

Controle de Tráfego Aplicado ao Sistema de Tempo Real

Processamento em Tempo Real - 26/06/2019

Gabriel Filipe Botelho Barbosa
12/0050935
gabrielffbbarbosa@gmail.com

Lucas Bonifácio Perez Nunes
14/0056581
lucasbpn@hotmail.com

Objetivo

Desenvolver uma aplicação que utilize um sistema de tempo real de forma a compreender os conceitos envolvidos como deadlines, períodos e processamentos de tarefas. Além disso, este projeto visa desenvolver um sistema a partir de uma visão prática do dia-a-dia, com o objetivo de melhorar o fluxo e o tráfego de automóveis nas cidades.

1. Introdução

Os sistemas computacionais estão presentes em diversas áreas do cotidiano e tem evoluído de forma intensa nos últimos anos. Essa evolução tem contribuído com o aumento da complexidade dos projetos de sistemas embarcados, que por consequência tem exigido novos níveis de abstração em soluções de software que possam interagir com o hardware da forma mais eficiente possível. Dentro de tal contexto, destaca-se um grupo de sistemas que trabalham com restrições de tempo, chamados Sistemas de Tempo Real. Sistemas de tempo real são submetidos a requisitos de natureza temporal nem sempre triviais e possuem requisitos definidos pelo ambiente. Seus aspectos temporais não estão limitados a uma questão de maior ou menor desempenho, mas sim, associados com a funcionalidade da aplicação. Deve-se necessariamente respeitar os requisitos temporais especificados de forma que os processamentos não extrapolem os prazos determinados (*deadlines*) causando falhas temporais. Atualmente no nosso cotidiano estão presentes diversos sistemas que utilizam requisitos de tempo real como base para suas aplicações. Pode-se listar como exemplo sistemas de aeronaves, automóveis e de máquinas de cartão de crédito. Por outro lado observa-se também a necessidade da implementação de um controle de “tempo real” para outros sistemas ao nosso redor, que acabam por atrasar o fluxo de pessoas, serviços e mercadorias diariamente. Uma pesquisa feita pelo SPC (Serviço de Proteção ao Crédito) para o Confederação Nacional de Dirigentes Lojistas (CNDL) detectou que o brasileiro passa, em

média, duas horas e vinte e oito minutos por dia no trânsito para se deslocar, de casa para o trabalho, para médicos, lazer e paradas as compras nas capitais. Ao multiplicar a média diária pelos dias do ano, o tempo de deslocamento equivale a trinta e sete dias e meio. Ou seja, mais de um mês inteiro passados no carro, ônibus, metrô, moto, bicicleta ou a pé.

2. Materiais

- Arduino Mega
- Motor de passo
- Sensor de presença
- LEDs
- Auto-falante
- Autorama
- Push-button



Figura 1. Imagem ilustrativa do ambiente da aplicação

3. Método

A aplicação consiste em simular por meio de um autograma, um ambiente com fluxo de automóveis, trens e pessoas como acontece nas grandes cidades. Porém, com o incremento da implementação de um sistema de tempo real para o controle dos semáforos. O sistema deve ser capaz de analisar a quantidade de carros presentes na pista de tal modo que possa controlar o período cíclico de cada semáforo. Como propriedades temporais da aplicação, tem-se que:

- Tarefas periódicas: Semáforos variando com suas cores verde, amarelo e vermelho
- Tarefas esporádicas: Passagem do trem pela pista
- Tarefas aperiódicas: Presença da ambulância com a movimentação da barreira e a presença do pedestre sendo “acionado” por um push button

Dessa forma, acrescenta-se a presença do pedestre que é caracterizada como uma interrupção de hardware (será acionada por um push-button simbolizando o botão presente nos semáforos indicando que alguém quer atravessar a faixa de pedestres). Assim, todo o sistema possui requisitos temporais como deadlines, períodos e intervalos que são requisitos a serem seguidos de forma a não causar falha temporal, no caso em questão, acidentes de trânsito. Divergente ao cenário de trânsito atual, os sensores utilizados no ambiente alimentam constantemente a malha, de modo que:

- A periodicidade do semáforo de pedestre varia de acordo com a presença de mais ou menos automóveis na pista sendo controlados pelo potenciômetro
- A passagem do trem resulta em “parada” de todos os semáforos e o correto posicionamento da barreira para que o mesmo possa passar sem que haja automóveis na pista
- A presença de um carro policial ou ambulância é vista como prioridade elevada durante o fluxo dos automóveis, mas ainda sim abaixo do trem
- O pedestre possui prioridade em relação ao fluxo de carros, mas com certa ressalva para que o trânsito não seja interrompido toda vez que houver um pedestre

Para implementar todo o processo, foi utilizado a biblioteca RTOS com tratador de interrupções e tasks para atender os requisitos temporais necessários.

3.1. Simulação com UPPAAL

Usamos a ferramenta UPPAAL, solicitada pelo professor, para fazer o projeto e simulação do sistema em tempo

real. Imagens dos módulos projetados que serão usados no projeto estão inclusos ao final deste documento.

Nós decidimos trabalhar com uma pista controlado por semáforo, onde também existe uma passagem de trem que ativa uma barreira e fecha os semáforos dos pedestres e do fluxo de carros. Também foi incluso dispositivos que permitem alterar momentaneamente o comportamento dos semáforos para priorizar o fluxo de veículos de serviço, caso estes estejam presentes.

4. Resultados

Devido aos recursos disponíveis, houve um redimensionamento do projeto. O escopo final foi um semáforo para veículos, um semáforo para pedestre e uma cancela para fechar a passagem para o trem. Os *inputs* são um *push button* para simular o botão de pedestre querendo atravessar, um *push button* para simular a aproximação e a presença de um trem, um potenciômetro para simular a detecção e cálculo da variação do fluxo de carros e um sensor de presença infravermelho (sensor PIR) para detecção de aproximação dos veículos de emergência.

4.1. Semáforos

Para os semáforos, foram usados 4 LEDs vermelhos (dois para cada), 1 LED amarelo (somente para o semáforo de veículos) e 2 LEDs verdes (um para cada). Todos de 5mm. Na figura 2 você pode ver o esquemático de ligação dos LEDs com o Arduino.

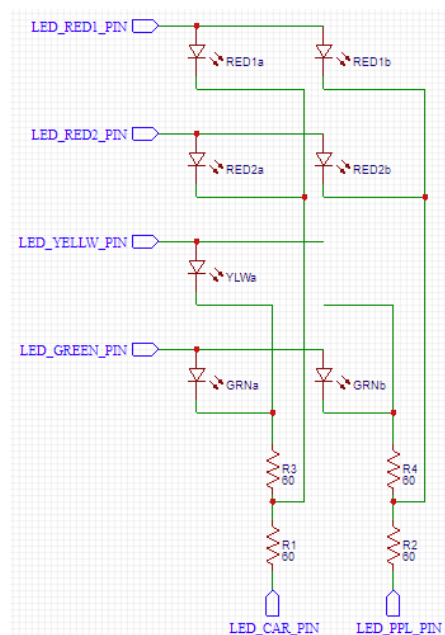


Figura 2. Esquemático da ligação dos LEDs

4.2. Atuadores e sensores

Para os atuadores e sensores, foi usado a construção com descrita na imagem 3. Onde o potenciômetro é usado com o objetivo de controlar uma simulação de não só um sensor de fluxo de veículos como o reporte do cálculo entre máximo e mínimo de fluxo a ser controlado (transformado em um número de ponto flutuante entre 0 e 1 por código). *Grove - PIR Motion Sensor* é o sensor de movimento que usa infravermelho utilizado no projeto na configuração *retriggerable*. Os pinos ligado aos push buttons estão em modo *PULLUP* e ligados no terra para não termos que lidar com debouncing e estados em alta impedância.

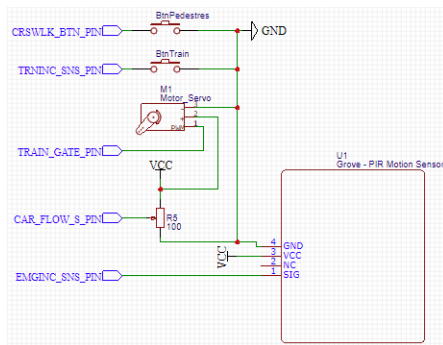


Figura 3. Esquemático da ligação dos LEDs

4.3. Limitação de tempo

Como o *Watchdog timer* do Arduino é, no mais rápido possível, 15ms, ou seja, 66.6Hz, isso não era rápido o suficiente para se aproveitar do efeito de persistência de visão para a multiplexação imperceptível dos LEDs. Por isso, a task responsável por isso roda sempre (nunca é bloqueada) e usando o *wait()* do próprio Arduino, resultando em *Busy Wait* mas um melhor controle da frequência de execução da tarefa. O ideal na verdade seria fazer com que todas as tarefas que precisem ser executadas com frequência maior que 66.6Hz sejam delegadas para um escalonador executivo cíclico com interrupções e o resto continuar com o escalonador do FreeRTOS.

5. Discussão e Conclusões

Dessa forma, compreende-se que a presença de requisitos temporais no sistema, como também prioridades entre os automóveis pode fazer com que o fluxo seja mais constante e mais adaptado com o momento presente. Alguns elementos são vistos como interrupções assim como na vida real e que foram tratados corretamente de forma que não aconteça nenhuma falha temporal na seção crítica.

Referências

- [1] <http://www.romulosilvadeoliveira.eng.br/livrotemporeal/cap-1-Caracterizacao-slides.pdf>.

- [2] <http://home.ufam.edu.br/lucascordeiro/str/slides/01-introducao-sistemas-de-tempo-real.pdf>.
[3] https://github.com/bestknighter/Ferrovia_PTR.

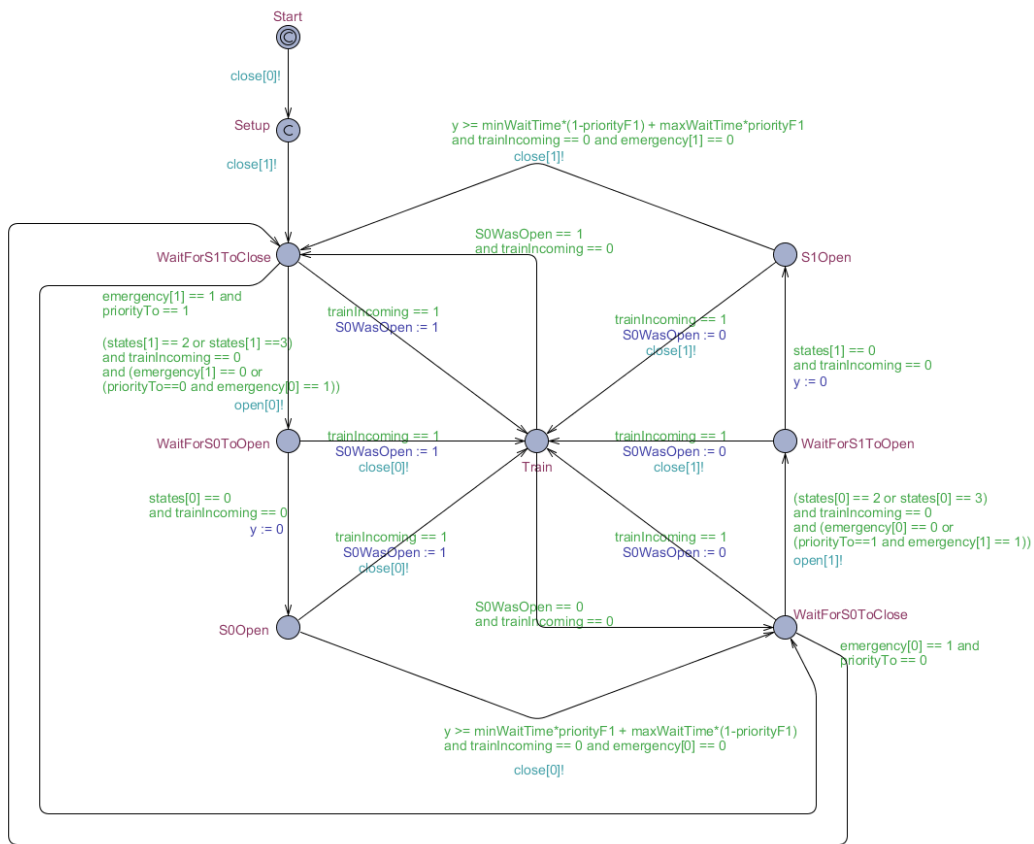


Figura 4. Controlador de todo o sistema semafórico da intercessão.

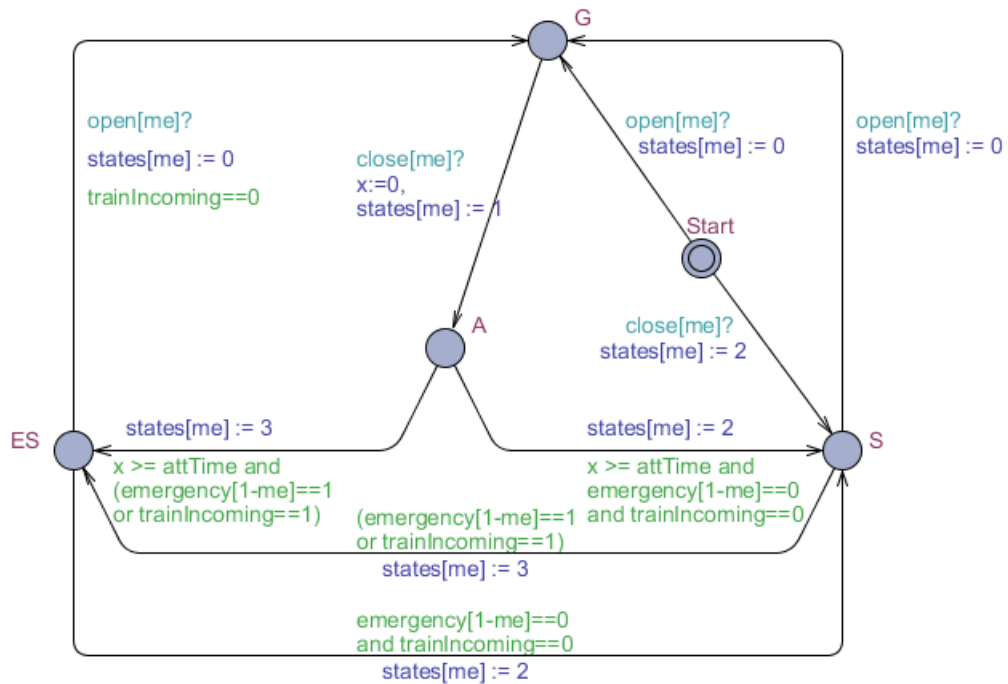


Figura 5. Controlador de uma única unidade semafórica.

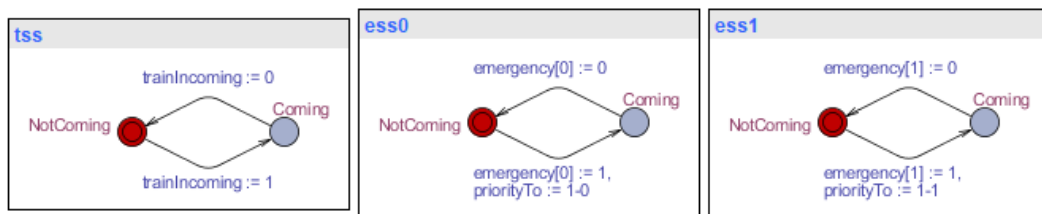


Figura 6. Visão de todos os módulos que emulam a entrada de um sensor, para uso da simulação.

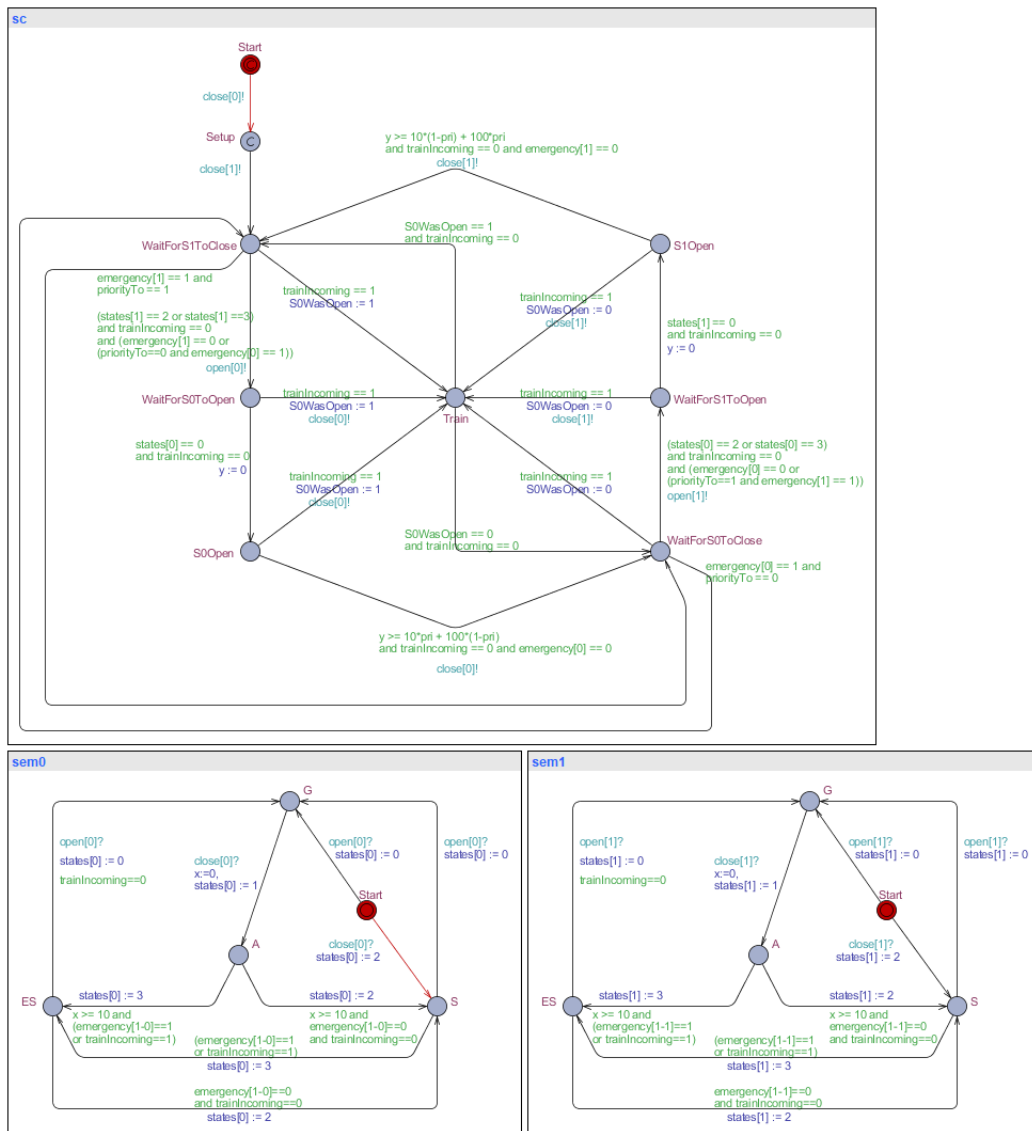


Figura 7. Visão de todos os controladores na simulação.

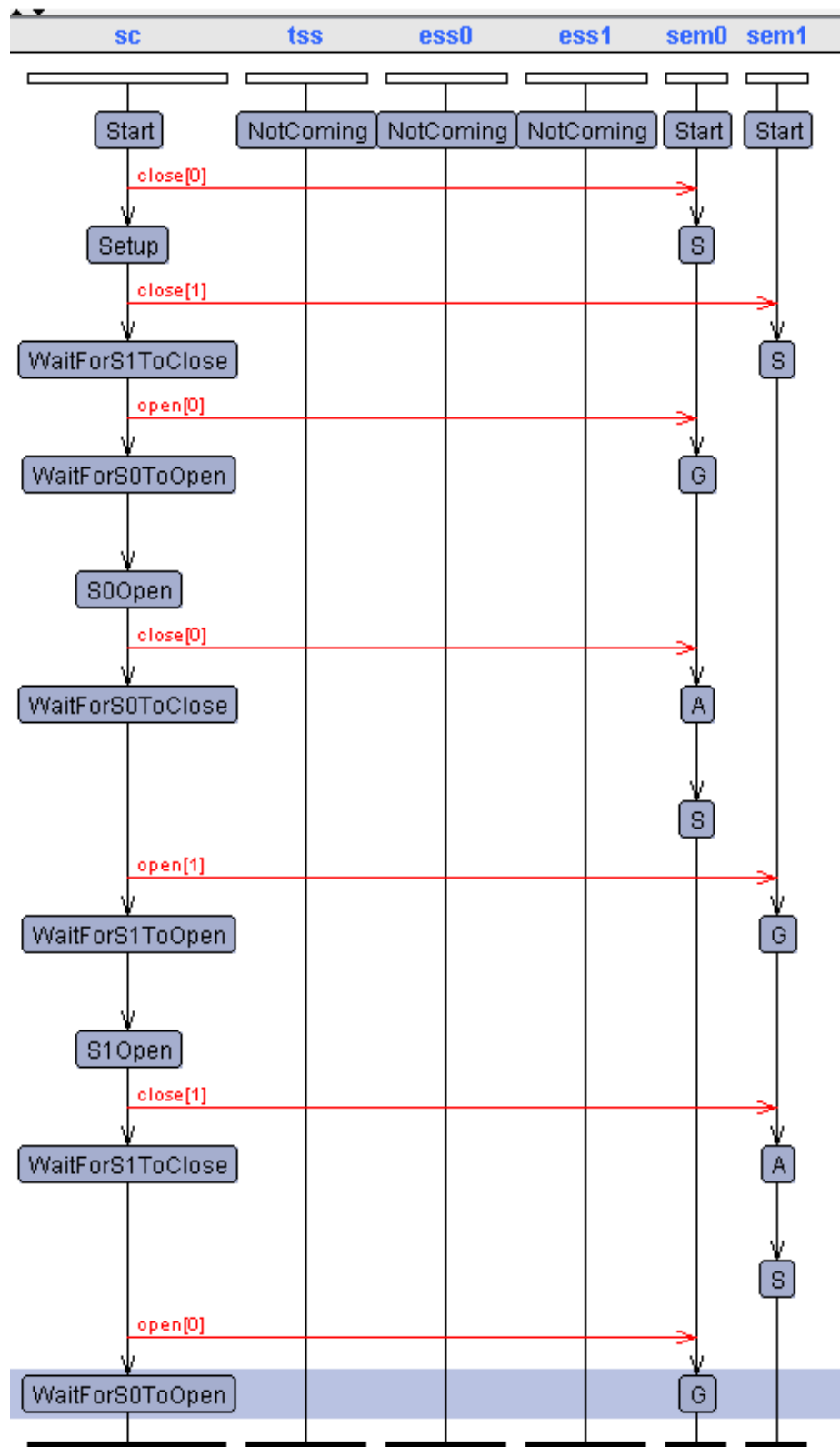


Figura 8. Fluxo de eventos do comportamento normal, sem eventos esporádicos ou aperiódicos.

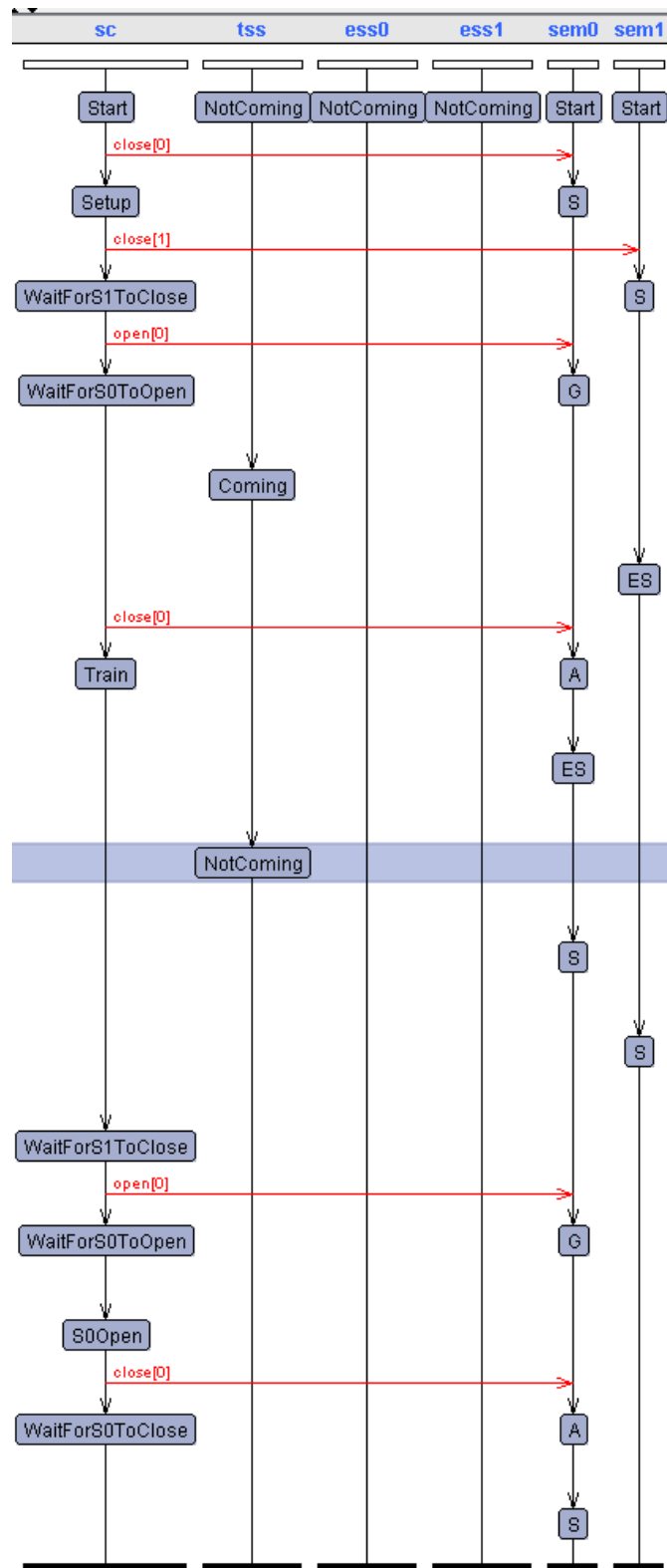


Figura 9. Fluxo de eventos do comportamento no caso de trem cruzando a pista. Em destaque o momento em que o cruzamento encerra.

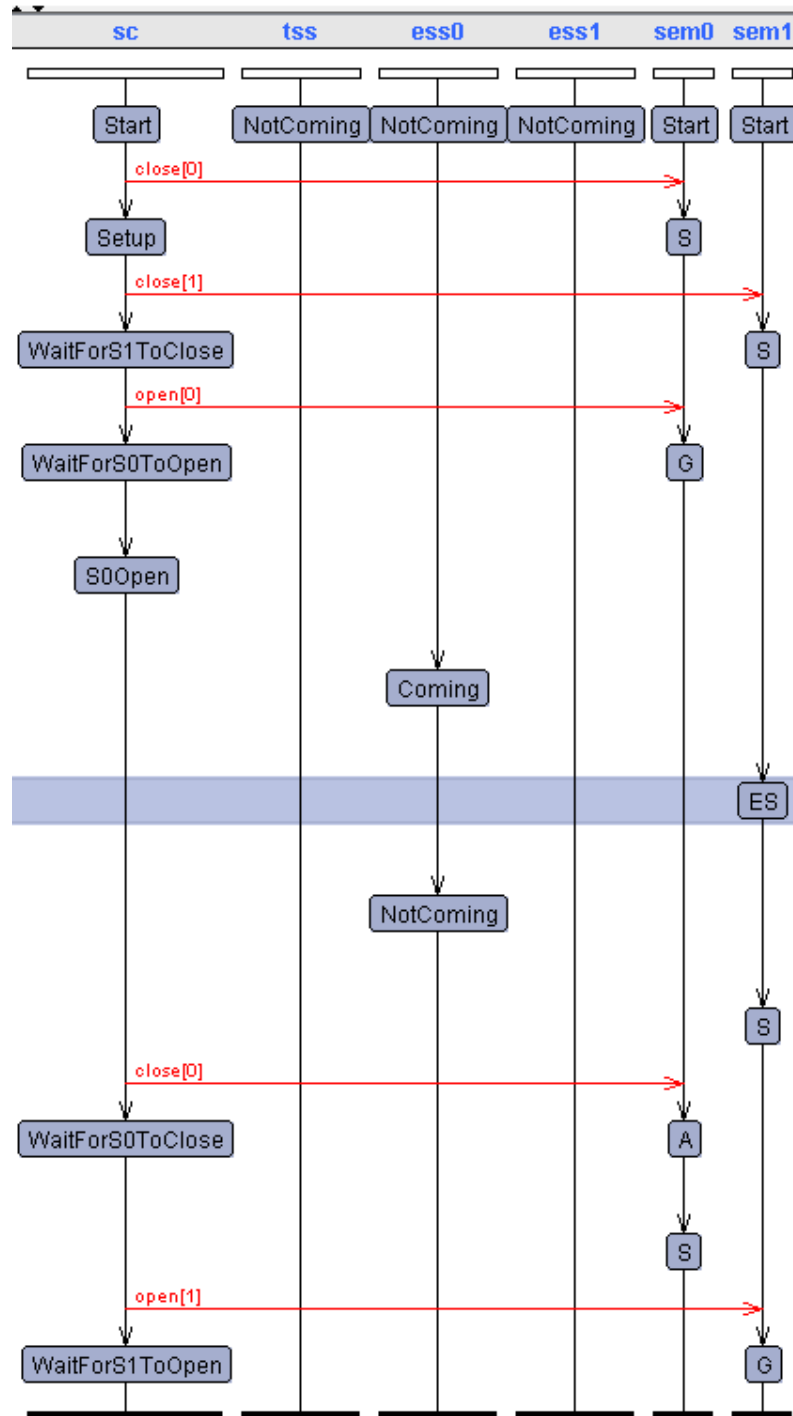


Figura 10. Fluxo de eventos do comportamento com fluxo prioritário para veículo de serviço. Em destaque o momento em que o semáforo da pista oposta muda para fechado emergencial.