos de áudio e vídeo. Suas principais características estão descritas abaixo:

- Frequência da portadora: 36 kHz (também visto em 38 kHz e 40 kHz);
- Usa modulação bifásica (Manchester);
- Tempo do *bit* '0' e '1': 1.778ms;
- Espaço para endereçamento de 5 *bits*;
- Espaço para comando de 6 *bits*;
- Não há redundância

Ele utiliza, por padrão, 36 kHz na portadora e modulação bifásica, também conhecida como Codificação Manchester. Este tipo de codificação opera dividindo

cada período de um *bit* em duas metades iguais, sendo a primeira metade o *bit* '1', ou vice-versa. Isso significa que a cada período ocorre uma transição de sinal e ambos os *bits* possuem a mesma largura. Abaixo (Figura 2.2.3.2.1) podemos visualizar como a modulação funciona na prática.

Figura 2.2.3.2.1 - Modulação do Protocolo Philips RC-5.

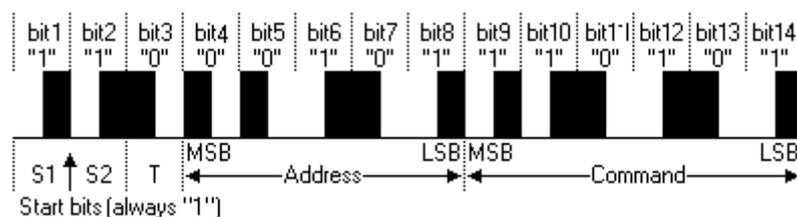


Fonte: SB-Projects - "Philips RC-5 Protocol - Modulation" (2024).

Em relação ao seu trem de pulsos (Figura 2.2.3.2.2), podemos notar que existem três *start bits*, onde os dois primeiros, *S1* e *S2*, são sempre nível lógico '1' e *toggle* 'T' é um *bit* variável, que é invertido toda vez que um botão é solto. A função do *toggle* é indicar ao receptor se um botão está sendo pressionado várias vezes ou o usuário está permanecendo com ele pressionado. Quando o botão permanesse pressionado a mensagem é repetida a cada 114ms.

Ainda no mesmo contexto, os próximos *bits* são relacionados ao endereçamento (cinco *bits*) e comando (seis *bits*), respectivamente. Existe o Protocolo RC-5 estendido, onde se abdica do *start bit toggle* e um *bit* é acrescentado ao comando, totalizando sete *bits*.

Figura 2.2.3.2.2 - Pulsos do Protocolo Philips RC-5.



Fonte: SB-Projects - "Philips RC-5 Protocol - Protocol" (2024).

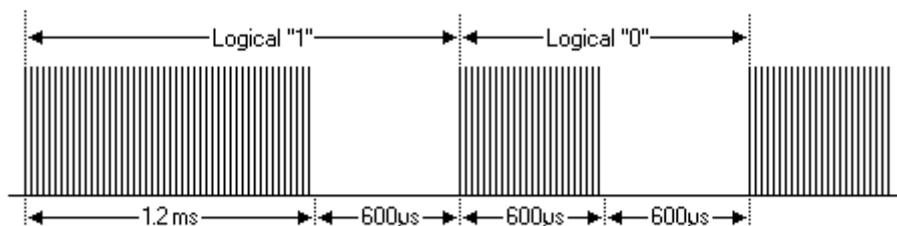
2.2.3.3 Protocolo Sony SIRC

O protocolo *SIRC*, também conhecido como *Sony Infrared Remote Control*, foi desenvolvido pela Sony e é amplamente utilizado em seus produtos, como TVs e sistemas de áudio.

- Frequência da portadora: 40 kHz;
- Usa modulação por distância de pulso;
- Tempo do *bit* '0': 0.6ms;
- Tempo do *bit* '1': 1.2ms;
- Espaço para endereçamento de 5 ou 8 *bits*;
- Espaço para comando de 7 *bits*;
- Não há redundância.

O protocolo da *Sony*, assim como o protocolo *NEC*, usa modulação por distância de pulso, porém sobre uma portadora de 40 kHz. Como podemos ver a seguir (Figura 2.2.3.3.1), o *bit* lógico '1' é representado por um pulso de 1.2ms e o *bit* '0' por um pulso de 0.6ms, ambos separados por um espaço de 0.6ms.

Figura 2.2.3.3.1 - Modulação do Protocolo Sony SIRC.

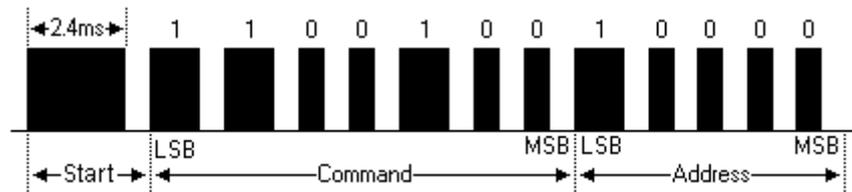


Fonte: SB-Projects - "Sony SIRC Protocol - Modulation" (2024).

Seu trem de pulsos (Figura 2.2.3.3.2) é bem simples, o início da transmissão é dado por um grande pulso de 2.4ms, seguido por um espaço de 0.6ms, este longo pulso é utilizado para ajustar o ganho do receptor *IR*. Os sete *bits* seguintes são destinados ao envio do comando e os cinco próximos ao endereçamento. Vale ressaltar que existem versões estendidas do protocolo que realizam um incremento no espaço de endereçamento de cinco para oito *bits*.

A repetição de comandos é emitida pelo emissor a cada 45ms, caso o botão do controle remoto siga pressionado pelo usuário.

Figura 2.2.3.3.2 - Pulsos do Protocolo Sony SIRC.



Fonte: SB-Projects - "Sony SIRC Protocol - Protocol" (2024)

2.2.4 Abordagem Utilizada

Na seção anterior foram vistos os protocolos para transmissão de dados via radiação infravermelha mais conhecidos e utilizados nos aparelhos eletrônicos. Com as informações e detalhes descritos seria viável realizar a implementação de alguns deles, porém, um dos propósitos principais do projeto seria comprometido: ser um controle remoto **universal**.

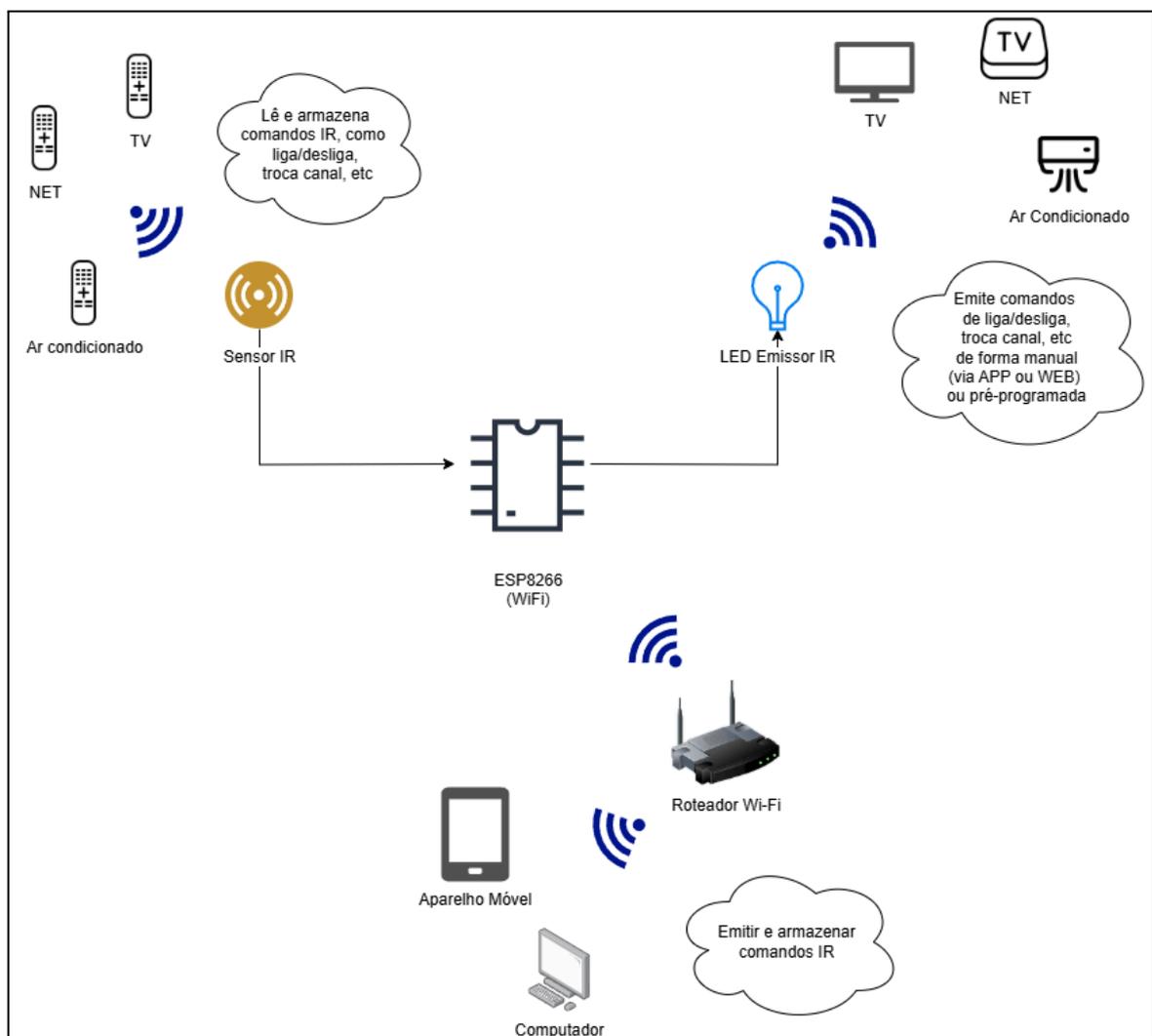
Os testes realizados em seis dispositivos distintos, abrangendo desde ar-condicionados até *home theaters*, revelaram que muitas empresas utilizam protocolos proprietários com pequenas variações em relação aos padrões mais conhecidos. Essas diferenças, embora sutis, são suficientes para ocasionar o não funcionamento do projeto com estes dispositivos.

Para garantir o funcionamento numa gama maior de equipamentos, o projeto adotou a abordagem de armazenar em memória os dados demodulados emitidos pelos controles remotos, sem tentar reconhecer os protocolos, funcionando basicamente como um dispositivo clone. Dessa forma, basta o aparelho utilizar a frequência da portadora de 38 kHz para ser compatível com o controle remoto *wireless* desenvolvido no trabalho.

2.3 Arquitetura do Sistema

A arquitetura do sistema (Figura 2.3.1) é baseada na plataforma de desenvolvimento *NodeMCU ESP-12E*, cujo microcontrolador presente é o *ESP8266*, que atende os requisitos mínimos necessários de processamento e conectividade para a execução do projeto. A arquitetura foi dividida em 3 módulos, que atendem necessidades específicas.

Figura 2.3.1 - Arquitetura do Sistema.



Fonte: Própria (2024).

2.3.1 Módulo de recepção de sinal infravermelho

Este módulo é de extrema importância para o funcionamento do sistema, pois, quando sua memória está vazia, nenhum sinal infravermelho estará armazenado e, portanto, não teremos como acionar remotamente nenhum dispositivo.

A função deste módulo é receber sinais infravermelho modulados, em 38KHz, demodulá-los e armazená-los em memória. O processo de demodulação é realizado através do módulo receptor *IR TSOP4838*, produzido pela empresa *Vishay*, que é amplamente utilizado para este tipo de projeto e facilmente encontrado no mercado brasileiro.

Figura 2.3.1.1 - Arquitetura do módulo de recepção.



Fonte: Própria (2024).

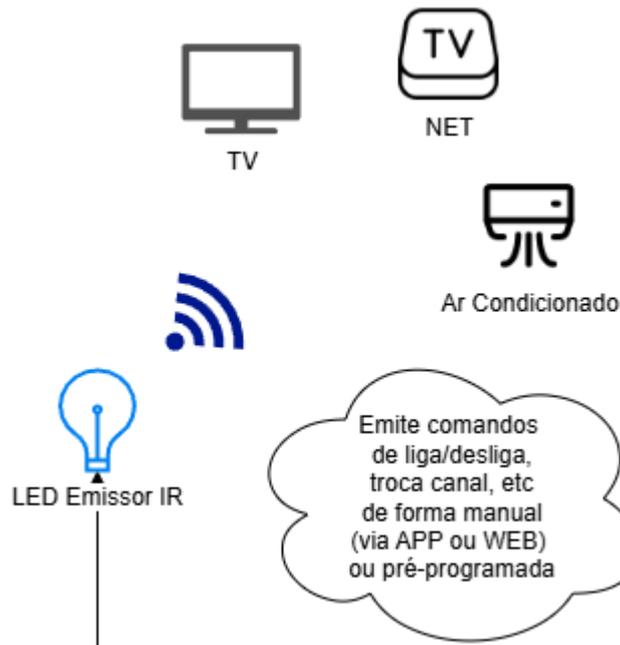
2.3.2 Módulo de emissão de sinal infravermelho

Este módulo (Figura 2.3.2.1), ao contrário do visto anteriormente, trata da leitura do sinal infravermelho demodulado em memória, modulação e, por fim emissão deste sinal. A execução destas tarefas ocorrem da seguinte maneira:

1. Leitura do sinal demodulado é realizada pelo microcontrolador que encaminha este ao *CI NE555P* através de uma das portas *GPIO*;
2. O *CI NE555P*, oscilador cuja precisão atende aos requisitos do projeto, recebe o sinal demodulado, o modula em 38 KHz, e envia o sinal modulado aos dois *transistores PN2222A*;

3. Os *transistores* fazem o chaveamento dos LEDs infravermelho que, por sua vez, acionam o dispositivo remoto.

Figura 2.3.2.1 - Arquitetura do módulo de emissão.



Fonte: Própria (2024).

2.3.3 Módulo de front-end voltado ao usuário

De nada adianta os módulos de emissão e recepção serem funcionais se o usuário não pode interagir com eles, e esta é a função deste módulo (Figura 2.3.3.1).

Toda a interação do hardware com o usuário é realizada através de uma página *WEB HTTP*, projetada seguindo a abordagem *WDR (Web design responsivo)*, e, portanto pode ser utilizada tanto em dispositivos com tela pequena (aparelhos móveis, *tablets*) quanto os de tela grande (*desktops*). A ideia é trazer uma página simples e funcional para que tanto usuários avançados quanto mais leigos consigam acionar e clonar dispositivos sem dificuldades.

Figura 2.3.3.1 - Arquitetura do módulo front-end.



Fonte: Própria (2024).

2.4 Projeto do Firmware

Para o pleno funcionamento do controle remoto wireless foi desenvolvido um *firmware*, escrito na linguagem C, que comporta todas as funcionalidades descritas nas seções acima. Em relação ao tempo investido, com certeza foi a etapa que exigiu mais esforço e as limitações do SoC *ESP8266* encontradas aqui exigiram alterações na estrutura do trabalho.

Nas subseções abaixo entraremos a fundo no desenvolvimento de cada etapa do *firmware*.

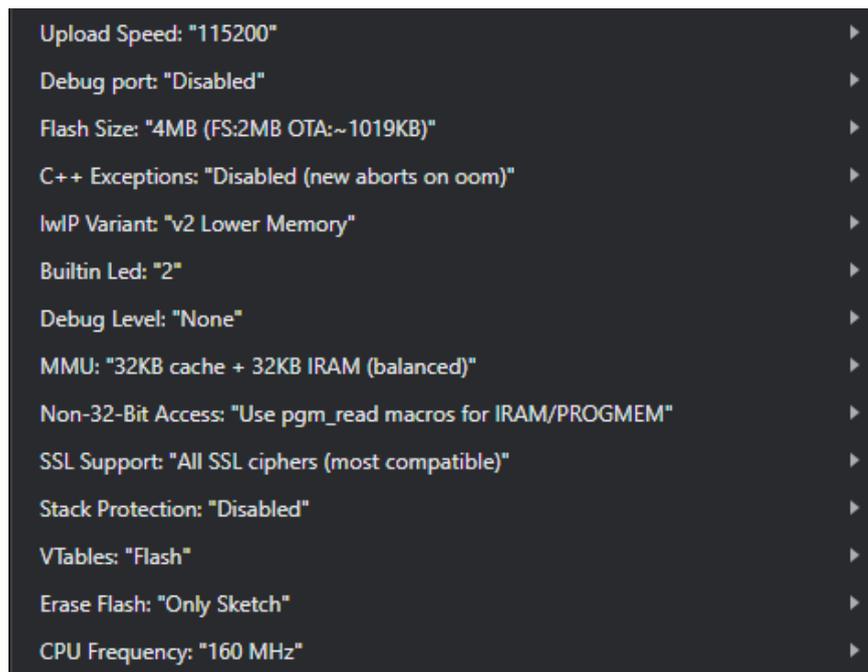
2.4.1 Ambiente de Desenvolvimento

Qualquer desenvolvimento em linguagem de programação exige um ambiente de desenvolvimento ágil, confiável e que integre com os equipamentos ou bibliotecas que serão utilizados. Levando em consideração estes três fatores foi

escolhida a *IDE* (ambiente de desenvolvimento integrado) do *Arduino*, no desenvolvimento de projetos.

A *IDE* integra perfeitamente com o *ESP8266*, possibilitando via interface gráfica realizar configurações específicas do *SoC* (Figura 2.4.1.1), como a frequência da *CPU*, memória destinada para atualizações *OTA* (*over-the-air*), etc.

Figura 2.4.1.1 - Configurações do *SoC* via *Arduino IDE*.



Fonte: Própria (2024).

Além disso, permite separar o programa em diversos arquivos, para melhor organização, realizar *upload* do *firmware* de maneira fácil e há uma parte da tela destinada especificamente a interface *serial* que, como sabemos, é indispensável para o *debug* da aplicação.

2.4.2 Organização dos Arquivos

O *firmware*, para organização, foi separado em seis arquivos distintos, sendo:

- ***TCC_IR.ino*** - arquivo *main* da aplicação, comporta as funções de *setup* do *firmware*, *defines* e constantes;
- ***EEPROM.ino*** - funções destinadas ao uso da memória *EEPROM* do *SoC*;

- **WebServer.ino** - gerencia o servidor *WEB* quando a placa não está resetada;
- **WebServerReset.ino** - gerencia o servidor *WEB* quando a placa está resetada;
- **SPIFFS.ino** - funções destinadas ao uso do sistema de arquivos *SPIFFS*;
- **Signal.ino** - funções relacionadas ao sinal *IR*.

2.4.3 Recepção de sinal IR

A recepção de sinal infravermelho é um processo assíncrono no projeto, visto que, para isso, foi adicionada uma interrupção, sensível a borda de descida, no pino onde o receptor *TSOP4838*, que já demodula o sinal, está conectado. Através deste mecanismo é possível fazer com que a recepção de um novo sinal seja feita a qualquer momento, sem necessidade do usuário realizar uma ação prévia.

O processo começa por uma validação, caso a interrupção seja acionada mas a placa esteja no meio de uma emissão ou recepção de sinal. Nesta situação o sinal é descartado, caso contrário recepção é iniciada. Iniciar a recepção significa que é habilitado um *timer* do *ESP8266*, este *timer* chama a função de recepção numa frequência ao redor de 300 kHz, que garante mais estabilidade no envio do sinal infravermelho e também é acionada a *flag* de recepção

No laço do *timer*, a primeira tarefa é validar se já passou o período necessário desde a última leitura, o tempo que está sendo utilizado é 130 us (7,69 kHz), valor que é facilmente parametrizável através de *defines* no código-fonte. Passando esta validação, o tempo atual e o valor lido do sensor *IR* são armazenados em memória, sendo este último num vetor.

Em seguida é realizada mais uma checagem, se o valor lido é '1', ou seja, desligado. Caso sim, realizamos um incremento na variável que controla a quantidade recebida de '1's, caso contrário, esta variável é zerada. Este mecanismo serve para identificar quanto tempo o sensor está sem receber um sinal, se a contagem passar de 500, é considerado que a recepção foi finalizada.

No final da recepção informações sobre o tamanho do comando recebido é exibido na porta *Serial* e salvo em memória (Figura 2.4.3.1), o *timer* é desabilitado e também funções de compressão do sinal são executadas. As funções de compressão servem para diminuirmos em oito vezes o espaço necessário em

memória para o armazenamento do sinal, este valor é alcançado através do armazenamento de oito leituras de sinal num único *byte*, como podemos ver na Figura 2.4.3.2.

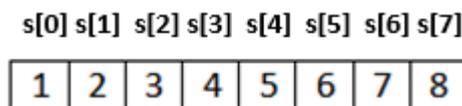
O fluxo detalhado do processo de recepção pode ser visto na Figura 2.4.3.3.

Figura 2.4.3.1 - Relatório de recepção na porta *Serial*.

```
IR Signal Detected!
IR Received sucessfully. Time: 246760 TimePerCycle: 131 Size: 1870
```

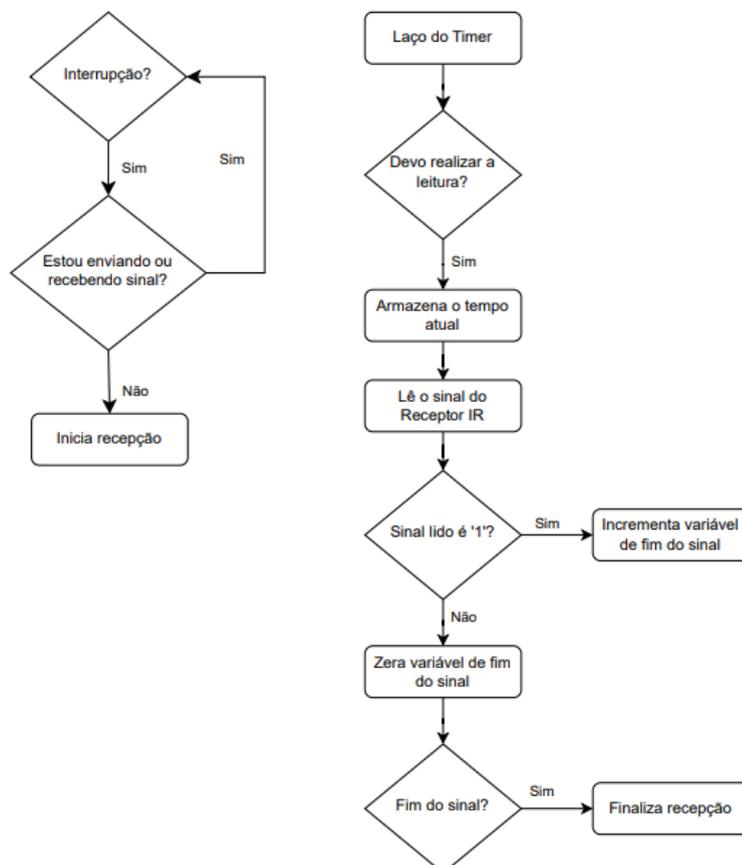
Fonte: Própria (2024).

Figura 2.4.3.2 - Armazenamento do Sinal *IR*.



Fonte: Própria (2024).

Figura 2.4.3.3 - Fluxograma de recepção *IR*.



Fonte: Própria (2024).

2.4.4 Emissão de sinal IR

Para realizar a emissão de comandos via radiação infravermelha é necessário que o usuário realize esta requisição através da página *HTTP* disponibilizada. É possível emitir um sinal que foi salvo no *SPIFFS*, e também o último sinal que foi recebido ou enviado. Se o sinal estava salvo, há um procedimento prévio para realizar a leitura dos dados no *SPIFFS*, descomprimí-lo e armazená-lo em memória.

Após garantir que o sinal está pronto para emissão, é realizada uma validação, assim como vimos na recepção, para avaliar se um sinal já está sendo enviado ou recebido no momento, caso não, se inicia o processo de recepção, que consiste em habilitar um *timer* com as mesmas especificações tratadas na seção anterior.

O início do laço de envio consiste em validar se já é o momento de emitir o próximo dado, ou seja, se já se passaram 130 us desde o último envio. Caso sim, o tempo atual é armazenado em memória, é feita uma leitura do sinal em memória na posição atual do arranjo, e este é enviado ao circuito integrado *NE555P*, que realiza a modulação do sinal.

Por fim, é feita uma validação se este era o último dado do arranjo, quando negativo, é incrementada a variável de posição atual e quando positivo são realizados os procedimentos de finalização da emissão, que se resumem a exibir na interface *Serial* informações sobre o envio (Figura 2.4.4.1) e desabilitar o *timer*. A partir deste momento o *SoC* já está disponível para realizar outras tarefas.

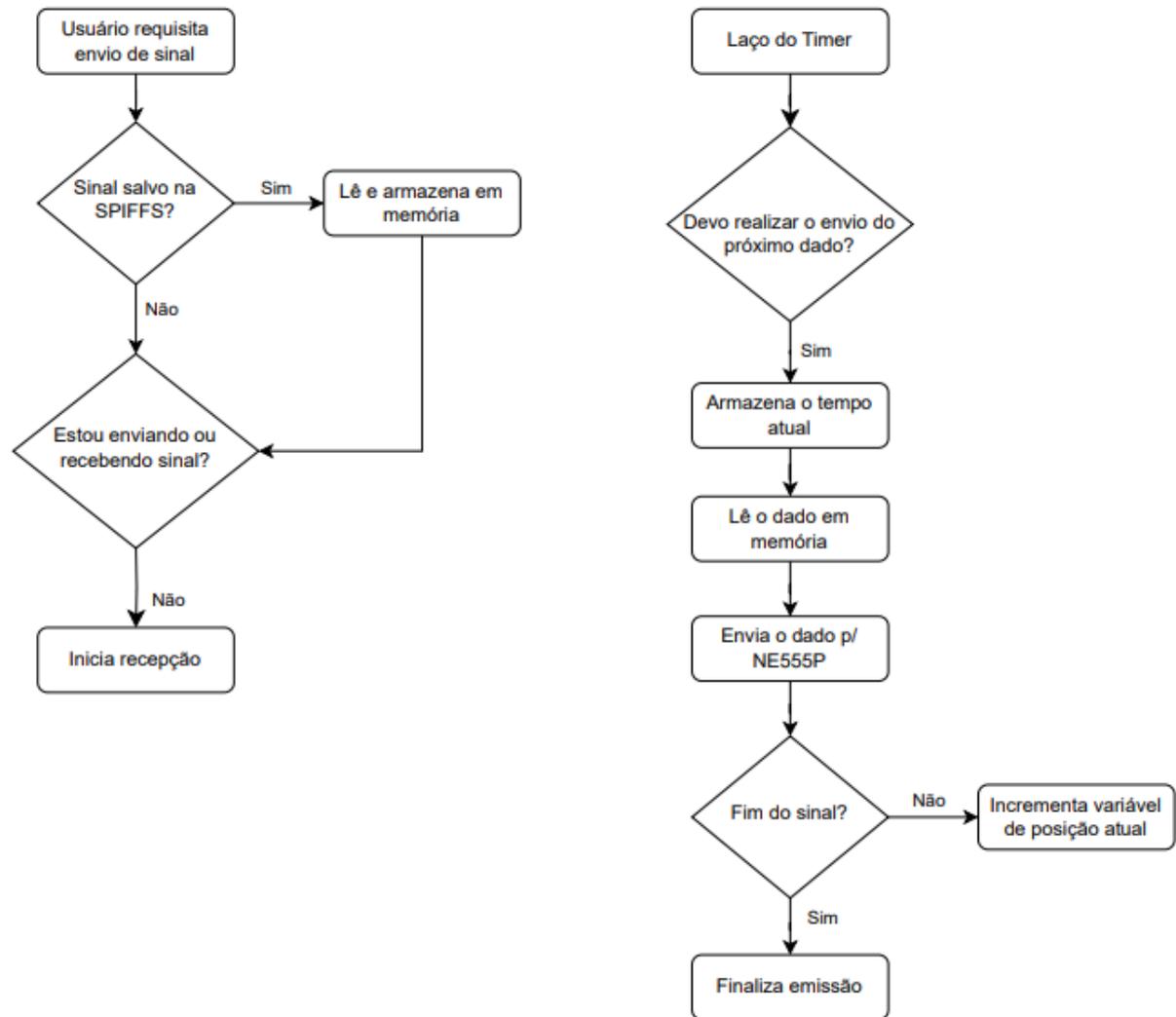
Todo este fluxo pode ser visto na Figura 2.4.4.2.

Figura 2.4.4.1 - Relatório de emissão na porta *Serial*.

```
Loading signal from /FRIO 24C:  
Send (via web request) !  
IR Emitted sucessfully. Time: 180204 TimePerCycle: 131.
```

Fonte: Própria (2024).

Figura 2.4.4.2 - Fluxograma de emissão IR.



Fonte: Própria (2024).

2.4.5 Inicialização

Este é o processo de *boot* do microcontrolador. Por conta da conectividade *Wi-Fi*, possui duas variações que serão vistas a seguir.

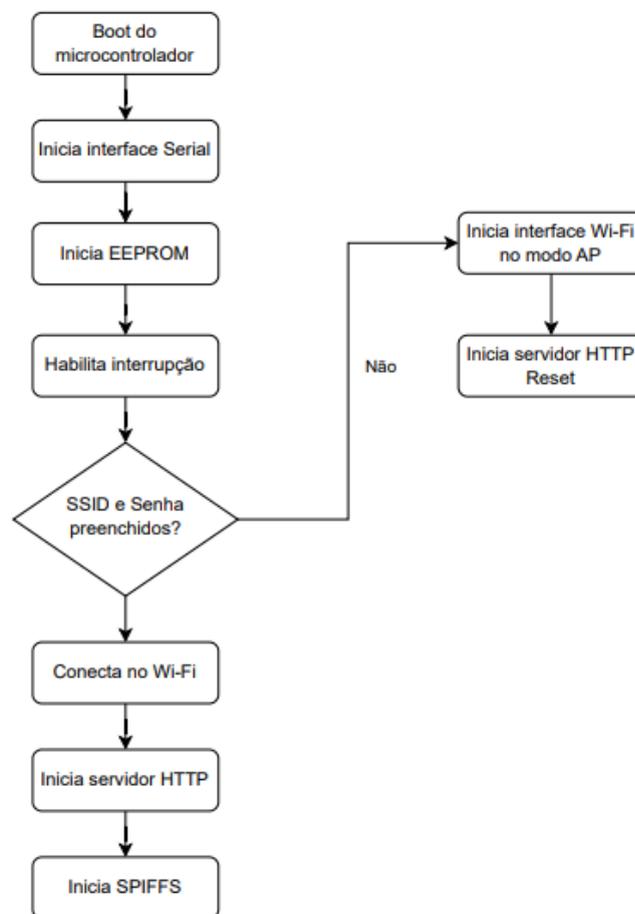
Após o equipamento o SoC ser alimentado, inicia-se o processo, que consiste em habilitar a interface *Serial*, interrupção para o *reset* do equipamento e, por fim, a memória *EEPROM*. Esta última é utilizada para realizar o armazenamento não volátil dos dados de conectividade *Wireless*: *hostname*, *SSID* e senha.

Se não houver informação (*SSID* e senha vazios) a inicialização entra no modo *reset*, ou seja, é considerado que placa está resetada de fábrica, sem informações. Neste caso, o *Wi-Fi* é iniciado no modo *AP* (ponto de acesso) e é

iniciado um servidor *WEB* específico para configuração da placa, que permite o usuário realizar a configuração dos dados *Wi-Fi*.

Caso os dados estiverem preenchidos, o microcontrolador se conecta na rede *Wireless*, é iniciado o servidor *HTTP* padrão (com possibilidade de armazenamento e emissão de sinais) e, para finalizar, é habilitado o *SPIFFS*, sistema de arquivos onde são armazenados os sinais.

Figura 2.4.5.1 - Fluxograma de *boot* do microcontrolador.



Fonte: Própria (2024).

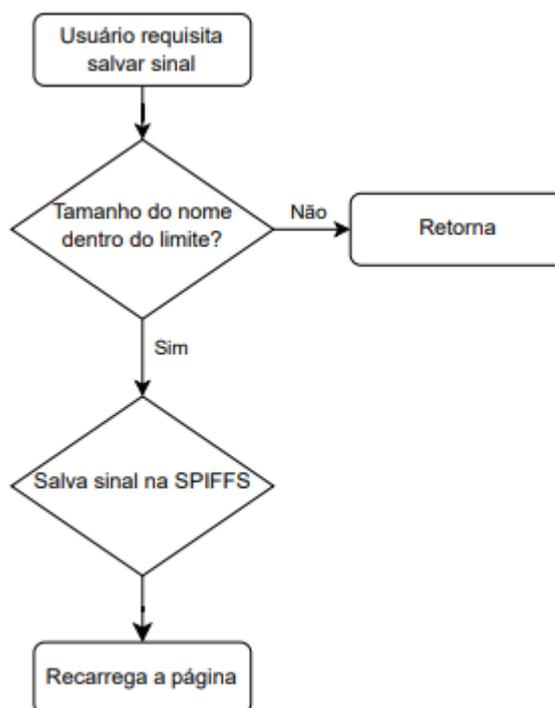
2.4.6 Armazenamento de sinal *IR*

A possibilidade de salvar um sinal para emissão posterior é uma funcionalidade essencial do projeto. Isso pode ser feito pelo usuário através da

página *WEB* ao pressionar o botão “Salvar último sinal” (Figura 2.4.6.1), localizado no topo, e inserir o nome desejado.

Ao enviar a requisição para o microcontrolador, o mesmo valida se o nome do sinal está dentro do tamanho limite de 32 caracteres. Caso não, a operação é cancelada, caso sim, é executada a função de adição de um novo arquivo no *SPIFFS* e a página é recarregada para que o usuário visualize o comando recém salvo.

Figura 2.4.6.1 - Fluxograma de armazenamento de sinal.



Fonte: Própria (2024).

2.4.7 Interface *WEB*

A interface *WEB* foi construída com as linguagens *HTML*, *CSS* e *Javascript*, utilizando o *Bootstrap*. O *Bootstrap*, criado por desenvolvedores do *Twitter*, é um *framework front-end* muito conhecido que auxilia no desenvolvimento de páginas *WEB* modernas e responsivas através do fornecimento de uma diversa gama de componentes com estilos pré-definidos.

A utilização dele, além de facilitar a definição do *design* da página, poupou uma vasta quantidade de memória *flash* do SoC, pois todas as regras de negócio do *framework* são disponibilizadas através de uma *CDN* (content delivery network), ou seja, os arquivos são hospedados em um servidor externo na *internet* e o *download* deles é feito de forma automática pelo navegador do usuário ao abrir a página.

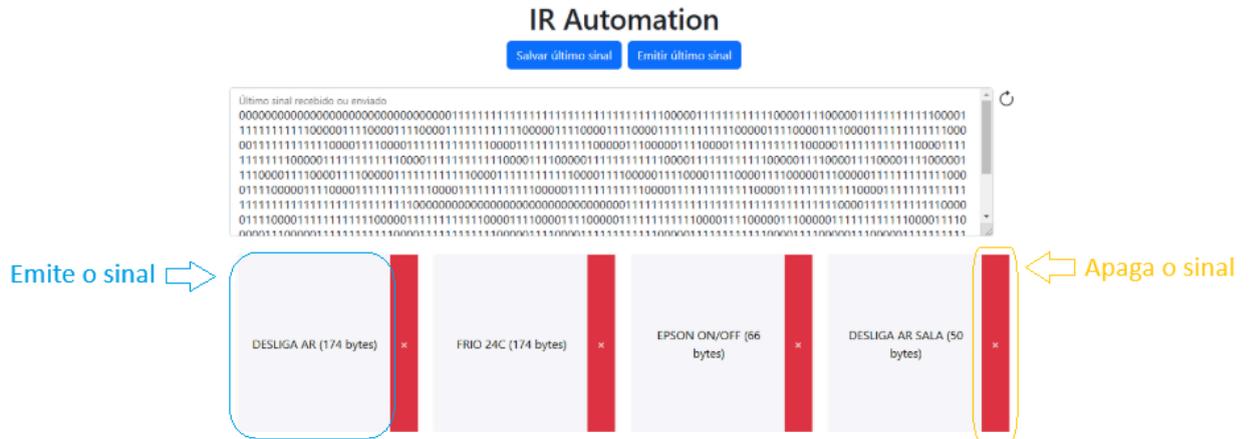
Como já comentado na Seção 2.4.5, o projeto possui dois modos de operação. Quando a placa está resetada de fábrica, ou seja, sem dados de conexão *Wi-Fi* na memória *EEPROM*, é iniciado o servidor *WEB* no modo “resetado”. Este modo conta com uma página bem mais simples (Figura 2.4.7.2), onde o usuário deve apenas inserir os dados do seu *Wi-Fi*. Inclusive, neste modo a página não conta com o *Bootstrap*, visto que o usuário, na maioria das vezes, não tem acesso a *internet* quando está conectado à rede disponibilizada pela placa.

Quando, no *boot*, é identificado que os dados *Wi-Fi* estão armazenados, é inicializado o servidor *WEB* padrão, que fornece a página mais completa ao usuário, permitindo realizar as ações relacionadas a recepção e envio de sinais infravermelho. Na Figura 2.4.7.1 é possível visualizar como ficou a distribuição dos elementos. No topo temos os botões “Salvar último sinal”, cujo funcionamento foi detalhado na Seção 2.4.6 e “Emitir último sinal” que, como o nome diz, irá emitir o último sinal infravermelho em memória, independente se foi emitido ou recebido.

Na área de texto abaixo, é listado o sinal demodulado em memória. Este é o sinal que pode ser armazenado ou emitido nos botões recém comentados. Por fim, são exibidos botões com todos os sinais salvos na *SPIFFS*, este sendo o componente mais complexo de toda a página.

Para gerar este componente foi necessário criar um *HTML* dinâmico, onde, para cada sinal lido no sistema de arquivos, é acrescentada uma parte do código dentro da *String*. Em cada botão é possível, no lado esquerdo, emitir o sinal e, no lado direito realizar a exclusão do mesmo. Da maneira que foi desenvolvido não é possível realizar a ordenação manual deles.

Figura 2.4.7.1 - Página *WEB* quando o dispositivo está configurado.



Fonte: Própria (2024).

Figura 2.4.7.2 - Página *WEB* quando o dispositivo está resetado.

IR Automation

Hostname:

SSID:

Senha:

Enviar

* O tamanho máximo é de 32 caracteres.

* Todos os campos devem ser preenchidos.

Fonte: Própria (2024).

Um ponto relevante que vale ser comentado é sobre o manuseio através do aparelho móvel. Como abordado no início desta Seção, foi utilizado o *Bootstrap* e ele é um *framework mobile-first*, ou seja, todos os seus componentes foram desenvolvidos com foco na utilização via dispositivos móveis. Com isso, a página pode ser utilizada através de celulares sem nenhuma perda de informação. Um exemplo pode ser visto na Figura 2.4.7.3.

Figura 2.4.7.3 - Página *WEB* via dispositivos móveis.

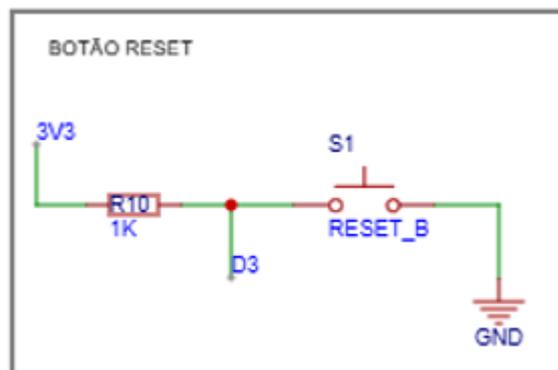
2.5.1 Esquemático

O desenvolvimento do esquemático foi baseado nas diversas iterações de testes e nas recomendações dos fabricantes dos componentes. As subseções a seguir foram divididas de acordo com os blocos do esquemático, cujo documento completo pode ser visualizado no APÊNDICE - A.

2.5.1.1 Botão reset

A funcionalidade do botão de *reset* é zerar a memória *EEPROM* do SoC, onde os dados de conectividade *Wi-Fi* são armazenados. Foi utilizado o próprio botão *flash* da Placa de Desenvolvimento (Figura 2.1.1.1) para o propósito, e o esquemático referente a ele (Figura 2.5.1.1.1) foi adicionado utilizando como base o *datasheet* oficial da placa.

Figura 2.5.1.1.1 - Esquemático - Botão reset.

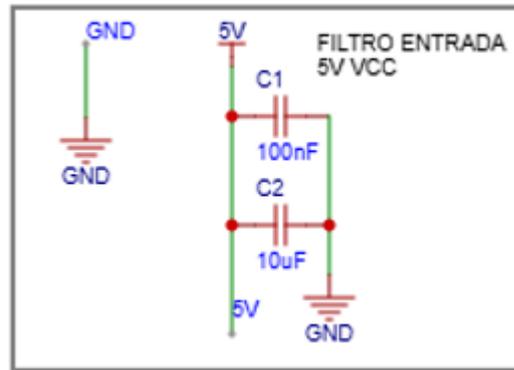


Fonte: Própria (2024).

2.5.1.2 Filtro de entrada

Durante o desenvolvimento do projeto surgiram problemas no envio de comandos IR, que estavam inconsistentes. Após diversos testes, identificou-se que o projeto estava sendo afetado por ruídos na linha de VCC, problema que foi solucionado com a adição de capacitores na entrada do circuito (Figura 2.5.1.2.1). Vale ressaltar que a alimentação vem de um conector *USB-C* (Figura 2.1.8.1).

Figura 2.5.1.2.1 - Esquemático - Filtro de entrada.

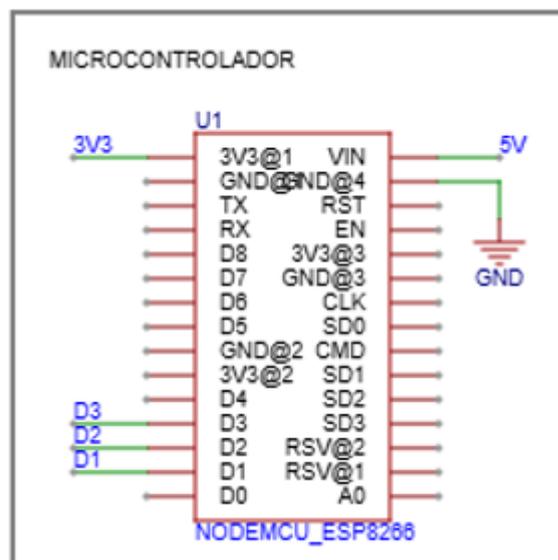


Fonte: Própria (2024).

2.5.1.3 Microcontrolador

Em relação a Placa de Desenvolvimento *ESP-12E*, não foram necessárias a utilização de muitos pinos. Os *GPIOs D1* e *D2* foram utilizados para conexão ao *TSOP4838* e *NE555P*, respectivamente, e o *D3* ficou em espera para *debug*. Sobre a alimentação, foi utilizado o 5V comentado na Seção 2.5.1.2.

Figura 2.5.1.3.1 - Esquemático - Microcontrolador.

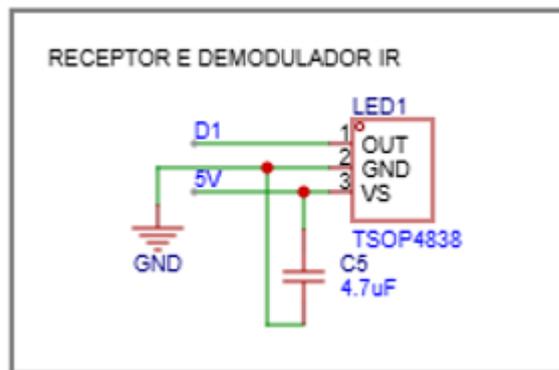


Fonte: Própria (2024).

2.5.1.4 Receptor e demodulador IR

O circuito utilizado para o pleno funcionamento do receptor e demodulador IR TSOP4838 pode ser visualizado na Figura 2.5.1.4.1. A adição do capacitor foi necessária para diminuir a entrada de ruídos e consta como recomendação no *datasheet* oficial do componente.

Figura 2.5.1.4.1 - Esquemático - Receptor e demodulador IR.



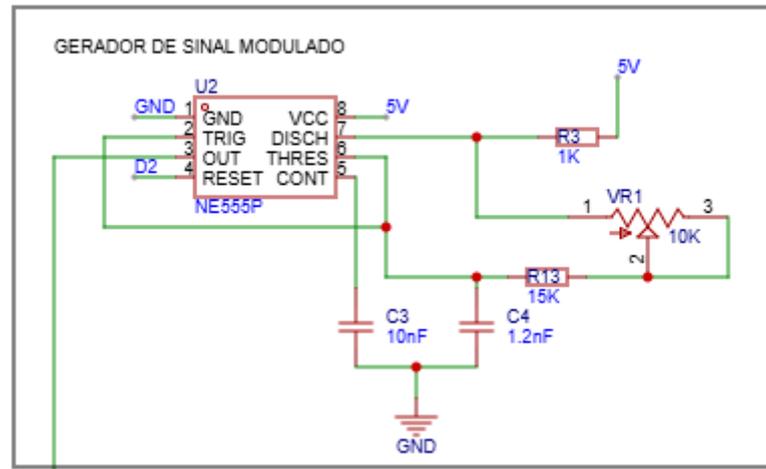
Fonte: Própria (2024).

2.5.1.5 Gerador de sinal modulado

Após diversos testes foi visto que o *ESP8266* não possui o poder de processamento necessário para realizar a modulação do sinal em 38 kHz e, então, a emissão foi dividida em duas etapas, primeiramente o SoC envia o sinal demodulado ao componente *NE555P* e este realiza a modulação do sinal, que é encaminhada aos *transistores* de acionamento dos *LEDs*.

O *NE555P*, como visto na Seção 2.1.4, é um oscilador cujas características atendem os requisitos do projeto, o seu pino *reset* foi utilizado para realizar efetivamente a modulação do sinal e está diretamente conectado ao pino *D2* do microcontrolador. O ajuste da frequência de saída é feito através dos capacitores e resistores conectados nos pinos cinco, seis e sete. Para o ajuste fino foi utilizado um potenciômetro de 10K (*VR1*) e um osciloscópio.

Figura 2.5.1.5.1 - Esquemático - Gerador de sinal modulado.



Fonte: Própria (2024).

2.5.1.6 Emissor IR 360 graus

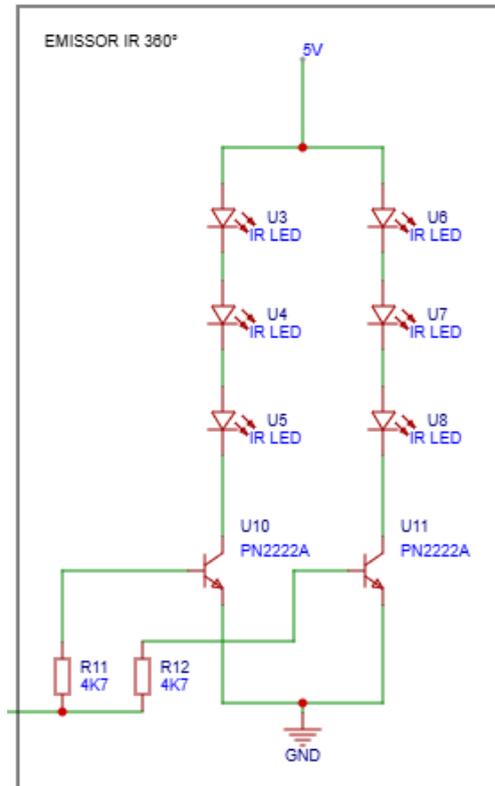
Para o acionamento dos *LEDs* infravermelho foi utilizado dois arranjos, onde cada um deles conta com três diodos em série. Como visto na Seção 2.1.3, eles possuem ângulo de emissão de 60 graus, portanto, a utilização de seis garante a transmissão de dados *IR* em todas as direções.

Cada um dos arranjos possui o seu *transistor* próprio, no caso o *PN2222A*, visto na Seção 2.1.5. Por serem *transistores* do tipo *BJT*, o acionamento deles ocorre por corrente, e esta característica é de suma importância quando estamos trabalhando num circuito de baixa tensão, neste caso, 5V.

A utilização de três diodos em série poupou a necessidade de adicionar um resistor extra nos arranjos, ou utilizar três arranjos, por exemplo. Em medições realizadas com os *LEDs* acionados continuamente foi lida a corrente de 125 mA em cada arranjo, que, para este uso, seria inviável, visto que os diodos possuem uma corrente nominal máxima de 50 mA.

Entretanto, no projeto, os arranjos trabalham para emissão de dados, numa portadora de 38 kHz e *duty cycle* pouco superior a 50%, além de cada transmissão durar, na média, poucos milésimos de segundo. Levando estes pontos em consideração, é seguro trabalhar com estas características.

Figura 2.5.1.6.1 - Esquemático - Emissor IR 360 graus.



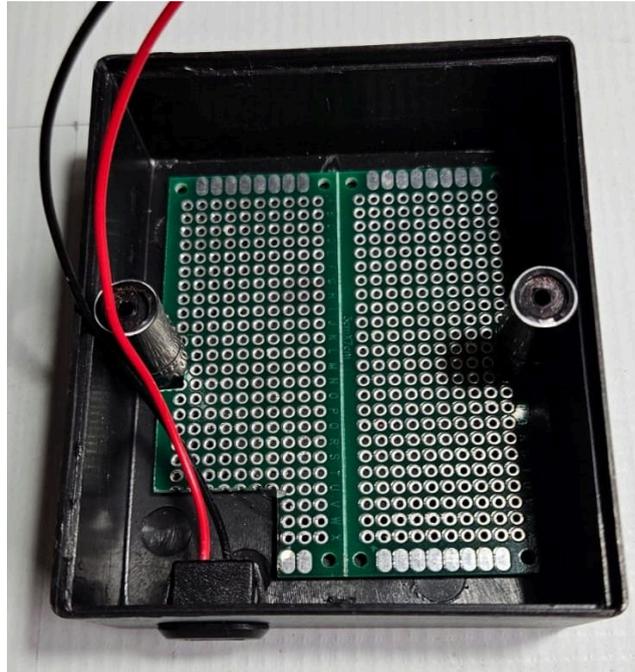
Fonte: Própria (2024).

2.5.2 Processo de montagem

Após a extensa etapa de validação do esquemático e do *firmware*, onde os componentes estavam todos montados numa *protoboard*, iniciou-se o processo de montagem. O planejamento é realocar todo o circuito da *protoboard* numa pequena caixa plástica, comentada na Seção 2.1.9.

Na Figura 2.5.2.1 é possível visualizar a caixa já com o conector *USB-C*, para alimentação, posicionado, e cortes na placa de circuito impresso universal para que possa ser posicionada dentro da caixa.

Figura 2.5.2.1 - Montagem - Parte 1.



Fonte: Própria (2024).

Na Figura 2.5.2.2 as placas de *CI* já estavam posicionadas dentro da caixa e, surgiu a ideia de adicionar uma cúpula na parte superior dela, onde ficariam posicionados os *LEDs* e receptor *IR*. Ela foi reaproveitada de uma lâmpada queimada, e o seu formato deixou o projeto com um visual único. Após posicionada a, o projeto ficou como na Figura 2.5.3.3.

Figura 2.5.2.2 - Montagem - Parte 2.



Fonte: Própria (2024).

Figura 2.5.2.3 - Projeto - Externo 1.



Fonte: Própria (2024).

Para que, após montado, ainda fosse possível realizar *debugs* a nível de Firmware, a placa de desenvolvimento foi posicionada na lateral da caixa, dessa forma um corte foi feito para que a porta *Micro USB* ficasse disponível externamente. É possível visualizar o resultado na Figura 2.5.2.4.

Figura 2.5.2.4 - Projeto - Externo 2.

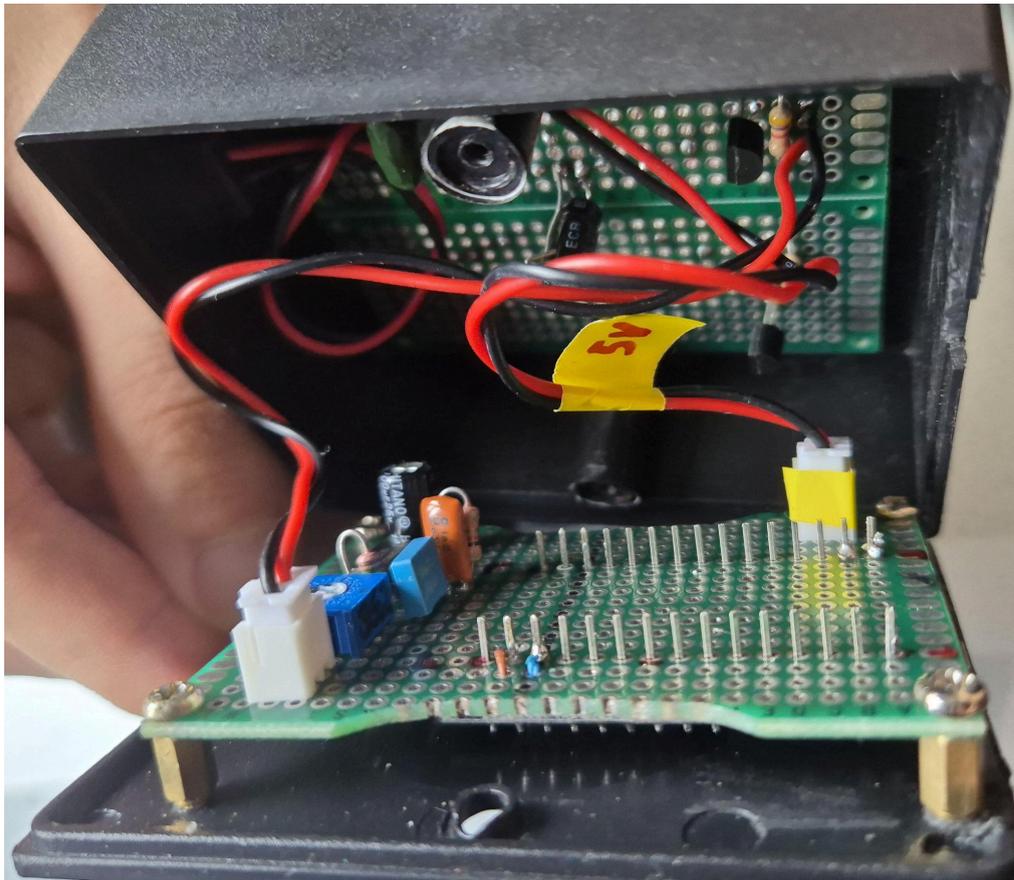


Fonte: Própria (2024).

No total foram necessárias quatro placas de circuito integrado universal. Na parte inferior do projeto ficaram posicionados o microcontrolador e o gerador de sinal modulado. Já a parte superior, próxima a cúpula, ficou dedicada a chegada do 5V, já com os capacitores para filtro, via porta *USB-C*, e a emissão de sinal infravermelho 360 graus.

Para facilitar a montagem, foram utilizados dois conectores nas placas inferiores, a chegada do *VCC*, marcado com fita amarela, e a saída do sinal modulado gerado pelo *NE555P*, que é consumido pelos *transistores* que fazem o chaveamento dos *LEDs*. Na Figura 2.5.2.5 é exibido o interior da caixa, já com todos os componentes instalados.

Figura 2.5.2.5 - Projeto - Interno.



Fonte: Própria (2024).

Outra definição realizada, foi o posicionamento do microcontrolador com a sua parte superior apontada para o face inferior da caixa. Isto permitiu que, através de cortes, fosse possível aproveitar os dois botões já presentes na placa de desenvolvimento, *flash*, utilizado para limpar a memória *EEPROM*, e *reset*, que basicamente realiza uma reinicialização forçada do *SoC*.

Na Figura 2.5.2.6 vemos a parte inferior do projeto, ao lado esquerdo o botão utilizado para realizar o *reboot* do *ESP8266*, este pode ser acionado com o dedo, e a direita o botão que limpa a memória, cujo acionamento pode ser feito com um *clips*.

Figura 2.5.2.6 - Projeto - Inferior.



Fonte: Própria (2024).

3. RESULTADOS

Neste capítulo são apresentados os resultados obtidos no projeto, em testes realizados após a sua conclusão. Cada teste está localizado em sua respectiva seção.

3.1 Armazenamento de Sinal IR

Conforme a Seção 2.4.6, o armazenamento de sinais infravermelho é feito no sistema de arquivos *SPIFFS* disponibilizado pelo microcontrolador *ESP8266*. No menu de configurações do *IDE* (Figura 2.4.1.1) há a opção *Flash Size*, que permite definir como a memória *flash* de 4MB será alocada no *SoC*. Foi utilizado o valor padrão que reserva 2MB para *FS (SPIFFS)*.

Com base nos testes realizados, protocolos mais complexos, como os presentes em controles remotos de ar-condicionados, ocupam em torno de 170 *bytes* de memória, pois eles armazenam todo o status do dispositivo, como nível de ventilação, temperatura, modo operação, etc. Aparelhos mais simples, como televisores, projetores e *home-theaters*, já ocupam um valor bem inferior sendo, em média, 60 *bytes*.

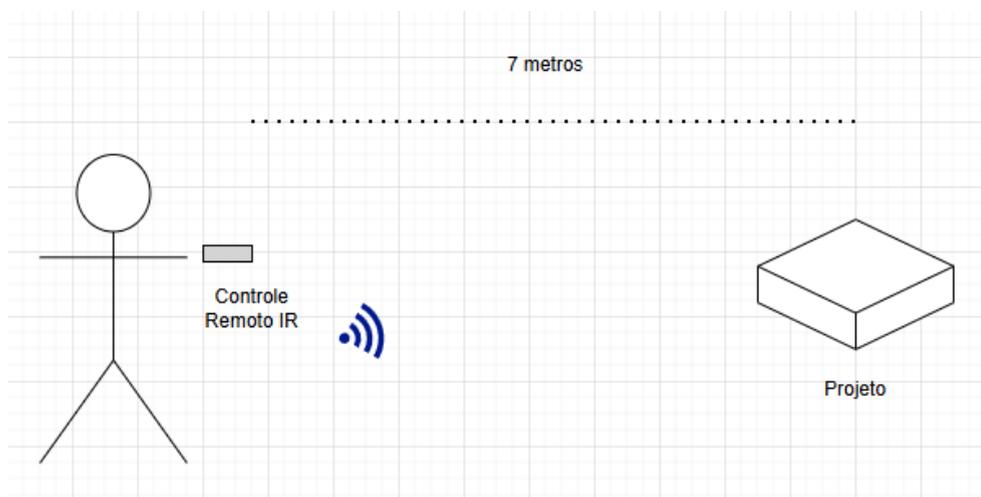
Levando em consideração estes valores, pode-se afirmar que os sinais armazenados pelo usuário ocuparão, na média, 120 *bytes* do sistema de arquivos. Além disso, temos o *overhead* acrescentado pelo próprio *filesystem* que, segundo a documentação da própria *Espressif* é, na média, de 75%, ou seja, dos 2MB, apenas 1,5MB realmente são utilizáveis. Através da divisão do tamanho médio dos sinais pelo espaço em memória utilizável, pode-se concluir que, em média, o usuário conseguirá armazenar 12500 sinais.

Apesar deste valor teórico calculado alto, em testes, foi possível exibir na página *WEB*, no máximo, 31 sinais, pois, ao passar deste valor, a *String* que armazena a seção de sinais da página *HTML* passa a ocupar mais memória do que há disponível no microcontrolador. A implementação de paginação, por exemplo, resolveria este problema, pois desta forma se poderia limitar a quantidade máxima de sinais que são exibidos por vez e, dessa forma, permitindo a utilização de todo o espaço de memória disponível.

3.2 Recepção de Sinal IR

Para esta validação o projeto foi posicionado numa mesa na altura de noventa centímetros do piso e uma pessoa apontou o controle remoto de um ar-condicionado diretamente a cúpula onde o receptor infravermelho está posicionado. A distância máxima que o microcontrolador conseguiu receber e armazenar corretamente o sinal foi de sete metros. A ilustração do teste pode ser visualizada na Figura 3.2.1.

Figura 3.2.1 - Ilustração do teste de recepção.

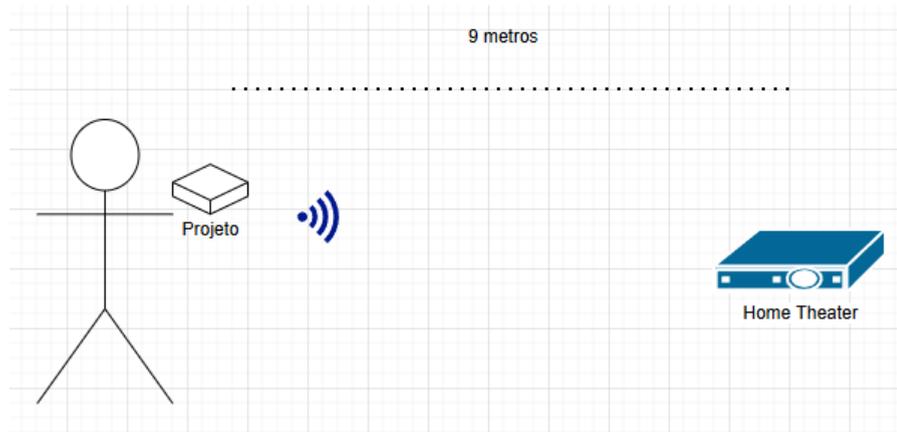


Fonte: Própria (2024).

3.3 Emissão de Sinal IR

Neste teste, o projeto ficou na mão de uma pessoa e apontado diretamente a um *home theater*. Enquanto isso, outra pessoa ficou um celular realizando a emissão de sinais liga/desliga ao aparelho. A distância máxima que o *home theater* conseguiu receber e decodificar corretamente o sinal enviado pelo projeto foi de nove metros. A ilustração do teste pode ser visualizada na Figura 3.3.1.

Figura 3.3.1 - Ilustração do teste de recepção.



Fonte: Própria (2024).

3.4 Consumo de Energia

O consumo energético da placa de desenvolvimento é bem baixo e, enquanto nenhum sinal é enviado, é basicamente ela que está operando. Através de um medidor *USB* (Figura 3.4.1) podemos notar que, em *idle*, o projeto consome, em média, apenas 100mA.

O projeto foi conectado a um *powerbank USB* de 15600mAh totalmente carregado e a duração da bateria foi de 98 horas.

Figura 3.4.1 - Medidor de potência *USB*.



Fonte: Própria (2024).

4. CONCLUSÃO

Este trabalho explorou a radiação infravermelha, abordando desde seu descobrimento e suas aplicações nos séculos passados até seu uso nos dias atuais. Foram analisados como a transmissão de dados ocorre por meio dessa tecnologia e os protocolos desenvolvidos para otimizar e padronizar a comunicação.

Através do desenvolvimento do controle remoto universal *wireless* utilizando o SoC *ESP8266* foi possível atingir o principal objetivo do projeto, dar uma nova vida a aparelhos eletrônicos mais antigos que não possuem a tecnologia *Wi-Fi* integrada e, portanto, não permitem o seu manuseio de forma remota através de dispositivos móveis ou *desktops*.

O *ESP8266* demonstrou ser um SoC de fácil utilização, altamente versátil, e sua integração com a *IDE* do Arduino proporcionou uma experiência de desenvolvimento do firmware fluida. No entanto, durante o desenvolvimento, foi identificado que o microcontrolador não apresentava desempenho suficiente para realizar simultaneamente a modulação do sinal infravermelho a 38 kHz, a conexão *Wi-Fi* e a operação de um servidor *HTTP*. Uma solução simples seria a substituição pelo *ESP32*, um SoC mais moderno com processador dual-core. Entretanto, alinhando-se ao objetivo do projeto de evitar obsolescência, optou-se por continuar utilizando o *ESP8266*, com a adição do *CI NE555P*, dedicado exclusivamente à modulação do sinal.

O desenvolvimento do *firmware*, que demandou o pleno entendimento do da transmissão de dados via radiação infravermelha, foi um grande desafio, pois basta a mudança de um único *bit* para que o dado enviado não seja devidamente lido pelo receptor, e o *debug* só pode ser feito através da utilização de um osciloscópio.

A etapa de resultados comprovou o funcionamento do projeto, que é capaz de emitir e receber dados via infravermelho até sete e nove metros de distância, respectivamente.

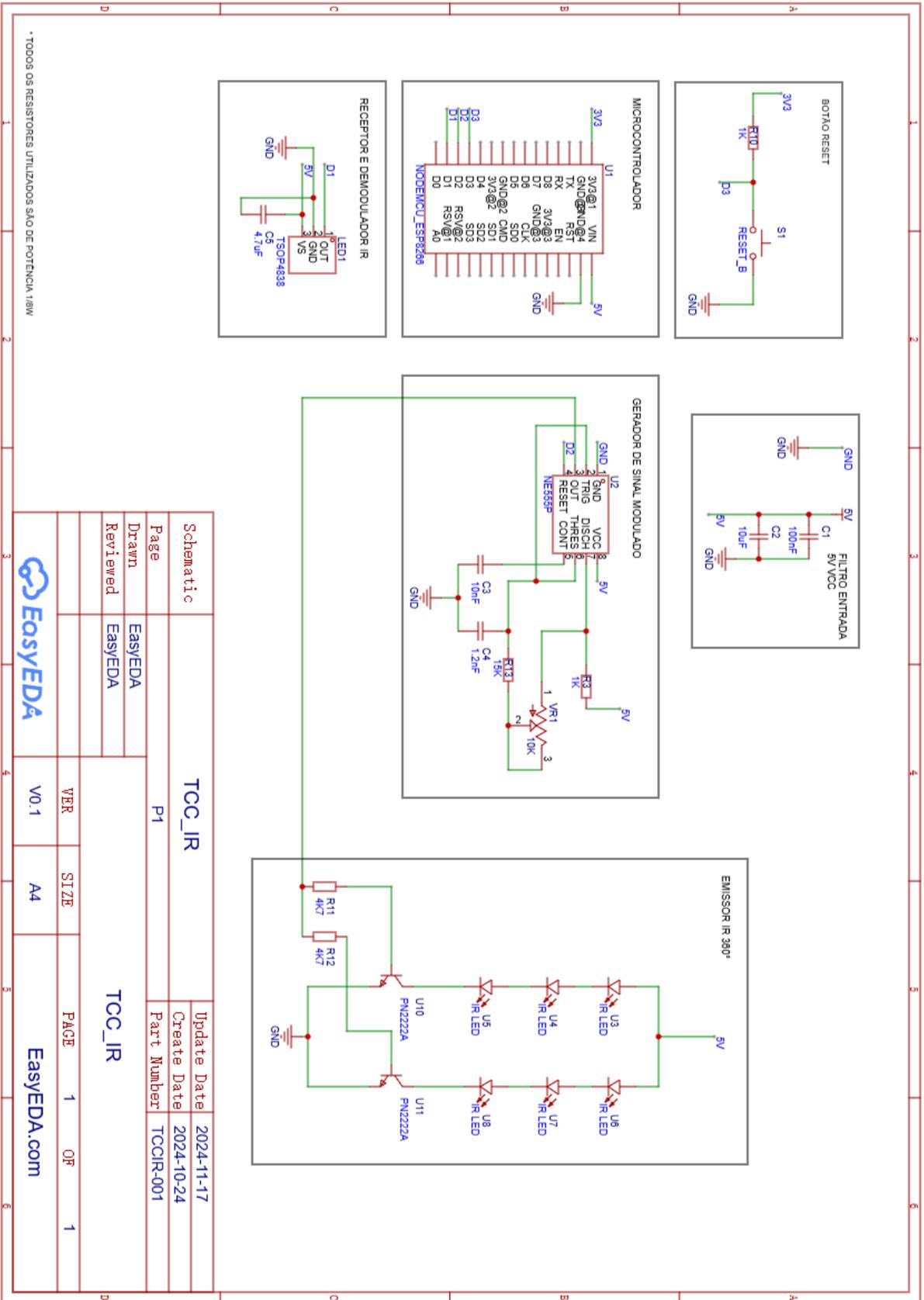
Entre as sugestões de melhorias para projetos futuros, destacam-se a implementação de paginação na lista de sinais, permitindo a exibição de mais de trinta sinais armazenados, o desenvolvimento de uma aplicação *WEB* externa ao microcontrolador, hospedada em um servidor na nuvem, com design mais moderno, além da possibilidade de compartilhamento de sinais entre dispositivos. Todas essas

ideias podem ser implementadas com base na estrutura construída durante este projeto. Por fim, conclui-se que os objetivos propostos foram alcançados.

REFERÊNCIAS

- SB-PROJECTS. **Infrared Remote Control**. Disponível em: <https://www.sbprojects.net/knowledge/ir>. Acesso em: 10 nov. 2024.
- WHITE, J. **Herschel and the Puzzle of Infrared**. Disponível em: <https://www.americanscientist.org/article/herschel-and-the-puzzle-of-infrared>. Acesso em: 11 nov. 2024.
- BESCHLOSS, S. **Object of Interest: Remote Control**. Disponível em: <https://www.newyorker.com/tech/annals-of-technology/object-of-interest-remote-control>. Acesso em: 11 nov. 2024.
- ESPRESSIF. **ESP8266EX Datasheet**. Disponível em: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. Acesso em: 14 nov. 2024.
- TEXAS INSTRUMENTS. **NE555P Datasheet**. Disponível em: <https://datasheet.octopart.com/NE555P-Texas-Instruments-datasheet-7284017.pdf>. Acesso em: 14 nov. 2024.
- VISHAY SEMICONDUCTORS. **TSOP48: IR Receiver Modules for Remote Control Systems**. Disponível em: <https://www.vishay.com/docs/82459/tsop48.pdf>. Acesso em: 15 nov. 2024.
- SINGH, R.; KAPOOR, A.; SINGH, R. **An overview of universal remotes: technologies and applications**. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S221478532036702X>. Acesso em: 17 nov. 2024.
- GETBOOTSTRAP. **Bootstrap**. Disponível em: <https://getbootstrap.com>. Acesso em: 18 nov. 2024.

APÊNDICE A. Esquemático da Placa de Circuito Impresso Universal



Schematic		TCC_IR	
Page	P1	Update Date	2024-11-17
Drawn	EasyEDA	Create Date	2024-10-24
Reviewed	EasyEDA	Part Number	TCCIR-001
VER		TCC_IR	
SIZE	A4	PAGE	1
ORF			1



EasyEDA.com