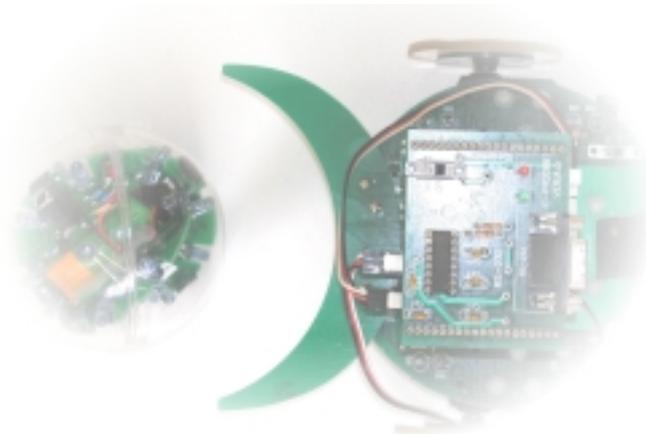




Ocupação Científica de Jovens nas Férias 2003

8 a 12 de Setembro 2003

Instituto Superior Técnico – Instituto de Sistemas e Robótica



Manual de construção de um Robot Futebolista

e

Interface Gráfica

Índice	Pág.
1 Introdução	3
1.1 Estrutura Funcional de um Robot	3
2 Factores a Ter em Conta na Construção de um Robot	4
2.1 Construção da Plataforma	4
2.2 Escolha dos Motores	4
2.2.1 Ajuste do Binário e da Velocidade	5
2.2.2 Controlo de Velocidade de um Motor DC	5
2.2.3 Controlo de Direcção	5
3 Material Incluído no <i>Kit</i> Electrónico	6
3.1 Descrição Geral	6
3.2 Funcionamento	7
4 Montagem dos Componentes	8
4.1 Noções Básicas de Soldadura	8
4.2 Componentes	9
4.2.1 Resistências	9
4.2.2 <i>Pack</i> de Resistências	10
4.2.3 Condensadores	11
4.2.4 Díodos e <i>Leds</i>	11
4.2.5 Circuitos Integrados	12
4.3 Construção das Placas constituintes do Robot	13
4.3.1 Construção da Placa de Controlo ID-PIC002	13
4.3.2 Construção da Placa de Interface Digital ID-DS002	14
4.3.3 Construção da Placa de Interface Analógico ID-AS002	15
4.3.4 Construção da Placa base de Sensores ID-BASEIR002	16
4.3.5 Montagem da Placa Programadora ID-PROG002	18
4.3.6 Interligação das várias placas que constituem o robot	19
4.3.7 Meia-lua	21
4.3.8 Sensores de Contacto	22
5 Interface Gráfica	23
5.1 Como funciona?	23
5.2 O que necessita para programar?	23
5.3 Como Programar?	24
5.4 Início do Programa	24
5.5 Colocação dos vários Blocos	26
5.6 Editar/Apagar/Substituir/Copiar Blocos	27
5.6.1 Editar/Substituir Blocos	28
5.6.2 Inserção de um bloco entre dois blocos	30
5.6.3 Copiar e inserir	31
5.6.4 Copiar e inserir	32
5.7 Estrutura do Programa	34
6 Tipos de Blocos	36
6.1 Blocos de Comparação Simples	36
6.2 Blocos de Comparação com AND Lógico	36
6.3 Blocos de Actuação de Variáveis Inteiras	38
6.4 Bloco de Actuação Binária	39
6.5 Bloco regulador de Velocidade	39
6.6 Bloco de Sensores de Infravermelho para actualização do PWM	40
6.6.1 Configuração dos Sensores a utilizar	40
6.6.2 Configuração de NÍVEIS	41
6.6.3 Configurações de PWM	41
6.6.4 Equivalência	42

6.7	Blocos para movimentos pré-definidos	43
7	Como fazer o Download para o Robot?.....	44
7.1	Mensagens de erro	46
8	Exemplos.....	47

1 Introdução

Os robots têm sido objecto da imaginação e das fantasias dos seres humanos através dos tempos. Até há bem pouco tempo, a imagem que se tinha deste tipo de dispositivos era a de que se tratavam de sistemas de uma grande complexidade, de elevado custo, compostos por uma amálgama de ligações eléctricas e controlados por complicados sistemas computadorizados.

No entanto, nos últimos anos, os avanços que se têm verificado na tecnologia dos microcontroladores, com a redução drástica de dimensão e custo a contrastar com o espantoso aumento de potencialidades, vem permitir a realização, de uma forma simples, de um conjunto de sistemas robóticos capazes de desenvolver autonomamente tarefas de alguma complexidade.

É de notar que a construção de um robot não se resume à sua programação. Na sua concepção estão envolvidos um conjunto de conhecimentos inerentes às áreas da mecânica, electrónica e programação, que tornam esse processo um desafio extremamente envolvente.

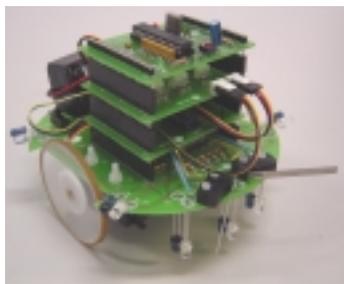


Fig 1.1 - Robot Circular Multi-Sensor

Neste manual são detalhados os pormenores de construção de um pequeno robot, tendo como base um *kit* electrónico desenvolvido pelos autores deste projecto (figura 1.1). O utilizador adquirirá ao longo do processo de construção a percepção das diferentes etapas de concepção de um robot, que vão desde a soldadura dos seus componentes até à sua programação recorrendo à utilização de uma interface gráfica.

1.1 Estrutura Funcional de um Robot

A estrutura funcional de um robot pretende dar resposta aos três problemas fundamentais que lhe são postos: *Onde Estou? Para onde vou? Como vou?*

Quando um robot se desloca num determinado ambiente, faz uso dos seus sensores para se localizar e para identificar os seus objectivos. Através dos actuadores poderá deslocar-se ou manipular algum objecto. Finalmente, estas acções de percepção e de actuação são coordenadas pelo controlador de bordo. A figura 1.2 resume este funcionamento.

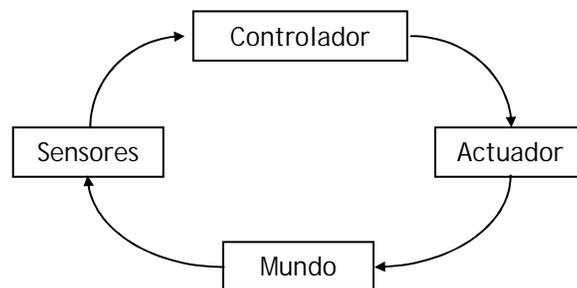


Fig. 1.2 - Ciclo de funcionamento de um robot

2 Factores a Ter em Conta na Construção de um Robot

Tendo em conta que este manual se centra sobre os pormenores de concepção das componentes electrónica e de programação do robot, nesta secção pretende-se apontar, de uma forma resumida, alguns detalhes adicionais a ter em conta na construção de um robot.

2.1 Construção da Plataforma

A plataforma poderá assumir diferentes formas, no entanto na sua concepção deverão ser tidos em conta os seguintes factores:

- 1 Simplicidade - minimizar o número de partes móveis e a complexidade do robot;
- 2 Robustez - resistência a impactos;
- 3 Modularidade - o robot deve ser composto por um grupo de módulos que se interligam, de tal forma que um dos módulos possa ser substituído sem necessidade de remoção dos restantes.

Poderão ser utilizados diferentes materiais, mas os mais comuns são o PVC, inox, alumínio, contraplacados e acrílicos.

A figura 2.1 apresenta uma estrutura comum a muitos robots, com dois motores independentes (tracção diferencial) e uma roda livre para manter o equilíbrio.

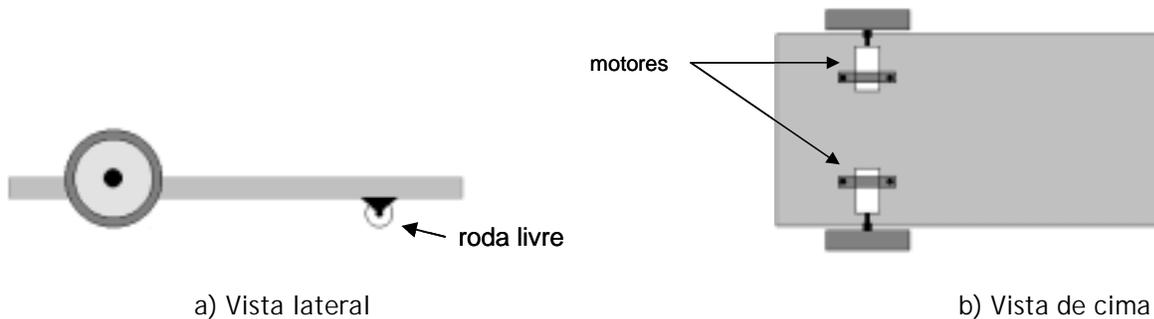


Fig. 2.1 - Robot com tracção diferencial

2.2 Escolha dos Motores

Na escolha do motor apropriado a uma determinada aplicação robótica, deverão ser tidos em conta os seguintes factores:

- 1 Tamanho e forma;
- 2 Massa;
- 3 Binário disponível;
- 4 Velocidade máxima (sem falha de comutação, sem falha nas engrenagens);
- 5 Máxima corrente/binário (sem sobreaquecimento, com dissipação das perdas).

De entre os diferentes motores que se podem encontrar no mercado - corrente contínua, passo-a-passo ou corrente alternada - os motores de corrente contínua, também conhecidos por motores DC (*Direct Current*), são os que se encontram mais difundidos devido à sua melhor relação potência/volume e também devido à grande variedade existente.

2.2.1 Ajuste do Binário e da Velocidade

Normalmente o motor não se encontra preparado para ser acoplado directamente a uma roda, quer porque a velocidade é excessiva, quer porque o binário que fornece não é suficiente. Para adaptar o motor às necessidades é utilizado um sistema de redução (engrenagens de rodas dentadas ou roldanas) para aumentar o binário e diminuir a velocidade do seu veio.

2.2.2 Controlo de Velocidade de um Motor DC

A velocidade de um motor pode ser controlada através da sua tensão (corrente) de entrada. No entanto, para evitar sobreaquecimentos nos componentes, em vez de um sinal contínuo é usado um sinal PWM (Pulse Width Modulation) onde a largura dos impulsos controla a tensão enviada e por sua vez a velocidade de rotação. A velocidade do motor varia proporcionalmente à área debaixo da porção positiva de cada período (ver figura 2.2).

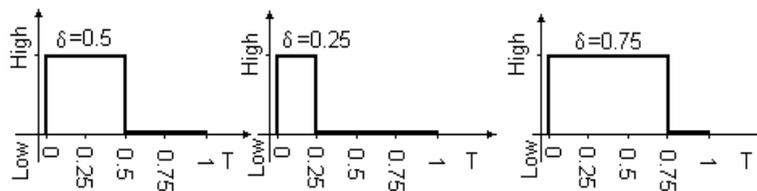


Fig. 2.2 - Sinal PWM

O facto de muitos microcontroladores possuírem este tipo de canal, torna bastante simples a sua utilização. Por outro lado, são reduzidas as perdas térmicas nos componentes, pois nem sempre se tem tensão aplicada.

2.2.3 Controlo de Direcção

A inversão de rotação dos motores pode ser conseguida revertendo a voltagem aplicada. A solução clássica para esta situação é o uso de um esquema tipo ponte-H, tal como é exemplificado na figura 2.3. O sinal PWM poderá ser aplicado no terminal *Enable*.

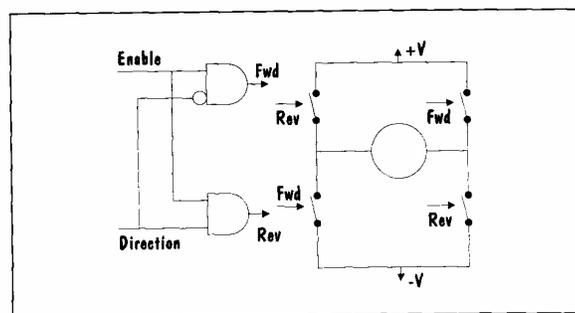


Fig. 2.3 - Ponte H

3 Material Incluído no *Kit* Electrónico

O *kit* electrónico disponibilizado para a construção do Robot Futebolista, é o seguinte:

- 1 Placa de Controlo c/ PIC (ID-PIC002)
- 2 Placa de Interface Digital (ID-DS002)
- 3 Placa de Interface Analógico (ID-AS002)
- 4 Placa Base com sensores Infravermelhos (ID-BASEIR002)
- 5 Placa Programadora (ID-PROG002)
- 6 LDR (Light dependent resistor)
- 7 Cabo de Ligação Série para PC
- 8 2 Motores Servos HITEC HS311
- 9 Componentes diversos

3.1 Descrição Geral

Este *kit* contém todos os componentes electrónicos necessários para construir um pequeno robot.

Este *kit* foi concebido para ser de fácil montagem e programação, permitindo a sua utilização por todos aqueles que estão interessados nas áreas da robótica e automação.

O robot a construir permitir-lhe-á desenvolver a sua capacidade de:

- 1) identificar e construir circuitos electrónicos;
- 2) entender as diferentes partes que o constituem
 - 1 Sensores;
 - 2 Actuadores;
 - 3 Microcontroladores.
- 3) desenvolver os seus próprios programas, levando o robot a executar um conjunto diverso de actividades.

Sensores

Engloba três receptores de infravermelhos que permitem detectar a bola emissora de infravermelhos; quatro pares de sensores led emissor e LDR, para permitir a localização do robot no campo; inclui ainda 2 interruptores de contacto, que permitem detectar a colisão com obstáculos.

Actuadores

Transformação de um par de servo motores de variação angular para variação de velocidade angular.

Microcontrolador

É o “cérebro” do robot. Aqui é tratada a informação vinda dos sensores e são tomadas as decisões de movimento a efectuar.

3.2 Funcionamento

A placa ID-PIC002 é uma placa de circuito impresso onde será colocado o microcontrolador PIC16F876. Esta placa permite a ligação de sensores e actuadores através de dois barramentos, os quais permitem o acesso a todos os pinos das portas de entrada/saída (E/S).

Estão incluídas no *kit* outras duas placas que permitem o acesso aos vários pinos do microcontrolador, sendo uma destinada aos pinos digitais (ID-DS002) e outra aos pinos analógicos (ID-AS002). Nesta última é possível a utilização de uma única entrada analógica para ler 8 entradas analógicas multiplexadas.

Foi também introduzido no *kit* um módulo de sensores (ID-BASEIR002) para permitir ao robot realizar o seguimento de uma bola emissora de infravermelhos, detectar a localização do robot no campo e ainda detectar a colisão do robot com paredes, robots e outros objectos. Como o processador utilizado não permite a leitura simultânea dos 7 sensores incluídos, esta placa foi criada para que só um dos sensores se encontre a funcionar em cada instante de tempo. Este método permite obter um baixo consumo de energia.

A ligação de sensores de saída analógica é simples, bastando ligar o pino de saída dos sensores às entradas de sinal analógico disponível neste robot, mais concretamente na placa de interface analógica ID-AS002. Nesta placa os terminais EA0 a EA7 estão multiplexados numa entrada analógica (canal 0 do PortA) e irão ser ocupados pelos sete sinais provenientes dos sensores. As restantes 4 entradas analógicas, canais 1 a 4 do PortA, estão disponíveis nos terminais PA1, PA2, PA3 e PA5. O terminal PA4 é uma entrada/saída digital.

A placa de interface digital ID-DS002 disponibiliza 11 entradas/saídas digitais, que podem ser utilizadas para aumentar o número de sensores/actuadores. Neste *kit* são incluídos 2 *micro-switches* que irão ocupar duas dessas entradas.

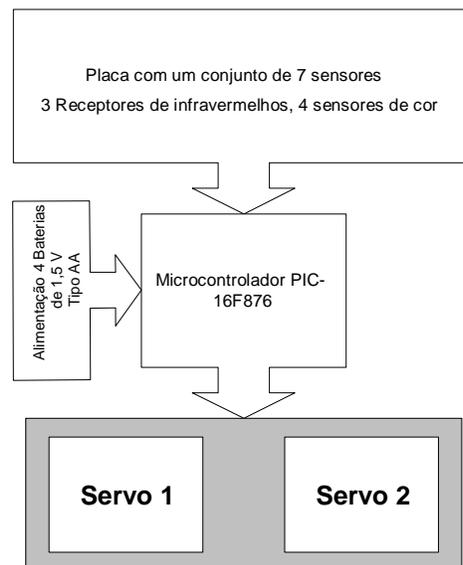


Fig. 3.1 - Arquitectura de *Hardware*

Como actuadores, optou-se pela utilização de dois servo-motores modificados que são actuados pelas saídas de PWM existentes no microcontrolador. Os servo-motores apresentam como vantagens o facto de já possuírem acoplado ao eixo do seu motor uma caixa redutora e também já possuírem internamente um circuito electrónico que realiza a ponte-H. A figura 3.1 resume o funcionamento do robot.

4 Montagem dos Componentes

4.1 Noções Básicas de Soldadura

O sucesso da montagem do *kit* depende em grande parte da soldadura dos componentes nas placas de circuito impresso (ver figura 4.1). Torna-se assim necessário o cumprimento de algumas regras básicas nesse processo:

- 3 Utilizar, sempre que possível, um ferro de soldar de ponta fina e de potência não superior a 25 W, podendo estes, ser encontrados em casas especializadas em electrónica;
- 4 Durante a sua utilização deverá manter sempre limpa a ponta do ferro de soldar utilizando para esse efeito uma esponja que deverá estar sempre húmida.
- 5 Nunca utilizar uma ponta suja ou lixada, sendo preferível a sua substituição se essa situação se verificar;
- 6 Deverá utilizar solda fina (0,7 mm);
- 7 É aconselhado a limpeza dos pinos do componente com uma lixa fina, antes de o soldar;
- 8 Pegue no ferro de soldar, coloque a sua ponta no terminal do componente e aplique o fio de solda (ver figuras 4.1 e 4.2). Este processo não deverá demorar mais do que 5 segundos, podendo o não cumprimento desta regra levar à destruição do componente em causa;
- 9 Após realizada a soldadura, a superfície desta deverá apresentar um brilho metálico, sendo este o momento ideal para cortar os extremos dos componentes;
- 10 Se necessitar retirar a solda deverá utilizar sempre um dessoldador;
- 11 Quando soldar um componente sensível deverá segurá-lo com umas pinças de modo a dissipar o calor, caso contrário o componente poderá ficar danificado;
- 12 Nunca utilizar mais solda do que a necessária para evitar o risco de curto-circuitos entre pistas.

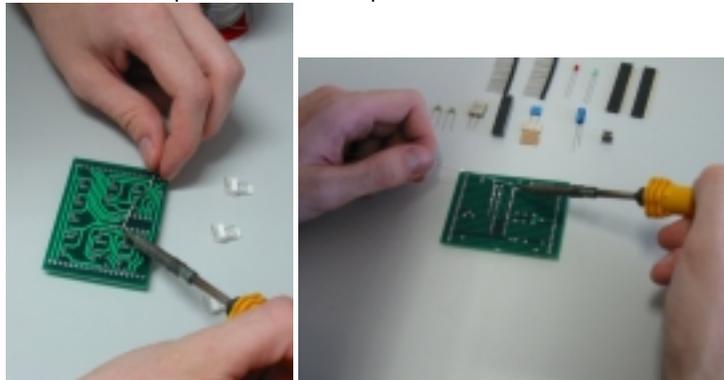


Fig. 4.1 - Soldadura dos componentes

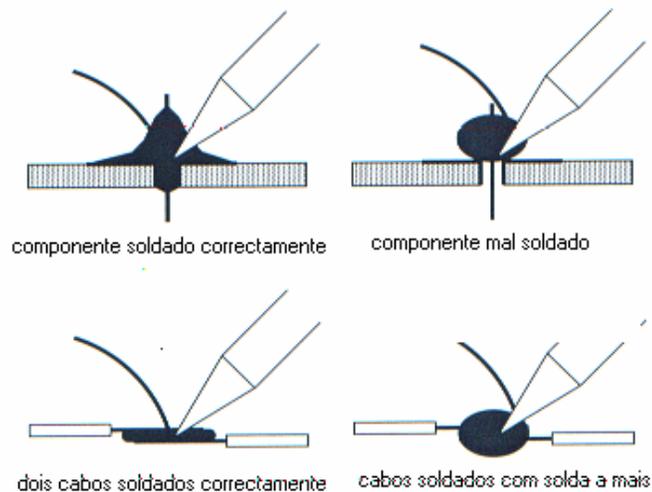


Fig. 4.2 - Métodos de soldadura

4.2 Componentes

Um circuito electrónico é constituído por uma vasta gama de componentes, com diferentes formas, tamanhos e códigos que permitem a sua distinção. Na construção de um circuito o utilizador deverá saber identificar os diferentes componentes.

É de notar que alguns dos componentes são polarizados, devendo a sua montagem ser efectuada de uma única maneira. A deficiente montagem destes componentes poderá provocar o mau funcionamento do circuito eléctrico ou até mesmo a sua destruição.

Esta secção tem como objectivo familiarizar o utilizador com os diferentes componentes existentes num simples circuito electrónico.

4.2.1 Resistências

Geralmente as resistências apresentam-se como pequenos cilindros com algumas riscas coloridas. As resistências utilizadas neste *kit* são de $\frac{1}{4}$ Watt, o que corresponde a uma baixa potência, mas suficiente para o bom funcionamento do *robot*. O código de cores utilizado é o standard, consistindo em 4 bandas de cores em torno do cilindro. As primeiras duas bandas formam a mantissa e a terceira banda representa um expoente.

Uma resistência pode ser identificada formando um número cujo algarismo das dezenas corresponde à primeira banda de cor e o algarismo das unidades corresponde à segunda banda de cor (Ver Tabela 1 para a respectiva correspondência de cores). A terceira banda de cores corresponde ao expoente de uma potência de base 10, pelo qual teremos de multiplicar o número obtido pelas duas primeiras bandas.

A quarta banda representa a tolerância da resistência. Se esta banda for prateada a resistência tem uma tolerância de 10 %, se for dourada então a tolerância é de 5%. A tolerância representa o intervalo de erro do valor nominal da resistência fornecida pelo fabricante. Exemplo: para uma resistência de 100 Ω com uma tolerância de 5%, o valor real da resistência poderá estar compreendido entre 95 e 105 Ω .

Cor	Valor da Banda	Factor Multiplicativo
Preto	0	1
Castanho	1	10
Vermelho	2	100
Laranja	3	1000
Amarelo	4	10000
Verde	5	100000
Azul	6	1000000
Violeta	7	
Cinza	8	
Branco	9	

Tabela 1 - Código de cores das resistências

Como exemplo, considere uma resistência composta pela seguinte sequência de cores: **laranja, branco, vermelho**.

$$\left. \begin{array}{l} \text{Laranja} = 3 \\ \text{Branco} = 9 \\ \text{Vermelho} = 2 \Rightarrow 10^2 \end{array} \right\} \Rightarrow 39 \left\{ \Rightarrow 39 \times 10^2 = 3900 \Omega \right.$$

(A resistência tem portanto o valor de 3900Ω)

4.2.2 Pack de Resistências

Num circuito electrónico pode-se também encontrar outro tipo de resistências, não unitárias, agrupadas em conjuntos que variam de 3 a 9 resistências (ver figura 4.3 que representa um *pack* de 7 resistências).

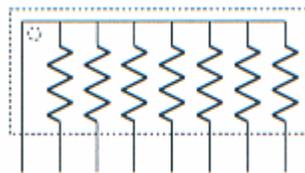


Fig. 4.3 - Pack de resistências

No *kit* disponibilizado, o conjunto utilizado tem 8 resistências mais um pino comum (identificado por um ponto). De notar que este componente é polarizado, sendo necessário verificar se está correctamente colocado antes de o soldar.

4.2.3 Condensadores

Encontram-se no mercado vários tipos de condensadores (monolíticos, electrolíticos e tantalum).

Monolíticos

Sendo a sua constituição de *poliester* ou cerâmica são, geralmente, condensadores pequenos e de baixa capacidade (pF ou nF). Estes condensadores nunca são polarizados.

Electrolíticos

Apresentam normalmente uma forma cilíndrica, revestida por um encapsulamento plástico, apresentando grandes valores de capacidade ($> 1 \mu\text{F}$). Este tipo de condensador é polarizado e a sua dimensão física aumenta com o aumento da sua capacidade.

Tantalum

São condensadores bastante compactos e em forma de gota. Os seus valores de capacidade são da mesma ordem que a do tipo anterior, apresentando a vantagem de serem mais fiáveis e a desvantagem de terem um custo mais elevado. Estes condensadores são sempre polarizados.

Nota: Os condensadores polarizados poderão explodir quando a sua polaridade não é respeitada. Por este motivo, recomenda-se ao utilizador o respeito da sua polaridade aquando da sua montagem, utilizando para tal a indicação de (+) e/ou (-) da sua embalagem. Existem várias maneiras de indicar o valor do condensador, sendo usual, para condensadores maiores que $1 \mu\text{F}$, o valor estar representado no componente. Em alguns casos o μ funciona como ponto decimal podendo um condensador de $4,7 \mu\text{F}$ ser representado por $4\mu 7$. Para pequenos condensadores os seus valores são representados em picofarads ($1000000\text{pF}=1\mu\text{F}$). Outra maneira de o representar é por exemplo 473, em que o procedimento é igual ao utilizado para as resistências, ou seja, $47 \times 10^3 \text{pF} = 47000\text{pF} = 0.047\mu\text{F}$.

4.2.4 Díodos e Leds

Os díodos e *leds* (*Light Emitting Diodes*) têm dois pinos: o ânodo e o cátodo. O ânodo deve ser ligado a um terminal de tensão positiva em relação ao cátodo para permitir o fluxo de corrente. Caso a polaridade não seja respeitada então o componente não permite a passagem de corrente. A figura 4.4 mostra como se podem identificar os pinos dos díodos e *leds*.

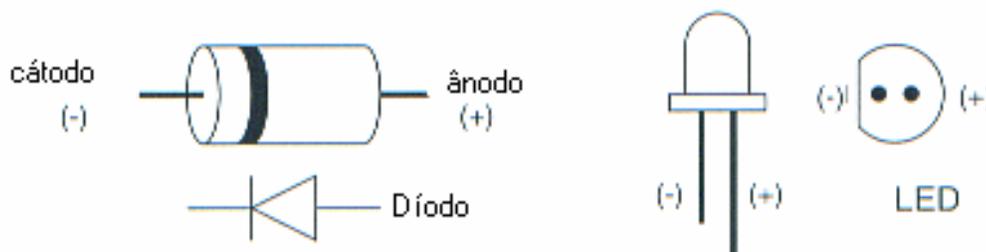


Fig. 4.4 - Polaridade dos díodos e *leds*

Geralmente os díodos são pequenos cilindros com uma pequena marca à sua volta, a qual permite a identificação do cátodo. Os *leds* são díodos que emitem luz quando uma corrente flui pelos seus terminais. A identificação da sua polaridade é feita pela verificação da "perna" mais comprida (ânodo(+)), ou então pelo lado em que a circunferência do *led* tem uma pequena parte achatada (cátodo(-)).

4.2.5 Circuitos Integrados

Os circuitos integrados (IC) são componentes constituídos internamente por circuitos complexos, podendo ser encontrados no mercado em diferentes formas e tamanhos. Geralmente, em montagens de circuitos manuais, são utilizados os chamados *DIPs* (*Dual-Inline Packages*).

Os circuitos integrados devem ser montados tendo em conta a sua numeração, a qual não está gravada no encapsulamento. Existem, no entanto, duas marcas que indicam o pino número 1. Estas duas marcas podem ser observadas como reentrâncias no *chip* - a primeira é uma pequena meia lua existente num dos lados sem pinos do *chip*, a qual indica que o pino 1 se encontra à sua direita (quando a meia lua está virada para o utilizador) e a segunda marca é um pequeno círculo existente por cima do pino número 1 (ver figura 4.5). Uma montagem incorrecta deste componente provoca geralmente a sua destruição.

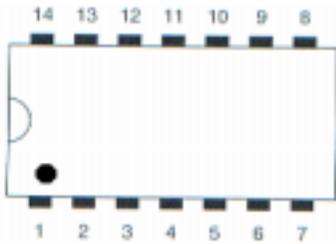


Fig. 4.5 - Formato de um circuito integrado do tipo *DIP*

4.3 Construção das Placas constituintes do Robot

Nesta secção é explicada a montagem do conjunto de placas que constituem o robot. A figura 4.6 apresenta o conjunto de placas de circuito impresso que constituem este *kit* electrónico.

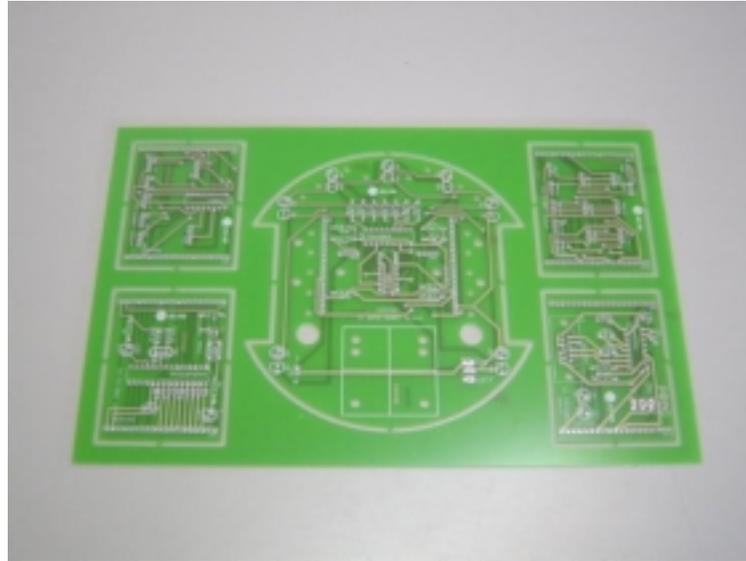


Fig. 4.6 - Conjunto de placas do robot

4.3.1 Construção da Placa de Controlo ID-PIC002

Comece por separar a placa ID-PIC002 (figura 4.7). Para tal corte as pequenas junções que a unem ao conjunto de placas e passe com uma lixa fina se necessário. Identifique os componentes utilizados nesta placa.

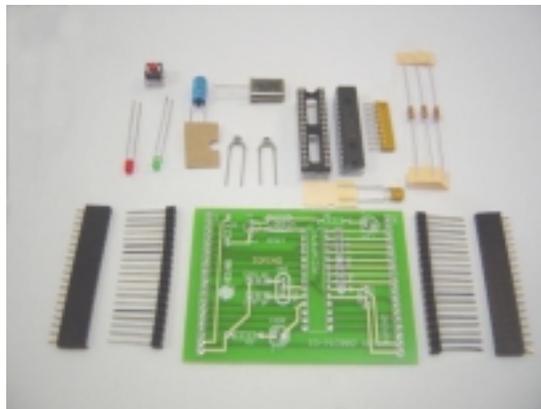


Fig. 4.7 - Material utilizado na construção da placa ID-PIC002

Inicie a construção da placa do microcontrolador soldando as três resistências de $3,3\text{ K}\Omega$ nos locais assinalados na placa. De seguida solde a resistência de 9 pinos tendo em atenção que o ponto na resistência corresponde à via quadrada. Para facilitar a orientação deste componente foi colocado uma marca circular branca na placa a qual deverá coincidir com a marca branca da resistência. Solde o *socket* de 28 pinos do microcontrolador. Tome em atenção a forma desenhada na placa, orientando o *socket* por forma a fazer coincidir os recortes em forma de meia lua. Solde os dois *leds* tendo em conta a sua polarização. Solde os dois

condensadores de 15 pF e o condensador de 100 nF. Solde o condensador electrolítico tendo em atenção a polarização indicada (ver secção 4.2.3). Solde o botão de pressão e o cristal nos locais indicados. Por fim, deverá soldar os dois barramentos macho de 19 pinos e encaixar nestes os 2 barramentos fêmea. Colocar o microcontrolador PIC16F876 no respectivo *socket*.

Na figura 4.8 está representada a placa do microcontrolador após a respectiva montagem dos componentes. É de notar que a placa tem dois barramentos macho e fêmea de 19 pinos, estando os primeiros soldados à placa e os segundos encaixados nestes.

Frente

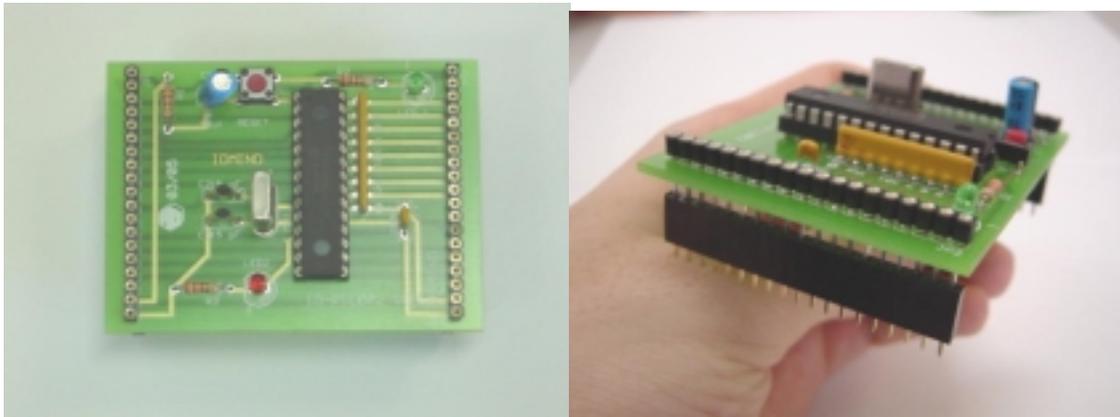


Fig. 4.8 - Placa do microcontrolador

4.3.2 Construção da Placa de Interface Digital ID-DS002

Separe a placa ID-DS002 (figura 4.9). Para tal corte as pequenas junções que a unem ao conjunto de placas e passe com uma lixa fina se necessário. Identifique os componentes utilizados nesta placa.



Fig. 4.9 - Material utilizado na construção da placa ID-DS002

Inicie a construção da placa digital soldando os vários terminais de três pinos com um ângulo de 90°, tendo em atenção que os dois terminais correspondentes ao PWM deverão fazer um ângulo de 180° com os primeiros (ver figura 4.10). Nesta placa deverá também soldar os terminais macho dos dois barramentos de 19 pinos e

encaixar nestes os 2 barramentos fêmea.

Na figura 4.10 é possível observar uma placa de interface digital montada. Nesta está acessível todo o barramento do PortB e alguns dos bits do PortC. Cada sensor encontra-se ligado a um dos terminais de 3 pinos. Na figura encontram-se assinalados, para cada terminal, os pinos correspondentes ao sinal do sensor e sinal de massa, respectivamente, "s" e "-". O pino central fornece um sinal de 5 V.

Como se pode verificar na figura os terminais de ligação C1 e C2 (PWM2, PWM1) estão rodados de 180° em relação aos restantes terminais, sendo as ligações também rodadas de 180°. Tal como a placa controladora esta placa tem os dois barramentos macho de 19 pinos soldados à placa e os outros barramentos (fêmea) encaixados nos primeiros.

Frente

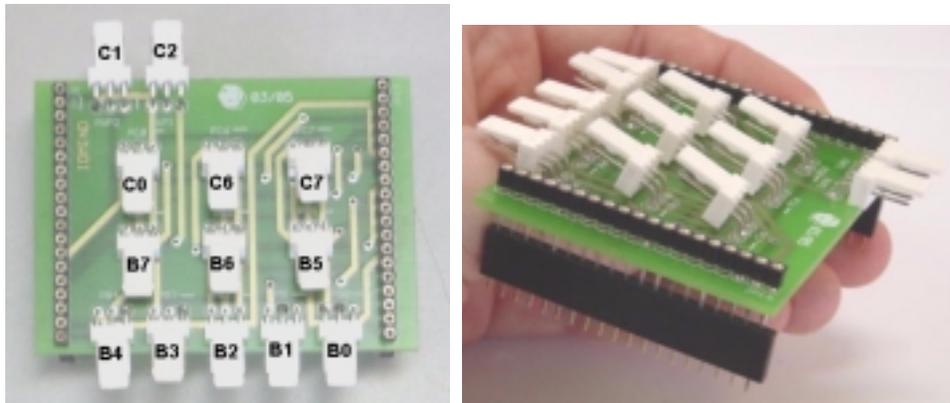


Fig. 4.10 - Placa de interface digital

4.3.3 Construção da Placa de Interface Analógico ID-AS002

Separe a placa ID-AS002 (figura 4.11). Para tal corte as pequenas junções que a unem ao conjunto de placas e passe com uma lixa fina se necessário. Identifique os componentes utilizados nesta placa.

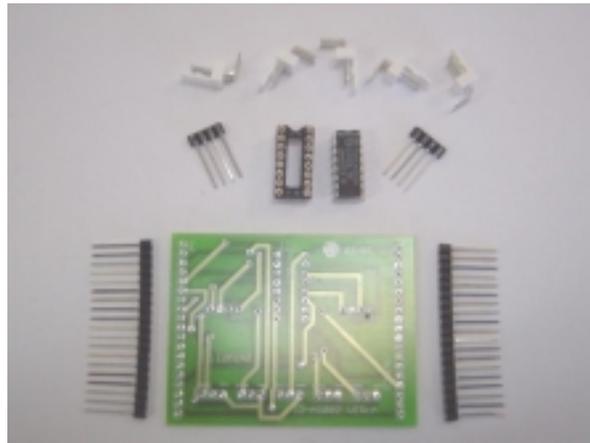


Fig. 4.11 - Material utilizado na construção da placa ID-AS002

Inicie a construção da placa analógica soldando os vários terminais de três pinos com um ângulo de 90° nos locais indicados por PA1, PA2, PA3, PA4 e PA5, seguindo a orientação expressa na figura 4.11. Solde os 2 barramentos macho de 4 pinos nos locais indicados por JP9 a JP10. Solde o *socket* de 16 pinos de acordo com o desenho da placa, e finalmente solde os barramentos macho de 19 pinos à placa. Introduza o integrado 74HC4051 no *socket* (meia-lua com meia-lua).

A figura 4.12 apresenta uma placa de interface analógica completamente montada. Nesta placa é possível ter acesso a todos os pinos do PortA (*port* analógico), estando os pinos 1 a 5 do PortA directamente acessíveis, e

o pino 0 do mesmo *port* multiplexado em 8 entradas correspondentes aos pinos EA0 a EA7.

Frente

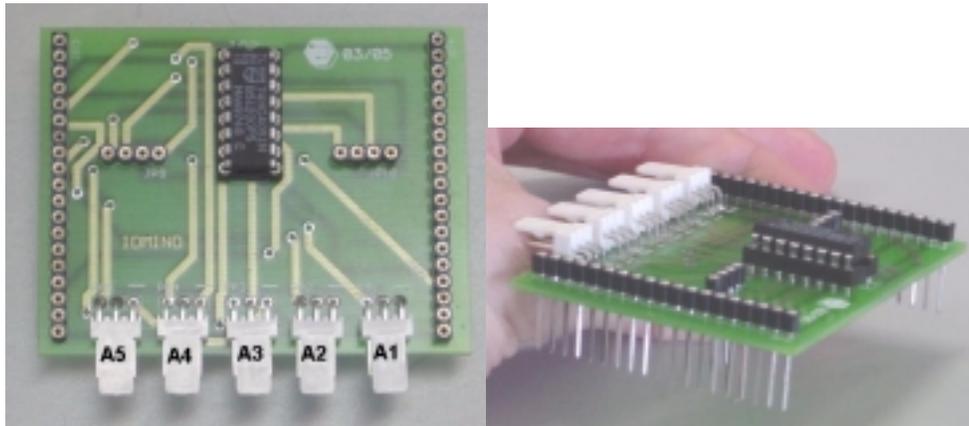


Fig. 4.12 - Placa de interface analógica

4.3.4 Construção da Placa base de Sensores ID-BASEIR002

Separe a placa ID-BASEIR002 (fig. 4.13). Para tal corte as pequenas junções que a unem ao conjunto de placas e passe com uma lixa fina se necessário.

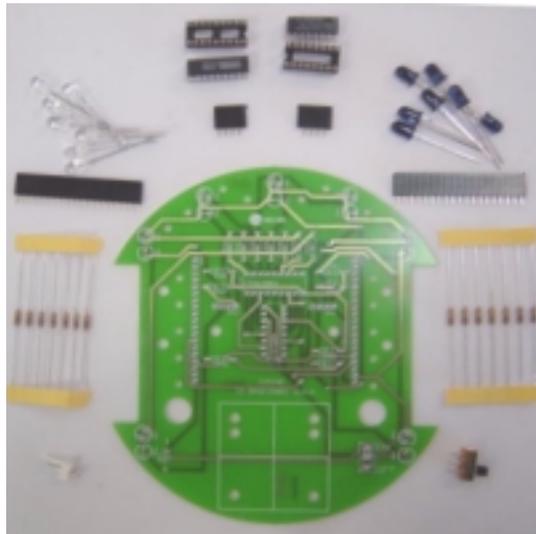


Fig. 4.13 - Material utilizado na construção da placa ID-BASEIR002

Inicie a construção deste módulo soldando as 4 resistências de 33Ω (RIR0, RIR1, RIR5 e RIR6) e as 7 resistências de $10 K\Omega$ (RTR0 a RTR6). Solde os dois sockets de 16 e 18 pinos respeitando os desenhos na placa. Solde os 2 barramentos de 4 pinos fêmea nos locais indicados por JP11 a JP12. Solde os barramentos fêmea de 19 pinos. Solde o interruptor ON-OFF.

Identifique os receptores de infra-vermelhos e a sua polarização (ver secção 4.2.4). Identifique os locais onde estes terão de ser montados. Os receptores (mais escuros) devem ser soldados nos locais indicados por TR2, TR3 e TR4. Estes sensores deverão ser montados de acordo com o fim a que destinam.

Deve montar as 4 LDRs nos locais indicados por TR0, TR1, TR5 e TR6. Monte estes componentes de maneira a

ficarem mais próximos do solo quanto possível. De igual forma monte os leds de emissão verdes nos locais indicados por IR0, IR1, IR5 e IR6. Não cortar as pontas aos leds e LDRs já que poderá ser necessário o seu ajustamento para ter leituras coerentes.



Fig. 4.14 - Suporte de pilhas

Monte o suporte de pilhas de acordo com a figura 4.14. Para tal deverá soldar os pequenos terminais metálicos aos fios do suporte e encaixe-os no respectivo suporte branco de duas entradas. Este suporte irá ser ligado nos terminais de alimentação "+" e "-" da placa ID-BASEIR002. Tome em atenção que quando o ligar deverá fazer coincidir o fio vermelho com o terminal positivo.

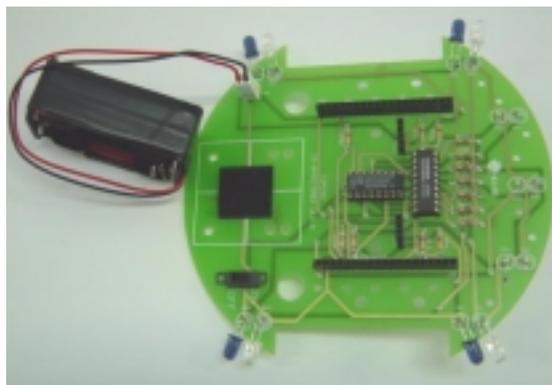
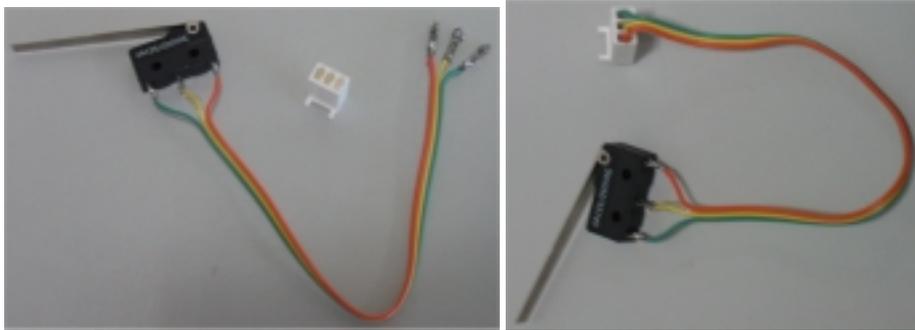


Fig. 4.15 - Montagem do suporte de pilhas

Se pretender montar os *micro-switches* incluídos no *kit* na frente do seu robot (no entanto, no robot futebolista, não deverá montar os *micro-switches* deste modo, ver secção 4.3.8), deverá começar por soldar de um dos lados dos cabos de três fios os *micro-switches*, e do outro lado os pequenos terminais metálicos fornecidos. Encaixe os terminais no suporte branco de três entradas. Tome em atenção que o terminal comum, representado como "COM" no *micro-switch*, tem de entrar no terminal esquerdo do suporte branco, quando os dois pequenos orientadores (pequenos pinos de plástico) do suporte se encontram virados para cima (ver figura 4.16).

Fig. 4.16 - Montagem do *micro-switch*

4.3.5 Montagem da Placa Programadora ID-PROG002

Separe a placa ID-PROG002 (fig. 4.17). Para tal corte as pequenas junções que a unem ao conjunto de placas e passe com uma lixa fina se necessário. Identifique os componentes utilizados nesta placa.

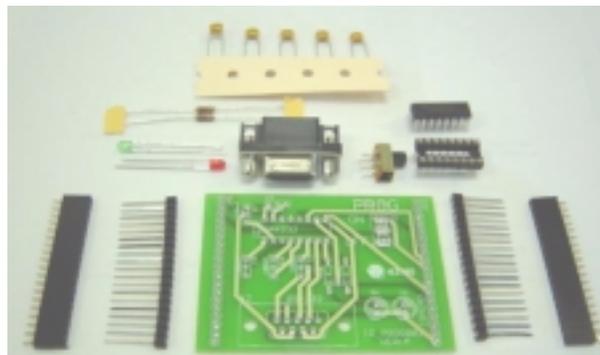


Fig. 4.17 - Material utilizado na construção da placa ID-PROG002

Esta placa permitirá programar o robot através da porta série do seu computador. Solde as duas resistências de $1K\Omega$. Solde o *socket* de 16 pinos tendo em atenção o desenho impresso na placa. Os dois *leds* devem ser soldados em TX (verde) e RX (vermelho), tendo em atenção a polaridade dos mesmos (ver secção 4.2.4). Solde os 5 condensadores de 100 nF (Cp1 a Cp5). Solde os dois barramentos macho de 19 pinos. Solde o interruptor ON-OFF. Solde o conector de porta série no local indicado como RS-232. Por fim encaixe os dois barramentos fêmea de 19 pinos nos barramentos macho e coloque o integrado 232 no respectivo *socket* (meia-lua com meia-lua). No final da montagem a placa deverá apresentar-se como é exemplificado na figura 4.18.

Frente

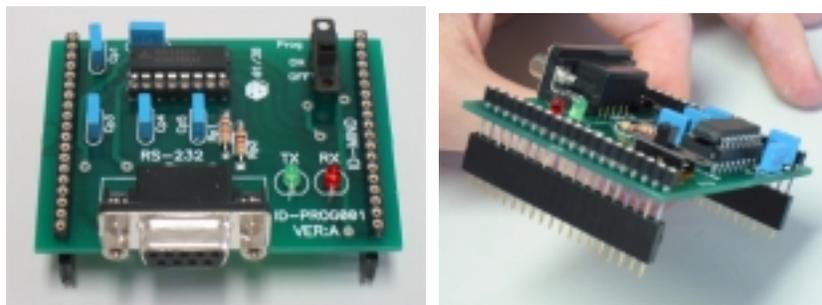


Fig. 4.18 - Vista lateral da placa ID-PROG002

4.3.6 Interligação das várias placas que constituem o robot

Esta secção demonstra como devem ser interligadas as várias placas e componentes que constituem o robot.

Disponha os motores e roda livre de acordo com a disposição expressa na figura 4.19.

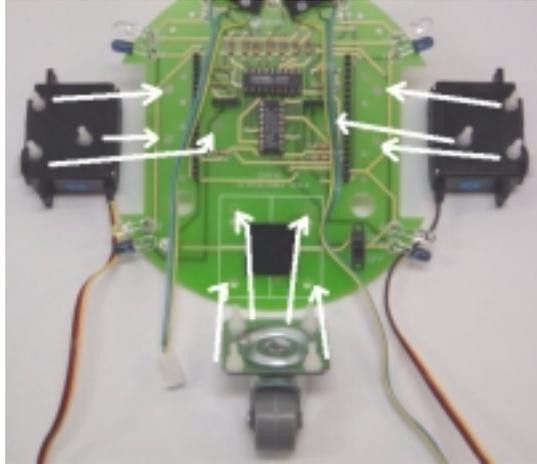


Fig. 4.19 - Motores e roda livre

Os dois motores são colocados na parte de baixo da placa ID-BASEIR002. Para cada um dos motores existem cinco orifícios, devendo utilizar os três orifícios mais atrás (ver figura 4.19). Coloque a roda livre no local respectivo (quatro furos na parte traseira). Ver figura 4.20. Passe os cabos provenientes dos motores servo pelo orifício circular existente na parte central da placa.

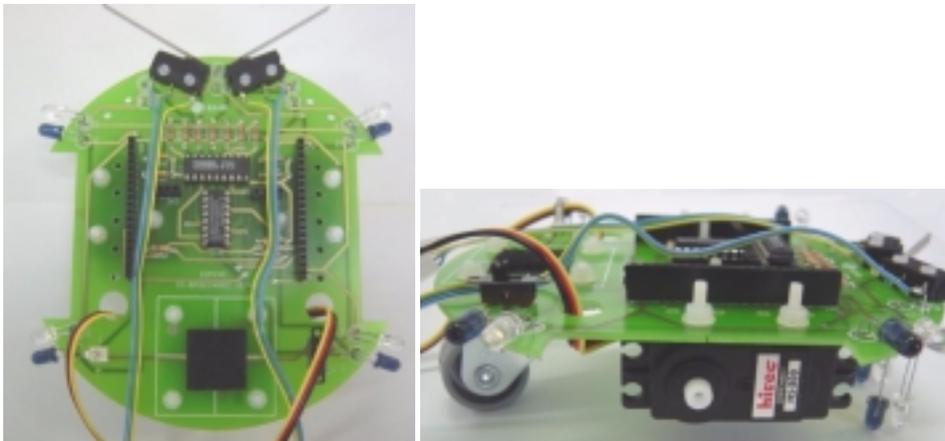


Fig. 4.20 - Montagem dos motores e roda livre

Coloque as rodas nos motores servo e aperte o parafuso no seu centro. Ver figura 4.21.

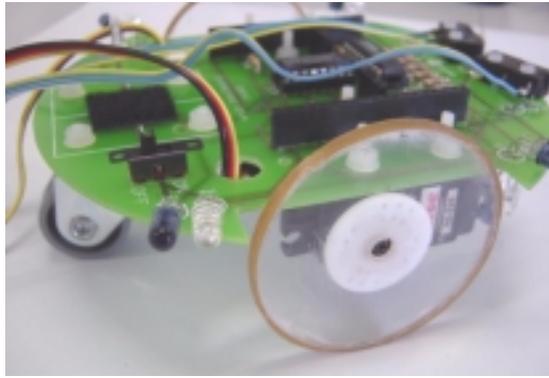


Fig. 4.21 - Montagem das rodas nos motores

Encaixe a placa ID-AS002 na placa ID-BASEIR002. Verifique que todos os pinos que correspondem aos barramentos macho estão a encaixar nos respectivos barramentos fêmea (barramentos de 19 e 3 pinos). Pressione os encaixes até que as duas placas estejam completamente encaixadas. Ver figura 4.22.

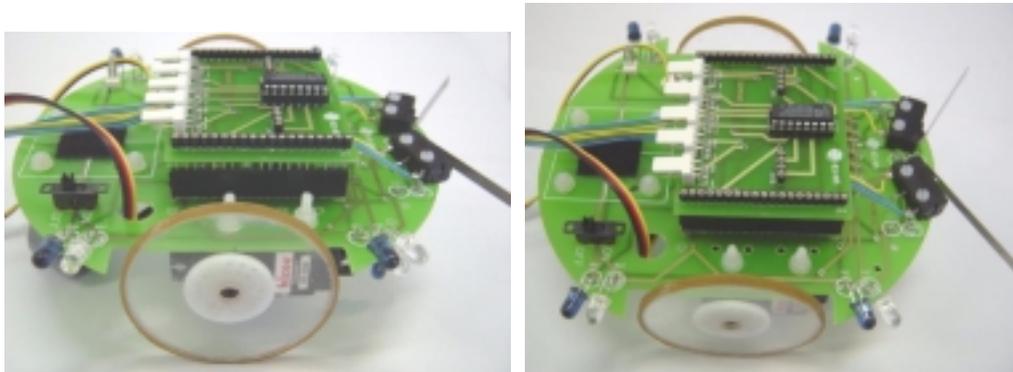


Fig. 4.22 - Encaixe das Placas ID-BASEIR002 e ID-AS002

Encaixe a placa ID-DS002 na placa ID-AS002. Pressione os encaixes até que as placas estejam completamente encaixadas.

Ligue as fichas de terminais dos *micro-switches*. O *micro-switch* esquerdo no terminal PB4 e o direito no terminal PB3 da placa ID-DS002. Ligue as fichas terminais dos dois motores nas saídas de PWM. Motor direito na saída PWM1 e o esquerdo na saída PWM2. Tome em atenção que o fio preto (massa) deverá ficar para o lado esquerdo, quando o robot está de "costas" para si. Ver figura 4.23.

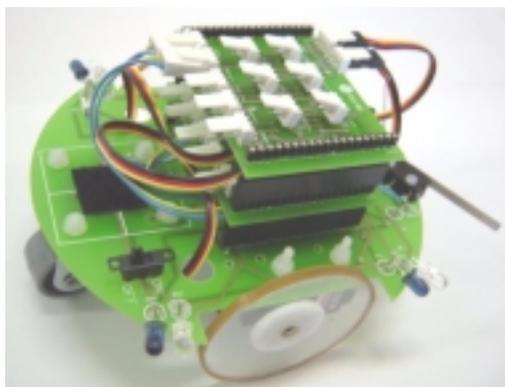


Fig. 4.23 - Encaixe da Placa ID-DS002

Finalmente encaixe a placa ID-PIC002 na placa ID-DS002. Pressione os encaixes até que as duas estejam completamente encaixadas. No final o robot deverá ficar com o aspecto expresso na figura 4.24.

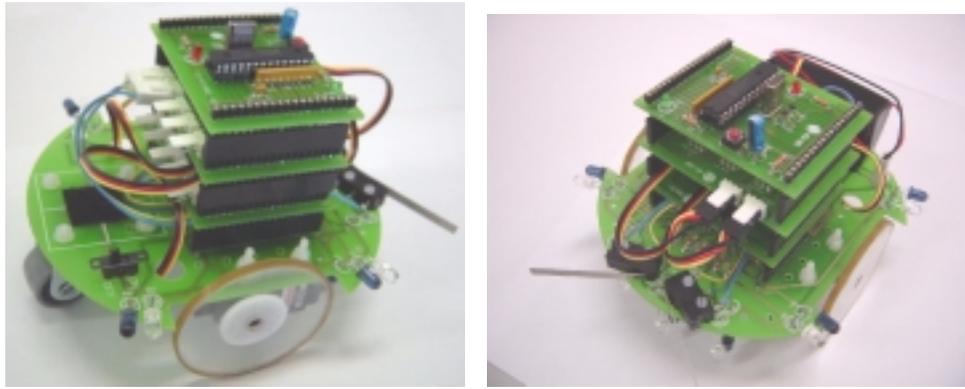


Fig. 4.24 - Robot montado

Tal como foi descrito na secção 3.2, os sensores de infra-vermelhos ocupam apenas uma das cinco entradas analógicas do microcontrolador, ficando disponíveis 4 entradas analógicas que o utilizador poderá usar para acrescentar outros sensores analógicos. A ligação de novos sensores de saída analógica é simples, bastando ligar o pino de saída dos novos sensores às entradas de sinal analógico disponível neste robot, mais concretamente na placa de interface analógica ID-AS002. A placa de interface digital ID-DS002 disponibiliza 11 entradas/saídas digitais, que podem ser utilizadas para aumentar o número de sensores/actuadores. Os 2 *micro-switches* ocupam duas dessas entradas.

4.3.7 Meia-lua

Um dos requisitos fundamentais para um jogo de futebol robótico é manter o domínio da bola. Idealmente, o robot deve poder manter a bola sempre à sua frente. Com vista a garantir esse requisito, recomenda-se a instalação, na parte da frente do robot, de uma meia-lua paralela ao chão. Esta permitirá o controlo da bola de forma segura, obtendo-se grande domínio sobre a sua direccionalidade.

O material que se sugere para a construção desta peça pode ser aproveitado da moldura de placa de circuito impresso que sobra após a montagem das várias placas. Desta placa aproveitar-se-á apenas uma parte, ilustrada na figura 4.25:

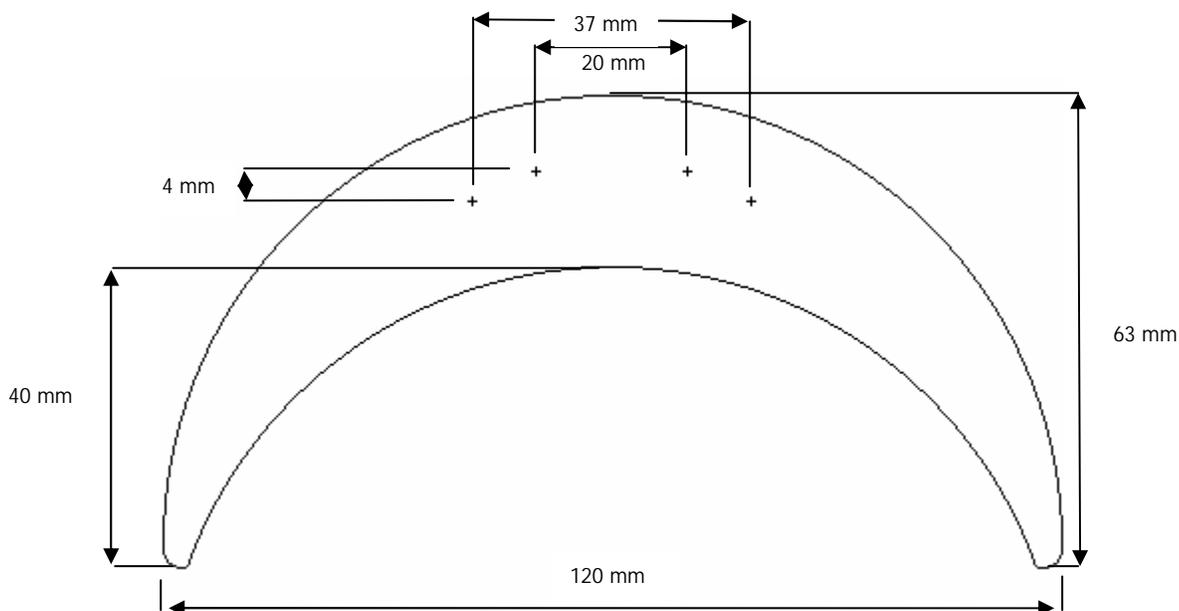


Fig. 4.25 - Meia-Lua

Os furos para fixação (+) devem estar de acordo com as marcações existentes na base do robot. Tendo a meia-lua pronta, aparafusa-se à base conforme é ilustrado na figura 4.26. Esta ocupará o local reservado aos sensores de contacto.



Fig. 4.26 - Meia-Lua montada no robot

4.3.8 Sensores de Contacto

Os sensores de contacto do robot circular são substituídos, no robot futebolista, pela meia-lua que permite o controlo da bola. Ficam portanto disponíveis para ser montados noutro local onde possam ser úteis para a detecção de colisões.

Recomenda-se que estes sensores sejam montados de forma a possibilitar a detecção de colisões não desejadas, dando assim informação ao controlador que permitirá evitar situações de conflito dentro do campo. Um local lógico para a sua montagem é na própria Meia - Lua.

5 Interface Gráfica

5.1 Como funciona?

O programa Prograf é uma interface gráfica que permite ao utilizador programar um microcontrolador de um pequeno robot sem ter de o programar em linguagem de baixo nível.

Esta programação gráfica é feita utilizando pequenos blocos (fig. 5.1) de funções em assembly pré-definidas.

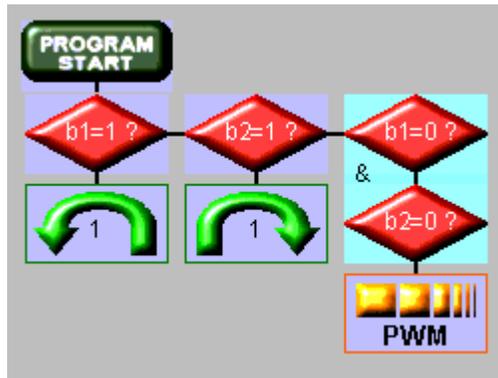


Fig. 5.1 - Blocos de Programação

Depois do utilizador acabar de construir graficamente o programa, o compilador vai compilar os vários blocos e gerar um programa em linguagem de baixo nível. Foi incluída nesta mesma rotina a possibilidade de enviar directamente o programa compilado para o robot.

A interface permite ainda gravar, abrir e imprimir qualquer ficheiro por ela criado.

5.2 O que necessita para programar?

Para programar o seu robot necessita de ter um microcontrolador PIC16F876 com o programa residente fornecido pela IdMind.



Fig. 5.2 - Comunicação série RS232

Sem este programa não é possível fazer o download do programa, já que é este que gere a programação do PIC.

É ainda necessário que o computador em uso tenha uma entrada série RS232 livre (Fig. 5.2).

5.3 Como Programar?

O utilizador poderá programar o robot utilizando um ou mais dos blocos apresentados na figura 5.3.

Existem 4 categorias de blocos:

Na primeira categoria são apresentados 4. Estes blocos executam operações de comparação simples, e operações de comparação com AND Lógico.

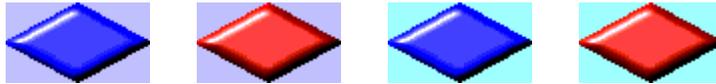


Fig. 5.3 - Blocos de comparação simples e de comparação c/ AND lógico

Na segunda categoria existem 4 blocos de actuação (figura 5.4). É permitido actuar em variáveis inteiras (entre 0 e 255) e actuar os sinais digitais do microcontrolador utilizando os seguintes blocos:

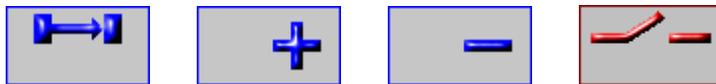


Fig. 5.4 - Blocos de actuação

Na terceira categoria são apresentados dois blocos que permitem actuar a velocidade dos motores.

Com o bloco seguinte (figura 5.5) é possível regular a velocidade de cada uma das rodas dada uma situação especial.



Fig. 5.5 - Bloco regulador de velocidade

Com o bloco da figura 5.6, é possível configurar os três pares emissores/receptores, receptores da frente para o seguimento de uma faixa no chão ou para o seguimento de uma bola emissora de infravermelhos.



Fig. 5.6 - Bloco que permite o seguimento de uma faixa no chão

Finalmente, o último conjunto de blocos (figura 5.7) é constituído por quatro blocos pré-definidos que permitem forçar o robot a ter um determinado comportamento durante x segundos.



Fig. 5.7 - Conjunto de blocos que força o robot a um determinado comportamento

O programa resultante é constituído pela composição de vários blocos.

5.4 Início do Programa

Quando se inicia a construção do programa, o utilizador deverá identificar e configurar as entradas e saídas do microcontrolador. É de notar que à partida todos os terminais podem ser entradas ou saídas, por isso é necessária a sua definição.

Para definir as entradas e saídas, deverá carregar no botão esquerdo do rato na imagem com o texto **Program start** (figura 5.8).



Fig. 5.8 - Botão PROGRAM START

Para poder utilizar uma porta de entrada analógica deverá marcar neste quadro (figura 5.9) qual a porta onde se encontra ligado o dispositivo analógico. Por exemplo se ligar um detector de distância analógico no pino 1 do PortA e um sensor analógico de temperatura no pino 3 do PortA, deverá marcar o quadrado à frente das Entradas analógicas correspondentes, tal como se mostra na figura.

No caso das entradas/saídas digitais basta em cada pino correspondente identificar quais os pinos que correspondem às entradas e quais correspondem às saídas.

As 5 variáveis auxiliares (V1, V2, V3, V4 e V5) inteiras (0 a 255), são variáveis para a exclusiva utilização pelo programador. Este poderá utilizar estas variáveis como contadores, valores de nível para os vários sensores analógicos ou como variáveis de sistema que permitem armazenar valores de entradas analógicas. No menu de configuração é possível pré definir os valores iniciais destas variáveis. Exemplo: na figura abaixo V1 é configurada inicialmente como tendo o valor 12, V2 o valor 200 e V3, V4 e V5 o valor 0.

Poderá a qualquer momento alterar novamente a configuração das entradas e saídas, bastando voltar a carregar no mesmo local.



Fig. 5.9 - Configuração de entradas e saídas

Quando acabar de configurar as entrada/saídas poderá começar a criar o programa, através da colocação dos vários blocos.

5.5 Colocação dos vários Blocos

Existem dois modos diferentes de inserir blocos na interface gráfica:

1- Por baixo de outro Bloco.

Todos os blocos podem ser colocados por baixo de outro bloco. Para introduzir um novo bloco deverá proceder do seguinte modo:

1 - Escolher na **Toolbox** o bloco de função a introduzir:



Fig. 5.10 - Toolbox

2 - Carregar com o botão do lado esquerdo do rato por baixo de um bloco já existente:

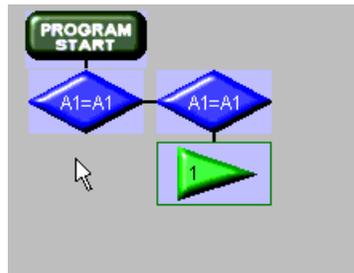


Fig. 5.11 - Colocação de bloco

3 - Configurar o menu da função e validar carregando em **OK** ou cancelar carregando em **Cancel**:

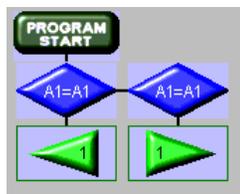


Fig. 5.12 - Validação de um bloco

2 - À direita de um bloco de comparação.

Só os blocos de comparação se podem introduzir deste modo. Para a sua inserção deverá proceder do seguinte modo:

1 - Escolher na **Toolbox** um dos blocos de comparação:



Fig. 5.13 - Toolbox

2 - Carregar com o botão do lado direito do rato sobre o bloco de comparação já existente. O novo bloco será posicionado imediatamente à direita do bloco de comparação anterior;

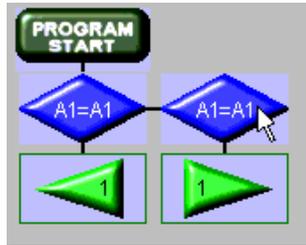


Fig. 5.14 - Colocação de um bloco

3 - Configurar o menu do bloco de comparação e validar carregando em **OK** ou cancelar carregando em **Cancel**:

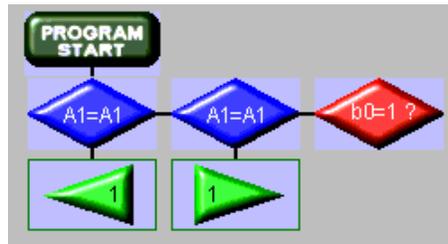


Fig. 5.15 - Validação de um bloco

nota1: No caso dos blocos de comparação com AND lógico só se pode introduzir um bloco à direita do primeiro bloco que forma o conjunto de comparações.

nota2: Se utilizar este procedimento num bloco que já tem um bloco de comparação ligado à sua direita, então vai criar um novo bloco entre estes dois.

Para poder distinguir entre blocos de comparação ligados e não ligados, basta verificar quais é que têm uma linha a ligá-los.

5.6 Editar/Apagar/Substituir/Copiar Blocos

Para editar ou apagar os blocos deverá posicionar o cursor do rato sobre o bloco e premir com o botão esquerdo do rato.

Aparecerá uma janela de controlo com as opções de apagar ou editar (fig 5.16).

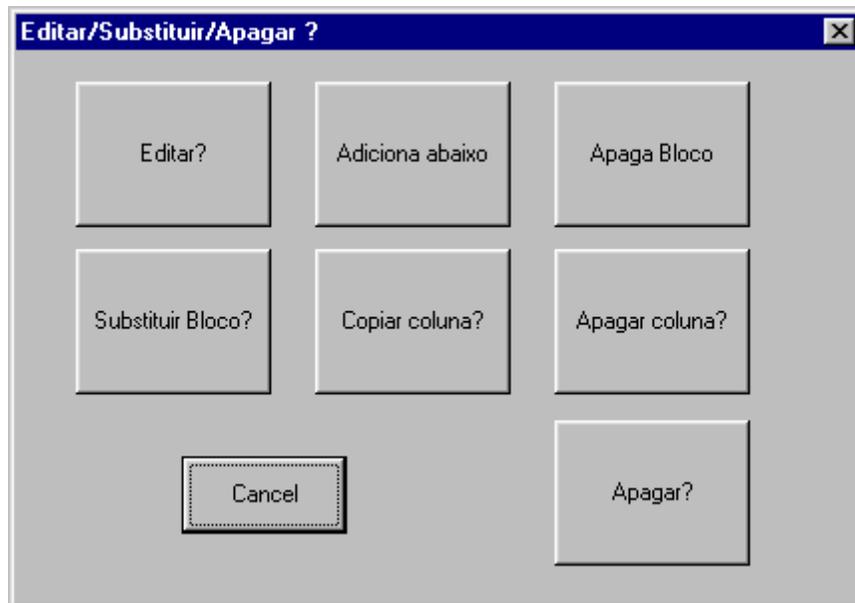


Fig. 5.16 - Janela de Controlo

Nesta janela é possível realizar as seguintes operações:

Operações de Alteração dos blocos:

- **Editar:** Permite alterar as definições de um bloco, sem alterar o tipo de bloco.
- **Substituir bloco:** Permite alterar o bloco existente por um novo bloco de outro tipo.

Operações de Introdução de blocos:

- **Adiciona abaixo:** Permite a introdução de um novo bloco entre dois blocos existentes pertencentes à mesma coluna.
- **Copia Coluna:** Permite copiar para a memória - *Clipboard* - um conjunto de blocos, para utilização posterior, no programa ou noutro programa.

Operações de remoção de blocos:

- **Apaga Bloco:** Permite a remoção de um bloco de qualquer zona do programa. Todas as suas dependências abaixo sobem um nível.
- **Apaga coluna:** Permite apagar uma coluna e as dependências abaixo do bloco seleccionado.
- **Apagar:** Apaga um bloco e todas as suas dependências abaixo e à sua direita.

5.6.1 Editar/Substituir Blocos

Para editar ou substituir os blocos, deverá posicionar o cursor do rato sobre o bloco e carregar com no botão esquerdo.

Aparecerá uma janela com várias opções, entre elas aparece a opção de Editar bloco ou substituir bloco.

Editar:

Caso opte por editar, poderá alterar os valores anteriormente definidos. Poderá validar carregando em OK ou cancelar carregando em CANCEL.

Substituir:

Caso opte por substituir o bloco existente, aparecerá uma janela (fig. 5.17) com os tipos de blocos que poderá utilizar. Nem sempre são apresentados todos os blocos, já que a substituição do bloco existente por um outro tem de seguir as regras definidas.

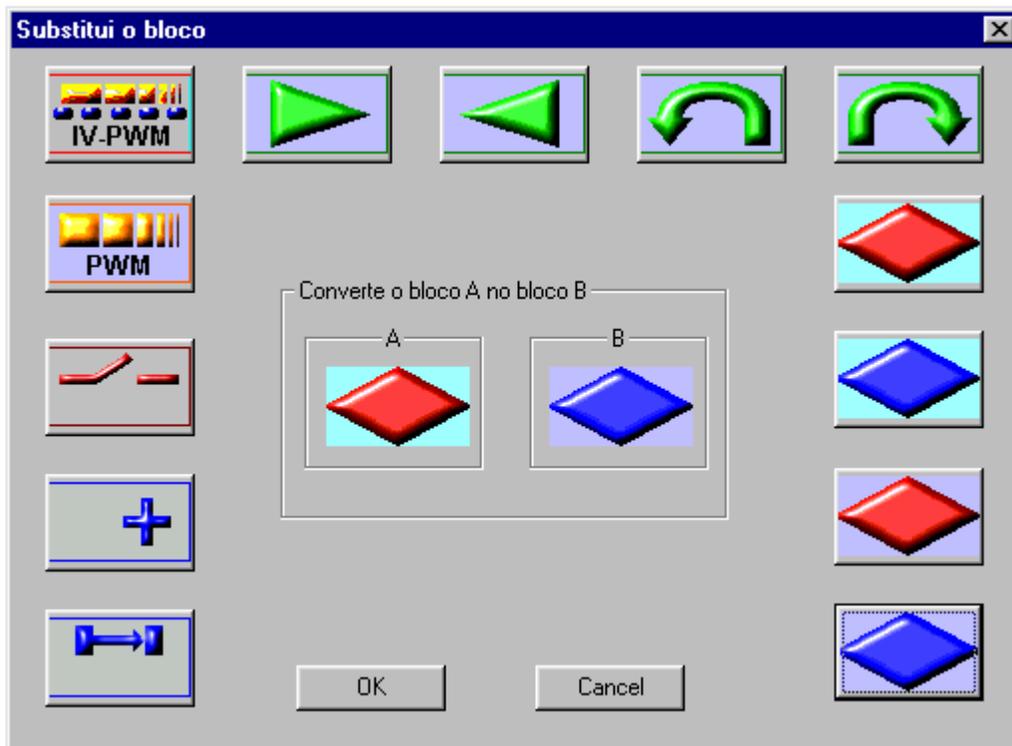


Fig. 5.17 - Janela de substituição de blocos

No centro da janela é apresentado o bloco que pretende substituir. Depois de escolher o novo bloco, seleccionando-o com o rato, este aparece também no centro da janela.

Poderá validar carregando em OK ou cancelar carregando em CANCEL.

De seguida aparecerá a janela de configuração do bloco.

Poderá validar carregando em OK ou cancelar carregando em CANCEL.

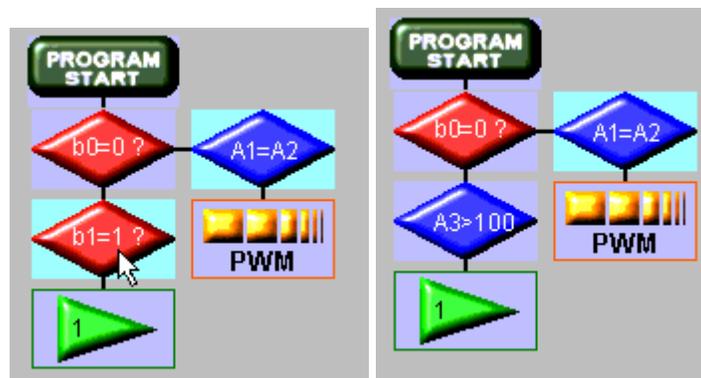


Fig. 5.18 - Substitui um bloco

5.6.2 Inserção de um bloco entre dois blocos

É possível introduzir um novo bloco entre dois blocos de uma coluna. Para a sua inserção deverá proceder do seguinte modo:

- 1 - Escolher na Toolbox o bloco que pretende introduzir;



Fig. 5.19 -Escolha de bloco a introduzir

- 2 - Carregar com o botão do lado esquerdo do rato sobre o bloco que se situa por cima do local onde deseja introduzir o novo bloco. Aparecerá uma janela de controlo, dando a possibilidade, entre outras, de introduzir por baixo. Escolhendo este controlo, o novo bloco será posicionado imediatamente por baixo do bloco anterior;

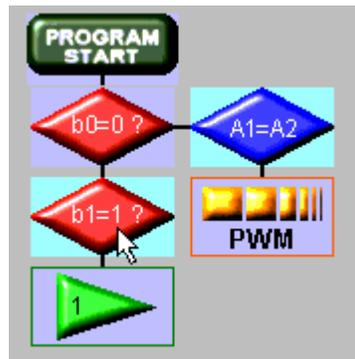


Fig. 5.20 - Introduz Bloco no meio

- 3 - Configurar o menu do bloco;
- 4 - Validar carregando em OK ou cancelar carregando em Cancel

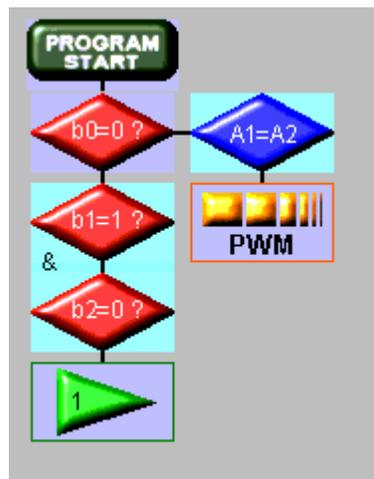


Fig. 5.21 - Resultado final

5.6.3 Copiar e inserir

Foi introduzida a possibilidade de copiar as colunas e suas dependências e inserir estas partes copiadas em outras zonas do programa ou em outros programas.

Para copiar um bloco e as suas dependências, deverá posicionar o cursor do rato em cima do primeiro bloco a copiar. Carregue com o botão do lado esquerdo do rato. Aparecerá uma janela de controlo. Entre as suas opções encontrará a opção de copiar. Selecione a opção Copiar coluna. Nesse momento, o bloco e todas as dependências abaixo são copiadas para o Clipboard.

Inserir por baixo

Para inserir os blocos copiados deverá posicionar o rato por baixo do último bloco da coluna, onde deseja inserir a coluna e as suas dependências.

Exemplo:

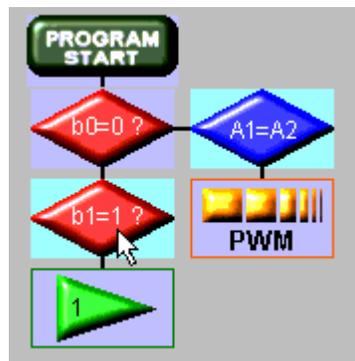


Fig. 5.21 - Cópia da coluna

Se o bloco escolhido for o $b0=0$, esse bloco e todos abaixo dele serão copiados para o Clipboard. Se desejar introduzir a informação copiada para baixo do bloco de PWM, então deverá colocar o ponteiro do rato por baixo desse bloco e carregar no botão do lado direito do rato. Nesse momento, se existir alguma informação no Clipboard, aparecerá uma janela de confirmação. Caso carregue em Paste, todo o conteúdo copiado é inserido por baixo do bloco de PWM.

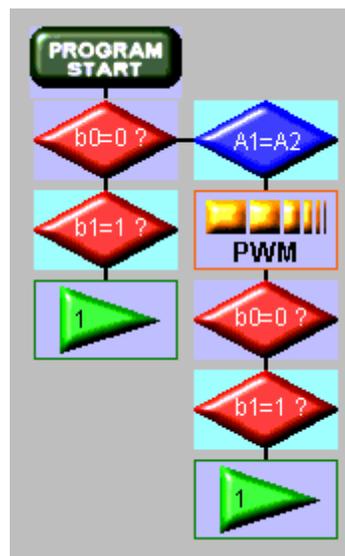


Fig. 5.22 - Inserir por baixo

Inserir no lado direito

Para inserir os blocos copiados deverá posicionar o rato em cima do bloco de comparação que ficará à esquerda do conjunto de blocos a serem inseridos. Se o primeiro elemento do conjunto copiado anteriormente for uma comparação, aparecerá uma janela de controlo. Carregue com o cursor do rato na função Paste Direita. O conjunto de blocos anteriormente copiados aparecerá no lado direito do bloco de comparação escolhido.

Inserir no lado esquerdo

Para inserir os blocos copiados deverá posicionar o rato em cima do bloco de comparação que ficará à direita do conjunto de blocos a serem inseridos. Se o primeiro elemento do conjunto copiado anteriormente for uma comparação, aparecerá uma janela de controlo. No caso de o bloco em que carregou ser um bloco de comparação sem qualquer ligação à sua esquerda, estará disponível na janela de controlo a possibilidade de Inserir à esquerda. Carregue com o cursor do rato na função Paste Esquerda. O conjunto de blocos anteriormente copiados aparecerá no lado esquerdo do bloco de comparação escolhido.

5.6.4 Copiar e inserir

Existem três tipos diferentes de funções para apagar blocos. É possível apagar um único bloco, apagar uma coluna e todas as suas dependências abaixo e ainda existe a possibilidade de apagar um bloco e todas as suas dependências abaixo e à direita.

Para qualquer uma destas funções deverá posicionar o rato sobre o primeiro bloco a apagar e carregar com o botão do lado esquerdo do rato. Aparecerá uma janela de controlo com as opções, entre outras, de apagar bloco, apagar coluna e apagar.

Apagar Bloco:

Se optar por apagar um bloco, o bloco escolhido será eliminado. Todos os blocos abaixo, e as suas dependências sobem uma posição.

1-Exemplo

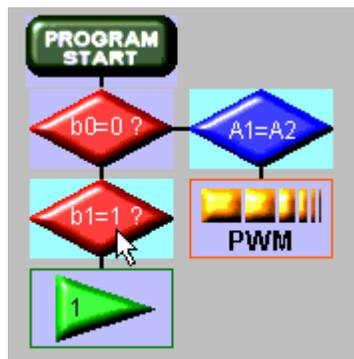


Fig. 5.23 - Escolher o bloco a apagar

Se o bloco escolhido for o $b0=0$ então o resultado é o seguinte:

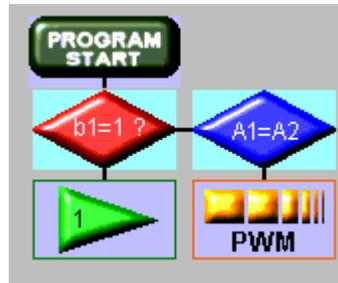


Fig. 5.24 - Resultado final

2-Exemplo

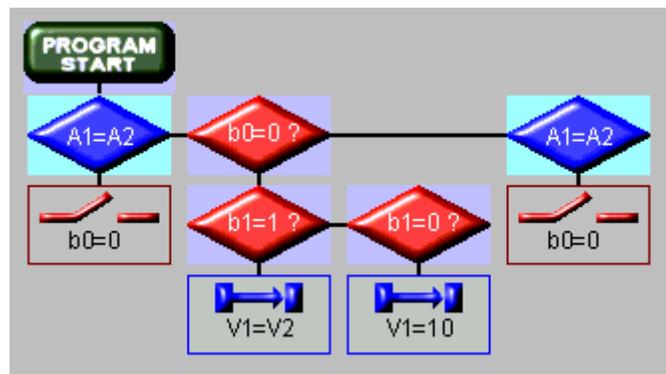


Fig. 5.25 - Escolher o bloco a apagar

Se o bloco escolhido for o $b0=0$ então o resultado é o seguinte.

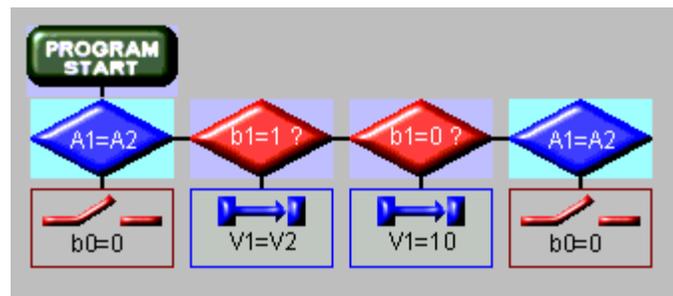


Fig. 5.26 - Resultado final

Existem algumas situações nas quais não se pode utilizar esta função. Sempre que uma destas situações ocorre aparece uma mensagem de erro que explica qual o problema encontrado.

Apagar Coluna:

Se optar por apagar a coluna, o bloco e todos os blocos que estejam abaixo deste serão eliminados.

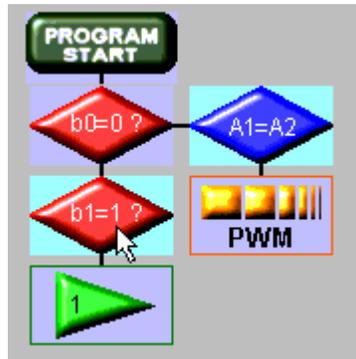


Fig. 5.27 - Escolher a coluna a apagar

Se o bloco escolhido for o $b0=0$ então o resultado é o seguinte.



Fig. 5.28 - Resultado final

Apagar:

Se optar por apagar, o bloco e todos os blocos que tenham dependência abaixo e à sua direita serão eliminados.

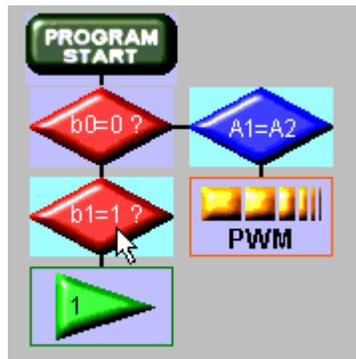


Fig. 5.29 - Escolher as ligações a apagar

5.7 Estrutura do Programa

O programa a criar deverá seguir sempre os seguintes princípios:

- 1 - O ciclo do programa será sempre de cima para baixo, i.e., o ciclo começa pelo bloco imediatamente abaixo do bloco de início do programa, PROGRAM START
- 2 - Caso não exista nenhuma comparação o bloco seguinte a ser executado é o bloco logo abaixo;
- 3 - Caso exista uma comparação, existem duas hipóteses:

3.1-Verifica-se a condição; Caso se verifique a condição imposta no bloco de comparação, o próximo

bloco a ser executado é o bloco logo abaixo deste.

3.2-*Não se verifica a condição*; Neste caso o bloco a ser executado é o bloco à direita da comparação, que será uma nova comparação ou poderá não existir. No caso de não existir mais nenhum bloco ao lado, então é dado como acabado este ciclo e é iniciado um novo.

4 - Quando não existirem mais blocos por baixo o ciclo acaba e é iniciado um novo ciclo.

nota: No caso da utilização de blocos de comparação AND, o procedimento é semelhante já que a utilização de conjuntos de dois ou mais blocos deste tipo correspondem efectivamente à utilização de apenas um bloco de comparação, só que utilizando mais elementos para essa comparação. Isto é, no caso de utilizar 5 blocos AND seguidos, se algumas das condições não se verificar, o próximo bloco a ser executado será aquele que se encontra à direita do primeiro elemento deste tipo (caso exista).

6 Tipos de Blocos

6.1 Blocos de Comparação Simples

Foram incluídos na interface gráfica dois blocos de comparação (figura 6.1).

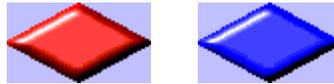


Fig. 6.1 - Blocos de Comparação

Um primeiro bloco permite ao utilizador fazer comparações binárias, isto é, permite verificar o estado lógico de uma variável. As variáveis que podem ser encontradas neste tipo de comparação são as seguintes: b0..b7, c0 e c3. Para poder utilizar estas variáveis como entradas, o utilizador deverá configurá-las no menu de configuração.

nota: É possível utilizar neste tipo de comparação os pinos configurados como entradas e os pinos configurados como saídas (figura 6.2).

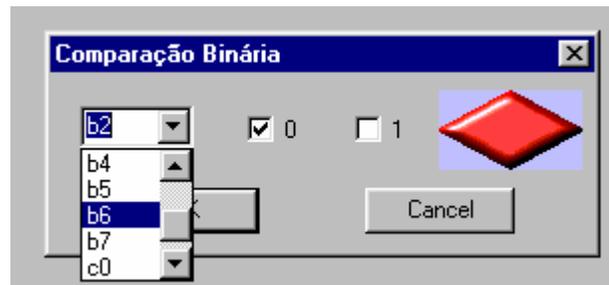


Fig. 6.2 - Comparação Binária

O segundo bloco permite ao utilizador fazer comparações de 8 bits (comparação de palavra). Isto é, podemos verificar se uma dada variável é menor, igual ou maior que outra, ou mesmo comparar uma variável com um número (figura 6.3). As variáveis de 8 bits que podem ser utilizadas nestas comparações são as seguintes: I0 a I6, A1, A2, A3 e A5 e V1 a V5.

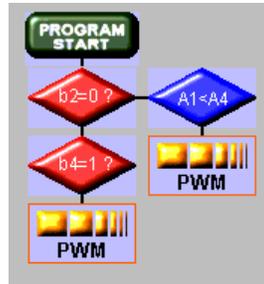


Fig. 6.3 - Comparação de números inteiros

Nota: I0 a I6 correspondem às 7 entradas multiplexadas ligadas à entrada A0 do microcontrolador.

6.2 Blocos de Comparação com AND Lógico

Este tipo de blocos assemelha-se aos blocos de comparação simples, mas com a particularidade de poder encadear vários conjuntos de blocos de comparação, permitindo assim existência de comparações com mais informação.



Caso 1:

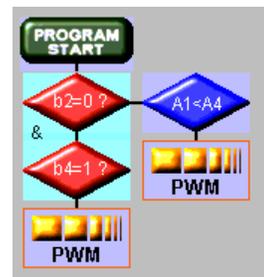


Fig. 6.4 - Caso 1

Na Figura 6.4 são apresentados dois programas. No lado esquerdo é apresentado um programa com três comparações simples e no lado direito é apresentado o mesmo programa mas com comparações com AND Lógico.

No primeiro caso o programa vai iniciar a sua execução com a verificação do bit b2. Se este estiver a 0 então o programa passa para o bloco seguinte de comparação, onde se verifica se b4 é igual a 1. É neste ponto que os dois tipos de blocos diferem. No caso dos blocos de comparação simples, se b4 for igual a 0, então, como não existe nenhum bloco de comparação ligado a este bloco, o programa vai recomeçar. No caso da utilização de um bloco de comparação com AND Lógico, caso b4 seja igual a 0, o programa não vai recomeçar, vai sim fazer o teste A1<A4?.

Caso 2:

Com este segundo caso (figura 6.5) mostra-se que podemos ter os dois tipos de comparação (com números inteiros e binários), a formarem o mesmo bloco de comparação AND.

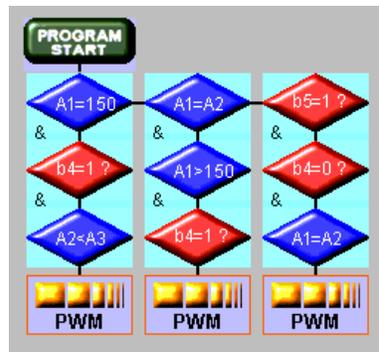


Fig. 6.5 - Caso 2

Como se pode ver na figura 6.5, a um bloco AND constituído por dois ou mais blocos de comparação, só é permitida uma única ligação directa à direita e esta estará sempre ligada ao primeiro bloco.

Caso 3:

Neste último caso (figura 6.6) mostra-se como é possível agrupar os vários blocos.

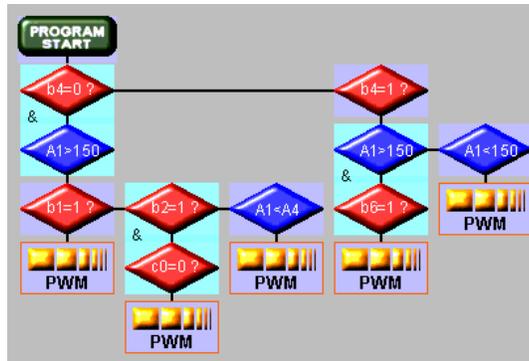


Fig. 6.6 - Caso 3

6.3 Blocos de Actuação de Variáveis Inteiras

Este tipo de blocos (figura 6.7), permite actuar variáveis inteiras.



Fig. 6.7 - Blocos de variáveis inteiras

O bloco que se encontra à esquerda permite ao utilizador atribuir um valor a uma das variáveis auxiliares V1 a V5.



Fig. 6.8 - Atribuição de valores às variáveis auxiliares

Existem dois tipos de valores que podem ser atribuídos a estas variáveis auxiliares.

- 1 Valores de outras variáveis inteiras, A1 a A4, I0 a I6 e V1 a V5;
- 2 Valores inteiros entre 0 e 255.

Os blocos que se encontram à direita permite ao utilizador incrementar ou decrementar o valor de uma das variáveis auxiliares V1 a V5.

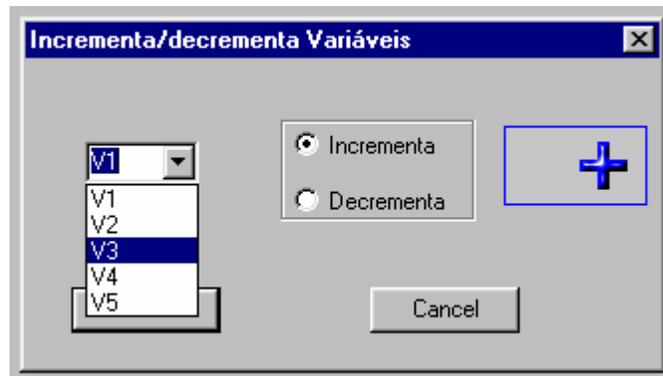


Fig. 6.9 - Incremento/Decremento de variáveis

Em conjunto estes blocos podem ser utilizados como contadores.

6.4 Bloco de Actuação Binária



Fig. 6.10 - Bloco de Actuação Binária

Este tipo de bloco, permite actuar uma saída digital.

É possível ligar ou desligar uma das seguintes saídas: b0 a b7, c0, c3 ou c7.



Fig. 6.11 - Set/Reset de saída digital

Para poder actuar a saída é necessário que esta esteja configurada como saída.

6.5 Bloco regulador de Velocidade



Fig. 6.12 - Bloco Regulador de Velocidade

Este bloco (figura 6.12) permite a regulação da velocidade de ambos os motores do robot, ou só de um, mantendo o outro.

O utilizador poderá escolher a velocidade dos motores numa escala de 255 valores (figura 6.13). Ao valor

médio dessa escala equivale a parar o motor, enquanto que nos extremos se verificam as velocidades mais elevadas para um lado ou para o outro.

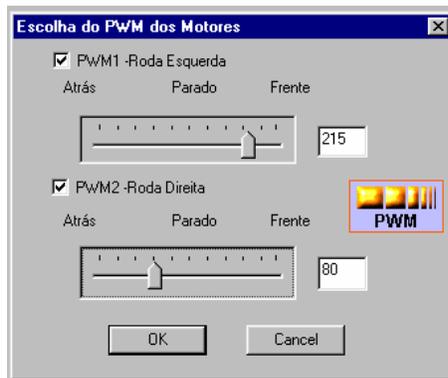


Fig. 6.13 - Escolha do PWM dos Motores

6.6 Bloco de Sensores de Infravermelho para actualização do PWM



Fig. 6.14 - Bloco de sensores IV

Este bloco (figura 6.14) permite relacionar as leituras dos sensores de IV existentes no robot, I0, I1, I2, I3, I4, I4, I5 e I6, com diferentes velocidades para cada uma das rodas.

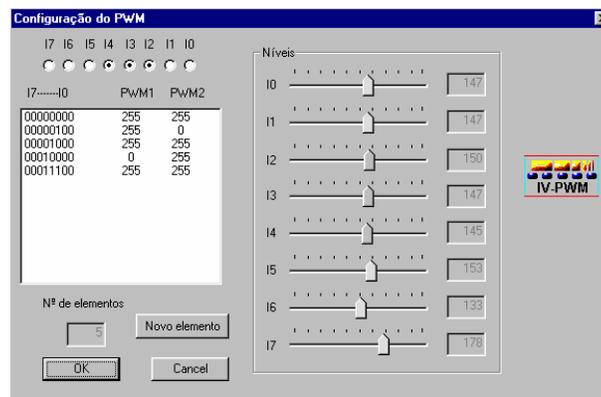


Fig. 6.15 - Configuração do PWM

No menu acima existem dois campos distintos.

6.6.1 Configuração dos Sensores a utilizar

No topo superior esquerdo desta janela é apresentado a configuração dos sensores que se pretende utilizar. Na figura é apresentada a configuração para a utilização de I2, I3 e I4. Para utilizar os outros sensores de infravermelhos, basta assinalar o sensor pretendido.

nota: todos os sensores são sempre lidos, mesmo que não estejam assinalados.

6.6.2 Configuração de NÍVEIS

O sinal recebido pelo microcontrolador, vindo de cada sensor, é um sinal analógico entre 0 e 5 V. Correspondendo o caso de 5V, à existência de uma superfície bastante reflectora, e os 0V à inexistência de reflexão. Geralmente as superfícies escuras não absorvem toda a luz. Reflectem sempre uma quantidade apreciável, por isso é necessário tentar determinar um nível entre 0 e 5 V que permita distinguir o branco do preto.

O microcontrolador contém um conversor Analógico/Digital incorporado que converte as tensões em valores entre 0 e 255, correspondendo o 0 a 0V e 255 a 5V. O utilizador terá de escolher o nível que permita ao robot distinguir o branco do preto. Só podem ser configurados os sensores que estejam assinalados.

Depois dos sensores escolhidos serem lidos, os valores são comparados com os níveis, sendo convertidos em binário, zero ou um, conforme estejam acima ou abaixo do nível.

Se o valor lido for superior ao valor do nível, o sensor estará sobre uma superfície reflectora, neste caso este bloco considera que o valor desse sensor é 0.

Se o valor lido for inferior ao valor do nível, o sensor estará sobre uma superfície não reflectora, neste caso este bloco considera que o valor desse sensor é 1.

6.6.3 Configurações de PWM

Depois dos sensores serem lidos e obtidos os valores lógicos, 0 branco e 1 preto, é possível tratar todas as situações, correspondendo à tabela lógica, até 8 bits. O utilizador poderá, para cada uma das situações, programar a velocidade dos dois motores.

Para introduzir um novo elemento na lista basta carregar na opção de introdução Novo Elemento. É aberta uma nova janela que permite seleccionar a nova configuração de infravermelhos, bem como os dois novos valores a serem enviados para os motores. Repare que só poderá adicionar elementos utilizado os sensores anteriormente assinalados.

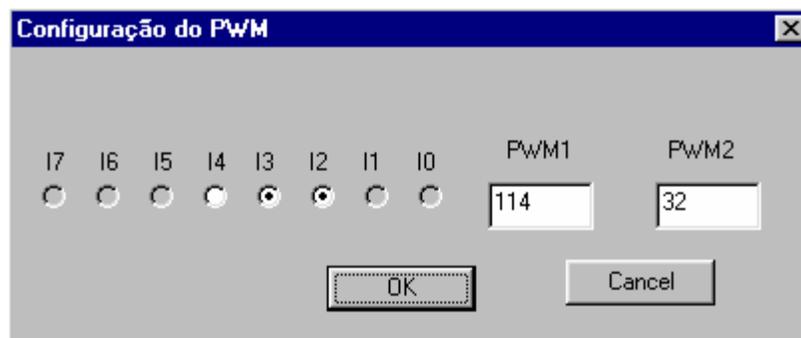


Fig. 6.16 - Escolha de um novo elemento da tabela

Para validar carregue em OK.

Um novo elemento será adicionado à lista. Esta lista de configurações encontra-se ordenada por ordem crescente. Sempre que um novo elemento é adicionado, este é colocado na ordem correcta.

Para editar ou retirar um elemento da lista, basta colocar o cursor do rato no elemento a alterar/apagar e carregar no seu botão do lado esquerdo.

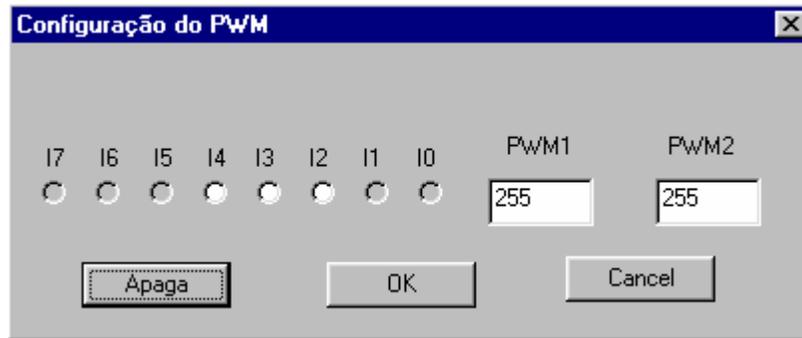


Fig. 6.17 - Apagar/editar

Aparece uma nova janela, com a configuração seleccionada sendo possível editar os dados ou mesmo apagar, bastando utilizar o botão de apagar.

No final a lista é novamente ordenada.

Quando acabar de alterar os dados da Lista basta carregar em OK para a sua validação.

6.6.4 Equivalência

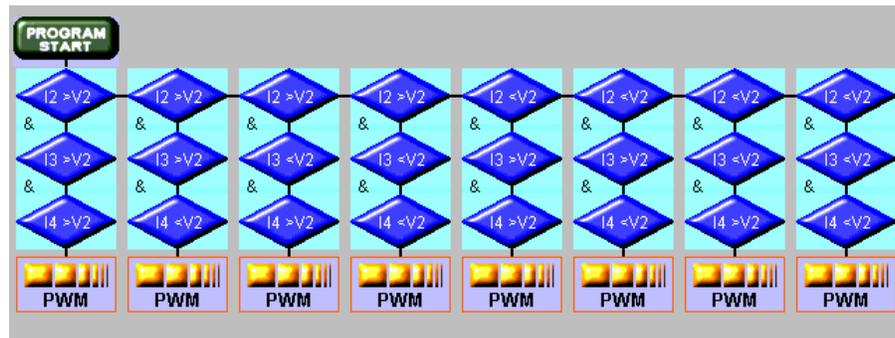


Fig. 6.18

O bloco acima descrito é equivalente ao conjunto de blocos da figura 6.18. Sendo $V2=150$ igual para todos os conjuntos de sensores.

6.7 Blocos para movimentos pré-definidos

Nesta interface gráfica foram incluídos 4 movimentos pré-definidos (figura 6.19).



Fig. 6.19 - Blocos para movimentos pré-definidos

O utilizador só terá de configurar o tempo de execução da manobra. O tempo encontra-se escalado em períodos de 0,25 segundos. O tempo definido deverá estar dentro do intervalo de 0 a 64 segundos (figura 6.20).

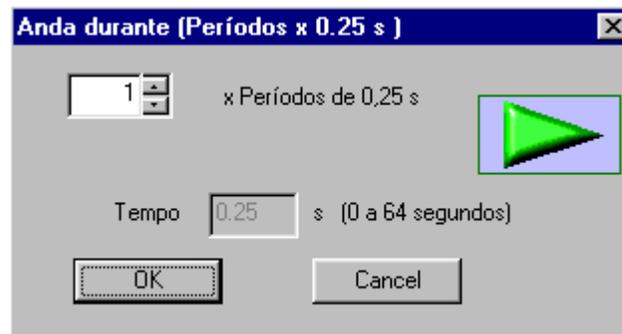


Fig. 6.20 - Movimento por períodos de 0,25 segundos

7 Como fazer o Download para o Robot?

Para fazer o Download do Programa deverá em primeiro lugar efectuar a sua compilação.

Para tal deverá carregar no botão de compilação existente na **Toolbar** (figura 7.1).

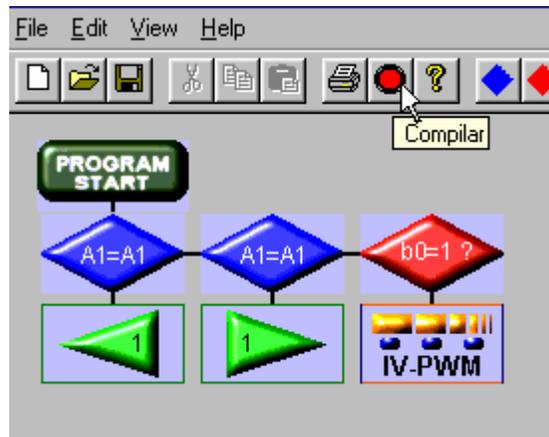


Fig. 7.1 - Toolbar

Depois de efectuada a compilação é automaticamente aberta a aplicação **LOADER**.

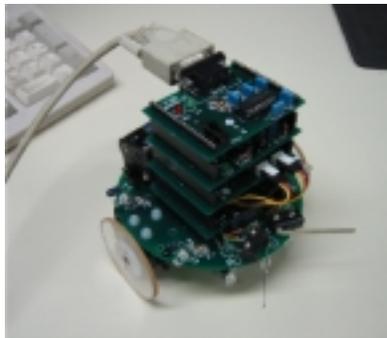


Fig. 7.2 - Ligação ao Robot

Nesta altura deverá montar a placa programadora ID-PROG002 no robot, encaixando-a, para tal, em cima da placa ID-PIC002. Ligue o terminal macho do cabo série à placa programadora e o terminal fêmea a uma entrada série do PC (figura 7.3).



Fig. 7.3 - Ligação do Robot ao PC

No programa **LOADER** deverá configurar a porta série que está a utilizar "Com1" ou "Com2".

Abra o ficheiro Prog.exe.



Fig. 7.4 - Prog.exe

Com o robot desligado (interruptor da placa ID-BASEIR002 "OFF"), ligue o interruptor da placa ID-PROG002 ("ON"). Passe agora o interruptor da placa ID-BASEIR002 a "ON".

Carregue em "Send" no programa Loader e verá o seu programa a ser enviado para o microcontrolador (ver figura 7.5).



Fig. 7.5 - Prog.exe a ser enviado

Quando aparecer a mensagem "Success" pode desligar o robot (interruptor da placa ID-BASEIR002 "OFF") e desligar o interruptor da placa programadora. Retire o cabo série da placa programadora.

Para pôr o robot em movimento basta agora ligar o interruptor da placa ID-BASEIR002. O robot poderá executar os programas com ou sem a placa programadora montada. Para os executar com a placa montada o interruptor desta deverá estar em "OFF". Tome em atenção que a execução dos programas com a placa programadora montada leva a um pequeno aumento de consumo energético.

7.1 Mensagens de erro

Aquando do envio do programa para o robot poderão ocorrer erros devidos a procedimentos incorrectos. De seguida são apresentados os erros, bem como as causas mais prováveis.

Erro: "ERROR: Serial Port Selected not Enable!!!"

Causa: A porta série escolhida já tem um periférico ligado (rato)

Solução: Escolher a outra "com" no programa

Erro: "ERROR:The Pic does not respond "

Causa: O robot não está ligado

O interruptor da placa controladora não está ligado

A porta série associada não é a que está definida

Solução: Com o robot desligado (interruptor da placa ID-BASEIR001 "OFF"), ligue o interruptor da placa ID-PROG001 ("ON"). Passe agora o interruptor da placa ID-BASEIR002 a "ON".

Verificar a porta série ou escolher a outra "com" no programa

Erro: "ERROR: The Pic can't be programmed "

Causa: Os dados vindos do computador não estão correctos

O Pic não consegue processar os dados enviados pelo computador.

Solução: Voltar a compilar os dados

Aumentar o "Delay" no programa

Erro: "ERROR: Stop by USER !!!"

Causa: O utilizador parou o download através do botão "Stop"

Solução: Voltar a enviar o programa "Send"

8 Exemplos

Exemplo 1: Programa de teste dos interruptores 1

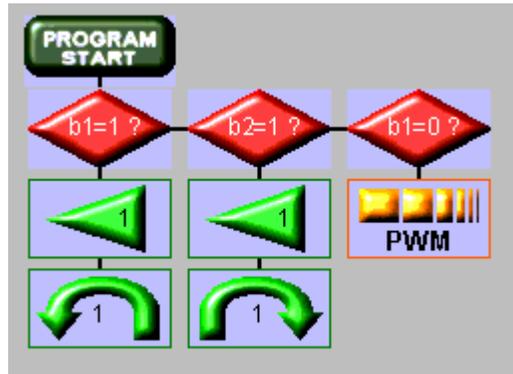


Fig. 8.1 - Programa de teste dos interruptores 1

Este programa, exemplo1.grf, é um simples programa onde se mostra como se podem utilizar os blocos de comparação simples, os blocos de movimentos prédefinidos e ainda como se pode utilizar os blocos de regulação de velocidade. Para poder utilizar este programa o utilizador necessita de ter ligado os dois interruptores aos pinos 1 e 2 do PORTB.

Explicação:

1-O interruptor que se encontra ligado ao pino b1 vai ser testado. Caso este esteja pressionado, o robot vai andar durante 0,25 s para trás e depois fará, durante 0,25 s, uma rotação para a esquerda após a qual vai voltar a fazer o teste ao pino b1. Caso não esteja pressionado, vai testar o bloco à sua direita (2).

2-O interruptor que se encontra ligado ao pino b2 vai ser testado. Caso este esteja pressionado, o robot vai andar durante 0,25 s para trás e depois fará, durante 0,25 s uma rotação para a direita, após a qual irá voltar para o início do programa. Caso não esteja pressionado, vai testar o bloco à sua direita(3).

3-O interruptor que se encontra ligado ao pino b2 vai ser testado. Caso este não esteja pressionado (é sempre necessário uma condição), vai utilizar as velocidades definidas no bloco de PWM. Depois de executar o último bloco, o programa volta ao início (1).

Utilize este programa para ficar a conhecer estes três blocos, alterando as velocidades dos motores e tempos dos blocos.

Exemplo 2: Programa de teste dos interruptores 2

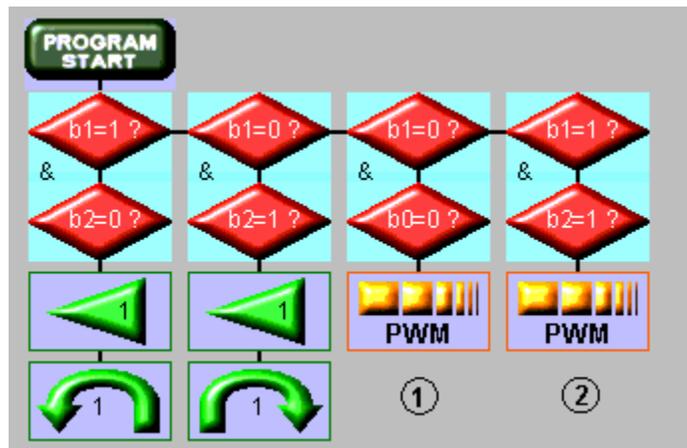


Fig. 8.2 - Programa de teste dos interruptores 2

Este programa, exemplo2.grf, demonstra a vantagem de, em certos casos, utilizar os blocos de comparação com AND Lógico. Para poder utilizar este programa o utilizador necessita de ter ligado os dois interruptores aos pinos 1 e 2 do PORTB.

Explicação:

1-Os dois interruptores vão ser testados. No caso do interruptor que se encontra ligado a b1 estiver a 1 (pressionado) e o interruptor que se encontra ligado a b2 estiver a 0 (desligado) então o robot vai andar durante 0,25 s para trás e depois fará, durante 0,25 s, uma rotação para a esquerda, após a qual irá voltar para o início do programa. Caso não se verifiquem as duas condições o programa vai executar o bloco AND à sua direita(2).

2-O segundo bloco é simétrico ao primeiro. Caso as condições se verifiquem (b1=0 e b2=1), o robot vai andar durante 0,25 s para trás e depois fará, durante 0,25 s, uma rotação para a direita, após a qual irá voltar para o início do programa. Caso não se verifiquem as duas condições o programa vai executar o bloco AND à sua direita(3).

3-O terceiro bloco AND vai testar se estão os dois interruptores desligados. Se estiverem então é atribuído um valor de PWM (1), que fará o robot andar para a frente e o programa volta ao início. Caso não se verifiquem as duas condições, o programa vai executar o bloco AND à sua direita(4).

4-O quarto e último bloco AND vai testar se os dois interruptores estão pressionados. Se estiverem, é atribuído um valor de PWM (2), que fará o robot parar. Depois de executar o último bloco, o programa volta ao início (1).

Exemplo 3: Programa de seguimento de uma linha 1



Fig. 8.3 - Programa de seguimento de uma linha 1

Com este programa, exemplo3.grf, pretende-se mostrar como se pode utilizar o bloco de sensores de infravermelhos para actualização do PWM, para seguir uma linha preta no chão branco.

Só é necessário utilizar um único bloco, bastando configurar a velocidade dos dois motores e o nível que define a separação entre o branco e o preto. Bastando para isso configurar o quadro que se segue na figura 8.4.

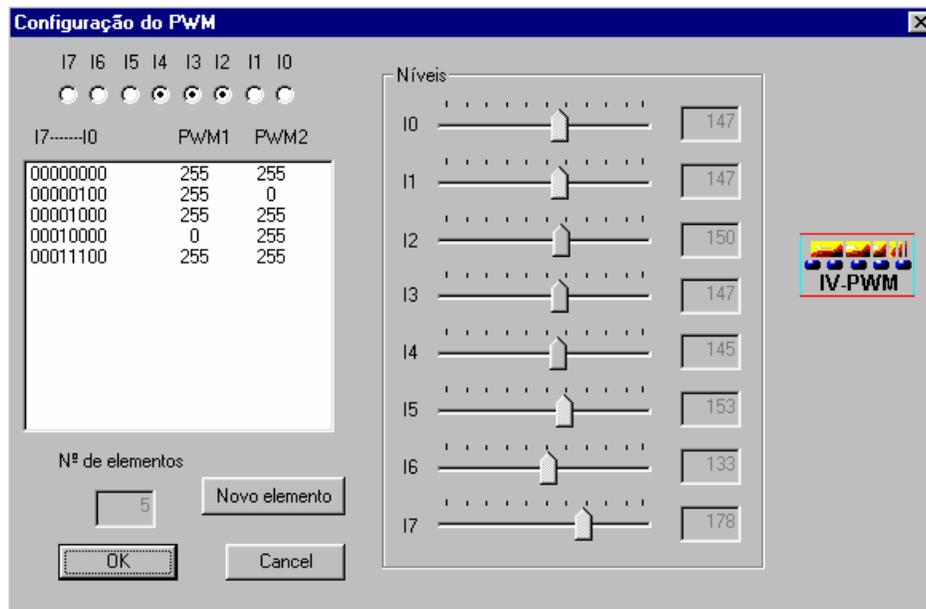


Fig. 8.4 - Configuração do PWM

nota: utilizando só este bloco o utilizador fica limitado aos valores que são introduzidos no quadro.

Exemplo 4: Programa de seguimento de uma linha 2

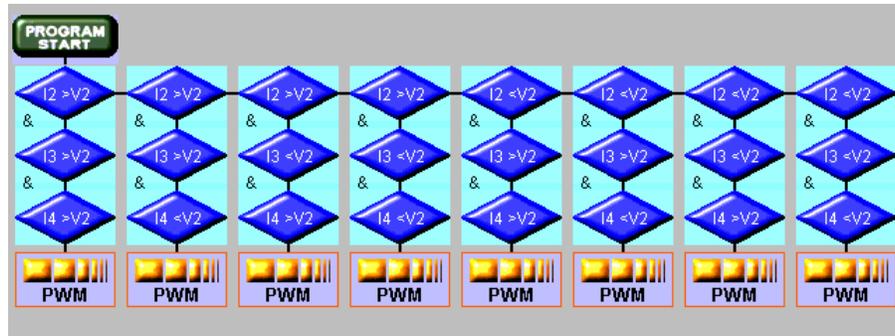


Fig. 8.5 - Programa de seguimento de uma linha 2

Com este programa, exemplo4.grf, mostra-se como se pode utilizar conjuntos de blocos de comparação com AND Lógico para realizar a mesma tarefa do Bloco de sensores de infravermelhos para actualização do PWM, para seguir uma linha preta no chão branco.

Como se pode ver na figura 8.6 por cada condição existente no bloco de sensores de infravermelho são utilizados 3 comparações com o nível V1 dentro de um AND Lógico e utilizado um bloco de regulação de velocidade.

Verifique que V1 está inicializado com o valor 150 na janela de configuração.

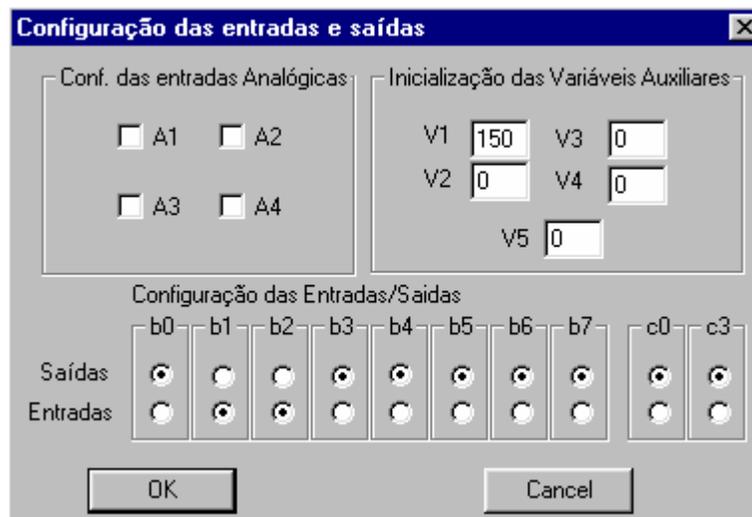


Fig. 8.6 - Configuração das entradas e saídas

Poderá ter níveis diferentes para cada um dos sensores, bastando para isso, fazer a comparação de I2 com V1, I3 com V2 e I4 com V3. Não esquecer de fazer a inicialização dos valores de V1, V2 e V3 na janela de configuração.

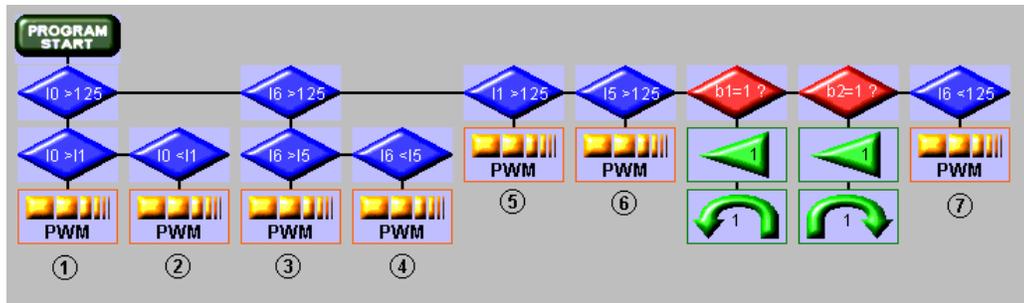
Exemplo 5: Programa robot evita obstáculos 1

Fig. 8.7 - Programa robot evita obstáculos 1

No programa exemplo5.grf, demonstra-se como se pode programar o robot para evitar os obstáculos que se encontram à sua volta. Utiliza os dois sensores de contacto (interruptores ligados a b1 e b2) e os 4 infravermelhos laterais (I0, I1, I5 e I6).

Explicação:

1-O robot testa o sensor I0, verificando a proximidade de um obstáculo, (reflexão superior a 125), se o valor lido no sensor for superior a 125, o programa vai verificar a existência de um obstáculo próximo de I1 (caso exista deve-se dar prioridade a este sensor), se $I0 > I1$ então as velocidades são as seguintes PWM (1) esquerda=255 e direita=255 (anda em frente), caso $I0 < I1$ PWM(2) esquerda=0 e direita=255 (roda para a sua esquerda);

2-O robot testa o sensor I6, verificando a proximidade de um obstáculo, (reflexão superior a 125), se o valor lido no sensor for superior a 125, o programa vai verificar a existência de um obstáculo próximo de I5 (caso exista deve-se dar prioridade a este sensor), se $I6 > I5$ então as velocidades são as seguintes PWM (3) esquerda=255 e direita=255 (anda em frente), caso $I6 < I5$ PWM(4) esquerda=255 e direita=0 (roda para a sua direita);

3- O robot testa o sensor I1, verificando a proximidade de um obstáculo, (reflexão superior a 125), se o valor lido no sensor for superior a 125 então as velocidades são as seguintes: PWM(5) esquerda=0 e direita=255 (roda para a sua esquerda);

4- O robot testa o sensor I5, verificando a proximidade de um obstáculo, (reflexão superior a 125), se o valor lido no sensor for superior a 125 então as velocidades são as seguintes: PWM(6) esquerda=255 e direita=0 (roda para a sua direita);

5- Caso não exista nenhum obstáculo colocado lateralmente ao robot, então são testados os sensores de contacto começando por testar o sensor ligado a b1. Caso este esteja em contacto com algum obstáculo então a rotina a ser executada é uma manobra de fuga, anda para trás e vira à esquerda durante 0,5, para ambos os blocos;

6- De seguida testa o interruptor ligado a b2, caso este esteja em contacto com algum obstáculo então a rotina a ser executada é uma manobra de fuga, anda para trás e vira à direita durante 0,5, para ambos os blocos;

7-Caso não existam obstáculos em redor do robot então este vai andar em frente.

Exemplo 6: Programa robot evita obstáculos 2

No programa exemplo6.grf, demonstra-se como se pode programar o robot para evitar os obstáculos que se encontram à sua volta, utilizando os vários tipos de blocos de comparação. Utiliza os dois sensores de contacto (interruptores ligados a b1 e b2) e os 4 infravermelhos laterais (I0, I1, I5 e I6).

Verifique que com a utilização dos vários blocos de comparação (simples e AND lógico) é possível acrescentar ao robot uma maior flexibilidade, tanto ao nível de movimento como ao nível de programação.

Exemplo 7: Programa sequência de movimentos

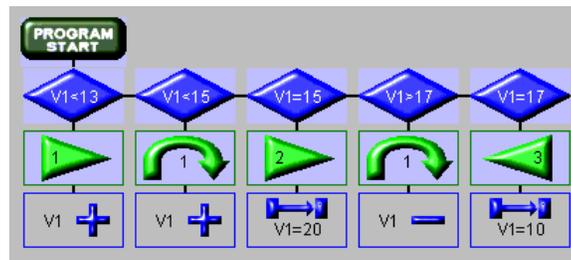


Fig. 8.8 - Programa sequência de movimentos

O exemplo7.grf, mostra como se podem utilizar as variáveis de auxiliares como contadores que são utilizados para desencadear novos movimentos. São aplicadas comparações de variáveis auxiliares com números inteiros, atribuições de números de 0 a 255 a essas variáveis e incrementos e decrementos dessas mesmas variáveis. A composição destes vários elementos vai permitir gerar uma sequência de movimentos.

Exemplo 8: Programa exemplo final

O último exemplo, exemplo8.grf, integra parte das funcionalidades da interface gráfica, incluindo a utilização de um sensor analógico ligado à entrada analógica A1. Verifique na configuração que a leitura da entrada analógica A1 se encontra activa. Este programa reúne o seguimento de uma linha no chão, com o programa que permite evitar obstáculos.

FIM