



UNIVERSIDADE
de AVEIRO

Departamento de Engenharia Mecânica

Relatório Final:

ATLAS II

*Desenvolvimento de um Robot Móvel para
Participação no Festival Robótica 2004,
Classe Condução Autónoma*

Orientador | Prof. Dr. Vítor Santos

Miguel Oliveira | N.º 16664

Miguel Neta | N.º 20256

Aveiro | Julho 2004

Resumo

O projecto tinha como objectivo principal o desenvolvimento de um robot autónomo e móvel capaz de competir na classe de condução autónoma do festival Robótica 2004. Este evento decorreu no Porto, nos dias 23 a 25 de Abril de 2004. A competição apresenta desafios tecnológicos muito interessantes, cuja resolução contribui para o desenvolvimento académico dos participantes. De referir que o ATLAS II, nome dado ao robot, foi este ano desenvolvido baseado na plataforma concretizada pelos alunos José Luís Silva e José Miguel Gomes no ano de 2003.

O relatório inicia com uma introdução a alguns assuntos relevantes, fazendo depois uma descrição dos objectivos a cumprir. Depois apresenta-se o plano de intervenção elaborado no início do ano lectivo. Segue-se uma descrição do robot, do software desenvolvido, terminando com uma avaliação final dos resultados obtidos e das soluções aplicadas.

Agradecimentos

Queríamos agradecer ao nosso orientador, **Professor Vítor Santos**, pelo apoio constante prestado ao longo de todo o projecto e pelo ânimo dado nos três dias muito difíceis do Festival.

Gostaríamos também de agradecer a várias pessoas pelo apoio prestado durante todo este ano de trabalho:

Eng. Camilo Christo,

Eng. António Festas,

Eng. André Quintã,

Luís Clara Gomes,

A todos os nossos colegas,

Gostaríamos ainda de agradecer especialmente aos colegas que iniciaram o projecto ATLAS no ano de 2003, **José Luís Silva** e **José Miguel Gomes**.

Índice

1	INTRODUÇÃO GERAL.....	8
2	CLASSE CA – OBJECTIVOS.....	10
3	PLANO DE INTERVENÇÃO	13
3.1	SISTEMA OPERATIVO E AMBIENTE DE PROGRAMAÇÃO	13
3.2	MICROCONTROLADOR.....	13
3.3	UTILIZAÇÃO DE DUAS CÂMARAS PARA SEGUIMENTO DE PISTA.....	14
3.4	UTILIZAÇÃO DE UMA CÂMARA PARA IDENTIFICAÇÃO DO SINALIZADOR.....	14
4	COMPONENTES DO ATLAS II.....	15
4.1	ALIMENTAÇÃO.....	15
4.2	BOTÃO DE EMERGÊNCIA.....	16
4.3	SENSORES	17
4.3.1	<i>Sensores Infravermelhos Túnel.....</i>	<i>17</i>
4.3.2	<i>Sensor Infravermelhos Passadeira.....</i>	<i>17</i>
4.3.3	<i>Câmaras Utilizadas.....</i>	<i>18</i>
4.4	MOTORES.....	19
4.4.1	<i>Tracção.....</i>	<i>19</i>
4.4.2	<i>Direcção.....</i>	<i>19</i>
4.5	PLACA DE REDE SEM FIOS.....	20
4.6	COMPONENTES ELECTRÓNICOS.....	20
4.6.1	<i>Microcontrolador.....</i>	<i>20</i>
4.6.2	<i>Multiplexer Analógico HEF4067B.....</i>	<i>21</i>
4.6.3	<i>Contador Binário HEF4040B.....</i>	<i>23</i>
4.6.4	<i>L297 e L298.....</i>	<i>23</i>
4.7	PLACAS ELECTRÓNICAS.....	25
4.8	DISPOSIÇÃO DOS COMPONENTES NA ESTRUTURA DO ROBOT.....	28
5	CONTROLO DOS MOTORES.....	30
5.1.1	<i>Ondas Quadradas – PWM.....</i>	<i>30</i>
5.1.2	<i>Controlo da Tracção.....</i>	<i>30</i>
5.1.3	<i>Controlo da Direcção.....</i>	<i>31</i>
6	SOFTWARE DESENVOLVIDO	32
6.1	BAIXO NÍVEL – MICROCONTROLADOR.....	32
6.1.1	<i>Programação do PIC.....</i>	<i>32</i>
6.1.2	<i>Comunicação Série.....</i>	<i>33</i>
6.1.2.1	<i>Comunicação PC→PIC.....</i>	<i>34</i>
6.1.2.2	<i>Comunicação PIC→PC.....</i>	<i>34</i>
6.1.3	<i>Funcionamento Global do PIC.....</i>	<i>35</i>
6.2	ALTO NÍVEL – PROCESSADOR CENTRAL.....	37
6.2.1	<i>Interpretação dos Sensores – Lookup Table.....</i>	<i>37</i>
6.2.2	<i>Contagem do número de Passagens na Passadeira.....</i>	<i>38</i>
6.2.3	<i>Navegação no Túnel / Zona de Obras.....</i>	<i>39</i>
6.2.4	<i>Processamento da Imagem do Semáforo.....</i>	<i>40</i>
6.2.4.1	<i>Modelos de Cor.....</i>	<i>41</i>
6.2.4.2	<i>Conversão RGB para HSV.....</i>	<i>41</i>
6.2.4.3	<i>Identificação de Cor.....</i>	<i>43</i>
6.2.5	<i>Navegação na Pista.....</i>	<i>47</i>
6.2.5.1	<i>Colocação das Câmaras.....</i>	<i>47</i>
6.2.5.2	<i>Configuração das Câmaras.....</i>	<i>50</i>
6.2.5.3	<i>Busca de Linha.....</i>	<i>51</i>

6.2.5.4	Definição de Postura Ideal na Pista	52
6.2.5.5	Busca da Postura Ideal	53
6.2.5.6	Orientação da Linha	56
6.2.5.7	Parâmetros de Validação	56
6.2.6	<i>Perspectiva Global de Funcionamento do Algoritmo Principal</i>	60
7	PARTICIPAÇÃO NO ROBÓTICA 2004	62
8	CONCLUSÃO	63
8.1	SOLUÇÃO MECÂNICA DE MOBILIDADE	63
8.2	ALTERAÇÃO DO SISTEMA OPERATIVO / PROGRAMAÇÃO EM C.....	63
8.3	MICROCONTROLADOR.....	63
8.4	DUAS CÂMARAS PARA SEGUIMENTO DE PISTA.....	64
8.5	CONFIGURAÇÃO DAS CÂMARAS	64
8.6	CÂMARA PARA ANÁLISE DE SEMÁFOROS.....	65
8.7	SENSORES INFRAVERMELHOS	65
8.8	DIFICULDADES ENCONTRADAS.....	65
8.9	CONTINUIDADE DO PROJECTO	67
9	BIBLIOGRAFIA	68

Índice de Figuras

FIGURA 1. LOGÓTIPO DO FESTIVAL DE 2004	8
FIGURA 2. FORMATO DA PISTA. A) VISTA ISOMÉTRICA. B) VISTA DE TOPO	10
FIGURA 3. SÍMBOLOS MOSTRADOS PELO SEMÁFORO	11
FIGURA 4. ZONA DE OBRAS	12
FIGURA 5. O ROBOT ATLAS II	15
FIGURA 6. BATERIAS DO ATLAS II	16
FIGURA 7. BOTÃO DE EMERGÊNCIA	16
FIGURA 8. SENSOR DE INFRAVERMELHOS SHARP GP2D12	17
FIGURA 9. SENSOR INFRAVERMELHOS DIGITAL SICK WT170-P112	18
FIGURA 10. CÂMARAS USB UTILIZADAS	18
FIGURA 11 - A) MOTOR MAXON RE040. B) ENCODER	19
FIGURA 12. A) MOTOR DE TRACÇÃO. B) POTENCIÓMETRO	20
FIGURA 13. ESQUEMA DO PIC	21
FIGURA 14. RÉGUA DE SENSORES	22
FIGURA 15. MULTIPLEXER ANALÓGICO DE 16 CANAIS	22
FIGURA 16. CONTADOR BINÁRIO HEF4040B	23
FIGURA 17. DISSIPADOR	24
FIGURA 18. L297	24
FIGURA 19. L298	25
FIGURA 20. PLACA PRINCIPAL, ESQUEMA 1	26
FIGURA 21. PLACA PRINCIPAL, ESQUEMA 2	27
FIGURA 22. PLACA DE CONTROLO DA DIRECÇÃO	28
FIGURA 23. DISPOSIÇÃO DOS COMPONENTES NA ESTRUTURA	29
FIGURA 24. PWM. ONDA QUADRADA DE DUTYCYCLE VARIÁVEL	30
FIGURA 25. SOFTWARE USADO NA PROGRAMAÇÃO DO PIC	33
FIGURA 26. FUNCIONAMENTO GLOBAL DO PIC	36
FIGURA 27. PINOS UTILIZADOS NO ROBOT	37
FIGURA 28. CURVA DE RESPOSTA DOS SENSORES DE INFRAVERMELHOS	38
FIGURA 29. A) MODELO RGG. B) MODELO HSV	41
FIGURA 30. ALGORITMO DE CONVERSÃO RGB PARA HSV	42
FIGURA 31. SIGNIFICADO DO VALOR DA COMPONENTE HUE	42
FIGURAS 32. CORES ELEMENTARES. A) VERMELHO. B) VERDE. C) AZUL	43
FIGURAS 33. HISTOGRAMA RGB DA IMAGEM: A) 33.A. B) 33.B. C) 33.C)	44
FIGURAS 34. HISTOGRAMA HSV DA IMAGEM: A) 33.A. B) 33.B. C) 33.C)	45
FIGURAS 35. SÍMBOLOS PARA: A) VIRAR À ESQUERDA. B) SEGUIR EM FRENTE. C) PARAR	46
FIGURAS 36. HISTOGRAMA RGB DA IMAGEM: A) 36.A. B) 36.B. C) 36.C)	46
FIGURAS 37. HISTOGRAMA HSV DA IMAGEM: A) 36.A. B) 36.B. C) 36.C)	46
FIGURA 38. SUPORTE PARA CÂMARAS	48
FIGURA 39. CONFIGURAÇÃO A 30. VISTA DE TOPO	49
FIGURA 40. CONFIGURAÇÃO A -60°. A) VISTA DE FRENTE. B) VISTA DE TOPO	49
FIGURA 41. CONFIGURAÇÃO A 90°. A) VISTA DE FRENTE. B) VISTA DE TOPO	50
FIGURA 42. AFINAÇÃO DE CÂMARA COM O CAMSTREAM	51
FIGURA 43. IMAGEM OBTIDA PELA CÂMARA ESQUERDA	52
FIGURA 44. IMAGEM BINARIZADA	52
FIGURA 45. POSTURA IDEAL	53
FIGURA 46. POSTURA DEFEITUOSA – LOCALIZAÇÃO	54
FIGURA 47. LOCALIZAÇÃO DEFEITUOSA. A) IMAGEM NORMAL. B) IMAGEM BINARIZADA	54
FIGURA 48. POSTURA DEFEITUOSA – ORIENTAÇÃO	55
FIGURA 49. ORIENTAÇÃO DEFEITUOSA. A) IMAGEM NORMAL. B) IMAGEM BINARIZADA	55
FIGURA 50. LINHA VIRTUAL	56
FIGURA 51. ENTRADA NA BIFURCAÇÃO	57
FIGURA 52. BIFURCAÇÃO. A) IMAGEM NORMAL. B) IMAGEM BINARIZADA	57
FIGURA 53. SITUAÇÃO DE PASSAGEM NO PARQUE	58
FIGURA 54. PASSAGEM NO PARQUE. A) IMAGEM NORMAL. B) IMAGEM BINARIZADA	58
FIGURA 55. ESQUEMA DO FUNCIONAMENTO DO ALGORITMO DE NAVEGAÇÃO	61
FIGURA 56. PISTA DE TESTES	66

Índice de Tabelas

<i>TABELA 1. FUNÇÕES DOS PINOS DO L297.</i>	25
<i>TABELA 2. FUNÇÕES DOS PINOS DO L298.</i>	25
<i>TABELA 3. TABELA DE COMUNICAÇÕES PARA A TRACÇÃO.</i>	34
<i>TABELA 4. TABELA DE COMUNICAÇÕES PARA A DIRECÇÃO.</i>	34
<i>TABELA 5. MEIO DE IDENTIFICAÇÃO DOS SÍMBOLOS DO SEMÁFORO.</i>	40
<i>TABELA 6. DIFERENÇAS ENTRE O MODELO RGB E HSV.</i>	43
<i>TABELA 7. CLASSIFICAÇÕES PARCIAIS E FINAL.</i>	62

1 INTRODUÇÃO GERAL

O projecto ATLAS II tem como objectivo primordial a participação na quarta edição do Festival Nacional de Robótica. Este festival engloba várias actividades que decorrem durante um fim-de-semana, incluindo encontros científicos, competições, conferências de imprensa entre muitas outras. O Festival pretende promover o gosto pela robótica entre os mais jovens ao mesmo tempo que funciona como catalisador desta área também a nível da investigação. O Festival Robótica 2004 decorreu este ano na cidade do Porto, mais concretamente no pavilhão Rosa Mota, no fim-de-semana de 23 a 25 de Abril.



Figura 1. Logótipo do Festival de 2004

As competições são várias, de entre as quais se destaca o futebol robótico, classe que também é disputada no evento *Robocup*, a decorrer este ano em Lisboa, de 5 a 25 de Julho. O Robocup tem uma dimensão internacional, coisa que não acontece com o Festival Robótica. Este último é, no entanto, o expoente máximo da área a nível nacional. Existem ainda outras competições, de nível universitário ou com equipas vindas de vários liceus do país. A competição onde se pretende que o ATLAS II participe é a chamada condução autónoma (CA). O objectivo desta, é percorrer uma pista de trajecto conhecido, no menor tempo possível, incorrendo no menor número de penalizações. Mais, o robot deve fazê-lo autonomamente, ou seja, utilizando um algoritmo de navegação que reage às condições momentâneas da pista. Aparentemente simples os desafios impostos nesta classe são, na verdade, tecnologicamente elaborados, dado que a robótica autónoma e móvel está ainda num estado embrionário e tarefas consideradas simples como estas são muitas vezes de difícil execução por um robot.

O carácter multidisciplinar deste projecto é bastante aliciente, dado que é necessário aprender matérias tão díspares como electrónica, linguagens de programação, uso de

microcontroladores, planeamento de trajectórias, controlo de motores, processamento de imagem entre muitas outras.

De referir ainda que o robot utilizado, denominado ATLAS II, foi inicialmente desenvolvido pelos colegas José Luís Silva e José Miguel Gomes no ano de 2003, orientados também pelo professor Vítor Santos. O objectivo primordial consistia então em desenvolver o robot, fazendo modificações em vários sectores, de forma a que aquele se tornasse mais competitivo.

2 CLASSE CA – OBJECTIVOS

Como já foi dito pretende-se que o robot participe na classe de condução autónoma. Importa assim saber quais os requisitos desta classe para depois se passar à explanação das abordagens utilizadas para a resolução dos desafios impostos.

O objectivo do Robot será percorrer uma pista construída para a competição na classe de condução autónoma (CA). O formato da pista pode ser visto na figura seguinte.

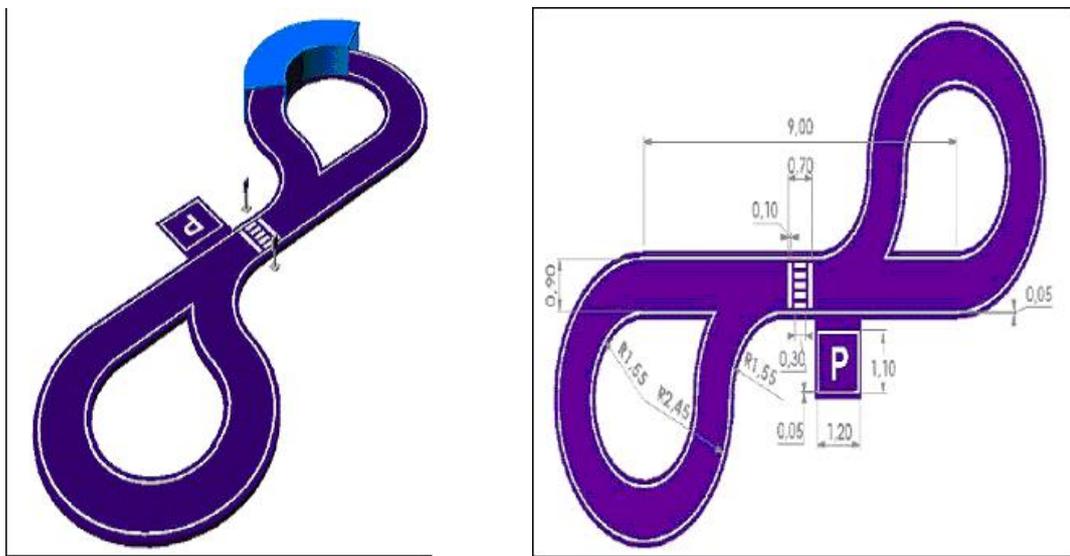


Figura 2. Formato da Pista. a) Vista isométrica. b) Vista de Topo.

De referir a presença de um túnel, passadeira, painel sinalético e parque de estacionamento. Todos estes são obstáculos adicionais que dificultam a tarefa e, se não forem cumpridos, penalizam o tempo de prova.

A prova será composta por três mangas distintas.

- A primeira manga consiste em percorrer a pista sem que esta apresente qualquer tipo de obstáculo adicional. Devem ser feitas duas voltas sendo que, no final, o Robot deve imobilizar-se na zona de chegada.
- A segunda ronda será realizada com o painel sinalético a funcionar. Este fornece ordens em forma de símbolos que o robot deve interpretar e cumprir. As ordens são várias incluindo indicações de direcção a tomar ou paragem na passadeira.

- A terceira e última manga será a mais complexa de executar. Na verdade existirão mais dois obstáculos. O túnel estará presente, o que irá impedir navegação baseada nas imagens adquiridas. Para além disso será construída uma zona de obras. Esta consiste num desvio da trajectória da pista, sinalizado com pinos. O trajecto de obras será aleatório.

Os símbolos utilizados em algumas das mangas para condicionar a actuação do robot podem ser vistas na figura seguinte.

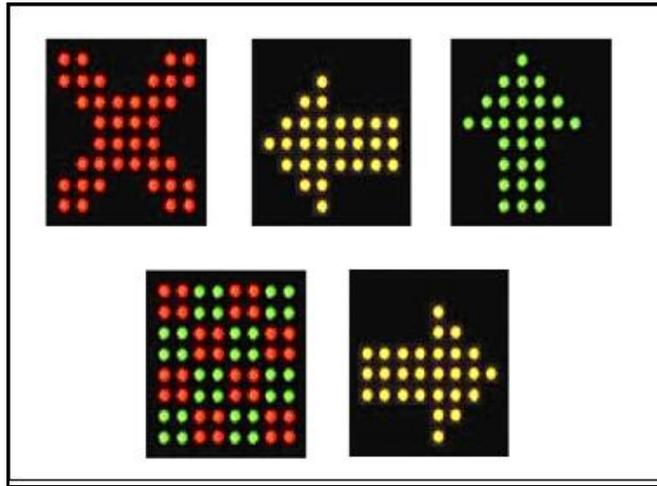


Figura 3. Símbolos mostrados pelo semáforo.

Da esquerda para a direita, de cima para baixo, os símbolos indicam que o robot deve:

- Imobilizar-se
- Seguir pelo trajecto esquerdo
- Seguir em Frente (trajecto direito)
- Indicação de última volta
- Estacionar

A zona de obras consiste numa área onde se devem ignorar as linhas da pista, navegando com base nos pinos dispostos aleatoriamente. Na seguinte figura pode ver-se uma representação da zona de obras.

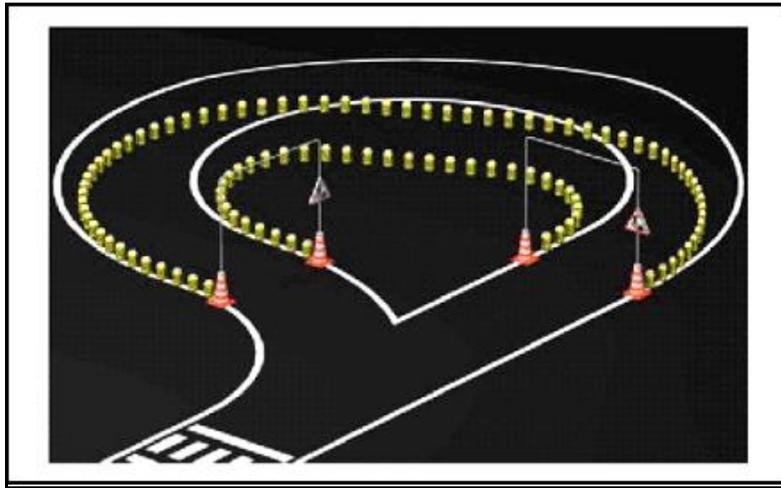


Figura 4. Zona de Obras.

3 PLANO DE INTERVENÇÃO

Expostos que estão os desafios a superar pelo robot importa agora definir quais as mudanças de fundo programadas logo no início do ano. Estas pretendiam dotar o robot de potencial suficiente para ombrear com os mais rápidos da classe.

3.1 SISTEMA OPERATIVO E AMBIENTE DE PROGRAMAÇÃO

Optou-se pela utilização do sistema operativo *Linux* e linguagem de programação c. Esta última foi eleita tendo em conta a sua versatilidade, rapidez de compilação e execução do código e também pela sua fiabilidade. Outra razão que contribuiu para esta mudança foi a dificuldade que existe em usar duas câmaras no sistema operativo precedente, problema que não se verifica no caso do *Linux*.

A utilização de duas câmaras é crucial para o aumento de eficácia do controlo. Permite uma melhor percepção das linhas, o que fornece mais informação ao algoritmo de navegação.

3.2 MICROCONTROLADOR

Outra grande mudança prende-se com a utilização de um microcontrolador, por oposição a uma carta de aquisição de sinais. O microcontrolador utilizado é o PIC 16F876 e tem como grandes vantagens em relação à alternativa anterior o facto de poder ser utilizado com qualquer sistema operativo bem como ser programável. Significa isto que o controlo de baixo e alto nível serão compartimentados. O PIC irá assegurar o primeiro gerindo eficazmente a aquisição de informação dos vários sensores e as questões relativas à modulação de sinais eléctricos para o correcto controlo dos motores. Isto permitirá simplificar o algoritmo de navegação pois este será unicamente responsável por ordens de alto nível. Por exemplo, imagine-se que, depois de analisadas as imagens vindas das câmaras o algoritmo de navegação decide rodar a roda dianteira para 30 graus. Esta ordem é enviada ao PIC, que se irá encarregar de verificar o ângulo actual da roda, calcular o sentido de rotação e gerar o PWM (onda quadrada) e outros sinais necessários ao funcionamento do motor. Fá-lo-á até que o sinal recebido do

potenciómetro seja correspondente aos trinta graus requeridos, altura em que desliga e immobiliza o motor da direcção.

3.3 UTILIZAÇÃO DE DUAS CÂMARAS PARA SEGUIMENTO DE PISTA

A presença de duas câmaras é um factor chave no controlo do Robot. Este facto assegura redundância ao sistema de navegação, melhorando-o consideravelmente. Pensava-se inicialmente que o sistema operativo Windows não permitia a utilização de duas câmaras. Em meados de Janeiro de 2004 encontrou-se uma *toolbox* do *MatLab* que possibilita aquela utilização. De qualquer modo, considera-se que a passagem ao sistema operativo *Linux* é proveitosa devido a muitos outros factores, pelo que esta descoberta não invalida a opção feita.

3.4 UTILIZAÇÃO DE UMA CÂMARA PARA IDENTIFICAÇÃO DO SINALIZADOR

Foi também decidida a utilização de uma câmara para identificação dos símbolos demonstrados pelo sinalizador. Esta alteração antevê dotar o ATLAS II dos meios necessários à correcta identificação dos símbolos. No ano ulterior foi utilizado um sensor de infravermelhos que não veio a funcionar correctamente.

4 COMPONENTES DO ATLAS II

O ATLAS II, nome dado ao robot desenvolvido no âmbito do projecto final de curso, é um robot autónomo e móvel. Quer isto dizer que deve ser capaz de se mover ao longo de um ambiente, captando com os sensores informações vindas daquele e agindo com base nelas para ter um comportamento que possa ser considerado lógico. Neste caso específico o ambiente é uma pista ou circuito que pretende simular uma estrada real, ao longo da qual o robot terá que navegar autonomamente, ou seja, sem qualquer intervenção humana.

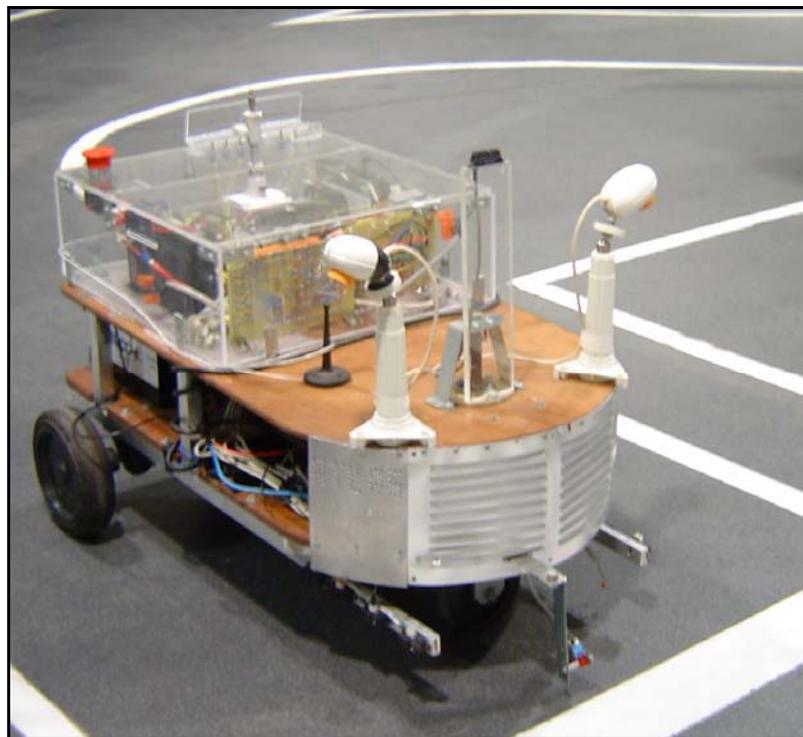


Figura 5. O Robot ATLAS II.

4.1 ALIMENTAÇÃO

O robot é alimentado por 4 baterias. Duas delas estão ligadas em paralelo e alimentam a fonte de alimentação do computador e a placa de controlo da direcção. Um conversor DC/AC encarrega-se de converter a corrente contínua das baterias para corrente alterna (220 V) da fonte de alimentação. O outro par de baterias está ligado em série (24 V) e encarrega-se da alimentação da carta de potência do motor de tracção.



Figura 6. Baterias do ATLAS II.

4.2 BOTÃO DE EMERGÊNCIA

O botão de emergência é usado para paragens de emergência no caso do robot se descontrolar ou caso seja necessário a sua paragem imediata por alguém estar na sua zona de passagem, entre outras. Funciona através do corte da voltagem que alimenta o motor de tracção, imobilizando o robot. Este botão é um requisito obrigatório para a participação na competição.



Figura 7. Botão de emergência.

4.3 SENSORES

O comportamento do robot será ditado, como foi dito, pela informação vinda dos sensores, a qual vai ser analisada pelo processador central. Feita esta análise são enviadas ordens para o microcontrolador, via RS232. Este, do qual se pode dizer que constitui o *firmware* do robot, encarregar-se-á de controlar os motores da direcção e tracção de forma a que o ATLAS II tenha o comportamento desejado pelo processador central. Compreende-se agora a importância de relatar detalhadamente o funcionamento e a informação disponibilizada por cada um destes para ter uma visão global do processo de interpretação dos sensores.

4.3.1 Sensores Infravermelhos Túnel

Estes sensores estão colocados na parte inferior lateral da frente do robot com o objectivo de detecção das paredes do túnel e navegação no mesmo. A leitura dos valores destes sensores só é feita quando um sensor, colocado na frente do robot a uma altura de cerca de 50 centímetros, detecta o tecto do túnel. Detectado o túnel o ATLAS II passa a guiar-se pelos sensores. Estes são sensores analógicos de infravermelhos (Sharp GP2D12) que permitem medir a distância a que se encontram as paredes do túnel e têm um alcance que vai desde os 8 cm até aos 80 cm.

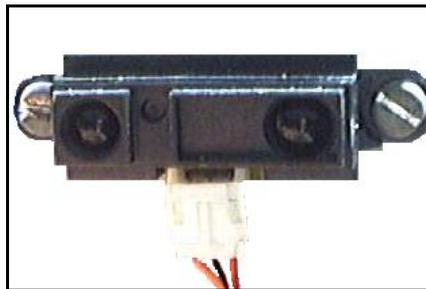


Figura 8. Sensor de Infravermelhos Sharp GP2D12.

4.3.2 Sensor Infravermelhos Passadeira

O sensor que permite detectar a passadeira é um sensor emissor/receptor de infravermelhos digital (Sick WT170-P112). Este sensor é utilizado na indústria o que dá garantias de elevada fiabilidade.

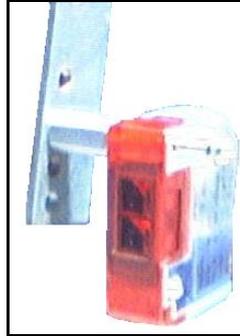


Figura 9. Sensor Infravermelhos Digital Sick WT170-P112.

4.3.3 Câmaras Utilizadas

Foi inicialmente decidido que, uma vez que o ambiente *Linux* permite facilmente a utilização de duas ou mais câmaras, utilizar um par de câmaras para seguimento de pista. Esta opção antevia-se muito vantajosa por ser bastante mais eficaz do que a solução do espelho utilizada pelos nossos antecessores. Para que o robot tome a direcção ordenada pelo semáforo foi também utilizada uma câmara com a qual se adquire a imagem do sinal do semáforo. As câmaras utilizadas são as *Philips Toucam pro pvc 740k*. Estas comunicam com o processador através do protocolo *USB*. São de boa qualidade, sendo consideradas as melhores no seu segmento.



Figura 10. Câmaras USB utilizadas.

4.4 MOTORES

São usados dois motores no robot, um motor (*Maxon RE040*) para a tracção e um segundo para a direcção (*Hybrid Stepper Motor RS 440-464*).

4.4.1 Tracção

O motor usado na tracção debita uma potência de 150 W às 7580 rotações por minuto e um binário máximo de 5,43 N.m. A caixa redutora tem uma redução de 1:15. Tendo em conta que o diferencial tem uma relação de redução de 1:2 a redução total é de 1:30. Para o seu controlo é usada uma carta de potência e um *encoder* que permitem o controlo da velocidade linear em malha fechada. O motor é alimentado a 24v de corrente contínua.

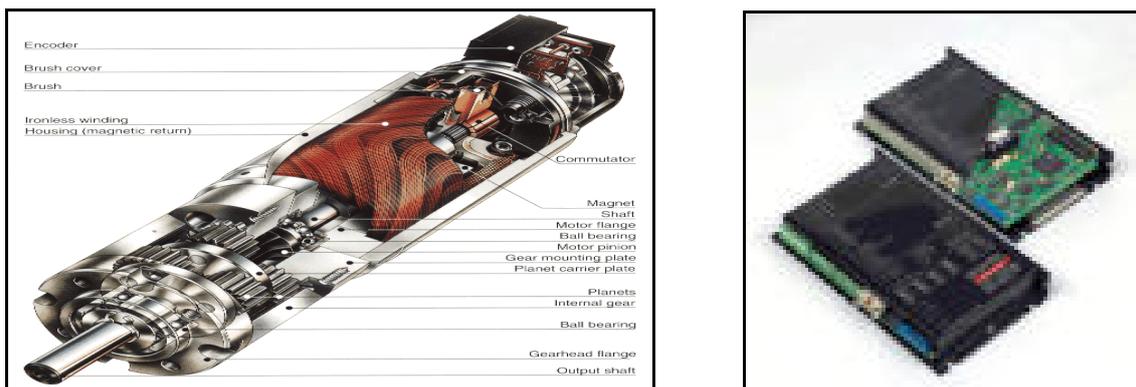


Figura 11 - a) Motor Maxon RE040. b) Encoder.

4.4.2 Direcção

O controlo da direcção é assegurado por um motor passo-a-passo em malha fechada.

Para controlar a parte lógica e de potência foi necessário construir um sistema electrónico.

Foi montado um potenciómetro no veio da direcção. Este permite fazer a leitura do ângulo da roda. Este valor é convertido num sinal eléctrico analógico e comparado com o valor que se pretende.

O motor da direcção tem um binário de 1.2 N.m e resolução de 1.8 °/passo e é alimentado a 12 V. A transmissão da direcção é executada por um sistema de correia dentada entre duas polias com uma relação de 1:2.



Figura 12. a) Motor de tracção. b) Potenciómetro.

4.5 PLACA DE REDE SEM FIOS

A necessidade de controlar o robot remotamente levou à aquisição de uma placa de rede sem fios. Foi assim possível prescindir do cabo de rede limitativo da mobilidade do robot na pista, caso se quisesse monitorizar o robot durante o percurso. Também para alterar o programa ou fazer *debugging* esta solução é bastante mais eficaz do que o cabo.

4.6 COMPONENTES ELECTRÓNICOS

4.6.1 Microcontrolador

Com o desenvolvimento dos circuitos integrados tornou-se possível armazenar centenas de milhares de transístores num único chip, constituindo-se assim um pré requisito para a produção de microprocessadores. Os primeiros computadores foram construídos adicionando periféricos externos tais como memórias, linhas de entrada e saída, temporizadores, entre outros.

Foi também possível o aparecimento de circuitos integrados contendo processador e periféricos. Assim apareceu o primeiro microcontrolador, projectado para ter conter todos os periféricos necessários integrados.

A lógica interna do microcontrolador é programável. Esta pode ser escrita em várias linguagens tais como *Assembler*, *C* e *Basic*. Para funcionar, o microcontrolador necessita de uma tensão e um oscilador. A execução das instruções do programa é regulada pelas pulsações do oscilador. É composto por vários pinos, entrada de tensão, e entradas e/ou saídas de sinais eléctricos, que são chamados portos. Na figura abaixo podemos observar o esquema de um PIC16F876A, que foi o modelo utilizado no robot.

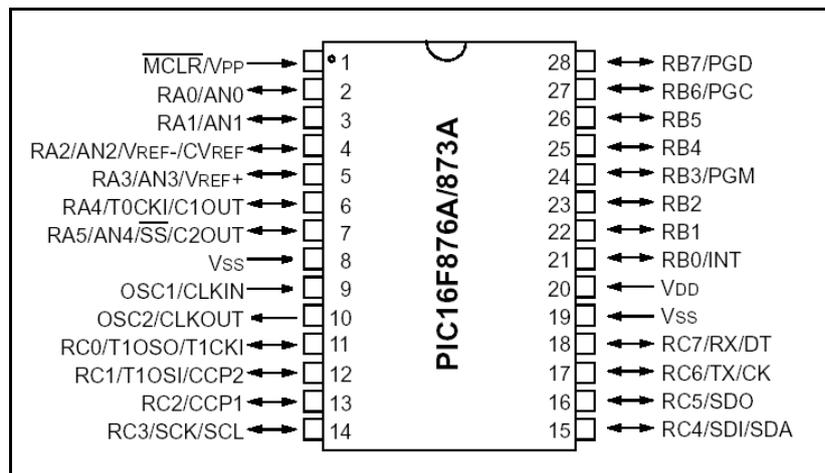


Figura 13. Esquema do PIC.

4.6.2 Multiplexer Analógico HEF4067B

No ano de 2003, o ATLAS utilizou três sensores de infravermelhos para a identificação e navegação no túnel. Esta solução funcionou eficazmente pelo que, para este ano, estava também programada a sua aplicação.

A variação em relação ao ano transacto surgiu na forma de um novo obstáculo: a zona de obras. Nesta os pinos que a demarcam estão distanciados entre si por uma distância igual ao seu diâmetro, cerca de 10 centímetros. Deste modo, a utilização de um sensor de cada lado era insuficiente, pois existiriam certamente alguns momentos em que nenhum dos dois estaria a detectar um pino. A solução encontrada foi o aumento do número de sensores, um total de cinco em cada lado, de forma a melhorar consideravelmente a probabilidade de pelo menos um dos sensores estar alinhado com um pino e fornecer informação válida ao algoritmo de navegação. Os sensores foram colocados numa régua, que foi acoplada ao andar inferior do ATLAS II.

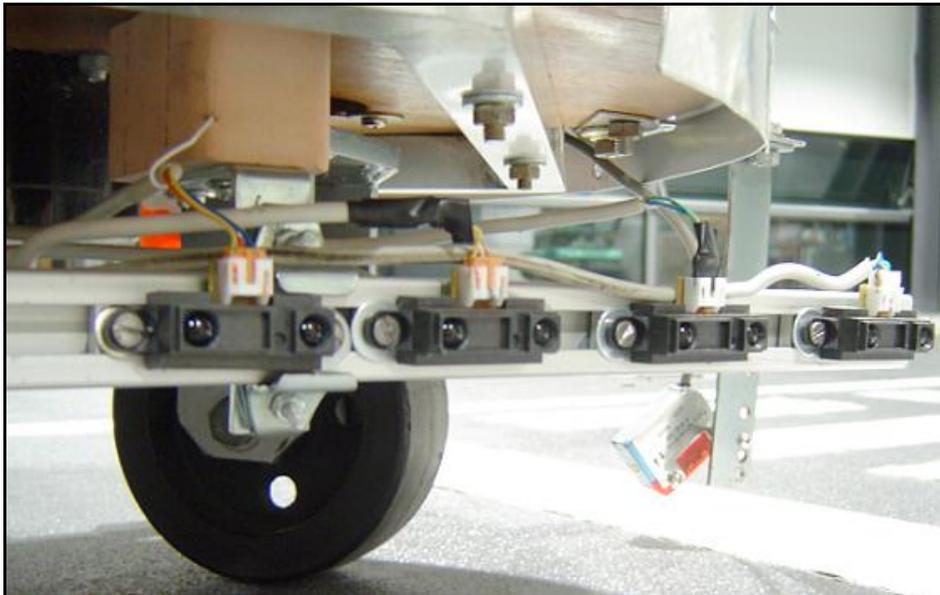


Figura 14. Régua de Sensores.

O sensor de identificação do túnel, que está direccionado para cima, manteve-se pois funciona muito bem.

Esta abordagem implicava a leitura rápida de mais de uma dezena de sensores. O microcontrolador não possui número tão elevado de canais de aquisição analógicos. Por este motivo foi necessário utilizar um *multiplexer* analógico. É um circuito electrónico no qual convergem vários sinais eléctricos. É possível escolher qual o canal que se quer receber bem como comutá-lo em curto espaço de tempo (da ordem dos nanossegundos). De seguida pode ver-se o esquema do multiplexer utilizado:

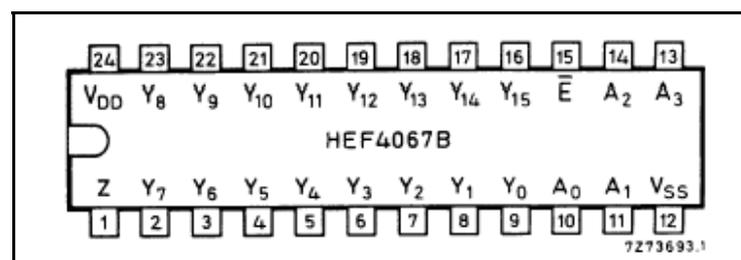


Figura 15. Multiplexer analógico de 16 canais.

A simbologia da figura significa o seguinte:

- V_{DD} é o pino de alimentação a 12V.
- V_{SS} é o pino do *ground*.
- Y_0 a Y_{15} são as 16 entradas analógicas.

- A_0 a A_3 são os pinos de endereçamento para definir qual o canal que se quer receber.
- Z é o pino comum onde é recebida informação de qualquer canal.
- \bar{E} é o pino de *enable*, que activa o *multiplexer* quando está a zero.

4.6.3 Contador Binário HEF4040B

Os PWM's são ondas quadradas, geradas pelo PIC, de *dutycycle* variável, ou seja, o tempo em que ficam no estado 0 ou 1. Aplicando um condensador ao fio que leva o sinal do PWM, obtém-se uma média do valor, conseguindo assim simular uma saída analógica. O problema surge quando se constata que o PIC não consegue gerar frequências suficientemente baixas para serem suportadas pelo motor passo-a-passo. O HEF4040 é usado, neste caso para dividir a frequência dos PWM's gerados.

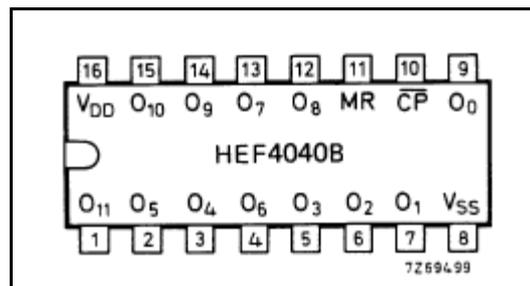


Figura 16. Contador Binário HEF4040B.

A simbologia da figura significa o seguinte:

- CP é a entrada do *clock* (*High to Low edge-triggered*),
- MR é *Master reset Input* (*active High*),
- O_0 to O_{11} são as entradas paralelas.

4.6.4 L297 e L298

O funcionamento do motor passo-a-passo é baseado num circuito de controlo L297, que realiza as funções de gerador de fases, e num *driver* de potência L298N. Estes dois

circuitos integrados permitem controlar o motor através de três sinais: *Clock*, *ClockWise* / *CounterClockWise* e *motor enable/disable*.

Quando o L298 muda de fase é gerada uma corrente muito alta que leva a um sobreaquecimento. Para resolver este problema foi acoplado um dissipador ao circuito integrado L298N.

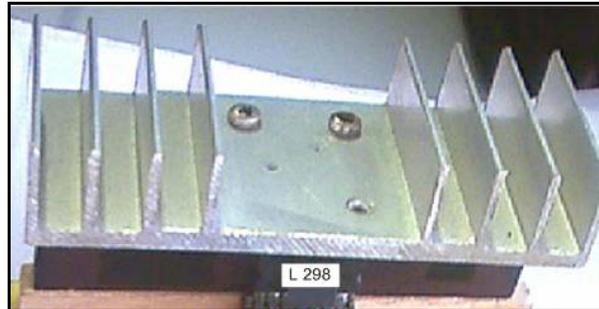


Figura 17. Dissipador.

Apresentam-se agora os esquemas de ambos os circuitos integrados.

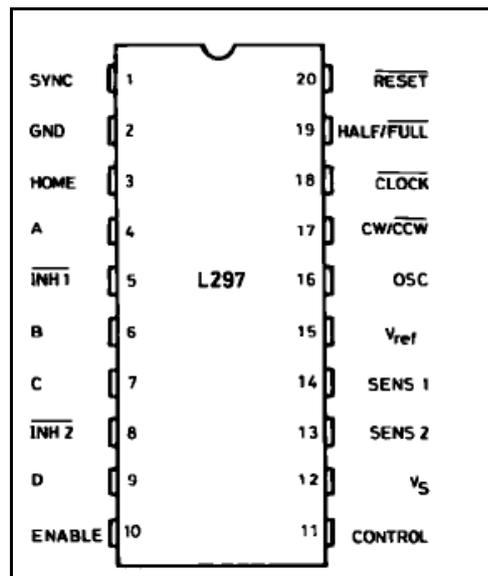
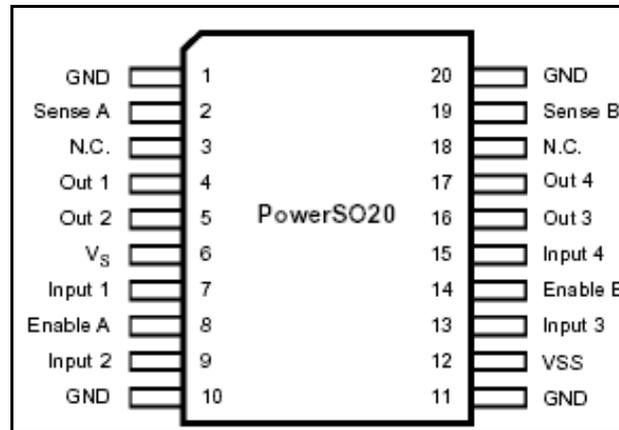


Figura 18. L297.

De seguida pode ver-se uma tabela com a explanação das funções dos pinos.

Tabela 1. Funções dos pinos do L297.

Symbol	Parameter	Value	Unit
V_s	Supply voltage	10	V
V_i	Input signals	7	V
P_{tot}	Total power dissipation ($T_{amb} = 70^\circ\text{C}$)	1	W
T_{stg}, T_j	Storage and junction temperature	-40 to + 150	$^\circ\text{C}$

*Figura 19. L298.*

De seguida pode ver-se uma tabela com a explanação das funções dos pinos.

Tabela 2. Funções dos pinos do L298.

Symbol	Parameter	Value	Unit
V_s	Power Supply	50	V
V_{ss}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel) - Non Repetitive ($t = 100\mu\text{s}$) - Repetitive (80% on -20% off; $t_{on} = 10\text{ms}$) - DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ\text{C}$)	25	W
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ\text{C}$

4.7 PLACAS ELECTRÓNICAS

Para integrar todos os componentes electrónicos utilizados foi necessário criar placas electrónicas. Foram criadas duas placas. A primeira contém o PIC e todos os componentes necessários ao seu funcionamento. Para além disto possui fichas para conectar todos os fios necessários ao transporte de sinais eléctricos para os motores e para a outra placa. Nos Anexos I e II podem consultar-se os esquemas das placas da direcção e principal respectivamente.

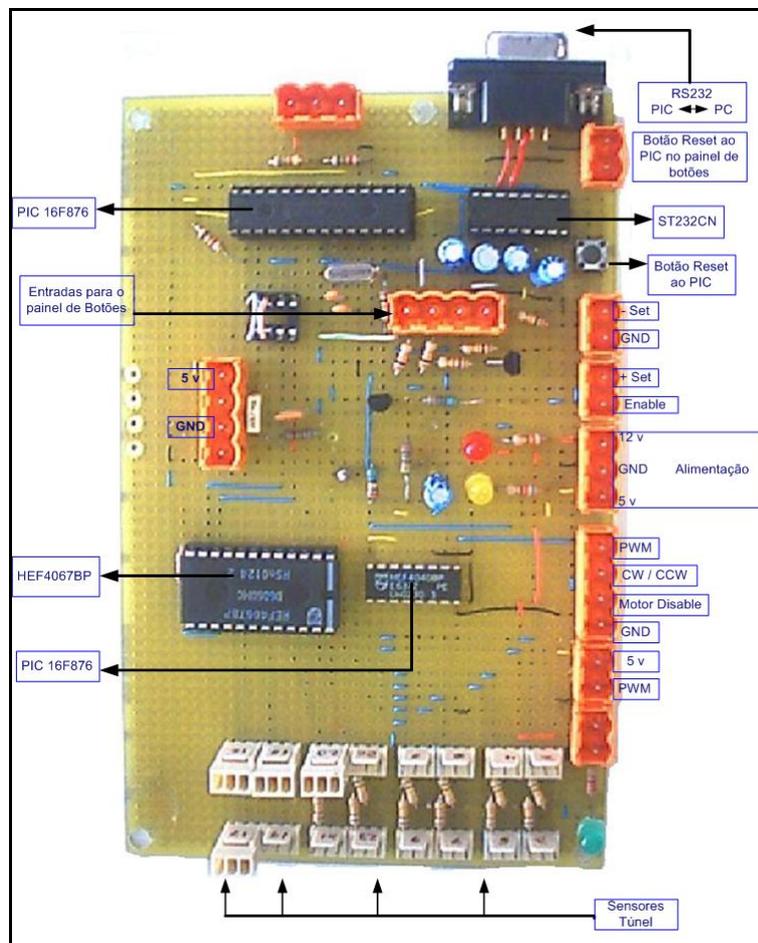


Figura 20. Placa Principal, esquema 1.

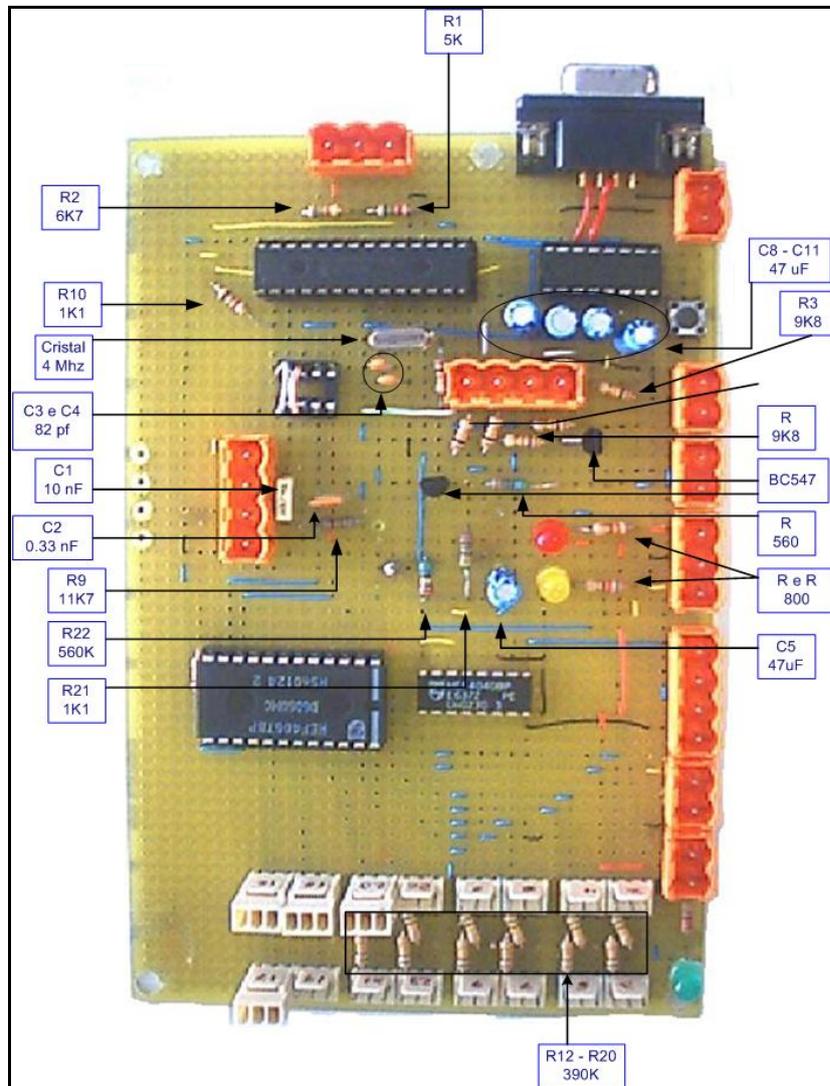


Figura 21. Placa Principal, esquema 2.

A segunda placa é dedicada inteiramente ao controlo do motor passo a passo da direcção.

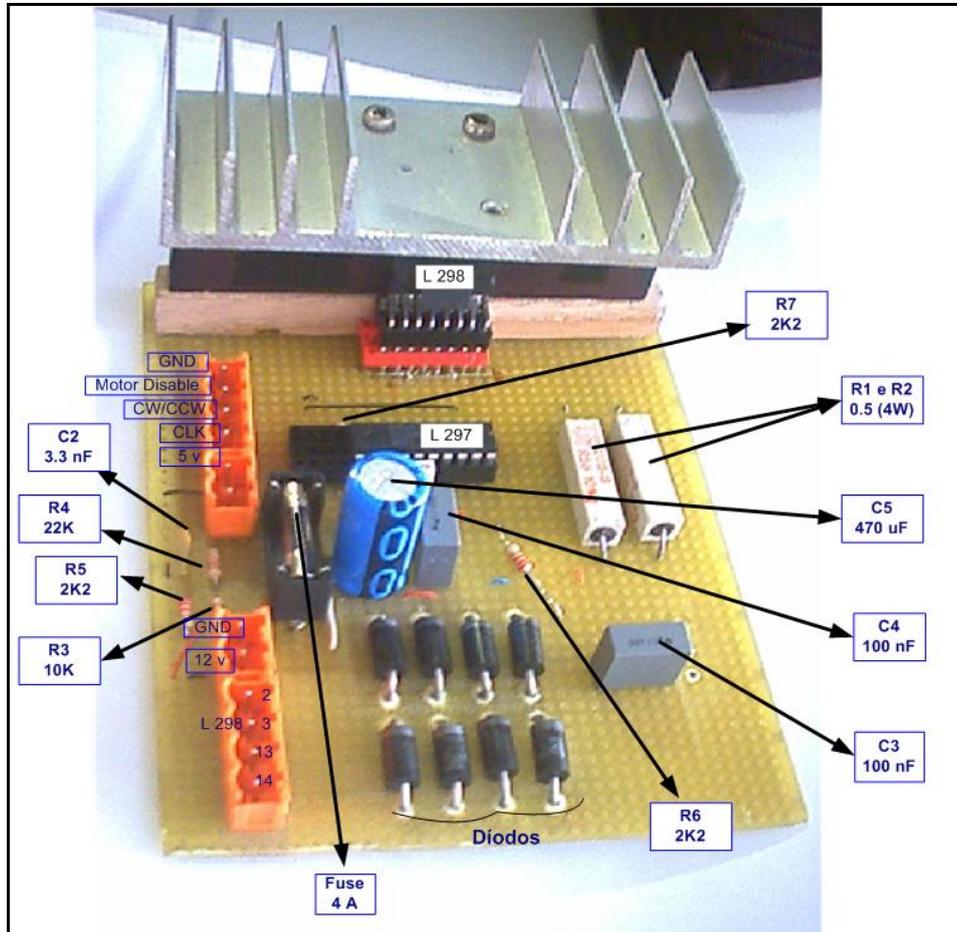


Figura 22. Placa de controlo da direcção.

4.8 DISPOSIÇÃO DOS COMPONENTES NA ESTRUTURA DO ROBOT

Apresentados que estão os mais importantes componentes utilizados na construção do robot, segue-se agora um diagrama onde se pode ver a disposição daqueles na estrutura do ATLAS II.

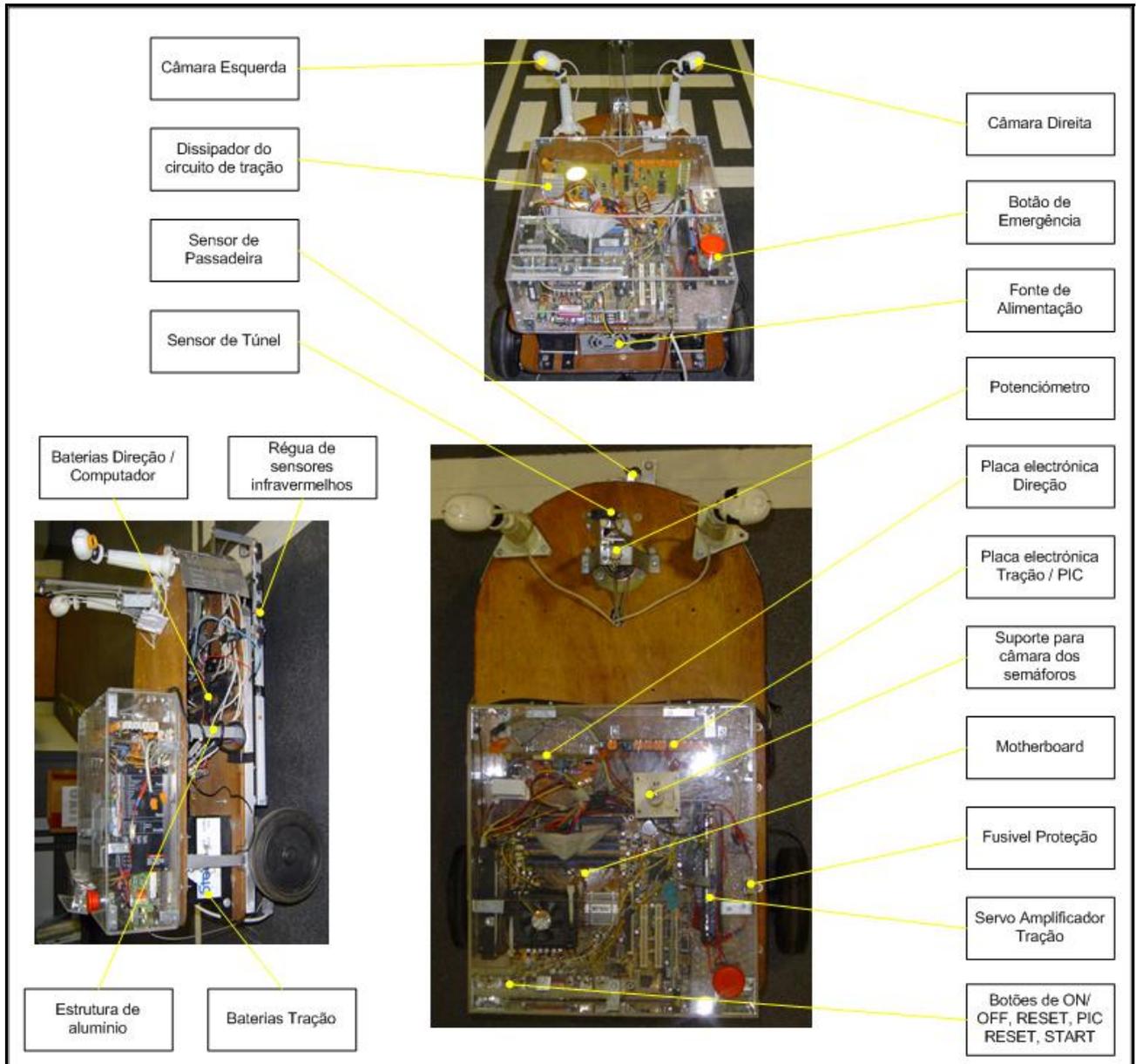


Figura 23. Disposição dos componentes na estrutura.

5 CONTROLO DOS MOTORES

Este breve capítulo pretende expor os problemas encontrados no controlo de ambos os motores do ATLAS II. Descreve depois como foram resolvidos aqueles, e justifica a presença de alguns componentes electrónicos pela resolução desses problemas.

5.1.1 Ondas Quadradas – PWM

O microcontrolador não possui saídas analógicas de tensão. Estas são no entanto necessárias para o controle dos motores. O que o PIC dispõe para suplementar esta situação é de pinos passíveis de gerar ondas quadradas em que o tempo de permanência nos seus dois estados booleanos pode ser manipulado. A aplicação de um condensador pode fazer uma média da onda obtendo-se assim um valor de tensão analógico cujo valor depende exactamente desse tempo atrás referido. O tempo pode ser expresso em percentagem no estado um, passando a denominar-se *dutycycle*.

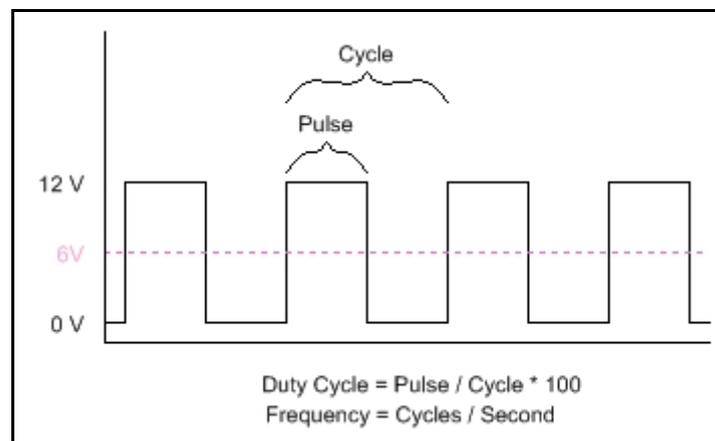


Figura 24. PWM. Onda quadrada de dutycycle variável.

5.1.2 Controlo da Tracção

O controlo do motor de tracção é bastante simples em virtude de, já no ano transacto se ter adquirido um servoamplificador. Este encarrega-se de gerar os sinais necessários ao funcionamento do motor. A velocidade do motor é definida pela diferença entre duas

tensões, uma de referência (foi utilizado o *ground*), e outra que varia consoante a velocidade desejada. Para gerar uma tensão analógica entre os zero (diferença igual a zero, logo robot parado) e os 10 V (requisito de tensão mínima do servoamplificador) foi necessário elaborar um circuito que fornecesse uma tensão entre estes dois valores. O PIC gera PWM's com uma tensão de 5V no estado 1. Foram acrescentados transístores ao circuito que, consoante o PWM gerado pelo PIC, modulavam uma onda quadrada semelhante mas com 10V no estado 1. Esta onda é filtrada por um condensador, gerando-se assim a tensão analógica requerida pelo servoamplificador.

De forma a dotar o ATLAS II da manobra de marcha-atrás foi posteriormente ligado o valor da tensão de referência a um pino do PIC. Para realizar a manobra este era activo e a tensão para comando de velocidade desactivada. A diferença entre ambas teria valores negativos, fazendo o ATLAS II mover-se em marcha-atrás.

5.1.3 Controlo da Direcção

O controlo do motor de direcção é um pouco mais complexo. Neste caso não foi necessário subir a tensão pois o circuito integrado de lógica funciona com os 5V habituais. Foi necessário gerar três sinais:

- *Motor enable/disable*
- *Clockwise/Counterclockwise*
- *Clock*

Destes sinais, os dois primeiros são digitais, ou seja, são muito facilmente gerados por um pino de saída do PIC. A lógica de activação destes pinos é programada no PIC. O problema surge quando se fala do terceiro sinal necessário. O *clock* é o relógio que irá comandar a frequência de actuação do motor passo a passo. O PIC pode facilmente gerar estas ondas, através dos PWM's. O problema surgiu quando se verificou que a frequência mínima gerada pelo PWM do PIC era muito superior aos limites permitidos pelo motor. Assim foi necessário diminuir essa frequência, razão pela qual se implementou um integrado denominado HEF4040, que é exactamente um divisor de frequência.

6 SOFTWARE DESENVOLVIDO

O software desenvolvido é uma componente muito significativa do projecto ATLAS II. Foi criado software a vários níveis, quer para o processador central quer para o microcontrolador. Em quase todas as aplicações, a linguagem utilizada foi o *c*, que tem a vantagem de ser muito robusto e compilar executáveis que são processados muito rapidamente. Outra vantagem desta linguagem é a compilação independente dos vários ficheiros criados (compila apenas se o código fonte foi alterado) o que em aplicações de tamanho considerável é um trunfo insubstituível em relação a outras linguagens. Dividir-se-á este capítulo em dois, referindo no primeiro todo o software desenvolvido para ser executado pelo microcontrolador e, no segundo, será abordado o *software* de alto nível que constitui o algoritmo de navegação e seus acessórios.

6.1 BAIXO NÍVEL – MICROCONTROLADOR

O *software* criado classificado como baixo nível inclui toda a programação do PIC. O seu propósito principal é o resolver tarefas muito específicas, de forma a que a programação de alto nível não contemple pormenores de menor importância, mas sim factores chave do comportamento do robot.

6.1.1 Programação do PIC

A programação do microcontrolador (*PIC*) faz-se em linguagem *c*, sendo que depois é necessário compilar o código escrito com um software dedicado chamado *MPLAB*. Este cria um algoritmo em código máquina que é descarregado para o *PIC* por *RS232* através de um programa chamado *PIC_DOWNLOADER*. É importante referir que esta operação só é possível pois o *PIC* foi previamente programado com um programa chamado *Boot loader*, que permite a programação do *PIC* pela porta paralela. Apresenta-se um esquema do *software* utilizado que demonstra como deve ser programado o PIC.

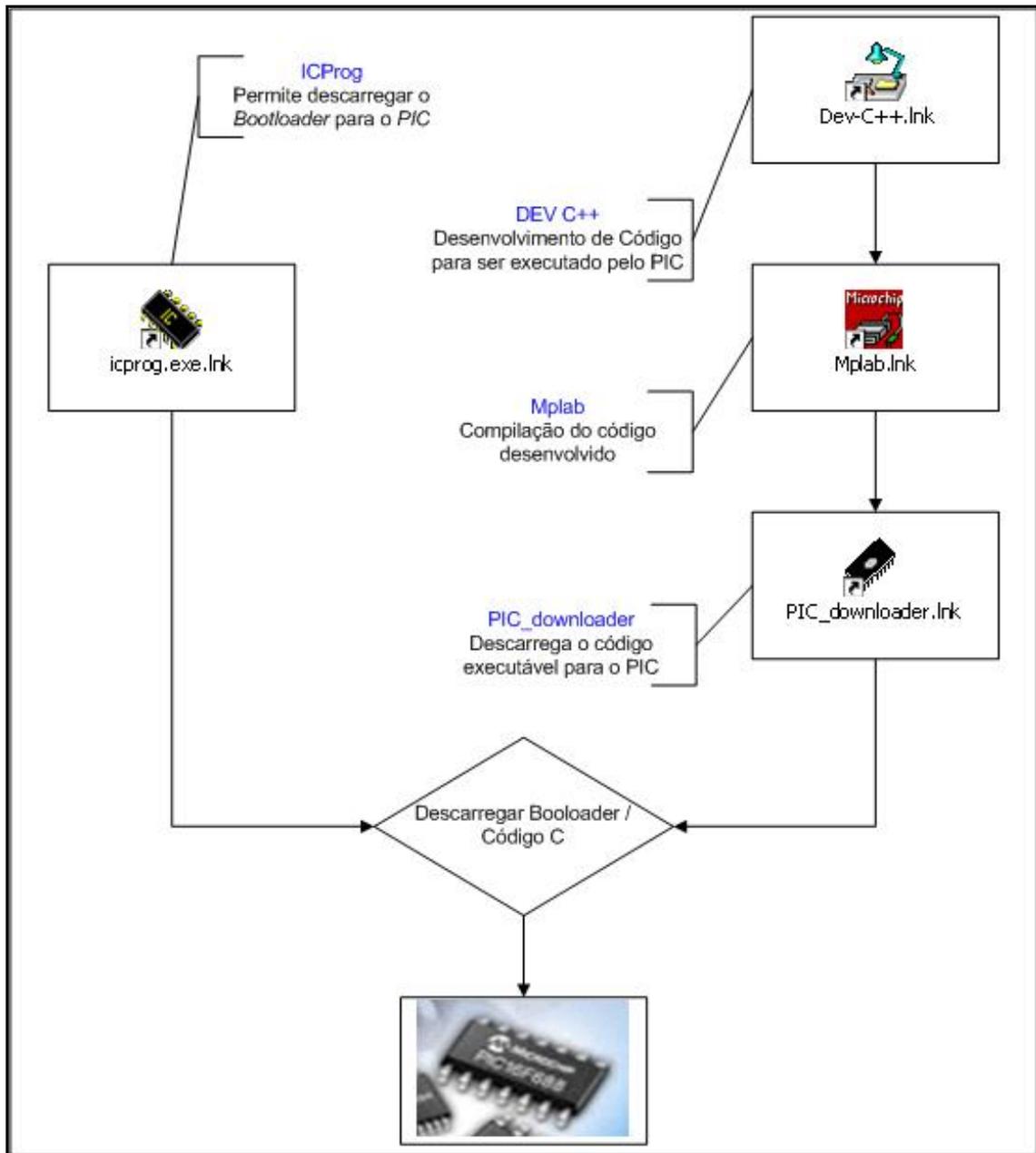


Figura 25. Software usado na programação do PIC.

6.1.2 Comunicação Série

Nesta aplicação em particular, em que o robot deve ser o mais rápido possível, o tempo dispendido nas comunicações entre o PC e o PIC deve o mínimo possível. Assim surgiu a necessidade de projectar uma comunicação bastante eficaz. Este capítulo será subdividido em dois uma vez que importa distinguir a comunicação consoante a direcção na qual a informação circula.

6.1.2.1 Comunicação PC→PIC

Todas as comunicações são feitas num único *byte*. Um *byte* contém oito *bits*, os quais terão, consoante a sua posição diferentes significados. O protocolo estabelecido foi o seguinte:

- Os três bits mais significativos compõem o chamado identificador. Esta variável terá 2³ combinações possíveis e identifica qual o tipo de ordem que se está a dar.
- O quarto bit mais significativo contém informação acerca do sentido. Esta variável nem sempre será utilizada mas existem casos onde será necessária, como se verá mais à frente
- Os quatro bits menos significativos quantificam a ordem dada.

De seguida podem ver-se duas tabelas com as possibilidades inerentes à comunicação de uma ordem de mudança de velocidade e de direcção.

Tabela 3. Tabela de Comunicações para a tracção.

TABELA DE COMUNICAÇÕES TRACÇÃO																
	M. Atrás	Parado	Em Frente													
Valor absoluto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Quantidade	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Sentido	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Identificador	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Byte a Enviar	10000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111
DutyCycle	0	60	140	220	300	360	420	500	560	620	680	740	800	860	940	1000

Tabela 4. Tabela de Comunicações para a direcção.

TABELA DE COMUNICAÇÕES DIRECÇÃO																																
	VIRAR ESQUERDA															VIRAR DIREITA																
Valor para BIN	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Valor absoluto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Quantidade	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Sentido	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Identificador	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Byte a Enviar	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111
Dir. Desejada	30	30	30	30	35	40	45	50	55	60	65	70	75	80	85	90	90	95	100	105	110	115	120	125	130	135	140	145	150	150	150	

6.1.2.2 Comunicação PIC→PC

Apesar de este trecho de código estar incluído no processador central, de forma a interpretar as mensagens vindas do microcontrolador, julgou-se conveniente este assunto ser abordado neste capítulo de forma a tratar de uma só vez toda a comunicação existente entre processadores. Esta componente da comunicação é um pouco mais elaborada, uma vez que a fluência de informação é maior nesta direcção. De facto, o microcontrolador necessita transmitir a informação relativa a um número elevado de

sensores, tais como os botões de início de prova, os sensores de túnel, de passadeira, etc. Não seria possível armazenar toda esta informação num único *byte*, razão pela qual se optou por uma outra técnica. O PIC transmite uma trama de vinte *bytes*. Eram cerca de dezasseis os sensores cujos valores deveriam ser enviados para o processador central. Adicionando a este valor dois *bytes* correspondentes aos caracteres de início e fim de trama, obtêm-se dezoito *bytes* de trama, valor que se resolveu arredondar para permitir uma rápida expansão do número de sensores utilizados.

Depois de enviados estes valores para o processador central convém saber como este os adquire. Uma vez que a rapidez de processamento é um factor chave nesta aplicação, também aqui foi necessário utilizar uma técnica rápida de leitura de dados. O algoritmo de navegação não utiliza processamento de tarefas paralelas, pelo que não convinha atrasar todos os outros processos esperando que a trama fosse totalmente adquirida pelo *buffer*, para só depois proceder à leitura. Por estas razões optou-se pelo modo de leitura não bloqueante. Este consiste em ler os *bytes* presentes no *buffer* e, utilizando como pontos de referência o início e final de trama e o índice do último *byte* lido, actualizar apenas os valores lidos. Tem este método a vantagem de não atrasar as outras tarefas lendo da porta de série apenas os valores disponíveis nesse instante. Como desvantagem destaca-se a complexidade da lógica de indexação dos trechos da trama lidos às suas variáveis representativas no algoritmo de navegação.

6.1.3 Funcionamento Global do PIC

Constituindo o cérebro de todo o *firmware*, o microcontrolador utilizado possui várias tarefas que deve desempenhar para que as decisões tomadas no algoritmo de navegação sejam devidamente cumpridas, bem como para informar correctamente o estado de alguns sensores. Uma das dificuldades encontradas foi conseguir coordenar as tarefas a desempenhar para que o PIC tivesse tempo para as executar correctamente. No código desenvolvido foram utilizados *interrupts*, que são na verdade funções que, caso se verifiquem determinadas condições, permitem que a normal execução de instruções seja suspensa, sendo executadas outras instruções que se tornaram entretanto prioritárias. No seguinte diagrama pode ver-se uma perspectiva global do funcionamento do código do PIC.

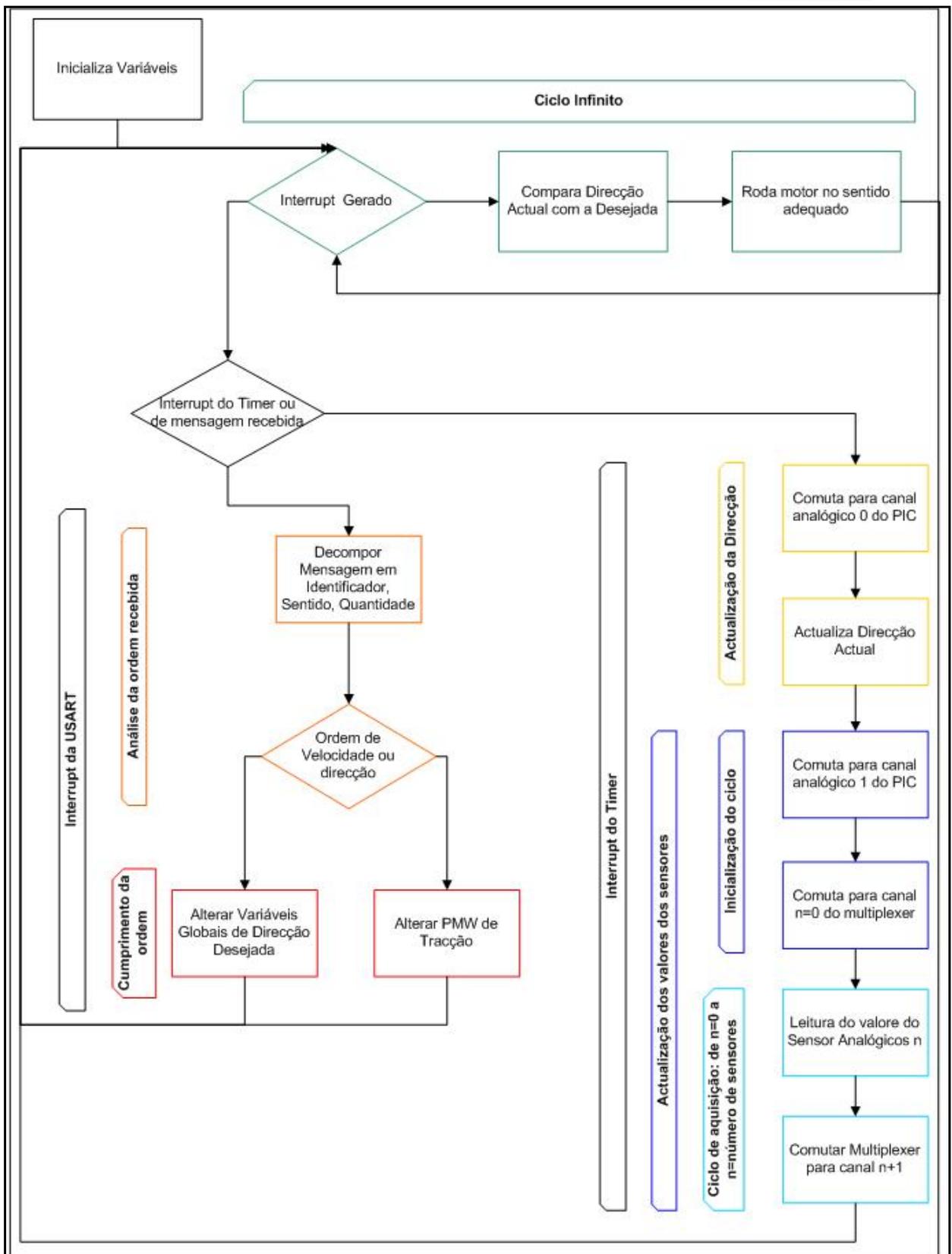


Figura 26. Funcionamento global do PIC.

De referir ainda que o microcontrolador utilizado é fabricado pela *microchip* e é denominado PIC 16F876. Este modelo revelou estar no limite das capacidades requeridas, quer a nível de velocidade de processamento quer a nível da quantidade de pinos utilizados.

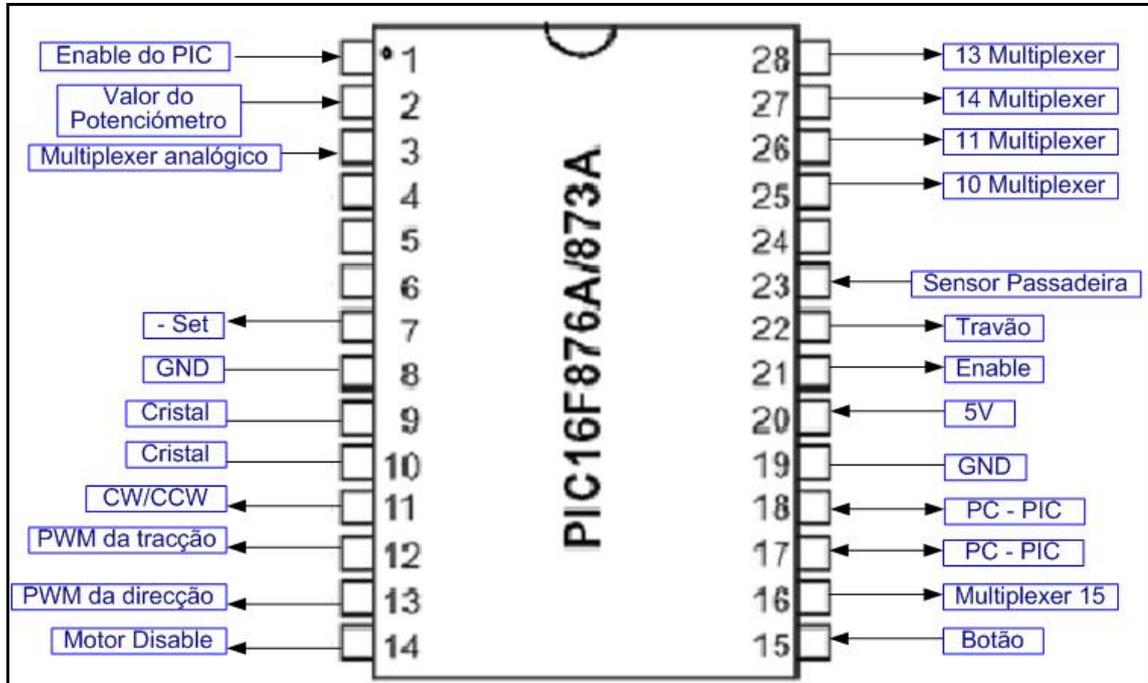


Figura 27. Pinos utilizados no robot.

6.2 ALTO NÍVEL – PROCESSADOR CENTRAL

Os algoritmos de alto nível são seguramente mais complexos do que aqueles abordados na secção 5.1. Têm a responsabilidade de tomar todas as decisões que vão afectar o comportamento do robot, ainda que estas sejam posteriormente postas em prática pelo microcontrolador.

6.2.1 Interpretação dos Sensores – *Lookup Table*

A resposta dada pelos sensores de infravermelhos utilizados não é linear. Esta pode ser visualizada na seguinte figura.

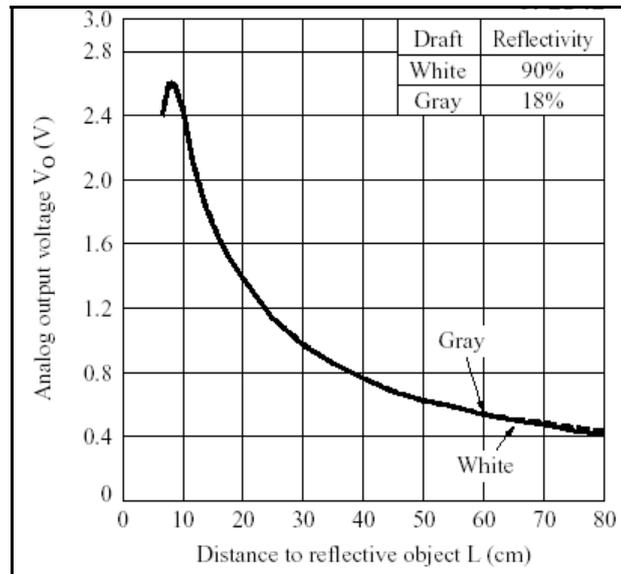


Figura 28. Curva de resposta dos sensores de infravermelhos.

Por este motivo foi necessário criar uma tabela com alguns valores de referência, que irá servir para calcular a transformação entre o valor da tensão recebido e a verdadeira distância. Esta tabela foi construída através de testes experimentais. Um algoritmo dedicado escrito para o efeito convertia os valores recebidos em distâncias realizando buscas cíclicas à tabela construída.

6.2.2 Contagem do número de Passagens na Passadeira

Como já foi referido a contagem do número de passagens na passadeira foi feita com um sensor digital de infravermelhos. Esta acção é conceptualmente simples, ou seja, sempre que o sensor esteja com valor um, o número de passagens deve incrementar. Esta técnica tem a vantagem da simplicidade, uma vez que basta atender ao valor do sensor e talvez de um temporizador (que contenha informação do tempo total de prova e do tempo desde a última passagem) para a efectuar. Existem, no entanto, algumas desvantagens. A primeira é a necessidade de condicionar a validade da leitura com base num temporizador. De facto, ao introduzir este tipo de condicionantes na lógica do sistema entra-se nos meandros do controle em malha aberta. Quer isto dizer que, uma vez que o sensor detecta qualquer tipo de linha branca, há que definir as seguintes directrizes na lógica de contagem: Incrementa um ao valor do número de passagens apenas se:

- Sensor de infravermelhos esteja com sinal um,

- Tempo desde a última passagem seja superior a um x predefinido.

A dificuldade surge em ajustar o tempo x . Este deve ser longo o suficiente para contemplar quase meia pista, sendo que se for maior do que este valor, o processo não funciona. Caso o tempo definido seja muito menor do que o tempo que o robot leva a percorrer meia pista, surge a possibilidade de o robot contar uma passagem se a sua frente pisar as linhas limítrofes da pista.

6.2.3 Navegação no Túnel / Zona de Obras

A navegação no túnel é também conceptualmente bastante simples. O primeiro passo é a identificação de uma zona de túnel. Este é facilmente suplantado através do sensor colocado na parte anterior da estrutura do Atlas II e orientado para cima, que fornece um valor de distância menor quando o robot entra no túnel. Depois de identificada esta situação, o algoritmo central saberá que deve ignorar as câmaras e navegar com base nos valores dos sensores de infravermelhos.

Os referidos sensores retornam uma distância à parede do túnel. Sabendo esta distância e a distância ideal (aquela em que o robot se encontra no meio do túnel) é uma questão de rodar a direcção no sentido de aproximar a distância à referida distância ideal.

O processo de navegação na zona de obras é muito semelhante. Existem apesar de tudo algumas variações. A identificação de zona de obras não pode agora ser feita pelo sensor de topo, por razões óbvias. Terá aquela de ser diagnosticada com base nos sensores laterais. Outra variação prende-se com o facto de os pinos estarem espaçados entre si e, por esse motivo, não constituírem uma parede sólida. Existem cinco sensores de cada lado e, nesta zona, nem todos estarão a detectar os pinos. É então necessário desclassificar os sensores com distância infinita, pois isto não significa que o robot esteja muito afastado dos pinos mas sim que esse sensor não está presentemente alinhado com nenhum pino.

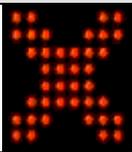
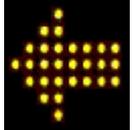
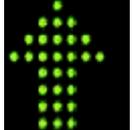
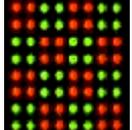
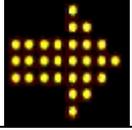
Quantificaram-se atrás os sensores laterais que permitem a navegação descrita neste capítulo. Interessa então saber como ponderar todas estas leituras para uma informação unitária que possa ser compreendida pela heurística do algoritmo. Logicamente, convém agrupar as leituras dos lados direito e esquerdo. Para agrupar estas leituras ponderaram-se várias hipóteses, sendo uma delas a média ponderada em que os sensores colocados

mais à frente teriam um peso superior aos outros. No entanto, verificou-se na prática que uma média simples (não ponderada) era suficiente para uma navegação eficaz, pelo que se optou por esta última.

6.2.4 Processamento da Imagem do Semáforo

Para efectuar uma análise correcta da imagem captada pela câmara destinada ao semáforo é necessário recorrer a duas técnicas muito utilizadas em processamento de imagem. A primeira, consiste em identificar a cor (ou as cores) presente (s) numa dada imagem. Quanto à segunda, esta não foi implementada por falta de tempo, existindo no entanto o argumento de que a análise de cor combinada com a contagem das passadeiras permite a identificação de todos os estados possíveis dos semáforos. Passa-se a explicar: existem cinco símbolos possíveis de serem mostrados pelos semáforos. Alguns destes não necessitam de ser reconhecidos por visão uma vez que a contagem de passagens na passadeira possui, informação redundante. A seguinte tabela descreve como é identificado cada símbolo:

Tabela 5. Meio de identificação dos símbolos do semáforo.

Símbolo	Acção	Meio de Identificação
	Parar.	Identificação de cor (excepto última volta).
	Seguir Esquerda.	Identificação de cor (excepto última volta).
	Seguir Frente.	Identificação de cor (excepto última volta).
	Última Volta.	Sabe-se da última volta pela contagem de passagens. Não atende à imagem.
	Estacionar.	Sabe-se que a prova terminou pela contagem de passagens. Não atende à imagem.

6.2.4.1 Modelos de Cor

Existem vários modelos matemáticos para representação da cor, sendo que o mais utilizado é o modelo RGB, em que a cor de cada pixel é dada por uma combinação, dos componentes vermelho (R - *red*), verde (G - *green*) e azul (B - *blue*). As câmaras recolhem informação, sendo que esta é armazenada em formato RGB. Este modelo de cor, apesar de ser dos mais utilizados, não é o mais interessante para fazer reconhecimento de cor. Na verdade, o que é verdadeiramente útil para a aplicação desta técnica denomina-se HSV (H - *hue*, S - *saturation*, V - *brightness*), e permite a isolar a cor numa só variável (*Hue*), separando-a de outras componentes pouco importantes tais como o brilho ou a saturação. A representação gráfica destes dois modelos de cor, é distinta, sendo que o primeiro é representado por um cubo e o segundo por um duplo cone. A figura seguinte representa os espaços tridimensionais dos modelos de cor RGB e HSV.

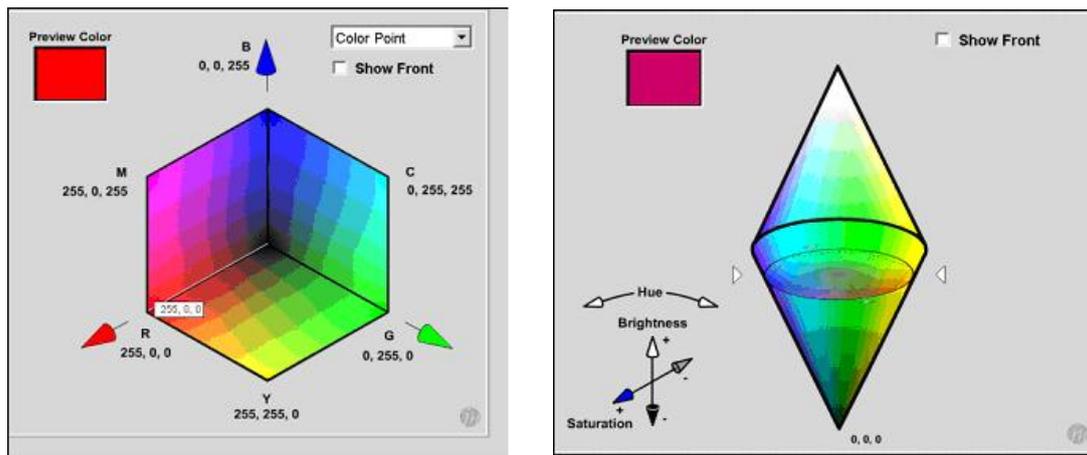


Figura 29. a) Modelo RGG. b) Modelo HSV.

6.2.4.2 Conversão RGB para HSV

Foi então necessário efectuar a conversão entre estes dois modelos de cor, uma vez que o segundo é muito mais eficaz para reconhecer cor. A conversão é complexa, tendo-se inclusivamente encontrado fórmulas em vários livros da área que não se conseguiram aplicar. Foi, posteriormente elaborado um algoritmo que, quando aplicado ao programa, se verificou funcionar. O algoritmo pode ver-se na figura que se segue:

```

TYPE
  EColorError = CLASS(Exception);
  TReal = DOUBLE;
...
// RGB, each 0 to 255, to HSV.
// H = 0.0 to 360.0 (corresponding to 0..360.0 degrees around hexcone)
// S = 0.0 (shade of gray) to 1.0 (pure color)
// V = 0.0 (black) to 1.0 (white)

// Based on C Code in "Computer Graphics -- Principles and Practice,"
// Foley et al, 1996, p. 592.

PROCEDURE RGBtoHSV (CONST R,G,B: TReal; VAR H,S,V: TReal);
  VAR
    Delta: TReal;
    Min : TReal;
BEGIN
  Min := MinValue( [R, G, B] ); // USES Math
  V := MaxValue( [R, G, B] );

  Delta := V - Min;

  // Calculate saturation: saturation is 0 if r, g and b are all 0
  IF V = 0.0
  THEN S := 0
  ELSE S := Delta / V;

  IF S = 0.0
  THEN H := NaN // Achromatic: When s = 0, h is undefined
  ELSE BEGIN // Chromatic
    IF R = V
    THEN // between yellow and magenta [degrees]
      H := 60.0 * (G - B) / Delta
    ELSE
      IF G = V
      THEN // between cyan and yellow
        H := 120.0 + 60.0 * (B - R) / Delta
      ELSE
        IF B = V
        THEN // between magenta and cyan
          H := 240.0 + 60.0 * (R - G) / Delta;

    IF H < 0.0
    THEN H := H + 360.0
  END
END (RGBtoHSV);

```

Figura 30. Algoritmo de conversão RGB para HSV.

A aplicação deste algoritmo permitiu reconhecer a cor, uma vez que, para isso, basta analisar o valor da variável *Hue*. O seu valor, que varia entre 0 e 360 graus, corresponderá a diferentes cores, como se pode ver na representação esquemática da seguinte figura.

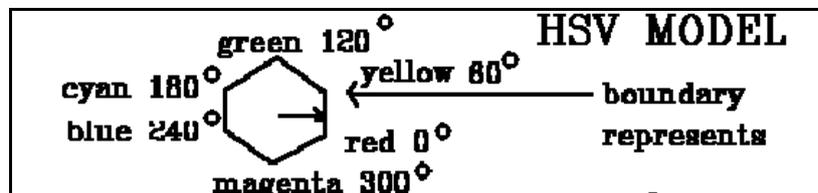


Figura 31. Significado do valor da componente Hue.

Segue-se agora uma tabela que discrimina as diferenças entre estes dois modelos:

Tabela 6. Diferenças entre o Modelo RGB e HSV.

	Modelo RGB	Modelo HSV												
Aplicações	Quase todas as aplicações que utilizam imagem.	Fundamental para Aplicações Robóticas, nomeadamente na identificação de cor.												
Representação	<p>Cubo de Eixos R (x), G (y) e B (z)</p>	<p>Duplo Cone. H (rotação XoY), S (valor do raio) e V (translação zz)</p>												
Componentes	<table border="1"> <tr> <td>R (Red)</td> <td>0 a 255</td> </tr> <tr> <td>G (Green)</td> <td>0 a 255</td> </tr> <tr> <td>B (Blue)</td> <td>0 a 255</td> </tr> </table>	R (Red)	0 a 255	G (Green)	0 a 255	B (Blue)	0 a 255	<table border="1"> <tr> <td>H (Hue)</td> <td>0 a 360°</td> </tr> <tr> <td>S (Saturation)</td> <td>0 a 1</td> </tr> <tr> <td>V (Brightness)</td> <td>0 a 1</td> </tr> </table>	H (Hue)	0 a 360°	S (Saturation)	0 a 1	V (Brightness)	0 a 1
R (Red)	0 a 255													
G (Green)	0 a 255													
B (Blue)	0 a 255													
H (Hue)	0 a 360°													
S (Saturation)	0 a 1													
V (Brightness)	0 a 1													

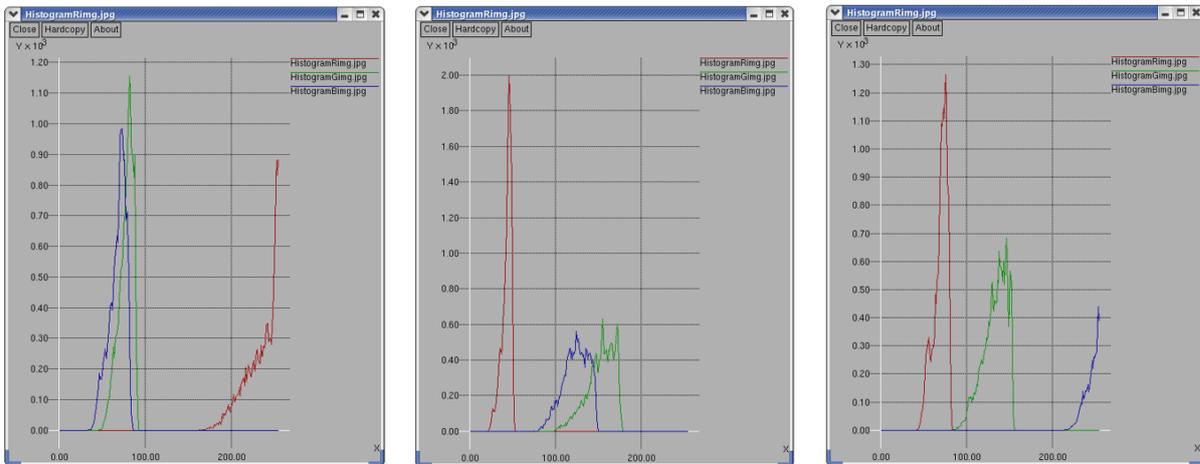
6.2.4.3 Identificação de Cor

Assimilada que estava a teoria inerente aos modelos de cor, as técnicas de conversão entre estes e características de cada um, passou-se à criação de um programa que permitisse identificar a cor de uma imagem adquirida. O programa desenvolvido em *c* permite a visualização da imagem adquirida bem como dos histogramas correspondentes às componentes R, G e B da imagem. Disponibiliza ainda um outro histograma da componente *Hue*, para fácil identificação da cor através do valor desta última. Vejamos agora alguns exemplos. Foram tiradas três fotos distintas, com cores básicas a ocupar toda a imagem em cada uma delas.



Figuras 32. Cores elementares. a) Vermelho. b) Verde. c) Azul.

Os histogramas retirados do formato RGB foram os seguintes:



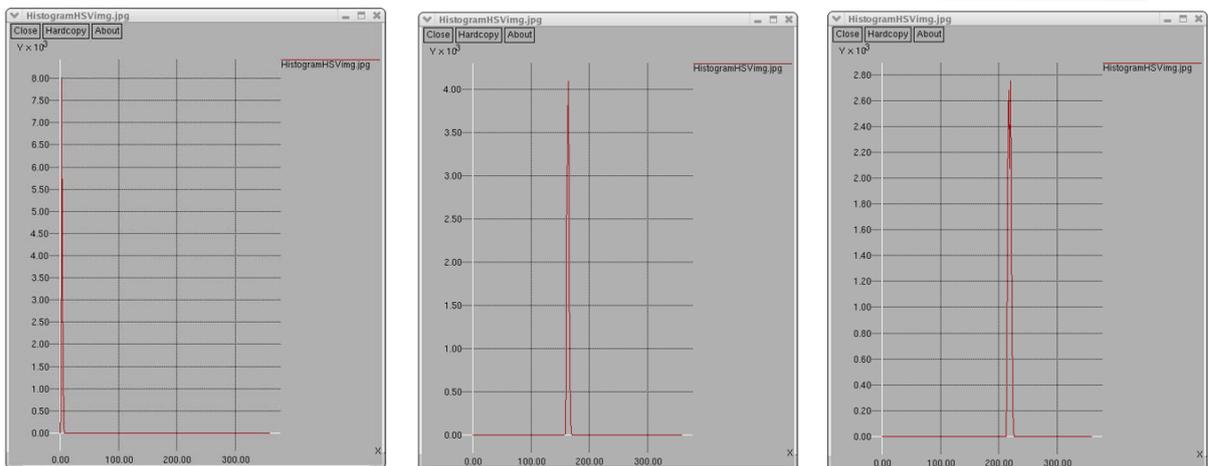
Figuras 33. Histograma RGB da imagem: a) 33.a. b) 33.b. c) 33.c).

Sabendo que as cores vermelha, verde e azul dos histogramas representam as componentes RGB respectivamente, da análise dos histogramas facilmente se chega às seguintes conclusões:

- O histograma da esquerda possui, na componente vermelha, vários valores muito próximos do valor máximo correspondente a 255. Daqui poderia supor-se que a cor maioritária seria o vermelho,
- O histograma do meio possui, nas componentes azul e verde, vários valores muito próximos do valor 200. Daqui poderia supor-se que a cor maioritária seria um azul esverdeado, correspondente a um meio-termo entre as duas cores,
- O histograma da direita possui, na componente azul, vários valores muito próximos do valor máximo correspondente a 255. Daqui poderia supor-se que a cor maioritária seria o azul.

Estas conclusões seriam, como se pode ver nas imagens, bastante acertadas, apesar da lógica algorítmica que as poderia elaborar ser algo complexa.

Vejam-se agora os histogramas correspondentes à componente *Hue* do modelo HSV.



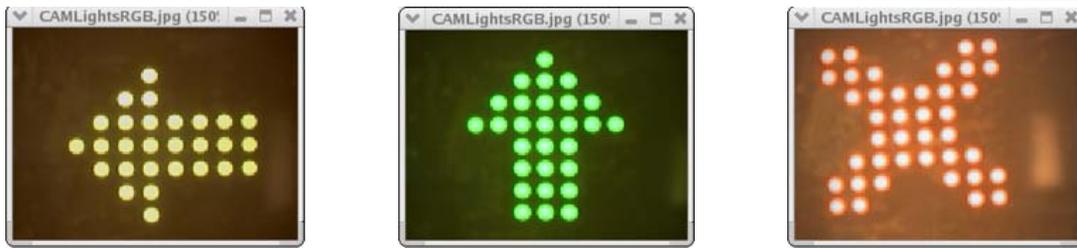
Figuras 34. Histograma HSV da imagem: a) 33.a. b) 33.b. c) 33.c).

Como se pode ver, estes histogramas picos, que representam grande preponderância de valores muito próximos. A ausência de outros valores indica que a imagem teria apenas uma cor. Esta primeira conclusão era já bastante difícil de tirar com base nos histogramas RGB, mais ainda se tivesse que ser laborada por um algoritmo dedicado. Observa-se também que o histograma da esquerda tem valores próximos de zero, o do meio valores aproximados a 190, e o último tem valores próximos de 220. Observando a figura 32 verifica-se que:

- O valor zero corresponde à cor vermelha,
- O valor 160 corresponde à cor azul esverdeado,
- O valor 220 corresponde à cor azul.

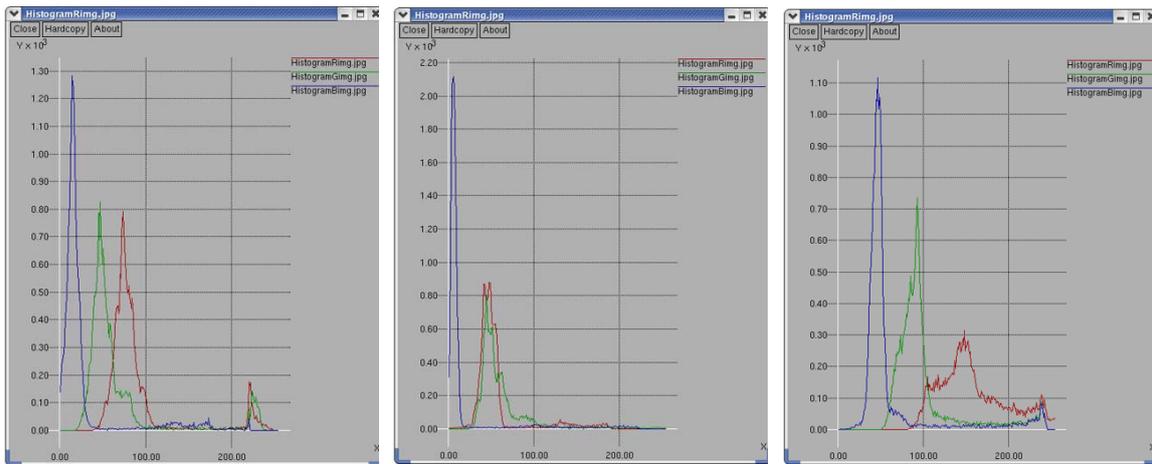
Estas conclusões estão todas correctas e constituem base suficiente para a análise dos semáforos. Mais ainda, o custo computacional da análise dos histogramas HSV é sem dúvida francamente menor do que aquele baseado no modelo RGB.

Estas imagens seriam possíveis de analisar em ambos os modelos, apesar de da análise em si ser mais complexa no primeiro. No entanto são imagens simples, com uma cor apenas, em nada semelhantes a uma imagem captada do semáforo em prova. Vejamos agora estas imagens, que são capturas dos símbolos mostrados pelos semáforos:



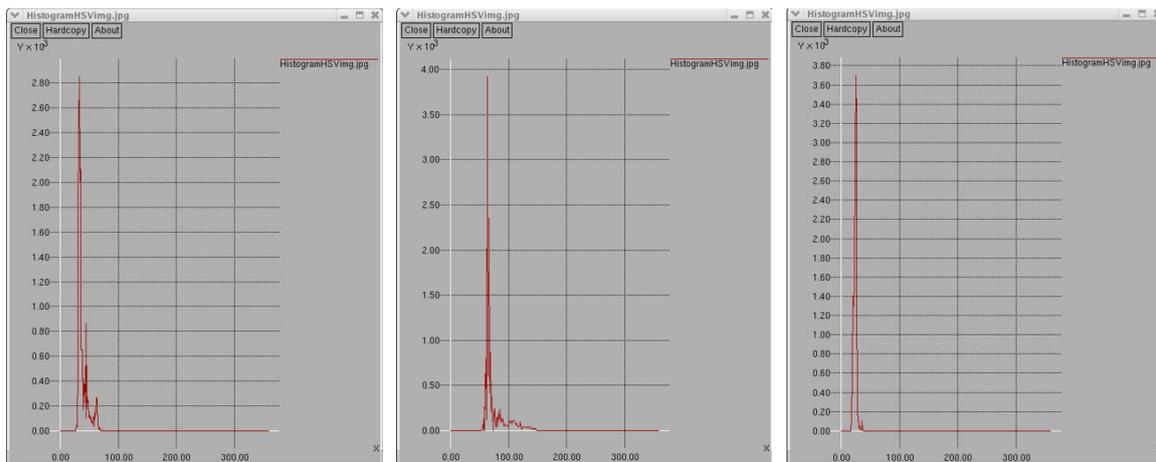
Figuras 35. Símbolos para: a) Virar à esquerda. b) Seguir em frente. c) Parar.

Os histogramas retirados do formato RGB foram os seguintes:



Figuras 36. Histograma RGB da imagem: a) 36.a. b) 36.b. c) 36.c).

Neste caso é já muito difícil tirar conclusões baseadas nos histogramas RGB. No entanto, observando os histogramas HSV:



Figuras 37. Histograma HSV da imagem: a) 36.a. b) 36.b. c) 36.c).

Estes histogramas continuam muito simples de analisar, senão vejamos:

- O primeiro histograma apresenta valores próximos de 60, que corresponde ao amarelo,
- O segundo é problemático, uma vez que apresenta valores da ordem dos 90, quando o verde é, na verdade localizado no 120. Ainda assim a informação é incomparável com a fornecida pelos histogramas RGB,
- O terceiro e último histograma apresenta valores aproximados ao número 30, o que também está ligeiramente desfasado do ideal, que seria o zero (vermelho).

6.2.5 Navegação na Pista

Esta componente é sem dúvida a mais complexa do trabalho realizado. Várias dificuldades aparecem neste tipo de navegação:

- Dificuldades inerentes ao processamento de imagem. Escolha do limite de binarização adequado,
- Afinação do algoritmo com base na colocação das câmaras,
- Afinação da reactividade do algoritmo.

Durante a maior parte do tempo de prova, o robot comportar-se-á com base nas decisões tomadas por este algoritmo, daí a sua importância.

6.2.5.1 Colocação das Câmaras

Tendo em conta que se pretendia utilizar duas câmaras, o que, no caso do ATLAS II ficou definido à partida por se achar que esta abordagem é aquela que possui mais potencial devido à redundância que se retira desta opção, o factor que maior influência teria no desenvolvimento do algoritmo de navegação seria a colocação das câmaras. Este facto é evidente dada a grande variedade de soluções possíveis. Para não comprometer qualquer possibilidade que pudesse vir a ser explorada foram adquiridos suportes para as câmaras que possuem os graus de liberdade suficientes para permitir qualquer orientação.



Figura 38. Suporte para câmaras.

Podem dividir-se em duas grandes vertentes as possibilidades de posicionamento das câmaras. A primeira consiste na local onde estas serão posicionadas, e a segunda engloba a orientação daquelas. Abordar-se-á primeiro a orientação, uma vez que é este o factor chave na navegação condicionando inclusive o local onde devem estar colocadas:

Inicialmente foi planeado orientar as câmaras num ângulo de aproximadamente 30 graus em relação ao eixo da pista. Este era utilizado por vários outros robots no ano de 2003, incluindo o vencedor. Esta configuração tinha algumas vantagens, nomeadamente a de o robot obter uma percepção da pista com algum alcance. Este factor pode ser contraditório uma vez que implica também um muito maior trabalho a nível de processamento de imagem. Uma desvantagem é que o robot facilmente perde ambas as linhas quando está mal colocado na pista.

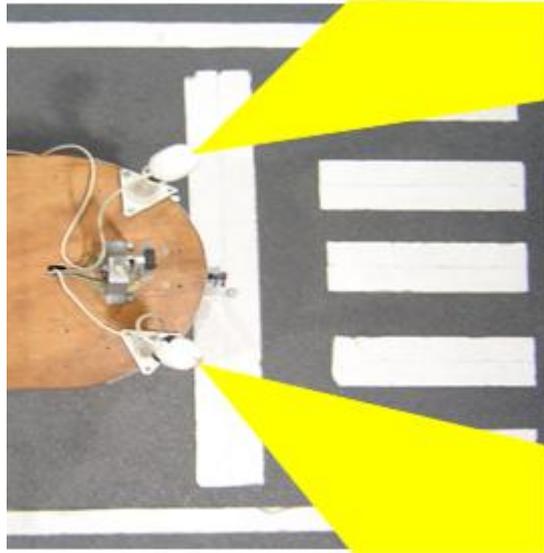


Figura 39. Configuração a 30. Vista de topo.

Pensou-se então noutra solução de configuração para minimizar a perda de linha. A solução encontrada foi cruzar as câmaras, ficando estas orientadas em aproximadamente menos sessenta graus. Assim a câmara esquerda visualizava a linha direita e vice-versa. Esta solução apresentava menos probabilidade de perda das duas linhas simultaneamente, mantendo no entanto a elevada dificuldade de processamento da imagem obtida.

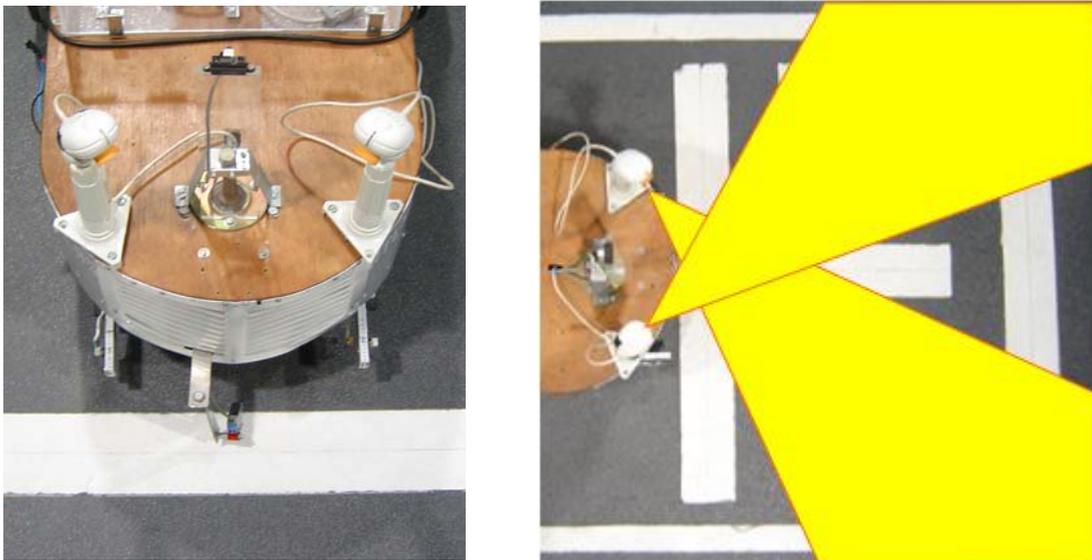


Figura 40. Configuração a -60°. a) Vista de frente. b) Vista de topo.

Assim, foi decidida uma configuração radicalmente diferente. Orientar as câmaras, não para a frente, mas para o lado. O ângulo pode ser aproximado a noventa graus. Esta solução parece um pouco ilógica, mas pesando as suas vantagens depressa se compreende porque se optou por ela. De facto, o algoritmo de processamento de imagem é consideravelmente simplificado uma vez ao não adquirir imagens da frente, a maioria das imagens a processar são bastante simples, pelo menos de um dos lados.

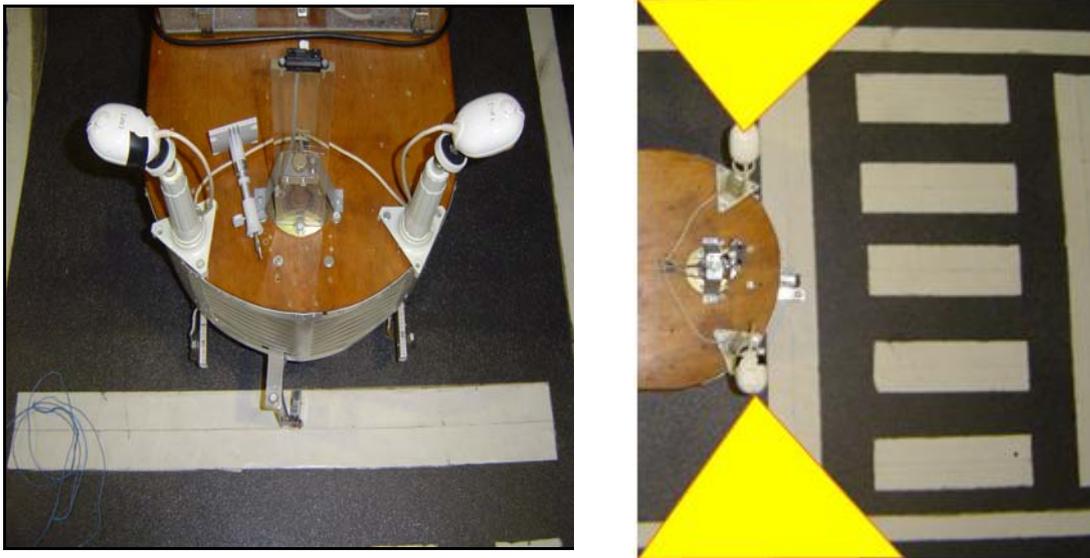


Figura 41. Configuração a 90°. a) Vista de frente. b) Vista de topo.

6.2.5.2 Configuração das Câmaras

No que respeita a configuração dos vários parâmetros das câmaras em *Linux*, foi utilizado um software denominado *camstream*. Este permite todas as afinações disponíveis nos *drivers* fornecidos pelo fornecedor, apesar destes serem apenas disponibilizados para *Windows*. Os parâmetros mais importantes são a saturação (para visualizar as cores) e o brilho (de forma a evitar reflexos). O balanço dos brancos é também extremamente importante, sobretudo na interpretação dos semáforos, já que os diferentes filtros disponíveis podem alterar por completo a cor visualizada.

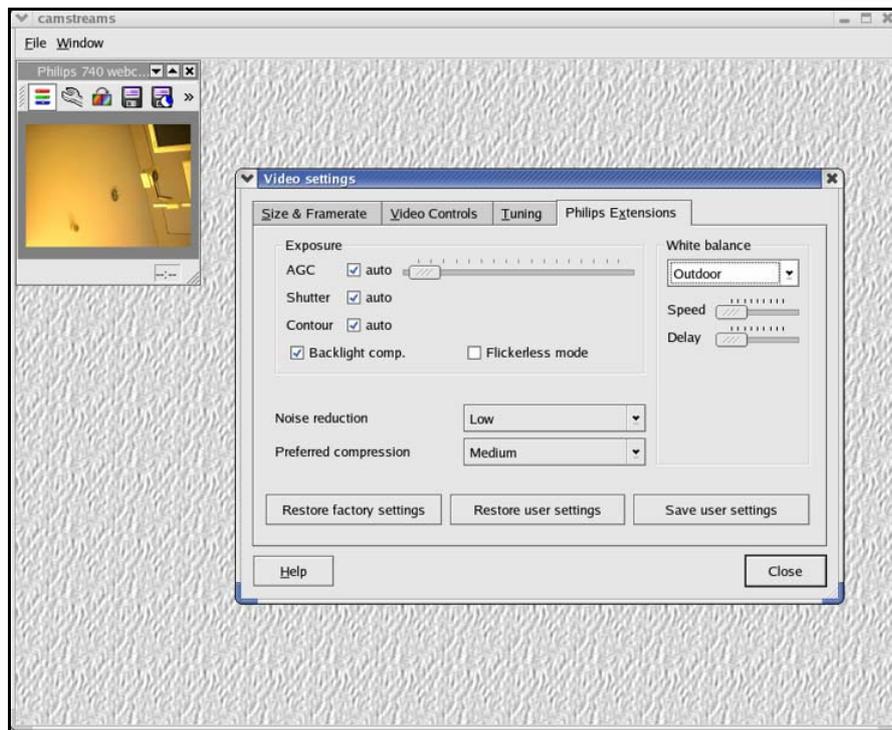


Figura 42. Afinação de câmara com o camstream.

6.2.5.3 Busca de Linha

A busca de linha é o processo mais complexo utilizado no atlas, sendo que a grande dificuldade reside não tanto nos aspectos técnicos da busca de linha (que se resumem a ciclos *for* em busca de pixels brancos) mas mais na tentativa de encontrar um solução que funcione para uma quantidade infinita de soluções. Em robótica denomina-se esta situação como a impossibilidade de formalizar um plano universal. Quer dizer que, dada a infinidade de situações com que um algoritmo se poderá deparar, é impossível utilizar uma programação concreta que se assemelha a um encadear de condições lógicas.

A busca de linhas efectuada pelo ATLAS II foi a seguinte: Observe-se a imagem.

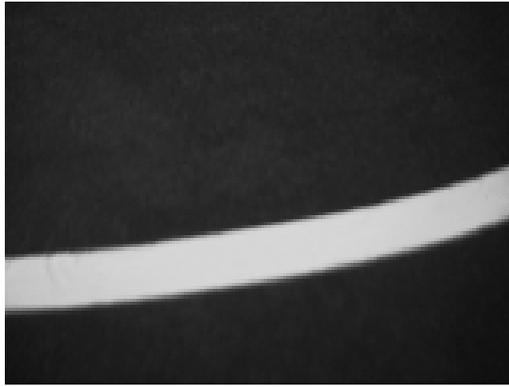


Figura 43. Imagem obtida pela câmara esquerda.

Esta imagem é bastante boa pois existe um contraste acentuado entre a linha e o resto do ambiente. Antevê-se assim uma boa imagem pós binarização. Quando binarizada, ficou com o seguinte aspecto:

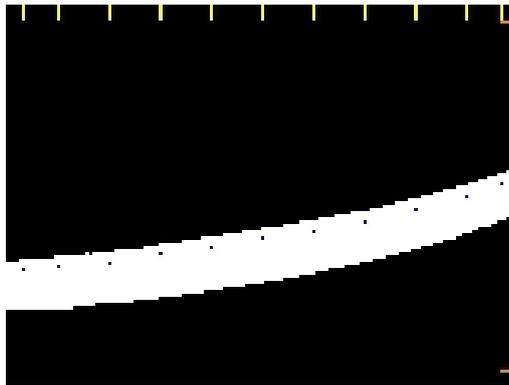


Figura 44. Imagem binarizada.

Como se pode ver esta imagem é excelente, dado que se conseguiu distinguir a linha de tudo o resto. Seguidamente, era executada uma busca de pixeis brancos em algumas colunas e linhas da imagem. As primeiras estão assinaladas a amarelo e as segundas a laranja. O algoritmo da navegação faria agora alguns cálculos, comparando os índices dos pixeis encontrados (assinalados a azul na imagem os correspondentes à busca vertical e a vermelho aqueles obtidos pela busca horizontal) com valores característicos de uma postura ideal do robot na pista.

6.2.5.4 Definição de Postura Ideal na Pista

Os valores ideais são números que correspondem aos resultados dos cálculos do algoritmo quando o robot está com a colocação e a orientação ideais na pista. Imagine-se que o robot está colocado tal como se vê na seguinte figura:

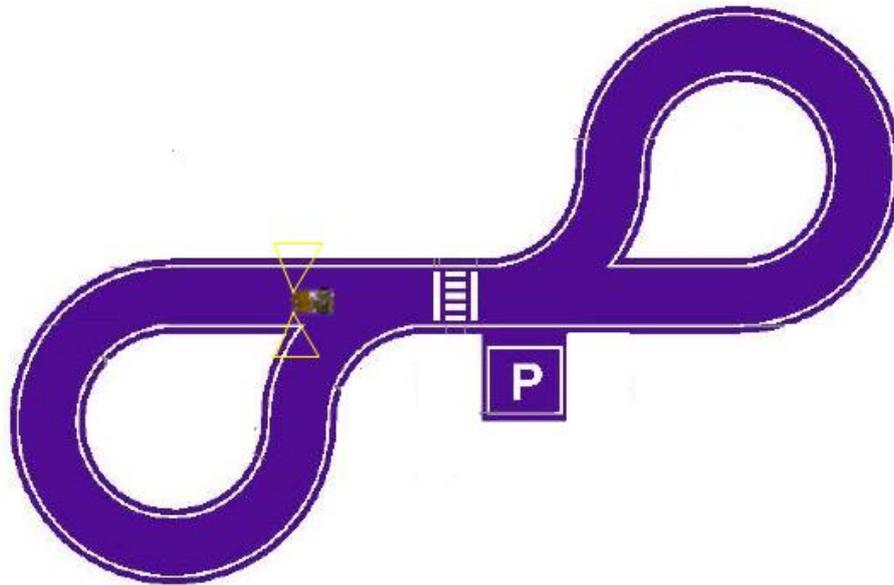


Figura 45. Postura ideal.

A cor amarela podem ver-se as áreas observadas por cada câmara. Nesta situação a imagem retirada pela câmara direita deve corresponder à tal situação ideal já que o robot tem a localização e a orientação ideais. Aquela seria muito simplesmente uma linha horizontal situada no centro da imagem.

6.2.5.5 Busca da Postura Ideal

Pretende-se agora demonstrar como funciona o algoritmo em termos de procura da postura ideal na pista. Como já foi dito, existem duas variáveis das quais depende a postura do robot na pista: a localização e a orientação. Na seguinte figura pode ver-se um desfasamento de localização com a orientação correcta.



Figura 46. Postura defeituosa – localização.

Como se pode ver, nesta figura o robot está bem orientado mas não está colocado no centro da pista. A imagem captada seria a seguinte:

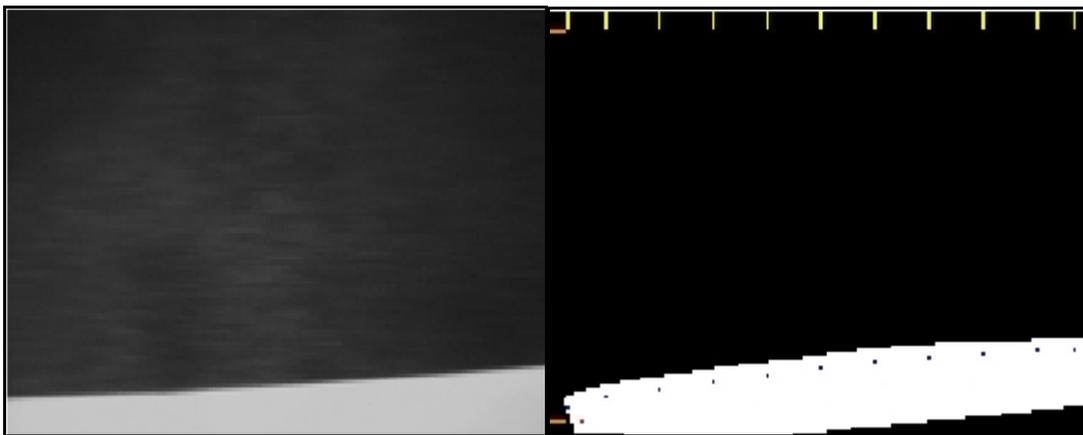


Figura 47. Localização defeituosa. a) Imagem normal. b) Imagem Binarizada.

Esta imagem apresenta a linha branca muito próxima do robot, o que levaria obviamente a uma necessidade urgente de o levar a afastar-se da linha. Os pixels encontrados, a azul, possuíam índices de linha muito elevados (mais do que metade da altura da imagem, que seria a situação ideal), o que faria com que o robot fosse levado a virar para a esquerda, afastando-se da linha.

A seguinte figura demonstra agora um desfasamento de orientação.



Figura 48. Postura defeituosa – orientação.

Adivinha-se já uma situação de emergência uma vez que se o robot não mudar imediatamente de direcção corre o risco de sair da pista. A imagem obtida pela câmara esquerda seria a seguinte:

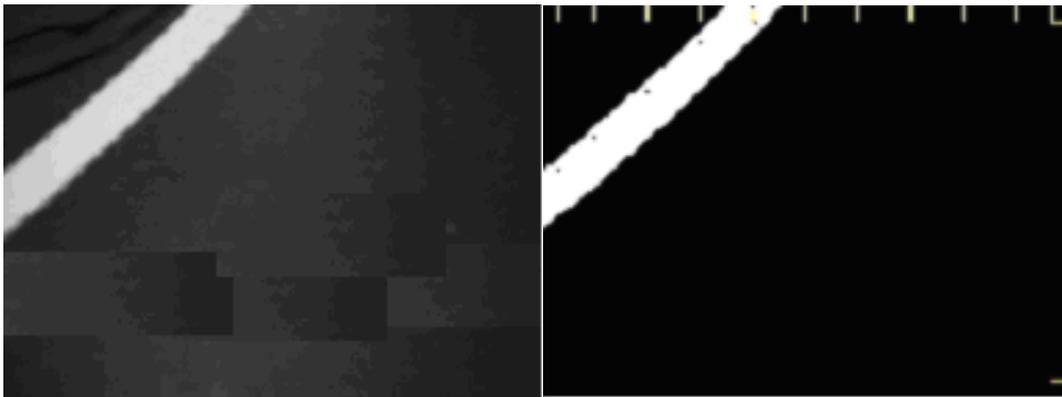


Figura 49. Orientação defeituosa. a) Imagem normal. b) Imagem Binarizada.

Esta imagem é muito diferente da imagem ideal. Nem todas as buscas verticais encontraram linha, mas as buscas horizontais (pixeis a vermelho) tornam-se agora fulcrais. Informam, juntamente com o cálculo da inclinação adiante abordado, que a linha está inclinada e que, portanto, o robot está muito distante do ponto virtual, obtido pela intercepção entre a linha e uma recta perpendicular à linha ideal.

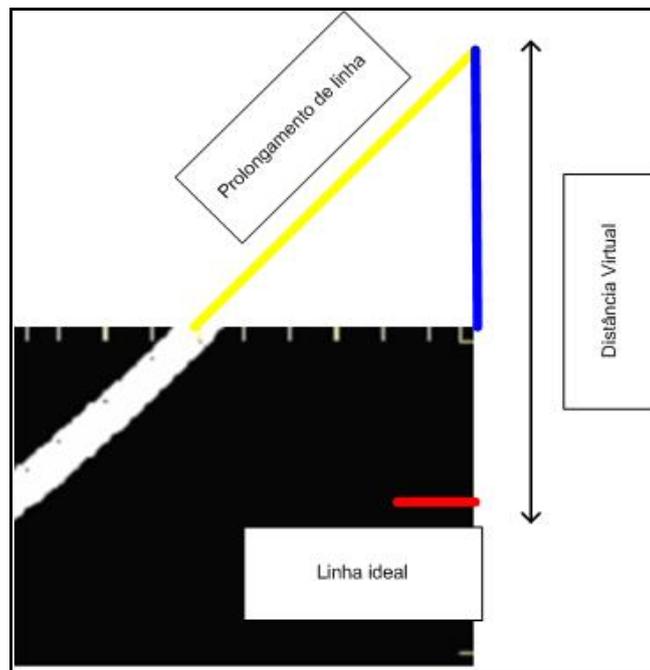


Figura 50. Linha virtual.

Esta abordagem permitia que o robot se apercebesse que estava muito distante da linha e portanto virar muito para a esquerda, compensando a orientação defeituosa.

6.2.5.6 Orientação da Linha

No capítulo anterior foi possível perceber a importância do cálculo da inclinação da linha. Este cálculo é bastante simples uma vez que se obtiveram as coordenadas de vários pontos na busca da linha. É calculada uma inclinação que é a média das inclinações entre vários pontos consecutivos.

6.2.5.7 Parâmetros de Validação

Aos problemas inerentes a uma má escolha do limite de binarização acrescentam-se ainda outros. Estes são alguns locais da pista em que as imagens obtidas não são compreensíveis para o algoritmo criado. Observe-se a seguinte imagem:



Figura 51. Entrada na bifurcação.

A imagem obtida pela câmara esquerda é, neste caso, problemática.

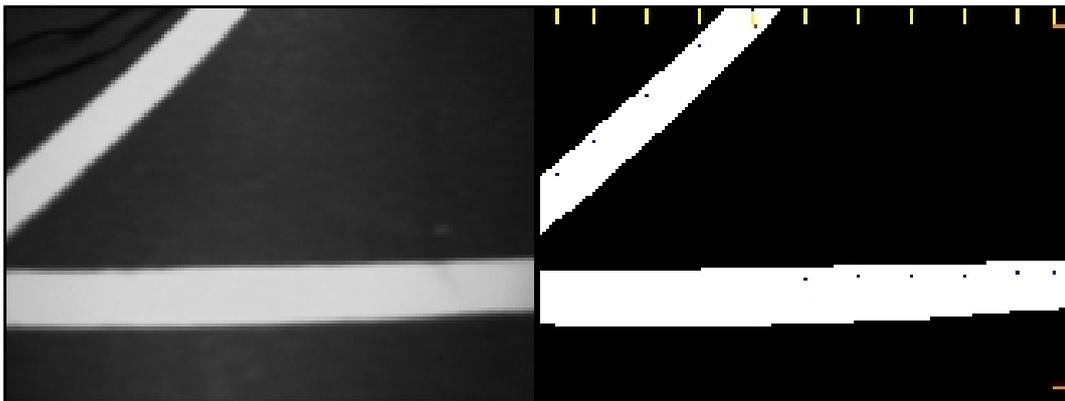


Figura 52. Bifurcação. a) Imagem normal. b) Imagem Binarizada.

Como se pode ver pela observação dos pixels azuis foram encontradas duas linhas. Acresce ainda que a busca horizontal cimeira também encontrou linha (pixel vermelho).

Existem ainda outras situações problemáticas, tais como a da seguinte figura:

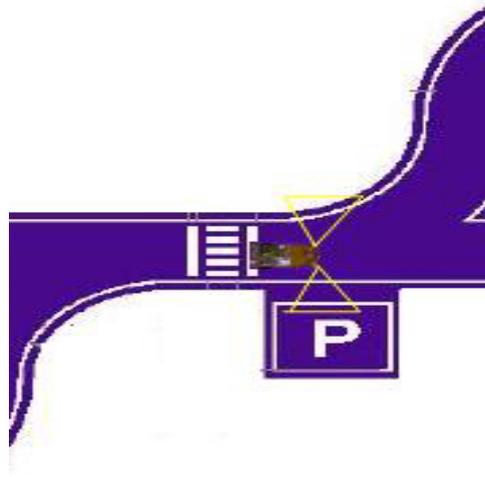


Figura 53. Situação de passagem no parque.

A imagem obtida pela câmara direita foi a seguinte:

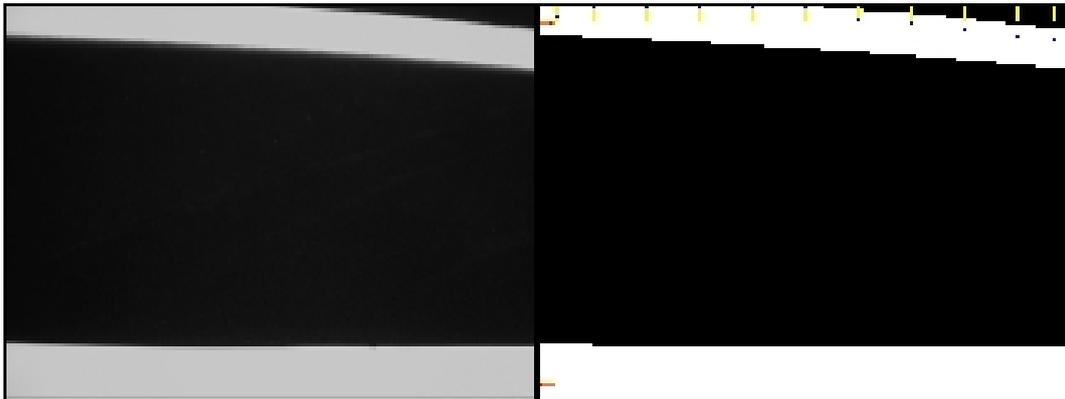


Figura 54. Passagem no parque. a) Imagem normal. b) Imagem binarizada.

Como se pode ver, foram capturadas duas linhas sendo que a encontrada foi a que pertence ao parque e por isso devia ser negligenciada.

Estas imagens são incompreensíveis nos termos da lógica desenvolvida. Aqui se obtém uma prova cabal da importância da utilização das duas câmaras. Seriam as câmaras direita, no primeiro caso, e a esquerda no segundo, as únicas disponíveis para tomar uma decisão correcta da direcção a tomar. Por este motivo é vital utilizar ferramentas que desqualifiquem uma imagem incoerente, de forma a que o algoritmo

seja forçado a aceder ao conselho da outra imagem. São estas ferramentas que se pretendem caracterizar neste capítulo.

Como já se disse foram calculados os ângulos entre cada dois pontos de linha consecutivos encontrados. Uma ferramenta de validação consistia em verificar uma mudança muito elevada entre os ângulos calculados. Imagine-se o ângulo entre o primeiro ponto de linha e o segundo era de trinta graus, enquanto que o cálculo seguinte indicava um ângulo de -50 graus. Obviamente a imagem era imediatamente desclassificada por não ser coerente com as características conhecidas da pista (não existem variações tão bruscas de ângulos na pista). Posteriormente, calcula-se o desvio padrão de todos os ângulos calculados, sendo que, para que a imagem seja validada este tem que ser inferior a um determinado valor.

Outra ferramenta utilizada para encontrar a linha analisa a sua espessura. Um pixel branco é considerado linha se e só se for seguido de um número configurável de pixels brancos. Com isto evita-se que algum ruído na imagem seja considerado um ponto de linha.

Resumindo, podem considerar-se as seguintes fases na tarefa de navegação na pista:

- Obtenção da imagem da câmara para uma variável,
- Binarização da imagem,
- Conversão da informação obtida para uma matriz bidimensional,
- Procura vertical e horizontal de linha,
- Validação da imagem,
- Calculo dos ângulos,
- Calculo da média de ângulos,
- Calculo do desvio padrão,
- Assinalar com cores pixels chave para o processo (opcional),
- Atribuição dos valores calculados à estrutura preparada para o efeito.

Estes processos são maioritariamente cálculos que analisam a imagem sob vários pontos de vista, e obtêm como resultado final um valor de ângulo de viragem com base naqueles. Inicialmente executa-se a binarização, da qual todo o processo vai depender (daí a importância da escolha de um limite de binarização adequado). Posteriormente, é na procura de linha que reside a importância do processo.

6.2.6 Perspectiva Global de Funcionamento do Algoritmo Principal

O algoritmo principal contém todos os outros e o seu formato final pode ser considerado complexo. No entanto, o conhecimento adquirido pela leitura dos capítulos anteriores facilita a sua compreensão. O algoritmo não é mais do que um contínuo processar de informações vindas dos sensores e respectivas tomadas de decisões. Estas acções ocorrem ciclicamente até que sejam verificadas condições de terminação, normalmente o número total de passagens na passadeira. Na seguinte figura pode ver-se um diagrama de fluxo do algoritmo que é processado em cada prova. De referir que, consoante a prova a realizar, algumas componentes do seguinte esquema podem ser desactivadas, tais como por exemplo a leitura dos sensores de infravermelhos numa prova sem túnel. Com a seguinte figura pretende-se dar uma ideia de como funciona o algoritmo e não representar fielmente os passos executados em cada uma das mangas. No entanto, se se quiser compreender o esquema como tal, ele representa fielmente o algoritmo utilizado na terceira e última manga, pois engloba as funções responsáveis pela leitura de semáforos e pela navegação no túnel / zona de obras.

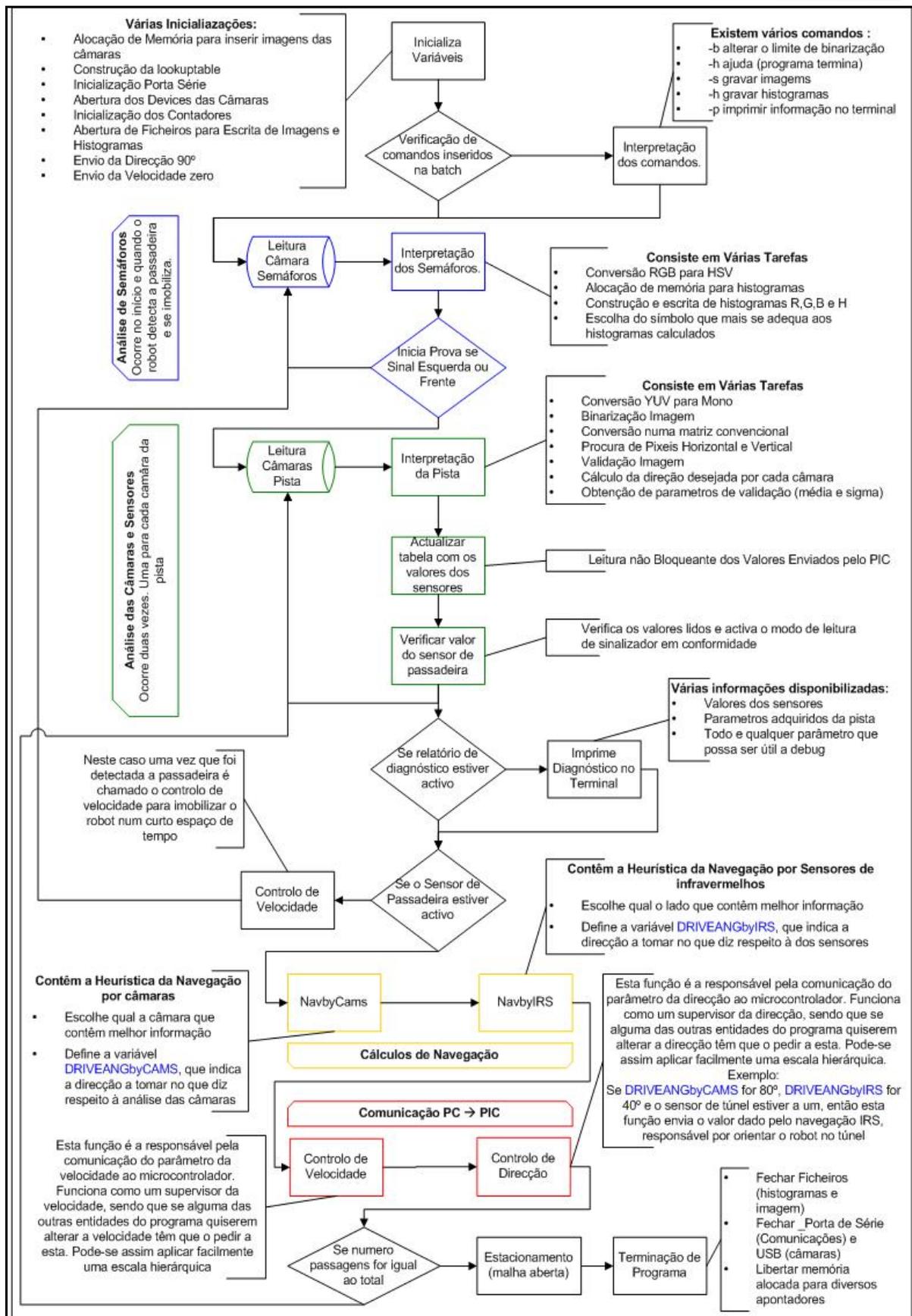


Figura 55. Esquema do funcionamento do algoritmo de navegação.

7 PARTICIPAÇÃO NO ROBÓTICA 2004

A participação no Festival Robótica 2004 decorreu no fim-de-semana de 23 a 25 de Abril de 2004. Os resultados obtidos pelo ATLAS II, discriminados por ronda e total, podem ser verificados na seguinte tabela, retirada do sítio oficial do festival. Este pode ser acedido em www.robotica2004.org.

Tabela 7. Classificações parciais e final.

Classe Condução Autónoma (CA)						
Clas.	Equipa	Instituição	1ª Ronda	2ª Ronda	3ª Ronda	Final
1	Made in Agueda	Esc. Sup. de Tec. e Gest de Águeda	35	66	71	172
2	Runner	ISEP/LSA	63	211	200	474
3	ATLASII	U. Aveiro - Departamento Mecânica	74	140	556	770
4	G-Force	IDMEC - Instituto Superior Técnico	157	200	597	954
5	Rasteirinho	IDMEC - Instituto Superior Técnico	138	246	627	1011
6	Spidwheel	ISEP/LSA	58	520	630	1208
7	Sergent Pepper	ISLA - Gaia	398	293	614	1305
8	Feup-Raly	FEUP	155	831	750	1736
9	ROBOTOP	ISLA - Gaia	605	610	631	1846
10	ANODUA	U. Aveiro - Departamento Mecânica	548	831	616	1995
11	DEIUMA	Universidade do Minho	782	831	750	2363
12	Góllas	CINEL	782	831	750	2363
13	turISTA	ISR - Instituto Superior Técnico	782	831	750	2363
14	BotRacer	FEUP/LIAC/ junifeup	782	831	750	2363

Como se pode ver, foi obtido o terceiro lugar da geral. Esta posição corresponde a uma melhoria em relação ao ano anterior, dado que o resultado obtido em 2003 foi o quarto lugar com um total de 882 segundos.

Infelizmente, aquando do festival, não estava ainda a funcionar a identificação de cor, facto pelo qual a análise dos semáforos funcionou mal. Isto aumentou em grande medida o tempo total obtido.

Ocorreram alguns problemas durante o festival, podendo-se no entanto, fazer um balanço global muito positivo, não tanto pelo lugar obtido mas muito mais pelo convívio com este ambiente e pela partilha de conhecimentos com outros colegas que também estiveram a preparar um robot um ano inteiro, e pelo gosto adquirido pela robótica.

8 CONCLUSÃO

Muitos factores contribuíram para o desenrolar deste projecto. Contudo, algumas decisões foram fulcrais para aquele. Comentam-se agora essas decisões, fazendo uma análise à posteriori dos seus resultados e outras implicações.

8.1 SOLUÇÃO MECÂNICA DE MOBILIDADE

Concebida e já utilizada pelos nossos colegas no ano passado, a solução utilizada para conferir mobilidade ao robot é, apesar de única entre os vários participantes, bastante eficaz. Facilita a escolha de uma direcção, de forma rápida e eficaz. O controlo da direcção a nível do PIC pode ser melhorado.

8.2 ALTERAÇÃO DO SISTEMA OPERATIVO / PROGRAMAÇÃO EM C

Esta revelou-se uma decisão muito bem tomada. De facto o *Linux* dispõe de muito mais potencial para aplicações robóticas, quando comparado com o *Windows*. Desde a utilização de três câmaras, à comunicação com o PIC, todos os processos foram simplificados com a utilização deste sistema operativo. Em defesa desta opinião, pode dizer-se que, os vencedores desde há muitos anos desta classe utilizam *Linux*. Obviamente, a aprendizagem necessária para trabalhar neste sistema operativo implica alguns meses de trabalho.

Quanto à programação em *c*, tem-se também uma opinião muito positiva, dado que é de execução muito mais rápida do que as possíveis alternativas (*MatLab* ou *Visual Basic*) o que é um factor fundamental para esta aplicação. Deve dizer-se no entanto que a aprendizagem desta linguagem é bastante difícil por ser diferente do habitualmente utilizado e de mais baixo nível.

Com este sistema operativo e esta linguagem de programação o potencial do robot aumenta exponencialmente.

8.3 MICROCONTROLADOR

O microcontrolador revelou-se muito mais eficaz do que a carta de aquisição utilizada pelo robot predecessor. A escrita de sinais analógicos é mais rápida, a comunicação também. Acresce ainda o facto de ser possível efectuar pré-processamento aos dados adquiridos sem sobrecarregar o processador com este processo. Pode ainda processar-se as ordens recebidas e convertê-las nos sinais eléctricos necessários à actuação dos motores. Agora é também possível fazer uma separação consistente e eficaz entre o *software* e o *firmware*.

Concluindo, a utilização do microcontrolador é sem dúvida um passo em frente na busca de um robot mais eficaz, mais rápido, e melhor organizado ao nível de distribuição de tarefas. Fica a sugestão de migrar para um microcontrolador superior, talvez da série 18, pois todos os pinos foram utilizados e a velocidade de processamento era baixa para as necessidades.

8.4 DUAS CÂMARAS PARA SEGUIMENTO DE PISTA

Opção sem dúvida acertada. A redundância subsequente da utilização de duas câmaras é vital para compensar eventuais desvios da posição ideal na pista, imagens por algum motivo inválidas, entre muitos outros factores. É, sem dúvida, uma solução a manter.

8.5 CONFIGURAÇÃO DAS CÂMARAS

Avaliar esta opção é difícil, por não ser possível saber o resultado de uma outra escolha. Pode, no entanto dizer-se que, para o tempo que estava disponível para o desenvolvimento do algoritmo, esta opção foi acertada. Facilita a análise das imagens obtidas e a distorção existente é menor, ou até negligenciável, pois acontece numa direcção não importante. Esta solução revelou, posteriormente, um problema não antecipado. Aconteceu que, sem qualquer informação acerca da passadeira que não a do sensor de infravermelhos, o robot não se imobilizava a tempo, incorrendo em penalizações. Se fosse outra a configuração, seria possível detectar a passadeira à distância e diminuir a velocidade em conformidade. Com a solução utilizada, foi necessário executar essa diminuição em malha aberta, com todas as desvantagens que esta metodologia acarreta.

Uma palavra para os suportes das câmaras que não eram os mais adequados por serem frágeis e desafinarem facilmente.

8.6 CÂMARA PARA ANÁLISE DE SEMÁFOROS

Apesar desta opção não estar a funcionar eficazmente aquando do festival, em teoria é a única opção possível. Contém muito trabalho a nível de processamento de imagem, reconhecimento de cor e talvez até de forma. Sugere-se talvez a utilização de um sistema que possibilite a mobilidade da câmara, apontando-a para o sinalizador, qualquer que seja a configuração do robot quando este se imobiliza na passadeira. Esta opção facilitaria o reconhecimento de forma e talvez até auxiliasse a navegação em situações críticas.

8.7 SENSORES INFRAVERMELHOS

A utilização destes sensores revelou-se uma opção correcta por simplificar a navegação no túnel e zona de obras. Conceptualmente o algoritmo de navegação é muito simples, o que faz com que se transforme uma desvantagem num trunfo. Quer isto dizer que, para outros navegar no túnel e zona de obras é uma dificuldade adicional (identificação da situação e orientação por marcos distintos), enquanto que a utilização destes sensores simplifica o problema, tornando a navegação nestes obstáculos inclusivamente mais fácil do que a navegação na pista.

8.8 DIFICULDADES ENCONTRADAS

Importa agora referir algumas dificuldades encontradas de forma a que, caso o projecto seja executado para o ano, estas possam rapidamente resolvidas na medida do possível.

Uma grande dificuldade prende-se com a pista de testes onde foram feitos todos os ensaios. O piso do Laboratório de Automação e Robótica (LAR) reflecte em demasia a luz, sendo muito complicado encontrar um limite de binarização adequado (em algumas alturas do dia, com a incidência do sol, este nem sequer existe). Seria este problema

solúvel com a aplicação de um tapete opaco na zona da pista ou com cortinas de outro formato.



Figura 56. Pista de testes.

Compreende-se que seja talvez impossível melhorar as condições disponíveis da forma sugerida, até porque o atlas não é o único robot a ser desenvolvido naquele laboratório e porque a maioria dos concorrentes não tem as condições tão vantajosas mas, pela impossibilidade de afinar correctamente o algoritmo de navegação na pista de testes, pensa-se que era importante referir este aspecto.

As tecnologias utilizadas (microcontroladores, programação em *c*), levam bastante tempo na aprendizagem, apesar de serem muito mais eficazes do que as alternativas.

Foi ainda desenvolvida uma técnica de travagem para o robot. Esta consistia em enviar para o servoamplificador um diferencial de tensões negativo. Assim o motor da tracção iria travar o robot por estar a tentar andar para trás. Esta técnica foi desenvolvida pelo facto de as câmaras não devolverem nenhuma informação relativa à aproximação da passadeira e assim se tornar imperativo imobilizar no menor espaço de tempo possível.

8.9 CONTINUIDADE DO PROJECTO

Reserva-se este capítulo para expor a opinião pessoal dos alunos acerca do potencial do ATLAS II no futuro. Deveria o projecto continuar? Foi atingido o potencial total? O ATLAS II é um robot com as soluções de fundo adequadas, mas que ainda carece de muito desenvolvimento, sobretudo a nível de *software*. Todas as mudanças efectuadas impossibilitaram a criação de um algoritmo de navegação realmente robusto. Deve dizer-se que, em alguns ensaios o ATLAS II conseguiu inclusivamente fazer a pista à velocidade máxima, apesar destes terem sido esporádicos e não oferecerem uma solução fiável. Com algum trabalho pode facilmente ombrear com os melhores da classe, pelo acerto das soluções de fundo implementadas. Para os alunos é muito aliciante participar num projecto destes, pela variedade de matérias que abrange e pela recompensa que é poder participar no festival, colocando em prática o trabalho realizado durante um ano.

9 BIBLIOGRAFIA

Websites

- [1] fóruns Linux, <http://forums.devshed.com/>
- [2] red hat Linux Homepage, <http://www.redhat.com/>, suporte Linux
- [3] Microchipc Homepage, <http://www.microchipc.com/>, *Program Microchip PIC micros with C*
- [4] MSDN Homepage, <http://msdn.microsoft.com/>, suporte para produtos da *Microsoft*
- [5] The Mathworks homepage, <http://www.mathworks.com/>, suporte *Matlab*
- [6] Festival Robótica2004 Homepage, <http://www.robotica2004.org>
- [7] Imaging Hardware Homepage, <http://www.imaginghardware.com>, Componentes para dispositivos de imagem e suporte
- [8] efg homepage, <http://www.efg2.com>, suporte para processamento de imagem e cor

Livros

- [1] W. K. Pratt, *Digital Image Processing*, 3rd ed., Wiley-interscience, New York, 2001
- [2] J. C. Russ, *The Image Processing Handbook*, 2nd ed., CRC Press, Boca Raton, FL, 1995
- [3] Luis Damas, *Linguagem C*, 8^a ed., FCA, Lisboa, 1999
- [4] J. L. Silva, J. M. Gomes, *Desenvolvimento de um Robot para participação no Robótica2003*, 2003
- [5] E. R. Davis, *Machine Vision: Theory, Algorithms, Practicalities (Signal Processing and Its Applications Series)*, 2nd ed., Academic Press, 1997
- [6] J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1993, Londres

Ficheiros PDF

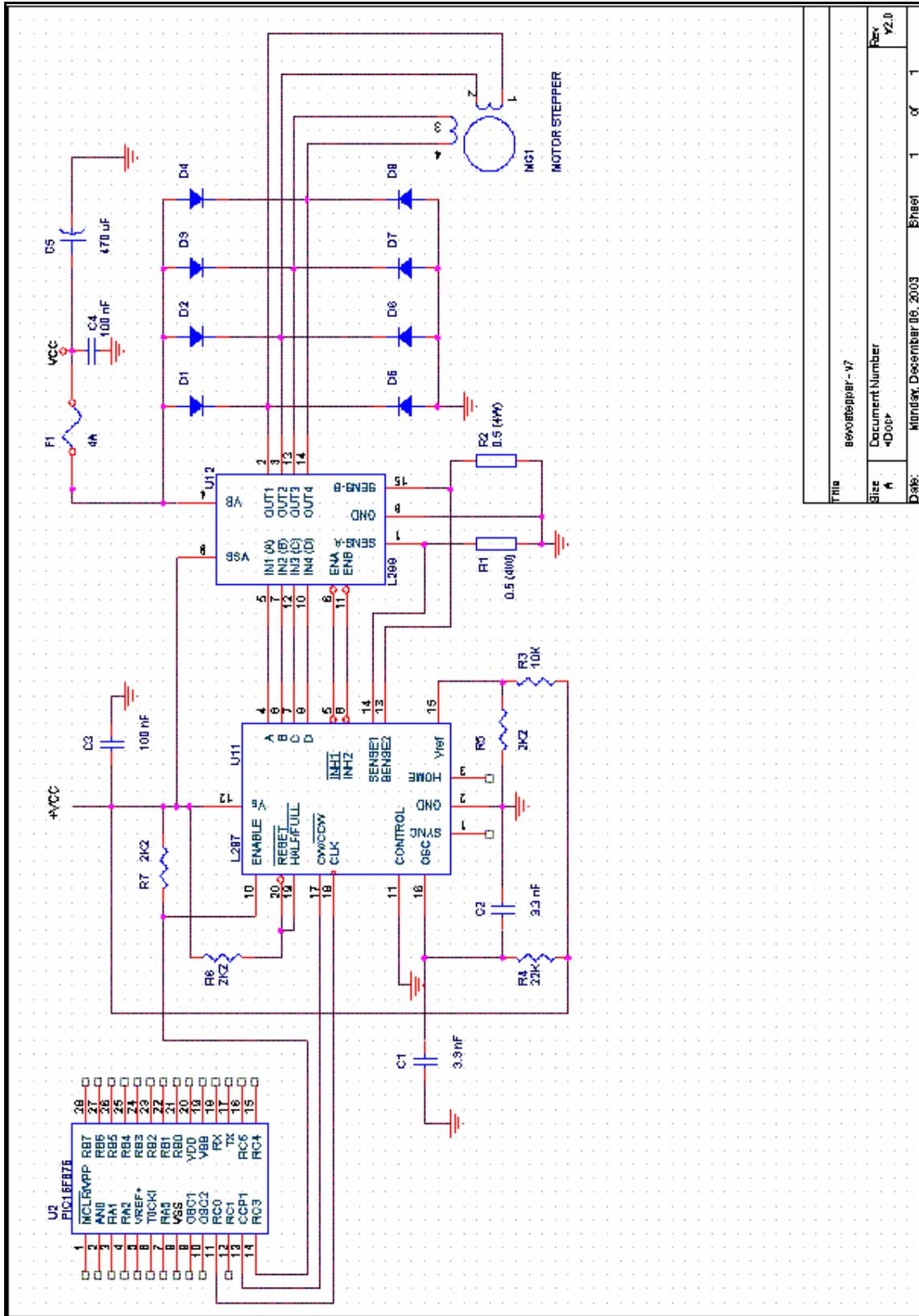
- [1] D. Collins, G. Wyeth, *Cerebellar Control of a Line Following Robot*, University of Queensland (Australia)

-
- [2] A. Kamga, A. Rachid, *A Simple Path Tracking Controller For Car-Like Mobile Robots*, Laboratoire des Systèmes Automatiques, Amiens (France)
- [3] A. Pozo-Ruz, C. Urdiales, A. Bandera, E. J. Pérez, F. Sandoval, *A Path Tracking Method For Autonomous Mobile Robots Based On Grid Decomposition*, Dpto. Tecnologia Electrónica. Universidade de Málaga (Espanha)
- [4] M. Kumano, A. Ohya e Shin'ichi Yuta, *Obstacle Avoidance of Autonomous Mobile Robot Using Stereo Vision Sensor*, Intelligent Robot Laboratory, University of Tsukuba, Ibaraki (Japan)
- [5] A. Jacoff, E. Messina e J. Evans, *Experiences in Deploying Test Arenas for Autonomous Mobile Robots*, Intelligent Systems Division, National Institute of Standards and Technology, Gaithersburg, MD
- [6] O. Lorch, R. Cupec, J. Denk, J. F. Seara, F. Freyberger e G. Schmidt, *Walking Robots – Vision and Locomotion*, Institute of Automatic Control Engineering, Technische Universität München
- [7] C. J. Taylor, D. J. Kriegman, *Vision-Based Motion Planning and Exploration Algorithms for Mobile Robots*, IEEE Transactions on Robotics and Automation, 1999.
- [8] Microchip, *Product Catalogue*, 2004

ANEXOS

ANEXO I

Esquema da Placa de Direcção



Title	servostepper-v7
Size	A
Document Number	4006
Date	Monday, December 06, 2003
Sheet	1 of 1
Rev	v2.0

ANEXO III

Software Utilizado durante o projecto

	Programa	Função
Windows	MatLab	Programação
	HyperTerminal	Comunicação RS232
	Kermit	Comunicação RS232
	DevC++	Editor de código C
	MPLAB	Compilador de código C
	PIC downloader	Descarregar código para o PIC
	Icprog	Descarregar Bootloader para o PIC
	Orcad –Capture CIS	Construção de Circuitos eléctricos
	Visio	Construção de diagramas
	Antenna Web Design Studio	Construção de websites
	Xcursion	Visualização de aplicações de um PC remoto Unix em ambiente windows
	Video Capture	Captura de vídeo com webcam
	SnatchIt	Captura de <i>frames</i> de videos
	Paragorn ext2fs	Recuperação de ficheiros
Linux	Telnet	Emulador de ligações TCP/IP
	Camstream	Programa para webcam
	gcc	Compilador
	Kwrite	Editor de texto
	Xgraph	Construção de gráficos
	Xview	Visualização de imagens
	Telnet	Emulador de ligações TCP/IP