

Supervisão Inteligente 2015 / 2016

Trabalho Prático 1

Projeto e Implementação de um Supervisor Inteligente

DEE / FCT/ UNL

Introdução

O primeiro trabalho prático de Supervisão Inteligente tem como objetivo abordar os conceitos base associados ao desenvolvimento de um Sistema de Controlo e Supervisão Inteligente (SCSI). O SCSI será aplicado ao armazém automático existente no Laboratório da disciplina, Fig. 1.

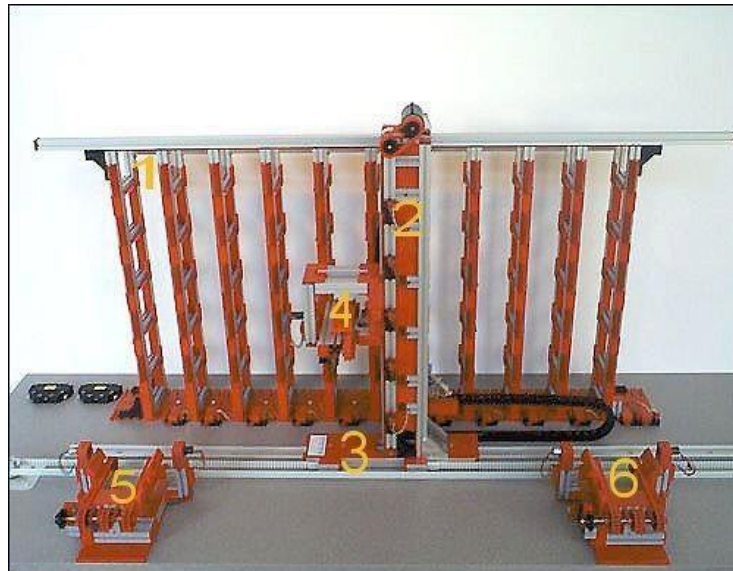


Figura 1 - Armazém Automático

Neste trabalho será implementada uma arquitetura de Supervisão Inteligente constituída por 5 módulos: Planeamento, Monitor, Despacho, Diagnóstico e Recuperador de Erros, Figura 2.

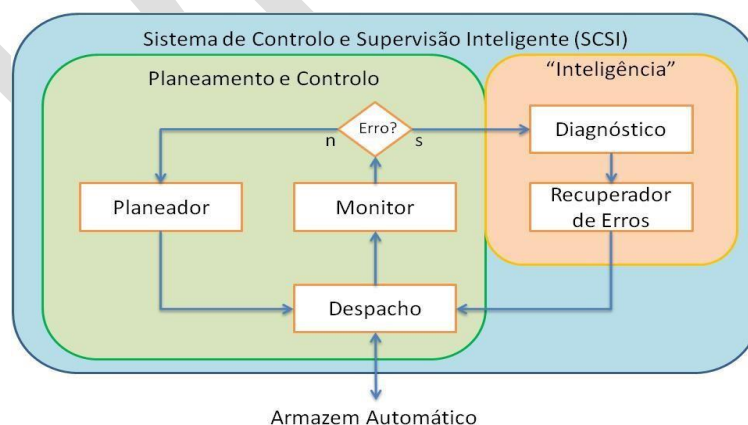


Figura 2 - Arquitetura do SCSI

As tecnologias a utilizar no desenvolvimento deste trabalho são as seguintes: **CLIPS**, **Linguagem C / C++** e **Linguagem Java**.

O CLIPS é uma ferramenta que fornece um ambiente completo para a construção de um Expert System (Sistema Pericial) que tem como principais características:

- ✓ Representação de Conhecimento, através da implementação de regras.
- ✓ Portabilidade, CLIPS é escrito em C e pode ser utilizado em diferentes Sistemas Operativos
- ✓ Integração, é possível colocar código processual através de sub-rotinas, e.g., chamadas a rotinas Java a partir do CLIPS, por mecanismos do estilo *Triggers*.

Os IDEs (Integrated Development Environmets) sugeridos que suportam o desenvolvimento de aplicações escritas nas **linguagens C/C++ e Java** são o [*Qt Creator da Digia \(c e c++ \)*](#) e o [*Intellij Idea da JetBrains \(Java \)*](#).

Este trabalho estará dividido em duas partes:

A primeira parte (3 semanas - data de entrega a **11 de Outubro**), terá como objetivo a implementação do módulo de Despacho e uma familiarização com as linguagens a utilizar e com o CLIPS. Esta parte será dividida em três fases:

I. Setup / Instalações e início do módulo de Despacho (1 aula)

Recuperação do trabalho de STR. Interação com simulador e *hardware*.

II. Desenvolvimento do módulo de despacho (1 aula)

Desenvolvimento do módulo de despacho, em diferentes níveis de abstração, utilizando a interface *JNI*.

III. Instalação CLIPS e Desenvolvimento de uma Demo sobre CLIPS (1 aula)

Familiarização com a programação / integração de CLIPS.

Na segunda parte (6 semanas - data de entrega a **22 de Novembro**), serão desenvolvidos os outros módulos do SCSI, novamente em três fases:

I. Planeamento e Controlo do SCSI (2 aulas)

Contacto com o hardware e desenvolvimento de um *Planner* base para a definição e organização das ações a desempenhar pelo Sistema, através do módulo de Despacho. Nesta fase deve ainda ser preparada uma

primeira versão de um módulo Monitor do funcionamento do Sistema, capaz de alertar o utilizador e parar o Sistema quando algum erro for detetado.

II. “Inteligência” do SCSI (2 aulas)

Numa segunda fase, será estendido o SCSI por forma a incluir os dois módulos constituintes da “inteligência” do mesmo - o Diagnóstico e a Recuperação de Erros.

III. Demonstração do SCSI (1 aula)

Nesta ultima fase, após estar tudo completo e devidamente testado no simulador, os alunos **deverão** demonstrar o seu trabalho no Sistema Real.

Arquitetura de Supervisão Inteligente – mais detalhes

As principais funcionalidades dos módulos mostrados na figura 2 são as seguintes:

1. **Planeador** Este módulo permite determinar o que fazer a cada momento, em função do “estado do mundo”. Por exemplo, no caso do *Warehouse*, armazenar uma nova peça implica que uma das estações de I/O deve ser escolhida para receber a referida peça, o transportador deve ser colocado na posição de receção de nova peça e esta, quando recebida, deve ser transportada e armazenada numa determinada célula. A definição da célula pode fazer parte do comando ou ser decidida em função da disponibilidade do armazém.
2. **Despacho**
Este módulo atua sobre o mundo real (seguindo um plano previamente definido). O despacho dispõe basicamente de operações de ativação de atuadores, capazes de garantir a materialização (no mundo real) das decisões do supervisor inteligente (e.g. acionar tapetes rolantes). É também neste módulo que é recebida toda a informação sensorial. Trata-se do único módulo que tem contacto com o *Hardware*.
3. **Monitor**
Verifica se tudo está a decorrer como planeado, detetando eventuais desvios na execução de uma determinada ação. O monitor usa operações de aquisição de informação sensorial fornecidas pelo Despacho, a fim de criar uma representação atualizada do “estado do mundo”. Tudo o que vá contra o previsto deve alterar o funcionamento do SCSI por forma a detetar a razão do desvio e tentar resolve-lo. No *Warehouse*, por exemplo: 1 – se for dada uma ordem para retirar uma determinada peça que se encontre numa célula específica (X,Y); 2 – se a célula (X,Y) está vazia, ou seja, a peça não pode ser retirada; 3 - algo está errado e o monitor deve assinalar tal facto para que a situação seja verificada com mais rigor. Este é um exemplo de erro recuperável, caso a peça esteja armazenada em outra célula, por exemplo.

4. **Diagnóstico**

Módulo ativado quando uma situação de desvio é detetada. O diagnóstico é o especialista que vai investigar as causas do desvio / problema detetado pelo monitor.

5. **Recuperador de Erros**

Este módulo verifica se o erro / problema caracterizado pelo diagnóstico é recuperável (automaticamente) ou não. Caso o erro seja recuperável automaticamente, isto é efetuado e o sistema volta a funcionar normalmente. Em termos práticos isto poderá corresponder a um “plano B” de ações para retomar o normal funcionamento do sistema. Caso a falha não seja recuperável, podem-se alterar as regras de controlo ou pode-se ainda abortar a operação do sistema.

Exemplos de erros possíveis: armazenar peça estando o armazém já cheio, retirar peça estando o armazém ainda vazio e retirar peça de uma célula vazia. O Anexo C mostra uma tabela contendo alguns exemplos de erros que podem ser considerados, entre outros.

ANEXO A – Sensores e Atuadores (Portos e Bits)

WAREHOUSE LEGEND					
11 Actuators			26 Sensors		
PORT	BIT	Definition	PORT	BIT	Definition
4	0	Tower, Left->Right	0	0	1st (Left,X/axis)
4	1	Tower, Right->Left	0	1	2nd
4	2	Tower, Slow motion	0	2	3rd
4	3	Get Part from Warehouse	0	3	4th
4	4	Put Part into Warehouse	0	4	5th
4	5	Lift DOWN	0	5	6th
4	6	Lift UP	0	6	7th
			0	7	8th
4	7	I/O Left, Output	1	0	9th
5	0	I/O Left, Input	1	1	10th (Right)
5	1	I/O Right, Output			
5	2	I/O Right, Input	1	2	Lift Platform, External Sensor (Retrieve Part)
			1	3	Lift Platform, Middle Sensor
			1	4	Lift Platform, Internal Sensor (Store Part)
			1	5	Z-axis above pos 1
			1	6	Z-axis below pos 1
			1	7	Z-axis above pos 2
			2	0	Z-axis below pos 2
			2	1	Z-axis above pos 3
			2	2	Z-axis below pos 3
			2	3	Z-axis above pos 4
			2	4	Z-axis below pos 4
			2	5	Z-axis above pos 5 (PUT)
			2	6	Z-axis below pos 5 (GET)
			2	7	Workpiece IN the LIFT (Presence = 1; absence=0)
			3	0	I/O Station Left, Part On
			3	1	I/O Station Right, Part On

ANEXO B – Exemplos de erros a serem tratados pelo SCSI

TIPO ERRO	TIMING	DESCRIÇÃO	RECUPERAÇÃO
Recuperável	Carregando peça da Station para Elevador	Detecta-se ausência de peça na Estação de entrada, quando o elevador está diante da estação	Re-LOAD INTO LIFT
Recuperável	Carregando peça da Station para Elevador	Detecta-se Elevador vazio após o LOAD da Estação	Re-LOAD INTO LIFT
Recuperável	Retirando peça do armazém para estação de saída	Detecta-se Estação de saída ocupada, quando o elevador está diante da Estação	Alarme, EJECT Piece ...?
Irrecuperável	Armazenando Peça	Detecta-se ausência de peça na Elevador, quando este está diante da célula destino	Irrecuperável.
Recuperação Condicionada	Armazenando Peça	Armazém Cheio	Análise das próximas operações. Se existir um comando RETIRAR, então troca a ordem de execução.
Recuperação Condicionada	Recuperando Peça	Armazém Vazio	Análise das próximas operações. Se ARMAZENAR, então troca a ordem de execução.
Recuperação Condicionada	Retirando peça do armazém para o Elevador	Após o LOAD da célula, detecta-se que o Elevador está vazio	Retira-se outra peça semelhante à desaparecida

Anexo C - Notas Importantes - Cuidados especiais a ter

1. O sensor que deteta a presença de uma peça no elevador é do tipo **active high** (i.e., elevador com peça - valor do sensor é igual a 1; elevador sem peça - valor do sensor é igual a 0), ao contrário de todos os outros sensores do Warehouse que são **active low** (e.g., estações de entrada/saída com peça - sensor == 0; sem peça - sensor == 1).
2. Ter em atenção que os elementos do kit (exceto o dispositivo principal) não têm qualquer proteção física de modo a serem parados nos seus limites aquando se verifique tal situação, pelo que podem “descarrilar”. **Especial atenção deve ser dada ao tapete do elevador, pois avançá-lo para além dos limites aceitáveis provoca, normalmente, a rutura da correia de movimentação do mesmo!**
3. O dispositivo principal não se movimentará ao longo do eixo X a não ser que o tapete do elevador esteja na posição do meio (detetável através de um sensor existente para o efeito). Esta proteção é feita a nível de kit pelos seus circuitos internos.
4. O dispositivo principal que se desloca no eixo X está protegido nos seus limites extremos por interruptores de modo a que este não saia dos carris.
5. No caso do dispositivo principal estar na posição 0 (posição mais à esquerda) quer dizer não só que está em frente da coluna nº0 como também está em frente da estação de entrada e saída de peças nº0 (I/O Station Left), e simetricamente, se estiver na posição número 9 (posição mais à direita) quererá dizer que não só está à frente da coluna nº9, como também está à frente da estação de entrada e saída de peças nº1 (I/O Station Right).
6. Ao contrário dos sensores que estão presentes ao longo do eixo X para informar da posição do dispositivo principal em relação às colunas, os sensores presentes ao longo do eixo Z não informam diretamente a posição do elevador em relação às linhas, mas sim se ele está acima ou abaixo de determinada linha de modo a poder deixar ou pegar uma peça, respetivamente. O mesmo se diz sobre a forma como ele deixa/recolhe uma peça nas/das estações de entrada e saída.