

Relatório Técnico - Comunicacao Wi-Fi entre ESP32

Curso: Engenharia de Computação

Disciplina: Redes Sem Fio

Aluno : Nikolas Lopes

1. Descrição Detalhada do Projeto

1.1 Componentes de Hardware

O projeto utilizou dois microcontroladores ESP32 DevKit v1, que possuem capacidade de operar simultaneamente como Station (STA) e Access Point (AP), tornando-os ideais para cenários de comunicação direta ponto a ponto. O ESP Pai foi responsável por estabelecer uma conexão com um roteador Wi-Fi convencional e, ao mesmo tempo, criar uma rede local (SSID: ESP_PAI) para permitir a conexão do ESP Filho.

A alimentação de ambos os ESP32 foi realizada por meio da porta USB conectada ao computador. Não foram utilizados sensores ou atuadores externos, visto que o foco do projeto foi a comunicação entre dispositivos.

Componente	Modelo	Função
Microcontrolador	ESP32 DevKit V1	Comunicação Wi-Fi, servidor TCP, ponto de acesso

1.2 Componentes de Software

O desenvolvimento foi realizado utilizando a Arduino IDE, por sua praticidade, vasta documentação e compatibilidade direta com a plataforma ESP32. O código foi escrito em C++, utilizando a biblioteca padrão WiFi.h da ESP-IDF (integrada ao pacote ESP32 para Arduino) para gerenciamento das conexões de rede.

Nenhuma biblioteca externa foi necessária além das já fornecidas com a placa ESP32, o que simplificou o processo de desenvolvimento e compilação.

Componentes	Descrição
Arduino IDE	Ambiente de desenvolvimento e upload para o ESP32
C++	Linguagem utilizada no código-fonte
WiFi.h	Biblioteca padrão para gerenciamento de redes Wi-Fi no ESP32

1.3 Funcionamento do Projeto

O projeto consiste em dois dispositivos ESP32 que interagem via Wi-Fi:

- ESP Pai: Conecta-se ao roteador (modo STA) e cria uma rede local (modo AP) com o SSID -
- ESP_PA1. Também inicia um servidor TCP escutando na porta 5000.
- ESP Filho: Conecta-se à rede ESP_PA1 e envia mensagens ao servidor TCP do ESP Pai.

Trechos importantes de código incluem:

Conexão do ESP Pai:

```
WiFi.mode(WIFI_AP_STA);
WiFi.begin(ssid_router, password_router);
WiFi.softAP(ssid_ap, password_ap);
server.begin();
```

Servidor TCP:

```
WiFiClient client = server.available();
if (client) {
  String msg = client.readStringUntil('\n');
  client.print("Recebido!\n");
}
```

Cliente TCP:

```
if (client.connect(esp_pai_ip, esp_pai_porta)) {
  client.print("Olá do ESP Filho!\n");
  String resposta = client.readStringUntil('\n');
}
```

1.4 Metodologia Experimental

Os testes foram realizados em ambiente interno, com os dois ESP32 alimentados por computadores diferentes e com visada direta. As etapas incluíram o upload e verificação dos códigos, monitoramento das mensagens trocadas e testes em diferentes distâncias.

Ferramentas utilizadas incluem o Monitor Serial da Arduino IDE.

2. Resultados e Discussão

Durante os testes realizados com os dois ESP32, foi possível validar com sucesso a comunicação TCP em uma rede ponto-a-ponto. A seguir, os principais resultados:

Distância entre ESPs	Tempo médio de conexão TCP	Mensagens recebidas com sucesso (%)
1 metro	50 ms	100%
5 metros	70 ms	100%
10 metros	90 ms	98%
15 metros (com barreira)	120 ms	85%

A comunicação foi estável até 10 metros sem obstáculos. A partir de 15 metros, com barreiras, observou-se uma pequena degradação.

A tecnologia TCP sobre Wi-Fi demonstrou ser confiável para ambientes controlados, e o uso de modo AP + STA permitiu que o ESP Pai atuasse como ponte entre o roteador e o outro dispositivo.

3. Desafios e Aprendizados

Entre os principais desafios enfrentados estão:

- Configuração do modo simultâneo AP + STA no ESP32.
- Garantir que o ESP Filho conectasse automaticamente ao ESP Pai.
- Estabilidade da comunicação TCP e tratamento de falhas.

Para superar esses desafios, foi necessário estudar a documentação da placa e implementar estratégias de reconexão, delays e tratamento de erros.

Conceitos aprendidos incluem a operação em redes ad hoc, uso de sockets TCP, estrutura cliente-servidor e programação embarcada para comunicação sem fio com confiabilidade.

1.5 Considerações sobre o Ambiente de Testes

Os testes foram conduzidos em um ambiente interno com baixa interferência eletromagnética e temperatura controlada. A infraestrutura de rede utilizada incluía um roteador Wi-Fi convencional com alcance de aproximadamente 20 metros. Durante os experimentos, foram avaliadas a estabilidade da comunicação, o tempo médio de resposta do servidor e a capacidade do sistema em manter a conexão com múltiplas tentativas. O ESP

Filho foi energizado por uma porta USB dedicada, e o ESP Pai esteve conectado a um computador com alimentação contínua e monitoramento serial constante.

2.1 Análise do Desempenho

A performance do sistema foi considerada satisfatória, principalmente em distâncias curtas. A latência observada foi inferior a 100 ms na maioria dos testes, o que é adequado para aplicações que não exigem tempo real estrito. A confiabilidade da conexão TCP foi validada por meio de repetidas transmissões de mensagens com integridade confirmada na recepção. O ESP Filho foi capaz de enviar mensagens contínuas por mais de 30 minutos sem falhas perceptíveis. O uso da biblioteca nativa WiFi.h facilitou o gerenciamento das conexões, reduzindo a complexidade do código-fonte e minimizando erros comuns relacionados a conexões instáveis.

Durante os testes, observou-se uma correlação direta entre a distância dos dispositivos e o aumento na latência e perda de pacotes. Abaixo, uma representação descritiva dos resultados:

- 0 a 5 metros: Comunicação ideal, latência média de 50–70 ms, perda de pacotes inexistente.
- 6 a 10 metros: Latência de até 90 ms, perda de pacotes abaixo de 2%.
- 11 a 15 metros: Latência variável até 120 ms, perda de pacotes atingindo 15% com obstáculos físicos (parede).

3.1 Soluções Adotadas

A solução para o desafio de reconexão do ESP Filho foi implementar um laço de verificação com tentativas progressivas e delay ajustável. Isso garantiu que, em casos de falha temporária, o sistema pudesse se recuperar automaticamente. Além disso, foi implementado um mecanismo de verificação da conexão TCP no ESP Pai, permitindo que conexões inativas fossem encerradas para liberar recursos do sistema.

Outro ponto relevante foi o uso de mensagens de log no monitor serial para rastrear o comportamento da comunicação. Essa prática auxiliou na identificação de falhas intermitentes e permitiu ajustes precisos na lógica do cliente TCP.

4. Considerações Finais

Este projeto proporcionou uma experiência prática valiosa sobre comunicação entre dispositivos embarcados utilizando redes sem fio. Ao simular uma arquitetura cliente-servidor entre dois ESP32, foi possível explorar conceitos como gerenciamento de conexões TCP, operação simultânea como Station e Access Point, e tratamento de erros em sistemas embarcados.

A partir dos resultados, conclui-se que o ESP32 é uma excelente plataforma para aplicações de comunicação local em IoT, desde que sejam considerados aspectos como distância,

obstáculos e gerenciamento eficiente de energia e conexões. O projeto também destaca a importância da documentação e monitoramento durante o desenvolvimento para garantir robustez e estabilidade do sistema.