

# Semáforo dos Carros

Integrantes:

Arthur J. C. Brasil - 16855560 - Semáforo de Carros

Júlio Cesar Braga Parro - 16879560 - Semáforo de Pedestres

## Testes Unitários

Implementou-se um semáforo de carros com arquitetura multithread, utilizando quatro threads especializadas. A `thread_controle_fn` coordena as transições entre estados, enquanto as threads `thread_verde_fn`, `thread_amarelo_fn` e `thread_vermelho_fn` gerenciam os tempos específicos de cada cor. A sincronização é garantida por mutex e semáforos, assegurando os tempos definidos de 3 segundos para verde, 1 segundo para amarelo e 4 segundos para vermelho. As enumerações usadas para garantir ótima comunicação entre as threads foram `estado_semaforo_t` e `sinal_transicao_t`.

Foram implementados três modos de operação: normal, noturno e travessia; controlados pela enumeração de tipo `modo_operacao_t`. No modo noturno, o amarelo pisca alternadamente a cada segundo. O botão no pino PTA16 aciona o modo travessia, que realiza transição segura com amarelo por 1 segundo, seguido de vermelho por 4 segundos por meio do uso da transição controlada implementada. Já o botão no pino PTA17 inicia o sistema, garantindo sincronia com outro microcontrolador através de sinal externo.

A comunicação entre threads utiliza sinais de transição e semáforos para coordenação, com máquina de estados bem definida para controle dos modos de operação e transições seguras entre eles.

Com base na implementação desses requisitos, especificados pelo enunciado da atividade, realizou-se um planejamento e execução de diferentes testes unitários, para garantir o funcionamento do semáforo de carros isoladamente, conforme descrito abaixo:

### Caso de teste 1: Ciclo dos LEDs Verde, Amarelo e Vermelho

- Pré-condição: `modo_atual = MODO_NORMAL`, `estado_atual = ESTADO_VERDE`, `travessia_solicitada = false` e `sistema_rodando = true`. Mutex inicializado e threads ativas.

- Etapas de teste: Compilar e executar o código “caso de teste 1”. Observar a sequência de acionamento dos LEDs.
- Pós-condição observada: LED verde acende por cerca de 3 s e apaga; em seguida, LED amarelo acende por cerca de 1s; vermelho acende por cerca de 4 s. Nenhum LED fica aceso simultaneamente.
- Conclusão: A sequência e os tempos de cada LED estão corretos, indicando que as threads e o mutex funcionam conforme esperado.

**Caso de teste 2:** Teste do modo noturno (comportamento isolado)

- Pré-condição: *modo\_atual = MODO\_NOTURNO, estado\_atual = ESTADO\_VERDE, sistema\_rodando = true, travessia\_solicitada = false e deixar comentado o timer\_modo += 100; dentro do while loop da main.*
- Etapas de teste: Executar o código “caso de teste 2”. Observar a piscagem dos LEDs.
- Pós-condição observada: LED amarelo dos carros pisca com ciclo de 2 segundos (1 s aceso, 1 s apagado). Nenhum outro LED interfere.
- Conclusão: Comportamento do modo noturno é correto e o mutex garante exclusão mútua entre os LEDs.

**Caso de teste 3:** Teste do modo de travessia

- Pré-condição: *modo\_atual = MODO\_NORMAL, estado\_atual = ESTADO\_VERDE, sistema\_rodando = true, travessia\_solicitada = false e tirar de comentado o “Alternância automática entre normal/travessia para teste” no fim do código*
- Etapas de teste: Executar o código “caso de teste 3”. Observar a piscagem dos LEDs.
- Pós-condição observada: O modo normal e o modo de travessia alternam de 15 em 15 segundos. Quando o modo travessia é ativado, o LED amarelo acende imediatamente por cerca de 1 segundo e, logo em seguida, o LED vermelho acende por 4 segundos, encerrando o ciclo e retornando para o modo normal.

- Conclusão: Comportamento do modo travessia é correto e o mutex garante exclusão mútua entre os LEDs.

**Caso de teste 4:** Teste do botão de travessia

- Pré-condição: `modo_atual = MODO_NORMAL, estado_atual = ESTADO_VERDE, sistema_rodando = true, travessia_solicitada = false`
- Etapas de teste: Executar o código “caso de teste 4”. Observar a piscagem dos LEDs e os LOGS.
- Pós-condição observada: O botão é ativado inúmeras vezes pois não há debounce, mas a execução depois dessas várias ativações, o código funciona perfeitamente. Não introduzir debounce não será problema, pois o microcontrolador dos carros irá receber um sinal único do microcontrolador dos pedestres.
- Conclusão: O botão de travessia funciona corretamente.

**Caso de teste 5:** Teste do botão de inicialização

- Pré-condição: `modo_atual = MODO_NORMAL, estado_atual = ESTADO_VERDE, sistema_rodando = false, travessia_solicitada = false`
- Etapas de teste: Executar o código “caso de teste 5”. Observar a piscagem dos LEDs e os LOGS.
- Pós-condição observada: O botão é ativado inúmeras vezes pois não há debounce, mas a execução depois dessas várias ativações, o código funciona perfeitamente. Não introduzir debounce não será problema, pois o microcontrolador dos carros irá receber um sinal único do microcontrolador dos pedestres.
- Conclusão: O botão de inicialização funciona corretamente.

As evidências da execução de cada teste estão registradas abaixo, por meio de printscreens dos logs impressos no serial monitor.

Caso de teste 1: Ciclos dos LEDs

```

[00:00:08.008,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:08.008,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:11.009,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:11.009,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:12.009,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:12.009,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:16.009,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:16.009,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:19.009,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:19.010,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:20.010,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:20.010,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:24.010,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:24.010,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos

```

### Caso de teste 2: Modo noturno

```

[00:06:22.076,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:23.076,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:24.076,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:25.076,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:26.076,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:27.076,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:28.077,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:29.077,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:30.077,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:31.077,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:32.077,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:33.077,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:34.078,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:35.078,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:36.078,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:37.078,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:38.078,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:39.079,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:40.079,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:41.079,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:42.079,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:43.079,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:44.080,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:45.080,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:46.080,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:47.080,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos
[00:06:48.080,000] <inf> SEMAFORO_VEICULOS: AMARELO: Piscando 1 segundos

```

### Caso de teste 3: Botão de travessia

```

[00:00:00.007,000] <inf> SEMAFORO_VEICULOS: Sistema inicializado via botão PTA17
[00:00:00.007,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:00.008,000] <inf> SEMAFORO_VEICULOS: Thread controle iniciada
[00:00:00.008,000] <inf> SEMAFORO_VEICULOS: Thread verde iniciada
[00:00:00.008,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:00.008,000] <inf> SEMAFORO_VEICULOS: Thread amarelo iniciada
[00:00:00.008,000] <inf> SEMAFORO_VEICULOS: Thread vermelho iniciada
[00:00:03.008,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:03.008,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:04.008,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:04.008,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:08.008,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:08.008,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:11.009,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:11.009,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:12.009,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:12.009,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:14.922,000] <inf> SEMAFORO_VEICULOS: >>> ALTERNANDO PARA MODO TRAVESSIA <<<
[00:00:15.023,000] <inf> SEMAFORO_VEICULOS: >>> INICIANDO MODO TRAVESSIA <<<
[00:00:15.023,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:16.023,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:20.031,000] <inf> SEMAFORO_VEICULOS: >>> FINALIZANDO TRAVESSIA - VOLTANDO AO NORMAL <<<
[00:00:20.031,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:20.119,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:23.119,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:23.119,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:24.119,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:24.119,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos

```

#### Caso de teste 4: Botão de travessia

```

[00:00:11.460,000] <inf> SEMAFORO_VEICULOS: >>> SINAL RECEBIDO: Travessia solicitada <<<
[00:00:11.460,000] <inf> SEMAFORO_VEICULOS: >>> SINAL RECEBIDO: Travessia solicitada <<<
[00:00:11.460,000] <inf> SEMAFORO_VEICULOS: >>> SINAL RECEBIDO: Travessia solicitada <<<
[00:00:11.519,000] <inf> SEMAFORO_VEICULOS: >>> INICIANDO MODO TRAVESSIA <<<
[00:00:11.519,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:12.519,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:16.527,000] <inf> SEMAFORO_VEICULOS: >>> FINALIZANDO TRAVESSIA - VOLTANDO AO NORMAL <<<
[00:00:16.527,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:16.615,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:19.615,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:19.615,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:20.615,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:20.615,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:24.616,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:24.616,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:27.616,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:27.616,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:28.616,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:28.616,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:32.617,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:32.617,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:35.617,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:35.617,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:36.617,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:36.617,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:40.617,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:40.618,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos

```

#### Caso de teste 5: Botão de inicialização

```
*** Booting Zephyr OS build zephyr-v40200 ***
[00:00:00.007,000] <inf> SEMAFORO_VEICULOS: === SEMÁFORO DE VEÍCULOS INICIALIZADO ===
[00:00:00.007,000] <inf> SEMAFORO_VEICULOS: Placa: FRDM-KL25Z
[00:00:00.007,000] <inf> SEMAFORO_VEICULOS: Threads: Controle, Verde, Amarelo (AZUL), Vermelho
[00:00:00.007,000] <inf> SEMAFORO_VEICULOS: Aguardando botão de inicialização (PTA17)...
[00:00:09.405,000] <inf> SEMAFORO_VEICULOS: >>> SINAL RECEBIDO: Sistema iniciado via botão PTA17 <<<
[00:00:09.405,000] <inf> SEMAFORO_VEICULOS: >>> Threads serão criadas e sequência iniciada <<<
[00:00:09.417,000] <inf> SEMAFORO_VEICULOS: Sistema inicializado via botão PTA17
[00:00:09.417,000] <inf> SEMAFORO_VEICULOS: Threads criadas: Controle, Verde, Amarelo, Vermelho
[00:00:09.417,000] <inf> SEMAFORO_VEICULOS: Modo atual: MODO NORMAL
[00:00:09.417,000] <inf> SEMAFORO_VEICULOS: Thread controle iniciada
[00:00:09.417,000] <inf> SEMAFORO_VEICULOS: Thread verde iniciada
[00:00:09.417,000] <inf> SEMAFORO_VEICULOS: Thread amarelo iniciada
[00:00:09.417,000] <inf> SEMAFORO_VEICULOS: Thread vermelho iniciada
[00:00:09.467,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:09.467,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:12.467,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:12.467,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:13.468,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:13.468,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:17.468,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:17.468,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:20.468,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
[00:00:20.468,000] <inf> SEMAFORO_VEICULOS: AMARELO: Aguardando 1 segundos
[00:00:21.468,000] <inf> SEMAFORO_VEICULOS: Estado: VERMELHO
[00:00:21.468,000] <inf> SEMAFORO_VEICULOS: VERMELHO: Aguardando 4 segundos
[00:00:25.469,000] <inf> SEMAFORO_VEICULOS: Estado: VERDE
[00:00:25.469,000] <inf> SEMAFORO_VEICULOS: VERDE: Aguardando 3 segundos
[00:00:28.469,000] <inf> SEMAFORO_VEICULOS: Estado: AMARELO
```