



universidade de aveiro
theoria poiesis praxis

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA

Curso: [8240] Mestrado Integrado em Eng. de Computadores e Telemática

Disciplina: [47137] Introdução à Eng. de Computadores e Telemática

Ano lectivo: 2012/2013

Relatório da Aula Prática 10

Programação do Robô DETI PIC

Autores:

[68535] Bruno SILVA

[68799] Rui OLIVEIRA

Turma/Grupo:

T5B / Prática 3

Docentes:

André ZÚQUETE

João BARRACA

Data:

19 de Dezembro de 2012

Resumo:

Pretende-se através deste relatório expor sob forma escrita, o nosso desempenho e objetivos alcançados na aula de Introdução à Engenharia de Computadores e Telemática sobre a programação do robô DETI PIC, desenvolvido pelo Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Índice

0.1	Introdução	3
0.2	Descrição do problema	6
0.3	Aparelhagem e equipamento	10
0.3.1	Robô DETI PIC	10
0.3.2	Software DETInchanting	10
0.3.3	Materiais utilizados	11
0.4	Procedimento	12
0.4.1	Problema 1	12
0.4.2	Problema 2	13
0.5	Resultados	14
0.5.1	Problema 1	14
0.5.2	Problema 2	15
0.5.3	Problemas 3, 4 e 5	16
0.6	Análise dos Resultados	17
0.6.1	Problema 1	17
0.6.2	Problema 2	18
0.7	Conclusões	19

0.1 Introdução

Como sabemos, um robô é um dispositivo, ou conjunto de dispositivos, eletromecânicos ou biomecânicos capazes de realizar uma determinada funcionalidade de forma independente, para isso terá de ser pré-programado, ou então controlado por um ser humano. Primordialmente os robôs foram programados para desenvolver trabalhos de baixa complexidade, como por exemplo, deslocarem-se sobre superfícies planas, interagir com obstáculos, entre outros. Atualmente, os robôs realizam tarefas muito completas, e em muitos dos casos substituindo o trabalho humano. Embora, nem sempre se tira partido de todas as capacidades do robô.

Neste relatório pretendemos demonstrar o nosso desempenho na programação do robô DETI PIC, desenvolvido pelo Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro. Mais à frente iremos mostrar as principais características do robô, e alguns dos equipamentos e materiais que utilizámos para conseguir concretizar esta actividade. Para a programação de robô utilizando um software, também este desenvolvido pelo DETI, chamado *DETInchanting* criado a partir de linguagem JAVA e C++ e adaptado do *Enchanting*, desenvolvido para robôs da LEGO¹. Neste caso, necessitámos de ter conhecimentos básicos de programação em JAVA, para conseguir testar as capacidades do robô, através da implementação blocos que permitem executar operações do tipo: *if... else...; forever... ; repeat until... ;* entre outros.

Ambicionamos com estes testes ao robô, neste relatório, continuar a explorar a movimentação do robô usando tanto os sensores de luminosidade, colocados na sua parte inferior, como os sensores de distância. Iremos também utilizar os quatro led's que o nosso equipamento possui, de modo a indicar valores de variáveis usadas pelo programa.

No problema 1, que se encontra apresentado na próxima secção, iremos preparar o robô de modo a desenvolver um programa "base" que irá sustentar os restantes problemas. Para a construção desta estrutura base, necessitamos de ter alguns conhecimentos básicos sobre numeração binária.²

Na tabela seguinte encontra-se listada a conversão de decimal para binário, que teremos que saber para resolver alguns dos problemas propostos.

¹Empresa conceituada no fabrico de brinquedos, e atualmente no desenvolvimento de robôs.

²Sistema de numeração posicional baseado nos numeros 0 e 1.

Numeração decimal	Numeração binária
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Tabela 1: Conversão de decimal para binário e vice-versa.

Para além da conversão, de 0 a 15, que necessitamos de conhecer, neste caso para confirmar os valor que nos irão ser apresentados através do robô, teremos que saber a forma generalizada de como calcular o valor binário de um determinado número decimal. Para isso devemos seguir os seguintes passos:

1. Dado um determinado número decimal;
2. Dividir esse número por 2;
3. Observar o resto que resulta desta divisão;
4. Dividir o resultado da divisão anterior novamente por 2;
5. Repetir o último passo sempre que seja possível a divisão;
6. Ordenar os "vários restos" da divisão por ordem decrescente do seu cálculo.

A figura seguinte representa o processo anteriormente descrito.

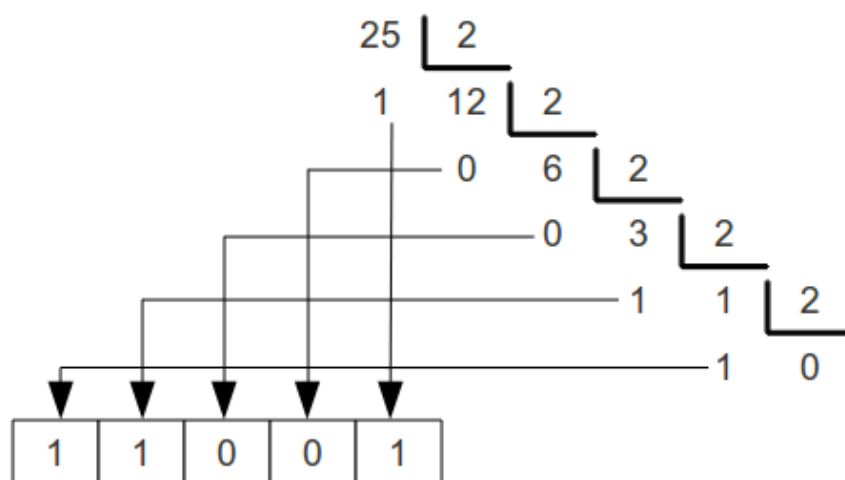


Figura 1: Determinar o binário de um número decimal.

0.2 Descrição do problema

Pretendemos através deste relatório dar resposta, programando e observando o comportamento do robô DETI PIC de modo a solucionar os seguintes problemas apresentados:

- **PROBLEMA 1** - Fazer um bloco que, dado um valor entre 0 e 15 mostre este valor em binário através dos quatro leds do robô.
- **PROBLEMA 2** - Neste exercício pretende-se descobrir o valor de Δ que permite equilibrar a velocidade efetiva dos dois motores do robô quando os mesmo são programados com uma velocidade base V (ou seja, um será programado com velocidade V , outro com $V + \Delta$). O valor de DELTA nunca é negativo, pelo que é preciso conhecer qual é o motor mais rápido.

Programar o robô de forma a ele seguir uma linha cuja largura é inferior à distância entre os dois sensores em torno do central (LEFT e RIGHT, no *DETInchanting*). O robô deverá seguir em frente até que qualquer dos sensores de linha, para além do central, indiquem a presença de linha. Nessa ponto deverá parar os motores (mas não o programa). O mesmo deverá acontecer no final da linha.

Seguidamente, vamos maximizar o percurso realizado pelo robô através do equilíbrio dos seus motores. Para isso vamos usar os botões de pressão para aumentar ou diminuir a diferença de velocidades entre os motores. O processo de equilíbrio dos motores deverá ser feito da seguinte forma, **sem nunca se alterar o programa**:

- Os motores são inicialmente programado com uma velocidade igual.
- O robô deveá iniciar o seu percurso após sentir um obstáculo perto de um dos sensores de distância e só deverá terminar nas circunstâncias indicadas no início do exercício.
- Após a paragem, a diferença Δ entre as velocidades dos motores pode ser alterada de forma a maximizar o percurso realizado. Assumindo que $\Delta = x \cdot \delta$, $x \in [0,7]$ e $\delta \in [0,7]$, a alteração deverá ser feita usando os botões de pressão:

- Pressão curta no botão preto: $x \leftarrow x + 1$
- Pressão longa no botão preto: $x \leftarrow x - 1$
- Pressão curta no botão vermelho: $\delta \leftarrow \delta + 1, x \leftarrow 0$;
- Pressão longa no botão vermelho: $\delta \leftarrow \delta - 1, x \leftarrow 0$;

Este processo permite experimentar valores de Δ desde 0 até 49, mas não todos os valores inteiros entre 0 e 49. No início da execução do programa tanto x como δ devem ter um valor nulo. Os valores de x e δ devem ser mostrados nos leds do robô, intervalados de 1 segundo. Como só existem 4 leds, um deles (D5) deverá ser usado para indicar qual o valor que se está a observar (desligado – δ , ligado – x), e os demais para mostrar o valor (em binário).

- O robô deverá ser novamente colocado no início da linha a percorrer e os motores deverão ser iniciados tal como no início desta sequência de passos, usando os sensores de distância.

- **PROBLEMA 3** - Alterar o programa anterior para mostrar a carga da bateria, no ecrã do computador (via *pterm*), no início da execução do programa (antes de fazer qualquer percurso). Repetir o exercício anterior para várias velocidades base do robô, de modo a obter uma curva de calibração em função da velocidade base. Esta curva deverá ser apresentada no relatório, bem como o número do robô no qual ela foi observada e a carga da bateria no início da calibração.

NOTA: a carga da bateria influencia a calibração. Ou seja, se a bateria tiver uma carga diferente, a curva de calibração será igualmente diferente.

- **PROBLEMA 4** - Uma vez conhecido o Δ apropriado para cada velocidade base, faça um programa que o explore e que siga a linha até ao seu término mas corrigindo a trajetória para maximizar o alinhamento com a linha. Este programa deverá usar os sensores de linha mais exteriores (FAR LEFT e FAR RIGHT) para detectar a presença de linhas com informação. Sempre que as detectar deverá acender um led correspondente (D5 e D2, respectivamente).

- **PROBLEMA 5** - Alterar o programa anterior para transformar o robô num dispositivo de leitura de informação sob a forma de linhas pretas impressas (algo similar a um código de barras). As linhas observadas pelo sensor far left fornece o sinal de leitura de informação, qual deverá ser obtida usando o sensor far right. Deste modo, sempre que existir um linha preta à esquerda da linha de deslocamento deverá ser obtido um bit, 1 ou 0, consoante à direita haja ou não uma linha preta (ver figura abaixo).

Durante o deslocamento do robô os leds devem apresentar a seguinte informação:

- **Lido bit 0:** apaga todos os leds durante 1 segundo.
- **Lido bit 1:** acende todos os leds durante 1 segundo.

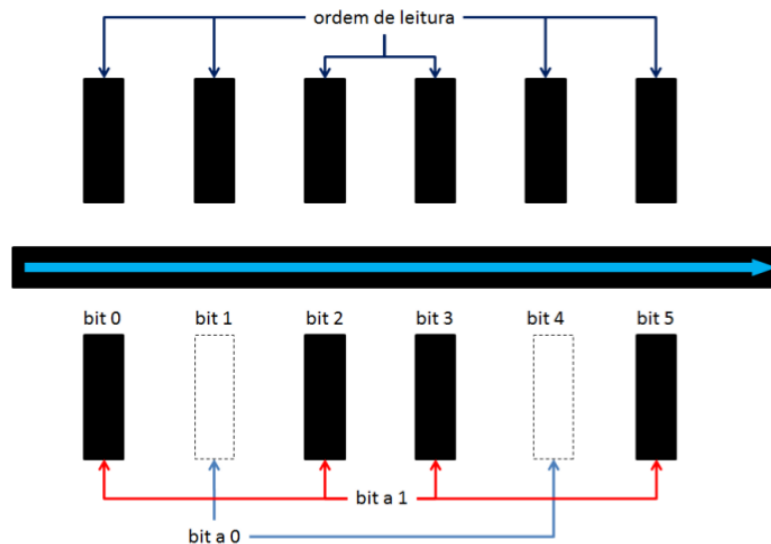


Figura 2: Diagrama de um cenário possível para o Problema 5

- **Demais situações:** acende os dois leds centrais (D3 e D4).

O robô não deverá parar o seu movimento enquanto estiver a mostrar, nos leds, a informação lida.

NOTA: por cada ordem de leitura dada por uma linha à esquerda não deverá ser obtido mais do que um bit, mas poderão ser feitas diversas leituras do sensor para determinar esse bit.

Para além dos cinco problemas acima descritos, pretendemos também adquirir alguns dos conhecimentos introdutórios do funcionamento e programação da robótica utilizando para isso o *DETinchanting*, programa este que iremos abordar mais à frente. Dado que nos foi imposto a realização deste relatório utilizando a linguagem tipo \LaTeX , pretendemos também desenvolver as nossas competências a nível desta linguagem, alargando assim os nossos conhecimentos já adquiridos em aulas anteriores.

0.3 Aparelhagem e equipamento

0.3.1 Robô DETI PIC

A nossa base de trabalho é o robô DETI PIC. Este robô é constituído por dois motores DC³ sem realimentação, três sensores de distância frontais cobrindo um ângulo de aproximadamente 45 graus para cada lado do robô, cinco sensores de brilho na parte inferior do robô e quatro led's na parte superior. O robô dispõe ainda de dois botões, um interruptor e uma porta USB. Todo este robô foi desenvolvido de forma prática e intuitiva à introdução à programação deste tipo de máquinas. Para fazer o upload entre o robô e o computador utilizá-mos um cabo USB⁴ 2.0.

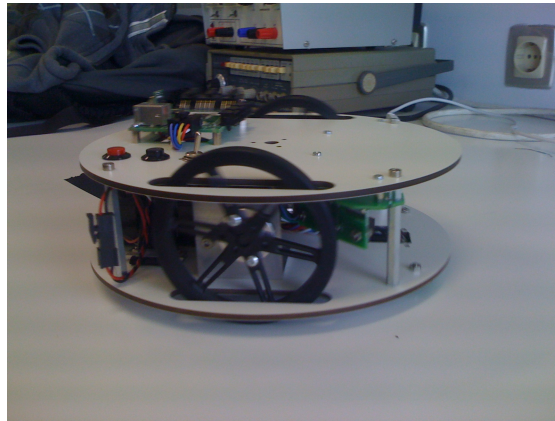


Figura 3: Robô DETI PIC

0.3.2 Software DETInchanting

O programa usado para a programação de instruções a dar ao robô neste trabalho foi o *DETInchanting*, uma adaptação do original *Enchanting* usado para programar robôs da LEGO. Este interface foi desenvolvido na Universidade de Aveiro com a intenção de facilitar a programação dos robôs utilizando um interface gráfico que se baseava em encaixes de blocos de código JAVA, na Figura 2 está representada o ambiente de trabalho deste programa. Este software usa um paradigma

³Corrente contínua

⁴Universal Serial Bus

gráfico do tipo Scratch⁵.



Figura 4: Ambiente do *DETInchanting*.

0.3.3 Materiais utilizados

O nosso ambiente de trabalho foi a bancada dos laboratórios do DETI, o que não nos dava muito espaço para manobrar o robô. Nestas condições usamos uma folha A3 com o esquema apresentado na Figura 2 da Descrição do Problema impresso. Este foi o único material que utilizamos para a resolução dos problemas apresentados, juntamente com os equipamentos já acima referidos.

⁵São interfaces gráficas que permitem que programas sejam criados através da sobreposição de blocos, tendo na sua base linguagens de programação.

0.4 Procedimento

0.4.1 Problema 1

Neste programa pretendemos fazer uma calibração ao robô para tornar o seu comportamento mais equilibrado e, de modo a servir como base para os restantes programas.

Começámos por criar uma função a que atribuímos o nome de “binLeds”. Esta função irá passar como argumento a variável “valor”, que foi posteriormente criada, tal como as variáveis “cont”, e “bit”. Após a criação da função, inicializámos um contador (“cont”) a zero, através do bloco **“set cont to 0”**, depois introduzimos um bloco que nos permite que todos os leds, nesta etapa do programa, estejam desligados. Depois criámos um ciclo repetitivo *“repeat until”* com a condição **“valor = 0”** ou seja a variável terá o seu início em zero. No final do ciclo, é incrementada uma unidade ao “cont”, de seguida tem-se que **“set bit to valor mod 2”**, nos irá permitir calcular o resto da divisão do valor, e **“set valor to valor / 2”**, nós permitirá determinar qual será o próximo valor a que teremos que determinar novamente o resto. Desta forma conseguimos determinar a numeração binária de um determinado número, tal como demonstrámos e exemplificámos na Introdução. Quando o resto da divisão “bit” é igual à unidade então, serão ligados os led’s correspondentes ao “cont”, devidamente incrementados, no caso da primeira interação não será ligado nenhum led.

No programa principal começámos por introduzir um bloco *“when clicked”* que dará início ao programa. Pelo facto de no início da criação do programa existirem problemas de sincronização dos leds, decidimos introduzir três blocos que nos permitem acender os mesmos durante 1 segundo, mais tarde verificámos que o problema estaria na função, que foi posteriormente corrigida.

Para finalizar introduzimos um ciclo repetitivo, no qual iremos “chamar” a função “binLeds” de modo a esta decorrer até ao “valor” 16, no final do ciclo estará presente um bloco que nos permitirá incrementar uma unidade à variável “valor” (daí a atribuição do valor 16 em vez de 15). Anteriormente ao ciclo *“repeat until”* inicializámos a variável “valor” a zero através do bloco *“set valor to 0”*.

0.4.2 Problema 2

Neste programa começámos com um ciclo *forever*, onde se encontra no seu interior um *if* que basicamente avalia a pressão efectuada no botão preto onde incrementa o x por uma unidade e de seguida altera o Δ (Delta) sendo ele o produto entre o valor actual de x com o δ (delta) . A próxima condição *if* avalia o mesmo procedimento mas para o botão vermelho. O código não pode ser acabado a partir deste ponto por falta de tempo, embora ainda assim tenhamos adicionado um ultimo *if* que fazia o robô andar com velocidade constante no motor esquerdo e a velocidade do motor direito ia sendo alterada de modo a que fosse igualar a do motor esquerdo.

0.5 Resultados

0.5.1 Problema 1

Através da implementação, criação e posterior upload do programa apresentado na Figura 5 e 6, conseguimos obdecer ao enunciado do Problema 1.

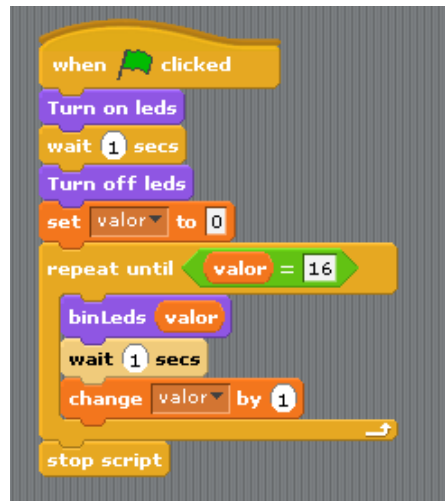


Figura 5: Programa relativo ao problema 1.



Figura 6: Função "binLeds" presente no programa 1.

0.5.2 Problema 2

Através da implementação, criação e posterior upload do programa apresentado na Figura 7, conseguimos obdecer, a parte do enunciado do Problema 2, embora com algumas dificuldades.

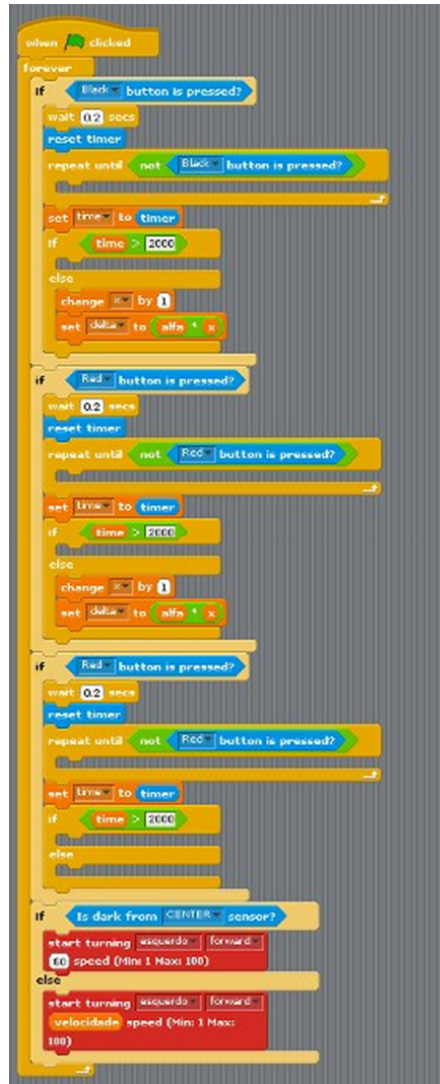


Figura 7: Programa relativo ao problema 1.

0.5.3 Problemas 3, 4 e 5

Devido à falta de tempo, não nos foi possível realizar e programar os algoritmos relativos aos problemas 3, 4 e 5, do guião a que equivale este relatório.

0.6 Análise dos Resultados

0.6.1 Problema 1

Inicialmente, neste problema pensávamos que poderíamos criar blocos para cada número decimal, ou seja de 0 a 15. Na verdade isso era possível, mas o programa iria tornar-se demasiado extenso e com possibilidade de dificuldade na execução. O nosso docente da aula prática solicitou-nos que criássemos uma função de modo que o programa se torna-se mais legível, e mesmo extenso. Com base na informação apresentada na Introdução (como determinar o número binário de um decimal), criámos o algoritmo apresentado na secção anterior. Embora tenhamos tido algumas dificuldade na implementação dos blocos, no final conseguimos atingir o resultados que pretendemos, sem grandes erros, estes foram posteriormente corrigidos. Na figura apresentada de seguida, está esquematizado o decorrer do programa em função do tempo. A cor amarelo (representado a luz acesa do led) e a branca (representado a luz apagada do led), correspondem respectivamente ao valor 1 e 0 da numeração binária. Conforme a Tabela 1 apresentada na Introdução observámos os mesmo resultados no robô, tal como esperávamos.

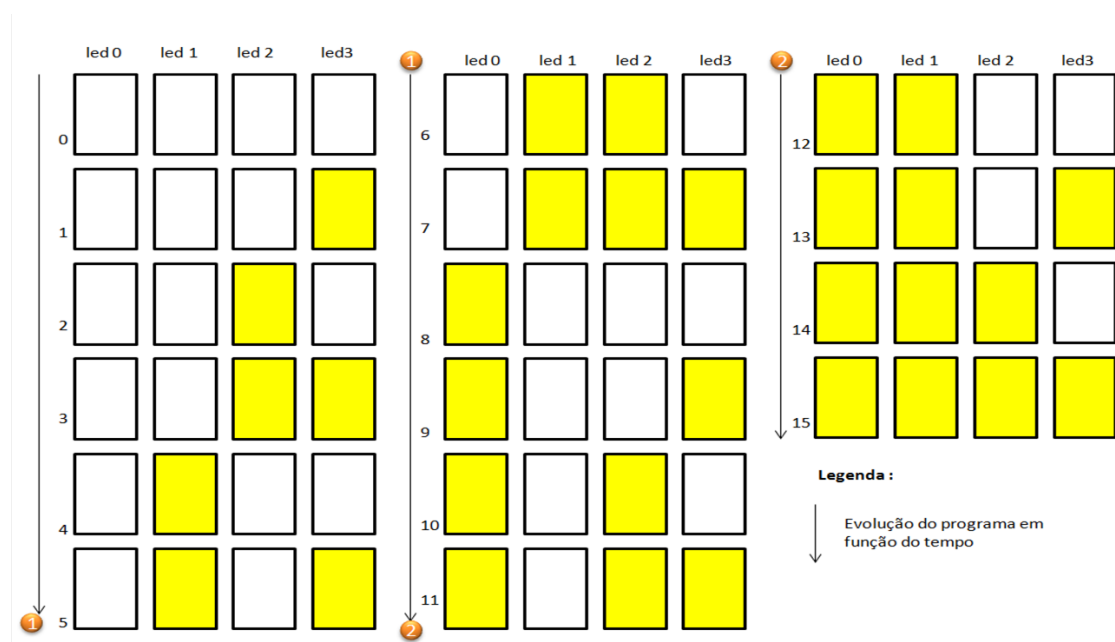


Figura 8: Esquema relativo ao Problema 1.

0.6.2 Problema 2

Este problema foi o que nos causou maior dificuldade um pouco pela complexidade mas também pelo tempo que tínhamos para a resolução. Com isto e devido muito à “pressa” de acabar o programa fizemos muitos erros que facilmente eram corrigidos ou mesmo evitados com mais tempo. Erros esses como o não ter atribuído a variável velocidade à velocidade inicial mais o Δ (delta) e depois ter usado essa variável no motor direito. Outro erro foi o ter colocado o mesmo motor a andar sendo um deles o direito e outro o esquerdo, e por último os motores deviam estar os dois dentro do ultimo *if* e não separados estando um no *else*. Concluindo não consideramos este programa como um resultado positivo mas achamos que embora tenha algum nível de complexidade poderíamos ter resolvido com um pouco mais de tempo.

0.7 Conclusões

Chegado ao final deste relatório, é nossa intenção efetuar uma retrospectiva da evolução do mesmo, tendo em conta os problemas que nos deparámos, objetivos, e principais metodologias utilizadas.

Apesar de não termos consigo testar e programar todos os problemas apresentamos na integra, devido ao fator tempo, pensamos que os problemas "base" foram bem conseguido, embora o ultimo com algumas dificuldades, que já frisámos em secções anteriores.

Achámos sem qualquer dúvida, que a programação do robô DETI PIC deu-nos bastante gozo e interesse a programar, dando-nos assim uma grande introdução aos conhecimentos sobre a programação em robótica. Na nossa opinião, esta disciplina foi uma mais valia não só devido a estes conhecimentos aprendidos, anteriormente citados, mas principalmente devido à introdução na aprendizagem da linguagem L^AT_EX.

Bibliografia

- [1] A.V.C.M. Zúquete , Site da Unidade Curricular - Introdução à Engenharia de Computadores e Telemática (18 Dezembro 2012). Diapositivos: robô DETI PIC [Online] *Available:* <http://www.ieeta.pt/avz/Aulas/IECT/12-13/docs/6-robo.pdf>

- [2] J. Barraca, D. Gomes, A. Zúquete , Site da Unidade Curricular - Introdução à Engenharia de Computadores e Telemática (18 Dezembro 2012). Guião da aula prática 10 [Online] *Available:* <http://www.ieeta.pt/avz/Aulas/IECT/12-13/docs/guiao-10.pdf>