

Projeto de Introdução à Arquitetura de Computadores

LEIC

IST-Taguspark

Sistema de controlo de comboios

2015/2016

1 – Objetivos

Este projeto pretende exercitar os fundamentos da área de Arquitetura de Computadores, nomeadamente a programação em linguagem *assembly*, os periféricos e as interrupções.

O objetivo deste projeto é implementar um sistema de gestão de comboios sobre um circuito predefinido, com agulhas e troços terminados por semáforos e sensores, com as seguintes funcionalidades:

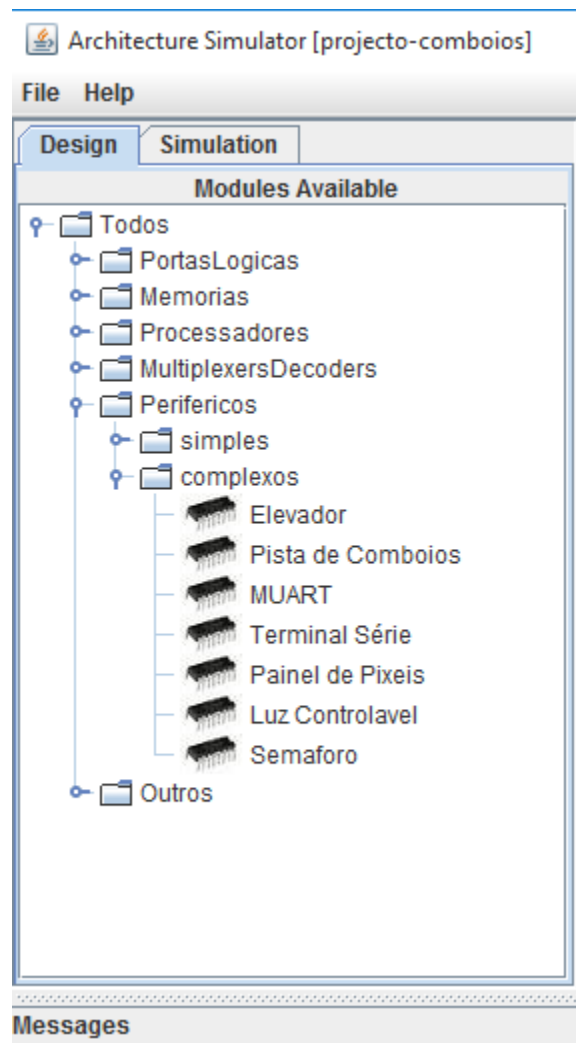
- Gestão automática dos semáforos de circulação (verde apenas se troço seguinte estiver livre);
- Gestão automática dos semáforos de uma passagem de nível (ligam em aproximação e desligam após passagem do comboio);
- Gestão manual das agulhas (o utilizador pode mudá-las em qualquer altura);
- Maquinista automático (com base nos sensores), seguindo o percurso determinado pelas agulhas, respeitando os semáforos (parando se vermelho) e parando automaticamente nas estações durante um tempo previamente definido.

O sistema deve ser implementado sobre o simulador da cadeira, com um programa de *assembly* do processador PEPE. O circuito de *hardware* (com o processador, memória e o módulo “Pista de Comboios”) é fornecido.

2 – Descrição do módulo “Pista de Comboios”

2.1 – Interfaces

Este módulo é um periférico complexo existente no simulador, disponível no seu menu de módulos:

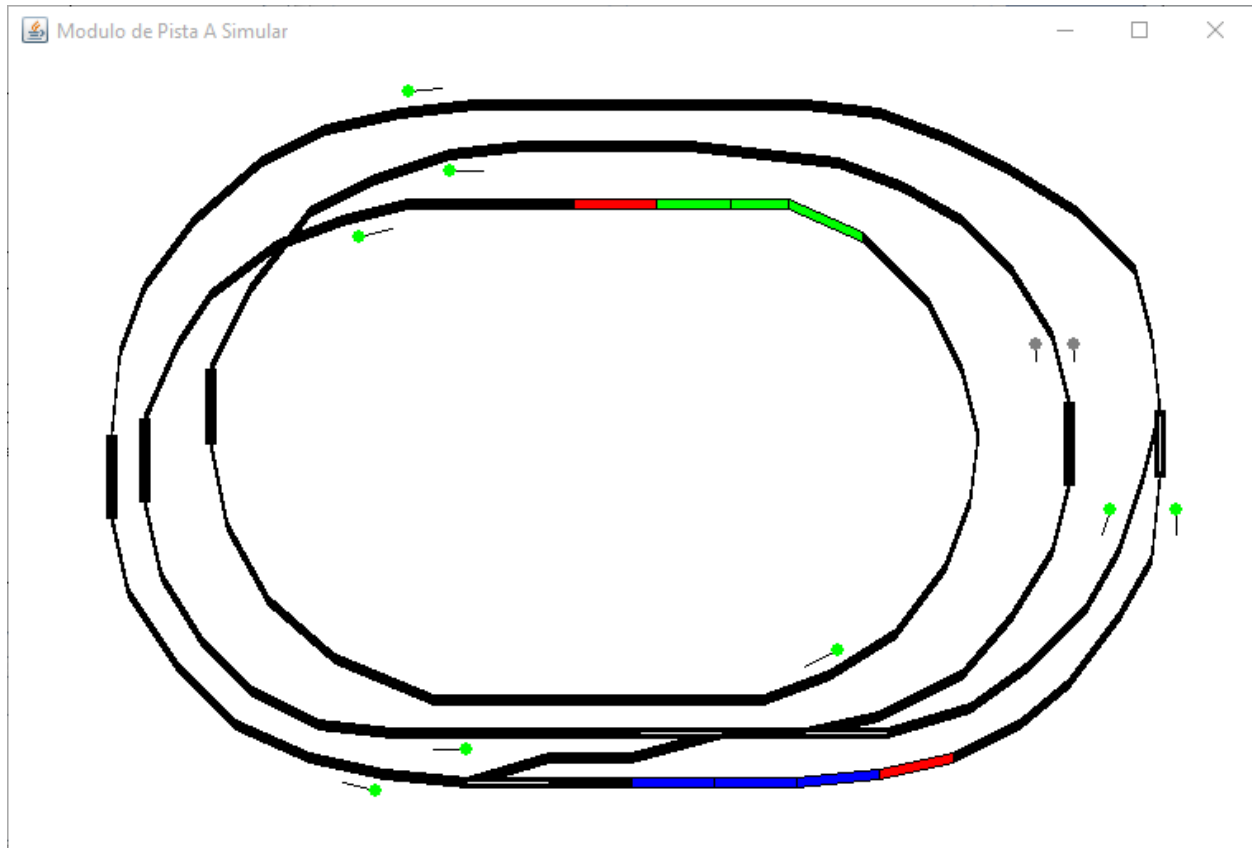


Este módulo aceita comandos a partir de um programa no PEPE. A sua funcionalidade está disponível no próprio simulador, fazendo duplo clique no módulo (no menu), em modo “Design”.

Em modo de “Simulation”, este módulo apresenta duas interfaces:

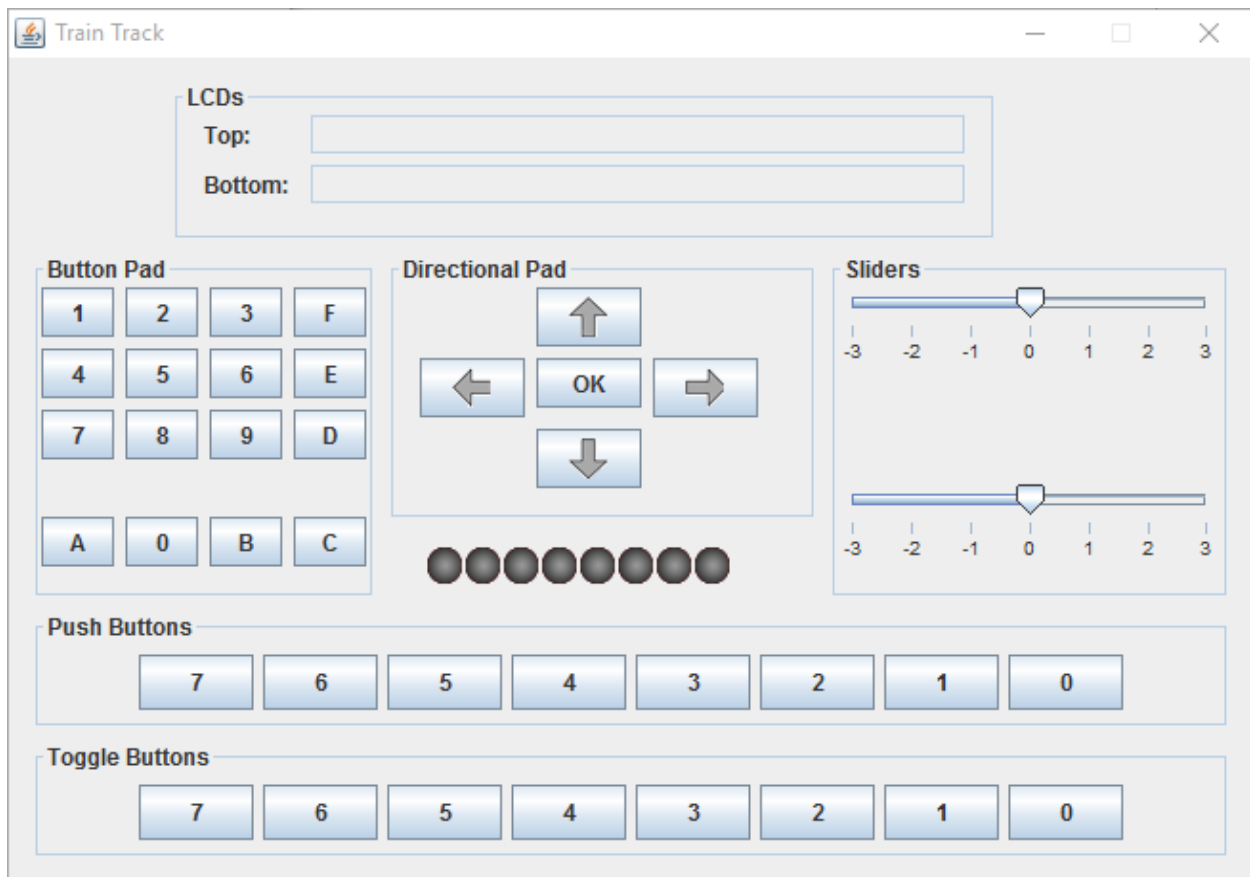
- Interface de visualização, que serve apenas para ver o circuito, a posição dos comboios e estado dos semáforos e agulhas. O estado do módulo, aqui visualizado, pode ser alterado a partir do programa no PEPE, com exceção da posição dos comboios. Uma vez que um comboio se ponha em andamento, evolui de forma autónoma e independente do programa (até este mudar a sua velocidade). Esta interface aparece automaticamente quando se carrega um programa no PEPE;
- Interface de controlo, constituída por um painel que oferece diversos elementos de controlo (botões e barras) e de visualização (leds e mostradores de texto), exclusivamente para utilização pelo programa. Se se pretender que um botão atue no estado do módulo, por exemplo, tem de ser o programa a ler o botão e a dar comandos ao módulo. Esta interface aparece fazendo clique duplo (em modo de “Simulation”) no módulo “Pista de comboios”, visível no circuito no simulador (onde aparece como “Train Track”).

A figura seguinte representa a interface de visualização com o estado inicial do circuito dos comboios, em que se pode observar:



- 2 comboios, um com carruagens a verde e outro a azul. As locomotivas estão representadas a vermelho. Os comboios circulam no sentido contrário ao dos ponteiros do relógio (locomotiva à frente do comboio);
- 4 agulhas. O traço branco no meio da linha preta indica para que lado cada agulha está virada (controlável manualmente);
- 8 semáforos de circulação, todos inicialmente a verde. Deverão estar a vermelho se um comboio se aproximar dele mas o troço seguinte ainda estiver ocupado pelo outro comboio;
- Cada semáforo termina um troço do circuito. Embora não visível, existe um sensor junto a cada semáforo, que permite saber ao comboio onde está;
- 2 semáforos de passagem de nível, que durante a passagem de um comboio devem alternar entre cinzento e vermelho (um cinzento, outro vermelho, depois trocam). Há sensores antes e depois da passagem de nível, para o sistema poder saber quando ligar (alternando) e desligar (ambos os semáforos a cinzento) estes semáforos.

A figura seguinte representa a interface de controlo deste módulo, em que se pode observar:



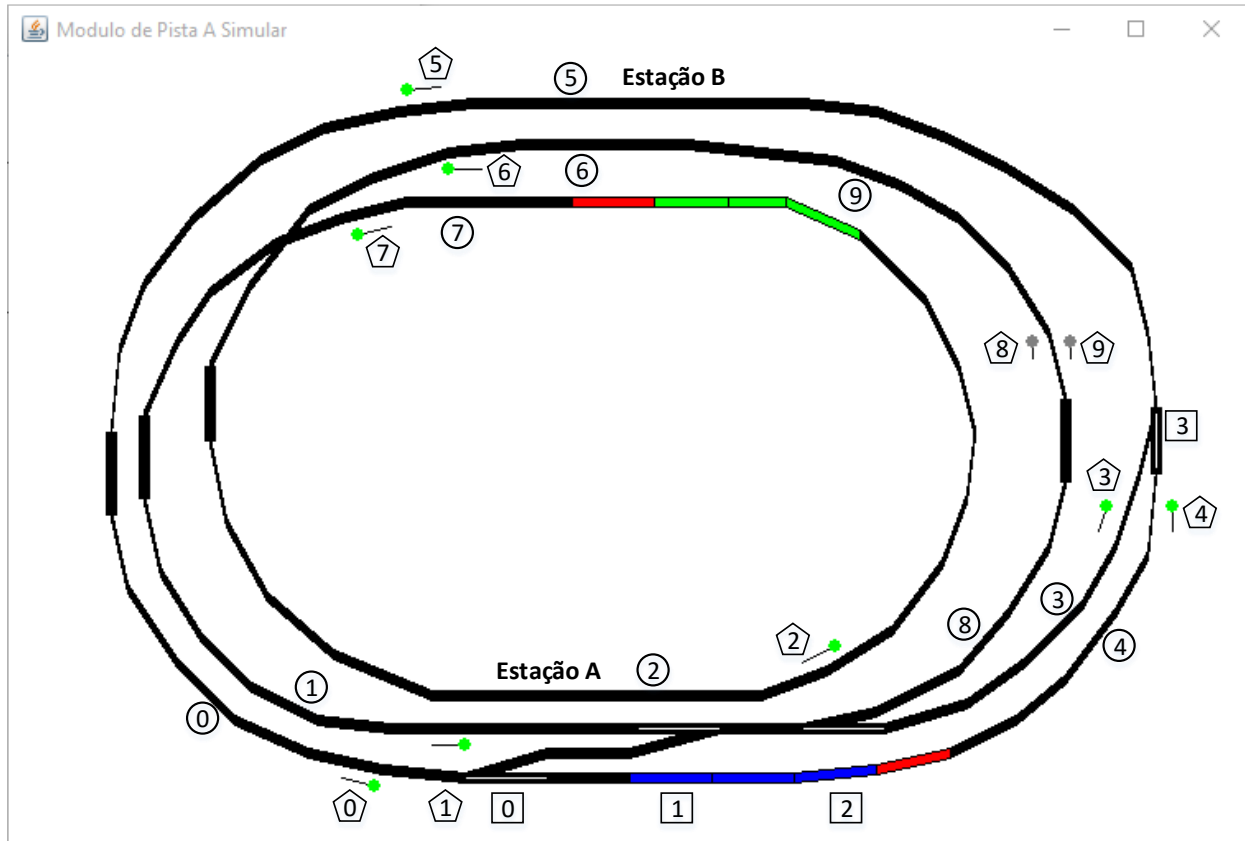
- Dois mostradores de texto (vulgarmente conhecidos por LCDs), com capacidade de 40 caracteres cada um;
- Um teclado de 16 teclas hexadecimais;
- Um teclado com setas e uma tecla OK;
- 8 leds;
- 2 barras de controlo (também conhecidas por *sliders*);
- 8 botões de pressão (em cada botão premido lê-se 1, caso contrário lê-se 0);
- 8 botões tipo alavanca (premindo e largando o botão, este fica premido, com estado 1. Para passar ao estado 0 tem de se premir e largar de novo).

É possível ao programa no PEPE ler o estado das teclas, botões e *sliders*, além, de permitir alterar o estado dos leds e escrever nos mostradores.

Toda a interação entre a interface de controlo e o estado do módulo tem de ser feita pelo programa.

2.2 – Funcionamento

A figura seguinte representa, de forma mais detalhada, o circuito a usar neste projeto. As definições dos comboios, do traçado e do posicionamento dos semáforos e sensores são especificadas em três ficheiros (**buildTrains.txt**, **buildTrack.txt** e **trafficLights.txt**), que devem estar no mesmo diretório que o simulador.



Nesta figura, pode observar-se a localização e número atribuído a cada um(a):

- Dos semáforos (pentágonos). Números 0 a 7 para os semáforos de circulação, 8 e 9 para os semáforos de passagem de nível;
- Dos sensores (círculos). Os sensores localizados junto aos semáforos de circulação têm o mesmo número;
- As agulhas (quadrados). Não há qualquer problema se um comboio circular por uma agulha “convergente”, vindo do lado não selecionado. Se os comboios circularem apenas no sentido contrário ao dos ponteiros do relógio, apenas as agulhas 0 e 2 terão efeito. Se se mudar uma agulha durante a passagem de um comboio, este descarrila (e dá erro na janela de mensagens do simulador);
- Estações A e B, que constituem apenas efeito visual e correspondem apenas ao troços que terminam nos semáforos 2 e 5, respetivamente.

Considera-se que o circuito está dividido em 8 troços (0 a 7), cada um terminando no semáforo com o mesmo número, precedido pelo sensor com o mesmo número. Este sensor permite saber quando um comboio passa por ele, caso em que o programa pode decidir parar o comboio (por exemplo, se o semáforo estiver vermelho).

A passagem da locomotiva por um sensor origina dois eventos, correspondentes aos dois topos da locomotiva, frente e trás (a passagem das carruagens não tem efeito). Use apenas o evento do topo da frente. O segundo evento deve também ser lido (para o retirar da fila dos eventos), mas ignorado.

Para saber quando ocorre um evento, o programa ou tem de estar continuamente a perguntar ao módulo dos comboios se houve um evento ou pode esperar que este gere uma interrupção

O programa deve ainda controlar o estado dos semáforos e das agulhas, para além da velocidade dos comboios.

Consulte os comandos que se podem dar ao módulo dos comboios, numa lista disponível no próprio simulador, fazendo duplo clique no módulo (no menu), em modo “Design”.

3 – Especificação do projeto

O projeto consiste em construir e testar um programa que controle o módulo de comboios, cumprindo as especificações desta secção. Deverá ser implementado em duas partes:

- Controlo manual, em que se usam os elementos da interface de controlo para controlar manualmente o estado do sistema. Esta parte, mais simples, destina-se a ganhar prática de programação do PEPE e de como controlar o módulo de comboios. É uma primeira versão do projeto;
- Controlo automático, em que será apenas possível controlar manualmente as agulhas e tudo o resto (semáforos, velocidade dos comboios, paragem nas estações e passagem de nível) deve ser gerido de forma automática pelo programa.

3.1 – Controlo manual

Pretende-se que, na interface de controlo:

- Variando os *sliders*, a velocidade dos comboios varie. O *slider* de cima controla o comboio 0 (carruagens azuis), o de baixo o comboio 1 (carruagens verdes). Os comboios poderão andar para trás (no sentido dos ponteiros do relógio);
- Carregando nas teclas do teclado, a luz dos semáforos alterne (entre verde e vermelho nos semáforos 0 a 7, entre cinzento e vermelho nos semáforos 8 e 9);
- Carregando nos botões de pressão 0 a 3, as agulhas alternem a direção selecionada;
- O mostrador de cima vá mostrando os sensores por que o comboio 0 vai passando no seu percurso (idem para o comboio 1 no mostrador de baixo)

Cuidado com os choques entre comboios!

3.2 – Controlo automático

Na interface de controlo, apenas se pode mudar o estado das agulhas, tal como no controlo manual.

Para que os comboios nunca choquem, só devem avançar para o troço seguinte do circuito, após cada semáforo, se esse troço estiver livre, caso em que o podem ocupar (reservar). Quando chegam ao fim de um troço, devem libertar o troço anterior (permitindo ao outro comboio ocupá-lo). Antes de entrar no troço seguinte, devem ocupá-lo.

Cada semáforo de circulação deve estar verde apenas se o troço seguinte estiver livre, ou se o troço que esse semáforo termina e o seguinte estiverem ocupados pelo mesmo comboio. O estado visual dos semáforos deve sempre refletir o estado de ocupação dos troços, o dele e o seguinte.

Se um comboio chegar a um semáforo vermelho, deve parar até que este esteja verde.

O número do troço seguinte a um dado troço pode variar, de acordo com o estado das agulhas.

Quando um comboio chega a uma estação, deve parar durante um tempo fixo (na ordem dos 3 ou 4 segundos) e depois prosseguir viagem (se e quando o semáforo após a estação estiver verde).

Quando um comboio passar pelo sensor 8 deve ligar a passagem de nível, desligando-os ao passar pelo sensor 9. Durante este tempo, os semáforos 8 e 9 devem alternar entre cinzento e vermelho, a cada 0,5 segundos (quando um está cinzento o outro está vermelho, e depois trocam). Fora deste tempo, os semáforos devem estar ambos cinzentos.

3.3 – Adicional ao controlo automático

Quando tiver o seu programa de controlo automático a funcionar, experimente mudar a agulha 2 quando a locomotiva de um comboio estiver entre o sensor 1 e a agulha 2.

Explique o porquê desta secção (no relatório do projeto) e indique se tal implica alterações (e quais) ao programa.

4 – Faseamento do projeto

O projeto decorrerá em duas fases, versão intermédia (controlo manual) e final (controlo automático). A versão intermédia vale 20% da nota do projeto (ou 4 valores em 20).

A versão intermédia do projeto deverá ser entregue até ao dia 11 de novembro de 2015, 23h59. A entrega, a submeter no Fenix (Projeto IAC 2015-16 - versão intermédia) deve consistir de um ficheiro (grupoXX.asm, em que XX é o número do grupo)

Sugere-se criar uma cópia da versão atual do código (eventualmente até já com coisas do controlo automático), de modo a compilar e executar apenas funcionalidade do controlo manual, limpando eventual “lixo” e coisas temporárias.

IMPORTANTE – Não se esqueça de identificar o código com o número do grupo e número e nome dos alunos que participaram na construção do programa (em comentários, logo no início da listagem).

A versão final do projeto deverá ser entregue até ao dia 2 de dezembro de 2015, 23h59. A entrega, a submeter no Fenix (Projeto IAC 2015-16 - versão final) deve consistir de um zip (grupoXX.zip, em que XX é o número do grupo) com dois ficheiros:

- Um pequeno relatório (modelo disponível em MS-WORD no Fenix), em pdf;
- O código, pronto para ser carregado no simulador e executado.

5 – Critérios de avaliação

Os critérios de avaliação e o seu peso relativo na nota final do projeto (expressos numa cotação em valores) estão indicados na tabela seguinte:

Critério	Versão intermédia	Versão final
Cumprimento da funcionalidade	2	4
Estrutura dos dados (tabelas, variáveis)		3
Estrutura do código (inclui processos)	1	4
Uso correto das interrupções		2
Qualidade dos comentários	1	2
Adicional ao controlo automático		1
Total	4	16

6 – Implementação

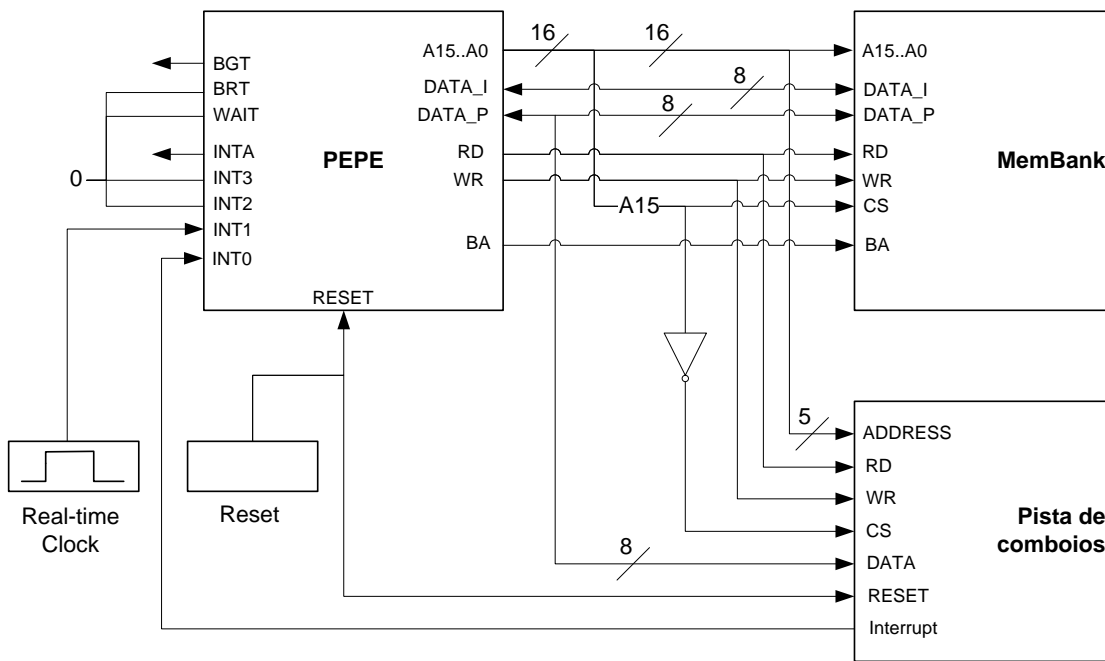
6.1 – Circuito a usar

A figura seguinte representa o circuito a usar no simulador e é fornecido, no ficheiro **projecto-comboios.cmod**), que inclui um relógio de tempo real, pré-programado para um período de 0,5 seg.

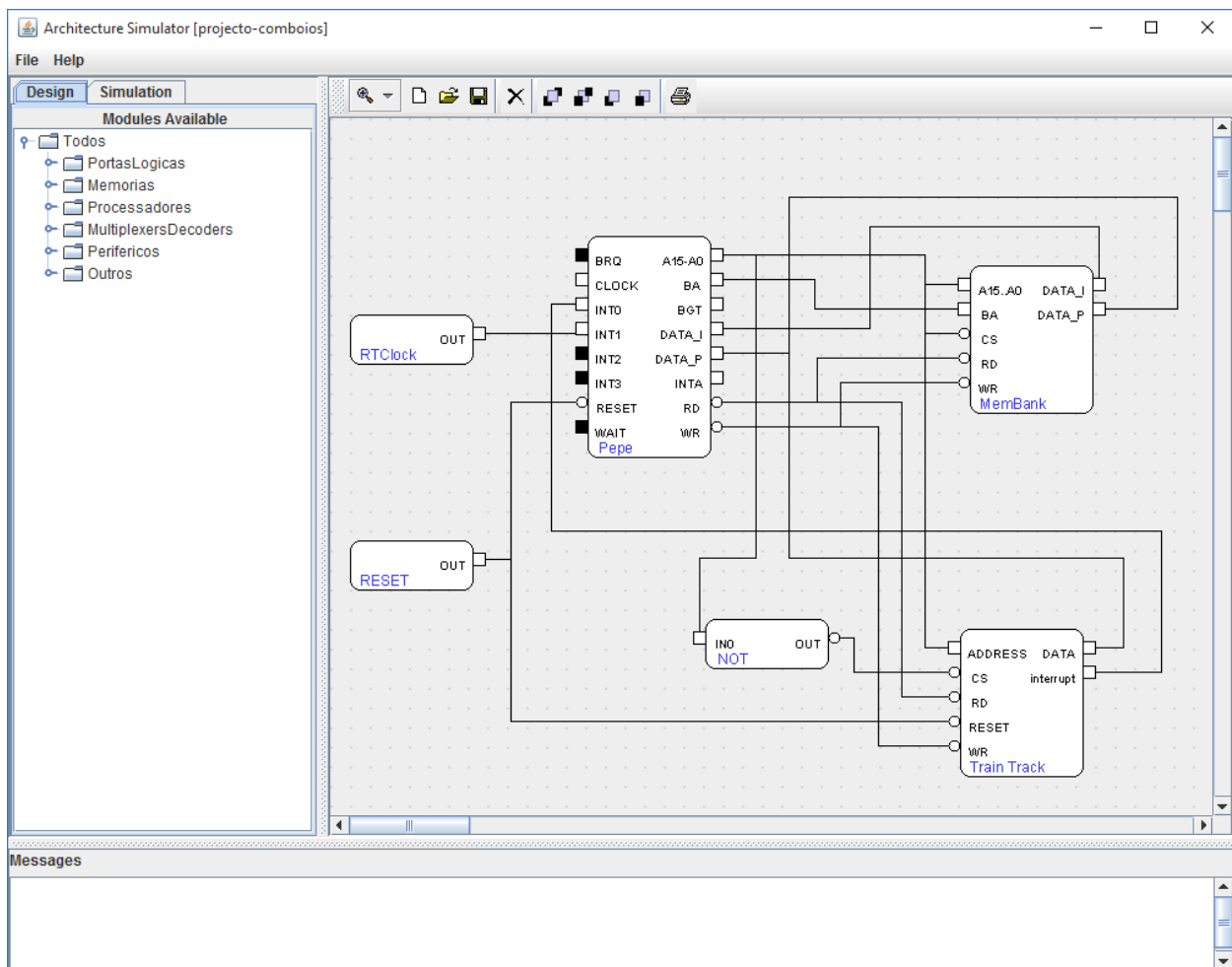
Este circuito é muito simples, pois toda a complexidade do sistema dos comboios é implementada pelo módulo dos comboios.

A interrupção 0 é gerada pelo módulo dos comboios sempre que ocorram eventos. A interrupção 1 é gerada periodicamente, a cada 0,5 segundos, desde que se tenha feito “Start” no relógio de tempo real.

NOTA IMPORTANTE – A interrupção 0 é ativada **enquanto** houver eventos por ler (o módulo dos comboios tem uma fila interna de 8 eventos, para não se perderem eventos caso surjam em sequência rápida e o programa não tenha tempo para os tratar logo que ocorram). Por isso, **a interrupção 0 deve ser programada para ser sensível ao nível 1**, enquanto a interrupção 1 (relógio) deve ser programada para ser sensível ao flanco (funcionamento por omissão).



No simulador, este circuito tem o seguinte aspeto:



O mapa de endereços (em que os dispositivos podem ser acedidos pelo PEPE) é o seguinte:

Dispositivo	Endereços
RAM (MemBank)	0000H a 7FFFH
Módulo dos comboios (TrainTrack)	8000H a 801EH

IMPORTANTE – Os ficheiros que definem a pista de comboios em concreto (**buildTrains.txt**, **buildTrack.txt** e **trafficLights.txt**), que também são fornecidos, devem estar no mesmo diretório que o simulador. Por essa razão, o simulador também é incluído no conjunto de ficheiros deste enunciado.

OUTRAS NOTAS IMPORTANTES:

- Os periféricos de 8 bits (neste caso, apenas o módulo dos comboios) e as tabelas com STRING devem ser acedidos com a instrução MOVB. As variáveis definidas com WORD (que são de 16 bits) devem ser acedidas com MOV;
- O módulo dos comboios inclui 16 portas internas (de 0 a F). Dado que este módulo é de 8 bits e só liga à metade par do bus de dados, os ports estão acessíveis apenas em endereços pares (de 2 em 2, de 8000H a 801EH). Por exemplo, os dois LCDs estão acessíveis em 8000H e 8002H;
- Tal como descrito atrás, a interrupção 0 deve ser programada para ser sensível ao nível 1;
- As rotinas de interrupção param o programa principal enquanto estiverem a executar. Por isso, devem apenas atualizar uma variáveis em memória, que os processos sensíveis a essas interrupções devem estar a ler. O processamento deve ser feito pelos processos e não pelas rotinas de interrupção, cuja única missão é assinalar que houve uma interrupção;
- Não se pode estar continuamente a escrever no módulo dos comboios (embora se possa ler), senão o efeito é a interface de visualização piscar muito. Escreva apenas informação quando ela for diferente da que já está no módulo.

6.2 – Estratégia de implementação

Alguns dos guiões de laboratório contêm objetivos parciais a atingir, em termos de desenvolvimento do projeto. Tente cumpri-los, de forma a garantir que o projeto estará concluído nas datas de entrega, quer a versão inicial quer a versão final.

Devem ser usados processos cooperativos para suportar as diversas ações do sistema de controlo dos comboios, aparentemente simultâneas. Recomendam-se pelo menos os seguintes processos:

- Varrimento e leitura das teclas ou botões, tal como descrito no guião de laboratório 5. Pode haver um processo destes para cada conjunto de teclas ou botões, cada um com dois estados (espera tecla carregada, espera tecla liberta), exceto no caso dos botões que mantêm o estado;
- Atualização do estado dos semáforos, de acordo com o estado dos troços (livre ou reservado);
- Passagem de nível. É preciso um estado à espera que a passagem seja ligada e mais dois, um para cada combinação cinzento-vermelho;
- Comportamento do maquinista. É preciso um estado para estar à espera de passar por um sensor e outros dois com comboio parado, para estar à espera de um semáforo verde ou do fim de tempo de paragem na estação. Havendo dois comboios, precisa de um processo para cada um deles. Pode ser a mesma rotina, com o número de comboio como argumento.

Como ordem geral de implementação, recomenda-se a seguinte:

1. Rotinas de implementação do controlo manual, já com ciclos de varrimento de botões e de passagem por sensores (por leitura contínua do módulo dos comboios);
2. Rotinas de gestão da ocupação dos troços e controlo automático dos semáforos, testadas usando os botões para libertar/ocupar cada troço e até os leds para ver o estado de cada troço. Tudo sem pôr os comboios a andar;
3. Tire um dos comboios (basta colocar um # no início da linha que descreve um comboio, no ficheiro buildTrains.txt), para reduzir a complexidade. Quando funcionar, volte a colocar o comboio;
4. Verifique o funcionamento do comportamento do comboio (pode continuar a usar os botões e leds para controlar a ocupação dos troços);
5. Ponha as interrupções a funcionar, tanto as dos sensores como as do relógio de tempo real;
6. Ponha a passagem de nível a funcionar, bem como o tempo de paragem nas estações;
7. Verifique o funcionamento global, com os dois comboios.

MUITO IMPORTANTE – O módulo dos comboios é um sistema de tempo real, em que a deteção de bugs não é fácil com *breakpoints* e execução passo a passo. Mesmo que se pare o programa, os comboios continuam em andamento. Por isso, use extensivamente a interface de controlo (mostradores e leds, nomeadamente), para saber continuamente o estado do sistema e o que está a suceder. Os mostradores são particularmente úteis, pois podem acumular até 40 caracteres de história.

Finalmente:

- Faça PUSH e POP de todos os registos que use numa rotina e não constituam valores de saída. É muito fácil não reparar que um dado registo é alterado durante um CALL, causando erros que podem ser difíceis de depurar. Atenção ao emparelhamento dos PUSHs e POPs;

- Vá testando todas as rotinas que fizer e quando as alterar. É muito mais difícil descobrir um bug num programa já complexo e ainda não testado;
- Estruture bem o programa, com zona de dados no início e rotinas auxiliares de implementação de cada processo junto a eles;
- Não coloque constantes numéricas (com algumas exceções, como 0 ou 1) pelo meio do código. Defina constantes simbólicas e use-as depois no programa;
- Produza comentários abundantes, não se esquecendo de cabeçalhos para as rotinas com descrição, registos de entrada e de saída;
- Não duplique código (com copy-paste). Use uma rotina com parâmetros para contemplar os diversos casos em que o comportamento correspondente é usado.