# Supervisão Inteligente – Relatório do 1º Trabalho

André Estevam – 38150

Gonçalo Freitas – 37555

Miguel Prego – 34001

**Introdução**

Neste trabalho será implementada a primeira parte de uma arquitetura de Supervisão Inteligente: Planeador e Despacho.

Posteriormente será implementado o resto da arquitetura.

**Implementação**

Para implementação destes dois módulos, foi criado um ciclo infinito onde são chamadas repetidamente duas funções. Cada uma referente a um módulo (Planeador e Despacho).

Estas funções não têm qualquer interação uma com a outra, as suas “decisões” têm como base flags que cada uma afecta na sua execução.

Para este fim, foram criadas estruturas X, Y, Z, L e R, em cada uma foram criadas variáveis correspondentes às possibilidades de movimento nesses eixos. No caso de L e R referimo-nos aos alimentadores esquerdo e direito.

Para fazer a alteração dessas variáveis foram criadas diversas funções, as quais são chamadas, numa outra função para simular o Planeador, consoante o premir da tecla correspondente no teclado. Por exemplo: Ao premir a tecla ‘D’ accionamos a função move\_right() que altera uma variável da estrutura X correspondente ao movimento para a direita.

Para que as coisas funcionem, criamos uma função que simula o Despacho. Esta função contém uma série de condições que verificam as variáveis alteradas nas estruturas e consoante os seus valores actua directamente no armazém.

O grupo optou por uma metodologia que permite apenas ligar um motor de cada vez, não sendo a mais eficiente em termos de deslocação no armazém mas a mais segura para o armazém, protegendo possíveis erros no nosso código.

**Exemplos de Estruturas e Funções**

|  |  |
| --- | --- |
| Descrição de Funções | |
| INIT | Função de inicialização dos canais de Input e Output com calibração do armazém, movendo o braço operacional até este activar um sensor do eixo do x, y e z. |
| Planner | É uma função que contém um switch de todas as teclas funcionais do teclado, com correspondência directa ao nosso programa, de modo a criar um joystick para mais facilmente controlar o armazém. |
| Dispatch | Nesta função são alterados os valores dos portos, de modo a controlar o armazém. A activação é feita por via de alteração de variáveis existentes nas estruturas criadas para esse efeito. |
| Move\_O  Take\_O  Give\_O | Sendo que O representa às várias hipóteses, neste conjunto de funções é alterada uma variável de acordo com o pressionar do teclado, para que o dispatch possa funcionar correctamente. |
| Stop\_O | Mais uma vez O representa várias hipóteses e neste caso é dada a informação de que é desejável parar o motor do eixo O. |

|  |  |
| --- | --- |
| Descrição de Estruturas | |
| X, Y, Z, L, R | Todas elas semelhantes e com a mesma funcionalidade, contêm 3 variáveis que são alteradas pela função existente na mesma estrutura e que é chamada pelas funções move, take e give. |
| Ports | Contém variáveis boleanas para cada um dos sensores existentes, de modo a ser mais fácil consultá-los. Também estão incluídas três funções que devolvem os valores de cada conjunto de variáveis (xSensors, ySensors e zSensors) e uma outra função que actualiza os valores existentes em todas as variáveis desta estrutura |

**Exemplo**

/\*Exemplo da estrutura com as variáveis para o movimento no eixo X\*/

struct {

bool x\_left = 0;

bool x\_right = 0;

bool x\_stop = 0;

void XX (bool left, bool right, bool stop) {

x\_left = left;

x\_right = right;

x\_stop = stop;

}

} X;

/\*Função que altera as variáveis da estrutura X, chamada pela função Planner, caso se pressione a tecla ‘D’\*/

void move\_right() {

if (X.x\_right == 1)

stop\_x();

else

X.XX(0, 1, 0);

Y.YY(0, 0, 0);

Z.ZZ(0, 0, 0);

L.LL(0, 0, 0);

R.RR(0, 0, 0);

}

/\*Parte da função Dispatch() em que se activam os motores\*/

void Dispatch() {

if (X.x\_right == 1)

out\_port(4, 1);

…

}