



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 9

### Programação do Robô DETI PIC

---

*Autores:*

[68535] Bruno SILVA

[68799] Rui OLIVEIRA

*Turma/Grupo:*

T5B / Prática 3

*Docentes:*

André ZÚQUETE

João BARRACA

*Data:*

12 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 . . . . .	4
0.3	Aparelhagem e equipamento . . . . .	7
0.3.1	Robô DETI PIC . . . . .	7
0.3.2	Software DETInchanting . . . . .	7
0.3.3	Materiais utilizados . . . . .	8
0.4	Procedimento . . . . .	9
0.4.1	Problema 1 . . . . .	9
0.4.2	Problema 2 . . . . .	10
0.4.3	Problema 3 . . . . .	11
0.4.4	Problema 4 . . . . .	11
0.5	Resultados . . . . .	12
0.5.1	Problema 1 . . . . .	12
0.5.2	Problema 2 . . . . .	13
0.5.3	Problema 3 . . . . .	14
0.5.4	Problema 4 . . . . .	15
0.5.5	Problema 5 . . . . .	16
0.6	Análise dos Resultados . . . . .	17
0.6.1	Problema 1 . . . . .	17
0.6.2	Problema 2 . . . . .	18
0.6.3	Problema 3 . . . . .	19
0.6.4	Problema 4 . . . . .	20
0.6.5	Problema 5 . . . . .	21
0.7	Conclusões . . . . .	22

## 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 LEGO<sup>1</sup>. 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, explorar essencialmente os sensores de brilho, embora também iremos utilizar os sensores de distância. Através dos sensores de brilho iremos programar o robô de modo que este siga uma linha preta em diferentes perspectivas. Mais à frente neste relatório iremos explicar detalhadamente quais serão estas perspectivas.

---

<sup>1</sup>Empresa conceituada no fabrico de brinquedos, e atualmente no desenvolvimento de robôs.

## 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** - Programar o robo 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 *DETIchinting*). Nesta linha poderá aparecer um entrocamento que deverá ser ignorado. No final da linha o robô deverá parar, de acordo com a figura seguinte.



Figura 1: Esquema do problema 1.

- **PROBLEMA 2** - Alterar o primeiro programa de modo a que o robô pare durante um segundo, com os 4 leds acesos, quando encontrar o entrocamento. Após essa paragem deverá apagar os leds, retomando o seu movimento e não parar novamente no mesmo entroncamente.

- **PROBLEMA 3** - Alterar o programa do Problema 1 de modo a que o robô volte para trás no final da linha. (Ver Figura 2)

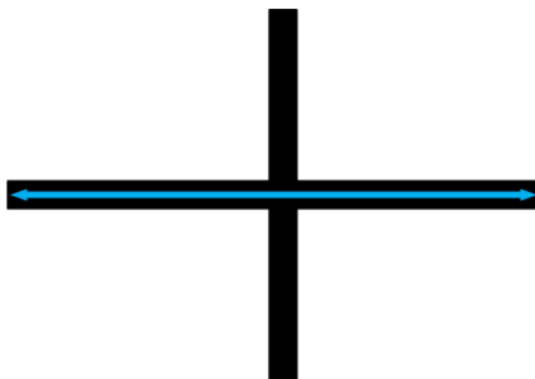


Figura 2: Esquema do problema 3.

- **PROBLEMA 4** - Alterar o programa anterior para que o robô siga a linha à direita sempre que encontra um entroncamento. (Ver Figura 3)

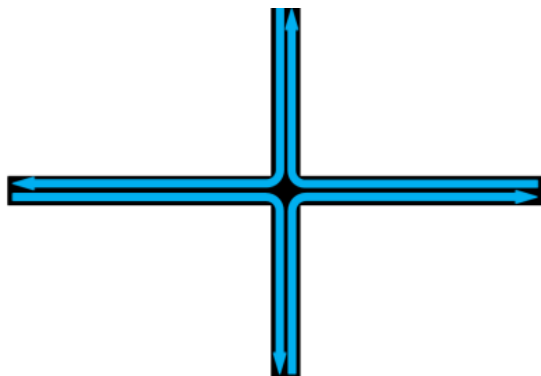


Figura 3: Esquema do problema 4.

- **PROBLEMA 5** - Alterar um dos quatro problemas acima mencionados, mas agora prevendo o facto de poder existir uma parede a interromper a linha. Neste caso o robô deverá abandonar a linha, seguir a parede e voltar a seguir a linha, abandonando a parede, quando reencontrar a primeira. (Ver Figura 4)

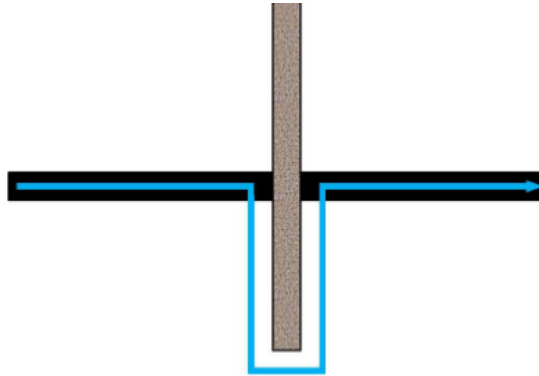


Figura 4: Esquema do problema 5.

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  $\text{\LaTeX}$ , 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 DC<sup>2</sup> 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 USB<sup>3</sup> 2.0.

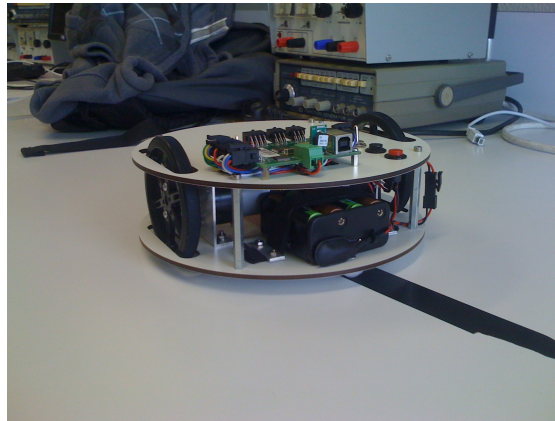


Figura 5: Robô DETI PIC

### 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

---

<sup>2</sup>Corrente contínua

<sup>3</sup>Universal Serial Bus



gráfico do tipo Scratch<sup>4</sup>.

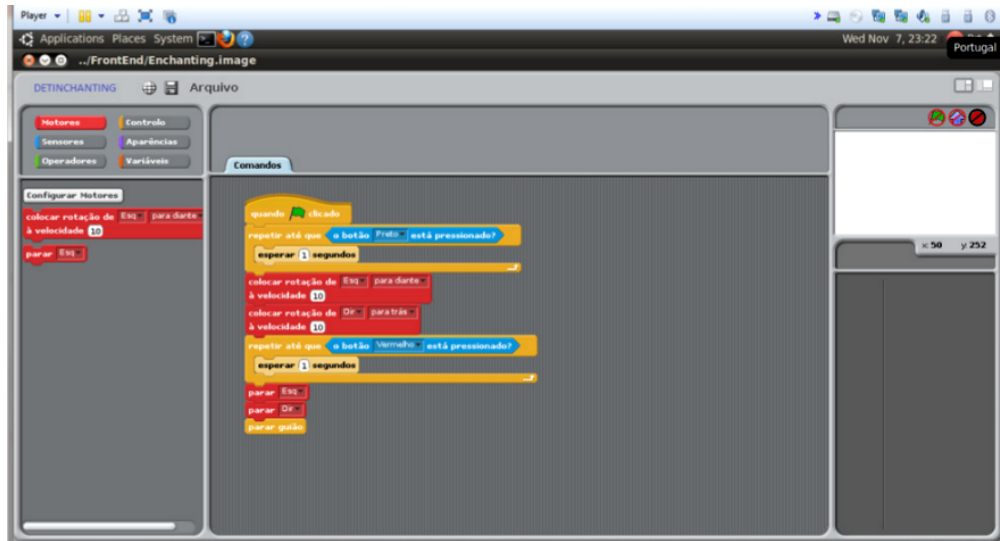


Figura 6: Ambiente do *DETInchanting*.

### 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 duas linhas de cor preta cruzadas entre si, para realização dos problemas propostos. Para a realização do Problema 5 utilizámos uma parede em K-Line.

---

<sup>4</sup>São interfaces gráficos 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 problema tal como nos restantes, começámos por introduzir um ciclo *repeat until*, que nos permite ao pressionar no botão red, que o programa seja executado.

Seguidamente, num ciclo repetitivo *forever*, introduzimos uma condição “Is dark from LEFT or RIGTH or CENTER sensor?”, esta opção irá possibilitar que o robô siga a linha escura, e caso se desvie(ou seja, se o sensor de brilho deixar de observar a cor preta) irá ser corrigida através da implementação de um bloco condicional. Por exemplo se a condição “Is dark from LEFT sensor?” se verificar, então o motor da roda esquerda irá parar e apenas o da direita se irá movimentar, de modo que a condição anteriormente descrita se deixe de verificar. O mesmo acontece no caso RIGTH, mas neste caso o movimento das rodas será trocado. Na situação “Is dark from CENTER sensor?” ambos os motores irão funcionar, ao mesmo tempo. Neste caso tivemos particular atenção à velocidade de cada roda, devido à folga que existe numa delas, por isso utilizámos a velocidade 60 e 55 para o motor esquerdo e direito, respectivamente.

Num outro ciclo *if* , caso se verifique a condição “Is dark from LEFT and RIGTH and CENTER and FAR LEFT and FAR RIGTH sensor?”, o dois motores irão deslocar-se em frente à mesma velocidade que anteriormente se movimentavam. Esta circunstância ocorre quando o robô encontra o entroncamento. Esta funcionalidade permite ao robô, quando os seus sensores de brilho mais laterais encontrarem o cruzamento, ou seja quando a condição anteriormente descrita se verificar, o robô irá ignorar o cruzamento, e seguirá em frente.

No final do problema, no *else* das condições acima descritas o robô irá parar (*stop all*), ou seja quando deixar de detetar qualquer linha preta nos seus sensores centrais, o robô irá por completo o seu percurso.

### 0.4.2 Problema 2

Para a resolução deste problema começámos por introduzir uma estrutura de repetição *repeat until* sempre que o sensor *far right* e o sensor *far left* detestassem escuro. Caso isso acontecesse, quando o sensor *center* detetasse a mesma condição, então o motor esquerdo e direita irão mover-se os dois, caso o sensor a detetar fosse o "right" então, a roda direita irá parar e apenas a do lado esquerdo irá movimentar-se, analogamente o mesmo acontece caso seja o sensor *left* a detetar presença da cor negra. A organização acima descrita permite que o robô corrija a sua posição em relação à linha, o mesmo aconteceu no programa 1.

Numa outra estrutura de estado, do tipo *if*, introduzimos uma condição que permite ao robô, ao encontrar o entroncamento, ou seja, caso o sensor *far right* e *far left* (estes localizados nas extremidade do robô) detete linhas escuras, então o robô irá parar (velocidade de ambos os motores será zero), e posteriormente irá ligar os seus led's durante 1 segundo.

De seguida, obtivemos um ciclo *forever* que permite ao robô, quando este chegar ao fim da linha que volte para trás. Para conseguir este sucedido procedemos da mesma forma tal como quando queríamos que o robô corrigir a linha no início do programa. Para finalizar o nosso projeto, voltamos a introduzir uma nova estrutura *if* que nos permite, quando o robô passar novamente pelo cruzamento irá ignorá-lo e seguirá em frente, ou seja quando "Is dark from LEFT and RIGTH and CENTER and LEFT and sensor? = false", então o motor direito e da esquerda irá seguir em frente.

Tal como no programa anterior, tivemos especial atenção à velocidade que atribuímos ao motor esquerdo e direito, pois devido à existência de uma pequena folga o robô desviava-se intencionalmente.

### 0.4.3 Problema 3

Neste programa começamos com um ciclo *forever* que vai acompanhar o resto do programa até ao fim contendo no seu interior uma sequência de 5 *if*'s acompanhados por os respectivos *elses*'s encadeados de forma a tornar o processo de processamento do código mais eficiente pelo robô, tentando com isto tornar o robô mais reactivo às diferentes situações que tinha de enfrentar. Os *if*'s estão delineados de forma a que o robô avalie as situações da melhor forma tentando neste caso ignorar o cruzamento a meio do trajecto, e no final da linha "dar a a volta" repetindo sempre este ciclo. O primeiro *if* do nosso código avalia se o robô está a andar em cima da linha preta de forma correcta, enquanto que o segundo e terceiro fazem as devidas correcções para a direita e esquerda respectivamente. O *if* seguinte trata de ignorar o cruzamento enquanto que o ultimo faz com que o robô "dê a volta" quando deixar de encontrar a linha preta.

Por ultimo adicionamos um *if* que de certa forma era um pouco opcional que fazia o código parar assim que pressionarmos o botão preto.

### 0.4.4 Problema 4

Este programa foi o mais complexo que resolvemos. Para o solucionar dentro do ciclo *forever* o programa encontra um *if* que verifica se o robô se encontra em cima da linha ou não, caso se encontre vai haver outro *if* que vai avaliar o caso de este se encontrar num cruzamento e começa o processo de virar a direita através de dois *repeat until* o primeiro fazendo virar até a linha passar o sensor direito (*right*) e o segundo até a linha encontrar o sensor (*center*) central do robô. Como faria sentido o *else* de encontrar o cruzamento quando se encontra na linha é de andar em frente seguindo a linha até o cenário que o robô encontre mude e ele tenha de agir de forma diferente. Com esta mudança de cenário entra o *else* do primeiro *if* que neste caso é quando o robô não encontra linha preta no sensor central (*center*) e com isto vai tomar uma de e decisões, corrigir a posição à esquerda, à direita ou então "dar a volta". Como nos outros programas é acrescentado um ultimo *if* para parar o programa a quando o botão preto é pressionado.

## 0.5 Resultados

### 0.5.1 Problema 1

Através da implementação, criação e posterior upload do programa apresentado na Figura 7, conseguimos obdecer ao enunciado do Problema 1.

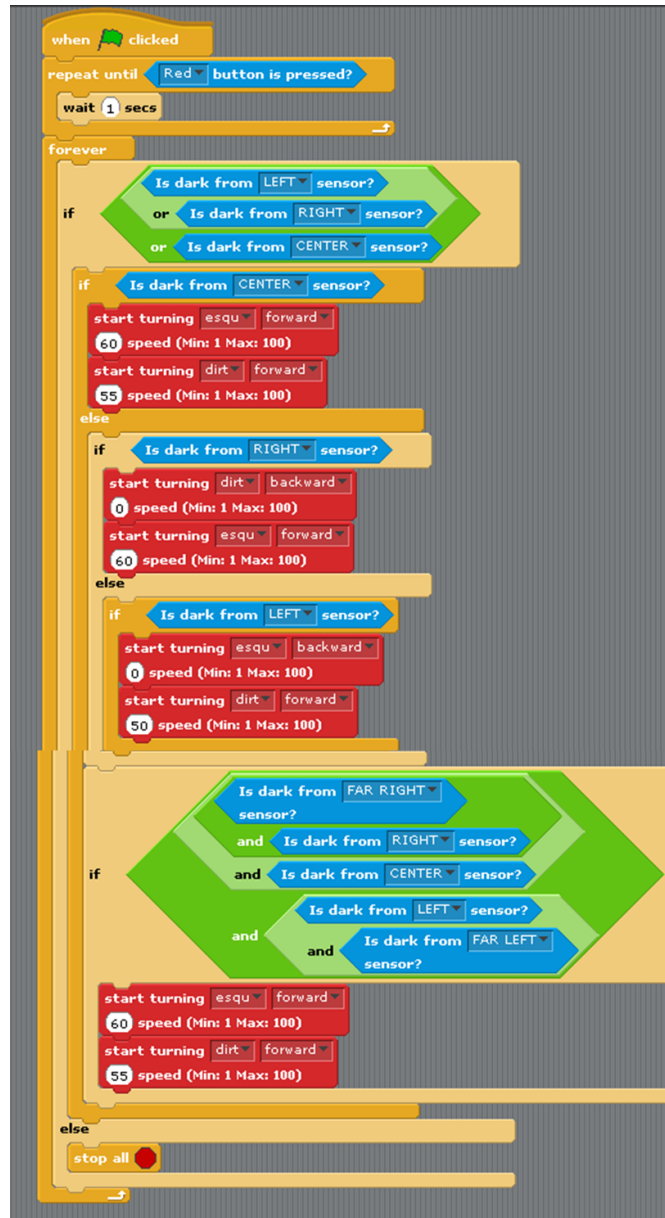


Figura 7: Resolução do problema 1 utilizando o software DETInchanting.

## 0.5.2 Problema 2

Através da implementação, criação e posterior upload do programa apresentado na Figura 8, conseguimos obdecer ao enunciado do Problema 2.

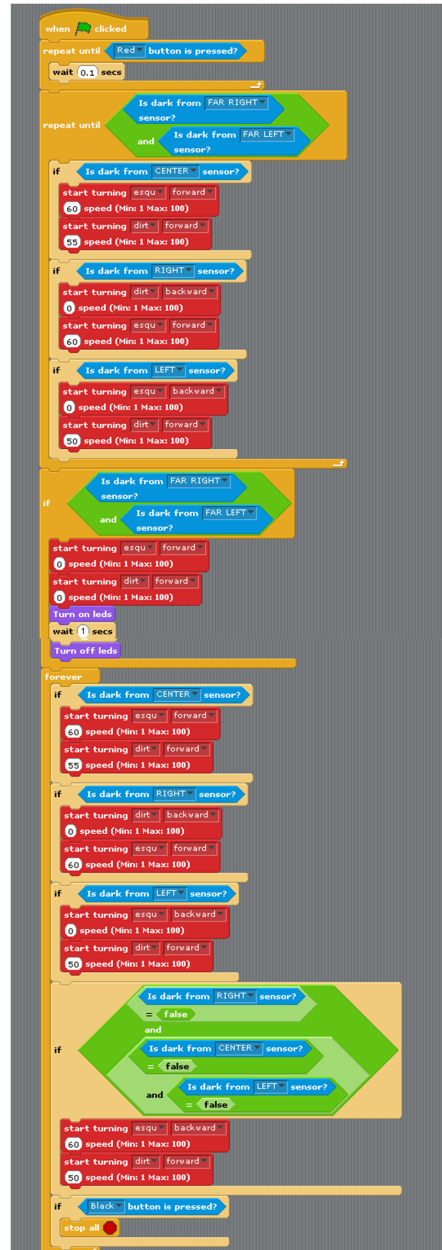


Figura 8: Resolução do problema 2 utilizando o software DETInchanting.

### 0.5.3 Problema 3

Através da implementação, criação e posterior upload do programa apresentado na Figura 9, conseguimos obdecer ao enunciado do Problema 3.

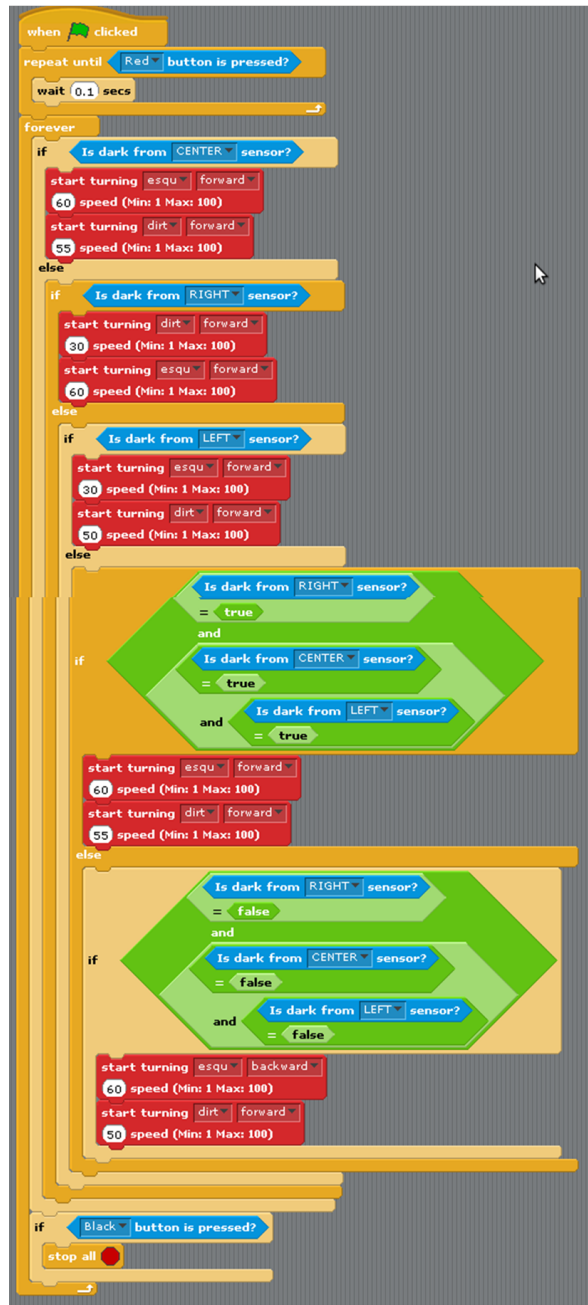


Figura 9: Resolução do problema 3 utilizando o software DETInchanting.

### 0.5.4 Problema 4

Através da implementação, criação e posterior upload do programa apresentado na Figura 10, conseguimos obdecer ao enunciado do Problema 4.

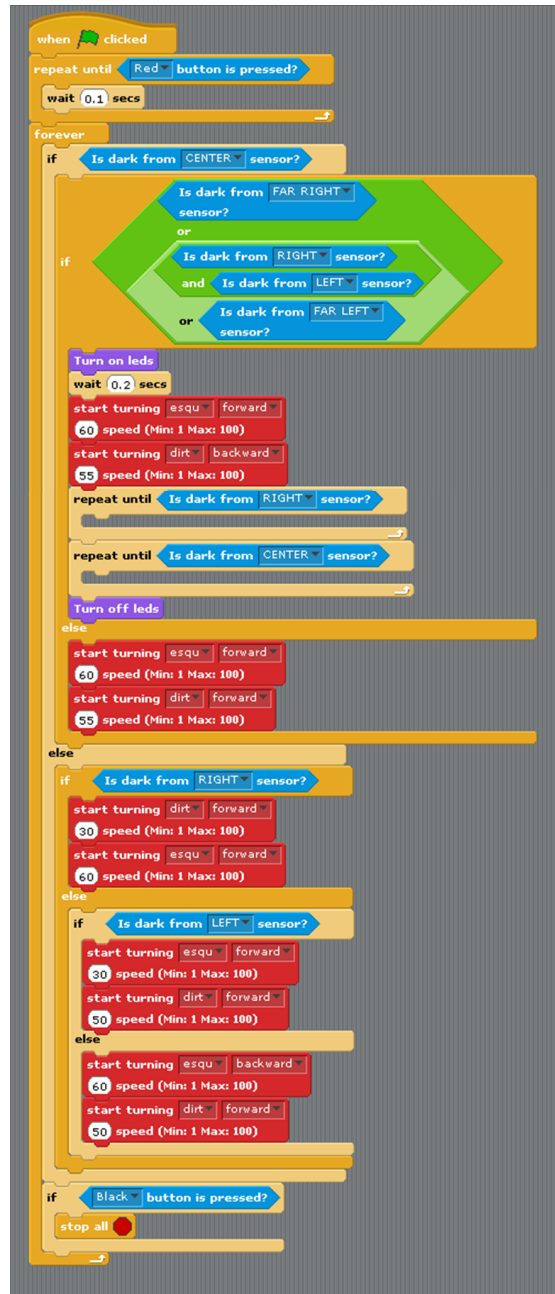


Figura 10: Resolução do problema 4 utilizando o software DETInchanting.



### **0.5.5 Problema 5**

Devido à falta de tempo, não nos foi possível realizar e programar o Problema 5. Na próxima secção iremos apresentar um esquema que permitirá compreender o funcionamento deste programa.

## 0.6 Análise dos Resultados

### 0.6.1 Problema 1

Através da execução do programa apresentado na secção acima, relativamente a este problema, não alcançámos grandes dificuldades no seu desenvolvimento. O nosso robô obedeceu ao esquema que se encontra representado na figura seguinte, daí pensamos que o programa estará a funcionar em perfeitas condições.

Como já dissemos anteriormente, não tivemos problemas na execução deste programa, embora inicialmente estávamos um pouco em dúvida em relação ao blocos que teríamos de utilizar.

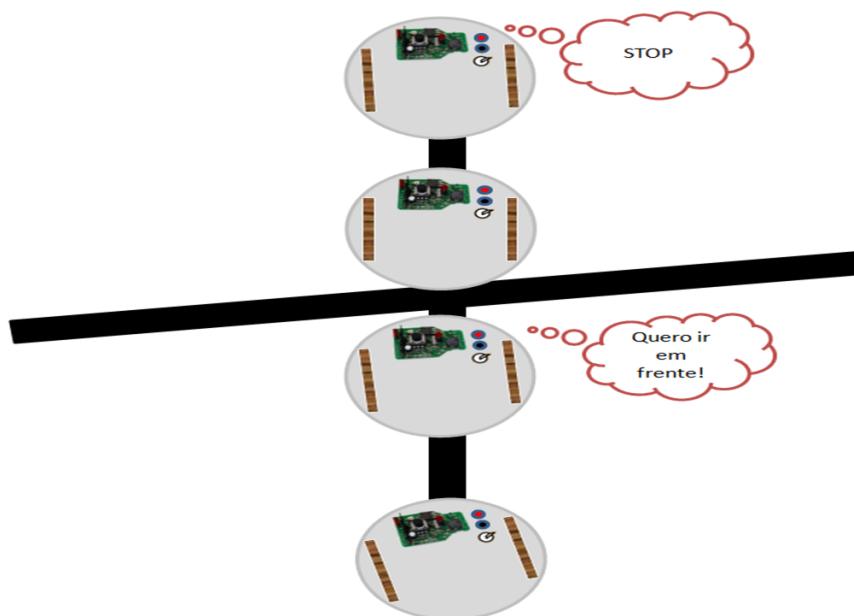


Figura 11: Esquema estroboscópico do Problema 1.

### 0.6.2 Problema 2

Como o problema 2 resume-se a uma adaptação do problema 1, não tivemos grande dificuldade na sua programação. Apenas tivemos que alterar o programa 1 de modo que o robô pare a primeira vez que encontrar o cruzamento e durante 1 segundo ligue os 4 led's, de seguida o robô terá de percorrer a linha e no final desta, voltar para trás. Nas restantes vezes que encontrar o cruzamento irá ignorar-lo e seguirá em frente, tal como está representado na figura abaixo.

Tal como no problema anterior não obtivemos grandes dificuldades na execução do algoritmo. Pensamos também que o programa está a funcionar em perfeitas condições.

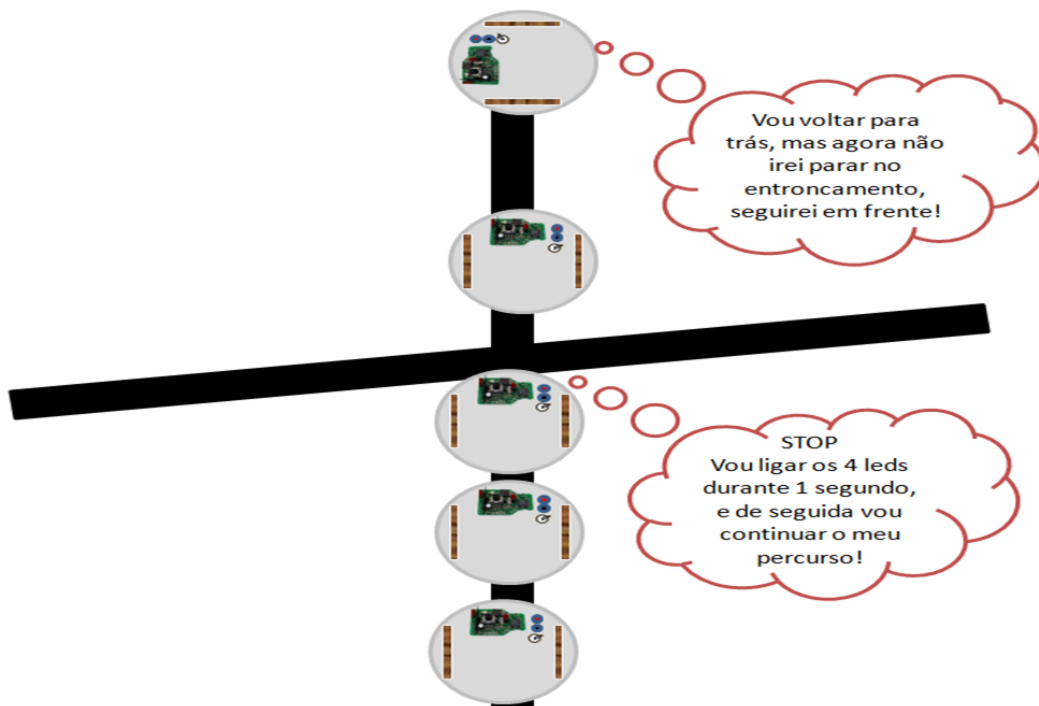


Figura 12: Esquema estroboscópico do Problema 2.

### 0.6.3 Problema 3

Na resolução deste problema não nos deparamos com grandes dificuldades, pois como todas as situações previstas para o robô tomar decisões já tinham sido tratadas indirectamente em outros exercícios quer de outras aulas quer na própria aula em exercícios anteriores, fez com que este exercício se torna-se de certo modo mais acessível em relação ao tempo que nos era dado. O robô conseguia sem grandes dificuldades atravessar o cruzamento sem qualquer reacção e sempre que necessário corrigia a sua posição assim como dava a volta quando chegava ao fim da linha preta.

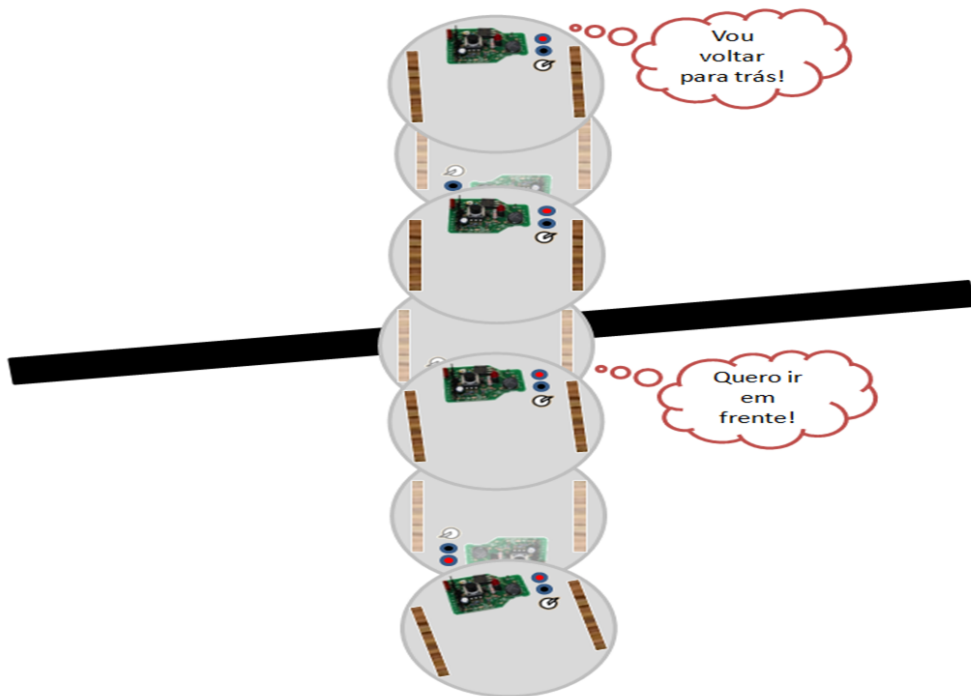


Figura 13: Esquema estroboscópico do Problema 3.

#### 0.6.4 Problema 4

A resolução deste problema foi sem dúvida a que nos causou mais problema a nível de execução, pois não estávamos a conseguir visualizar o algoritmo que fazia o robô virar no cruzamento, fazendo com isto o robô em muitas das nossas tentativas seguir em frente, tal como se encontra representado na figura abaixo. Depois de muitas tentativas conseguimos descobrir o algoritmo certo onde depois foi preciso apenas corrigir posições após a viragem à direita assim como a própria viragem uma vez que também nos deparamos com o problema de o robô após virar não conseguir encontrar a linha preta. Foi sem dúvida o programa que nos consomiu mais tempo e uma das principais razões pelo que não tivemos tempo para resolver o problema 5. Mas no fim consideramos este problema um sucesso uma vez que o robô conseguiu executar o programa na perfeição fazendo exactamente o que o enunciado pedia.

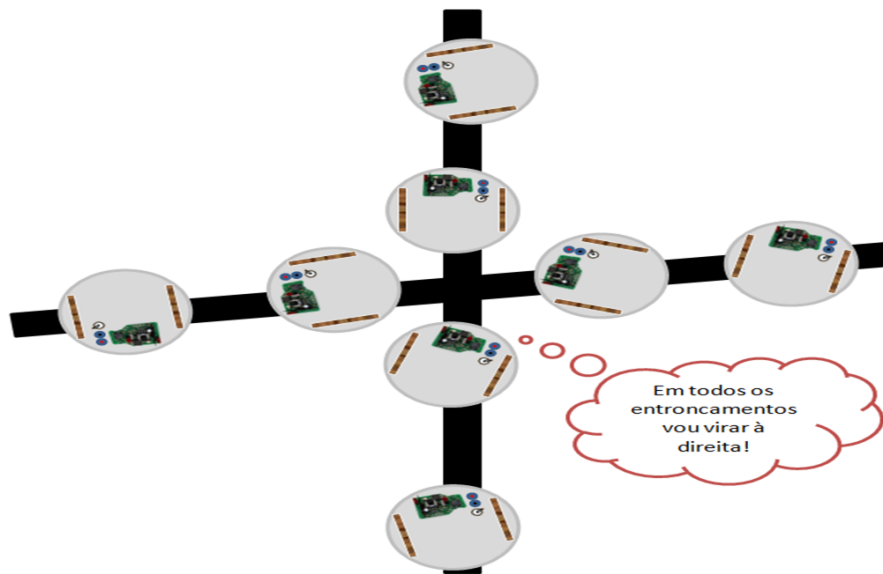


Figura 14: Esquema estroboscópico do Problema 4.

### 0.6.5 Problema 5

Apesar de não termos tido oportunidade de resolver o programa relativo ao problema 5, devido à falta de tempo como anteriormente frisámos, deixamos aqui o esquema estroboscópico do percurso do robô respeitando o enunciado do problema.

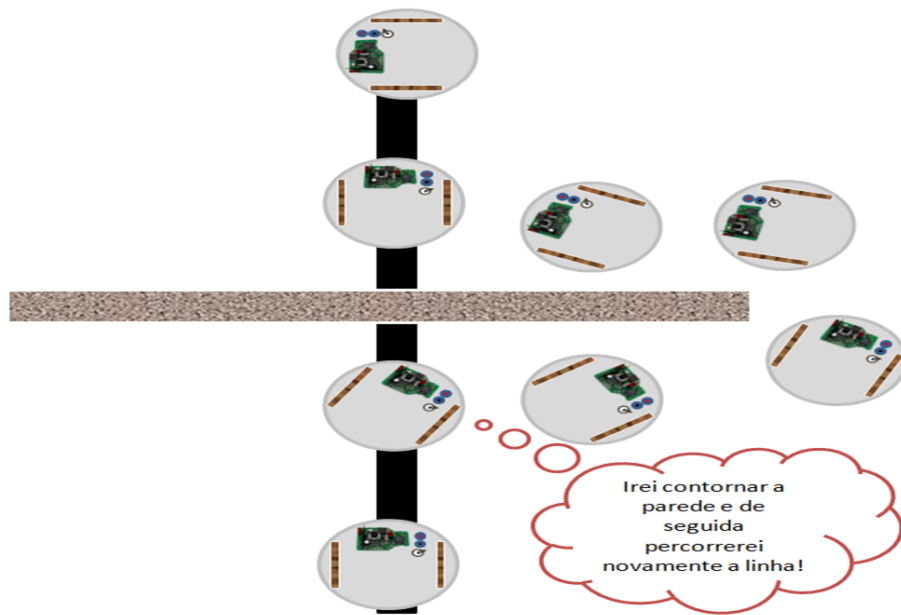


Figura 15: Esquema estroboscópico do Problema 5.

## 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 apenas um dos problemas apresentamos, devido ao fator tempo, pensamos que os restantes foram bem conseguidos, embora com algumas dificuldades. Em todos os programas tivemos algumas duvidas na escolha dos blocos a utilizar, embora no final da aula conseguimos executar quase a totalidade dos problemas enunciados com sucesso.

# Bibliografia

- [1] A.V.C.M. Zúquete , Site da Unidade Curricular - Introdução à Engenharia de Computadores e Telemática (12 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 (12 Dezembro 2012). Guião da aula prática 9 [Online] *Available:* <http://www.ieeta.pt/avz/Aulas/IECT/12-13/docs/guiao-9.pdf>