

EA075 – Laboratório de Sistemas Embarcados

Prof. Eric Rohmer

Data de entrega: 12/04/2022



Projeto 1 – Descrição da máquina de estados

Máquina de estados do semáforo durante o dia

Estado 0:

- Led vermelho dos carros desligado;
- Led amarelo dos carros desligado;
- Led verde dos carros ligado;
- Led vermelho para pedestres ligado;
- Led verde dos pedestres desligado;
- Display dos carros desligado;
- Display dos pedestres desligado;

Estado 1:

- Led vermelho dos carros apagado;
- Led amarelo dos carros ligado;
- Led verde dos carros apagado;
- Led vermelho dos pedestres ligado;
- Led verde dos pedestres apagado;
- Display dos carros desligado;
- Display dos pedestres desligado;

Estado 2:

- Led vermelho para carros ligado;
- Led amarelo dos carros apagado;
- Led verde dos carros apagado;
- Led vermelho dos pedestres apagado
- Led verde para pedestres ligado;
- Display dos carros ligado decrementando o tempo de 9 segundos;
- Display dos pedestres ligado decrementando o tempo de 5 segundos;

Estado 3:

- Led vermelho dos carros ligado;
- Led amarelo dos carros desligado;
- Led verde dos carros desligado;
- Led vermelho para pedestres ligado;
- Led verde dos pedestres desligado;
- Display dos carros ligado decrementando o tempo de 9 segundos;
- Display dos pedestres ligado mostrando o tempo de 0 segundos;

Estado 4:

- Led vermelho dos carros ligado;
- Led amarelo dos carros desligado;
- Led verde dos carros desligado;
- Led vermelho dos pedestres desligado;
- Led verde dos pedestres desligado;
- Display dos carros ligado decrementando o tempo de 9 segundos;
- Display dos pedestres desligado;

Obs: A combinação dos estados 3 e 4 produzem o efeito piscante do led vermelho e do display dos pedestres quando o tempo de travessia chega a zero.

Máquina de estados do semáforo durante a noite

Estado 5:

- Led vermelho dos carros desligado;
- Led amarelo dos carros ligado;
- Led verde dos carros desligado;
- Led vermelho dos pedestres ligado;
- Led verde dos pedestres desligado;
- Display dos carros desligado;
- Display dos pedestres desligado;

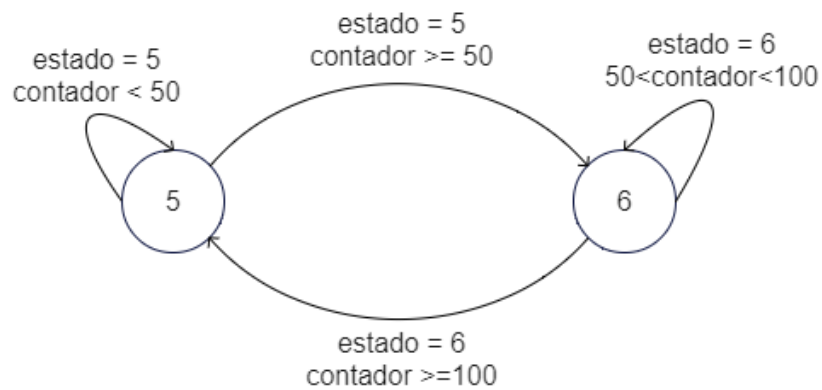
Estado 6:

- Led vermelho dos carros desligado;
- Led amarelo dos carros desligado;
- Led verde dos carros desligado;
- Led vermelho dos pedestres desligado;
- Led verde dos pedestres desligado;
- Display dos carros desligado;
- Display dos pedestres desligado;

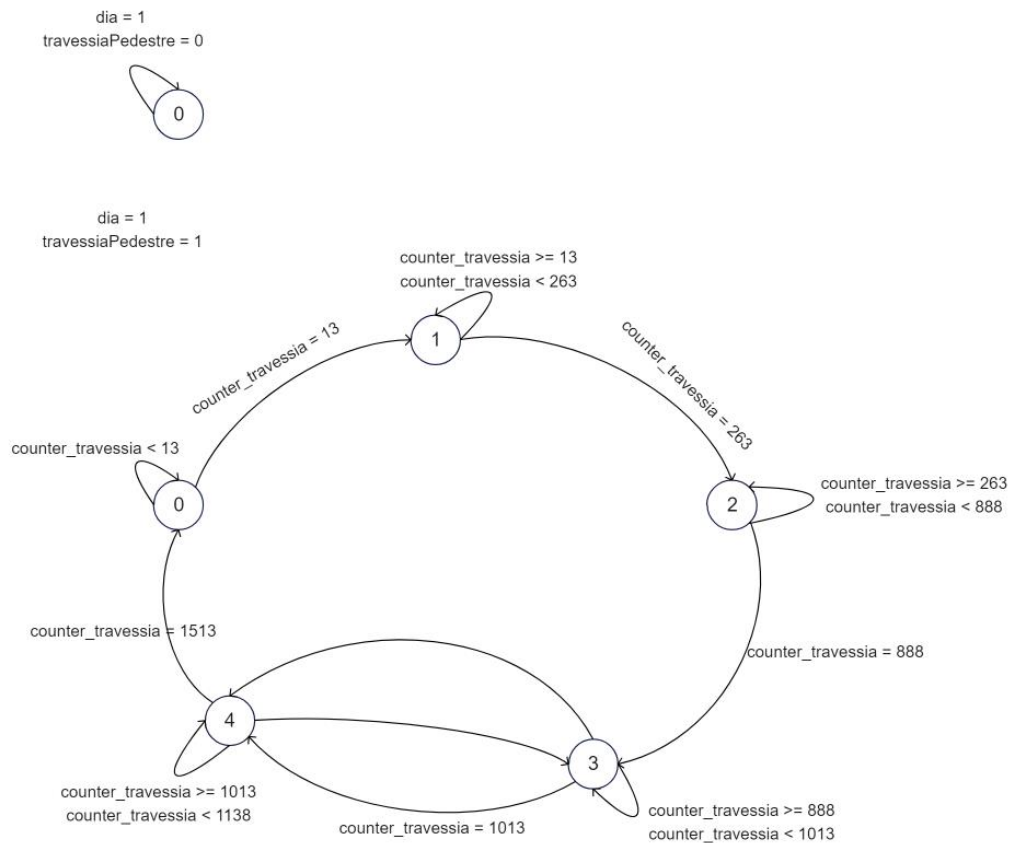
Obs: A combinação dos estados 5 e 6 produz o efeito piscante dos leds amarelo dos carros e vermelho dos pedestres.

Fluxograma máquina de estados

Modo noturno:



Modo diurno:



Obs: Quanto a contagem de tempo do display dos pedestres chegar a zero haverá alternância de estados entre 3 e 4 até que a contagem dos carros também chegue a zero (que ocorre quando counter_travessia atinge a contagem de 1513) e o ciclo de travessia de pedestres se encerra, voltando a máquina de estados a ficar no estado 0.

- Pseudo-código

Se estiver de noite (dia = 0):

Se contador <= 50:

Estado 5;

Se contador > 50;

Estado 6;

contador = 0;

Se estiver de dia e não houver pedestre (dia = 1 E travessiaPedestre = 0):

Estado 0;

Se estiver de dia e houver pedestre (dia = 1 E travessiaPedestre = 1):

//Esperar 100ms -> contador_travessia = 13

Se ((contador_travessia >= 13) && (contador_travessia < 263)):

//263 – 13 = 250 -> 2 segundos

Estado 1;

Se ((contador_travessia >= 263) && (contador_travessia < 388)):

//263 – 13 = 250 -> 2 segundos

//display dos carros em 9 segundos

//display dos pedestres em 5 segundos

Estado 2;

Se ((contador_travessia >= 388) && (contador_travessia < 513)):

//display dos carros em 8 segundos

//display dos pedestres em 4 segundos

Estado 2;

Se ((contador_travessia >= 513) && (contador_travessia < 638)):

//display dos carros em 7 segundos

//display dos pedestres em 3 segundos

Estado 2;

Se ((contador_travessia >= 638) && (contador_travessia < 763)):

//display dos carros em 6 segundos

//display dos pedestres em 2 segundos

Estado 2;

```

Se ((contador_travessia >= 763) && (contador_travessia < 888)):
    //display dos carros em 5 segundos
    //display dos pedestres em 1 segundos
    Estado 2;
Se ((contador_travessia >= 888) && (contador_travessia < 1013)):
    //display dos carros em 4 segundos
    //display dos pedestres em 0 segundos
    Estado 3;
Se ((contador_travessia >= 1013) && (contador_travessia < 1138)):
    //display dos carros em 3 segundos
    //display dos pedestres em 0 segundos
    Estado 4;
Se ((contador_travessia >= 1138) && (contador_travessia < 1263)):
    //display dos carros em 2 segundos
    //display dos pedestres em 0 segundos
    Estado 3;
Se ((contador_travessia >= 1263) && (contador_travessia < 1388)):
    //display dos carros em 1 segundos
    //display dos pedestres em 0 segundos
    Estado 4;
Se ((contador_travessia >= 1388) && (contador_travessia < 1513)):
    //display dos carros em 0 segundos
    //display dos pedestres em 0 segundos
    Estado 3;
Se (contador_travessia >= 1513):
    //Resetar as flags e os contadores
    //Retorno da máquina de estados ao estado 0
    travessiaPedestre = 0
    conter_travessia = 0
    delayDebouncing = 6

```

O projeto consiste em um semáforo inteligente com dois modos de operação: noturno e diurno.

No modo noturno não há leitura do estado do botão, e ocorre apenas o padrão piscante dos leds dos carros e do vermelho para os pedestres. A interação da placa com o meio externo ocorre apenas através do LDR, cujos valores são continuamente lidos.

No modo diurno é adicionado mais um meio de interagir com o ambiente, que é o botão de pedestres. A leitura do pino onde está conectado esse botão é feita continuamente (assim como o pino do LDR). Se o botão não for pressionado, mantém-se o estado padrão: Verde para os carros aceso, vermelho para os pedestres apagado. Caso o botão dos pedestres seja apertado, inicia-se o processo de travessia de pedestres descrito no projeto da máquina de estados.

No código existem 4 contadores responsáveis pelo controle de tempo das tarefas do programa, sendo que todos eles são incrementados ou decrementados na função que trata a interrupção no timer0 do microcontrolador. Os contadores utilizados são:

- counter: Responsável pelo padrão piscante dos leds amarelo dos carros e vermelho dos pedestres durante a noite (só é incrementado quando a flag dia=0, que equivale a ser noite).
- counter travessia: Controla o tempo de todo o processo de travessia dos pedestres (só é incrementado quando a flag dia=1 e a flag travessiaPedestre=1). Também é utilizada no controle do acionamento do displays da seguinte forma: Se counter_travessia%2 for 1, aciona-se o display dos carros; Se counter_travessia%2 for 0, aciona-se o display dos pedestres.
- delayDebouncing: Inicia o decremento logo que o botão de pedestres é pressionado (flag botaoPressionado = 1), e é responsável por aplicar uma espera de 48ms entre o pressionamento do botão e a execução do restante do código.
- timerDelay: É incrementado sempre que for detectado variações na leitura do LDR. Esse contador introduz um tempo de espera 5 segundos entre a mudança no valor do LDR e a mudança do valor da flag dia. Para efeitos práticos, esse contador é responsável por evitar que um carro que passe pelo semáforo a noite com os faróis acesos altere o padrão de funcionamento do semáforo de noturno para diurno.

No código existem também 3 flags que habilitam ou desabilitam determinados comportamentos especificados no projeto. São elas:

- dia: Flag que está atrelada a leitura do LDR e indica se está de dia (dia = 1) ou se é noite (dia = 0). Através dessa flag é condicionado o padrão noturno ou diurno do semáforo.
- botaoPressionado: Flag que está atrelada a leitura do botão, e indica se o mesmo foi pressionado (botaoPressionado = 1) ou se não foi pressionado (botaoPressionado = 0).

- travessiaPedestre: Flag que indica que o pedestre está atravessando e habilita o funcionamento do semáforo (transições de estado). A combinação entre as flags `dia=1` e `botaoPressioando=1` alteram o valor de `travessiaPedestre` para 1.

- Principais dificuldades

Como pode ser observado no código do projeto, os decrementos no tempo de travessia dos pedestres e dos carros foi feito explicitamente, isto é, foi declarado um estado diferente para cada instante de tempo. Para uma contagem pequena de apenas 10 segundos como a desse projeto isso talvez não seja um problema, mas em projetos maiores implementar essa mesma lógica seria uma tarefa difícil. Como melhoria pode-se propor um algoritmo mais iterativo, que decremente sozinho os tempos de cada etapa.