

SECCAO AUTONOMA DE ENGENHARIA MECANICA

PROJECTO DE AUTOMACAO

Projecto seguido pelo o Senhor Professor

VITOR SANTOS

Ano lectivo 1998-99

Marcos Henriques Da Silva

ÍNDICE

ÍNDICE	1
1. INTRODUÇÃO	3
2. O BRAÇO E SUA COMUNICAÇÃO	4
3. FLUXOGRAMA DO BRAÇO	5
4. DETERMINAÇÃO DOS COMPONENTES ELECTRÓNICOS	6
4.1. Ver	6
4.1.1. Sensor de cores	7
4.1.1.1. As vantagens	7
4.1.1.2. Os inconvenientes	7
4.1.2. Sensor com reflector polarizado	8
4.1.2.1. As vantagens	8
4.1.2.2. Os inconvenientes	8
4.1.3. Conclusão	9
4.1.4. Teste	9
4.2. Tocar	10
5. DESCOBERTA DO BRAÇO	11
5.1. Dados	11
5.1.1. O braço	11
5.1.2. O ambiente	12
5.1.3. O robot móvel	13
5.2. Estudo	14
5.2.1. Primeira parte	14
5.2.1.1. Ponto de origem	14
5.2.1.2. Os limites	14
5.2.1.3. O funcionamento	15
5.2.1.4. Os problemas	15
5.2.2. Segunda parte	16
5.2.2.1. As “words”	16
5.2.2.2. A comunicação	17
5.2.2.3. Os problemas	17
5.2.3. Terceira parte	18
5.2.3.1. Os problemas	18
5.2.4. Quarta parte	19
5.2.4.1. A missão	19
5.2.4.2. A precisão do braço	20
5.2.4.3. Movimentos lineares	20
5.2.4.4. Repetibilidade	20
5.2.5. Quinta parte	21

6. PROGRAMAÇÃO EM LABVIEW	22
6.1. Primeira parte	22
6.1.1. As “strings” e as “words”	22
6.1.2. As malhas	22
6.1.3. Outras funções	22
6.2. Segunda parte	23
6.2.1. Comunicação	23
6.2.2. Problema	23
6.3. Terceira parte	24
6.3.1. As “words”	24
6.3.1.1. “Move”	24
6.3.1.2. “Speed”	24
6.3.1.3. “Zero” e “Nest”	25
6.3.2. A comunicação	26
6.3.2.1. “com”	26
6.3.2.2. Problemas	27
6.4. Quarta parte	28
6.4.1. Placa de aquisição	28
6.4.1.1. Configuração	29
6.4.1.2. “daq”	29
6.4.2. Problemas	31
6.5. Quinta parte	33
6.5.1. Busca	33
6.5.1.1. Diagramas	33
6.5.1.2. Hierarquia	35
6.5.2. Chamada	36
6.5.2.1. Diagramas	36
6.5.2.2. Hierarquia	38
6.5.3. Referência	39
6.5.3.1. Diagramas	39
6.5.3.2. Hierarquia	40
7. TESTES	41
8. CONCLUSÃO	41
9. ANEXOS	42

1. INTRODUÇÃO

O objectivo deste projecto é realizar a programação e adaptação de um braço robot didáctico, para permitir a um robot móvel de utilizar o elevador, sem a actuação de uma pessoa.

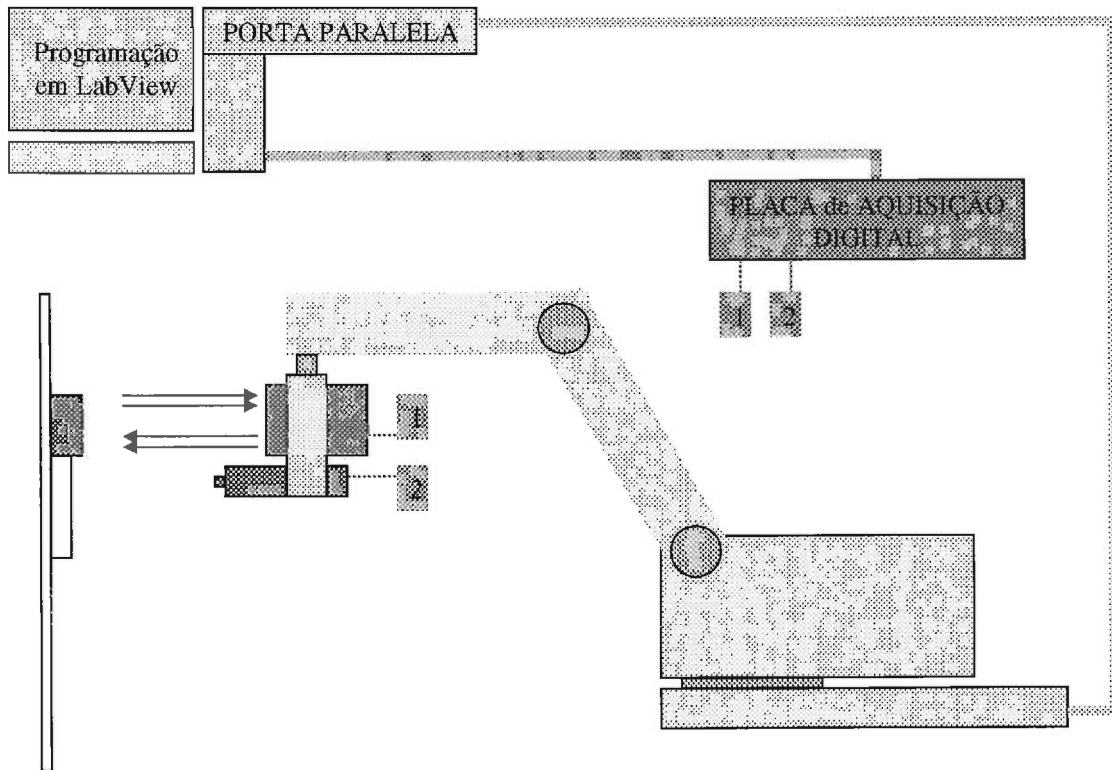
Neste projecto o braço tem de realizar uma busca para detectar o painel, e depois accionar o botão para chamar o elevador.

Para poder alcançar este objectivo, foram realizadas várias etapas para obter este resultado.

No início do projecto foi planeado um programa, que ao longo do projecto se tentou respeitar. Este plano era dividido em quatro grandes partes:

- Primeira parte: determinação dos componentes electrónicos que lhe permitem ver e tocar.
Devia decorrer até 31 de Janeiro.
- Segunda parte: conhecer o comportamento do braço, com o software de origem. De seguida, utilização do braço em ambiente MS-DOS.
Devia decorrer até 15 de Abril.
- Terceira parte: introdução à programação em LabView, para poder programar o braço com este software, mais versátil. Com a utilização deste software, também são utilizadas as portas paralela e série, e a carta de aquisição de dados.
Devia decorrer até 30 de Junho.
- Quarta parte: esta ultima é a mais difícil, sendo aqui que se adicionam todas as diversas componentes permitindo o funcionamento desejado do braço. Aqui, também são realizados os diversos testes para permitir uma viabilização máxima do braço.
Devia decorrer até 15 de Julho.

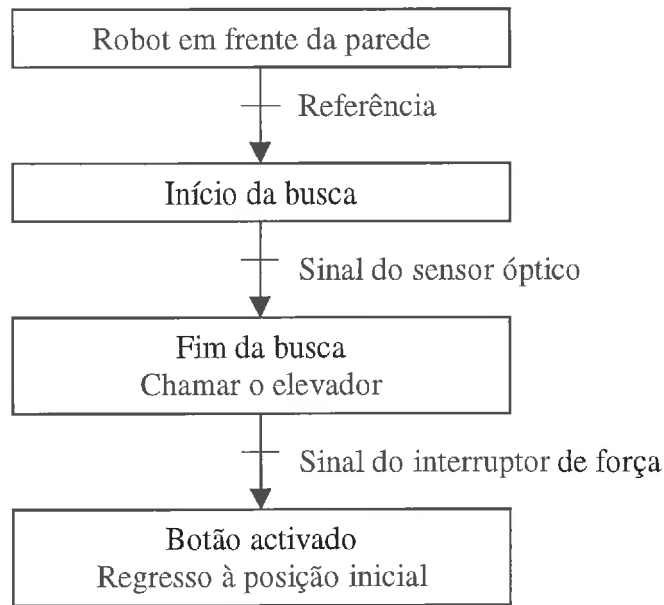
2. O BRAÇO E SUA COMUNICAÇÃO



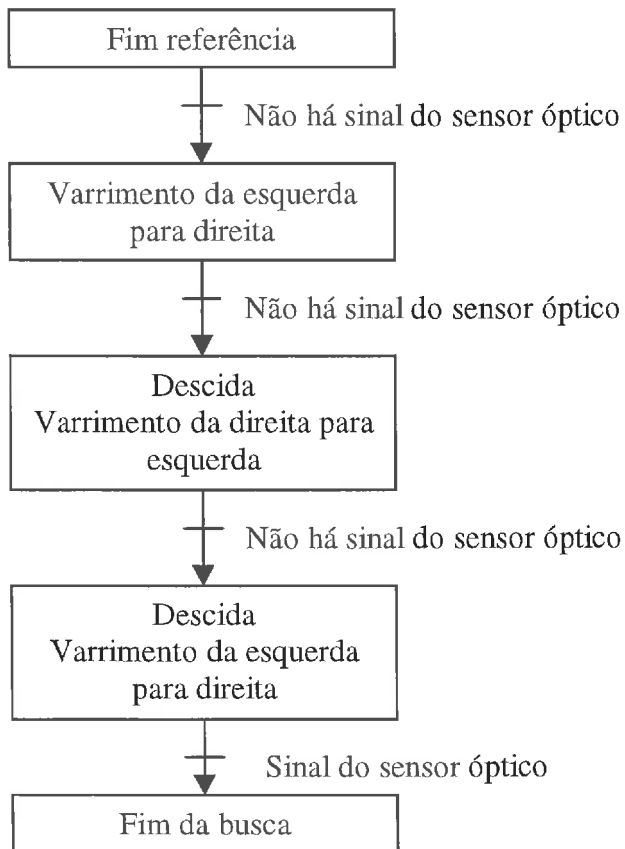
O computador utilizando o LabView para a programação, comunica com o braço através da porta paralela para lhe enviar informações, e através da placa de aquisição para receber informações sobre o sensor óptico e o interruptor de força.

Para detectar o painel é posicionado um reflector, que serve para reflectir a luz enviada pelo sensor óptico.

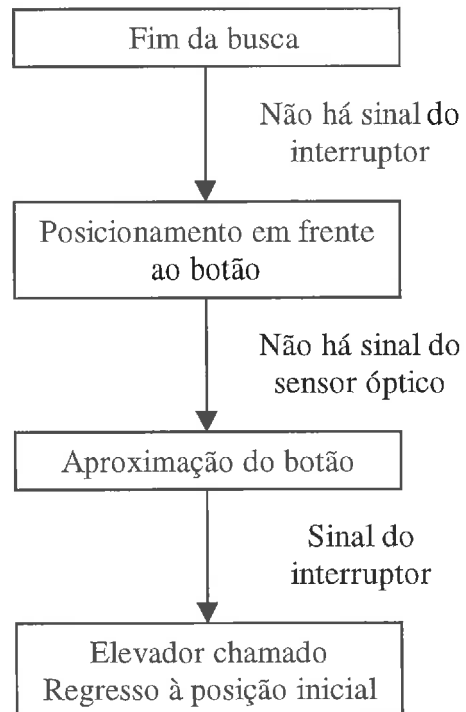
3. FLUXOGRAMA DO BRAÇO



BUSCA



CHAMAR



4. DETERMINAÇÃO DOS COMPONENTES ELECTRÓNICOS

A cabeça permite ao robot ver e tocar, estas características são necessárias para o robot poder utilizar o elevador, tanto no exterior, como no interior. Os dois sentidos são realizados graças a um sensor óptico e a um interruptor de força.

Para o projecto foram escolhidos componentes com uma dada referência, mas foram utilizados componentes com outras referências equivalentes, disponíveis no departamento.

4.1. Ver

Esta função é necessária ao robot, para ele poder encontrar o painel, que permite chamar o elevador.

Esta função servirá igualmente para marcar os pontos de referência do braço robot, para que ele possa alcançar o botão de chamada do elevador.

Estas duas etapas tem de ser realizadas no exterior do elevador, e no interior do elevador.

Actualmente, no mercado estão disponíveis vários tipos de sensores ópticos.

Os tipos de sensores ópticos são os seguintes:

1. Foto-celula de barreira.
2. Foto-celula de reflexão no espelho.
3. Foto-celula de reflexão no objecto.
4. Foto-celula com fibras ópticas.
5. Supressão de primeiro plano.
6. Supressão de segundo plano.
7. Sensores de contraste.
8. Sensor de luminescência.
9. Sensor de distancia.
10. Sensor de cores.
11. Espectro electromagnético.

O projecto limita a escolha do sensor óptico a dois tipos:

1. sensor de cores
2. e o sensor no espelho (reflector polarizado)

4.1.1. Sensor de cores

O sensor de cores pode ser utilizado para a detecção da caixa. Isto é possível, se a caixa de chamada do elevador estiver equipada de um indicador de posição do elevador vermelho. Esta luz vermelha sendo utilizada como alvo do sensor.

Segundo as características do fabricante, temos a disposição dois tipos de sensores de cores. As características de cada um dos sensores são idênticas, menos num ponto. A diferença apresenta-se no número de cores que o sensor pode identificar: segundo o modelo uma ou três cores. No projecto, só precisamos de identificar uma cor.

A referência do sensor de cores é: SENSICK CS1.

As características dadas pelo fabricante estão disponíveis em anexo.

4.1.1.1.As vantagens

1. Adaptação a qualquer tipo de elevadores equipado de um indicador de posição ao alcance do braço robot.
2. Não é preciso montar componentes suplementares para que o sistema possa funcionar.

4.1.1.2.Os inconvenientes

1. Alcance do sensor é bastante pequeno, o que dificulta o funcionamento do sistema.
2. Preço do sensor é bastante elevado.

4.1.2. Sensor com reflector polarizado

O sensor com reflector pode ser utilizado, fixando um reflector ao lado do painel, permitindo assim o uso deste sensor. Sendo então o reflector o alvo do sensor.

Segundo as características do fabricante, temos a disposição vários tipos de sensores com reflector polarizado. As características de cada um dos sensores variam segundo a utilização que desejamos. A diferença principal apresenta-se no alcance mínimo, máximo, e no tipo de saída. O alcance é a característica mais importante para a escolha do sensor.

A referência do sensor de cores é: SENSICK WL160.

As características dadas pelo fabricante estão disponíveis no anexo.

4.1.2.1.As vantagens

1. alcance do sensor é grande, e tem uma grande precisão, o que permite um funcionamento mais seguro.
2. preço do sensor é relativamente baixo.

4.1.2.2.Os inconvenientes

1. A utilização deste sensor no nosso sistema obriga-nos a adaptar o elevador, montando os reflectores. Não podendo assim ser utilizado em qualquer elevador.

4.1.3. Conclusão

Sendo a eficiência o elemento mais importante num sistema, o sensor com reflector destaca-se, neste ponto, em relação ao sensor de core.

O preço sendo igualmente um dos elementos mais importantes na escolha de qualquer componente, o sensor com reflector revela-se neste ponto também o mais económico. Se bem que temos que adaptar reflectores.

Visto as vantagens, e os inconvenientes de cada um dos sensores escolhidos acima. O sensor com reflector destaca-se como o melhor sensor para o projecto.

Referência do sensor escolhido: SENSICK WL160

O sensor acima escolhido não estando disponível no departamento, foi utilizado um sensor equivalente: SENSICK WL170

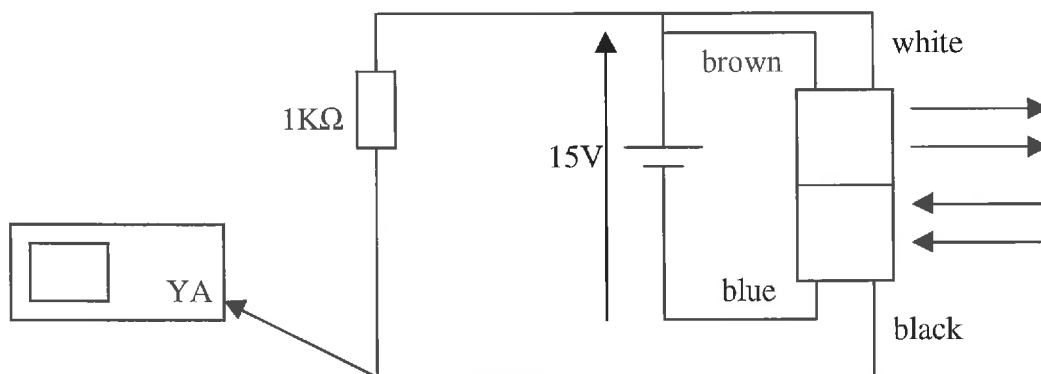
As características, e os esquemas eléctricos de cada um dos sensores estão disponíveis em anexo.

4.1.4. Teste

O teste do sensor óptico permitiu ver de que modo ele actuava, e verificar as suas futuras ligações.

O teste era a visualização, com um osciloscópio, do sinal à saída do sensor. Para isso alimentamos o sensor com uma tensão de entrada de 15V.

Montagem:



Esta montagem permitiu-nos ver qual era o funcionamento, e as possibilidades do sensor.

4.2. Tocar

Depois do braço ter detectado o painel, ele tem que actuar o botão do painel para poder chamar o elevador, tanto no exterior como no interior.

Para confirmar que o robot activou o botão de chamada, o sensor escolhido só deve activar o sinal “high” a partir de uma certa força, equivalente a força de um dedo que activa o botão de chamada do elevador.

Actualmente, no mercado estão disponíveis vários tipos de sensores de medição de forças.

Os diferentes sensores são os seguintes:

1. Interruptores de força.
2. Sistemas equivalentes as maquinas de pesar electrónicas.
3. Sistemas de medição da compressão.
4. Etc... Isto inclui todos os sistemas de medição de forças existentes, mas que não podem ser utilizados neste projecto.

O projecto limita a escolha a um sensor: o interruptor de força. Este é o mais adequado ao projecto graças a sua simplicidade de funcionamento.

A tarefa desempenhada por este sensor não necessita grande tecnologia. O preço deste componente sendo relativamente baixo, não há nenhuma restrição especial. Por isso, o interruptor de força é o sensor para este projecto.

As características do interruptor de força estão disponíveis em anexo.

5. DESCOBERTA DO BRAÇO

Para poder utilizar o braço didáctico no projecto, é preciso conhecê-lo, sendo assim possível mandar-lhe fazer o que desejamos.

Além das suas características físicas, tal como as dimensões, temos igualmente que conhecer os seus limites, quais são os movimentos que ele pode executar, como ele os executa. Isto tudo reunido deve nos permitir de utilizar ao máximo as suas possibilidades.

Para otimizar o trabalho do braço, é igualmente necessário conhecer o ambiente onde ele vai evoluir. Por isso temos que saber qual são as dimensões do robot móvel, dos painéis do elevador, e onde vai se situar o braço no robot móvel.

5.1. Dados

5.1.1. O braço

As dimensões do braço podem ser consultadas em anexo.

As características dos motores estão disponíveis em anexo.

O funcionamento do braço: ele é movido por motores passo a passo, pelo intermédio de correntes que transmitem o movimento até ao grau de liberdade. Este braço possui 5 graus de liberdade (DOF).

O espaço de trabalho do robot é quase equivalente a uns 3/8 de esfera. Nos cálculos do espaço de trabalho só consideramos este 3/8 de esfera. Temos então:

$$V = \frac{3}{8} * \frac{4}{3} * \pi * (R^3 - r^3) \quad \text{com } R=197+150+98$$

$$R=445\text{mm}$$

$$\text{e } r=197\text{mm}$$

$$V = \frac{3}{8} * \frac{4}{3} * \pi * ((445 * 10^{-3})^3 - (197 * 10^{-3})^3)$$

$$V = \frac{1}{2} * \pi * 80 * 10^{-3}$$

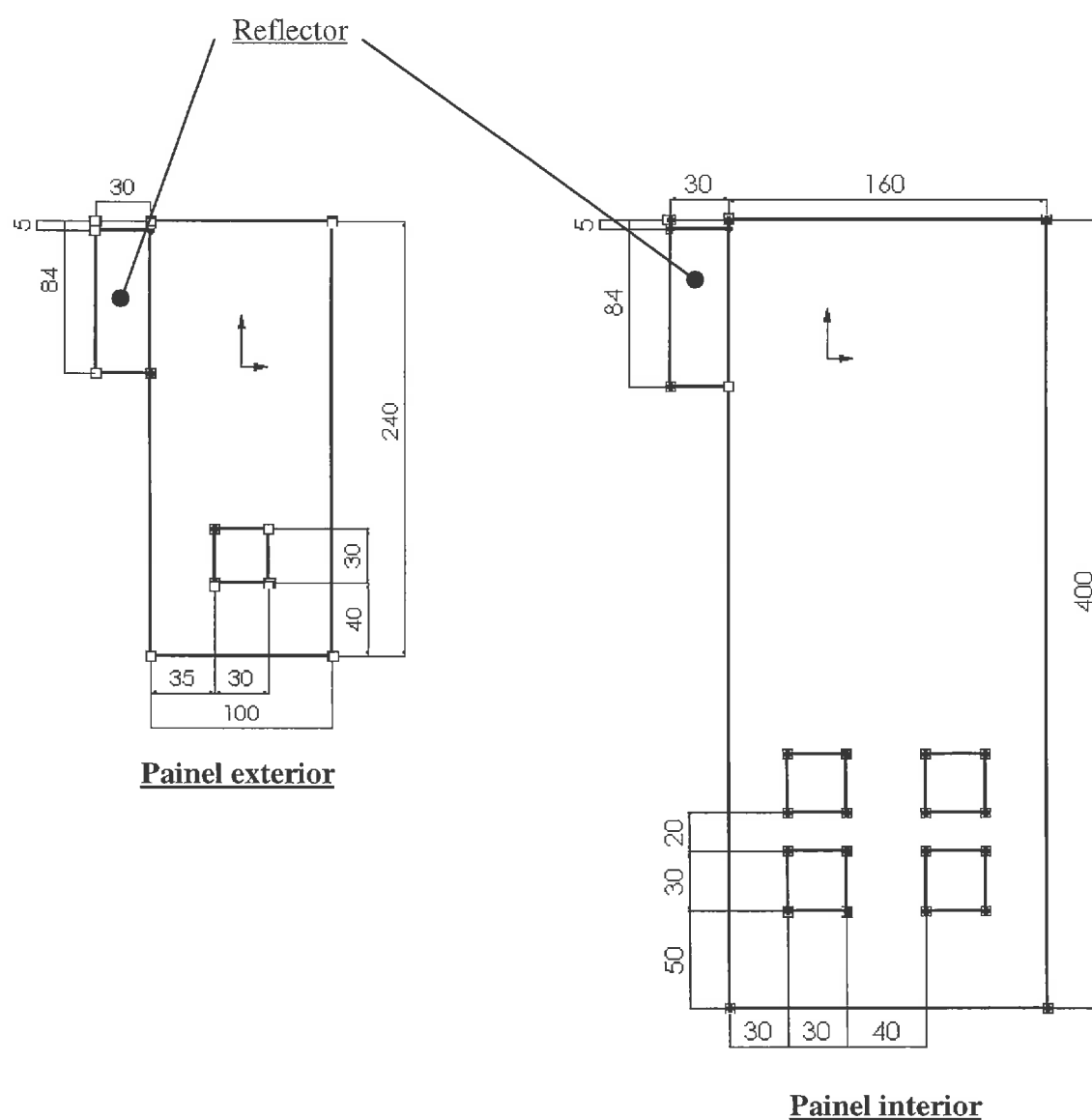
$$V = 126 * 10^{-3} \text{ m}^3$$

5.1.2. O ambiente

O reflector esta colocado na esquina esquerda de cima, em ambos os painéis exterior e interior, permitindo assim a sua detecção pelo braço.

O reflector é o mais pequeno possível para permitir uma maior precisão no seu objectivo: a chamada do elevador.

As dimensões dos painéis e do reflector estão descritas a baixo.



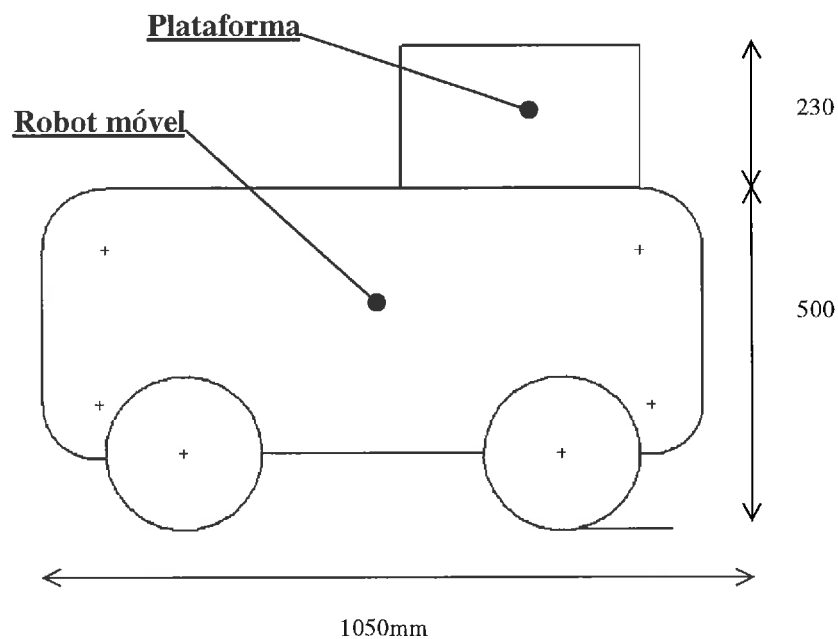
5.1.3. O robot móvel

O braço tem de ser posicionado em cima do robot móvel em função das características do painel, mas também em função das características do robot móvel.

O braço situar-se-á na retaguarda do robot móvel, em cima de uma plataforma própria. Esta posição foi definida em colaboração com os responsáveis do robot móvel.

O robot móvel e o braço sendo muito lentos para entrar no elevador, esta posição foi escolhida para permitir bloquear as portas do elevador, para que a missão do robot não seja perturbada por eventos exteriores durante a realização da missão.

As dimensões do robot móvel e posição do braço estão descritas a baixo:



5.2. Estudo

Nesta etapa do projecto estão incluídas várias partes, avançando passo a passo.

5.2.1. Primeira parte

Nesta parte o robot foi controlado através do software fornecido pelo o fabricante, sendo a maneira mais simples de familiarizar-se com o braço. Aqui estudamos varias sequências já existentes no software, e outras que realizamos.

Uma explicação de utilização do software está disponível em anexo.

5.2.1.1.Ponto de origem

O ponto de origem do braço era dependente da posição na qual ele iniciava a sua operação. Esta posição tinha que ser realizada manualmente para garantir um funcionamento eficiente do braço, quando o braço era utilizado através do software do fabricante. O fabricante indica a posição de referência através de setas posicionadas nos eixos do braço que devem coincidir entre elas.

O software memorizava a posição de origem no início da operação, e regressava a ela quando acabava a operação.

Isto era uma boa maneira de verificar se tivesse ocorrido algum problema em comparando a posição de inicio com a posição final.

5.2.1.2.Os limites

Quando era utilizado o software do fabricante, na realização de uma sequência, não podíamos realizar todos os movimentos desejados, isto porque o software contava os passos, e quando chegava ao seu limite parava o braço, mesmo se este ainda não tinha chegado ao seu limite físico. Tínhamos assim um funcionamento do braço pouco versátil.

Para poder modificar estes limites era necessário modificar a posição de origem, e evitar que o braço funcione mal, batendo num obstáculo sem poder avançar, e os motores continuando a rodar, por exemplo.

Este problema, se acontecesse, inviabilizava todo o resto da sequência introduzindo um erro em todos os movimentos seguintes.

5.2.1.3.O funcionamento

Realizando várias sequências e depois editando-as, descobrimos que o braço funcionava em relação a posição anterior, isto é em coordenadas incrementais.

Este tipo de coordenadas são mais simples para a compreensão do funcionamento do braço.

Facilitando igualmente a descoberta de eventuais erros, e a realização de uma programação sem a ajuda do software.

5.2.1.4.Os problemas

Nesta primeira parte o grande problema foi a comunicação através da porta paralela (porta idêntica a da impressora).

O primeiro problema era devido ao facto do software trabalhar com o sistema operativo WINDOWS 95. Como era utilizado o WINDOWS NT era impossível comunicar.

O segundo problema de comunicação era devido a activação da tecla “Num Lock”, que impossibilitava a comunicação.

5.2.2. Segunda parte

Nesta parte, controlamos o braço a partir de um ambiente MS-DOS, sob WINDOWS 95, utilizando e testando as diferentes “words” reconhecidas pelo braço, com as ferramentas de programação do MS-DOS.

Com esta parte vimos que o braço oferecia mais possibilidades neste ambiente de programação, ao contrario do ambiente software do fabricante.

5.2.2.1.As “words”

As “words” reconhecidas pelo braço, estão disponíveis em anexo.

Nesta parte descobrimos quais eram as “words” necessárias ao funcionamento do braço. As “words” que foram consideradas necessárias para o funcionamento do braço foram:

- “Move” implementado na programação por: Ma1,a2,a3,a4,a5,a6, onde a1,a2,a3,a4,a5,a6 são os passos incrementados nos diferentes motores.

Com esta “word” o braço vai deslocar-se em função das coordenadas definidas.

- “Zero” implementado na programação por: Z.

Com esta “word” o braço vai memorizar a posição actual, habitualmente correspondendo a posição de origem, para regressar a esta ao fim da sequência. Geralmente esta “word” é utilizada no início das sequências.

- “Nest” implementado na programação por: N.

Com esta “word” o braço regressa a posição memorizada com a “word” “Zero”. Geralmente esta “word” é utilizada no fim das sequências.

- “Speed” implementado na programação por: Sa0, onde a0 é um valor de 1 à 5.

Com esta “word” o braço move-se a uma velocidade diferente, sendo 1 a velocidade mínima e 5 a velocidade máxima. Quando uma velocidade é implementada, ela é sempre a mesma até que seja introduzida outra.

5.2.2.2.A comunicação

Nesta parte a comunicação é igualmente realizada através da porta paralela. Aqui como utilizamos as ferramentas de programação do MS-DOS, temos que implementar a função “type_”nome do ficheiro”_>_lpt1” para poder enviar o programa ao braço.

5.2.2.3.Os problemas

Nesta segunda parte o grande problema teve como origem os motores passo a passo. Estes sem “feedback” de controlo da posição, não informam o programa da posição dele. Sendo o seu controlo de malha aberta um problema quando o braço chega ao seu limite físico, bloqueando-se, sem que os motores parem.

Este problema é quase idêntico ao encontrado na parte anterior, resultando então num programa totalmente errado para o resto da operação. Sendo o regresso do braço a posição de origem errado.

5.2.3. Terceira parte

Nesta parte tentamos implementar na programação em MS-DOS as matrizes utilizadas em Robótica Industrial no projecto de prática do robot 5 DOF. Através destas matrizes, criámos utilizar a cinemática inversa para ter uma programação mais rápida. Isto seria realizado com a ajuda do software Matlab.

5.2.3.1.Os problemas

Nesta parte só apareceram problemas que influenciaram o resto do projecto. Estes problemas obrigaram nos a mudar por completo a estratégia de utilização do braço. Sendo irrealizável a realização do objectivo desta parte.

- Descobriu-se que os elos do braço não eram independentes um dos outros, mas são função da posição inicial da garra. As únicas partes totalmente independentes são a base, e a abertura/fechadura da pinça.

Isto é muito importante porque é obrigatório mover em primeiro a garra para a posição final, e depois é que podemos fazer mover em conjunto os dois elos para a posição desejada

Também obriga-nos a impor varias condições para a utilização das matrizes.

- Apareceram igualmente problemas de comunicação, mas que de seguida revelaram se ter origem no funcionamento do reset do braço.

O reset só autoriza o funcionamento do braço, quando o fim de curso interno, tiver accionado pelo esticador da garra.

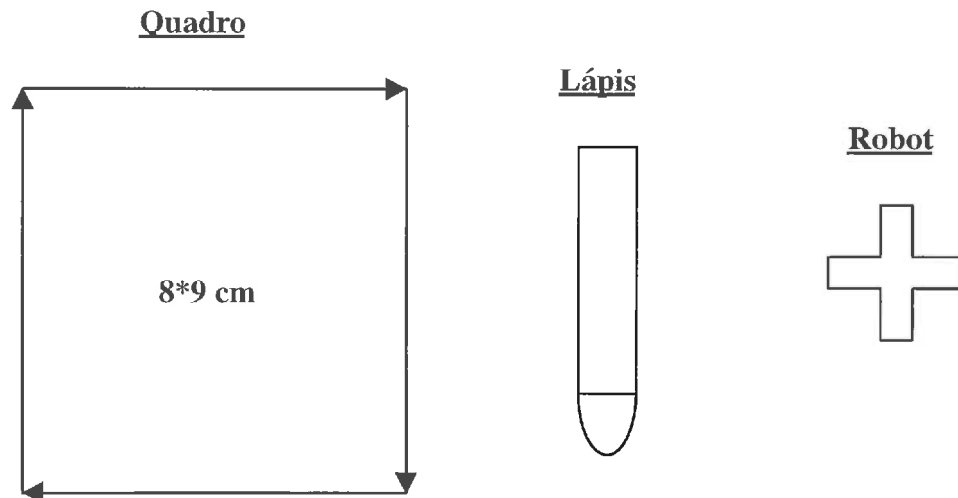
5.2.4. Quarta parte

Nesta parte programou-se o braço para ele executar uma missão para estudar o seu comportamento em relação a um conjunto de movimentos.

5.2.4.1.A missão

O braço tem que começar na posição de origem, imposto pelo construtor, ir buscar um marcador, agarrá-lo com a pinça, e depois realizar um percurso escrevendo numa folha de papel um quadro. E finalmente voltar a pôr o marcador no seu lugar, e regressar a sua posição de origem.

O percurso era o seguinte:



Com esta missão confirmaram-se vários problemas, e descobriram-se outros novos importantes para alcançar o objectivo final do projecto

5.2.4.2.A precisão do braço

A precisão do braço é função de diversos factores, que se podem tornar num problema se não forem resolvidos correctamente.

Em anexo estão disponíveis as características da busca.

- A referência inicial (ou ponto de origem), se não for realizada correctamente, a operação pode tornar-se difícil e errada, o que se torna ao decorrer da missão um erro de precisão.

Para que esse problema se torna muito importante, temos que realizar antes do início de cada operação uma verificação do ponto de origem

- A melhor opção para a realização de uma referência é utilizar o comando manual através do software do fabricante.
- O mecanismo do braço, pela sua construção, com engrenagem e correntes, induz naturalmente erros que nos impede ter uma grande precisão. Mas no nosso caso este erro não é significativo, porque não induz grandes erros, só se for pequenos incrementos.
- Nos limites físicos, o braço força sem avançar, mas o motores eles continuam a rodar, devido ao facto de eles terem uma malha aberta, como foi explicado na parte anterior (terceira parte), o que induz um erro quando este problema se repetir muitas vezes, ou erro de precisão se este problema durar um momento curto.

5.2.4.3.Movimentos lineares

Analisando o quadro desenhado pelo braço, reparou-se que quando o robot se desloca entre dois pontos distantes, ele não se move segundo uma trajectória linear. Para permitir este movimento linear, temos que definir vários pontos entre o ponto inicial e final, tanto no plano XY como em Z.

Um exemplo de percurso esta disponível em anexo.

5.2.4.4.Repetibilidade

Ela é boa se nenhum erro, descrito no parágrafo “precisão” ocorrer. Neste caso o erro que mais se realiza é o facto do braço estar bloqueado, e os motores continuarem a rodar.

5.2.6. Quinta parte

Na quinta e última parte, realizou-se um programa de simulação de busca, tal como se realizará no elevador.

Para poder realiza-lo no ambiente MS-DOS, realizamos em primeiro uma sequência com o software do fabricante. Esta fase permitiu ver qual seriam as “words” necessárias para a realização da busca. As “words” correctas, foram anotadas para serem ulteriormente implementadas na programação em LabView.

O programa realizado com o software do fabricante, e no ambiente MS-DOS estão disponíveis em anexo.

Nesta parte igualmente, teve-se a ideia da velocidade de execução de um programa pelo braço. Infelizmente, o braço ainda demora bastante tempo para realizar a sua tarefa.

Esperamos que na execução do seus objectivo, a sua lentidão não venha a perturbar a operação toda.

6. PROGRAMAÇÃO EM LABVIEW

Esta parte do projecto é a parte que se aproxima mais da finalidade do projecto. Nela encontramos tudo o que foi visto nas partes anteriores, permitindo nos assim evoluir mais rapidamente com os problemas ligados ao braço, permitindo-nos concentrar nos problemas de programação, que se podem tornar difícil pela falta de conhecimento do software.

6.1. Primeira parte

Nesta parte foram realizados exercícios fornecidos pelo fabricante do software para estudar as várias ferramentas necessárias à programação do braço.

6.1.1. As “strings”, e as “ words”

As “strings” corresponde na nossa programação a um carácter da “word”, que será enviada para o braço como se fosse um comando. Ao juntar todas as “words” obteremos um programa com os movimentos que desejamos.

Nisto destacaram-se os comandos ligados as “string”, que nos vão permitir a realização das “words”, através de funções que adicionam as varias “strings”.

6.1.2. As malhas

Nesta parte também estudou-se os vários tipos de malha, destacando-se a função “while loop”, que nos permite realizar varias “loop”. No nosso caso ciclos “for” com as condições que nos desejamos.

6.1.3. Outras funções

Mas a maioria das funções necessárias a toda programação foram descobertas ao decorrer da programação do braço. Alargando os conhecimentos da versatilidade do software LabView.

6.2. Segunda parte

Nesta parte realizou-se pequenos programas de comandos simples para mover o braço, verificando se ele aceitava bem a programação com este software. Isto utilizando “string” simples numa primeira fase, e a seguir utilizando as funções que nos permitem realizar “words”.

Quando conseguimos o funcionamento do braço através das “words” todas, de seguida juntamos estes comandos todos, realizando um programa, para verificar se ele conseguia realizar uma operação sem qualquer problema.

6.2.1. A comunicação

Para poder enviar os comandos que se programaram no LabView, temos que comunicar com o robot através de uma porta série, pelo o intermédio da porta paralela.

6.2.2. Problemas

Mas para realizar esta operação foi necessário configurar a porta série, com a porta paralela para podermos comunicar. Isto tornou-se um problema sendo necessário descobrir e inserir uma função num ficheiro *.ini do software. Sendo assim possível comunicar com o braço.

A função inserida é a seguinte:

```
labview.serialDevices=COM1; COM2; COM3; COM4; COM5; COM6;  
COM7; COM8; COM9; COM10; LPT1; LPT2; LPT3; LPT4;
```

Esta função configurava as portas todas ligadas à porta série. Sem esta função é impossível comunicar com o braço. Isto impediu-nos, avançar rapidamente, porque o problema era parecido com os problemas encontrados nas partes anteriores.

6.3. Terceira parte

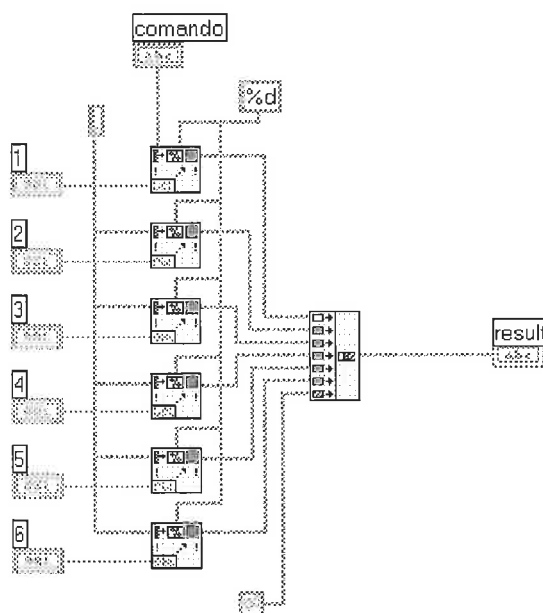
Na terceira parte começou-se a programação em LabView do braço com o objectivo de realizar a busca. Para isso baseou-se no programa realizado com as ferramentas do MS-DOS, copiando um equivalente em LabView.

No princípio, realizou-se “words” equivalentes as da implementadas nas programações anteriores, simulando e verificando-as independentemente. Quando estas pequenas partes foram todas realizadas sem problemas, fez-se um primeiro agrupamento destas.

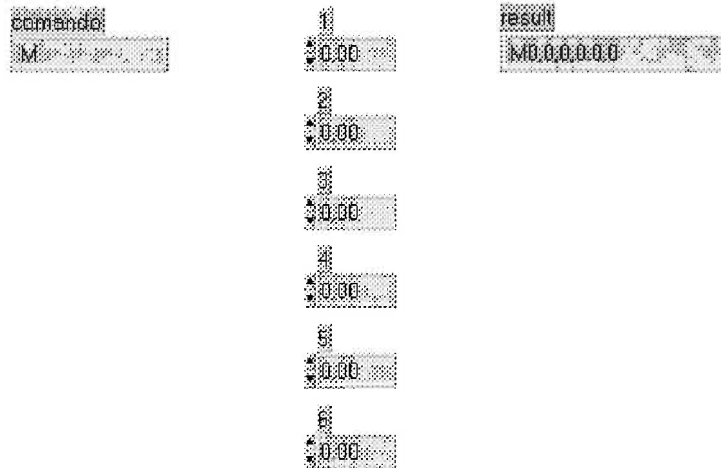
6.3.1. As “words”

6.3.1.1. “Move”

O diagrama seguinte corresponde a implementação da função “Move” na programação no software LabView.



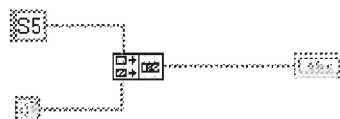
No painel seguinte, que corresponde ao diagrama, podemos visualizar o comando que o robot vai receber.



No painel “comando” está inserido o comando “M”, que corresponde como explicamos anteriormente, ao comando “Move”. Depois encontramos as seis coordenadas, e no painel “result” a “word” que o robot vai executar.

6.3.1.2.Speed

O diagrama seguinte corresponde a implementação da função “Speed” na programação no software LabView.



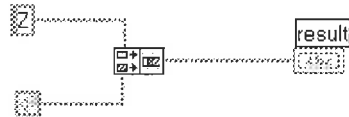
No painel seguinte, que corresponde ao diagrama, podemos visualizar o comando que o robot vai receber.



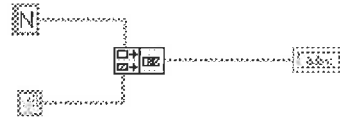
No painel “result” está inserido o comando “S”, que corresponde como explicamos anteriormente, ao comando “Speed”. Depois encontramos “5” que corresponde a velocidade máxima. É esta “word” que o robot vai executar.

6.3.1.3. “Zero” e “Nest”

O diagrama seguinte corresponde a implementação das funções “Zero” e “Nest”, que funcionam juntas, na programação do software LabView.



“Zero”



“Nest”

Nos painéis seguintes, que correspondem aos diagramas, podemos visualizar os comandos que o robot vai receber.



“Zero”



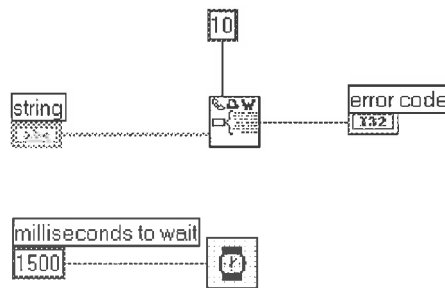
“Nest”

Nos painéis “result” estão inserido os comandos “Z” e “ N”, que correspondem como explicamos anteriormente, aos comandos “Zero” e “Nest. São estas “word” que o robot vai executar.

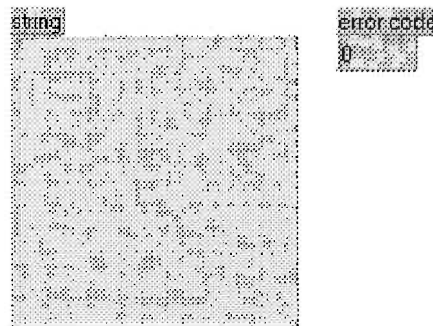
6.3.2. A comunicação

6.3.2.1. “Com”

O diagrama seguinte corresponde a implementação da função “Com”, na programação do software LabView. Este é uma função programada.



No painel seguinte, que corresponde ao diagrama, podemos visualizar os comandos que o robot vai receber.



No painel “string” está normalmente escrito o comando que o robot vai receber. No painel “error code” aparece um código de erro se houver um problema de comunicação.

A lista dos códigos de erro estão disponíveis em anexo.

6.3.2.2.Os problemas

A realização dos comandos dos parágrafos anteriores, não foi muito simples, porque apareceram vários problemas.

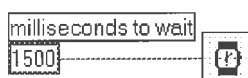
1. A comunicação

Foi o maior problema desta parte, e levou bastante tempo a resolver.

Descobrimos que o LabView enviava demasiadas informações num espaço de tempo reduzido, causando um engarrafamento à entrada do braço, de seguida provocava um cancelamento da operação.

Isto é, o braço é demasiado lento na execução dos movimentos em relação a velocidade com a qual o LabView envia as informações. Para resolver este problema foi inserido, entre as partes da programação mais críticas, uma temporização suficiente para impedir o engarrafamento à entrada do robot.

A função de temporização é a seguinte:



Onde no quadro azul está inserido o tempo (em milisegundos) da pause.

2. O reset

Para resolver o problema da comunicação anterior, que era muito simples, foi muito difícil, porque o reset do braço, como na parte da descoberta do braço voltou a dar problemas cancelando a operação, não sabendo se era um problema de comunicação ou de reset.

Desta vez resolveu-se o problema neutralizando o reset. Sendo depois mais fácil resolver o problema da comunicação.

A partir desse momento, os problemas devidos ao engarrafamento foram muito mais simples de encontrar e de resolver.

6.4. Quarta parte

Nesta parte os sensores foram integrados na programação. Permitindo assim ter um programa que seja realmente eficiente.

O primeiro, e o mais importante, a ser instalado na placa de aquisição, foi o sensor óptico. Antes de ser utilizada a placa de aquisição, o sensor já tinha sido integrado no programa mas era uma constante. Não sendo representado o estado real do sensor.

Depois foi instalado o interruptor de força, para permitir ao programa de decorrer, como foi especificado no início do projecto.

E em último foram integrados os vários fins de curso para poder realizar a referência automaticamente.

6.4.1. A placa de aquisição

É ela que nos permite comunicar entre o programa e os sensores. Sem ela, seria impossível, realizar uma parte essencial deste projecto. Na placa que dispomos no laboratório, só vão ser utilizadas os ports digitais.

Os números das entradas/saídas digitais da placa, e igualmente as características destas estão disponíveis em anexo

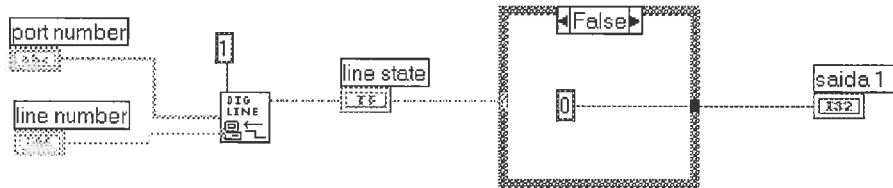
6.4.1.1.A configuração

A configuração da placa é à dada pelo fabricante. Não foi necessário nenhuma alteração. Só foi necessário instalar a placa.

Na nossa utilização as entradas/saídas digitais só vão ser utilizadas como entradas, lendo o sinal gerido pelo sensor ou pelo interruptor.

6.4.1.2. “daq”

O diagrama seguinte corresponde a implementação da função “daq”, na programação no software LabView. Está representada na posição “high”.



Nos painéis seguintes, que correspondem ao diagrama, podemos visualizar as principais informações desta função.



Nos painéis “port number” e “line number” são implementados os números dos slots em que deve ser lido o sinal. O painel “line state” permite-nos visualizar em que estado esta a entrada (“low” ou “high”). E o painel “saída 1” corresponde a informação que o programa vai ler, para depois enviar os comandos correctos ao robot.

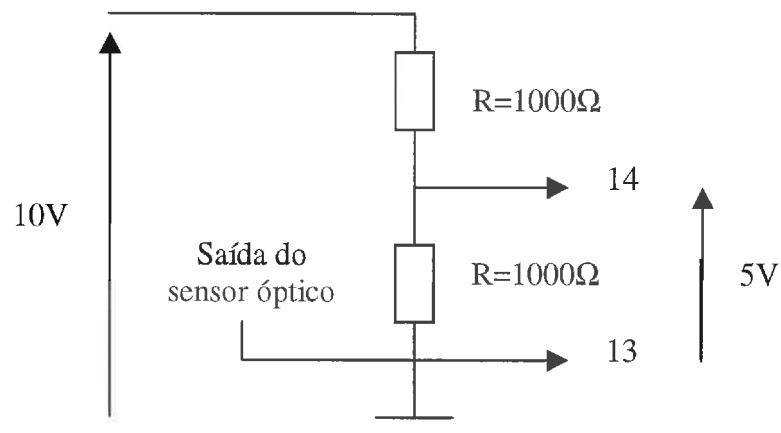
6.4.2. Os problemas

Quando, pela primeira vez utilizamos as funções que necessitavam da placa na programação, elas não funcionavam, porque havia um problema com a placa.

O problema foi resolvido com a instalação e configuração da placa de aquisição. Mas foi necessário instalar novamente o software LabView, porque o problema ainda estava presente devido à versão do LabView inicialmente instalada, que não incluía a utilização da placa de aquisição.

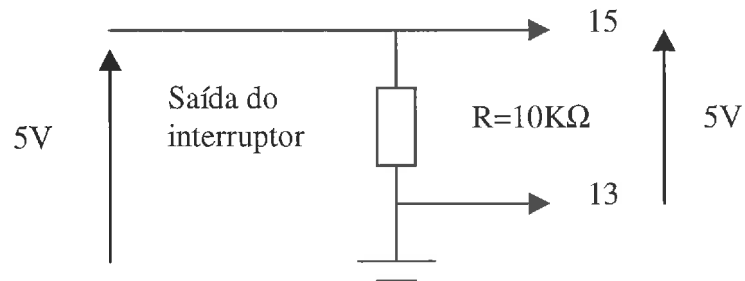
Os problemas de software resolvidos, apareceu nos o problema com a aquisição de dados. De facto a placa só aceitava em entrada digital, um sinal de 0 à 5V, (sem intensidade, cf. anexo: características da placa). Para resolver o problema foi implementado uma ponte divisora de tensão, sendo a tensão de saída do sensor ao mínimo superior à 9V.

Esquema de ligação da saída de sensor:



Apareceu igualmente um problema de estabilidade com o estado “low” do interruptor de força. Isto foi resolvido com a ligação à massa através de uma resistência, para evitar um curto-circuito.

Esquema de ligação do interruptor:



No nosso projecto o interruptor está alimentado por uma alimentação de 5V, independente da alimentação do sensor óptico, que está alimentado em 10V.

A resistência permite estabilizar o sistema quando o interruptor está em estado "low".

6.5. Quinta parte

Numa última parte programou-se as diferentes outras partes necessárias a execução do objectivo do braço, como:

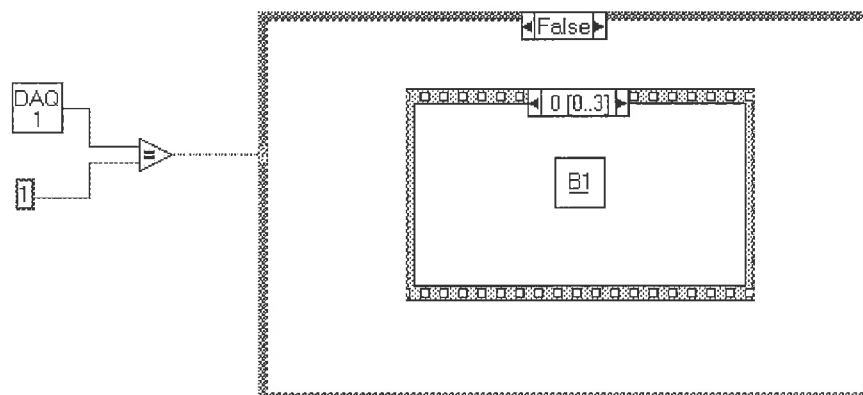
- a busca
- a chamada do elevador
- o ponto de referência.

6.5.1. Busca

A programação da busca é um programa geral agrupando vários sub-programas. Estes estão disponíveis no anexo.

6.5.1.1. Diagramas

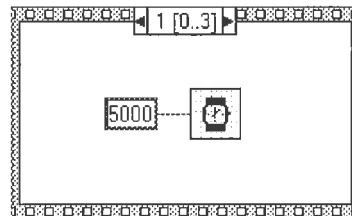
O diagrama da programação de busca:



O sub programa “B1” está representado em anexo, com o seu diagrama, e sua hierarquia.

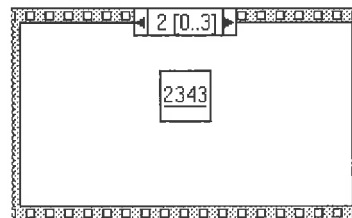
As diversas sequências do “Movie”

Sequência 1/3



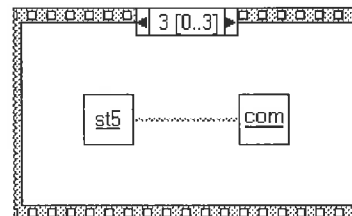
Aqui temos uma temporização de cinco segundos para que não acontecem engarrafamentos.

Sequência 2/3



O sub programa “2343” esta representado em anexo, com o seu diagrama, e sua hierarquia.

Sequência 3/3

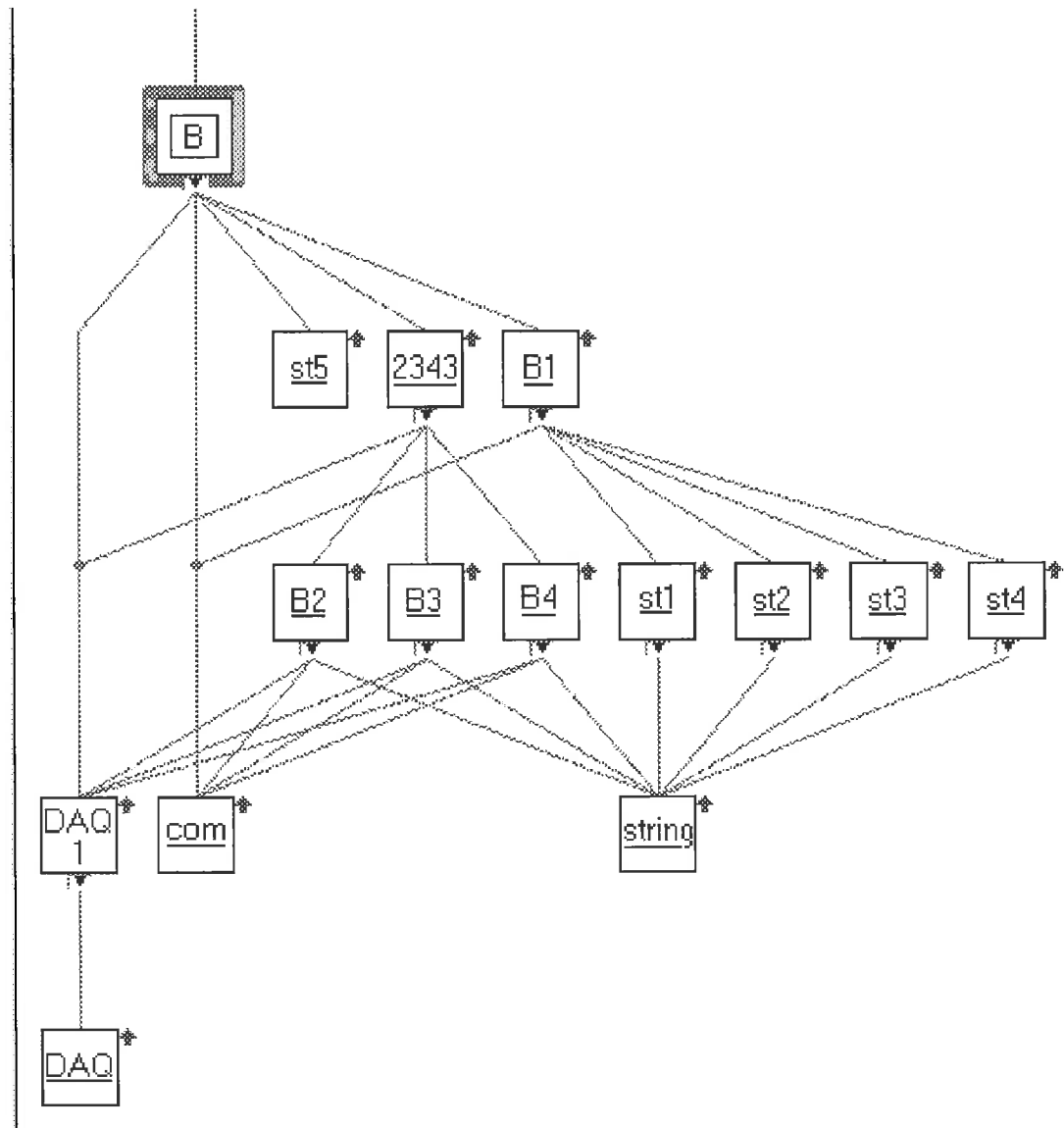


O Sb programa “st5” está representado em anexo, com o seu diagrama, e sua hierarquia.

O sub programa “com” esta representado no parágrafo 6.3.2.1.

6.5.2.1. Hierarquia

Aqui podemos ter uma ideia do funcionamento do programa com a hierarquia do programa principal:



Os programas que não estão representados nesta parte estão disponíveis em anexo.

Os programas são os seguintes: “B1” a “B4”, ”2343”, “st1” a “st5”, “daq1”

Os outros programas estão representados nos parágrafos anteriores, tal como: “string” no parágrafo 6.3.1.1, “daq” na parágrafo 6.4.1.2, “com” no parágrafo 6.3.2.1.

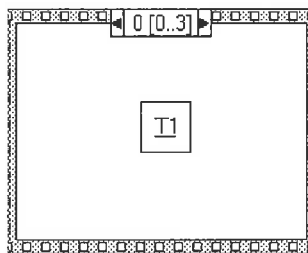
6.5.2. Chamada

A programação da chamada é um programa geral grupando vários sub-programas. Estes estão disponíveis em anexo.

6.5.2.1. Diagramas

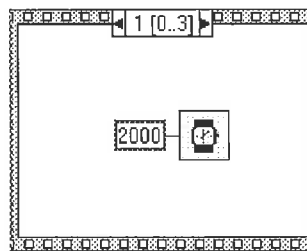
O diagrama da programação da chamada é composto por um “movie”:

Sequência 0/3:



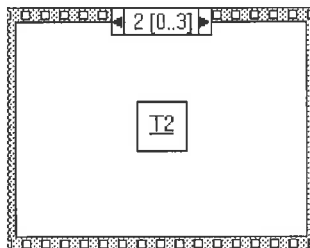
O sub programa “T1” está representado em anexo, com o seu diagrama, e sua hierarquia.

Sequência 1/3:



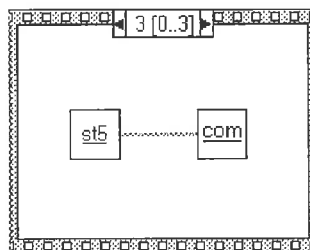
Aqui temos uma temporização de cinco segundos para que não aconteçam engarrafamentos.

Sequência 2/3:



O sub programa “T2” esta representado em anexo, com o seu diagrama, e sua hierarquia.

Sequência 3/3:

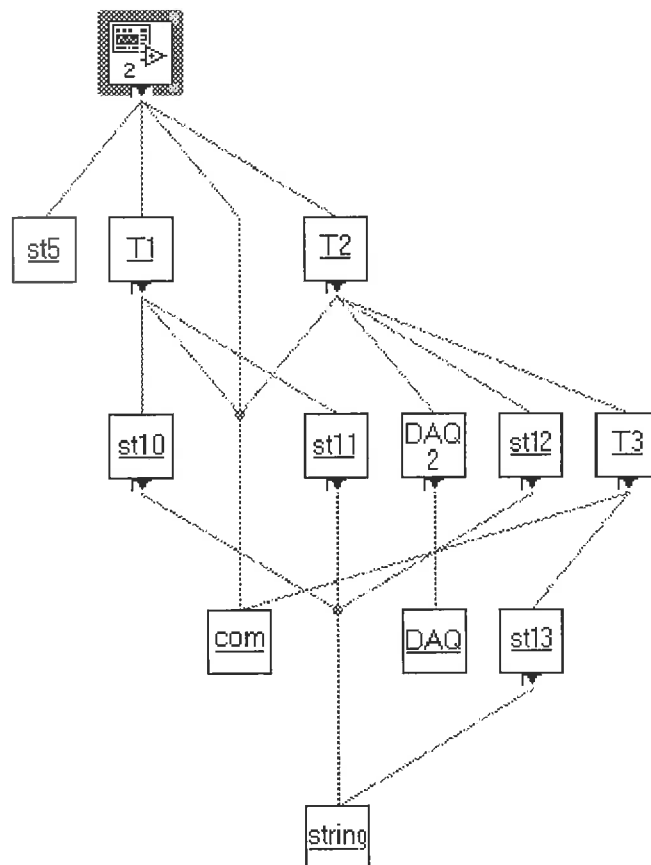


O Sub programa “st5” esta representado em anexo, com o seu diagrama, e sua hierarquia.

O sub programa “com” esta representado no parágrafo 6.3.2.1.

6.5.2.1. Hierarquia

Aqui podemos ter uma ideia do funcionamento do programa com a hierarquia do programa principal:



Os programas que não estão representados nesta parte estão disponíveis em anexo.

Os programas são os seguintes: “B1” a “B4”, ”2343”, “st1” a “st5”, “daq1”

Os outros programas estão representados nos parágrafos anteriores, tal como: “string” no parágrafo 6.3.1.1, “daq” na parágrafo 6.4.1.2, “com” no parágrafo 6.3.2.1.

6.5.3. Referência

A programação da referência é um programa geral grupando vários sub-programas. Estes estão disponíveis em anexo.

Esta parte não foi infelizmente levada a cabo. Aqui só está disponível o sub programa para a referência da base do robot.

Para poder realizar a referência vai ter de ser instalados vários fim de curso no braço. A posição deles será definida em função do tipo de fim de cursos utilizados.

6.5.3.1.Diagrama

Este diagrama corresponde a programação da referência da base do robot.

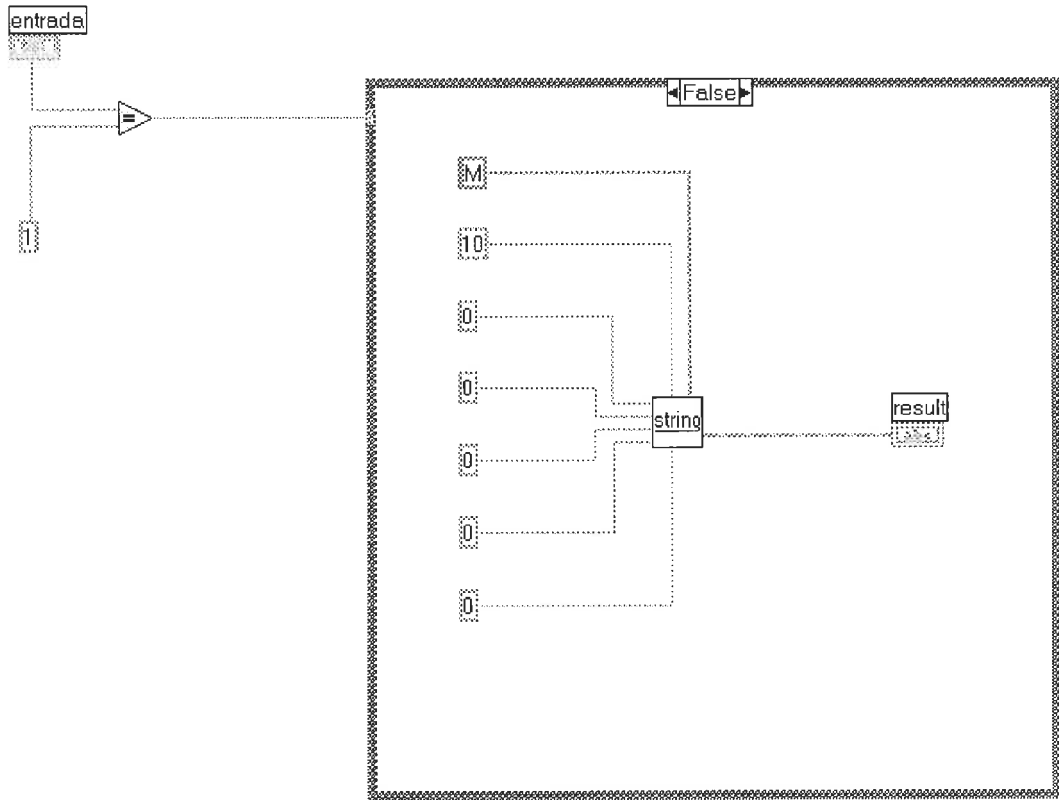
Para poder realizar a referência nos outros eixos, basta modificar os parâmetros da "Move".

Isto é sucessivamente:

1. M0,10,0,0,0,0
2. M0,0,10,0,0,0
3. M0,0,0,10,0,0
4. M0,0,0,0,10,0
5. M0,0,0,0,0,10

Ao fim de realizar este sub programas, eles devem ser inseridos num programa principal do tipo "Movie", onde cada sub programa correspondera a uma sequência.

Diagrama da referência da base.



Nesta parte esta representada o painel do sub programa da referência da base:



O painel “result “ indica-nos qual e o comando a ser enviado ao robot, e o painel “entrada” indica-nos se a referência já esta realizada, através do estado do fim de curso.

6.5.3.2.Hierarquia

Não estando totalmente realizado, não se obteve um esquema da hierarquia de todo o programa.

7. TESTES E SIMULAÇÃO

Os vários teste realizados nas diferentes etapas permitem de reduzir esta parte, ao agrupamento das diversas partes num só programa.

O ultimo teste não se realizo frente ao painel do elevador, mas frente a uma parede simulando o painel do elevador.

Neste último teste realizado ocorreu um problema, que não era relacionado com o braço. A origem do problema era um mau funcionamento do software LabView.

Chegamos a esta conclusão quando arrancamos de novo o software, e não ocorreu qualquer problema, ficando o braço a funcionar sem nenhum problema.

8. CONCLUSÃO

Actualmente não foi possível obter uma total autonomia do braço devido ao facto dos fins de cursos não estarem instalados.

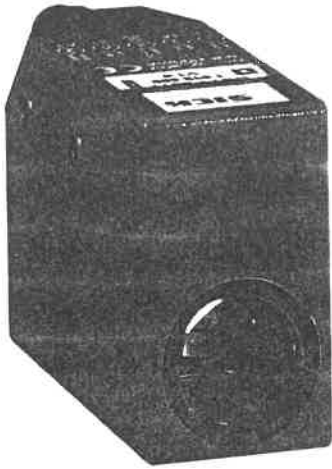
A realização deste projecto além do seu objectivo (permitir ao robot móvel de utilizar o elevador sem a ajuda de uma pessoa), que me ofereceu uma nova perspectiva em relação a robótica e o seu mundo, permitiu-me igualmente adquirir um método que devo utilizar quando me encontro frente a um problema seja qual for a sistema.

Com a minha formação, a resolução de problemas será com certeza a área na qual vou evoluir, este projecto ajudou-me a ter mais maturidade nesta área.

9. Anexos

Índice do anexo

- I. Características dos sensores ópticos.
 - CS1
 - WL 160
 - WL 170
- II. Características do braço.
 - Dimensões
 - Motores
- III. Comandos reconhecidos pelo braço.
- IV. Manual de utilização do software do fabricante.
- V. Sequências utilizadas com o software do fabricante.
- VI. Características da busca.
- VII. Diagramas de todos sub programas e programas.
- VIII. Lista dos erros de comunicação no LabView.
- IX. Características da placa de aquisição.



Scanning Range

12.5 and 60 mm



Range

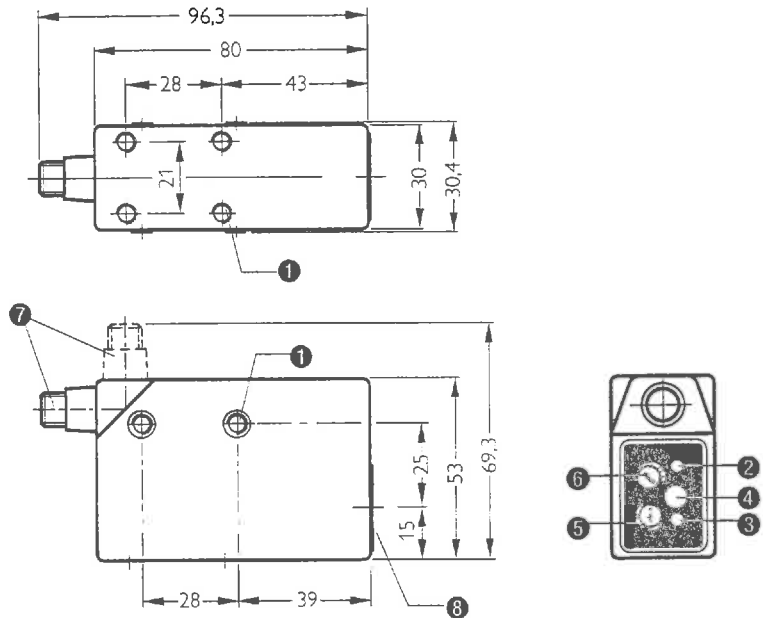
50 to 1000 mm



Features:

- activ three way process
- color detection in reflected and transmitted light
- Programming using the teach-in method (manually or via the input line)
- Individual color selectivity setting
- 2 operating modes: permanent or synchronised (blanked)
- Pilot light
- Switching frequency 1 kHz
- Die-cast housing
- Hinged connector socket (can be rotated through 90°)
- CE

CS 1



- ① Mounting holes M5 – 5,5 deep
- ② Operating indicator (green)
- ③ Function indicator: switching output/teach (yellow)
- ④ Teach button
- ⑤ Program selector switch
- ⑥ color tolerance selector switch
- ⑦ Connector M 12 x 1; 5-pole (can be rotated through 90°)
- ⑧ Lens system (aperture)

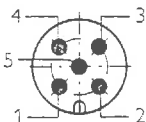
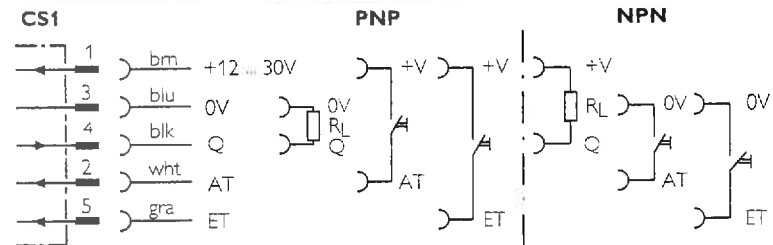
Notes on Setting

In order to simplify setting and operation of the device, the reference color is programmed using the teach-in method.

The color selectivity can be set individually and is defined using the rotary switch. The reference colors can then be stored by simply pressing a button.

The programming procedure can also be carried out automatically via the External Teach (ET) input line. Each pulse causes the sensor to switch to programming mode and determine the color of the scanned object. The system then immediately returns to sensing mode.

Connection Diagram



brn	wht	blu	blk	gra
brown	white	blue	black	gray

Color Sensor CS 1

	CS I	-P36II	-N36II	-P36II	-N36II
Part No.		1 012 858	1 012 862	1 012 859	1 012 863
Scanning distance SD¹⁾		12,5 mm		60 mm	
Scanning range tolerance		± 2 mm		± 9 mm	
Light spot dimensions		2 x 4 mm at distance of 12,5 mm		∅ 13 mm at distance of 60 mm	
Light spot position		Longitudinal			
Scanning range SR²⁾					
With reflector PL 80 A		100 to 250 mm (object-specific)		250 to 1000 mm (object-specific)	
With reflector PL 30 A		50 to 150 mm (object-specific)		200 to 750 mm (object-specific)	
Supply voltage V_S		12 to 30 VDC ³⁾			
Power consumption		≤ 80 mA ⁴⁾			
Residual ripple		< 5 V _{pp} ⁵⁾			
Service life of light source		Average service life 100,000 hrs at 25°C ambient temperature			
No. of colors which can be stored		1 color programmable			
Programming of reference color		Teach-in method (either with Teach-in button or External Teach ET input)			
Color selectivity/tolerance		Variable (Rotary selector switch, 5-step)			
Operating mode		Either permanently or synchronised (blanked)			
Immunity to external light sources		Up to 20 kLx with indirect constant light radiation.			
Switching output		Q (Color detected/not detected)			
Switching method		PNP	NPN	PNP	NPN
Output voltage HIGH		V _S - (≤ 2 V)	V _S	V _S - (≤ 2 V)	V _S
Output voltage LOW		0 V	≤ 2 V	0 V	≤ 2 V
Output current max.		100 mA			
Switching frequency		1 kHz			
Response time		≤ 700 μs			
Timer (drop-out delay)		Variable: 0 ms (run), 20 ms drop-out delay (run delay)			
External Teach ET input		Programming of reference color via input line ET			
Type		PNP		NPN	
Teach-in (programming)		> 12 V to < V _{S max.}		0 V to V _S - (≥ 12 V)	
Run		< 2 V or open-circuited		V _S - (≤ 2 V) or open-circuited	
Teach pulse duration		≥ 0,5 ms			
Blanking input AT		PNP		NPN	
Blanked (triggered)		> 12 V to < V _{S max.}		0 V to V _S - (≥ 12 V)	
Asynchronous		< 2 V or open-circuited		V _S - (≤ 2 V) or open-circuited	
Response time		< 0,2 ms			
Degree of protection		IP 67			
VDE protection class		Ⓜ ⁶⁾			
Protective circuits ⁷⁾		A, B, C			
Ambient temperature T _A		- 10 to + 55 °C			
Storage temperature T _S		- 25 to + 70 °C			
Shock/vibration		IEC 68			
Weight		Approx. 400 g			

1) From edge of housing
2) Dependent on the spectral transmission ratio/absorption of the object
3) Limit values, reverse polarity protected

4) Without load, at 24 V
5) Must not exceed or fall below V_S tolerances
6) Totally insulated, class 2 to VDE 0110/0160 (Rated voltage 50 V)

7) A = V_S terminals, polarity-reversal-proof
B = Switching output Q standard switch on/off, short-circuit and overload protection
C = Spurious pulse suppression

Accessories

Socket M 12 x 1; 5-pole, straight, 2 m connecting cable with integral plug, Part No. 6 008 899, 5 m connecting cable with integral plug, Part No. 6 008 868
Socket M 12 x 1; 5-pole, angled, 2 m connecting cable with integral plug, Part No. 6 008 900, 5 m connecting cable with integral plug, Part No. 6 008 869
Reflector PL 80 A, Part No. 1 003 865, Reflector PL 30 A, Part No. 1 002 314 (see page 236)



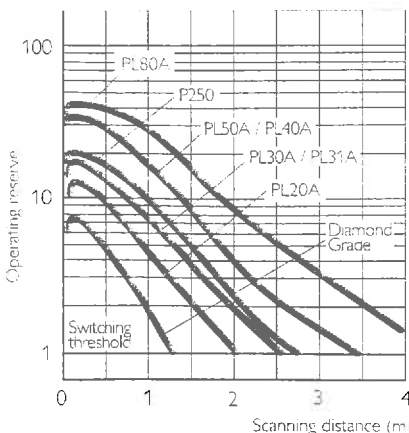
Scanning Range

2 m



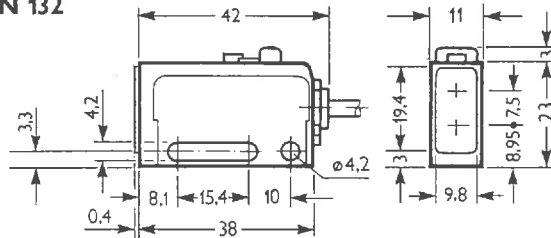
Features:

- Horizontal or vertical design
- Connecting cable or M8 plug, 4-pin
- Adjustable sensitivity (two-turn)
- Red and green indicator LEDs for ease of alignment
- Red LED to show system reserves
- Alarm output
- Test input to test the device and the complete system
- Transistor output short-circuit and overload protected
- Reverse-polarity protection
- Light/Dark-switching selector
- Polarizing filter for detection of targets with reflecting surfaces
- CE / UL / SP

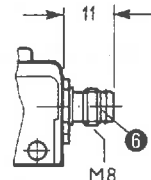


WL 160

WL 160
-P/N 132

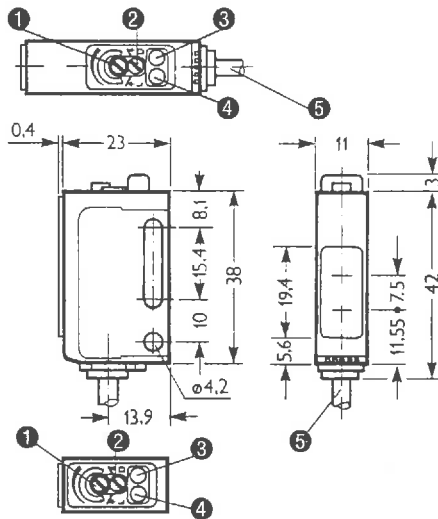


Dimensions in mm

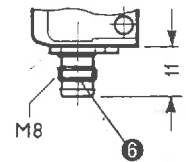


WL 160
-P/N 430

WL 160
-P/N 142



WL 160
-P/N 440

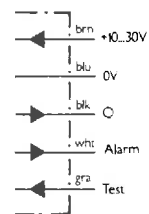


- 1 Sensitivity control (two-turn)
- 2 Light/Dark-switching selector (L: light-switching, D: dark-switching)
- 3 Red reception indicator (sender active)
- 4 Green reception indicator (WE)
- 5 Connecting cable
- 6 Equipment plug

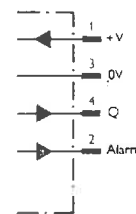
Mounting bracket and reflector included, see accessories (page 238, 236).
Order cable receptacles separately, see accessories (page 242).

Connection Diagram

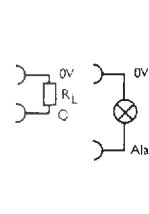
WL 160
-P/N 132
-P/N 142



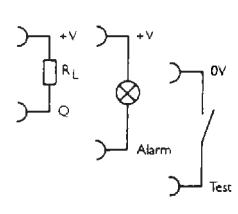
-P/N 430
-P/N 440



PNP



NPN



brn	blu	blk	wht	gra
brown	blue	black	white	gray

WL 160 Photoelectric Reflex Switch

WL 160	-N	-P
Part No.	see selection table	
Typical max. operating range / with reflector	0 to 2.5 m / P 250	
Suggested operating range with P 250 reflector	0 to 2 m	
Light spot diameter	approx. 150 mm at a distance of 2 m	
Supply voltage V_S¹⁾	10 to 30 V DC	
Current consumption (without load)	≤ 30 mA	
Ripple max. ²⁾	$\leq 5 V_{PP}$	
Light sender	LED, visible red light	
Light receiver switching mode	Light/Dark-switching, selectable	
Switching outputs open collector	NPN	PNP
Signal voltage HIGH	approx. V_S	$V_S - (\leq 1.5 V)$
Signal voltage LOW	$\leq 1.5 V$	approx. 0 V
Output current max.	≤ 100 mA	
Alarm output (statical)	≤ 100 mA	
Test input ³⁾	Sender deactivated 0 V	Sender deactivated + V_S
Response time, switching frequency max.	≤ 0.9 ms; 550/s	
VDE protection class⁴⁾	□	
Enclosure rating	IP 67	
Circuit protections ⁵⁾	A, B, C, D	
Ambient operating temperature	- 25 to + 55 °C	
Storage temperature	- 40 to + 70 °C	
Connecting cable	2 m, 5 x 0.2 mm ² , ϕ 4.2 mm	
Equipment plug	M 8, 4-pin	
Weight	20 g with plug, 60 g with connecting cable	

- 1) Limit values
 2) Must be within V_S tolerances
 3) No test input on models with plug
 4) Withstand voltage 50 V

- 5) A = Supply connections reverse-polarity protected
 B = In/Outputs reverse-polarity protected
 C = Interference pulse suppression
 D = Outputs overload and short-circuit protected

Selection Table



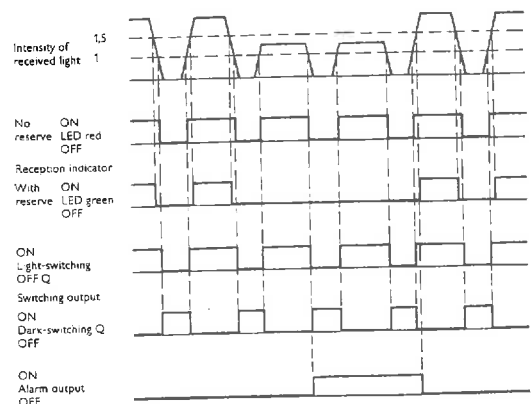
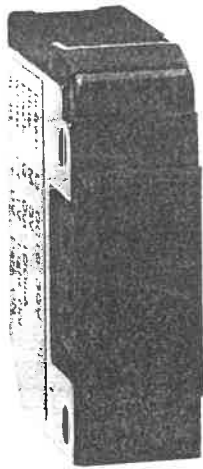
		Switching output	
		NPN	PNP
Horizontal design 	with connecting cable		
	Model	WL 160-N 132	WL 160-P 132
	Part No.	6008 807	6008 813
	with equipment plug, 4-pin		
Model	WL 160-N 430	WL 160-P 430	
Part No.	6008 809	6008 815	
Vertical design 	with connecting cable		
	Model	WL 160-N 142	WL 160-P 142
	Part No.	6008 808	6008 814
	with equipment plug, 4-pin		
Model	WL 160-N 440	WL 160-P 440	
Part No.	6008 810	6008 816	

Diagram of Functions





Scanning Range

0.01 to 2.5 m



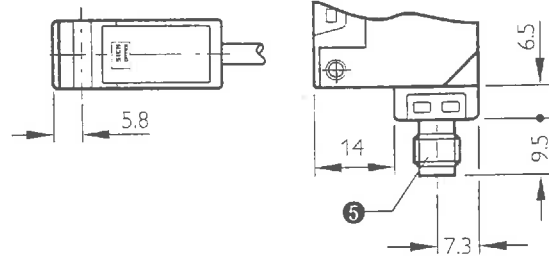
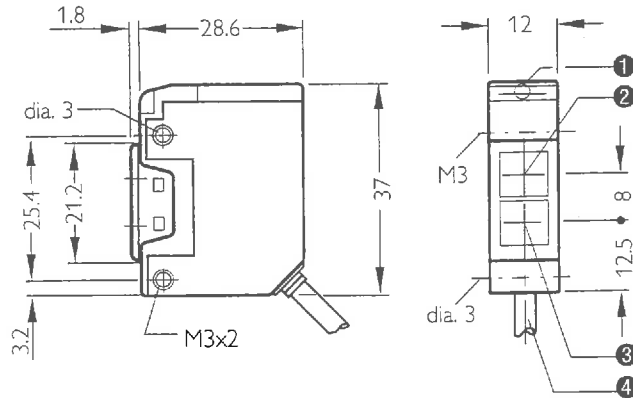
Features:

- Polarisation filter ensures that objects with reflective surfaces are also detected
- Robust stainless steel/plastic housing
- Switching output protected against short-circuits and overload
- Red-light transmitter LED
- Non-interchangeable
- Easy to adjust (with red receiver display)
- Light/dark setting – switchover via control line
- Connecting cable or connector socket M8, 4-pole
- CE /

WL 170

WL 170-P/N 132

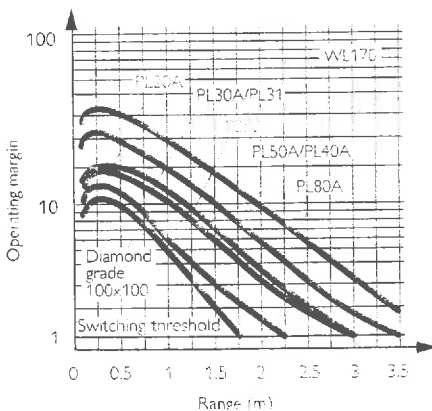
Dimensions in mm



WL 170-P/N 430

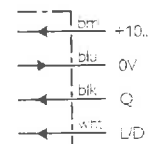
- ① Red reception display
- ② Reception axis
- ③ Transmission axis
- ④ Connecting cable
- ⑤ Connector socket M8, 4-pole

See accessories for line sockets and cable couplers (page 242).
Retaining bracket in scope of supply, see accessories (page 238).

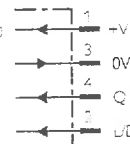


Connection diagram

WL 170 -P/N 132



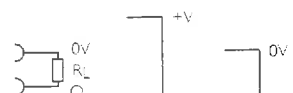
WL 170 -P/N 430



N



P



L/D = control line

light switching dark switching

brn	blu	blk	wht
brown	blue	black	white

WL 170

Photoelectric reflex switch with polarisation filter

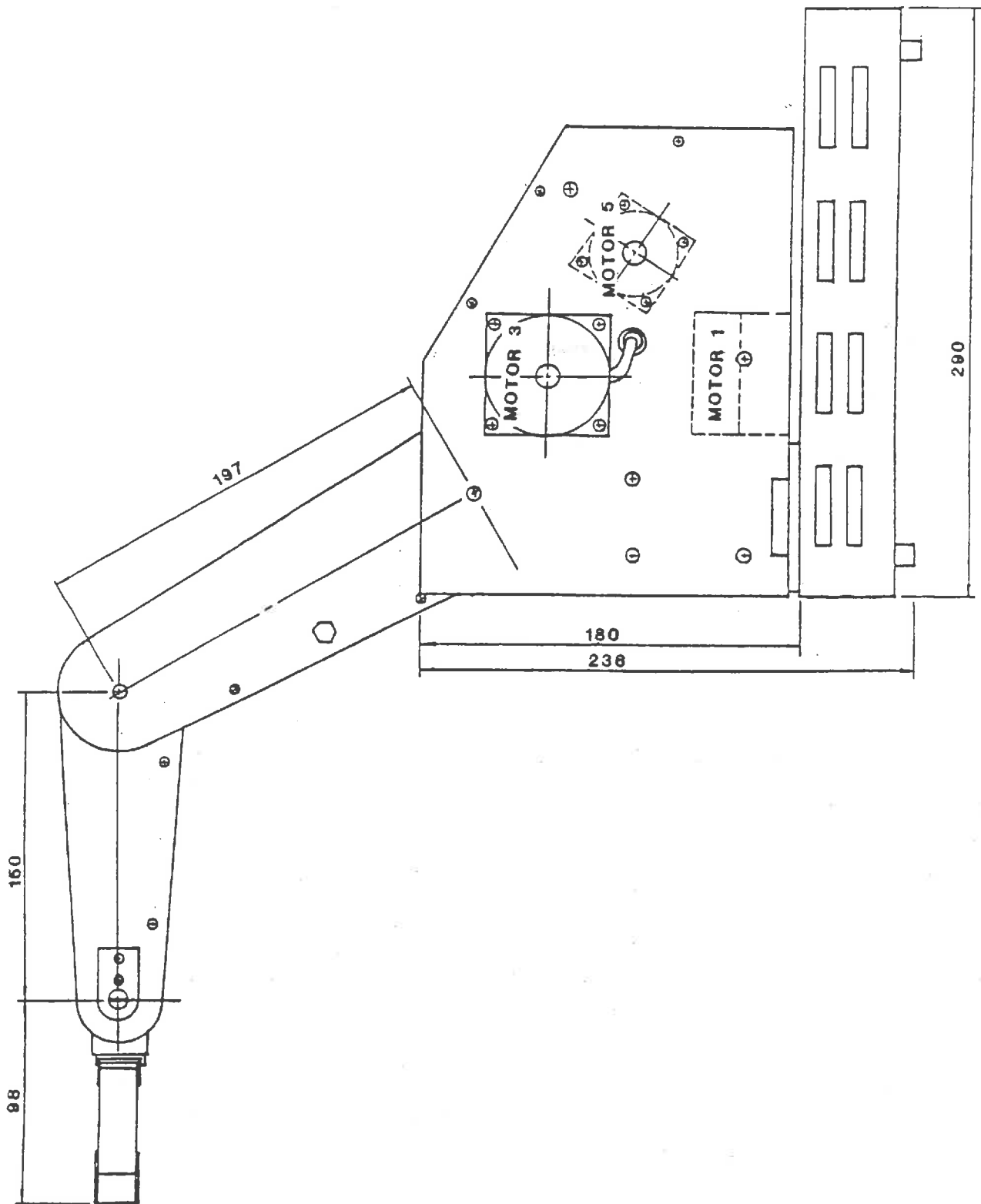
Type	WL 170	-N	-P
Part-No.	see table		
Typical max. operating range / with reflector	0.01 to 3.1 m / P 250		
Suggested operating range (RW) with reflector P 250	0.01 to 2.5 m		
Light spot diameter	approx. 200 mm at distance of 2.5 m		
Light transmitter	LED, visible red light		
Supply voltage V_S¹⁾	10 to 30 V DC		
Current consumption (no load)	≤ 30 mA		
Max. residual ripple ²⁾	$\pm 10\%$		
Light receiver switching type ³⁾	Light/dark switchover via control lead		
Polarisation filter	Yes		
Switching outputs, open collector	NPN	PNP	
Signal voltage HIGH	approx. V_S	$V_S - (\leq 1.5 \text{ V})$	
Signal voltage LOW	$\leq 1.5 \text{ V}$	approx. 0 V	
Max. output current	≤ 100 mA		
Response time; switching frequency	0.7 ms max.; 700/s min.		
VDE Protection class	III		
Type of protection (IEC 144)	IP 67		
Protection circuits ⁴⁾	A, B, C, D		
Ambient operating temperature T_U	- 25 to + 55 °C		
Storage temperature T_L	- 40 to + 70 °C		
Connecting lead	2 m, 4 x 0.2 mm ² , ϕ 4.0 mm		
Connector socket	M8-4 pole		
Weight	25 g with connector socket, 66 g with 2 m lead		

- 1) Limit values
 2) Must not exceed or fall below V_S tolerances
 3) Control line open
 NPN: light / PNP: dark switching
 Control line +V: light /
 Control line 0V: dark switching
 4) A = V_S connections non-interchangeable
 B = inputs/outputs non-interchangeable
 C = Spurious pulse suppression
 D = Outputs short-circuit and overcurrent-protected

Selection table

	Switching output	
	NPN	PNP
with connecting cable 2 m		
Type	WL 170-N 132	WL 170-P 132
Part-No.	6 010 191	6 010 189
with connector socket M8-4 pole		
Type	WL 170-N 430	WL 170-P 430
Part-No.	6 010 192	6 010 190

FIG 1.2 ROBOT - VISTA LATERAL DERECHA



Capítulo 2 - ESTRUCTURA MECÁNICA

El robot tiene cinco grados de libertad y es del tipo de acoplamiento vertical múltiple.

La precisión de colocación es de ± 0.9 mm.

La capacidad de carga es de 500 gramos (MAX) mientras que el peso total del ROBOT es de 8 Kg.

Las pinzas prevén una apertura máxima de 45 mm, mientras que la velocidad máxima puede ser de 300 mm/segundo.

Se ha concebido la distribución de las masas para mantener una buena estabilidad, incluso en situaciones de carga más gravosas, es decir cuando las pinzas se encuentran en una posición muy trasladada con respecto al eje principal del ROBOT. De hecho, la base y el soporte son particularmente robustos y pesados para contrapesar las masas en estos casos.

BASE

Tipo di movimiento: rotación en el plano horizontal. Intervalo de desplazamiento: 240 grados (120 grados en sentido dextrorso y 120 en sentido sinistrorso). Número de pasos realizados: 2000.

Ángulo de funcionamiento por paso: 0.12 grados.

ESPALDA

Tipo de movimiento: rotación en el plano vertical. Intervalo de desplazamiento: 144 grados. (72 grados hacia arriba y 72 grados hacia abajo). Número de pasos realizados: 1200.

Ángulo de funcionamiento por paso: 0.12 grados.

ANTEBRAZO

Tipo de movimiento: rotación en el plano vertical. Intervalo de desplazamiento: 100 grados. Número de pasos realizados: 1000 pasos (50 grados hacia arriba y 50 grados hacia abajo).

Ángulo de funcionamiento: 0.1 grados.

ELEVACIÓN DE LA MUÑECA

Tipo de movimiento: rotación en el plano vertical. Intervalo de desplazamiento: +90 grados -90 grados. Número de pasos realizados: 1800 pasos.

Ángulo de funcionamiento por paso: 0.1 grados.

ROTACIÓN DE LA MUÑECA

Tipo de movimiento: rotación alrededor del eje principal de las pinzas.

Intervalo de desplazamiento: +180 grados -180 grados. Número de pasos realizados: 1800 pasos en sentido dexterorso y 1800 pasos en sentido opuesto.

Ángulo de funcionamiento por paso: 0.1 grados

PINZAS

Tipo de movimiento: rotación alrededor del eje principal de las pinzas.

El peso máximo que se puede manipular es de .5 Kg, incluso el peso de las pinzas.

El motor que se utiliza es de avance gradual, así como también para el movimiento de las articulaciones. Si las pinzas están cerradas, se puede efectuar un movimiento de apertura máxima de 1800 pasos, lo que equivale a una apertura de 0.025 mm por paso. En cambio, si las pinzas están abiertas, se puede efectuar un movimiento análogo de cierre de 1800 pasos.

Capítulo 3. MANDOS DE DESPLAZAMIENTO

Todas las operaciones del robot se controlan mediante unos mandos particulares que el ORDENADOR INDUSTRIAL IC 88/EV envía a la unidad de control del robot a través de la línea paralela.

El programa de gestión realizado con el intérprete QUICK BASIC, utiliza la Puerta paralela como Puerta de salida para los MANDOS, por consiguiente éstos se consideran como mandos de SALIDA hacia una impresora de tipo paralelo.

La sintaxis general será del tipo:

LPRINT "Mando"

A continuación, se indica la lista de los mandos para el control directo del ROBOT:

ZERO (Posición de CERO - HOME position)

Mediante este mando (LPRINT "Z"), el Robot toma como posición inicial absoluta de referencia, la posición actual.

Ante todo, hay que destacar que los motores de avance gradual de este ROBOT están desprovistos de control automático de posición y, por consiguiente, de toda referencia para su posición inicial absoluta. Con objeto de tener un punto de arranque de referencia para toda la serie de movimientos sucesivos, habrá que recurrir al punto físico en que las posiciones de memoria donde se memorizan los pasos realizados por todos los motores son nulas.

Es conveniente que el ROBOT cumpla este mando cuando el mismo se encuentra en la posición de cero mecánico. Sin embargo, si el robot se coloca en una posición cualquiera y si en esta posición se le da el mando HOME, el robot tomará esta posición como punto cero para todos los movimientos sucesivos.

Si no se introduce esta instrucción, se toma como posición de cero el punto en que se encuentra el ROBOT en el momento del encendido.

MOVE

Con esta instrucción (LPRINT Ma1,a2,a3,a4,a5,a6) cada uno de los 6 ejes del ROBOT toma la posición especificada por los parámetros: a1, a2, a3, a4, a5 y a6.

El signo "+" puede desatenderse.

Los intervalos de los valores de los parámetros, eje por eje, son los siguientes:

parámetro	eje	+	-
a1	BASE	DEXTRORSO	SINISTRORSO
a2	ESPALDA	ARRIBA	ABAJO
a3	BRAZO	ARRIBA	ABAJO
a6	PINZAS	CERRADAS	ABIERTAS

Los motores a4 y a5 determinan juntos el movimiento de la muñeca.

Si los mismos se accionan con un número igual de pasos de signo opuesto, dichos motores hacen subir o bajar la muñeca. En cambio, si se accionan con un número igual de pasos del mismo signo, los motores provocan una rotación de la muñeca en sentido dextrorso o sinistrorso.

Los intervalos de desplazamiento de todos los ejes son los siguientes:

EJE	límite MÍNIMO	límite MÁXIMO
BASE	-1000	+1000
ESPALDA	-600	+600
BRAZO	-500	+500
MUÑECA ARRIBA/ABAJO	-900	+900
ROTACIÓN MUÑECA	-1800	+1800
PINZAS CERRADAS	0	+1800
PINZAS ABIERTAS	-1800	0

HERE

Mediante esta instrucción (LPRINT"Ha0" con a0=1...100), en la memoria RAM del sistema de control se memoriza la posición en la que se encuentra el ROBOT y, por consiguiente, el valor de los 6 parámetros relativos a los ejes.

Pueden memorizarse como máximo 100 posiciones diferentes. Si se añade un segundo chip de RAM en la tarjeta de control del robot (HM6116), ese número puede elevarse a 270.

POSITION

Con este mando (LPRINT"Pa0,a1,a2,a3,a4,a5,a6), se memoriza con un índice "a0" la posición especificada por los parámetros a1...a6.

Todas las posiciones se refieren al punto CERO fijado con este mando. Las posiciones pueden ser a lo sumo 100 ó 270 en el caso de que se extienda la memoria RAM, como vimos en el mando H.

La diferencia con el mando precedente H consiste en el hecho de que no es necesario desplazar físicamente el ROBOT a la posición deseada para memorizarla, sino que en este caso es suficiente especificar el número de la posición y los parámetros relacionados con los relativos ejes. Los límites de los parámetros a1...a6 son los mismos que vimos en el mando MOVE.

GO TO

Mediante esta instrucción (LPRINT "Ga0") el ROBOT se desplaza a la posición a0 correspondiente, previamente especificada con un mando Ha0 o Pa0.

El valor a0 varía de 0 a 100 en condiciones normales y puede alcanzar 269 con memoria RAM añadida.

El mando LPRINT "G0" determina la colocación del ROBOT en la posición CERO fijada con el mando ZERO.

NEST

Con esta instrucción (LPRINT "N") el robot se desplaza a su posición mecánica de arranque, previamente fijada con el mando ZERO o tomada por el ROBOT en el momento del encendido.

GRIP CLOSE

Este mando (LPRINT"C") determina el cierre de las pinzas que alcanzan su límite mínimo de apertura (-1800 pasos).

GRIP OPEN

Con esta instrucción (LPRINT "O") las pinzas se abren y alcanzan su límite máximo de apertura (+1800 pasos).

El ROBOT ignora el mando si se transmite el mismo más veces seguidamente.

SPEED

Con este mando (LPRINT "Sa0") se selecciona la velocidad de desplazamiento de los ejes. Para cargas considerables de las pinzas este valor puede ser 1 ó 2, mientras que para cargas leves la velocidad puede aumentarse hasta 5. El valor de a0 varía entre 1 y 5.

La velocidad fijada con este mando se mantiene estable hasta que se introduzca nuevo mando de velocidad.

DELAY

Tras recibir este mando (LPRINT "Da0"), el ROBOT queda a la espera de más instrucciones durante un tiempo correspondiente al número de segundos indicados mediante a0. La precisión es de +1/100 segundos.

LIMIT

Este mando (LPRINT "L0" o bien LPRINT "L1") permite controlar los límites de desplazamiento de los ejes. Con el mando "L1" se controlan los límites de movimiento con relación a la posición CERO. Si se rebasan dicho límites, el ROBOT se bloquea en estos puntos y empieza a centellar el LED de ERROR en la base fija del ROBOT.

Sucesivamente, el ROBOT interpreta como única instrucción ejecutable el mando "N" de vuelta a la posición CERO.

Con el mando "L0" el control de los límites no se efectúa y esta es la condición tras el encendido.

OUT 4

Mediante esta instrucción es posible controlar la SALIDA de una Puerta de 4 bitios. Se puede acceder a las líneas relativas mediante un conector presente en la tarjeta de control del ROBOT. Para acceder a dicho conector, es necesario quitar la protección metálica de la base fija del ROBOT. El conector (CON 4 en el diagrama serigrafiado del C.I) presenta la siguiente disposición de líneas:

Capítulo 4 - PROGRAMA DE GESTIÓN

El Robot se suministra junto a un programa que permite su utilización inmediata. Véanse las instrucciones indicadas en la etiqueta del disco para la instalación y la puesta en marcha del programa.

El menú principal del programa prevé las funciones básicas siguientes:

- 1) MANDO DE UN EJE INDIVIDUAL
- 2) MANDO GLOBAL DE LOS EJES
- 3) DESPLAZAMIENTO CONTROLADO DESDE EL TECLADO
- 4) MEMORIZACIÓN DE UNA SECUENCIA
- 5) EDICIÓN DE UNA SECUENCIA
- 6) ANULACIÓN DE LOS MOVIMIENTOS LLEVADOS A CABO
- 7) REALIZACIÓN DE UNA SECUENCIA
- 8) REALIZACIÓN DE UNA SECUENCIA POR MEDIO DE LA ROM
- 9) LISTA DE LAS SECUENCIAS
- F) SALIDA

A nivel de operaciones, dichas funciones están descritas a continuación.

1 MANDO DE UN EJE INDIVIDUAL

Esta sección permite accionar cada eje del robot individualmente.

Se selecciona la velocidad y luego se eligen los parámetros de desplazamiento relativos a un eje en particular.

Cada vez, se visualizan los límites dentro de los cuales pueden seleccionarse los parámetros.

El programa bloquea automáticamente la elección de valores fuera de los intervalos permitidos, proponiendo otra elección.

Si se selecciona el desplazamiento de las pinzas, que corresponde a la elección del eje N.6, se pueden escoger valores tales que permiten la apertura de las pinzas, puesto que ésta última siempre se encuentra en posición cerrada a la puesta en marcha.

Se sale de esta sección mediante la tecla <ESC>.

Con la tecla <Home> se pueden llamar en cualquier momento unas instrucciones sobre el uso de la sección de programa que se está utilizando (HELP).

2 MANDO GLOBAL DE LOS EJES

Seleccionando esta opción, se pueden desplazar simultáneamente todos los ejes y, por consiguiente, se pueden mover completamente todas las articulaciones del robot.

La definición de los parámetros relativos a cada eje y a las pinzas se realiza como en la sección precedente, impidiendo los mandos erróneos.

Apretando la tecla <ESC> se sale del menú principal; el movimiento se lleva a cabo automáticamente tras la introducción del valor de desplazamiento relativo al sexto eje.

Si no se quiere mover ningún eje, es suficiente apretar sólo <Enter> cuando se introduce el parámetro de desplazamiento relativo al eje mismo.

En caso de que se lleven a cabo más desplazamientos globales, el número de pasos efectuados por cada eje se acumula y por consiguiente el programa sigue controlando los límites de desplazamiento.

Con la tecla <Home> se pueden llamar en cualquier momento unas instrucciones sobre el uso de la sección de programa que se está utilizando (HELP).

3 DESPLAZAMIENTO CONTROLADO DESDE EL TECLADO

Mediante esta sección se consigue desplazar al robot desde el teclado. A cada tecla de función le corresponde un eje determinado y, si se aprietan otras teclas particulares, es posible variar la velocidad de desplazamiento y los parámetros de las pinzas.

La correspondencia entre las teclas de función y los ejes es la siguiente:

- F1 :rotación BASE en sentido sinistrorso
- F2 :rotación BASE en sentido dextrorso
- F3 :rotación ESPALDA hacia abajo
- F4 :rotación ESPALDA hacia arriba
- F5 :rotación BRAZO hacia abajo
- F6 :rotación BRAZO hacia arriba
- F7 :rotación MUÑECA hacia abajo
- F8 :rotación MUÑECA hacia arriba
- F9 :rotación MUÑECA en sentido sinistrorso
- F10 :rotación MUÑECA en sentido dextrorso

TECLAS DE DIRECCIÓN DEL TECLADO NUMÉRICO:

DERECHA : apertura PINZAS
IZQUIERDA : cierre PINZAS
ARRIBA : incremento del paso de desplazamiento
ABAJO : disminución del paso de desplazamiento

No hay que activar la tecla CAPS LOCK.

Con la tecla <End> es posible abrir totalmente las PINZAS, sin posiciones intermedias. Si se vuelve a oprimir dicha tecla, es posible cerrar completamente las PINZAS sin posiciones intermedias, y así sucesivamente.

A través de la tecla <PgUp> es posible acrecer la velocidad de accionamiento de todos los ejes, de 1 a 5. Al comienzo este valor es 3. Cada vez que se oprime la tecla, dicho valor aumenta de una unidad. Al rebasar el límite de 5, el mismo vuelve al valor 1, y así sucesivamente.

Si se mantiene apretada la tecla <+>, las posiciones en las que se encuentra el robot se memorizan en la memoria RAM de la unidad de control (se utiliza el mando POSITION) y quedan disponibles para el control sucesivo de las posiciones. De hecho, si se ha apretado la tecla <+> por lo menos en una posición, a la salida de la fase de DESPLAZAMIENTO CONTROLADO DESDE EL TECLADO, en lugar del menú principal aparece un sub-menú mediante el cual se puede construir una secuencia con las posiciones previamente memorizadas y luego llevarla a cabo.

El número máximo de posiciones es 40.

Esta fase puede resultar útil para analizar secuencias de movimientos repetidos (lazos de movimientos) antes de construir una secuencia de operaciones propiamente dicha.

4. MEMORIZACIÓN DE UNA SECUENCIA

Una vez que se haya seleccionado esta opción, el programa pregunta el nombre bajo el cual se va a memorizar la secuencia. Dicho nombre, que consiste en una cadena de 8 caracteres, permitirá luego el acceso a las funciones de movimiento en cualquier momento.

Si previamente se ha memorizado una secuencia con el mismo nombre, el programa avisa al operador. No es posible ninguna superposición accidental, ya que es necesario entrar en la fase de EDIT para borrar una secuencia de movimientos ya existente.

Si el nombre que se ha elegido para la secuencia no coincide con ningún otro nombre existente, aparece el menú de correspondencia entre teclas y ejes, el mismo que se visualiza en la fase de DESPLAZAMIENTO CONTROLADO DESDE EL TECLADO. El programa se prepara para aceptar los pasos de la secuencia y para memorizar los mismos en el disco. En cambio, si existe ya una secuencia con el mismo nombre, aparece el mensaje: SECUENCIA YA EXISTENTE USAR EDIT PARA BORRARLA.

Después de cada paso de la secuencia hay que apretar la tecla <+> para proceder al paso sucesivo.

Durante cada paso es posible mover más ejes sucesivamente.

En este caso, durante la ejecución se llevará a cabo el desplazamiento global compuesto. Si se quiere conservar una secuencia compuesta de movimientos de un único eje, hay que asignar un paso a cada desplazamiento. Para cerrar la secuencia de desplazamiento, basta con apretar la tecla <ESC>.

Durante la construcción de la secuencia, es importante evaluar bien las operaciones que prevén el desplazamiento de las PINZAS, puesto que en los tiempos sucesivos de ejecución influye mucho el tiempo de apertura y de cierre de las PINZAS, más bien reducido con respecto a los demás ejes. Normalmente, para las operaciones de manipulación necesarias en una secuencia, es conveniente efectuar el desplazamiento de las pinzas únicamente a través de la programación de los pasos relativos al eje N.6, evitando su apertura total o su cierre completo mediante la tecla <End>, a partir de la posición de PINZAS CERRADAS.

Con la tecla <Home> se pueden llamar en cualquier momento unas instrucciones sobre el uso de la sección de programa que se está utilizando (HELP).

5. EDICIÓN DE UNA SECUENCIA

Con esta opción es posible modificar los pasos de cada secuencia y el número de éstos o bien borrar completamente la secuencia misma sin efectuar ningún desplazamiento del ROBOT. Se visualizan los valores de los números de pasos por cada eje y por cada paso de la secuencia. Por consiguiente, es posible modificar cada valor desde el teclado. A través de la tecla <F1> puede truncarse la secuencia en el número de pasos indicado por el cursor; con la tecla <F2> se salva la secuencia en el disco, con todas las modificaciones efectuadas; con <ESC> es posible salir de la fase sin llevar a cabo ninguna modificación; mientras que la tecla <F9> permite borrar definitivamente la secuencia.

6. ANULACIÓN DE LOS MOVIMIENTOS LLEVADOS A CABO

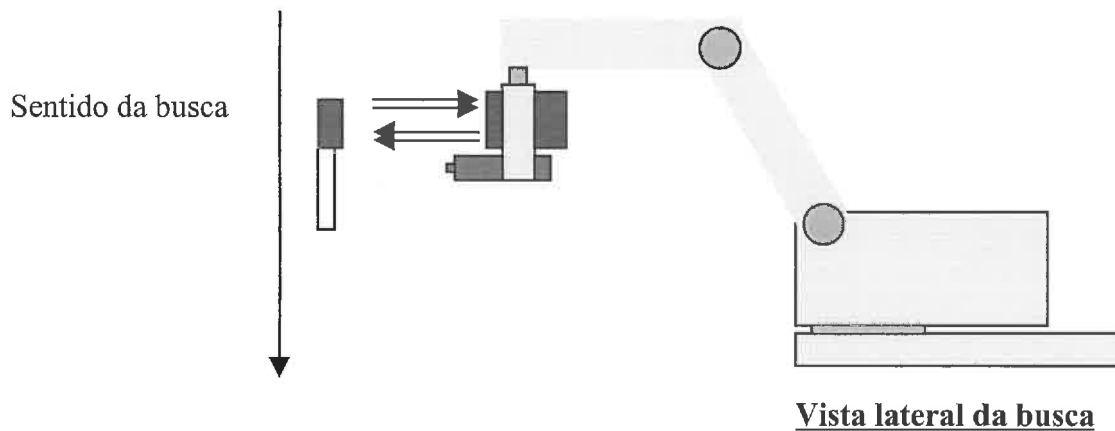
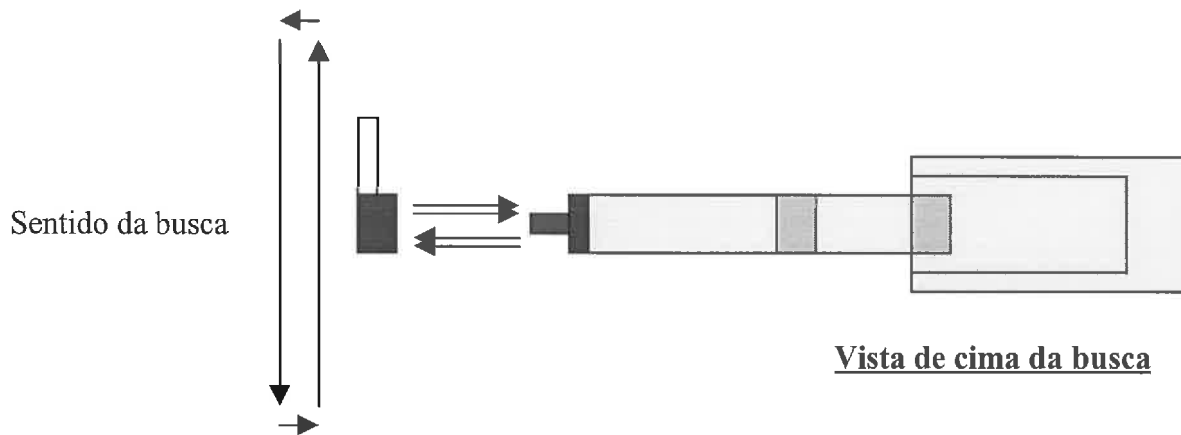
Seleccionando esta opción, el programa anula totalmente todos los movimientos llevados a cabo hasta el momento en que se imparte este mando. El robot toma la posición cero fijada durante el encendido.

7. REALIZACIÓN DE UNA SECUENCIA

Mediante esta sección, el ROBOT lleva a cabo la secuencia de movimientos anteriormente memorizada utilizando la opción 4 del menú principal. La secuencia se identifica mediante su nombre, si ésta efectivamente está memorizada.

Si el documento memorizado al que corresponde la secuencia no se encuentra, aparece el mensaje: LA SECUENCIA NO EXISTE.

REALIZAÇÃO DA BUSCA



A busca do painel, começa com uma varimento horizontal da esquerda para direita, como nos é mostrado na vista de cima. De seguida efectua uma descida de um grau como nos mostra a vista lateral, e depois começa uma varimento da direita para esquerda

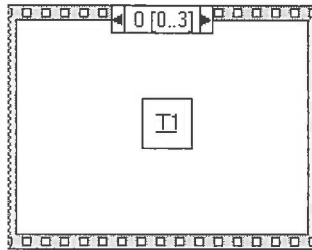
No plano horizontal, o braço realiza um percurso de 180° , com uma definição de 12° . No plano vertical, o braço realiza um percurso de 90° , com uma definição de 6° .

O braço realiza o percurso todo se ele não detectar o reflector, e o fim regressa a sua posição inicial. Se ao contrario ele detectar o reflector, o braço para no momento onde ele o encontrar. Só depois é que ele acciona a chamada do elevador.

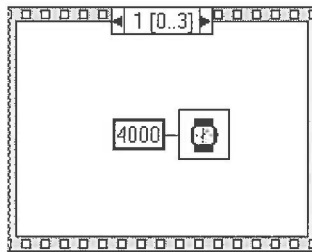
Depois de chamado o elevador, ele entra para dentro e volta a realizar este ciclo.

Diagrama de 1235

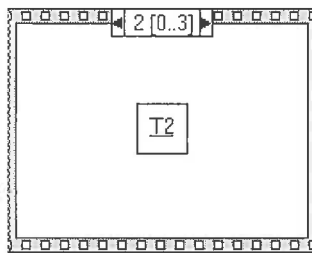
Sequência 0/3



Sequência 1/3



Sequência 2/3



Sequência 3/3

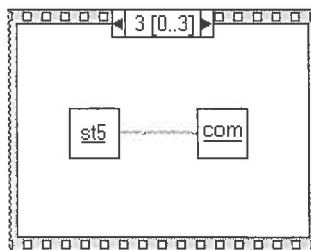
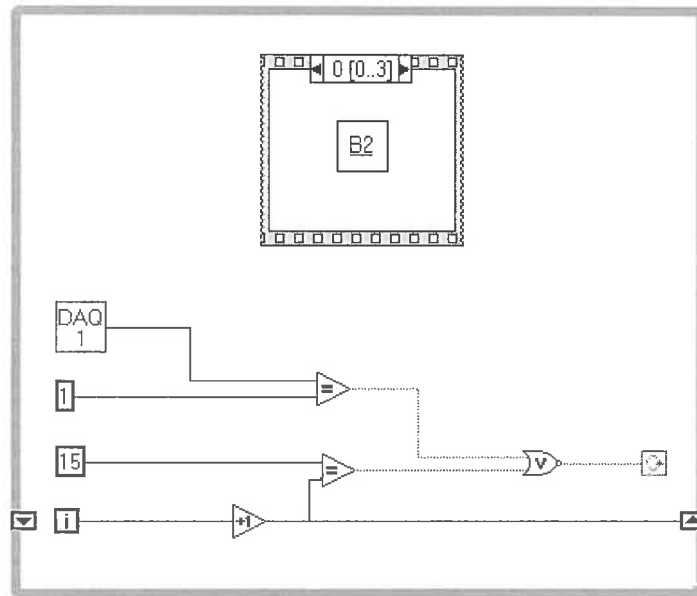
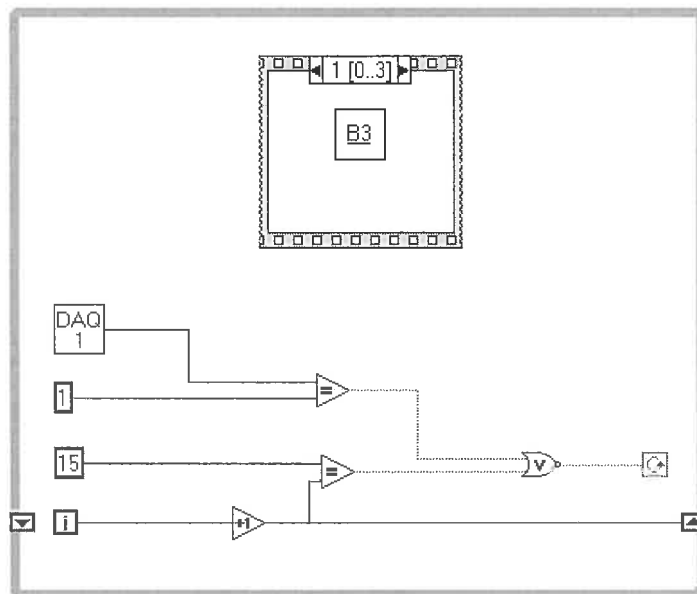


Diagrama de 2343

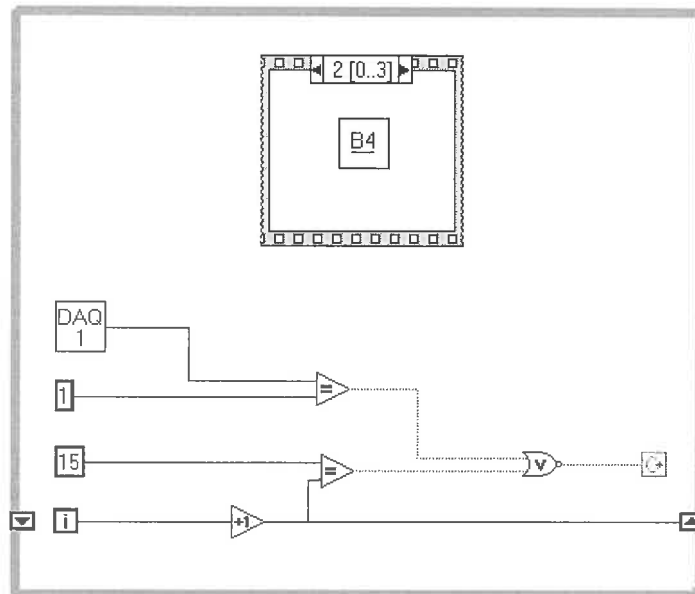
Sequência 0/3



Sequência 1/3



Sequência 2/3



Sequência 3/3

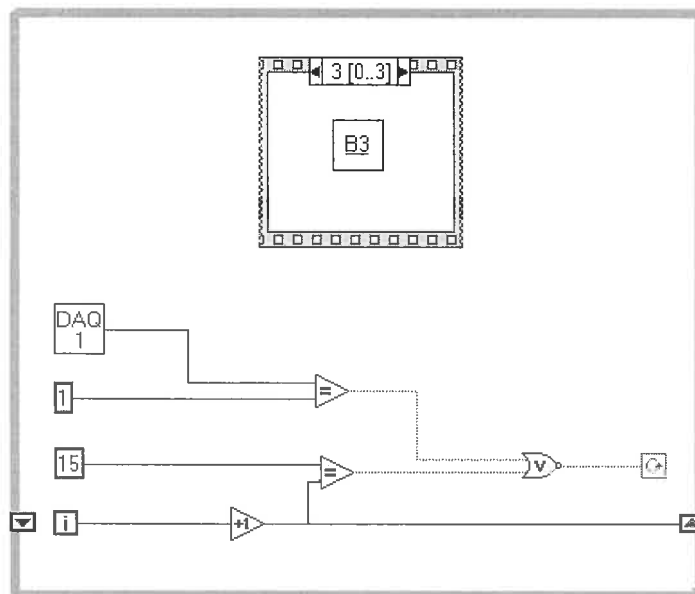


Diagrama de B1

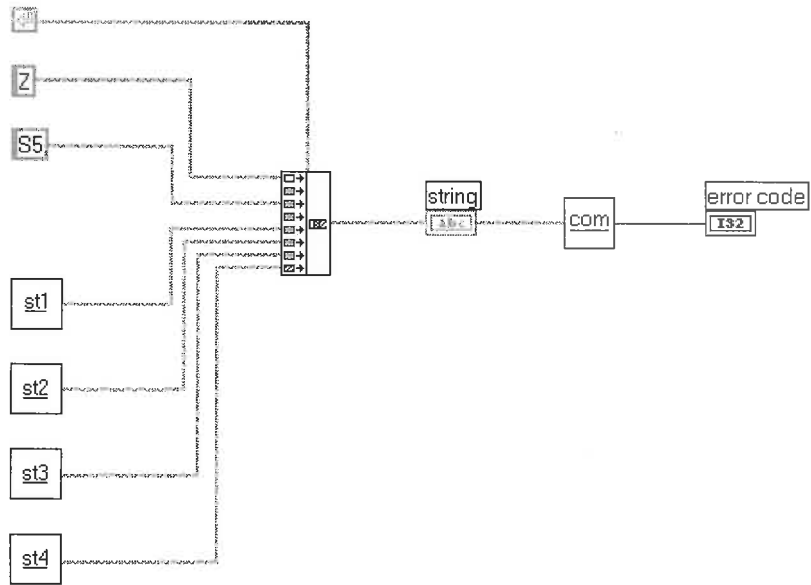


Diagrama de B2

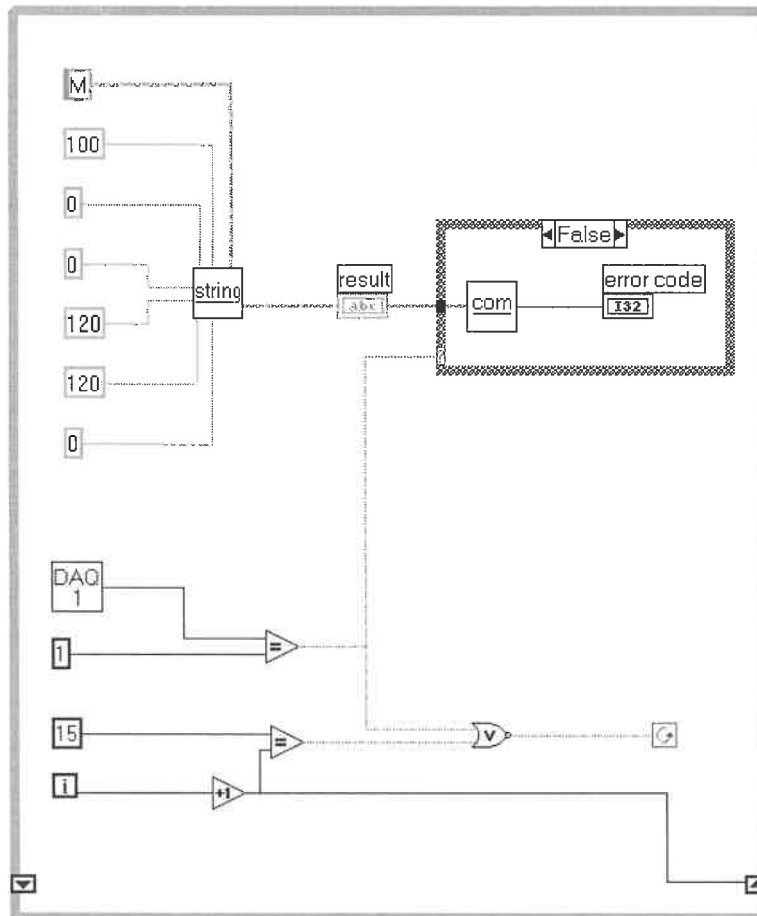


Diagrama de B3

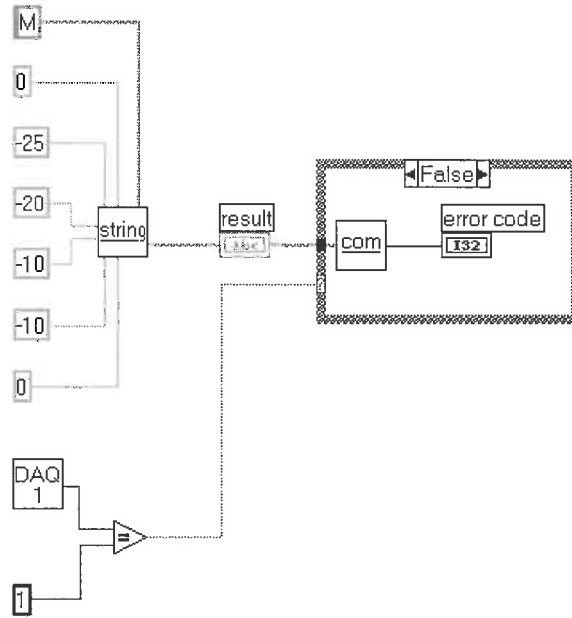


Diagrama de B4

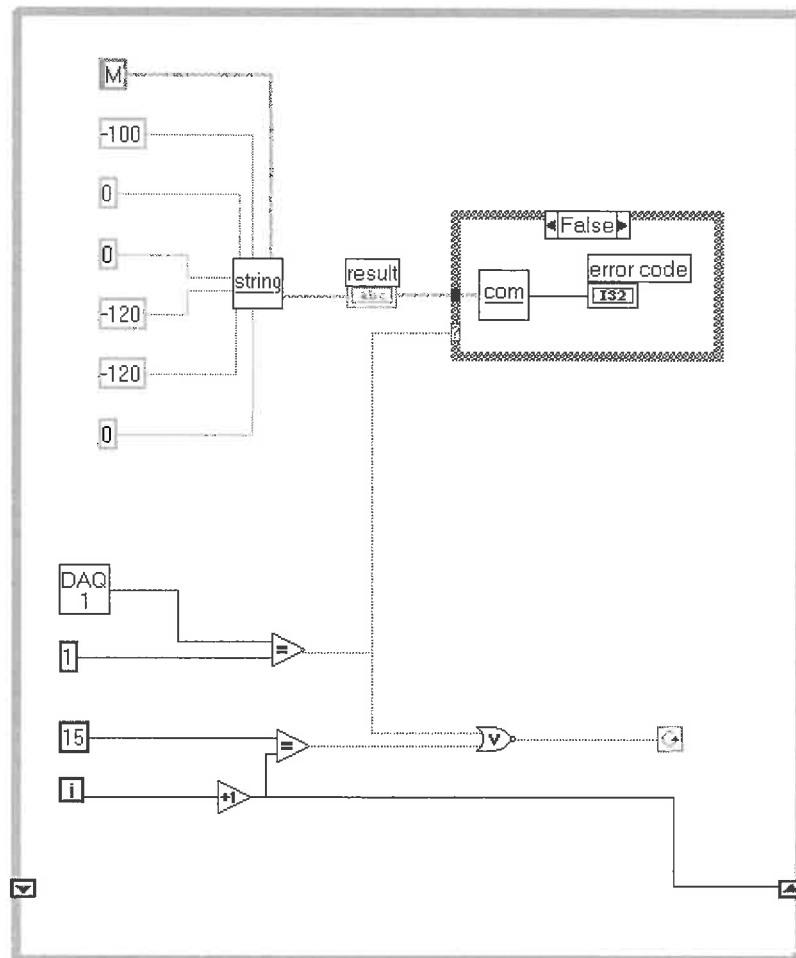


Diagrama de B5

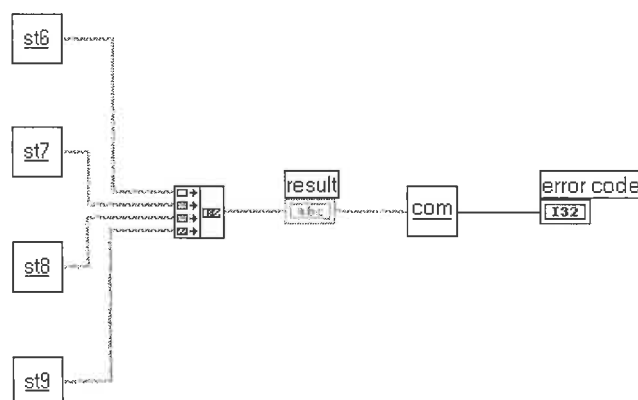
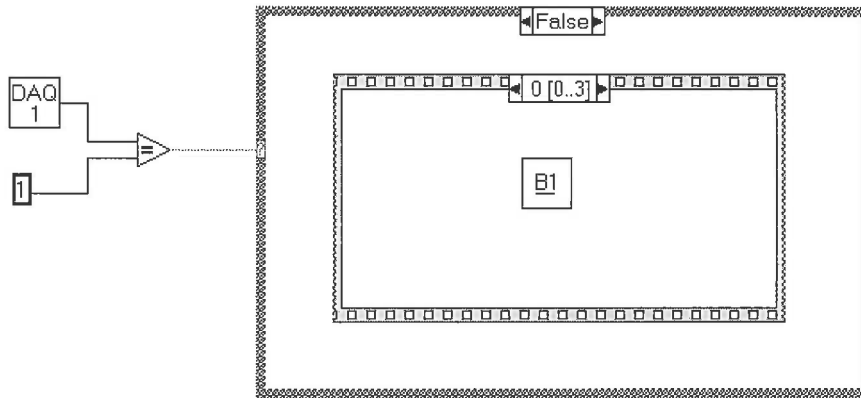
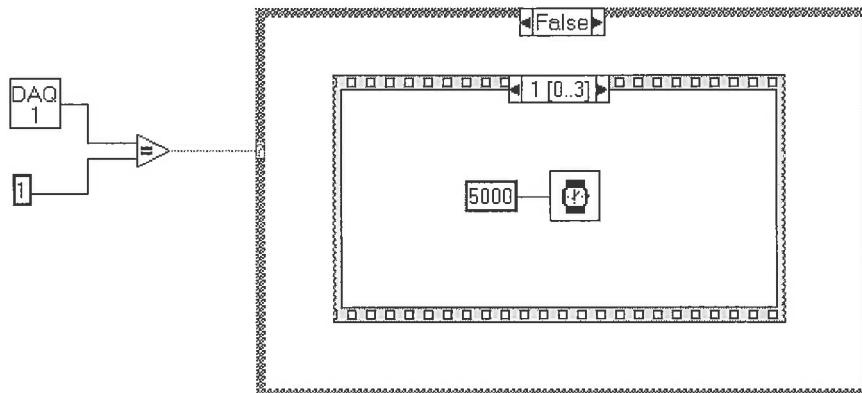


Diagrama de Busca

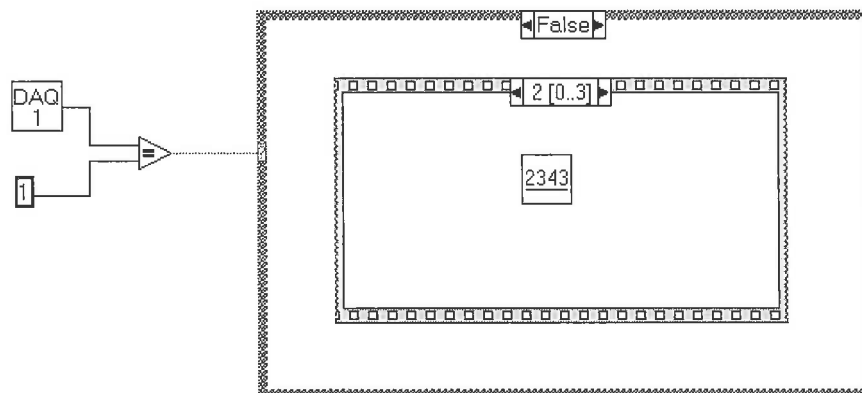
Sequência 0/3



Sequência 1/3



Sequência 2/3



Sequência True

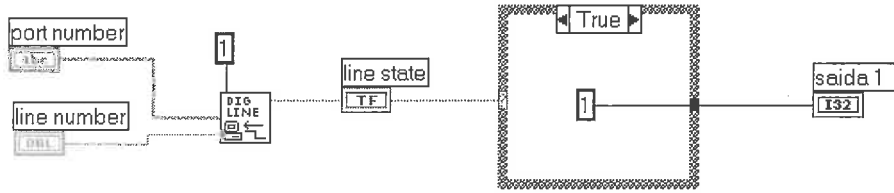


Diagrama de daq1

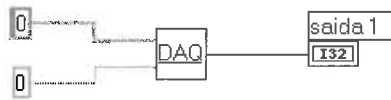


Diagrama de daq2

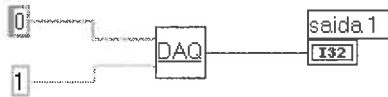


Diagrama de refl

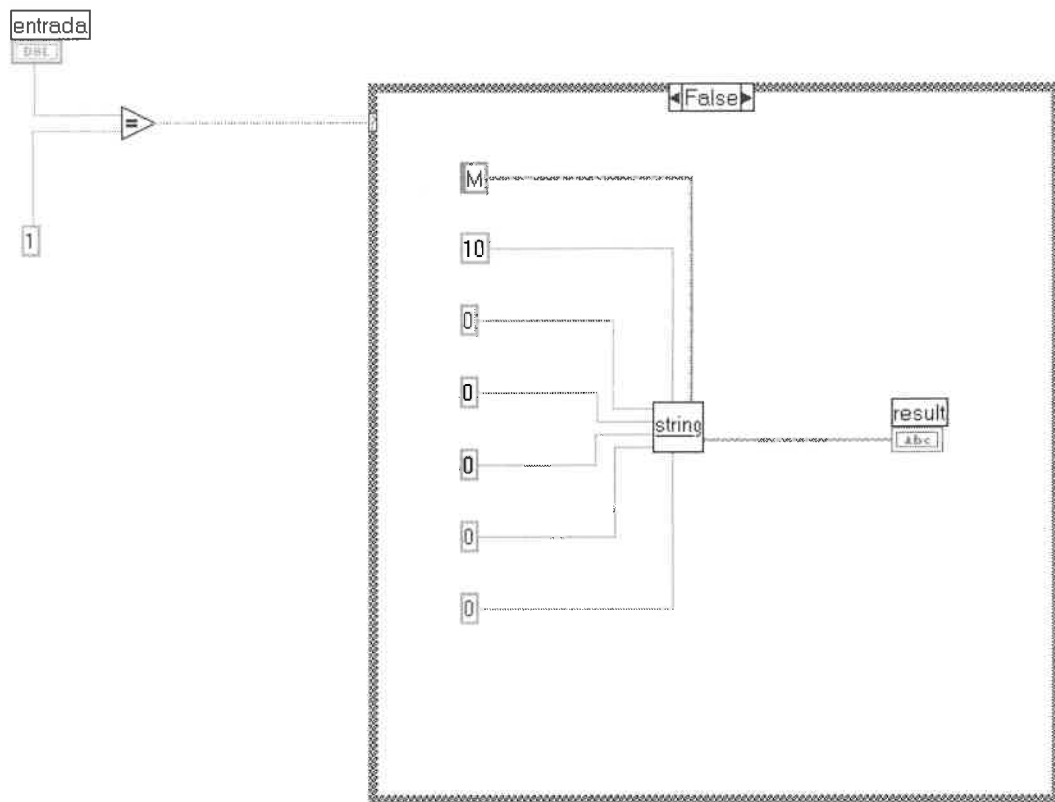


Diagrama de st1

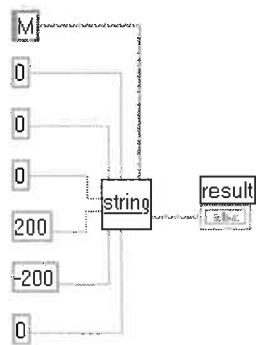


Diagrama de st2

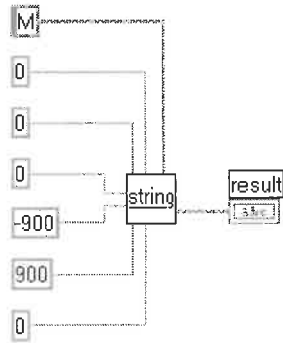


Diagrama de st3

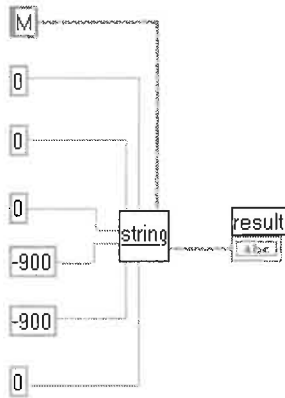


Diagrama de st4

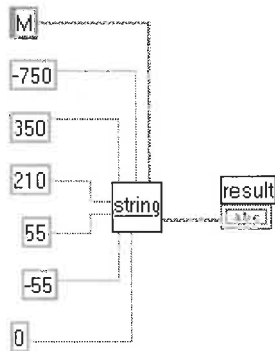


Diagrama de st5



Diagrama de st6

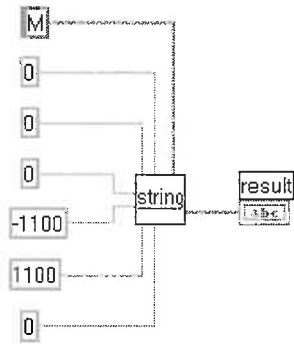


Diagrama de st7

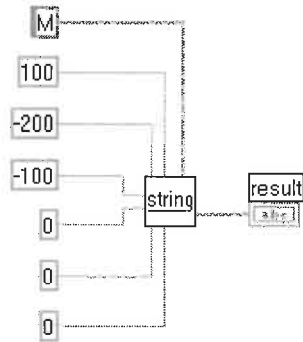


Diagrama de st8

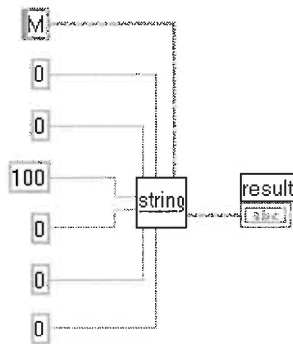


Diagrama de st9

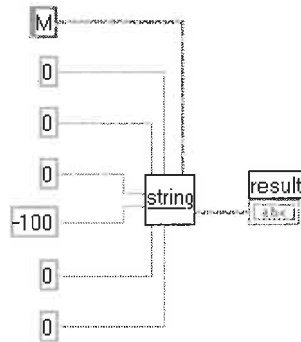


Diagrama de st10

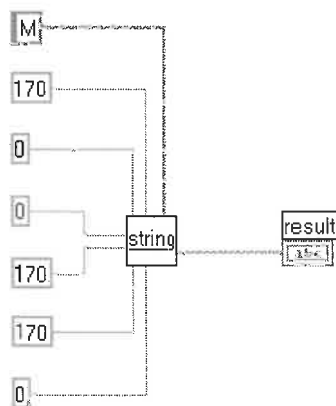


Diagrama de st11

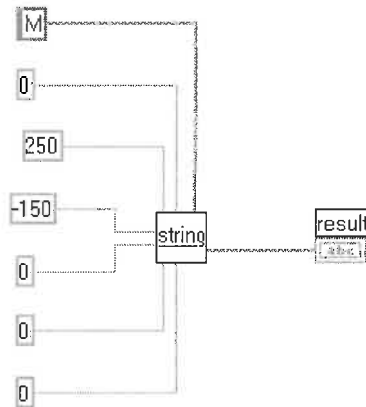


Diagrama de st12

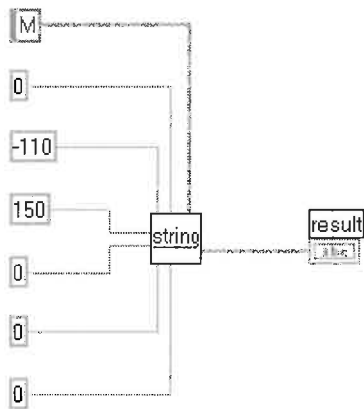


Diagrama de st13

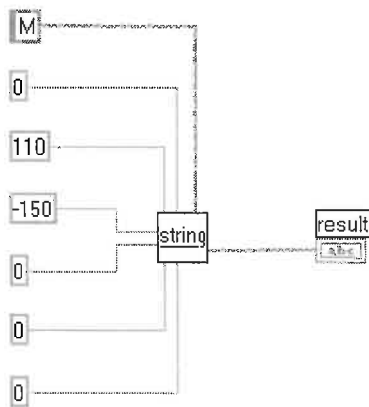


Diagrama de string

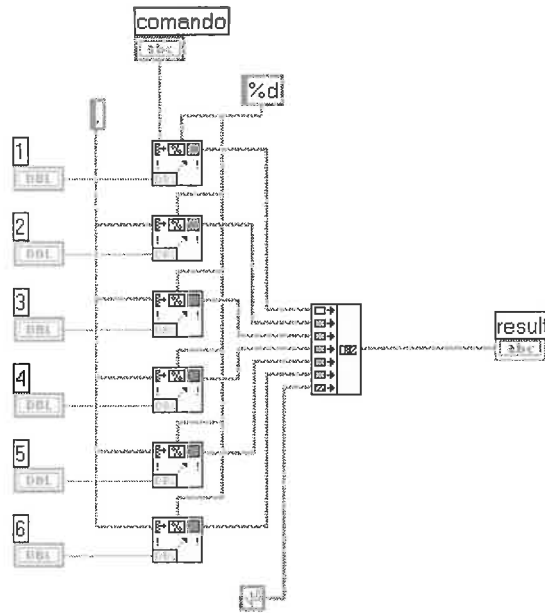
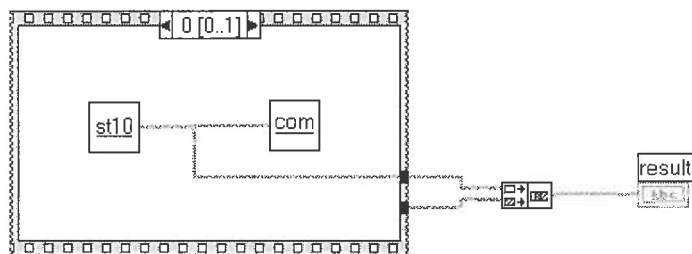


Diagrama de T1

Sequência 0/1



Sequência 1/1

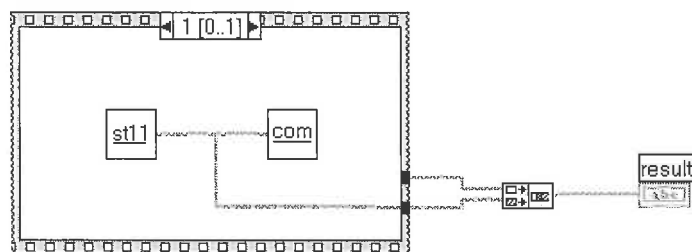
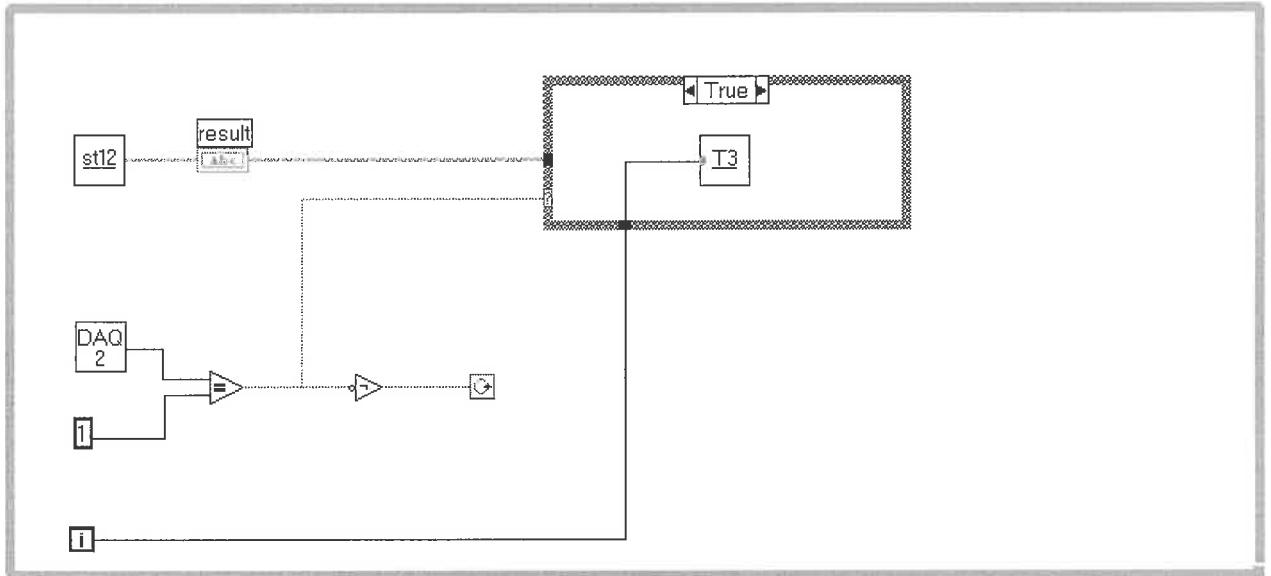


Diagrama de T2

Sequencia True



Sequencia False

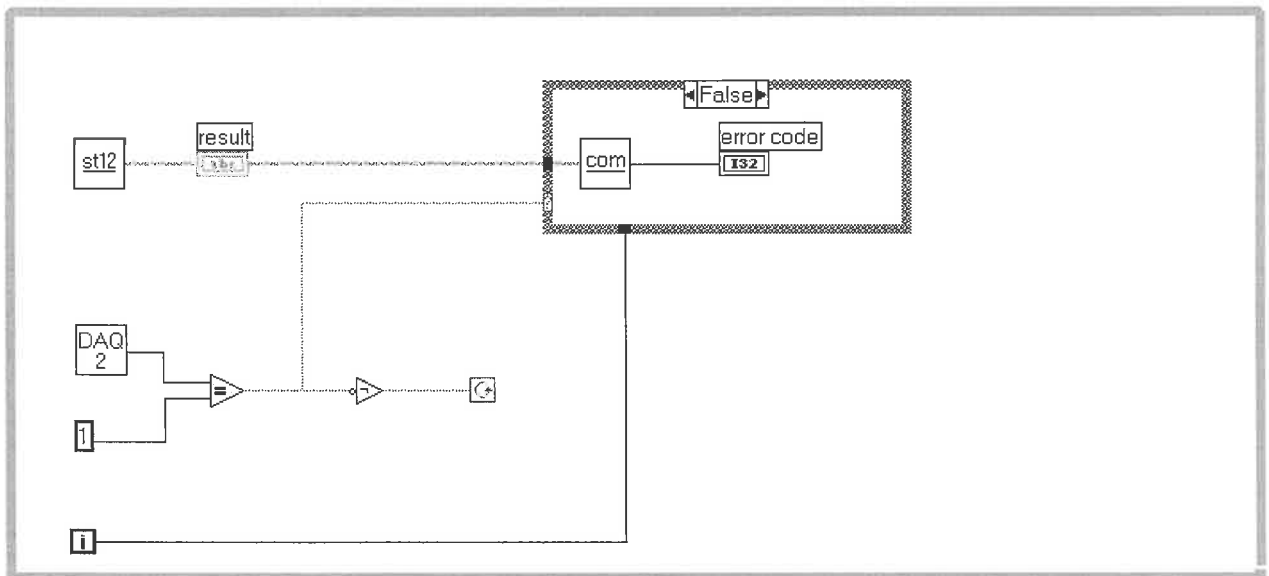


Diagrama de T3

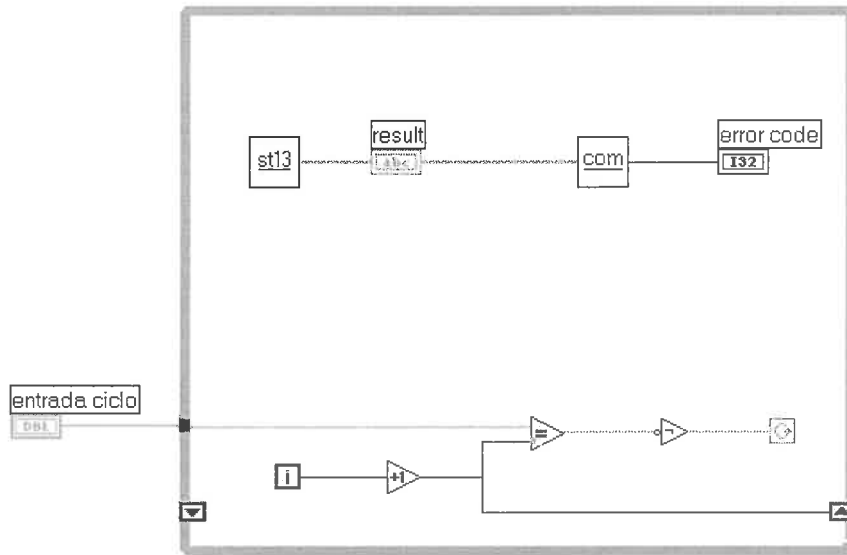
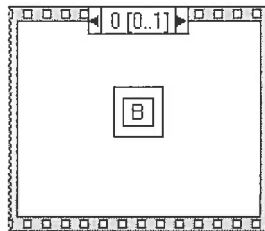
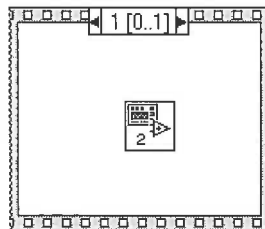


Diagrama de untitled1

Sequência 0/1



Sequência 1/1



LabVIEW Function Error Codes

Code	Description
0	No error.
1	Manager argument error.
2	Argument error.
3	Out of zone.
4	End of file.
5	File already open.
6	Generic file I/O error.
7	File not found.
8	File permission error.
9	Disk full.
10	Duplicate path.
11	Too many files open.
12	System feature not enabled.
13	Resource file not found.
14	Cannot add resource
15	Resource not found.
16	Image not found.
17	Image memory error.
18	Pen does not exist.
19	Config type invalid.
20	Config token not found.
21	Config parse error.
22	Config memory error.
23	Bad external code format.
24	Bad external code offset.
25	External code not present.
26	Null window.
27	Destroy window error.
28	Null menu.
29	Print aborted.
30	Bad print record.
31	Print driver error.
32	Windows error during printing.
33	Memory error during printing.
34	Print dialog error
35	Generic print error.
36	Invalid device refnum.
37	Device not found.
38	Device parameter error.
39	Device unit error.
40	Cannot open device.
41	Device call aborted.
42	Generic error.
43	Cancelled by user.
44	Object ID too low.
45	Object ID too high.
46	Object not in heap.

- 47 Unknown heap.
- 48 Unknown object (invalid DefProc).
- 49 Unknown object (DefProc not in table).
- 50 Message out of range.
- 51 Invalid (null) method.
- 52 Unknown message.
- 53 Manager call not supported.
- 54 Bad address.
- 55 Connection in progress.
- 56 Connection timed out.
- 57 Connection is already in progress.
- 58 Network attribute not supported.
- 59 Network error.
- 60 Address in use.
- 61 System out of memory.
- 62 Connection aborted.
- 63 Connection refused.
- 64 Not connected.
- 65 Already connected.
- 66 Connection closed.
- 67 Initialization error (interapplication manager)
- 68 Bad occurrence.
- 69 Wait on unbound occurrence handler.
- 70 Occurrence queue overflow.
- 71 Datalog type conflict.
- 72 Unused.
- 73 Unrecognized type (interapplication manager).
- 74 Memory corrupt.
- 75 Failed to make temporary DLL.
- 76 Old CIN version.
- 77 Unknown error code
- 81 Format specifier type mismatch
- 82 Unknown format specifier
- 83 Too few format specifiers
- 84 Too many format specifiers
- 85 Scan failed

Copyright © 1994-1995 National Instruments Corporation. All Rights Reserved. Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation. LabVIEW is a registered trademark of National Instruments Corporation.

Pin	Signal Name	Description
1-8	ACH0 through ACH7	Analog input Channels 0 through 7 (single-ended).
9	AISENSE/AIGND	Analog input ground in RSE mode, AISENSE in NRSE mode. Bi-directional.
10	DAC0 OUT	Voltage output signal for analog output Channel 0.
11	AGND	Analog ground. Analog output ground for analog output mode. Analog input ground for DIFF or NRSE mode. Bi-directional.
12	DAC1 OUT	Voltage output signal for analog output Channel 1.
x 13	DGND	Digital ground. Output.
14-21	PA0 through PA7	Bidirectional data lines for Port A. PA7 is the MSB, PA0 the LSB.
22-29	PB0 through PB7	Bidirectional data lines for Port B. PB7 is the MSB, PB0 the LSB.
30-37	PC0 through PC7	Bidirectional data lines for Port C. PC7 is the MSB, PC0 the LSB.
38	EXTTRIG	External control signal to start a timed conversion sequence. Input.
39	EXTUPDATE*	External control signal to update DAC outputs. Input.
40	EXTCONV*	External control signal to trigger A/D conversions. Bi-directional.
41	OUTB0	Counter B0 output.
42	GATB0	Counter B0 gate. Input.
43	COUTB1	Counter B1 output or pulled high (selectable).
44	GATB1	Counter B1 gate. Input.
45	CCLKB1	Counter B1 clock (selectable). Input.
46	OUTB2	Counter B2 output.
47	GATB2	Counter B2 gate. Input.
48	CLKB2	Counter B2 clock. Input.
49	+5V	+5 V out, 1 A maximum. Output.
50	DGND	Digital ground. Output.
*Indicates that the signal is active low.		

Differential nonlinearity (DNL) in a D/A system is a measure of deviation of code width from 1 LSB. In this case, code width is the difference between the analog values produced by consecutive digital codes. A specification of ± 1 LSB differential nonlinearity ensures that the code width is always greater than 0 LSBs (guaranteeing monotonicity) and is always less than 2 LSBs.

Digital I/O

Number of channels 24
 Compatibility TTL
 Digital logic levels

Level	Min	Max
Input low voltage	-0.3 V	0.8 V
Input high voltage	2.2 V	5.3 V
Input low current ($V_{in} = 0.8$ V)	-	-1.0 μ A
Input high current ($V_{in} = 2.2$ V)	-	1.0 μ A
Output low voltage ($I_{out} = 2.5$ mA)	-	0.4 V
Output high voltage ($I_{out} = -2.5$ mA)	3.7 V	-

Darlington drive output current (Ports B and C only)
 ($R_{EXT} = 700 \Omega$, $V_{EXT} = 1.7$ V)..... ± 2.5 mA min, ± 4 mA max
 Handshaking 3-wire (requires 1 port)
 Power-on state Configured as input
 Data transfers Interrupt, programmed I/O

Timing I/O

Number of channels 3 counter/timers
 Resolution 16 bits
 Compatibility TTL, gate and source pulled high with
 4.7 k Ω resistors
 Base clocks available 2 MHz
 Base clock accuracy 0.01%
 Max source frequency 8 MHz
 Min source pulse duration 60 ns
 Min gate pulse duration 50 ns
 Data transfers Programmed I/O