

Implementação de um semáforo com redes de Petri no SolidThinking Embed

1 Objetivo

Neste exemplo, será montado no Embed o modelo de um semáforo modelado através de redes de Petri. Uma vez simulado, iremos gerar o código com o Embed e implementar o semáforo usando um microprocessador Arduino Uno.

2 Implementação no Embed

Primeiramente, precisamos montar o diagrama em rede de Petri referente ao controle do semáforo. Um modelo simples pode ser visto na Figura 1.

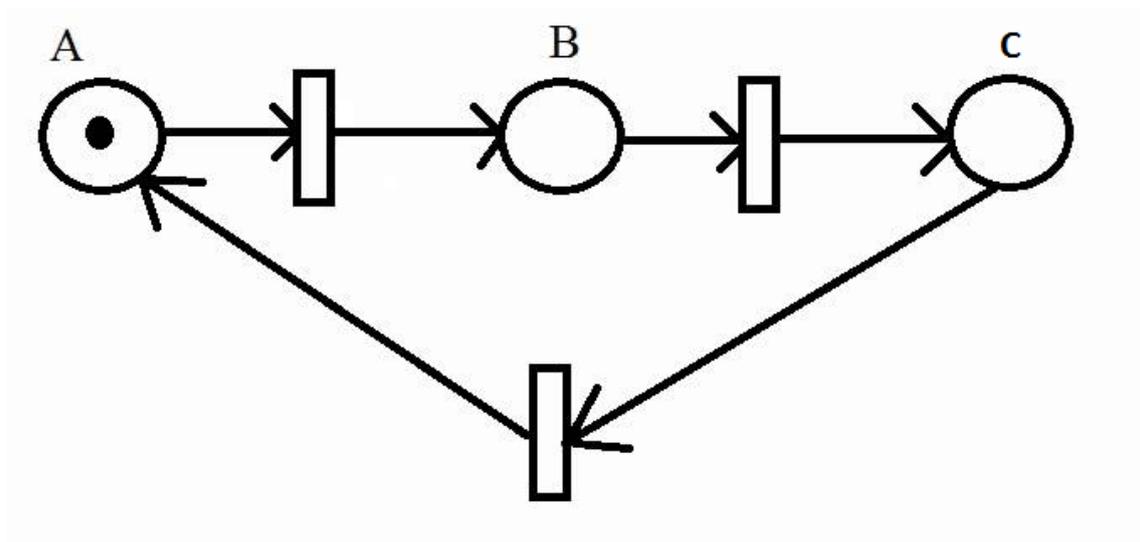


Figura 1 Rede de Petri de um semáforo simples

Cada estado (A, B e C) se refere a uma cor (Vermelho, amarelo e verde) e as transições são acionadas após uma contagem de tempo desejada.

Com a rede de Petri estabelecida, podemos implementar o controle no Embed usando State Charts. A implementação pode ser vista na Figura 2.

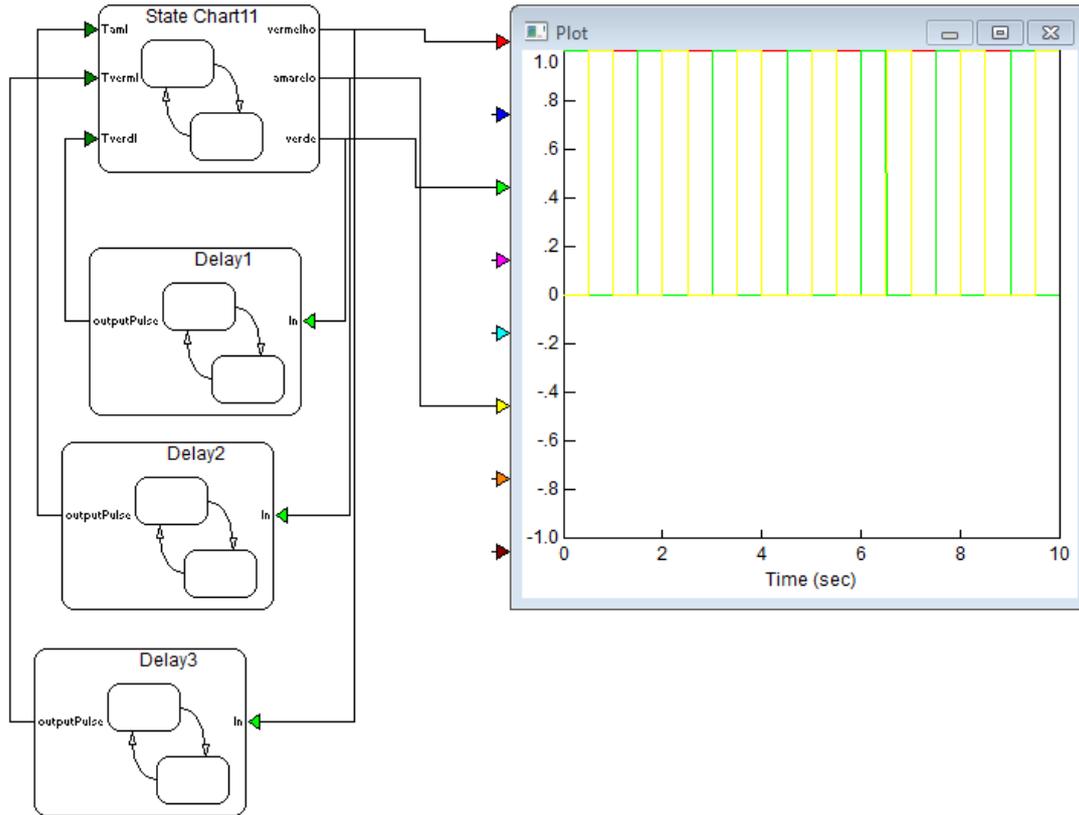


Figura 2 Rede de petri da figura 1 implementada no Embed

O primeiro State Chart se refere ao controle do semáforo, enquanto que os outros três serão responsáveis pela contagem de tempo de cada cor, de forma que os tempos escolhidos possam ter valores diferentes. Os três State Charts referentes a contagem de tempo possuem a mesma estrutura.

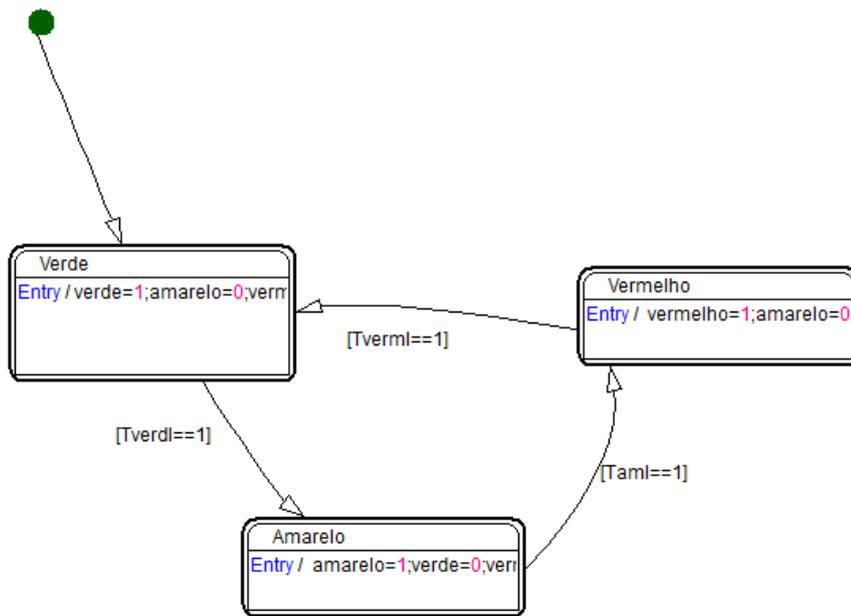


Figura 3 Composição do State Chart principal, referente ao controle do semáforo

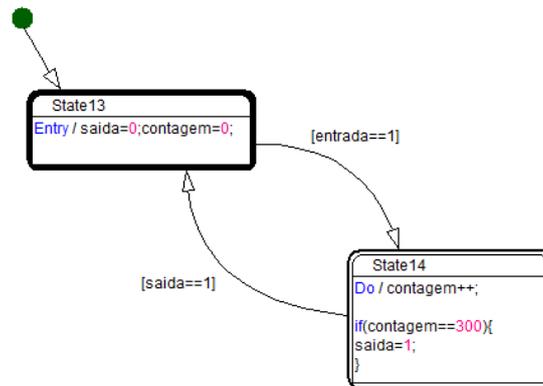


Figura 4 Composição do State Chart referente à contagem de tempo

Para o bloco principal, iniciamos com a variável “verde” igual a 1 e as variáveis “amarelo” e “vermelho” iguais a zero. Quando a variável externa Tverdl for igual a um, a primeira transição será acionada e mudar o estado, do bloco “verde” para o bloco “amarelo”, que irá zerar a variável “verde” e mudar a variável “amarelo” para um. O processo se repete passando para o estado “vermelho” e voltando para o verde, dando fim ao loop.

Para o bloco de contagem de tempo, visto na Figura 4, quando a entrada for igual a um, o estado de contagem será ativado e irá contar até 300. A função Do é executada a cada time step do programa, neste exemplo configurada para 0.01 segundos. Desta forma, passados 3 segundos, a variável 'contagem' será igual a 300 e a variável 'saida' será ajustada para um, fazendo com que o sistema volte para o estado inicial.

Testando com o auxílio de um Scope, podemos ver na Figura 5 que a implementação foi bem sucedida.

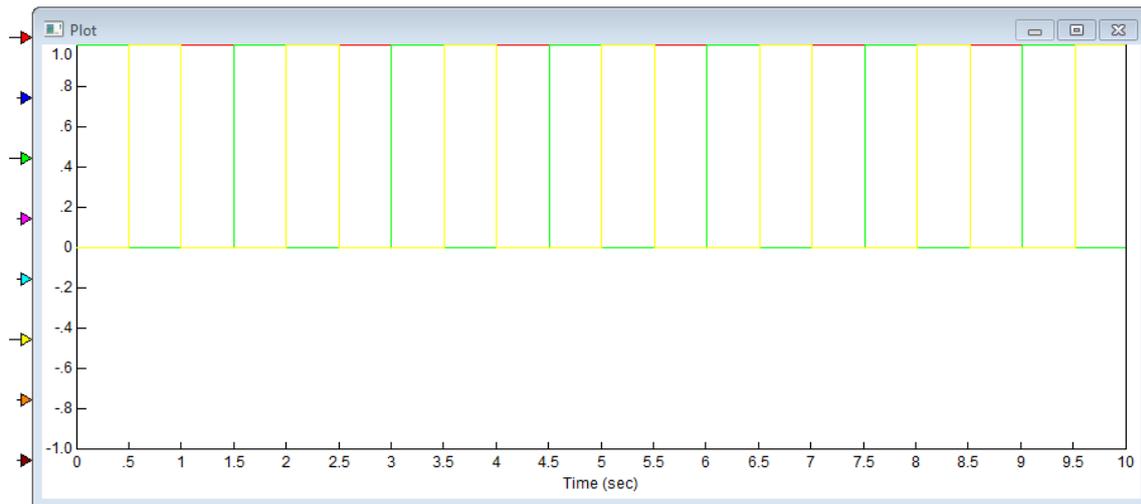


Figura 5 Os três sinais de saída ao final da implementação

3 Implementação no Arduino

Uma vez testado, podemos implementar o programa no Arduino. Para tal, basta adicionar as configurações do microprocessador e as portas de saída que iremos usar. Com isso, podemos gerar o código e fazer o download para o microprocessador, como pode ser visto na Figura 6.

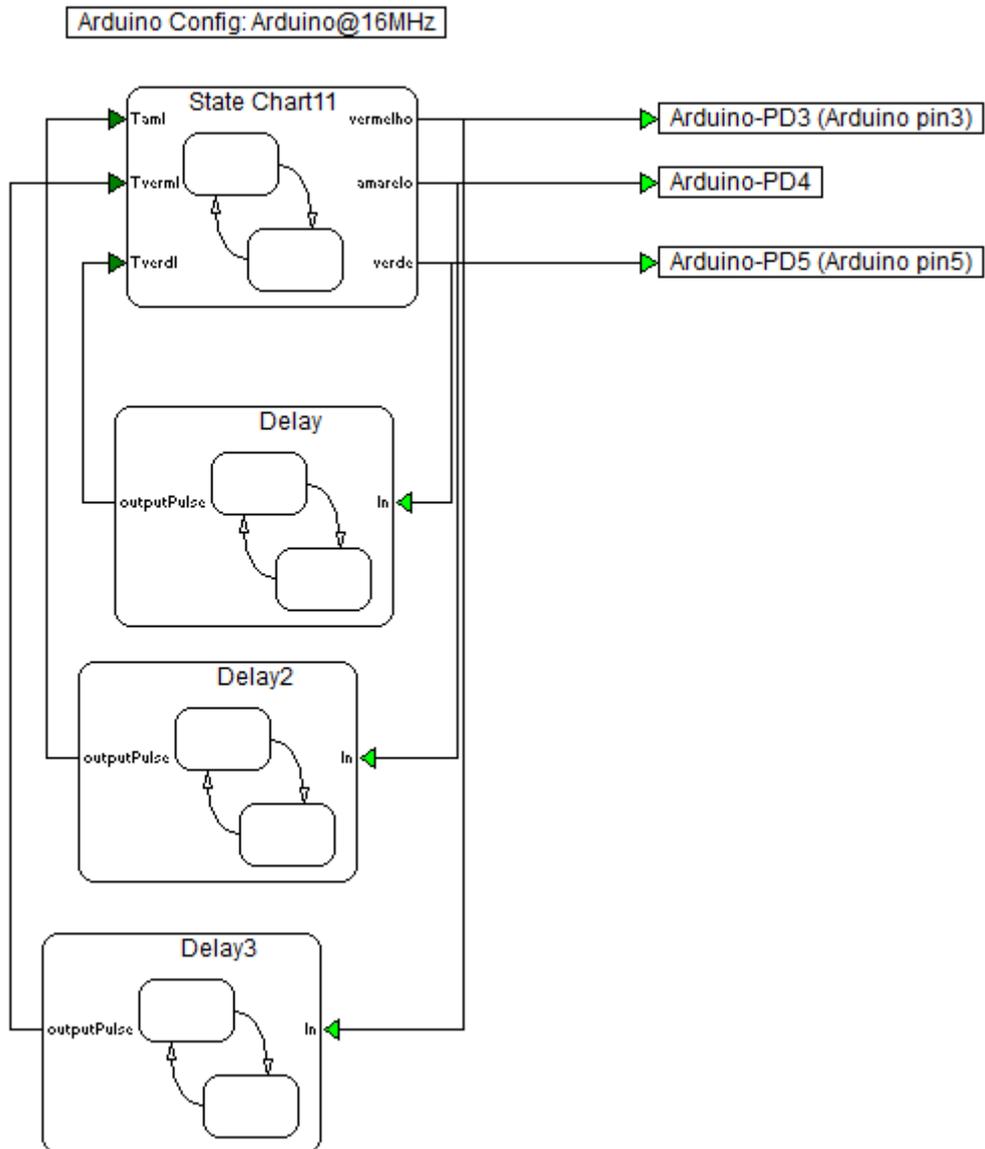


Figura 6 Geração do código para implementação no Arduino

Depois do download para a placa, basta realizar as conexões elétricas usando LEDs e resistores de 330Ω para limitar a corrente, como pode ser visto nas Figuras 7, 8 e 9.

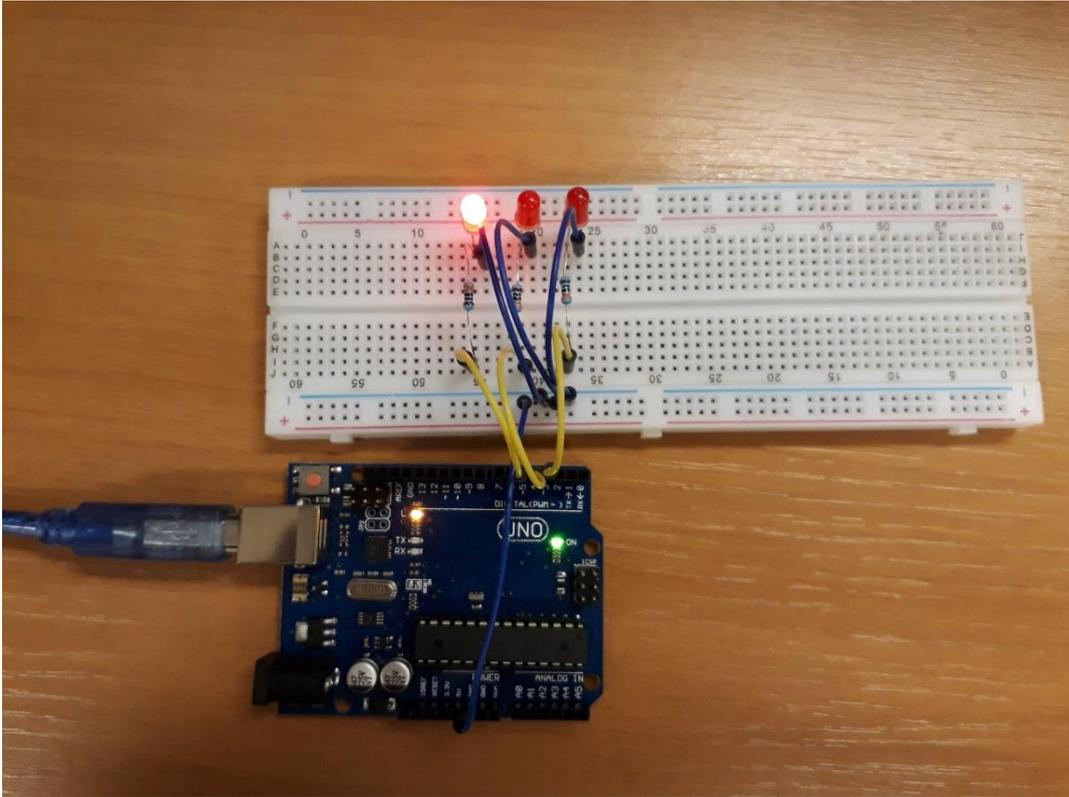


Figura 7 Primeiro LED, referente à cor verde, sendo acionado

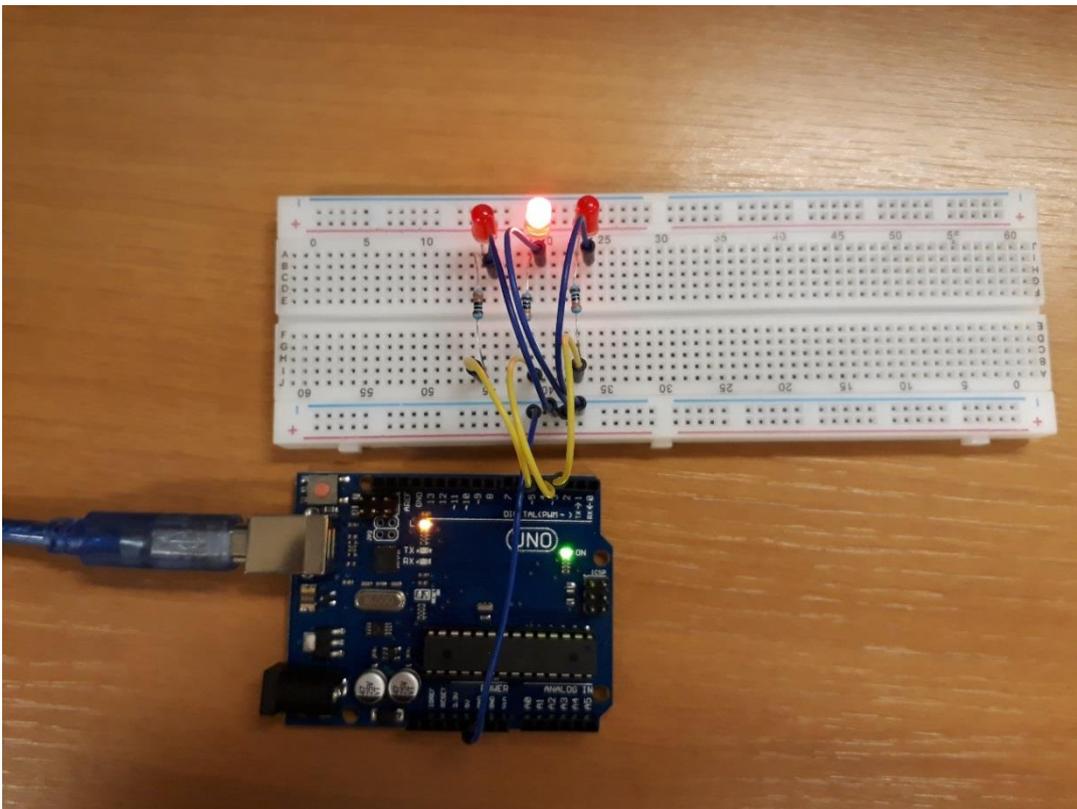


Figura 8 Segundo LED, referente à cor amarela, sendo acionado 1 segundo depois

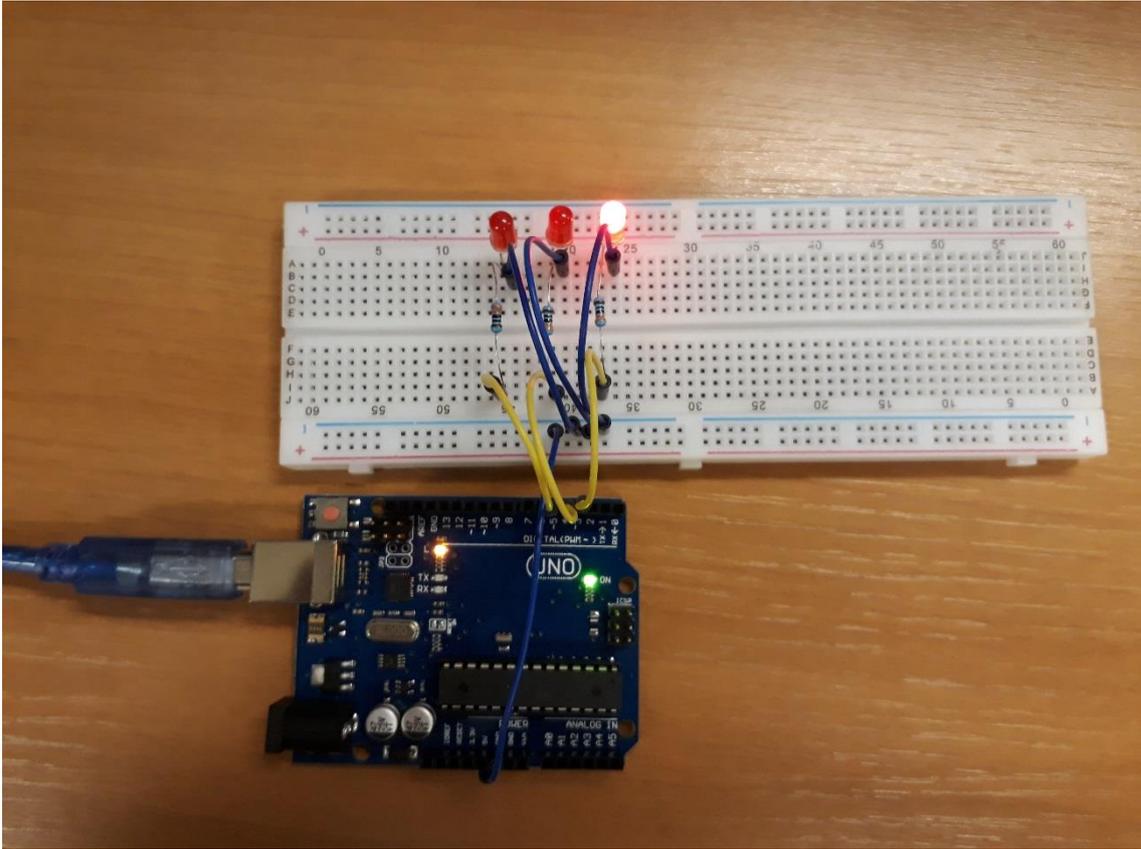


Figura 9 Terceiro LED, referente à cor vermelha, sendo acionado 1 segundo depois