

Relatório - Codificador e Decodificador com Protothreads e TDD

Nícolas Jordani

9 de setembro de 2025

1 Introdução

Este relatório apresenta a implementação de um protocolo de comunicação utilizando *protothreads*. O protocolo segue o formato:

STX (1 B) | QTD (1 B) | DADOS (N B) | CHK (1 B) | ETX (1 B)

Foram desenvolvidas duas protothreads: uma transmissora e uma receptora. Além disso, adotou-se a metodologia de *Test Driven Development (TDD)* para validar cada etapa (cálculo do checksum, montagem do pacote, decodificação correta e detecção de erro).

2 Testes Manuais

A Figura 1 apresenta a execução dos testes unitários. Neles, observamos que:

- O cálculo do checksum foi validado com sucesso;
- A montagem do pacote gera a sequência esperada [STX, QTD, DADOS..., CHK, ETX];
- Pacotes corretos são reconhecidos e geram ACK;
- Pacotes corrompidos são rejeitados com NACK;
- Foi simulado o mecanismo de timeout e retransmissão.

```

==== INICIANDO TESTES ====
Teste checksum: esperado=0x63 obtido=0x63 -> OK
Teste montagem:
  buffer[0] = 0x02
  buffer[1] = 0x02
  buffer[2] = 0xAA
  buffer[3] = 0xBB
  buffer[4] = 0x67
  buffer[5] = 0x03
  -> OK
Teste decodificação correta: resposta=ACK
Teste decodificação com erro: resposta=NACK
Tentativa 1: enviando pacote...
Tentativa 2: enviando pacote...
Tentativa 3: enviando pacote...
Timeout após 3 tentativas -> REENVIO NECESSÁRIO
==== FIM DOS TESTES ====

```

Figura 1: Saída dos testes manuais (TDD).

3 Simulação com Protothreads

A Figura 2 mostra a execução das protothreads. Podemos ver a transmissão byte a byte do pacote, a recepção correta, a validação do checksum e a resposta da receptora enviando o ACK.

```

TX -> 0x02
TX -> 0x10
TX -> 0x30
TX -> 0x03
TX -> 0x00
TX -> 0x00
TX -> 0x00
[RX] Pacote OK! Enviando ACK
[TX] ACK recebido!
[TX] Enviando pacote...
TX -> 0x02
TX -> 0x10
TX -> 0x30
TX -> 0x03
TX -> 0x00
TX -> 0x00
TX -> 0x00
[RX] Pacote OK! Enviando ACK
[TX] ACK recebido!
[TX] Enviando pacote...
TX -> 0x02
TX -> 0x10
TX -> 0x30
TX -> 0x03
TX -> 0x00
TX -> 0x00
TX -> 0x00
[RX] Pacote OK! Enviando ACK
[TX] ACK recebido!

```

Figura 2: Execução da transmissão e recepção com Protothreads.

4 Conclusão

Os testes realizados validaram a implementação do protocolo tanto em nível unitário (TDD) quanto integrado (Protothreads). O sistema foi capaz de detectar pacotes válidos, responder com ACK, e simular retransmissão em caso de erro.