

ISCTE  **IUL**
Instituto Universitário de Lisboa

Departamento de Ciências e Tecnologias de Informação
Mestrado em Engenharia de Telecomunicações e Informática

poRsCHE – *Remote Control via Head movEs*

José Alberto Porfírio Pequeno Lourenço

Dissertação submetida como requisito parcial para obtenção do grau de
Mestre em Engenharia de Telecomunicações e Informática

Orientador:
Doutor Alexandre Passos de Almeida, Professor Auxiliar,
ISCTE - IUL

Lisboa – setembro – 2016 – ISCTE-IUL

Agradecimentos

Em primeiro lugar, agradecer à minha família, à minha Mãe, ao meu Pai, aos meus irmãos Ana e Rui, aos meus Avós e à minha tia Paula por todo o apoio e motivação que me deram para terminar esta fase da minha vida, e que sem eles não seria possível.

Um agradecimento especial aos Engenheiros Biomédicos Mariana Matos e Tiago Robalo e aos pacientes Mónica Silva, Artur Neves e Ricardo Neves do Centro de Medicina de Reabilitação de Alcoitão pela afável receção e por todo o apoio prestado para o aperfeiçoamento do sistema implementado no decorrer da dissertação.

Finalmente, agradeço ao Prof. Doutor Alexandre Passos de Almeida por aceitar orientar esta dissertação e por toda a ajuda e conselhos que me foi dando ao longo da mesma.

A todos, muito obrigado.

Resumo

Esta tese descreve um Sistema Embebido ergonómico, não intrusivo e de baixo consumo energético que permite que pessoas tetraplégicas comandem uma cadeira de rodas elétrica através dos seus movimentos de cabeça. Os movimentos de comando foram implementados tendo em consideração as limitações físicas dos utilizadores. Utilizando apenas um leve equipamento colocado na cabeça do utilizador e transmitindo dados através de um sistema sem fios implementado através do protocolo Bluetooth 4.0 (Bluetooth Low Energy) assegura-se um sistema ergonómico (inexistência de cabos) e de baixo consumo energético. Como funcionalidade extra, o sistema tem a capacidade de disparar um alarme de emergência após identificação de um desmaio do utilizador.

Palavras-chave: Tetraplegia; Controlo Remoto; Movimentos de cabeça; Cadeira de rodas; *Low Energy*; SensorTag; Sistema de sensores; Alarme de desmaio.

Abstract

We present an ergonomic, nonintrusive and low power Embedded System which allows tetraplegic people to control an electric wheelchair by using their head movements. These command movements are implemented by taking into account people's physical limitations. The main sensing device is installed in a low weight board attached to the user head via an elastic band. Wireless communication is implemented with Bluetooth 4.0 (Bluetooth Low Energy) which assures an ergonomic (non-existence of cables) and a low power battery operated system. A smartphone connection provides extra functionality which triggers an emergency contact call after identifying a user fainting condition.

Keywords: Tetraplegia; Remote Control; Head Moves; Wheelchair; Low Energy; SensorTag; Sensor System; Faint Alarm.

Índice

Capítulo 1. Introdução	15
1.1. Motivação e Enquadramento	15
1.2. Questões de Investigação e Objetivos	16
1.3. Método de Investigação.....	18
Capítulo 2. Revisão da Literatura	19
2.1. Hardware	20
2.2. Software	23
2.2.1. Interpretação dos comandos de controlo.....	23
2.2.2. Modos de funcionamento do sistema.....	28
2.2.3. Utilização da interface gráfica	29
2.2.4. Calibração do sistema	30
2.2.5. Personalização do sistema para vários utilizadores	31
2.2.6. Alarme médico	31
2.3. Testes aos protótipos	32
2.4. Dúvidas sobre as implementações estudadas	32
2.5. Limitações encontradas nos sistemas.....	33
2.6. Síntese	34
Capítulo 3. Definição do sistema poRsCHE	35
3.1. Seleção do protocolo <i>wireless</i>	36
3.2. Estrutura do sistema	41
3.3. Definição dos movimentos de comando	43
Capítulo 4. Protocolo <i>Wireless BLE</i>	45
4.1. Introdução	46
4.2. Arquitetura	48
4.2.1. <i>Application</i>	48
4.2.2. <i>Host</i>	50
4.2.3. <i>Controller</i>	50
4.3. BLE Device Address	52
4.4. Topologia da rede	53
4.4.1. <i>Broadcast</i>	53
4.4.2. <i>Connected</i>	54
4.5. BLE na arquitetura poRsCHE.....	55
Capítulo 5. Hardware	56
5.1. Seleção da PD com sensor de movimento.....	57
5.2. Seleção da PD com portas <i>input/output</i> programáveis.....	61
5.3. Implementação do protótipo	63
5.3.1. Componentes e estrutura	63
5.3.2. Alimentação energética.....	65
5.3.3. Ligações eletrónicas	66
5.4. Estudo dos sensores de movimento do SensorTag.....	69
5.4.1. Funcionamento do acelerómetro MEMS	70
5.4.2. Funcionamento do magnetómetro MEMS.....	71
5.4.3. Funcionamento do giroscópio MEMS	72
5.4.4. Estudo do acelerómetro MEMS	73
5.4.5. Estudo do magnetómetro MEMS.....	77
5.4.6. Estudo do giroscópio MEMS	79
5.4.7. Seleção do sensor de movimento	83
Capítulo 6. Software	84
6.1. Seleção da plataforma móvel.....	85

6.2. Ligações <i>wireless</i> BLE	86
6.3. Funcionamento do sistema poRsCHE	92
6.3.1. Pré-condução da cadeira de rodas	93
6.3.2. Condução da cadeira de rodas	97
Capítulo 7. Conclusão.....	103
7.1. Validação do Sistema	105
7.2. Ilacões	106
7.3. Trabalho futuro	107
Referências.....	108
Anexo.....	111

Índice de Figuras

Figura 1 – Comandos especiais atualmente comercializados para pessoas tetraplégicas [2]	19
Figura 2 – Funcionamento do SE	20
Figura 3 – Fluxograma do sistema [3] (dois movimentos de cabeça)	24
Figura 4 – Fluxograma do sistema [3] (quatro movimentos de cabeça)	25
Figura 5 – Representação gráfica dos comandos de controlo do sistema [8]	26
Figura 6 – Representação das sequências de comando do sistema [9]	27
Figura 7 – Interface gráfica do projeto [13]	29
Figura 8 – Interface gráfica do projeto [10]	30
Figura 9 – Movimento de flexão	44
Figura 10 – Movimento de extensão	43
Figura 11 – Movimento de inclinação lateral	43
Figura 12 – Os três protocolos Bluetooth [25]	46
Figura 13 – Arquitetura BLE [20]	48
Figura 14 – Hierarquia GATT [24]	49
Figura 15 – Largura de Banda utilizada pelo BLE [20]	51
Figura 16 – Topologia BLE <i>Broadcast</i> [20]	53
Figura 17 – Topologia BLE <i>Connected</i> [20]	54
Figura 18 – Arquitetura poRsCHE	55
Figura 19 – SensorTag CC2650, dimensões e benefícios [27]	57
Figura 20 – SensorTag CC2650, sensores e módulo rádio BLE integrados [27]	58
Figura 21 – SensorTag CC2650, sistema de alimentação energética à base de bateria [27]	58
Figura 22 – Fita desportiva para segurar o SensorTag à cabeça do utilizador	59
Figura 23 – Fita desportiva colocada sobre um boné	59
Figura 24 – RaspberryPi Model B+ [35]	61
Figura 25 – <i>Pen</i> ASUS BT-400 [36]	62
Figura 26 – Base para o protótipo poRsCHE	63
Figura 27 – Protótipo poRsCHE, visão lateral	64
Figura 28 – Protótipo poRsCHE, visão superior	64
Figura 29 – Bateria para alimentar servos e sonar	65
Figura 30 – Bateria para alimentar RaspberryPi	65
Figura 31 – Circuito eletrónico protótipo poRsCHE	66
Figura 32 – Sinal PWM-rotação do servo no sentido horário	66
Figura 33 – Sinal PWM-rotação do servo no sentido anti-horário	67
Figura 34 – Sinal PWM-imobilização do servo	67
Figura 35 – Divisor de tensão 5-3,3V	68
Figura 36 – Estrutura de um Acelerómetro MEMS [47] (imagem editada)	70
Figura 37 – Demonstração do efeito <i>Hall</i> para medição do campo magnético I [47] (imagem editada)	71
Figura 38 – Demonstração do efeito <i>Hall</i> para medição do campo magnético II [47] (imagem editada)	71
Figura 39 – Medição de rotação por parte de um Giroscópio MEMS [48] (imagem editada)	72
Figura 40 – Estrutura de um Giroscópio MEMS [47] (imagem editada)	72
Figura 41 – Colocação do SensorTag na cabeça do utilizador através de um boné	73
Figura 42 – Eixos referenciais do Acelerómetro do SensorTag	73
Figura 43 – Colocação do SensorTag na cabeça do utilizador através de um boné	79
Figura 44 – Eixos referenciais do Giroscópio do SensorTag	79
Figura 45 – 1ª ligação BLE estabelecida pelo sistema	86
Figura 46 – 3ª ligação BLE estabelecida pelo sistema	87
Figura 47 – 2ª ligação BLE estabelecida pelo sistema	89
Figura 48 – Comunicação BLE alarme de emergência	91
Figura 49 – Fluxograma do sistema poRsCHE	92
Figura 50 – Calibração dos movimentos de comando através da aplicação Android	93
Figura 51 – Aplicação Android - <i>upload</i> dos valores de calibração	94
Figura 52 – Aplicação Android – seleção do contacto de emergência	95
Figura 53 – Aplicação Android - comunicação à PD RaspberryPi dos valores de calibração	96
Figura 54 – Receção da SMS de alerta de emergência no <i>SmartPhone</i> do cuidador	102

Índice de Tabelas

Tabela 1 – <i>Hardware</i> utilizado nos projetos estudados	21
Tabela 2 – Protocolos <i>wireless</i> estudados [15-19].....	36
Tabela 3 – Consumos energéticos dos protocolos analisados nos artigos [15-16].....	36
Tabela 4 – Características dos módulos rádio testados no artigo [17].....	38
Tabela 5 – Consumos energéticos dos protocolos analisados nos artigos [17]	38
Tabela 6 – Consumos energéticos do protocolo BLE [18].....	39
Tabela 7 – Principais diferenças entre o Bluetooth Clássico e o BLE [21].....	47
Tabela 8 – Venda de <i>SmartPhones</i> (unidades e quotas de mercado) a nível Mundial, por Sistema Operativo utilizado, no quarto quadrimestre de 2014 e 2015 [49].....	85
Tabela 9 – Serviço BLE <i>Movement Service</i> [27].....	87
Tabela 10 – Serviço BLE <i>Commands Calibration</i>	89

Índice de Gráficos

Gráfico 1 – Consumos energéticos, em mW, dos protocolos analisados nos artigos [15]	37
Gráfico 2 – Consumo energético, em mJ/Mb, dos protocolos analisados nos artigos [15]	37
Gráfico 3 – Consumo energético, em mJ/Mb, dos protocolos analisados nos artigos [16]	37
Gráfico 4 – Consumo energético dos protocolos analisados nos artigos [17]	39
Gráfico 5 – Acelerómetro, Movimento de flexão	74
Gráfico 6 – Acelerómetro, Movimento de extensão	74
Gráfico 7 – Acelerómetro, Movimento de inclinação para o lado direito	75
Gráfico 8 – Acelerómetro, Movimento de inclinação para o lado esquerdo	75
Gráfico 9 – Magnetómetro, Movimento de flexão	77
Gráfico 10 – Magnetómetro, Movimento de extensão	77
Gráfico 11 – Magnetómetro, Movimento de inclinação para o lado direito	78
Gráfico 12 – Magnetómetro, Movimento de inclinação para o lado esquerdo	78
Gráfico 13 – Giroscópio, Movimento de flexão	80
Gráfico 14 – Giroscópio, Movimento de extensão	80
Gráfico 15 – Giroscópio, Movimento de inclinação para o lado direito	81
Gráfico 16 – Giroscópio, Movimento de inclinação para o lado esquerdo	81
Gráfico 17 – Valores de calibração (amplitudes máximas)	90
Gráfico 18 – Identificação do Movimento de comando Deslocação em frente	97
Gráfico 19 – Identificação do Movimento de comando Redução de velocidade / Stop	98
Gráfico 20 – Identificação do Movimento de comando Deslocação em marcha-atrás	99
Gráfico 21 – Identificação do Movimento de comando Rotação para a direita	99
Gráfico 22 – Identificação do Movimento de comando Rotação para a esquerda	100
Gráfico 23 – Identificação do Movimento de comando Alteração de Estado	101
Gráfico 24 – Identificação de um falso comando	101

Lista de Acrónimos

AVC	Acidente Vascular Cerebral
SE	Sistema Embebido
PD	Placa de Desenvolvimento
CMRA	Centro de Medicina de Reabilitação de Alcoitão
PC	<i>Personal Computer</i>
LCD	<i>Liquid Crystal Display</i>
LEDs	<i>Light Emitting Diodes</i>
FIPA	<i>Foundation for Intelligent Physical Agents</i>
UWB	<i>Ultra-Wide Band</i>
BLE	<i>Bluetooth Low Energy</i>
PWM	<i>Pulse Width Modulation</i>
SMS	<i>Short Message Service</i>
SIG	<i>Special Interest Group</i>
GATT	<i>Generic Attribute Profile</i>
ACK	<i>Acknowledgmente</i>
UUID	<i>Universal Unique Identifier</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>
SDIO	<i>Secure Digital Input/Output</i>
CRC	<i>Cyclic Redundancy Check</i>
ISM	<i>Industrial, Scientific and Medical</i>
FHSS	<i>Frequency-hopping spread spectrum</i>
GFSK	<i>Gaussian frequency-shift keying</i>
MAC	<i>Media Access Control</i>
IP	<i>Internet Protocol</i>
IEEE	Instituto dos Engenheiros Eletrotécnicos e Eletrónicos
MEMS	Sistema Mecânico Microeletrónico
PVC	<i>Polyvinyl Chloride, Vinyl</i>
GPIO	<i>General-Purpose Input/Output</i>

Capítulo 1. Introdução

1.1. Motivação e Enquadramento

Todos os dias ocorrem acidentes e são diagnosticadas doenças que limitam a mobilidade das pessoas. Além dos inúmeros acidentes que provocam paraplegias ou tetraplegias, existem também várias doenças que impossibilitam as pessoas de ter uma locomoção independente [1]. De seguida enumeram-se algumas:

- lúpus;
- esclerose múltipla;
- alterações da função vascular, que poderão originar Acidentes Vasculares Cerebrais (AVC);
- tumores;
- infeções;
- malformações;
- processos degenerativos ou compressivos.

O uso das tradicionais cadeiras de rodas elétricas melhora consideravelmente a qualidade de vida de quem ficou com a sua locomoção limitada ou negada. Contudo, não estão adaptadas para pessoas que além de paralisia dos membros inferiores, tenham também paralisia dos membros superiores.

As tradicionais cadeiras de rodas elétricas são propulsionadas por um motor elétrico comandado pelos movimentos realizados num *joystick*. Esta implementação é eficaz para pessoas com limitações dos membros inferiores, contudo é ineficiente para pessoas com limitações conjuntas dos membros inferiores e superiores, uma vez que estes não conseguem controlar o *joystick*.

Com o intuito de ultrapassar a dificuldade acima exposta, propõe-se a utilização de um microcontrolador associado a um acelerómetro para efetuar as funções do *joystick*, ou seja, através dos movimentos efetuados pela cabeça do utilizador, enviar ordens de comando para a condução da cadeira de rodas ao motor da mesma.

Já foram realizados alguns estudos com o intuito de ultrapassar este obstáculo. No entanto, devido ao facto de serem demasiado complexos, intrusivos e/ou de custos avultados, acabaram por não ser colocados ao dispor da sociedade.

1.2. Questões de Investigação e Objetivos

A presente dissertação está focada no desenvolvimento de um Sistema Embebido (SE) que permita que a condução de uma cadeira de rodas elétrica seja feita a partir dos movimentos da cabeça do utilizador. O objetivo é superar as implementações já desenvolvidas de modo a que o produto final seja colocado à disposição daqueles que dele necessitem.

Através do estudo do Estado da Arte, apresentado mais à frente no capítulo 2., e dos planos iniciais para a dissertação, identificam-se os seguintes objetivos de implementação para o protótipo final, que se explicam posteriormente:

- ergonomia e não intrusão;
- baixo consumo energético;
- controlo eficiente;
- deteção de obstáculos;
- alerta de emergência.

Com o objetivo de alcançar um produto final que consiga ser utilizado pelo público alvo, considera-se que os dispositivos do SE terão que ser o mais ergonómicos possível, para poderem ser utilizados durante longos períodos de tempo, e não intrusivos, por forma a minorar qualquer tipo de desconforto. Com esta declaração de objetivos, aborda-se a implementação do equipamento principal do sistema. Equipamento que tem como função a obtenção de comandos de controlo por parte dos movimentos efetuados pela cabeça do utilizador e que, estando em contacto direto com este, levanta mais cuidado na sua implementação física. Tendo como plano inicial para este equipamento, a utilização de uma placa de desenvolvimento (PD)¹ com um acelerómetro integrado, colocam-se as seguintes questões: Qual a placa a utilizar? Qual a fonte de energia a utilizar para alimentar a placa? Qual o equipamento/vestuário a utilizar para segurar a placa à cabeça do utilizador?

Tendo como preocupação o consumo energético do sistema, de modo a promover a maior durabilidade do funcionamento do mesmo, colocam-se as seguintes questões: Qual ou quais as fontes de energia a utilizar para alimentar o SE? Qual o protocolo de comunicação *wireless* a implementar na comunicação entre os equipamentos do sistema? Quais as placas de desenvolvimento a utilizar?

Para alcançar a eficiência idealizada para o SE é crucial uma implementação dos comandos de controlo para a condução da cadeira de rodas que correspondam o melhor possível às necessidades do utilizador. Para satisfazer tal objetivo é necessário obter resposta às seguintes questões: Quais os movimentos mais adequados para servirem de comandos de controlo? Quais os movimentos mais ergonómicos?

¹ Vulgarmente designadas por microcontroladores, as placas de desenvolvimento (ex: Arduino e RaspberryPi) são placas únicas de *hardware* livre desenvolvidas com o objetivo de criar ferramentas de baixo custo, flexíveis e de fácil utilização.

Tendo como objetivo adicional a implementação de um alerta de emergência, colocam-se algumas questões para investigação: Quais os movimentos a considerar relevantes para despoletar o alarme? Qual o tipo de comunicação a acionar?

Para se poder responder às questões acima expostas irá ser realizado um estudo:

- de outros trabalhos já realizados;
- dos dispositivos disponíveis no mercado para implementação do protótipo.

Por outro lado, irão ser contactados profissionais do Centro de Medicina de Reabilitação de Alcoitão (CMRA) para compreender as limitações do público alvo de modo a ajustar a implementação do sistema a um maior número de indivíduos.

1.3. Método de Investigação

O método de investigação usado no âmbito desta dissertação é composto por quatro fases distintas, a saber:

- investigação das questões identificadas na secção anterior - no final desta fase define-se a implementação do *hardware* e do *software* (Capítulos 2. e 3.);
- implementação do *hardware* – implementação da estrutura que irá formar o SE (construção do protótipo que irá ser utilizado para testar o sistema (Capítulo 5.));
- implementação do *software* – desenvolvimento do *software* que faz funcionar o SE (Capítulo 6.);
- validação do protótipo – nesta fase são executados testes ao funcionamento do protótipo, para correções à implementação do *software* e/ou do *hardware* e para final validação deste (secção 1. do Capítulo 7.).

Capítulo 2. Revisão da Literatura

No mercado atual de cadeiras de rodas elétricas, os equipamentos construídos para que as pessoas tetraplégicas possam comandar a sua cadeira de rodas não são ergonômicos (obrigam a que a pessoa mantenha a cabeça inclinada para que a cadeira se desloque) e são intrusivos (apresentam um grande impacto visual). Como apresentado na figura 1, podem ser grandes estruturas de *switches*² instalados nas laterais e na retaguarda da cabeça do utilizador e/ou grandes estruturas com um *joystick* a ser controlado pelo utilizador através da sua boca ou queixo (comando mentoniano) [2].



Figura 1 – Comandos especiais atualmente comercializados para pessoas tetraplégicas [2]

Com intuito de ultrapassar as limitações identificadas, algumas investigações científicas já foram realizadas. Neste capítulo analisa-se uma amostra dessas investigações com o objetivo de, identificando os prós e contras de cada uma delas, definir a melhor abordagem para o presente projeto.

A análise é realizada segundo os seguintes tópicos: *hardware*, *software* e testes aos protótipos. Terminada a análise aos artigos, são expostas dúvidas decorrentes da análise feita às implementações dos sistemas estudados, limitações encontradas nos mesmos e uma síntese sobre os principais conhecimentos a reter para a implementação do sistema poRsCHE.

² Componentes que ativam/desativam/alternam funcionalidades do sistema através de pressão efetuada sobre estes.

2.1. Hardware

Através da figura 2 representa-se, numa perspetiva alto nível, o funcionamento dos sistemas estudados.

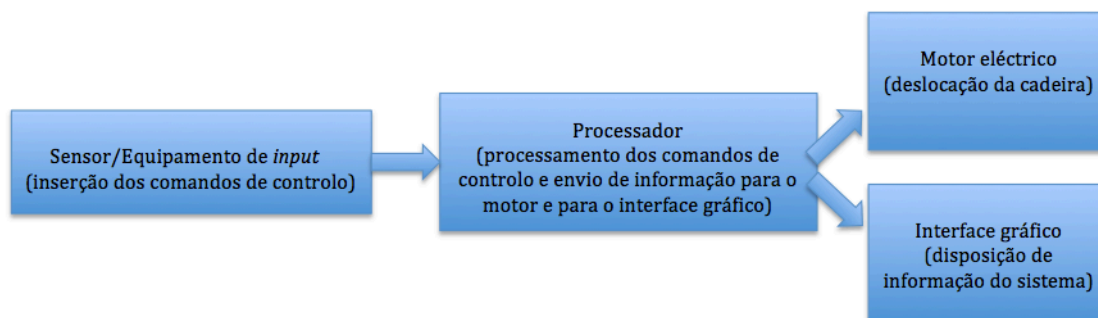


Figura 2 – Funcionamento do SE

Comparando as implementações dos projetos, com base na estrutura apresentada na figura 2, identificam-se algumas diferenças com respeito ao:

- tipo de sensor/equipamento de *input*;
- processamento do sistema;
- interface gráfica.

Embora quase todos os projetos estudados tenham implementado o controlo da cadeira de rodas por parte de movimentos da cabeça do utilizador, nem todos o fazem de forma exclusiva. Parte dos projetos adicionam outros meios de comando para o controlo da cadeira de rodas e um deles apresenta como comando de controlo, o movimento dos olhos do utilizador. Através da análise deste leque de implementações, avalia-se o tipo de controlo projetado para a presente dissertação e apreendem-se conhecimentos de possíveis *inputs* que poderão ser adicionados caso se encontre alguma adversidade à implementação que assim o justifique.

Como segunda diferença identificada, apresenta-se o processamento do sistema. Os projetos estudados implementaram o processamento do sistema através de um *Personal Computer* (PC) ou através de um microcontrolador. Nesta diferença encontrada, não existindo informação de que uma das implementações seja superior à outra pela sua capacidade de processamento, identifica-se como vantagem de implementação, o processamento através de um microcontrolador uma vez que este necessita de menores recursos energéticos, apresenta menores dimensões e tem um valor comercial consideravelmente mais baixo.

Por último, nas diferenças gerais entre projetos, identifica-se a existência/inexistência de uma interface gráfica que informa o utilizador do corrente funcionamento do sistema.

Na tabela 1 apresenta-se o *Hardware* utilizado por cada projeto.

Projeto <i>Hardware</i>	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]
Interface Multimodal (<i>joystick</i> , microfone, acelerómetro, teclado, rato e <i>webcam</i>)										X	X	X
Giroscópio do Emotiv EPOC Headset (movimentos da cabeça do utilizador)	X											
Acelerómetro (movimentos da cabeça do utilizador)		X	X		X		X	X				
Sistema de sensores infravermelhos (movimentos da cabeça do utilizador)				X								
Sensor de inclinação (movimentos da cabeça do utilizador)						X						
<i>Eye-ball sensor</i> (movimentos dos olhos do utilizador)									X			
PC	X			X						X	X	X
Microcontrolador		X	X		X	X	X	X	X			
Interface gráfica	X	X	X		X	X	X	X		X	X	X

Tabela 1 – *Hardware* utilizado nos projetos estudados

Além das diferentes implementações de *Hardware* apresentadas na tabela 1, é importante referir algumas características que são identificadas como relevantes para o estudo da presente dissertação, a saber:

- ao contrário dos restantes, os projetos [3-5] e [7], apresentam comunicações *wireless* entre o equipamento com o sensor de *input* colocado na cabeça do utilizador e o processador do sistema. Esta implementação torna os sistemas mais ergonómicos, proporcionando ao utilizador uma utilização do equipamento de comando de controlo (movimentos da cabeça) não limitada pelos cabos, que nos outros projetos, ligam este equipamento ao processador;
- as interfaces gráficas dos projetos têm maior ou menor impacto em termos de relevância para o funcionamento do sistema³ e são implementadas sobre os seguintes componentes de *Hardware*: PC

³ Na secção 2.2. será feita uma maior abordagem a este tema.

[3-4], [8], [12-14], *Liquid Crystal Display* (LCD) [5], [7], [10] e sinalizador com base em *Light Emitting Diodes* (LEDs) [9]. Destes três tipos de implementação, identifica-se como mais vantajosa, a que utiliza o LCD como interface gráfica. Através da utilização do LCD consegue-se uma boa relação entre dimensões positivas para a apresentação de informação e dimensões aceitáveis para um equipamento a ser adicionado à estrutura da cadeira de rodas;

- a colocação do acelerómetro na cabeça do utilizador é implementada com auxílio à fixação deste num boné [4], [9-10]. Esta implementação faz parte dos planos iniciais para a dissertação e é discutida mais à frente na secção 5.1.;
- cinco projetos adicionam ao seu funcionamento de controlo da condução da cadeira de rodas, sistemas de deteção de obstáculos a partir de tecnologias como sonares e infravermelhos [5], [11-14]. Como referido no capítulo anterior, o sistema poRsCHE irá também contar com uma funcionalidade do género por modo a aumentar a segurança do utilizador.

Como última referência às implementações de *Hardware* estudadas é de referenciar o controlo díspar da cadeira de rodas por parte do protótipo do projeto [9]. Ao contrário dos outros projetos, em que os comandos de controlo são enviados para os motores, neste projeto, os comandos de controlo são enviados para um equipamento mecânico que altera a posição do *joystick* por forma a controlar a condução da cadeira. Não existindo uma avaliação na primeira pessoa da execução do sistema, não existe uma opinião formada face à eficácia ou não deste. Contudo identifica-se como uma melhor implementação, o envio dos comandos de controlo diretamente aos motores das rodas da cadeira em detrimento do envio por parte do *joystick*.

2.2. *Software*

Nesta secção apresentam-se as principais implementações de *Software* dos projetos estudados. A apresentação das implementações é feita abordando os seguintes tópicos: interpretação dos comandos de controlo, modos de funcionamento do sistema, utilização da interface gráfica, calibração do sistema, personalização do sistema para vários utilizadores e sistema de alarme médico.

2.2.1. Interpretação dos comandos de controlo

Analisando os diferentes projetos identificam-se três tipos de controlo da cadeira de rodas: controlo através de análise multimodal, controlo através da análise dos movimentos dos olhos do utilizador e controlo através da análise dos movimentos da cabeça do utilizador.

Os projetos que implementaram análise multimodal não referem como foram projetados os comandos de controlo, pelo que não se sabe quais serão as instruções necessárias à condução da cadeira. Estes referem apenas que a tomada de decisão de controlo é feita com base na sequência de comandos inseridos e que esta sequência é analisada segundo fluxogramas inerentes ao sistema. Não existindo muitos dados para formar uma opinião concreta sobre a eficácia deste tipo de sistemas, apenas se refere que este aparenta ter um funcionamento de elevada complexidade, ao nível do número de diferentes *inputs* [12-14].

O projeto [11] distingue-se dos restantes por ter como equipamento de controlo um sensor de movimentos dos olhos. No artigo analisado, os autores, para explicarem o funcionamento do sensor, referem que a parte branca dos olhos pode ser localizada por infravermelhos e que através desta localização se podem identificar os movimentos efetuados pelos olhos de uma pessoa (cima, baixo, esquerda e direita). Não obstante a tecnologia utilizada, sensor de infravermelhos, ter sido explicada com sucesso, o artigo em questão não refere quais os movimentos dos olhos que foram associados aos movimentos de condução da cadeira. Pelo que, como nos projetos anteriormente referidos, também neste não se sabe quais são as instruções necessárias à condução da cadeira de rodas, ficando igualmente por saber a complexidade ou não de utilização deste tipo de equipamento para controlar a cadeira.

O projeto [3], projeto com implementação de controlo através de movimentos da cabeça, apresenta dois modos de funcionamento: controlo baseado em dois movimentos de cabeça e controlo baseado em quatro movimentos de cabeça. Nas figuras 3 e 4 ilustra-se o funcionamento dos dois modos respetivamente.

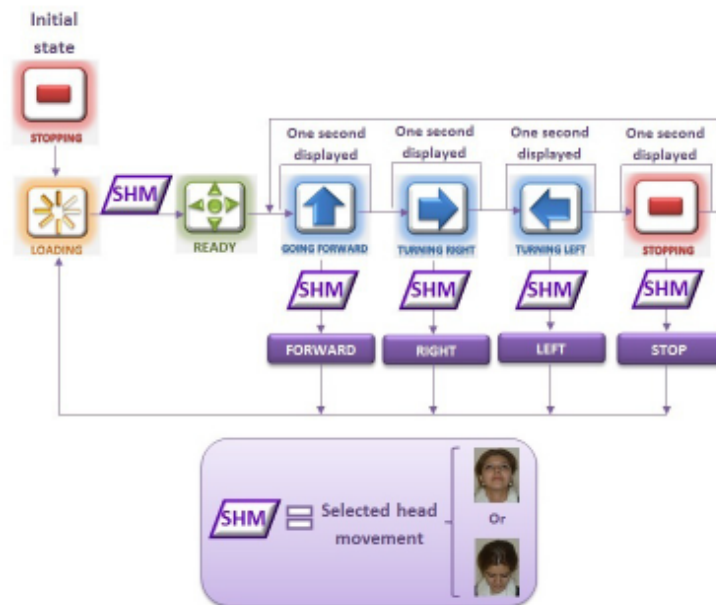


Figura 3 – Fluxograma do sistema [3] (dois movimentos de cabeça)

Como ilustrado na figura 3, o funcionamento do primeiro modo, é executado segundo uma seleção de comandos através de uma interação entre o utilizador e a interface gráfica do sistema. Desta interação ocorrem duas ações: a apresentação dos comandos pela interface gráfica e a seleção dos mesmos por parte do utilizador através de dois movimentos de cabeça (movimento ascendente para confirmar a seleção e movimento descendente para negar a seleção do comando apresentado) [3].

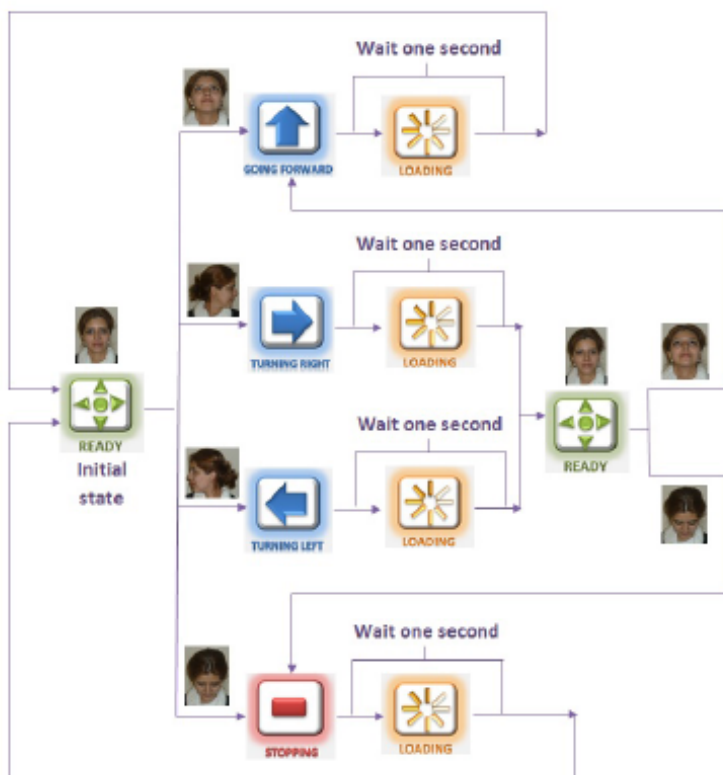


Figura 4 – Fluxograma do sistema [3] (quatro movimentos de cabeça)

Como ilustrado na figura 4, o funcionamento do segundo modo já não depende de uma interação entre o utilizador e a interface gráfica, depende apenas dos movimentos da cabeça do utilizador: movimento ascendente, para dar ordem de avanço à condução da cadeira de rodas; movimento descendente, para dar ordem de paragem à condução da cadeira de rodas; movimentos laterais, para dar ordens de deslocação lateral da cadeira de rodas [3].

Uma dúvida que surge imediatamente da inspeção à figura é como, nesta implementação, é distinguido o olhar ocasional para a direita ou para a esquerda, do respetivo comando para a mudança de direção da cadeira.

O projeto [6], projeto com implementação de controlo através de movimentos da cabeça detetados por um equipamento de sensores de infravermelhos, apresenta uma abordagem diferente aos comandos de controlo já apresentados. Segundo o artigo estudado, este projeto tem implementado:

- para o avanço e recuo da cadeira - movimentos de cabeça para a frente e para trás;
- para a deslocação lateral da cadeira - inclinação, para o lado pretendido, da cabeça do utilizador como comando de controlo.

Segundo os autores do projeto, a implementação da velocidade da cadeira foi feita com base na velocidade dos movimentos de cabeça que comandam o avanço ou o recuo da cadeira. Para o utilizador parar a deslocação da cadeira, os autores referem que os comandos de controlo implementados foram os mesmos que para a deslocação, mas com movimentos de cabeça em sentido oposto ao da condução e mais acelerados.

Realizando uma análise crítica à implementação anteriormente exposta, identifica-se como possíveis limitações do projeto, o controlo eficiente da velocidade de condução da cadeira de rodas, o controlo de imobilização da mesma e a implementação de comandos de controlo com recurso a movimentos de cabeça acelerados que são facilmente identificados como prejudiciais à saúde do utilizador.

Os restantes projetos com implementação de controlo através de movimentos da cabeça, tendo como equipamento de controlo um acelerómetro ou um sensor de inclinação, baseiam os seus comandos de controlo nas possíveis posições do tradicional *joystick* das cadeiras de rodas elétricas [4-5], [7-10]. Através da figura 5 ilustram-se os respetivos comandos.

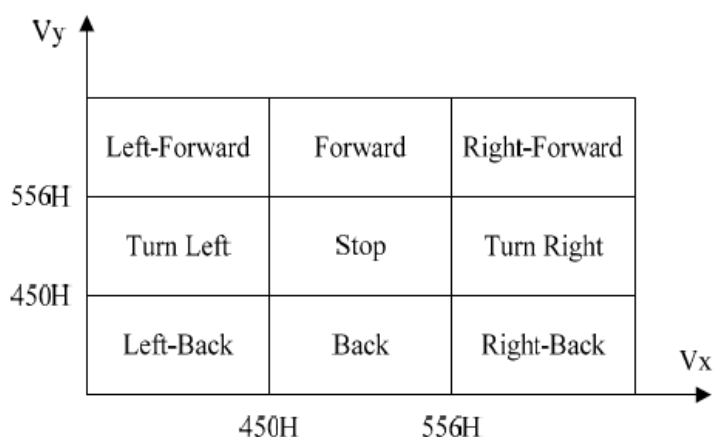


Figura 5 – Representação gráfica dos comandos de controlo do sistema [8]

A figura 5 apresenta os comandos implementados baseados no funcionamento do tradicional *joystick* das cadeiras de rodas elétricas. A apresentação gráfica destes comandos, é feita tendo como referenciais os valores de leitura de tensão por parte do acelerómetro convertidos para Hexadecimal [8].

Estando incluído nos projetos que apresentam um controlo da cadeira de rodas com base nas explicações dadas nos dois últimos parágrafos, o projeto [9], apresenta uma explicação, sobre a implementação, um pouco mais detalhada. Este projeto além de ter em conta, na sua implementação, os movimentos de cabeça efetuados pelo utilizador, tem também em conta possíveis sequências de comando e dois níveis de velocidade para a condução da cadeira.

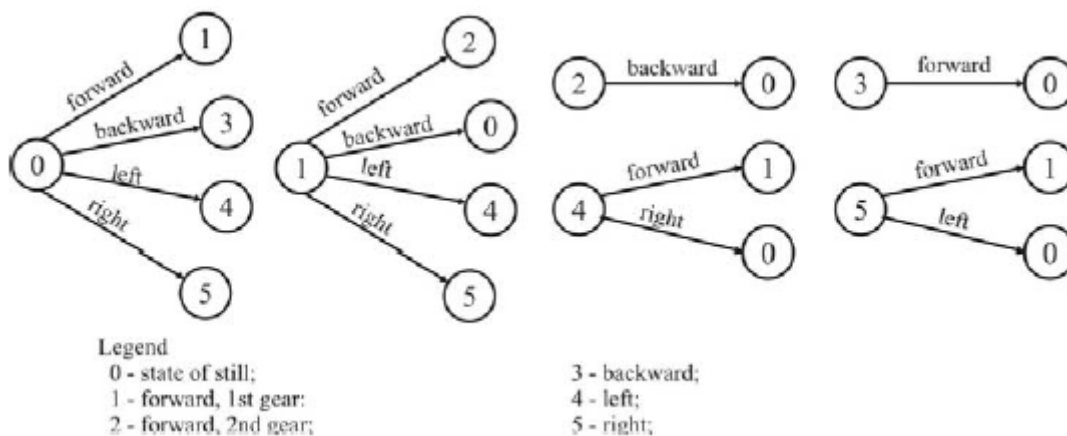


Figura 6 – Representação das sequências de comando do sistema [9]

Através da figura 6 pode-se compreender como se comporta o protótipo do projeto [9] segundo as sequências de comando apresentadas.

Ao iniciar o sistema, o utilizador encontra-se no estado 0 (posição de cabeça "olhar em frente"). Uma vez nesta posição, o utilizador tem quatro hipóteses de comando de controlo: inclinar a cabeça para a frente (locomoção avante da cadeira - estado 1), inclinar a cabeça para trás (locomoção à retaguarda da cadeira - estado 3), inclinação para a esquerda (virar a cadeira para a esquerda - estado 4) e inclinação para a direita (virar a cadeira para a direita - estado 5). Caso o utilizador opte por comandar a cadeira para avançar, irá ter em seguida mais quatro hipóteses de comando de controlo: inclinar a cabeça para a frente (locomoção avante da cadeira para uma segunda velocidade - estado 2), inclinar a cabeça para trás (imobilização da cadeira - estado 0), inclinação para a esquerda (virar a cadeira para a esquerda - estado 4) e inclinação para a direita (virar a cadeira para a direita - estado 5). Estando a cadeira na segunda velocidade de condução avante e, sendo efetuado pelo utilizador, o movimento de inclinação da cabeça para trás, ou estando a cadeira numa condução à retaguarda e, sendo efetuado pelo utilizador, o movimento de inclinação da cabeça para a frente, a cadeira passa ao estado 0 (imobilização da cadeira). Estando a cadeira a ser virada para um dos lados e, sendo efetuado pelo utilizador, o movimento de inclinação da cabeça para o lado oposto, a cadeira passa também ao estado 0 (imobilização da cadeira) [9].

Os autores deste projeto referem também no artigo que os comandos implementados apenas são válidos durante um período de tempo estabelecido, pelo que se o comando se mantiver o mesmo depois desse intervalo de tempo, deixa de ser válido passando o sistema ao comando neutro, imobilizando a cadeira (estado 0) [9].

2.2.2. Modos de funcionamento do sistema

Além da já referida implementação de dois modos de funcionamento no projeto [3], também outros dois projetos incluem na sua implementação mais do que um modo de funcionamento.

O projeto [13], apresenta uma arquitetura desenhada segundo o standard da Fundação para Agentes Físicos Inteligentes (FIPA - *Foundation for Intelligent Physical Agents*) de modo a promover um Sistema Inteligente que oferece três modos de funcionamento: navegação autónoma (condução da cadeira de rodas única e exclusivamente por parte do sistema implementado), interação homem-máquina estendida (condução da cadeira segundo uma conjugação de decisões entre o sistema e o utilizador) e condução semiautónoma com aviso de colisão e comportamento cooperativo e colaborativo (condução da cadeira por parte do utilizador através dos equipamentos de controlo disponibilizados mas com "supervisão" do sistema).

O projeto [14], apresenta uma implementação de *software* que oferece três modos de funcionamento do sistema: real (condução da cadeira de rodas no mundo real), simulado (condução da cadeira de rodas num simulador disponibilizado no PC do sistema) e realidade aumentada (condução da cadeira de rodas no mundo real com adição de obstáculo virtuais no simulador do sistema, para um teste ao sistema de dificuldade mais elevada e um treino mais exigente para o utilizador).

2.2.3. Utilização da interface gráfica

Nesta secção, de análise à implementação da interface gráfica do sistema, apresentam-se duas abordagens distintas. Uma que apresenta muita informação sobre comandos inseridos, leituras feitas pelo sistema, decisões tomadas e opções de comando por parte do utilizador [12-14]. E outra apenas com apresentação de *feedback* do funcionamento do sistema para o utilizador ter a perceção de algum erro que possa ter ocorrido no sistema [3], [5], [7], [9].

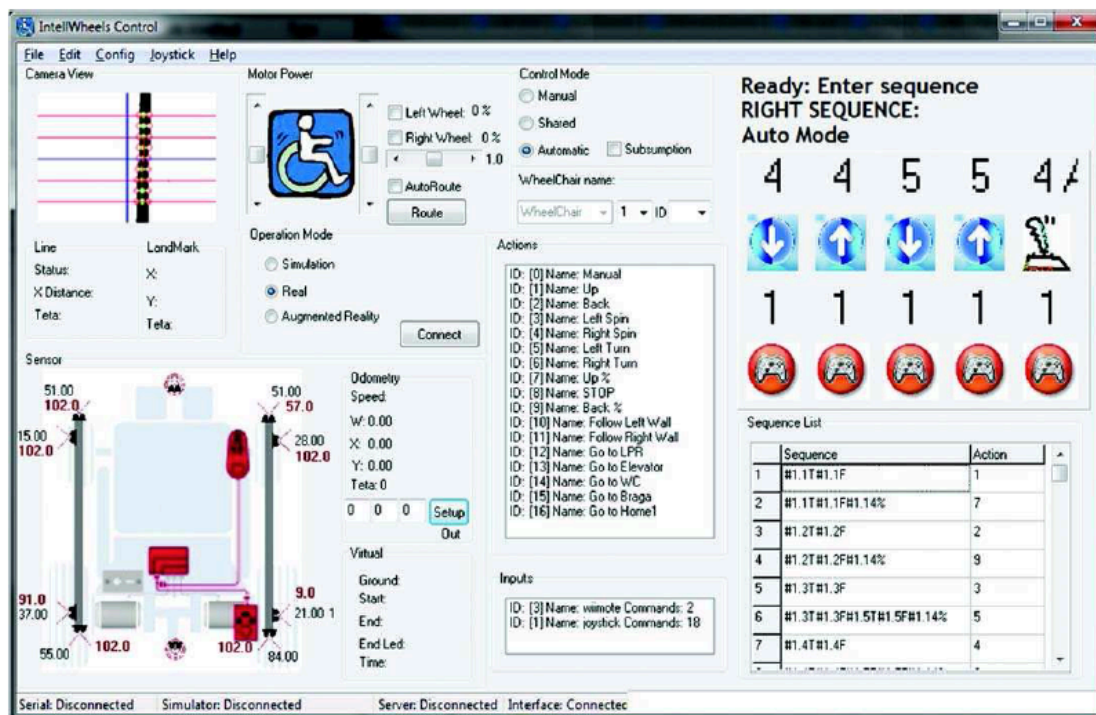


Figura 7 – Interface gráfica do projeto [13]

Através da figura 7 apresenta-se um exemplo da primeira abordagem, ilustrando a enorme quantidade de informação que a interface gráfica do projeto [13] apresenta. Este tipo de implementação não é exclusiva deste projeto. Também os projetos [12] e [14] apresentam uma implementação idêntica à ilustrada. Não obstante este tipo de interface gráfica apresentar informação útil para os engenheiros que conceberam o sistema, por forma a terem uma ferramenta de apoio ao *debug*⁴, esta apresenta informação considerada excessiva para a utilização do sistema por parte de um elemento do público alvo, podendo até ser causa de descontrolo da condução da cadeira.

⁴ Processo de encontrar e corrigir defeitos num *software*.



Figura 8 – Interface gráfica do projeto [10]

Na sequência da figura 3, apresenta-se na figura 8 outro exemplo de uma interface gráfica simples e que de igual modo apresenta interação com o utilizador. Esta última identifica-se como sendo a interface gráfica do projeto [10] e apresenta uma interação com o utilizador que será desenvolvida na secção seguinte.

Dos artigos estudados, não se obtém informação sobre o local onde as interfaces gráficas são implementadas.

Não obstante a informação apresentada na interface gráfica poder ser relevante para o utilizador ter um *feedback* do corrente funcionamento do sistema, não se sabe qual a influência que esta tem no controlo da condução da cadeira de rodas. Será que o utilizador consegue uma melhor condução da cadeira tendo acesso à informação apresentada no ecrã ou será que o movimento da cabeça, ao olhar para o ecrã, se reproduz num comando errado para a condução da cadeira?

2.2.4. Calibração do sistema

No seguimento do antepenúltimo parágrafo da secção anterior, faz-se referência a uma característica considerada importante e relevante para o bom funcionamento de um sistema como este.

A calibração do sistema é função inicial e inerente a quatro projetos estudados e define-se sucintamente como sendo uma conjugação de leitura e armazenamento de valores de amplitude dos movimentos da cabeça do utilizador. Esta função é não só essencial para uma boa condução da cadeira de rodas (leitura correta dos comandos de controlo), como para uma utilização do sistema por parte de mais do que um utilizador [3-4], [9-10].

Não sendo claro nos artigos estudados, subentende-se que a calibração seja feita no início da utilização do sistema e que os dados desta sejam guardados para futuras utilizações. Relativamente aos projetos que também têm incluído nas suas implementações sistemas de perfis de utilizador, tema abordado na próxima secção, estes dados são guardados nos dados de perfil do utilizador, permitindo que o sistema fique calibrado consoante o utilizador que o utilize.

2.2.5. Personalização do sistema para vários utilizadores

Existindo a necessidade de o sistema guardar dados pessoais do utilizador, como exige a funcionalidade referida na secção anterior, dois dos projetos estudados referem a implementação de perfis de utilizador.

O projeto [13] refere como trabalho futuro, a implementação de uma arquitetura servidor-cliente onde são guardados dados sobre os utilizadores, como: *inputs* favoritos, modos de funcionamento favoritos, histórico de decisões, entre outros que possam melhorar o funcionamento do sistema.

Por oposição à arquitetura servidor-cliente sugerida no parágrafo anterior, o projeto [3], implementa um sistema local de perfis de utilizador onde, entre outros possíveis dados, são guardados os valores de amplitude de movimentos lidos na funcionalidade de calibração do sistema.

Esta característica de sistema é considerada útil para um produto versátil, ou seja, que possa ser utilizado por mais do que um utilizador do público alvo. Existindo a necessidade de um produto destas características ser utilizado em Centros de Medicina de Reabilitação, como é o caso do CMRA, é crucial que o sistema tenha a possibilidade de ser calibrado para diferentes utilizadores e que essa calibração seja guardada até não ser mais precisa (quando o utilizador deixa de necessitar da utilização deste tipo de cadeira ou deixa de ser utente do Centro de Medicina).

2.2.6. Alarme médico

Em três dos projetos estudados é referida uma funcionalidade de preocupação com a saúde do utilizador. Esta funcionalidade resume-se à interpretação por parte do sistema de uma perda dos sentidos do utilizador que irá originar um *PowerOff* dos motores que permitem a deslocação da cadeira. A interpretação da perda dos sentidos é feita através da leitura de uma posição desmaiada da cabeça do utilizador, leitura feita pelo sensor colocado na cabeça do mesmo (sensor de inclinação ou acelerómetro) [8-9], ou através do deslocar da posição correta do sensor colocado nos ombros, deslocação provocada de igual modo pelo desmaio deste [11]. Este último sensor, colocado nos ombros do utilizador, é exclusivo do projeto [11] e corresponde a uma fita magnética com reação idêntica à de um condensador, ou seja, quando existe algum material entre as duas placas (corpo do utilizador) as cargas elétricas armazenadas mantêm-se nestas, quando esse corpo (dielétrico) deixa de existir entre estas, as cargas passam de uma para outra. Desta forma, analisando a condução ou não de corrente elétrica, o sistema consegue perceber se o sensor se encontra ou não colocado nos ombros do utilizador e consequentemente se este se encontra consciente ou, pelo contrário, se perdeu os sentidos.

2.3. Testes aos protótipos

Segundo os autores dos projetos estudados, os testes aos respetivos sistemas de um modo geral obtiveram resultados positivos. De salientar apenas, que nem todos os testes realizados foram executados por utilizadores com necessidades motoras e que alguns dos *inputs* dos projetos de análise multimodal, nomeadamente o controlo por comandos de voz, não obtiveram resultados positivos.

2.4. Dúvidas sobre as implementações estudadas

Dos artigos estudados ficam algumas dúvidas por esclarecer; a saber:

- não se sabe como foram analisados os comandos de controlo através dos movimentos da cabeça do utilizador [12-14];
- não se compreende como são enviados mais do que um *input* para o controlo da cadeira de rodas, assim como não se sabe como o sistema lida com estes num tempo eficiente para uma condução interrompida da cadeira [12-14];
- à exceção do projeto [9], não se compreende como é implementado o controlo sobre a velocidade de condução da cadeira de rodas;
- não se tem conhecimento da localização exata da interface gráfica assim como não se sabe que influência tem o olhar para esta nos comandos de condução através de movimentos de cabeça [3-5], [7-10], [12-14].

2.5. Limitações encontradas nos sistemas

Além das dúvidas apresentadas, identificam-se algumas limitações aos projetos, a saber:

- uso ineficiente dos comandos de voz;
- abordagem considerada desadequada à condução da cadeira;
- limitação económica;
- limitação ergonómica.

Como referido pelos autores dos projetos, a utilização de comandos de voz para controlo do sistema não é eficiente, entre outras causas devido à má perceção destes causada pelo ruído ambiente. Não tendo sido referido pelos autores as consequências de tal limitação, fica também por esclarecer a dimensão da redução do desempenho dos sistemas provocada pela limitação em causa [12-14].

O projeto [3] destaca-se dos restantes por apresentar dois modos de funcionamento com base em movimentos de cabeça. Um deles é igual ao dos restantes projetos, utilizando as quatro direções de movimento da cabeça (anterior/posterior e laterais) para conduzir a cadeira. O outro, utiliza apenas os movimentos anterior/posterior para seleccionar qual o comando que o utilizador pretende dar à condução da cadeira (frente/trás/esquerda/direita). Embora o segundo modo possibilite a utilização do sistema por um maior número de indivíduos, entende-se que este não seja adequado para o propósito a que se destina. O intervalo de tempo que irá existir entre o desejo do utilizador de alterar a deslocação da cadeira e a consequente alteração, irá ser demasiado grande para uma condução eficiente. Ineficiência que, em vez de trazer um bem-estar ao utilizador por se conseguir deslocar autonomamente, irá trazer um mal-estar a este por não conseguir comandar a condução da cadeira como desejaria.

A utilização de vários equipamentos de input, de um PC para processamento do sistema e de equipamentos como o *Emotiv EPOC Headset* condicionam financeiramente os projetos [3], [6], [12-14]. No caso do projeto [13], os autores apresentam o valor de 1500 US\$ como quantia necessária para aquisição de todos os equipamentos, valor considerado bastante avultado. Relativamente à implementação do equipamento *Emotiv EPOC Headset* no projeto [3] considera-se que o seu valor comercial, 399 US\$, é elevado para a utilização atribuída. Como será apresentado no capítulo 5., existem equipamentos de igual utilidade com valor oito vezes inferior ao deste.

A utilização de vários equipamentos de *input*, de arquiteturas com fios e de equipamentos intrusivos, como é o caso do *Shoulder Sensor*, condicionam ergonomicamente os projetos [6], [8], [10-14]. Na presente dissertação considera-se como não ergonomia, a utilização de vários equipamentos de *input*, uma vez que estes além de ocuparem bastante espaço na arquitetura da cadeira de rodas eléctrica, prejudicam o aspeto visual da mesma e limitam a atenção do utilizador a outras tarefas que tenha para além do controlo da cadeira de rodas [12-14]. Considera-se que uma arquitetura, que necessite de cabos para a comunicação entre os equipamentos utilizados para comandar a cadeira e o processador do sistema, não é ergonómica por condicionar a área que o utilizador necessita para através dos seus movimentos conseguir comandar a condução da cadeira [6], [8], [10-14]. No que diz respeito a equipamentos como o *Shoulder Sensor* [11], identificam-se estes como intrusivos, uma vez que para serem utilizados necessitam de ser colocados em zonas particulares do corpo. Zonas que levantam certos cuidados na colocação, para que o equipamento não comprometa o funcionamento do sistema, e que podem provocar algum desconforto ao utilizador.

2.6. Síntese

Do estudo realizado neste capítulo, destacam-se os seguintes aspetos:

- para realizar o processamento do sistema, identifica-se como vantajosa a utilização de uma PD (equipamento que contém um microcontrolador) por oposição a um PC. Uma PD apresenta menores dimensões, necessita de menores recursos energéticos e tem um valor comercial consideravelmente mais baixo;
- uma arquitetura *wireless* proporciona um sistema mais ergonómico. A não existência de cabos, que dificultam a mobilidade do utilizador, na estrutura do sistema é considerada benéfica;
- de um modo geral, os movimentos de cabeça adotados para a condução da cadeira de rodas baseiam-se nos movimentos efetuados no tradicional *joystick* das cadeiras elétricas;
- quatro projetos estudados incluem no seu sistema uma função de calibração. Esta função serve não só de ferramenta para aperfeiçoar a condução da cadeira de rodas como para permitir que mais do que um utilizador possa usufruir do mesmo sistema. Partindo do pressuposto que diferentes utilizadores irão ter diferentes amplitudes de movimento, uma função de calibração, função que adapta os movimentos de comando à mobilidade do utilizador, torna-se fulcral para que o sistema possa ser utilizado por utilizadores com diferentes graus de mobilidade;
- uma função que detete uma indisposição do utilizador, é considerada importante por mais do que um projeto. Tendo em consideração as limitações que o público alvo apresenta, limitações que tornam os indivíduos dependentes de outros, é de relevante importância que um sistema que traga alguma independência a estes, tenha também, na sua implementação, uma função que salvguarde o bem-estar destes;
- cinco projetos adicionam ao seu funcionamento de controlo da condução da cadeira de rodas, sistemas de deteção de obstáculos a partir de tecnologias como sonares e infravermelhos. Funcionalidade considerada importante e que irá também ser implementada no sistema poRsCHE.

Capítulo 3. Definição do sistema poRsCHE

Concluído o Estudo do Estado da Arte, onde foram identificadas as limitações e as mais valias dos projetos desenvolvidos com objetivo idêntico ao do presente, investiga-se qual o protocolo *wireless* que melhor se adequa aos objetivos propostos, define-se a estrutura e o funcionamento geral do sistema que se pensa ser o melhor para alcançar todos os objetivos traçados e, com auxílio dos técnicos e pacientes do CMRA, definem-se os movimentos que irão servir de comando à condução da cadeira de rodas.

3.1. Seleção do protocolo *wireless*

No capítulo 1. da presente dissertação, foram levantadas algumas questões para investigação. Qual o protocolo *wireless* a implementar, foi uma dessas questões.

Nesta secção apresenta-se a investigação realizada com intuito de perceber qual o melhor protocolo *wireless*, em termos energéticos, a implementar num sistema embebido com recurso a sensores.

Na investigação realizada foram considerados oito protocolos: UWB, Wi-Max, Wi-Fi, Bluetooth, BLE, ZigBee, GSM/GPRS e ANT. Através da tabela 2 apresentam-se as principais características dos protocolos.

Protocolo	UWB	Wi-Max	Wi-Fi	Bluetooth	BLE	ZigBee	GSM/GPRS	ANT
Espetro	3.1 – 10.6 GHz	2.4; 5.1 – 66 GHz	2.4; 5 GHz	2.4 GHz	2.4 GHz	2.4 GHz	850/900; 1800/1900 MHz	2.4 GHz
Alcance	10 – 102 m	0.3 – 49 Km	100 m	10 – 100 m	10 – 100 m	10 – 100 m	2 – 35 Km	30 m
Débito	110 Mbps	35 – 70 Mbps	54 Mbps	1 – 3 Mbps	1 Mbps	250 Kbps	168 Kbps	60 Kbps
Potência Tx	-41.3 dBm /MHz	23 dBm	15 – 20 dBm	0 – 10 dBm	0 dBm	0 dBm	0 – 39 dBm	0 dBm
Número canais RF	15	10 – 20	14;64	79	37	16	124	125

Tabela 2 – Protocolos *wireless* estudados [15-19]

Para sistemas como o presente, de curto alcance e com alimentação energética à base de baterias, os artigos [15] e [16] referem que os protocolos mais indicados são o Bluetooth e o ZigBee. Segundo testes apresentados nestes artigos (Tabela 3 e Gráfico 1), o protocolo que apresenta menor consumo energético é o ZigBee (24,7mA para transmissão de dados e 27mA para receção) seguido do Bluetooth (54mA para transmissão de dados e 47mA para receção).

Protocolo	Bluetooth	UWB	ZigBee	Wi-Fi	Wi-Max	GSM/GPRS
Módulo rádio testado	BlueCore2	XS110	CC2430	CX53111	AT86 RF535A	SIM300
VDD (volt)	1.8	3.3	3.0	3.3	3.3	3
Tx (mA)	57	~227	24.7	219	320	350
Rx (mA)	47	~227	27	215	200	230
Débito (Mbps)	0.72	114	0.25	54	70	0.164

Tabela 3 – Consumos energéticos dos protocolos analisados nos artigos [15-16]

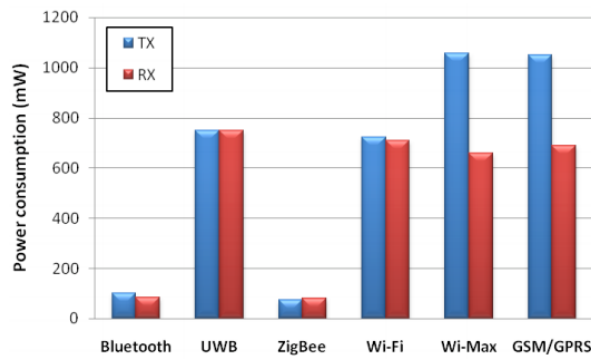


Gráfico 1 – Consumos energéticos, em mW, dos protocolos analisados nos artigos [15]

Não obstante o ZigBee apresentar menores consumos energéticos, os mesmos artigos [15] e [16], referem que em termos de eficiência energética, o protocolo que apresenta melhores resultados é o Bluetooth. Para a mesma quantidade de dados transmitidos, o protocolo Bluetooth consome menos energia do que o protocolo ZigBee.

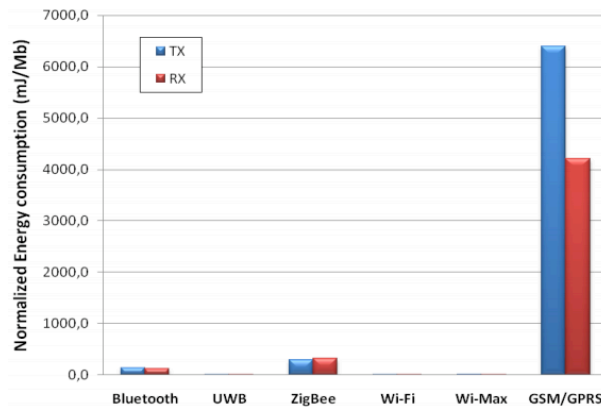


Gráfico 2 – Consumo energético, em mJ/Mb, dos protocolos analisados nos artigos [15]

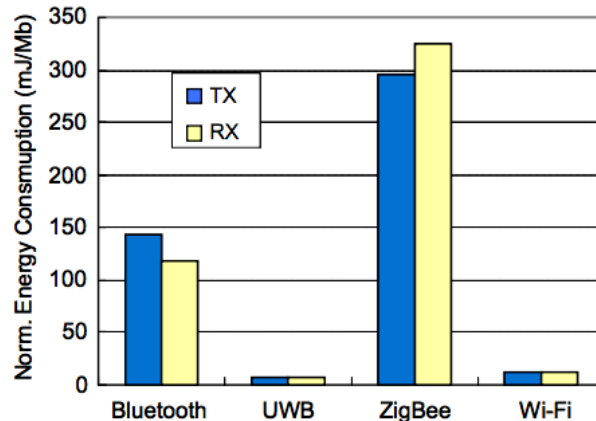


Gráfico 3 – Consumo energético, em mJ/Mb, dos protocolos analisados nos artigos [16]

Enquanto que o protocolo ZigBee apresenta valores de 300mJ/Mb para transmissão de dados e 325mJ/Mb para receção de dados, o protocolo Bluetooth apresenta valores inferiores a metade dos apresentados pelo primeiro.

Através dos resultados apresentados pode-se concluir que dos seis protocolos avaliados, o que apresenta melhor avaliação em termos energéticos é o Bluetooth.

Além de apresentar melhores resultados na avaliação energética realizada, o protocolo Bluetooth também apresenta uma maior taxa de transmissão de dados quando comparado com o ZigBee, o que favorece a eficiência do sistema que incorpora.

O artigo [17], além de avaliar o protocolo ZigBee, já avaliado pelos artigos [15] e [16], avalia mais dois protocolos, o BLE (Bluetooth Low Energy) e o ANT. Os testes realizados em [17] além de avaliarem a energia despendida por módulos rádio dos protocolos em análise, avaliam essa energia tendo em consideração os dois períodos existentes numa rede de sensores (período ativo (troca de dados – configuração do equipamento que integra o sensor e envio dos valores medidos pelo sensor) e período inativo (inexistência de troca de dados – períodos de tempo que não sendo necessários para envio de valores medidos pelos sensores, são aproveitados para economizar a corrente elétrica)).

Protocolo	BLE	ZigBee	ANT
Sensibilidade Rx	-87 dBm	-102 dBm	-85 dBm
Potência Tx	0 dBm	0 dBm	0 dBm
Alteração Frequência da Portadora	Sim	Não	Não
Espetro	2.4 GHz	2.4 GHz	2.4 GHz
Módulo rádio	CC2450	XBee S2	AP2
Período entre envio de pacotes de aviso¹	100 ms	100 ms	10 ms
Período entre pacotes de dados	100 ms	100 ms	250 ms

Tabela 4 – Características dos módulos rádio testados no artigo [17]

Terminados os testes aos módulos rádio apresentados na tabela 4, testes que foram realizados com diferentes intervalos de descanso (tempo em que os equipamentos com sensores não enviaram dados), os autores do artigo [17] concluíram que o protocolo mais eficiente é o BLE.

Protocolo	BLE	ZigBee	ANT
Tempo para estabelecimento da ligação	1150 ms ± 260 ms	250 ms ± 9.1 ms	930 ms ± 230 ms
Consumo energético Período descanso	0.78 uA	4.18 uA	3.1 uA
Consumo energético Período transmissão	4.5 mA	9.3 mA	2.9 mA
Consumo energético mínimo	10.1 uA	15.7 uA	28.2 uA
Intervalo de tempo ótimo para o período de descanso	10.0 s	14.3 s	15.3 s

Tabela 5 – Consumos energéticos dos protocolos analisados nos artigos [17]

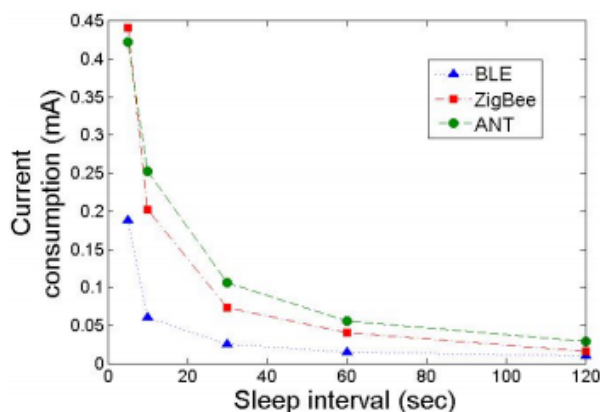


Gráfico 4 – Consumo energético dos protocolos analisados nos artigos [17]

Como ilustrado no gráfico 4, o BLE apresenta em todos os testes realizados valores de consumo energético consideravelmente abaixo dos outros dois protocolos que rivalizam com este. Para transmissões com períodos de descanso entre 0 e 20 segundos, o BLE apresenta consumos energéticos entre 0,05 e 0,2mA; para transmissões com períodos de descanso superiores a 20 segundos, o BLE apresenta consumos energéticos próximos de 10,1uA, valores bastante baixos que permitirão que uma simples bateria CR2032 (pilha de lítio tipo moeda) mantenha o sistema em funcionamento por vários meses.

O artigo [18] apenas compara os protocolos BLE e ZigBee e corrobora o artigo [17]. Segundo os autores deste artigo, o BLE apresenta melhores consumos energéticos do que o seu rival, apresentando como resultados a testes os seguintes valores:

Consumo na transmissão de pacotes de aviso (equipamento que integra o sensor – <i>slave</i>)	0.2 mW
Consumo na descoberta de pacotes de aviso (equipamento destinatário dos valores medidos pelo sensor – <i>master</i>)	33 mW
Receção de dados (<i>slave</i>)	66 mW
Transmissão de dados (<i>slave</i>)	84 mW
Receção de dados (<i>master</i>)	90 mW
Transmissão de dados (<i>master</i>)	72 mW
Período de inatividade	~1 uA

Tabela 6 – Consumos energéticos do protocolo BLE [18]

Embora os autores do artigo [18] não apresentem comparação de valores de consumo entre os dois protocolos, concluem que o protocolo BLE é superior ao ZigBee do ponto de vista energético, apresentando valores bastante atrativos de consumo de energia por bit transmitido.

Entre os protocolos Wi-Fi, Bluetooth e BLE, o artigo [19] considera que o BLE é o protocolo que melhor se adequa para um sistema de sensores aplicado à monitorização médica. Os autores deste artigo referem que o protocolo é o único que preenche todos os requisitos para um sistema deste tipo: interoperabilidade, baixo consumo energético, personalização das comunicações, compatibilidade eletromagnética com equipamentos médicos, transmissão segura e ligação com equipamentos que possuam ligação à Internet e às Telecomunicações Móveis.

Além dos autores do artigo [19] corroborarem os baixos valores de consumo energético já apresentados sobre o protocolo BLE, estes evidenciam a importância deste protocolo para utilizadores que necessitem de auxílio médico. Através da ligação do protocolo BLE com protocolos de Internet e de Telecomunicações Móveis suportada por equipamentos como *SmartPhones* e *Tablets*, o sistema ganha a capacidade de envio de

alertas de emergência rápidos e eficazes. A título de exemplo os autores referem o seguinte: se um sistema que meça os níveis de glicemia de um paciente detetar que este se encontra em estado de hipoglicemia (níveis baixos de glicemia), este, através de um *SmartPhone* ou *Tablet*, despoleta dois alarmes. O primeiro avisa o paciente para o seu estado, dando-lhe uma indicação que deverá comer algo para equilibrar os níveis de glicemia. Caso o paciente atenda ao alarme, comendo algo, o sistema continua o seu normal funcionamento; caso o paciente não atenda ao alarme, este despoleta um segundo alarme. O segundo alarme, enviado através da rede Internet ou de uma rede de Telecomunicações Móveis, é dirigido a um responsável pelo paciente (familiar ou profissional de saúde) por forma a que este possa prestar o auxílio necessário.

Tendo como objetivo a implementação de um alarme de emergência que avise alguém de cuidado, que o utilizador do sistema poRsCHE se encontra indisposto, a presente dissertação terá em consideração a implementação proposta pelo artigo [19].

Concluída a investigação apresentada e tendo em consideração todos os dados estudados, a seleção do protocolo para a implementação do sistema poRsCHE recai sobre o protocolo BLE (protocolo de baixos consumos energéticos, que possibilita o funcionamento do sistema com alimentação através de baterias por longos períodos de tempo e que tem capacidade de comunicação com equipamentos que possuam ligação a outros protocolos, nomeadamente protocolos de Internet e de Telecomunicações Móveis).

No capítulo 4. é apresentada a estrutura, o funcionamento do protocolo BLE e uma introdução da sua utilização no sistema proposto.

3.2. Estrutura do sistema

Para cumprir os objetivos traçados em 1.2., o sistema poRsCHE é implementado com cinco equipamentos diferentes:

- para detetar os movimentos de cabeça do utilizador, o sistema irá utilizar uma PD com um sensor de movimento integrado. Esta PD detetará os movimentos efetuados através do sensor e enviará a informação para o processador do sistema através de um módulo rádio BLE;
- para conduzir a cadeira de rodas, o sistema irá utilizar uma PD com portas de *input/output* programáveis e com um módulo rádio BLE. Recebendo a informação sobre o movimento que o utilizador efetuou com a cabeça, esta PD irá comandar os motores da cadeira através de sinais PWM⁵ enviados pelas suas portas *output* programáveis;
- para detetar obstáculos, de modo a evitar colisões, o sistema irá utilizar um sonar. Através da receção de uma reflexão da onda sonora emitida, o sistema conseguirá saber da existência de um obstáculo no seu caminho. Ao ser calculado um intervalo de tempo entre a emissão da onda sonora e a receção do seu eco, que identifique um obstáculo a uma distância igual ou inferior a metade do comprimento da cadeira, o sistema interromperá a sua deslocação;
- para informar o utilizador sobre o seu corrente funcionamento, o sistema irá utilizar uma interface gráfica com recurso a três LEDs. Um dos LEDs indicará ao utilizador a disponibilidade de inserção de um novo comando (o utilizador pode efetuar um novo movimento para que seja interpretado como comando de controlo). Os restantes LEDs, indicarão qual o estado atual do sistema: *Standby* (o utilizador pode movimentar a cabeça livremente; estado implementado para que o utilizador possa efetuar outras tarefas, como conversar com alguém ou interagir como o seu telemóvel ou computador portátil, sem que os movimentos que realize sejam interpretados como movimentos de comando de condução) ou *Driving* (estado onde o sistema interpreta os movimentos de cabeça como movimentos de comando de condução). No capítulo 6. é apresentada a implementação dos dois estados mencionados.
- para calibrar os movimentos de comando e para envio de um alerta de emergência, caso o utilizador se sinta indisposto, o sistema irá utilizar um *SmartPhone*. Através de uma aplicação móvel, o sistema irá disponibilizar: uma primeira funcionalidade que, conectando-se com a PD com o sensor de movimento integrado, permitirá ao utilizador, com auxílio de uma segunda pessoa, calibrar os movimentos que irão servir de comando à condução da cadeira de rodas; uma segunda funcionalidade que, recebendo uma *flag*⁶ do processador do sistema, enviará uma SMS ao contacto definido como contacto do cuidador do utilizador. No capítulo 6. é apresentada a implementação da aplicação móvel mencionada.

⁵ Sinais *Pulse Width Modulation*, são sinais retangulares que sinalizam uma determinada informação consoante o seu período a *High* (tensão positiva de corrente elétrica). No caso do sistema poRsCHE, o período irá indicar qual o sentido em que a roda motora irá rodar e qual a sua frequência de rotação.

⁶ Valor definido para sinalizar um acontecimento.

A escolha do último equipamento, um *SmartPhone*, prende-se a três razões:

1. a primeira razão é económica. Podendo o sistema recorrer às capacidades de um equipamento adquirido à priori pelo utilizador, o custo financeiro do sistema fica contido ao valor dos restantes três equipamentos;
2. a segunda razão prende-se à implementação do alerta de emergência. Através de um *SmartPhone*, o sistema passa a ter acesso às Comunicações Digitais Móveis, podendo através destas alertar eficaz e atempadamente quem de direito de uma possível indisposição do utilizador. Proposta de implementação considerada no artigo [17] e apresentada na secção **3.1.**;
3. a terceira razão prende-se com a crescente utilização deste tipo de equipamentos. Com o desenvolvimento das Telecomunicações Móveis, não só os equipamentos se tornaram economicamente mais acessíveis como também são utilizados, cada vez mais, para diversas funcionalidades. Em consequência da crescente utilização, os *SmartPhones* tornaram-se um equipamento do dia-a-dia, de fácil compreensão e utilização.

No capítulo **5.** é apresentada a seleção de todos os equipamentos aqui mencionados bem como a integração destes para formar o protótipo que irá servir de prova ao funcionamento do sistema embebido proposto.

3.3. Definição dos movimentos de comando

Nesta secção definem-se os movimentos que irão servir como comandos de controlo ao sistema.

Segundo o estudo feito no capítulo 2., Revisão da Literatura, os movimentos adotados, por um maior número de projetos, baseiam-se nos movimentos efetuados no tradicional *joystick* das cadeiras de rodas elétricas.

Embora a abordagem, descrita no parágrafo anterior, permita tantas movimentações da cadeira quantas as que existem nas tradicionais cadeiras de rodas elétricas, entende-se que ao nível da saúde da coluna vertebral do utilizador, esta não seja a mais adequada. Ao exigir-se que o utilizador mantenha a cabeça, durante vários segundos, na posição de comando que pretende dar à condução da cadeira, está a obrigar-se que a coluna do utilizador tenha durante todo esse intervalo de tempo uma posição não natural que irá resultar em dores e possíveis lesões.

Para contornar a limitação exposta no parágrafo anterior e querendo alcançar um nível intuitivo de comando como o exposto, o sistema poRsCHE apresenta os seguintes movimentos: para dar ordem de comando de condução em frente, movimento de flexão (movimento ilustrado na figura 9 – flexão e conseqüente retorno da cabeça à posição vertical para que o sistema interprete o movimento como um comando); para parar a cadeira, colocá-la em marcha-atrás e alternar os estados do sistema (*Standby* e *Driving*), movimento de extensão (movimento ilustrado na figura 10 – extensão e conseqüente retorno da cabeça à posição vertical para que o sistema interprete o movimento como um comando); para rodar a cadeira, movimento de inclinação lateral para o lado pretendido (movimento ilustrado na figura 11 – inclinação da cabeça para que a cadeira rode e retorno à posição vertical para que esta pare de rodar).

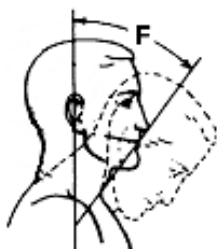


Figura 9 – Movimento de flexão

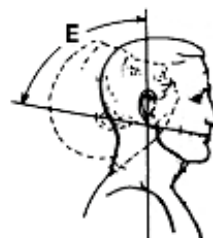


Figura 10 – Movimento de extensão

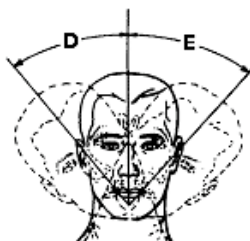


Figura 11 – Movimento de inclinação lateral

Para a definição dos comandos acima apresentados tem-se em consideração os seguintes aspetos:

- o público alvo tem capacidade motora para movimentar a cabeça segundo os movimentos anterior/posterior e laterais;
- para não afetar a saúde do utilizador, a posição prevalecente de comando do sistema tem que ser a posição natural da coluna vertebral. O sistema não pode obrigar o utilizador a ter a cabeça, durante mais do que alguns segundos, numa posição que não a vertical.

Para conseguir um compromisso entre uma eficaz condução lateral da cadeira de rodas e um movimento de cabeça ergonómico para este tipo de movimento, optou-se por um movimento rotacional da cadeira por oposição a um deslocamento angular.

No capítulo **6**, é apresentada a implementação dos comandos de controlo com base nos quatro movimentos de cabeça expostos.

Capítulo 4. Protocolo *Wireless* BLE

No capítulo anterior, capítulo 3., foi selecionado o protocolo a ser implementado na arquitetura do sistema poRsCHE. O protocolo selecionado denomina-se BLE (Bluetooth Low Energy ou, também designado, Bluetooth Smart).

Neste capítulo expõe-se os temas considerados mais relevantes à compreensão do funcionamento das comunicações *wireless* deste protocolo.

4.1. Introdução

O Bluetooth é um protocolo rádio desenvolvido e gerido pelo SIG (Bluetooth Special Interest Group) que atualmente lidera as tecnologias *wireless* em termos de volume de vendas e de arquiteturas instaladas.

Anteriormente ao Bluetooth 4.0 (BLE), dispositivos *wireless* como auscultadores, ratos e teclados utilizavam o Bluetooth Clássico para transferência de grandes pacotes de dados a elevadas taxas de transmissão. Essas taxas refletem um elevado consumo energético e não permitem a utilização de baterias de pequenas dimensões nem da mesma bateria durante um elevado período de tempo.

Para solucionar a limitação exposta, o SIG criou o Bluetooth Smart. Através da redução da largura de banda, e consequentemente das taxas de transmissão de dados, o SIG conseguiu criar um protocolo de consumo energético tão baixo que permite que a mesma bateria alimente o sistema por vários meses.

Além do Bluetooth Clássico e do BLE, existe uma outra tecnologia *wireless* que garante a interligação destas duas, a esta, o SIG dá o nome de Bluetooth Smart Ready. Com a criação desta tecnologia o SIG garantiu que equipamentos de ampla utilização, como o caso de portáteis e *SmartPhones*, tenham capacidade de comunicação tanto com dispositivos desenvolvidos com base no Bluetooth Clássico como com dispositivos que são implementados sobre BLE [20-25].







Bluetooth low energy (Bluetooth Smart)	Dual Mode Bluetooth	Classic Bluetooth
		
Connect low power applications to a smart phone or tablet	Bluetooth connection across any end point (Bluetooth low energy or classic); enables bridge between Bluetooth low energy (Bluetooth Smart) and classic	Bluetooth connection with high data rate (up to 3Mbps)
		
<ul style="list-style-type: none"> • Custom profile, allows any application • Multiyear on Coin Cell Battery • Less than 200Kbps data rate 	<ul style="list-style-type: none"> • Supports new and old phones (BT and BLE) • Up to 3Mbps data rate 	<ul style="list-style-type: none"> • A2DP Profile to stream music from phones • Up to 3Mbps data rate

Figura 12 – Os três protocolos Bluetooth [25]

Technical Specification	Classic Bluetooth	Bluetooth Smart
Primary use cases	Mobile phones, gaming, headset, stereo audio, automotive, PC HID	Large number of use-cases due to proprietary profiles.
Application throughput	Up to 3 Mbps	Up to 0.2 Mbps
Power consumption	Can run multiple years on 2 x AAA batteries	Can run multiple years on coin cell
Latency (from a non connected state)	100 ms	< 3 ms
High quality audio	Yes	No
Voice quality audio	Yes	Yes
Nodes/Active slaves	7	unlimited
Network topology	Scatternet	Star topology
Profiles	Adopted SIG profiles	Adopted SIG profiles Proprietary profiles

Tabela 7 – Principais diferenças entre o Bluetooth Clássico e o BLE [21]

A tabela 7 ilustra as principais diferenças entre o Bluetooth Clássico e o BLE (Bluetooth Smart). Como anteriormente referido, podemos verificar que a taxa de transmissão de dados ("*Application throughput*") é menor no BLE, principal razão para um menor consumo energético. Contudo, não tendo taxas de transmissão tão elevadas quanto o Bluetooth Clássico, o BLE apresenta valores de latência bastante mais baixos. Estes valores representam menores intervalos de tempo entre a ordem de comando e a sua execução e que se refletem numa maior eficiência nas aplicações desenvolvidas.

Para além do consumo energético e dos valores de latência, bastante diferenciados entre as duas tecnologias, existem mais duas características interessantes e próprias do BLE que o evidenciam em relação ao Bluetooth Clássico. Enquanto que no Bluetooth Clássico, o número de nós (dispositivos periféricos conectados com o principal) está limitado a um valor de 7 unidades, o BLE não tem limites para o número de nós numa rede, o que permite a construção de um sistema sem limites na sua infraestrutura (característica que permite a construção de sistemas expansíveis, dinâmicos e ecléticos). A outra característica de relevante importância presente no BLE por oposição ao Bluetooth Clássico, é a possibilidade de utilização de perfis personalizados pelo Engenheiro do sistema; característica que permite o desenvolvimento de sistemas diversos e inovadores, que irão promover um forte crescimento do mercado BLE [20-25].

4.2. Arquitetura

O protocolo BLE é constituído por três níveis diferentes: *Application*, *Host* e *Controller*.

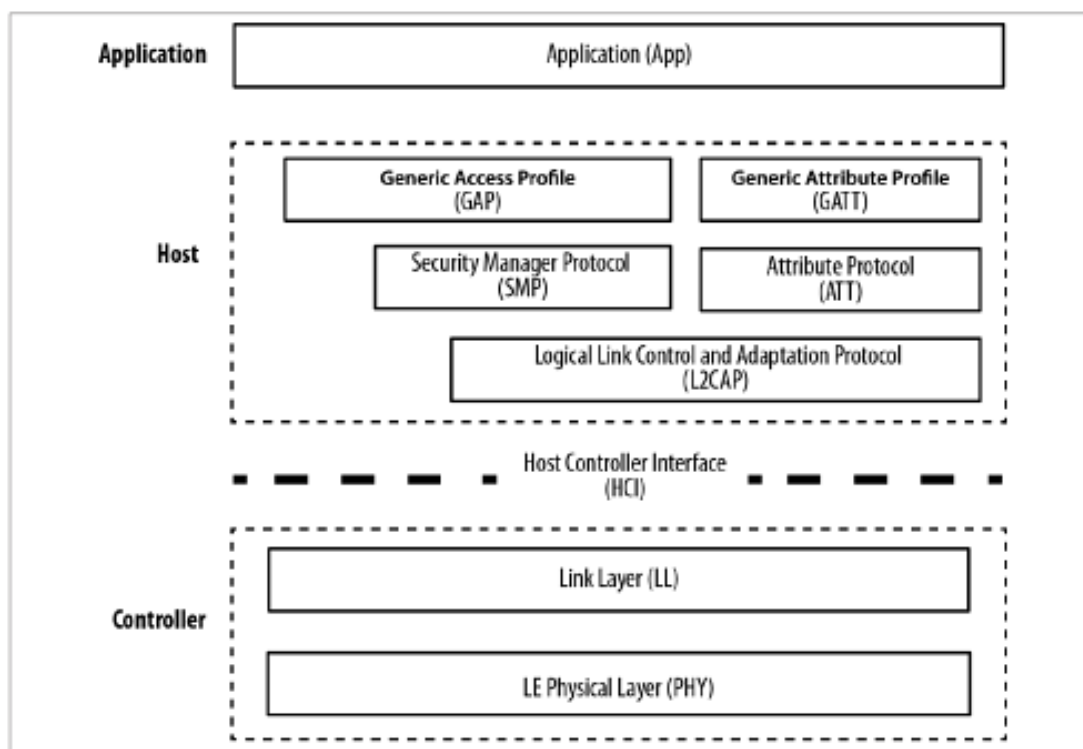


Figura 13 – Arquitetura BLE [20]

4.2.1. Application

O nível Aplicação (*Application*), nível topo do protocolo, define três tipos de especificações: Característica (*Characteristic*), Serviço (*Service*) e Perfil (*Profile*). Cada uma destas especificações é implementada em cima do GATT (*Generic Attribute Profile*). O GATT organiza os dados a serem trocados entre dispositivos em Perfis, Perfis públicos (desenvolvidos e publicados pelo SIG) ou Perfis privados (desenvolvidos por um Engenheiro de *Software* para um sistema específico) [26].

O Perfil representa o dispositivo enquanto informação que tem para trocar com os restantes dispositivos do sistema que integra e é organizado da seguinte forma:

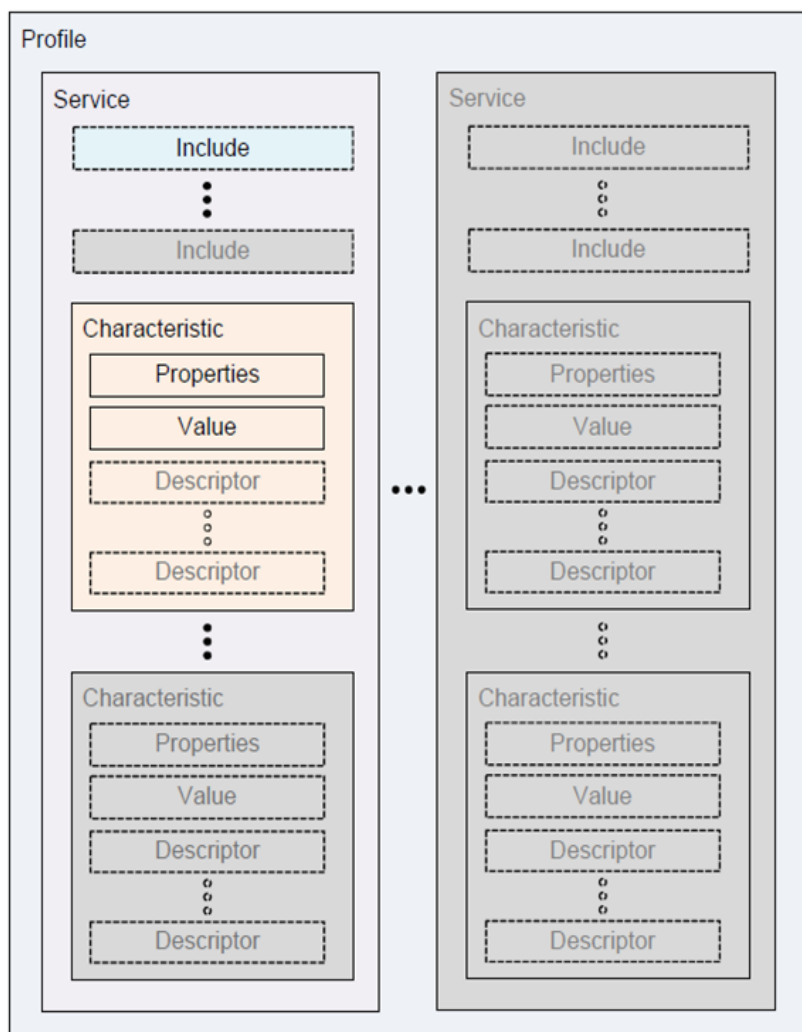


Figura 14 – Hierarquia GATT [24]

Cada Perfil tem um ou mais Serviços e cada Serviço tem uma ou mais Características. Cada Característica representa um tópico de informação a ser trocado com os outros dispositivos e é formada pelos seguintes elementos: Propriedades (*Properties* - informação sobre como é realizado o acesso ao valor, como este é apresentado e como é representado), Valor (*Value* - valor da variável a ser acessada pelos dispositivos) e Descritor (*Client Characteristic Configuration Descriptor* - elemento onde o outro dispositivo pode ativar ou desativar notificações (comunicação sem ACK⁷) ou indicações (comunicação com ACK⁸) de alteração de valor).

Cada um dos elementos anteriormente referidos tem um UUID (Identificador Universal Único) e é através deste que os dispositivos comunicam com cada elemento de cada um. Exemplificando: após o estabelecimento da ligação estar completo entre o dispositivo X e Y, para que o dispositivo Y (Cliente) tenha acesso ao Valor da Característica α do dispositivo X (Servidor), sempre que este sofrer alterações, este tem que aceder ao Descritor da Característica através do $UUID^{\text{Descritor}}$ para ativar as notificações de alteração de Valor (escrita no elemento, o valor 1) e posteriormente, sempre que receba uma notificação, terá que efetuar a leitura desse Valor através do $UUID^{\text{Valor}}$ [20-25].

⁷ *Acknowledgment*, em português: Confirmação.

⁸ O BLE tem um sistema de deteção de erros (CRC) que permite que os dispositivos detetem possíveis erros na comunicação e consequentemente peçam o reenvio do(s) pacote(s) em questão.

4.2.2. *Host*

O nível Anfitrião (*Host*) é considerado como o herói anónimo do Bluetooth. É o nível responsável por uma vasta lista de funções como: multiplexagem, negociação protocolar (negociação de como estabelecer a segurança na ligação), estabelecimento da ligação, gestão de troca de dados e desemparelhamento (término da ligação entre dois dispositivos).

A formar esta camada protocolar, estão presentes cinco elementos: o *Logical Link Control and Adaptation Protocol* (responsável pela multiplexagem dos pacotes de dados entre a camada inferior e as camadas superiores), o *Security Manager Protocol* (responsável pela configuração e gestão da segurança da ligação com os outros dispositivos), o *Attribute Protocol* (responsável pela organização da estrutura dos perfis BLE), o já mencionado *Generic Attribute Profile* (responsável pela organização e gestão da informação a ser trocada entre os dispositivos) e o *Generic Access Profile* (responsável pela gestão processual de transmissão de pacotes de aviso (*advertising*), pelo estabelecimento e gestão de ligação entre dois equipamentos (*connecting*) e pela gestão de futuro reconhecimento de dispositivos previamente emparelhados (*bonding*)).

De salientar, que é através do *advertising* (transmissão de pacotes de aviso) que os dispositivos conhecem a existência de outros nas suas proximidades, e que é através deste processo que toda a relação entre estes se inicia [20-23].

4.2.3. *Controller*

O nível Administrador (*Controller*), facilmente identificado pelo público como o módulo rádio, é composto pelas partes analógicas e digitais dos componentes de radiofrequência de modo a suportar a transmissão e receção de pacotes de dados. Comunica com o mundo exterior (outros dispositivos BLE) através de uma antena e comunica com o Anfitrião através da Interface Administrador–Anfitrião (*Host Controller Interface*).

A Interface Administrador–Anfitrião é o elo de ligação entre estes dois níveis. Esta interface, que pode ser implementada através de UART⁹, USB¹⁰ e SDIO¹¹, é essencial devido à crucial diferença entre os dois níveis protocolares. Enquanto que o Administrador lida com operações em tempo real, devido à sua ligação com a camada física da rede (*LE Physical Layer*), o Anfitrião executa operações que não são em tempo real devido à complexidade protocolar destas. Com esta interface consegue-se uma ligação entre os comandos mais alto nível, para leitura e escrita de dados, com os comandos da camada física, de transmissão e receção de dados.

A camada de enlace (*Link Layer*) é provavelmente o elemento mais complexo do protocolo BLE. Comunica diretamente com a camada física e é responsável pelo *advertising* (transmissão de pacotes de aviso), *scanning* (procura de pacotes de aviso de outros dispositivos) e *connecting* (estabelecimento e gestão de ligação entre dois equipamentos). Operações de custo computacional alto que justificam a implementação das mesmas ao nível *Hardware*, de modo a não sobrecarregar o processador central [20-23].

⁹ Tecnologia de ligação de componentes *Hardware Universal Asynchronous Receiver/Transmitter*

¹⁰ Tecnologia de ligação de componentes *Hardware Universal Serial Bus*

¹¹ Tecnologia de ligação de componentes *Hardware Secure Digital Input/Output*

De modo a executar as operações referidas, a camada de enlace tem implementadas as seguintes funcionalidades:

- preâmbulo (configuração da potência e frequência de transmissão),
- produção e verificação de CRC (*Cyclic Redundancy Check*),
- *Data whitening*¹²,
- geração aleatório de números de modo a alternar a frequência de transmissão (frequência da portadora),
- *Advanced Encryption Standard*¹³.

A camada física da rede (*LE Physical Layer*) é a responsável pelas transmissões e receções de dados através de ondas eletromagnéticas (sinais analógicos). Além do envio e receção de sinais analógicos por meio de ondas eletromagnéticas, a camada física da rede é também responsável pela modulação e desmodulação dos mesmos para símbolos digitais (tipo de dados com que as camadas superiores trabalham).

O módulo rádio, pertencente a esta camada protocolar, utiliza a largura de banda dos 2,4GHz ISM (*Industrial, Scientific and Medical*) para comunicar com os restantes dispositivos da mesma tecnologia e divide-a em quarenta canais, dos 2,4000GHz aos 2,4835GHz. Na figura 15 ilustram-se os quarenta canais referindo-se as suas respetivas utilizações.

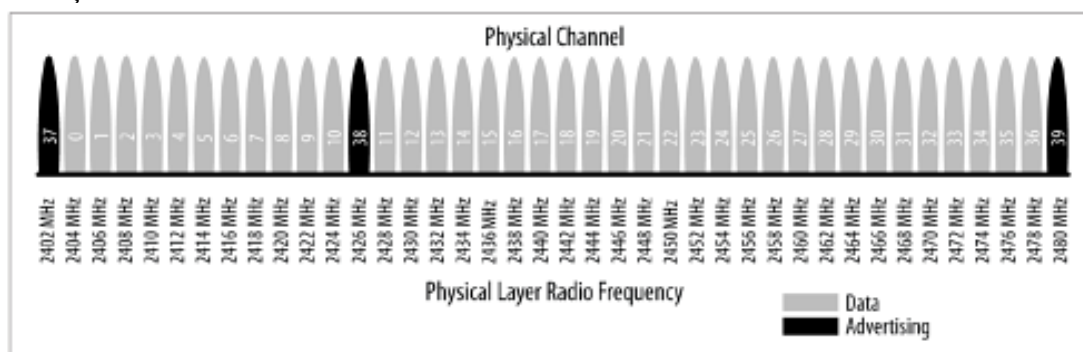


Figura 15 – Largura de Banda utilizada pelo BLE [20]

Dos quarenta canais, trinta e sete são de transmissão de dados e três (37, 38 e 39) são de transmissão de pacotes de aviso.

Como mencionado anteriormente, o protocolo BLE alterna a frequência de transmissão (frequência da portadora). A alternância da frequência da portadora é efetuada através da técnica FHSS (*Frequency-hopping spread spectrum* - espectro de difusão em frequência variável em tradução livre), o que permite uma menor probabilidade de interferência entre comunicações que utilizem a mesma largura de banda. A técnica FHSS utiliza a seguinte fórmula para alternar a portadora:

$$\text{próximo_canal} = (\text{canal_atual} + \text{salto}) \text{ mod } 37$$

A variável "salto" é um valor aleatório que é gerado pela camada de enlace.

Para modulação e conseqüente desmodulação dos sinais analógicos, o BLE utiliza a técnica GFSK (*Gaussian frequency-shift keying*), técnica utilizada por outros protocolos como o Bluetooth Clássico, e que utiliza um filtro Gaussiano para discriminar os desvios positivos/negativos de frequência (bit 1/0) [20-23].

¹² Método que consiste em baralhar a informação digital, através de uma palavra-chave, de modo a evitar a polarização conseqüente da corrente contínua aplicada na transmissão do sinal analógico. Para uma correta receção da mensagem, a mesma palavra-chave é aplicada pelo recetor para descodificar o sinal recebido.

¹³ As mensagens trocadas através do protocolo BLE são encriptadas através do protocolo de encriptação AES (*Advanced Encryption Standard*).

4.3. BLE *Device Address*

O identificador do dispositivo BLE, similar ao endereço MAC (*Media Access Control*) do protocolo Ethernet, denomina-se como *Bluetooth device address*. Este endereço, formado por seis bytes (quarenta e oito bits), identifica-se como sendo um UUID, pelo que existe a necessidade de garantir que na mesma rede de operações não exista mais do que um dispositivo com o mesmo endereço de comunicação.

Existem dois tipos de endereço para os dispositivos BLE:

- Endereço público: idêntico ao endereço MAC, nunca é alterado durante o tempo de vida do dispositivo. Endereço registado pela autoridade de registo IEEE¹⁴.
- Endereço privado aleatório: idêntico ao endereço IP, pode ser programável pelo Engenheiro do sistema ou gerado aleatoriamente pelo dispositivo. A segunda situação ocorre quando os dois dispositivos conectados acordam para segurança de ambos (*non-tracking*¹⁵) a sua utilização, sendo combinado entre estes um IRK (*Identity Resolving Key* - chave que lhes permite gerar endereços que ambos conheçam) [20-22].

¹⁴ Instituto dos Engenheiros Eletrotécnicos e Eletrónicos - organização profissional sem fins lucrativos, fundada nos Estados Unidos da América. É a maior organização profissional do mundo (em número de sócios).

¹⁵ Entende-se por *non-tracking*, a capacidade que o dispositivo tem de evitar que um possível atacante saiba qual a sua localização na topologia da rede e/ou a sua localização física (geográfica).

4.4. Topologia da rede

Os dispositivos implementados sobre o protocolo BLE podem ter quatro tipos diferentes de ação:

- *Broadcaster*,
- *Observer*,
- *Peripheral (server / slave)*,
- *Central (client / master)*.

4.4.1. *Broadcast*

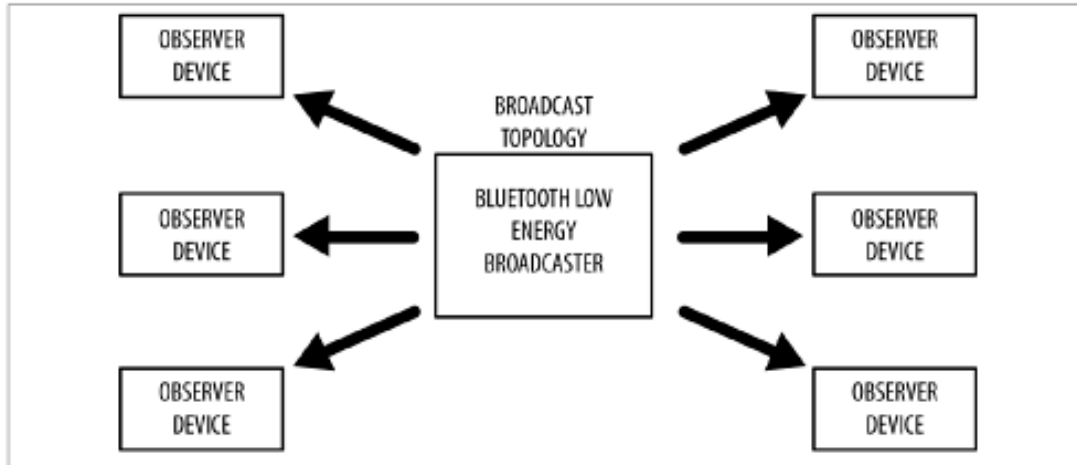


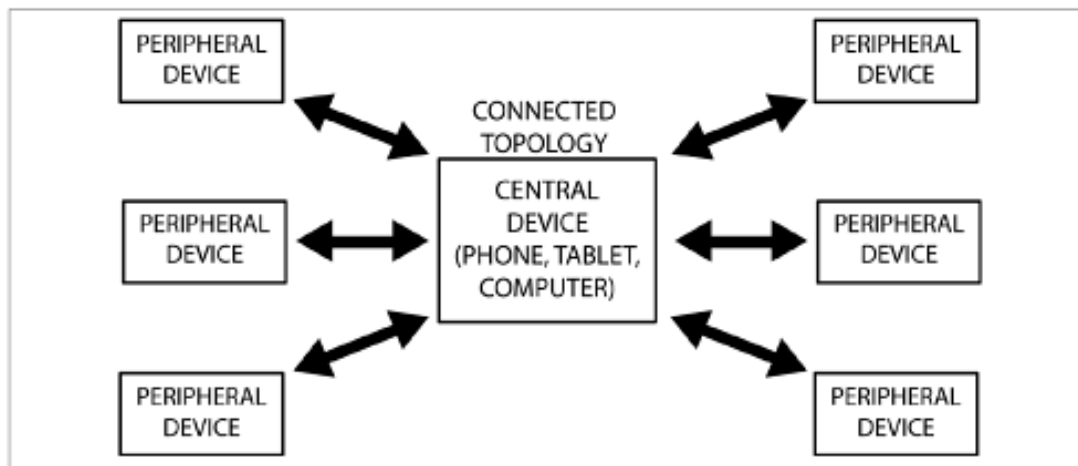
Figura 16 – Topologia BLE *Broadcast* [20]

A topologia de rede ilustrada na figura 16, apresenta os dois primeiros tipos de dispositivos listados.

O primeiro dispositivo, *Broadcaster*, envia periodicamente pacotes de aviso com o seu endereço único de identificação, o seu nome e com informação que queira partilhar com eventuais dispositivos que estejam na sua vizinhança.

O segundo dispositivo, *Observer*, efetua um scan a todas as frequências da largura de banda do BLE de modo a receber a informação que procura obter do primeiro tipo de dispositivos.

Neste tipo de topologia não existe estabelecimento de ligação [20-21].

4.4.2. *Connected*Figura 17 – Topologia BLE *Connected* [20]

A topologia de rede ilustrada na figura 17, apresenta o terceiro e quarto dispositivos.

O terceiro dispositivo, *Peripheral*, envia periodicamente pacotes de aviso e aceita pedidos de estabelecimento de ligação. Uma vez estabelecida a ligação com o dispositivo do tipo *Central*, este segue o seu *timing* (*Central timing*) de modo a existir uma troca regular de dados.

O quarto dispositivo, *Central*, efetua um scan a todas as frequências da largura de banda do BLE de modo a encontrar o dispositivo com quem se pretende conectar. Uma vez encontrado o dispositivo alvo, envia-lhe um pacote de pedido de estabelecimento de ligação, aguardando uma resposta favorável para que ambos possam trocar dados.

De salientar que na topologia *Connected*, embora se refira que a troca de dados é mútua, uma vez que para se configurar o correto funcionamento da ligação ambos têm que trocar dados, existe um tipo de dispositivo, *Peripheral*, que tem como função o envio de dados (informação útil ao sistema) e outro, *Central*, que tem como função a receção dos mesmos para conseqüente trato e execução das restantes funcionalidades do sistema em causa [20-21].

4.5. BLE na arquitetura poRsCHE

A arquitetura poRsCHE irá ser formada por três equipamentos que comunicarão através do protocolo BLE. Através da figura 18 representa-se a arquitetura proposta.

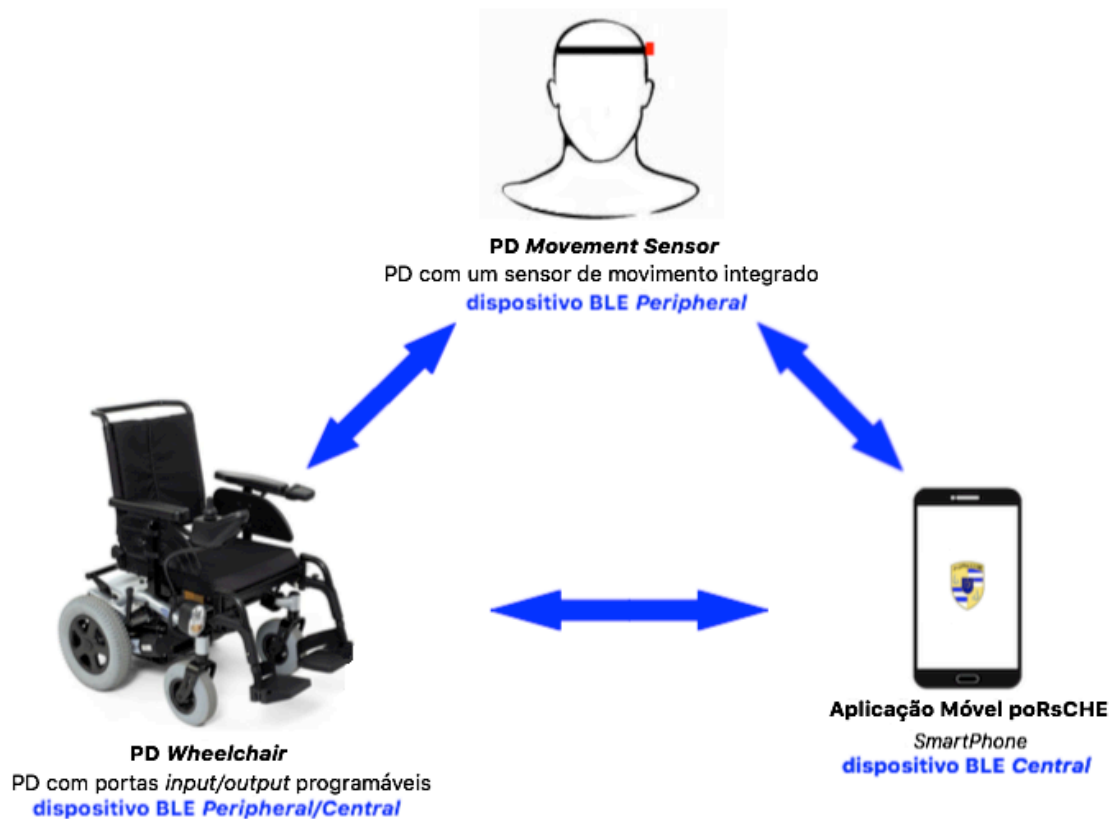


Figura 18 – Arquitetura poRsCHE

A *PD Movement Sensor* atuará como um dispositivo BLE *Peripheral*. Esta PD funcionará como um servidor, comunicando ao cliente (*SmartPhone* ou *PD Wheelchair*) os valores de movimento medidos.

O *SmartPhone* funcionará como um dispositivo *Central*. Embora já existam alguns *SmartPhones* equipados com módulos rádio que têm capacidade de operar nos dois modos protocolares, *Central* e *Peripheral*, a maioria destes equipamentos apenas está equipado com um módulo rádio com capacidade de operar no modo protocolar *Central*.

A *PD Wheelchair* irá ser implementada como um módulo rádio que operará nos dois modos protocolares, *Central* e *Peripheral*. Esta implementação deve-se à limitação apresentada no parágrafo anterior. Uma vez que a maioria dos *SmartPhones* apenas está equipada com um módulo rádio com capacidade de operar no modo protocolar *Central*, e existindo a necessidade de comunicação desta PD com a *PD Movement Sensor* e com o *SmartPhone*, é indispensável que esta seja implementada com um módulo rádio com capacidade de operar nos dois modos protocolares.

As explicações apresentadas nesta secção sobre as comunicações *wireless* do sistema poRsCHE irão ser desenvolvidas no capítulo 6..

Capítulo 5. *Hardware*

Definida a estrutura do sistema poRsCHE e o protocolo *wireless* a ser implementado, investiga-se quais os equipamentos que melhor poderão formar o sistema e apresenta-se como será implementado o protótipo que irá servir de prova ao funcionamento deste.

5.1. Seleção da PD com sensor de movimento

O equipamento estudado nesta secção é o mais importante do sistema poRsCHE. Não só por ser o equipamento que envia as ordens de comando para a condução da cadeira de rodas, mas também por ser o equipamento que, colocado na cabeça do utilizador, necessitará de ser o mais ergonómico possível (equipamento de reduzidas dimensões, leve, que incorpore um sensor de movimento, um módulo rádio e um sistema de alimentação energética à base de bateria).

No presente mercado eletrónico apenas um equipamento satisfaz as condições acima expostas. Este equipamento denomina-se SensorTag e é fabricado pela Texas Instruments. Disponibiliza três sensores de movimento: giroscópio, acelerómetro e magnetómetro; tem implementado um módulo rádio BLE e um sistema de alimentação energética com base numa bateria CR2032 (pilha de lítio tipo moeda), com duração de cerca de um ano; é de reduzidas dimensões: 5 x 6,7 x 1,4 cm; é revestido por uma proteção de borracha que evita possíveis estragos provocados por quedas; e é vendido pelo valor de 25,72€, valor considerado baixo face à soma de valores dos vários componentes que este integra [27].

Os três sensores, acima referidos, fazem parte de apenas um dos componentes que integra o SensorTag. O equipamento em questão é um sistema mecânico microeletrónico (MEMS) e caracteriza-se por ser um *chip*¹⁶ de reduzidas dimensões que explora as capacidades mecânicas do silicógeno para detetar movimento em direções distintas. O MEMS integrado no SensorTag é fabricado pela InvenSense, denomina-se MPU-9250 e caracteriza-se por ser a nova geração de MEMS desta marca, tendo dimensões 44% inferiores às anteriores versões e gastos energéticos de apenas 9,3 μA [28].

Nas figuras 19, 20 e 21 ilustra-se o equipamento fabricado pela Texas Instruments, fazendo-se referência às características acima descritas.

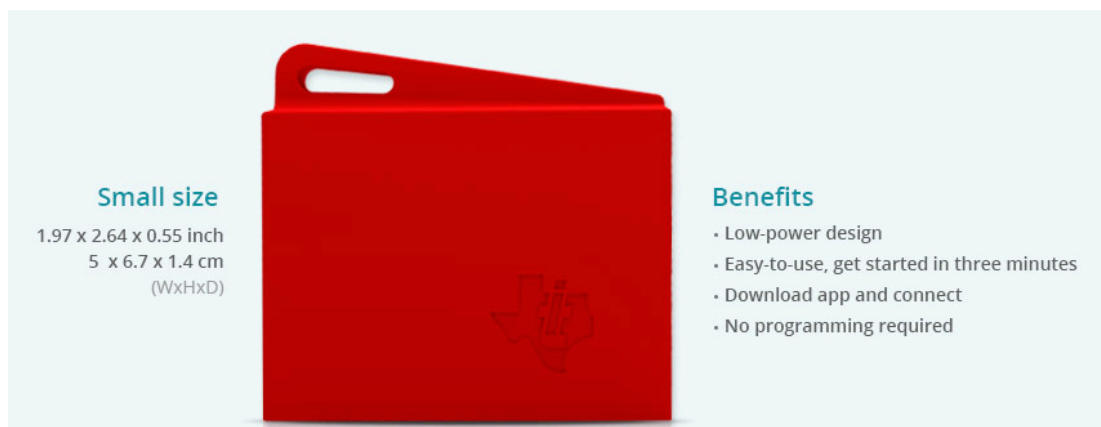


Figura 19 – SensorTag CC2650, dimensões e benefícios [27]

¹⁶ Circuito integrado que tem como suporte uma base de um material semicondutor na qual são gravados ou inseridos componentes eletrónicos que, em conjunto, desempenham uma ou mais funções.

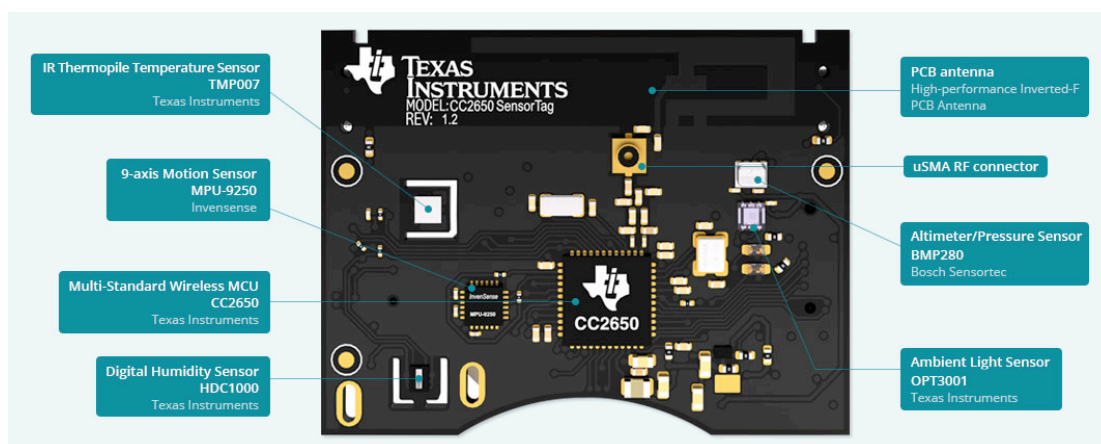


Figura 20 – SensorTag CC2650, sensores e módulo rádio BLE integrados [27]

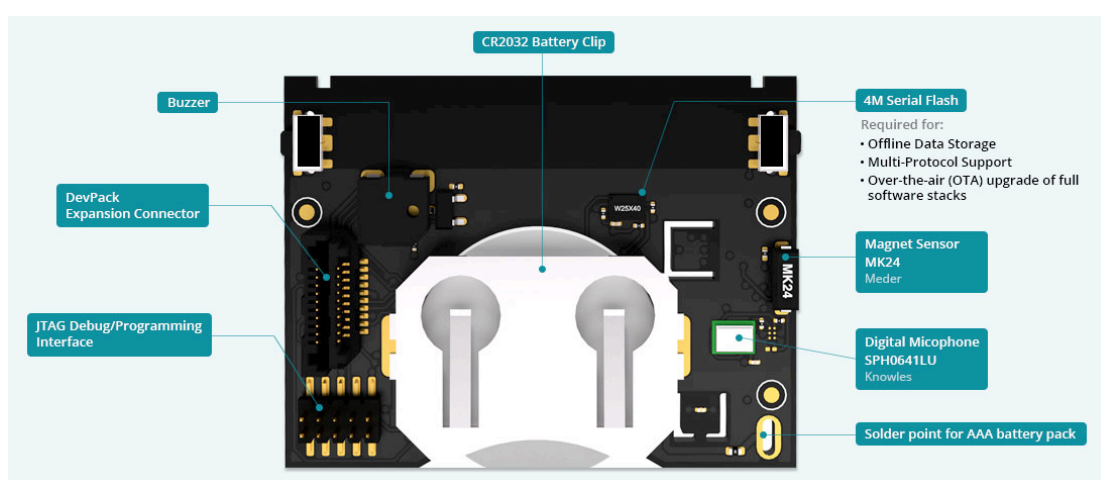


Figura 21 – SensorTag CC2650, sistema de alimentação energética à base de bateria [27]

A única opção que, de certo modo, compete com o equipamento acima exposto, chama-se IMUDUINO. Este equipamento é fabricado por uma *Startup* americana e compete com o SensorTag pela razão de: ter os três sensores de movimento acima referidos, ter um módulo rádio do mesmo protocolo e ser de dimensões mais reduzidas. No entanto, este não possui um sistema de alimentação energética à base de bateria, não possui uma proteção como a do SensorTag e é vendido por mais de quatro vezes o valor do equipamento da Texas Instruments [27], [29].

Além das comparações já feitas, o SensorTag destaca-se do IMUDUINO pela razão do seu fabricante ser reconhecido mundialmente pelos produtos que foi comercializando ao longo dos seus anos de existência [27], [29].

Para a implementação do sistema, foi selecionada a segunda versão do SensorTag (SensorTag CC2650).

A primeira versão considerada para a presente dissertação, foi a CC2541 (primeira versão do SensorTag). Esta versão além de não apresentar valores de consumo energético tão positivos quanto os da versão atual, apresentou, nos testes realizados, uma falha de *firmware*¹⁷ considerada comprometedora para o bom funcionamento do sistema¹⁸ [30].

¹⁷ Conjunto de instruções operacionais programadas diretamente no *Hardware* do equipamento eletrónico.

¹⁸ No capítulo 6. explica-se a falha encontrada e como esta iria afetar o bom funcionamento do sistema.

Com a necessidade de alteração da versão do equipamento SensorTag, o desenvolvimento do projeto foi atrasado. O atraso deveu-se não só à espera da receção do equipamento, como à espera da divulgação de informação, por parte da Texas Instruments, das informações necessárias à correta comunicação BLE entre este e os outros equipamentos do sistema poRsCHE (informação para uma correta configuração das comunicações BLE e para uma correta descodificação das mensagens trocadas).

Na secção 2.1. fez-se referência à utilização de um boné, por parte de um dos projetos estudados, como meio de segurar o sensor de movimento à cabeça do utilizador. Nessa mesma secção também se referiu que, tal equipamento/vestuário, seria uma possível opção para a implementação do presente projeto.

A utilização do boné é uma hipótese considerada, mas não é única. O sistema poRsCHE adota dois equipamentos/vestuários na sua implementação. Para a fixação do SensorTag na cabeça do utilizador, é utilizada uma fita desportiva como ilustrado na figura 22 [31].



Figura 22 – Fita desportiva para segurar o SensorTag à cabeça do utilizador

A colocação da fita desportiva poderá ser feita diretamente na cabeça do utilizador do sistema ou então recorrendo a um boné como é ilustrado na figura 23 [31].



Figura 23 – Fita desportiva colocada sobre um boné

Através da implementação exposta, pretende-se alcançar dois objetivos:

- tornar o sistema o mais versátil possível (com a implementação da fita para fixar o SensorTag, o utilizador pode optar ou não pela utilização do boné (vestuário mais indicado para ambientes exteriores));
- tornar o sistema disponível para vários utilizadores (com a implementação da fita conforme apresentada na figura 22, o sistema poRsCHE permite que o mesmo SensorTag possa facilmente ser utilizado por mais do que um utilizador. Cada utilizador irá ter a sua fita, sendo o SensorTag trocado entre fitas consoante necessidade. Deste modo garante-se uma utilização higiénica do equipamento que deteta os movimentos de comando à condução da cadeira de rodas).

5.2. Seleção da PD com portas *input/output* programáveis

Uma vez selecionada a PD para detetar os movimentos de cabeça do utilizador, é necessário escolher a PD que irá comunicar com a primeira e com o *SmartPhone*.

No presente mercado eletrónico, as PDs com portas *input/output* programáveis, mais exploradas para a implementação de projetos como o presente, são fabricadas pelas marcas: Arduino e RaspberryPi [32-33].

Referentemente às placas desenvolvidas pela marca Arduino, qualquer integração “*plug and play*” de um módulo rádio terá que ser realizada através de uma placa de expansão que será colocada em cima dos pinos (portas *input/output* programáveis) da PD. Atualmente, não existe no mercado eletrónico nenhuma placa de expansão para PDs Arduino, com um módulo de rádio BLE que funcione nos dois modos protocolares.

Por outro lado, as PDs desenvolvidas pela marca RaspberryPi, possuem interfaces *input/output* USB que permitem a conexão de módulos rádio BLE com capacidade de funcionar nos dois modos protocolares. Além de estas PDs possuírem entradas USB que permitem inserir *Pens* USB BLE, também têm capacidades de processamento bastante superiores às PDs desenvolvidas pela Arduino, melhorando desta forma o desempenho do processamento dos dados recebidos pelo SensorTag e conseqüentemente a condução da cadeira de rodas.

Face ao exposto, o presente projeto será implementado com recurso a uma PD RaspberryPi Model B+, no valor de 32,78€. Esta PD embora apresentando um valor em 7€ superior ao da PD mais explorada da Arduino (a Arduino UNO), possibilita, além de um acesso a um módulo rádio BLE, uma capacidade de processamento vinte e cinco vezes superior ao da Arduino UNO [34-35].

Na figura 24 ilustra-se a PD selecionada.



Figura 24 – RaspberryPi Model B+ [35]

Selecionada a PD, é necessário selecionar a *Pen* BLE que irá disponibilizar a esta a capacidade de comunicação com o SensorTag e com o *SmartPhone*. Existem no mercado algumas opções para este equipamento. Todas são de valor aproximado e todas oferecem ao sistema as mesmas funcionalidades, ou seja, a integração de um módulo rádio BLE que funciona em ambos os modos protocolares. Não existindo diferenças, entre as opções existentes no mercado que justifiquem uma crítica detalhada, a escolha recai sobre a *Pen* fabricada pelas ASUS, marca mundialmente conhecida pelos produtos de informática fabricados. A *Pen* em questão tem o nome BT-400 e tem um valor de 14,50€ [36-37].

Na figura 25 ilustra-se a *Pen* selecionada.



Figura 25 – *Pen* ASUS BT-400 [36]

Anteriormente à seleção da PD RaspberryPi Model B+, foi testada a integração do módulo rádio BLE RN4020 da Microchip (módulo rádio que opera nos dois modos protocolares) na PD Arduino UNO. A integração deste módulo foi feita soldando os pinos de transmissão e receção aos pinos UART da PD [34], [38].

Depois de feita a ligação explicada no parágrafo anterior e a correta alimentação do módulo rádio, foram testadas as comunicações entre a PD Arduino UNO e o Sensortag. Embora se tenha conseguido uma correta conexão dos dois dispositivos, SensorTag e Arduino UNO, as comunicações entre estes não obtiveram o sucesso pretendido. Através da versão de *firmware* testada (versão 1.23), não foi possível a descoberta de todos os serviços do SensorTag nem tão pouco a configuração e ativação do serviço Movimento (serviço utilizado pelo sistema poRsCHE para obter os dados provenientes dos sensores de movimento) [26], [34], [38].

Com o intuito de solucionar a lacuna encontrada foram estabelecidos contactos com a Microchip. Após um longo período de troca de informações e consequentes testes ao equipamento, que não alcançaram os resultados pretendidos, a abordagem foi abandonada.

5.3. Implementação do protótipo

Para que o sistema embebido poRsCHE possa ser submetido a testes sem colocar em causa a integridade física do utilizador, é implementado um protótipo, de dimensões 30x25cms, de condução externa ao utilizador. O protótipo a ser implementado terá deslocação idêntica a de uma cadeira de rodas elétrica, sendo propulsionado por duas rodas motoras traseiras.

Para que a implementação do protótipo seja apresentada de forma clara, esta secção irá ser dividida em três subsecções: componentes e estrutura, alimentação energética e ligações eletrónicas.

5.3.1. Componentes e estrutura

A construção do protótipo é feita recorrendo aos seguintes componentes:

- uma placa em PVC;
- duas rodas motoras propulsionadas por dois servos Futaba S3104 [39];
- duas rodas não motoras de rotação livre 360°;
- a PD, com pinos *input/output* programáveis, selecionada na secção anterior (RaspberryPi Model B+) [35];
- um sonar HC-SR04 [40];
- dois suportes de bateria (baterias que irão ser formadas por conjuntos de seis pilhas AA);
- dois reguladores de tensão LM7805C [41];
- uma *breadboard* com 400 pontos de conexão;
- uma *breadboard* com 170 pontos de conexão;
- três LEDs;
- um conjunto de 25 *jumper wires*;
- seis resistências (1 x 330Ω; 1 x 4,7KΩ; 1x 10KΩ; 3 x 330Ω);
- um cabo de alimentação com terminal MicroUSB.

Através da figura 26 ilustra-se a base utilizada como base para o protótipo (placa em PVC, duas rodas motoras, duas rodas não motoras de rotação livre e dois suportes de bateria).



Figura 26 – Base para o protótipo poRsCHE

Através das figuras 27 e 28 ilustra-se a integração dos restantes componentes na base apresentada na figura 26.

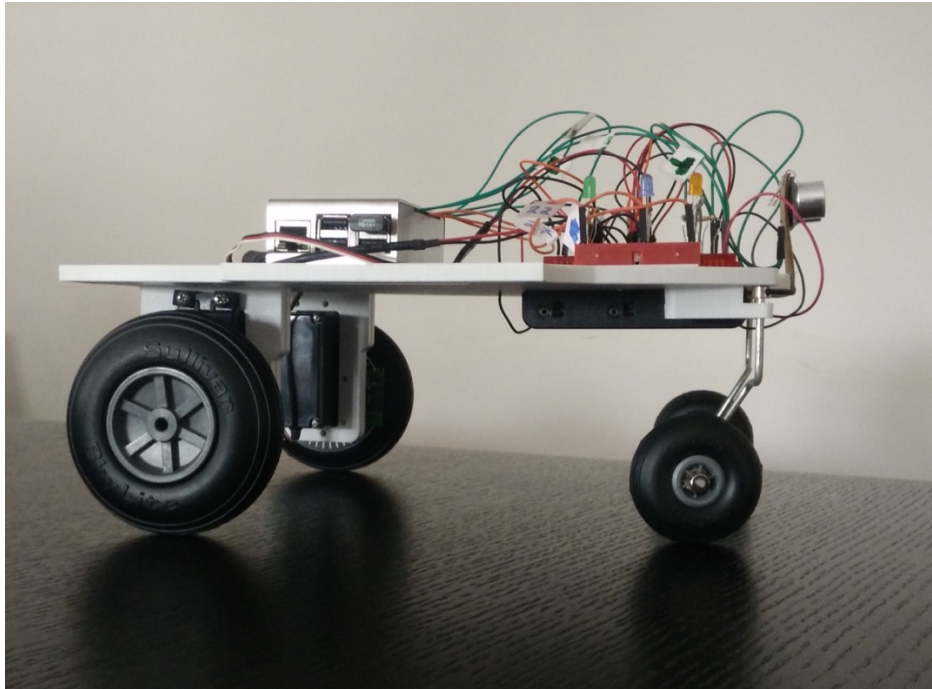


Figura 27 – Protótipo poRsCHE, visão lateral

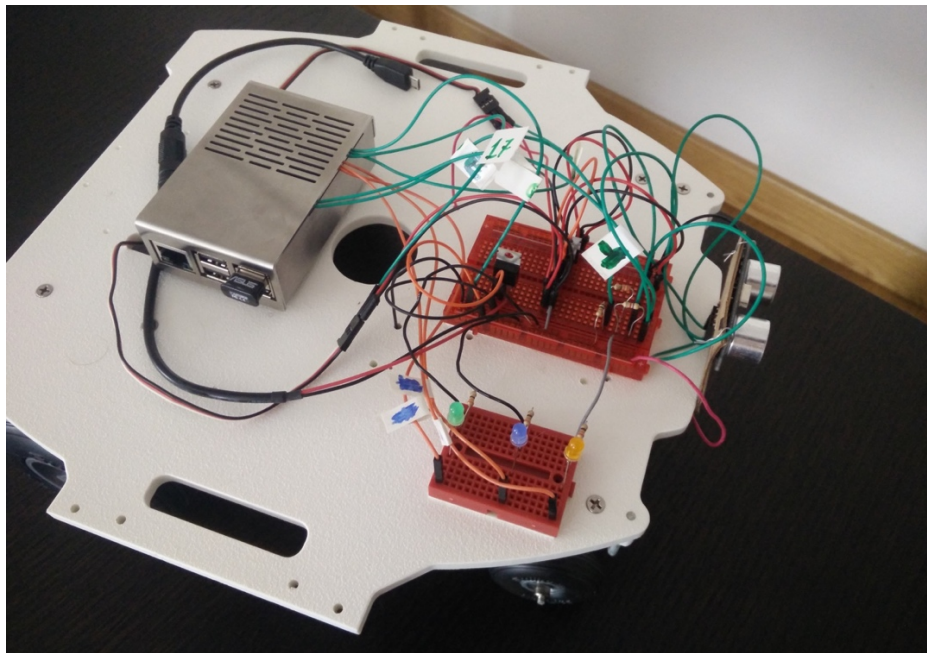


Figura 28 – Protótipo poRsCHE, visão superior

5.3.2. Alimentação energética

A alimentação energética do protótipo é composta por duas baterias independentes:

- para alimentar os servos, o sonar e a interface gráfica constituída por três LEDs é utilizada uma bateria composta por seis pilhas alcalinas AA, cada uma com 2500mAh e 1,2V, o que resulta numa bateria de 2500mAh e 7,2V. Bateria ilustrada através da figura 29;



Figura 29 – Bateria para alimentar servos e sonar

- para alimentar a PD RaspberryPi é utilizada uma bateria composta por seis pilhas alcalinas AA, cada uma com 1000mAh e 1,2V, o que resulta numa bateria de 1000mAh e 7,2V. Bateria ilustrada através da figura 30;



Figura 30 – Bateria para alimentar RaspberryPi

Para se obter uma tensão estável de 5V é utilizada para cada bateria um regulador de tensão LM7805C da Texas Instruments. A limitação da tensão a 5V é feita dadas as características de alimentação dos equipamentos utilizados [39-40] [42].

Para alimentar a PD RaspberryPi, é indicada a utilização de uma fonte de alimentação com corrente estável a 5V e amperagem entre 700mAh e 1000mAh, consoante os periféricos utilizados [42].

Para alimentar os servos Futaba S3104 e o sonar HC-SR04, é indicada a utilização de uma fonte de alimentação com tensão estável a 5V. Tendo em conta a energia que irá ser despendida para propulsionar o protótipo, a bateria utilizada para alimentar os servos é composta por pilhas com amperagem igual a 2500mAh [39-40].

5.3.3. Ligações eletrônicas

Através da figura 31 ilustram-se as ligações eletrônicas estabelecidas para conectar os equipamentos do protótipo poRsCHE.

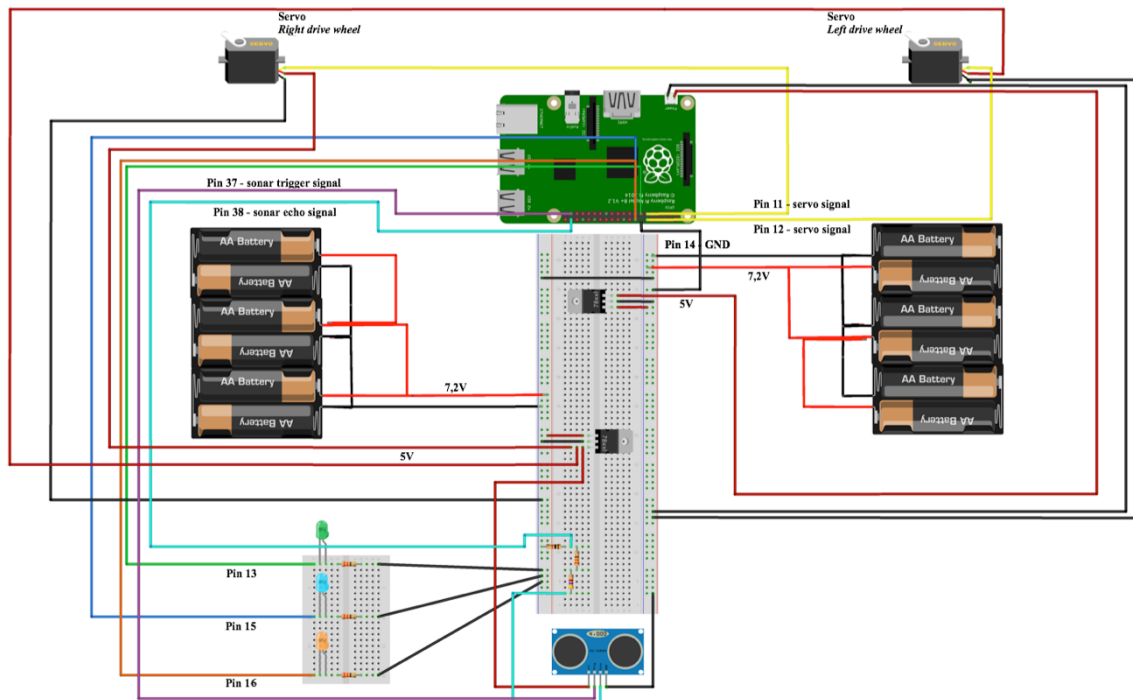


Figura 31 – Circuito eletrônico protótipo poRsCHE

A vermelho-vivo são representadas as ligações de corrente elétrica de 7,2V, a preto as de 0V (GND) e a vermelho escuro as de 5V.

A amarelo são representadas as ligações entre a PD RaspberryPi e os servos Futaba. No pino 11 da RaspberryPi é gerado o sinal PWM de comando do servo que propulsiona a roda motora direita. No pino 12 é gerado o sinal PWM de comando do servo que propulsiona a roda motora esquerda.

Os sinais PWM gerados para comandar os servos Futaba S3104 têm frequência de 50Hz (período = 20000uS) e *duty-cycle* consoante ordem de comando:

- para rotação no sentido horário, o *duty-cycle* é inferior a 7,005% (<1401uS). Quanto menor, maior a velocidade de rotação;

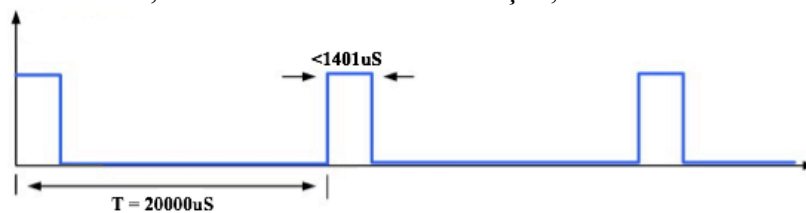


Figura 32 – Sinal PWM-rotação do servo no sentido horário

- para rotação no sentido anti-horário, o *duty-cycle* é superior a 7,005% (>1401uS). Quanto maior, maior a velocidade de rotação;

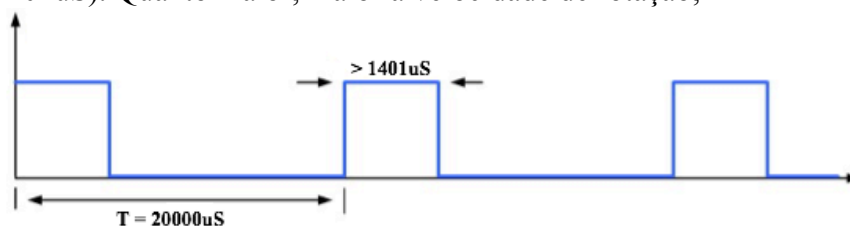


Figura 33 – Sinal PWM-rotação do servo no sentido anti-horário

- para interromper a rotação, o *duty-cycle* é igual a 7,005% (1401uS).

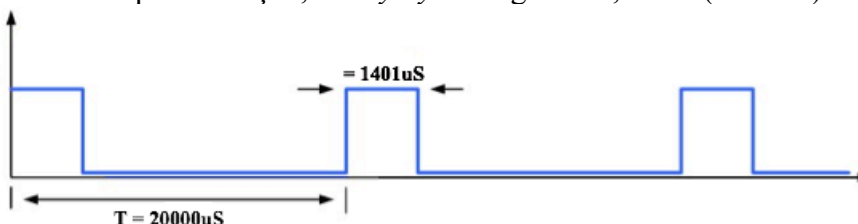


Figura 34 – Sinal PWM-imobilização do servo

A verde, a azul e a laranja são representadas as ligações entre a PD RaspberryPi e os LEDs que formam a interface gráfica do sistema. Para limitar a corrente, de modo a não queimar as portas *input/output* da RaspberryPi, são utilizadas três resistências de 330Ω , uma por cada ligação RaspberryPi – LED. No pino 13 da RaspberryPi é gerada uma tensão de 5V quando o sistema se encontra no estado *Standby* (liberdade de movimentos). No pino 15 é gerada uma tensão de 5V quando o sistema, estando no estado *Driving*, está preparado para receber um novo comando de condução. No pino 16 é gerada uma tensão de 5V quando o sistema se encontra no estado *Driving* (os movimentos de cabeça do utilizador são interpretados como movimentos de comando de condução).

A violeta e a azul-marinho são representadas as ligações entre a PD RaspberryPi e o sonar HC-SR04. No pino 37 da RaspberryPi é gerado o sinal PWM que ativa o funcionamento do sonar (sinal com impulsos de 5V de duração igual a 10uS). No pino 38 é recebido o sinal PWM que irá ser utilizado para calcular a distância a que o obstáculo detetado se encontra (sinal com impulso de 5V de duração igual ao intervalo de tempo entre a emissão da onda sonora gerada e a receção do eco da mesma). Para o sinal gerado pelo sonar (*echo signal*) baixar de 5V para 3,3V, de modo a não danificar a porta GPIO da RaspberryPi, é utilizado um divisor de tensão com recurso a três resistências [40], [43]. O divisor de tensão é representado através da figura 35.

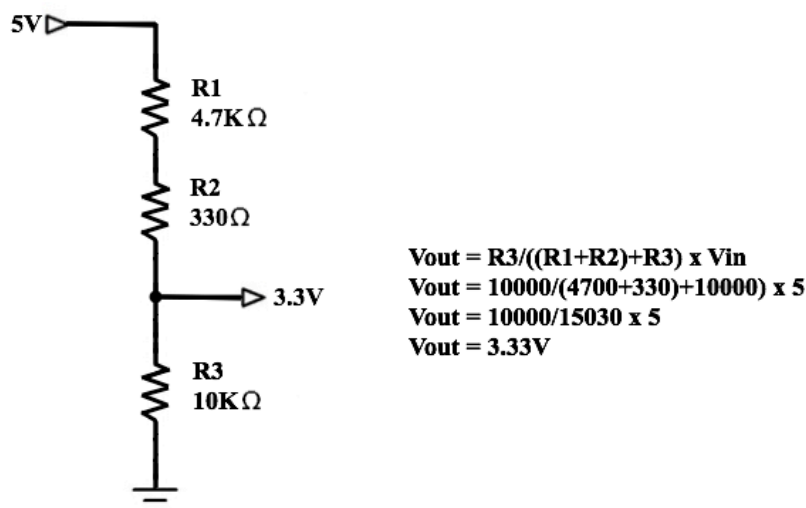


Figura 35 – Divisor de tensão 5-3,3V

Através do *echo signal*, obtém-se o valor de tempo decorrido entre a emissão da onda sonora e a receção do seu eco, valor que vai permitir o cálculo da distância ao obstáculo [40]. Em seguida apresenta-se a dedução da fórmula de cálculo da distância:

$$distância = velocidade \times \Delta t$$

$$distância_{obstáculo} = velocidade_{som}^{ar} \times \frac{\Delta t_{medido\ pelo\ sonar}}{2}$$

À constante $velocidade_{som}^{ar}$, corresponde o valor 340,3 m/s, velocidade de propagação das ondas sonoras no ar.

À variável $\Delta t_{medido\ pelo\ sonar}$, corresponde o tempo medido pelo sonar entre a emissão da onda sonora e a receção do seu eco. Para cálculo da distância ao obstáculo, este valor é dividido por dois. O tempo decorrido entre a emissão da onda sonora e a sua colisão com o obstáculo, corresponde a metade do tempo cronometrado entre a emissão da onda sonora e a receção do seu eco.

Substituindo a variável $velocidade_{som}^{ar}$ pelo seu valor, chega-se à fórmula final que permite o cálculo da distância ao obstáculo detetado:

$$distância_{obstáculo} = \frac{velocidade_{som}^{ar} \times \Delta t_{medido\ pelo\ sonar}}{2}$$

$$distância_{obstáculo} = \frac{340,3 \times \Delta t_{medido\ pelo\ sonar}}{2}$$

$$distância_{obstáculo} = 170,15 \times \Delta t_{medido\ pelo\ sonar} (m).$$

5.4. Estudo dos sensores de movimento do SensorTag

O presente projeto tem como ideia inicial a utilização de um acelerómetro para a deteção dos movimentos efetuados pela cabeça do utilizador.

A PD selecionada, para deteção de movimento, tem incorporado um sistema mecânico microeletrónico que possui três sensores de movimento: um acelerómetro, um magnetómetro e um giroscópio.

Uma vez definidos os movimentos que irão servir de comando à condução da cadeira de rodas e existindo, na PD, mais dois sensores para além do inicialmente pensado para o projeto, estabelece-se como nova etapa, o estudo da deteção dos movimentos, que irão servir de comando de controlo à condução da cadeira de rodas, por parte dos três diferentes sensores.

Antes de se iniciar o estudo dos sensores e consequente seleção do sensor para o sistema poRsCHE, apresenta-se uma pequena explicação do funcionamento de cada um.

5.4.1. Funcionamento do acelerómetro MEMS

O princípio de funcionamento de um acelerómetro MEMS passa pela medição, através de pentes (identificados na figura 35 como “Fixed plates”), do deslocamento de um corpo (massa) preso à superfície de silicone do sistema MEMS [44-46]. Através da figura 36 ilustra-se o princípio descrito.

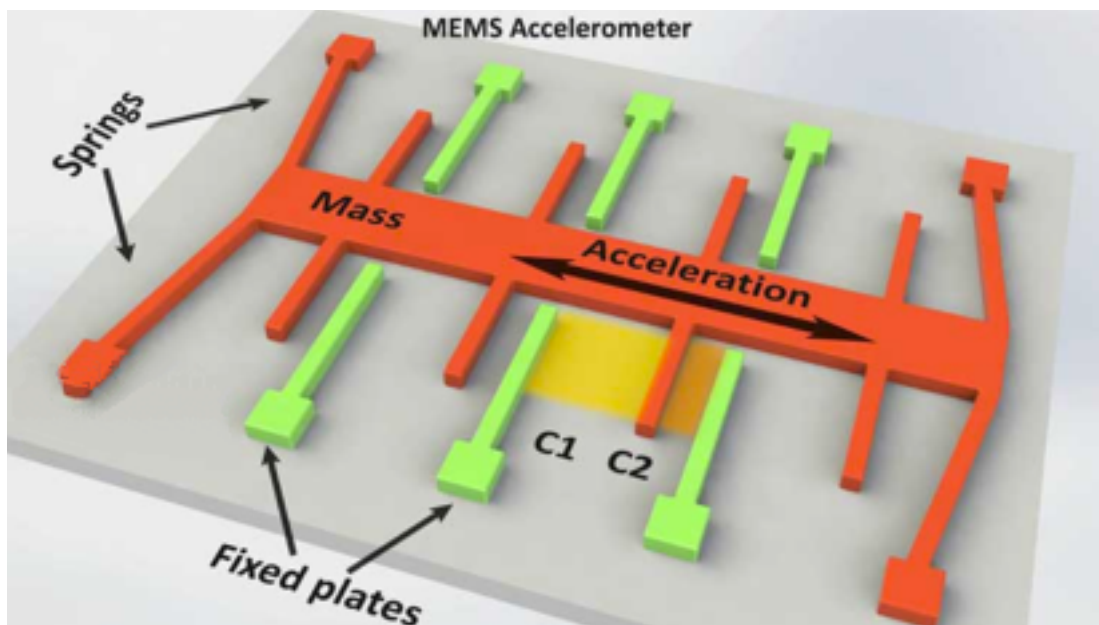


Figura 36 – Estrutura de um Acelerómetro MEMS [47] (imagem editada)

Segundo a segunda lei de Newton ($F = m \times a$), ao deslocar-se o SensorTag, o corpo presente no *chip* irá sofrer uma força que irá ser medida pelo sistema. Dividindo a força medida pela massa do corpo, obtém-se o valor de aceleração correspondente ao deslocamento efetuado pelo SensorTag [44-46].

Embora na figura 35 apenas esteja representada a medição de força segundo um eixo, o sensor mede as forças que atuam sobre a sua massa segundo os três eixos de referência (X, Y e Z) [44-46].

Os valores de movimento medidos pelo acelerómetro são apresentados através de múltiplos de 1G.

A força Gravitacional, força exercida pelo Planeta Terra a qualquer corpo (massa) presente na sua superfície, tem o valor de $9,8 \text{ m/s}^2$. A unidade de aceleração não SI¹⁹, G, representa o valor de aceleração que o Planeta Terra exerce sobre uma massa à sua superfície [44-46].

Através deste sensor consegue-se medir qualquer tipo de movimento à exceção da queda livre (movimento não considerado relevante para o sistema). A impossibilidade de medição deste movimento deve-se ao facto da única força que atua sobre a massa do *chip* (força gravitacional), atuar sobre os três eixos de referência ao mesmo tempo. Ou seja, não existe nenhum movimento relativo entre a massa e o referencial (massa do Planeta Terra) [44-46].

A medição do movimento por parte deste sensor está dependente de um referencial, a força gravitacional [44-46].

¹⁹ Unidade de medida de aceleração que não pertence ao Sistema Internacional de Unidades.

5.4.2. Funcionamento do magnetómetro MEMS

O princípio de funcionamento deste sensor passa pela medição da orientação e magnitude do campo magnético, envolvente a este, segundo o efeito *Hall*. Através das figuras 37 e 38 ilustra-se como ocorre tal medição [44-46].

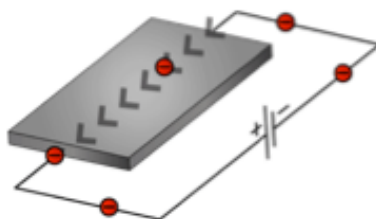


Figura 37 – Demonstração do efeito *Hall* para medição do campo magnético I [47] (imagem editada)

Ao montar-se um circuito elétrico como ilustrado na figura 37, os eletrões que se deslocam no circuito seguem um percurso direto entre as extremidades da placa condutora [44-46].

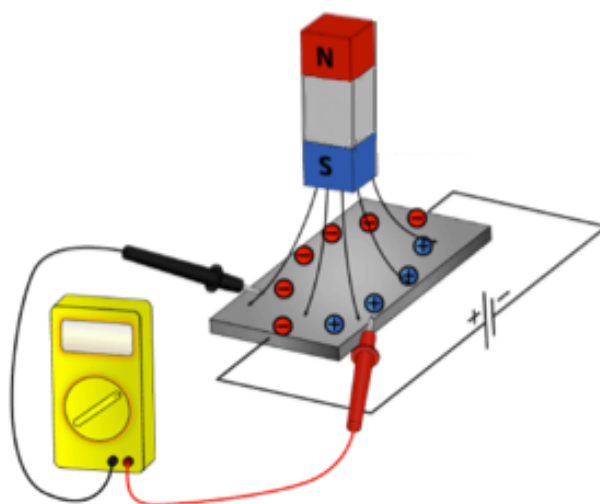


Figura 38 – Demonstração do efeito Hall para medição do campo magnético II [47] (imagem editada)

Por outro lado, ao aproximarmos da placa condutora um campo magnético, as cargas elétricas irão posicionar-se de forma a originar um diferencial de tensão. Como ilustrado na figura 38, os eletrões defletem para uma extremidade da placa e os prótons para a extremidade oposta [44-46].

Através de uma correlação entre o diferencial de tensão medido no magnetómetro MEMS e valores conhecidos à priori, através de estudos, pode aferir-se qual a polaridade do campo magnético presente e qual a sua magnitude. A unidade de medida do magnetómetro MEMS é o μT (micro Tesla) [44-46].

Além da medição do movimento por parte deste sensor estar dependente de um referencial, o campo magnético, esta medição encontra-se também dependente do ambiente envolvente. O referencial para a medição do movimento, resulta não só do campo magnético da Terra, como dos possíveis objetos magnetizados que se encontrem no raio de ação do sensor [44-46].

5.4.3. Funcionamento do giroscópio MEMS

O princípio de funcionamento do giroscópio MEMS passa pela medição de uma disposição adotada pela massa presente no *chip* aquando de uma rotação. Quando é aplicada uma rotação ao SensorTag, duas forças são exercidas sobre a massa do giroscópio MEMS. Uma das forças é a velocidade do movimento, representada na figura 39 pelo vetor de cor vermelha. A outra força, é o efeito de Coriolis²⁰, representada na figura 39 pelo vetor de cor azul. Como resultado das duas forças referidas, a massa presente no *chip* posiciona-se num ponto 2D que corresponde a um determinado valor de velocidade angular. A unidade de medida da velocidade angular é o rad/s, podendo também ser apresentada em graus/s [44-46].

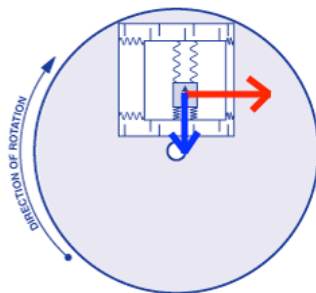


Figura 39 – Medição de rotação por parte de um Giroscópio MEMS [48] (imagem editada)

A estrutura do giroscópio MEMS é apresentada na figura 40. A massa representada a laranja, que se encontra em constante oscilação, ao sofrer uma rotação sobre o seu plano, irá posicionar-se num determinado local dentro da área compreendida pelo retângulo vermelho. A posição adotada pela massa corresponde a um valor de velocidade angular [44-46].

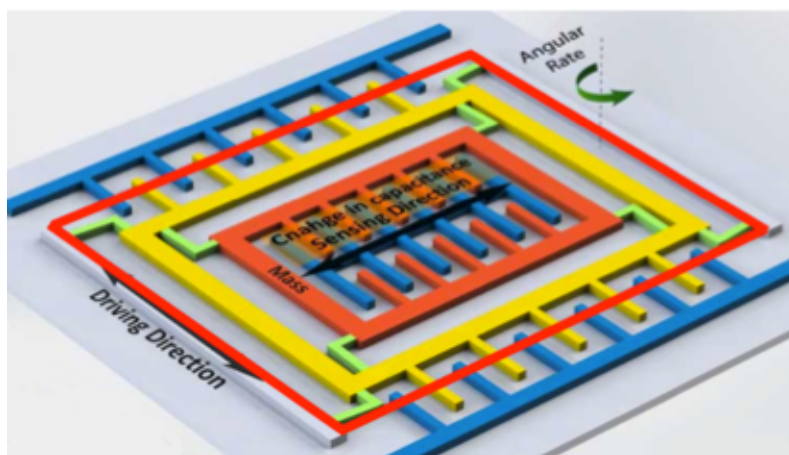


Figura 40 – Estrutura de um Giroscópio MEMS [47] (imagem editada)

Ao contrário dos dois sensores anteriormente apresentados, acelerómetro e magnetómetro, que medem o movimento segundo um referencial, gravidade terrestre e campo magnético respetivamente, o giroscópio MEMS mede o movimento sem estar dependente de um referencial externo. Este sensor mede a sua própria rotação [44-46].

²⁰ Força resultante da rotação do plano de referência de um objeto em constante oscilação.

5.4.4. Estudo do acelerómetro MEMS

Nesta secção é estudado o sensor inicialmente pensado para o projeto. O estudo é feito à resposta de medição que o sensor apresenta aos movimentos definidos para comandar a cadeira de rodas.

Uma vez colocado o SensorTag na cabeça do utilizador, como ilustrado na figura 41 (boné colocado na cabeça com a pala para a frente), os movimentos de comando são detetados segundo os eixos referenciais apresentados na figura 42.

O envio dos valores de aceleração é feito em intervalos de tempo de 100ms, o que corresponde a uma receção de 10 amostras de valor por segundo.



Figura 41 – Colocação do SensorTag na cabeça do utilizador através de um boné

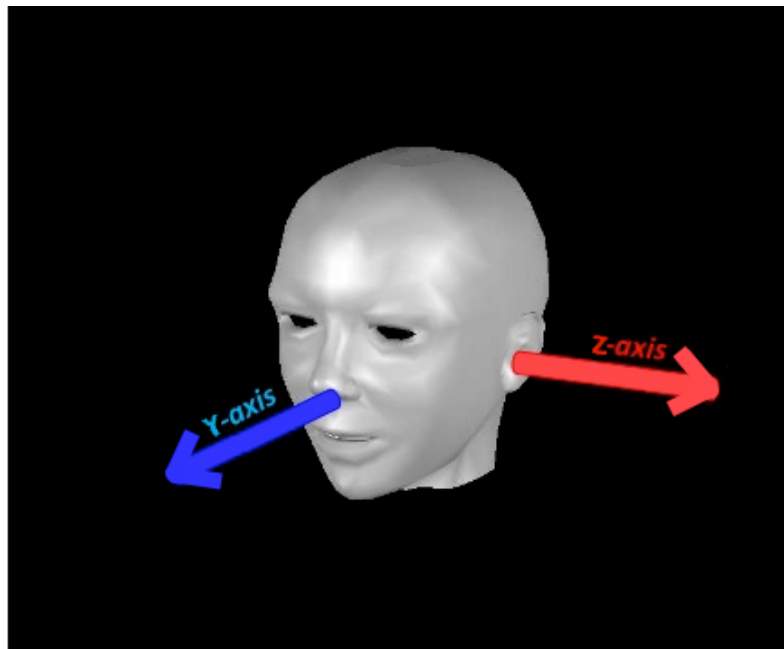


Figura 42 – Eixos referenciais do Acelerómetro do SensorTag

Efetuando o movimento de flexão, o sensor mede um diferencial de aceleração, segundo o eixo Y, representado graficamente por uma parábola com a concavidade para cima. A medição de um movimento deste tipo é apresentada no gráfico 5.

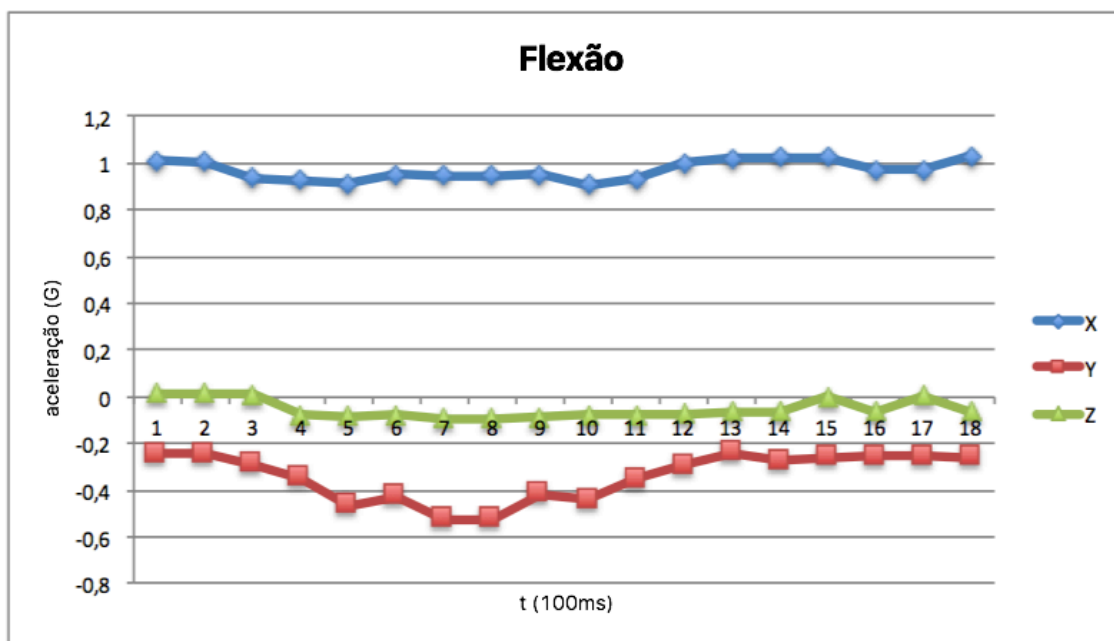


Gráfico 5 – Acelerômetro, Movimento de flexão

Efetuando o movimento de extensão, o sensor mede um diferencial de aceleração, segundo o eixo Y, representado graficamente por uma parábola com a concavidade para baixo. A medição de um movimento deste tipo é apresentada no gráfico 6.

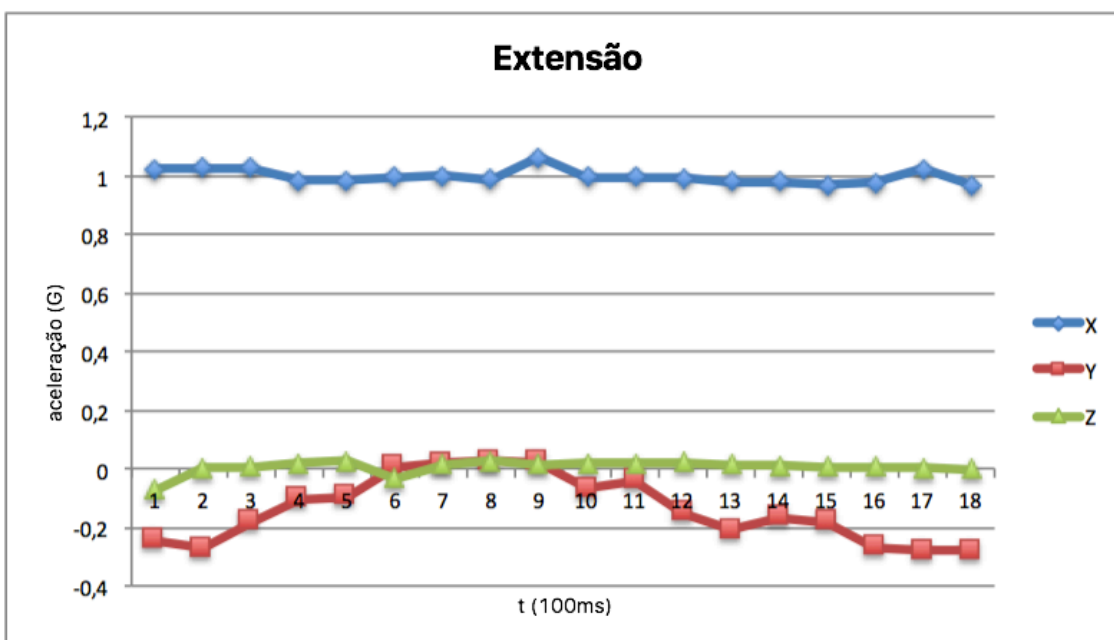


Gráfico 6 – Acelerômetro, Movimento de extensão

Efetando o movimento de inclinação para o lado direito, o sensor mede um diferencial de aceleração, segundo o eixo Z, representado graficamente por uma parábola com a concavidade para baixo. A medição de um movimento deste tipo é apresentada no gráfico 7.

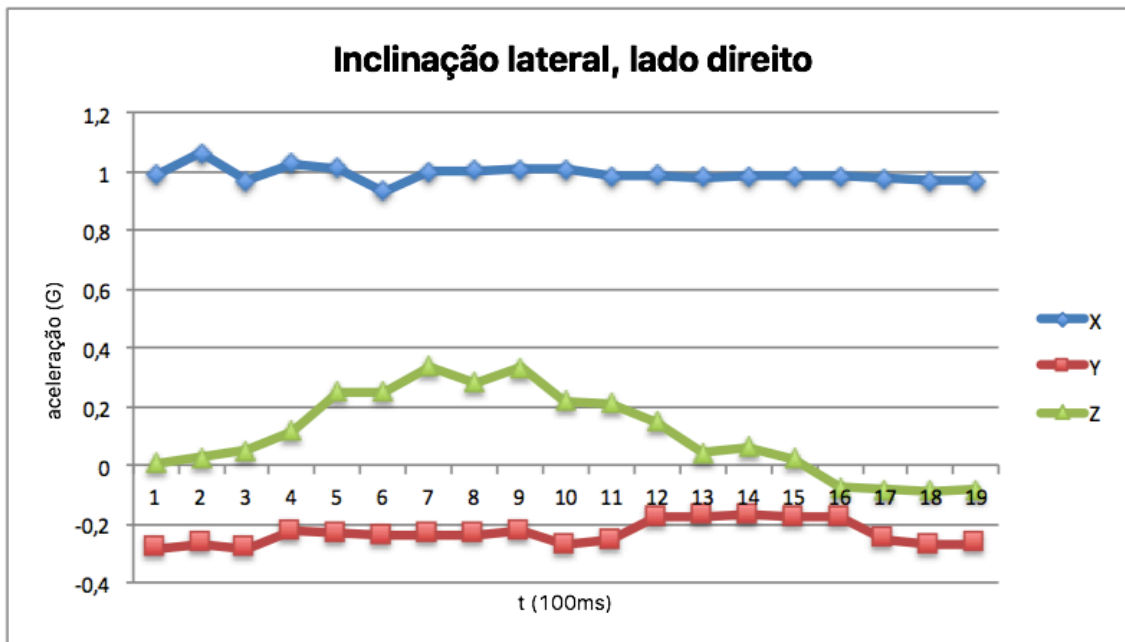


Gráfico 7 – Acelerómetro, Movimento de inclinação para o lado direito

Efetando o movimento de inclinação para o lado esquerdo, o sensor mede um diferencial de aceleração, segundo o eixo Z, representado graficamente por uma parábola com a concavidade para cima. A medição de um movimento deste tipo é apresentada no gráfico 8.

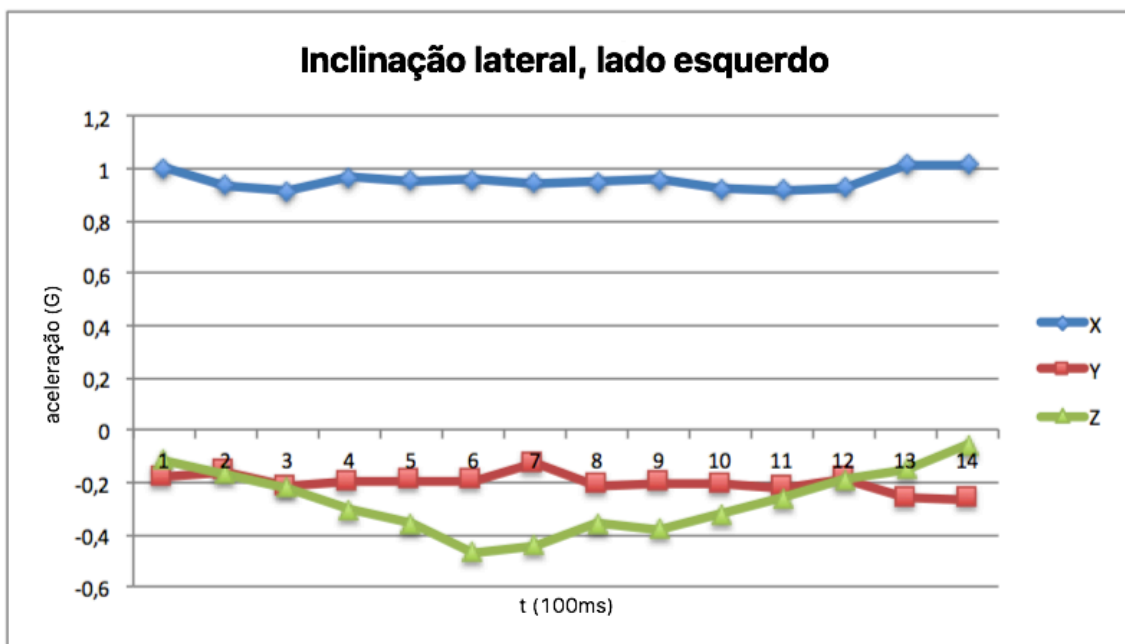


Gráfico 8 – Acelerómetro, Movimento de inclinação para o lado esquerdo

Através dos gráficos apresentados anteriormente, consegue-se identificar com clareza a medição dos movimentos segundo os eixos de referência do acelerómetro. No entanto, identifica-se também que, em termos de análise gráfica, o diferencial de aceleração entre o instante de tempo em que não existe movimento e o instante de tempo em que o movimento efetuado se traduz numa maior aceleração, é baixo. Os diferenciais identificados correspondem a uma ordem de grandeza de cerca de 0,4 da unidade de medida.

A unidade de medida utilizada pelo SensorTag para apresentar os valores de aceleração é o G, unidade de medida não SI.

Este diferencial de aceleração de baixo valor, é facilmente compreendido. Qualquer pessoa, ao movimentar a cabeça, não aplica uma grande aceleração ao movimento efetuado.

Os movimentos ilustrados graficamente apresentam valores de aceleração na ordem de $3,92 \text{ m/s}^2$.

5.4.5. Estudo do magnetómetro MEMS

Nesta secção é estudado o segundo sensor do SensorTag apresentado neste capítulo. O estudo é feito à resposta de medição que o sensor apresenta aos movimentos definidos para comandar a cadeira de rodas.

O envio dos valores de magnitude é feito em intervalos de tempo de 100ms, o que corresponde a uma receção de 10 amostras de valor por segundo.

Embora o SensorTag dê informação sobre a medição do campo magnético segundo um referencial de três eixos, os movimentos medidos não apresentam uma correspondência clara entre o movimento realizado pela cabeça do utilizador e uma alteração de valores num eixo em particular.

Através dos gráficos 9, 10, 11 e 12, apresenta-se a medição dos movimentos de comando do sistema. A apresentação gráfica dos movimentos segue a seguinte ordem de movimentos: flexão, extensão, inclinação lateral (lado direito) e inclinação lateral (lado esquerdo).

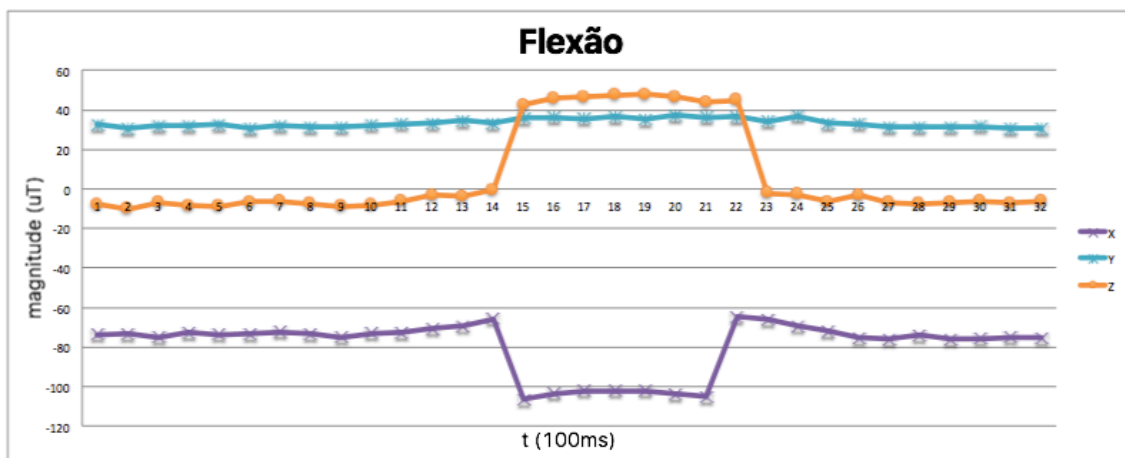


Gráfico 9 – Magnetómetro, Movimento de flexão

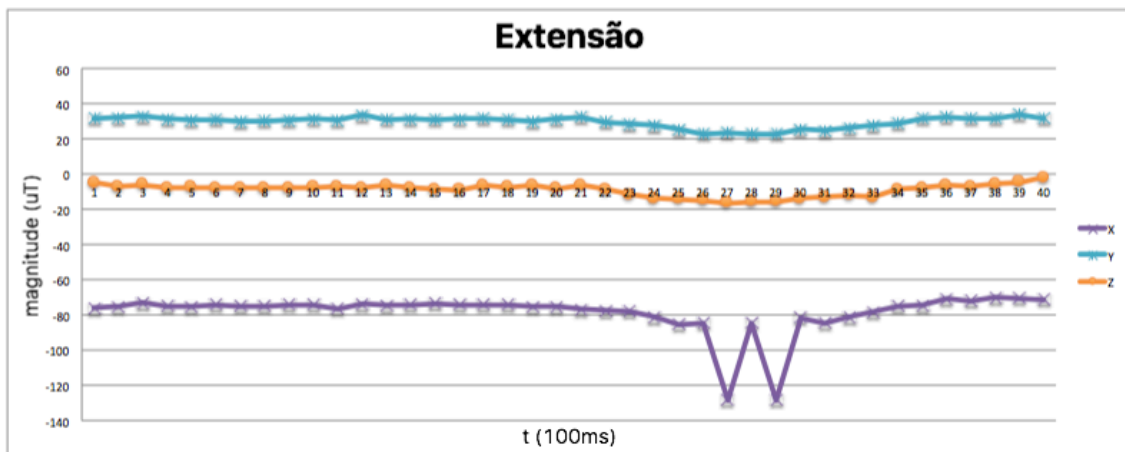


Gráfico 10 – Magnetómetro, Movimento de extensão

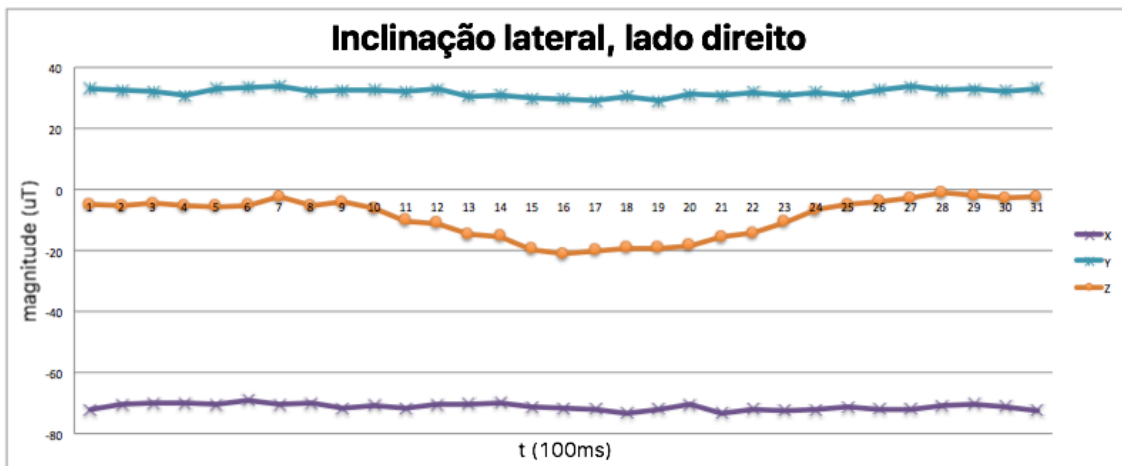


Gráfico 11 – Magnetómetro, Movimento de inclinação para o lado direito

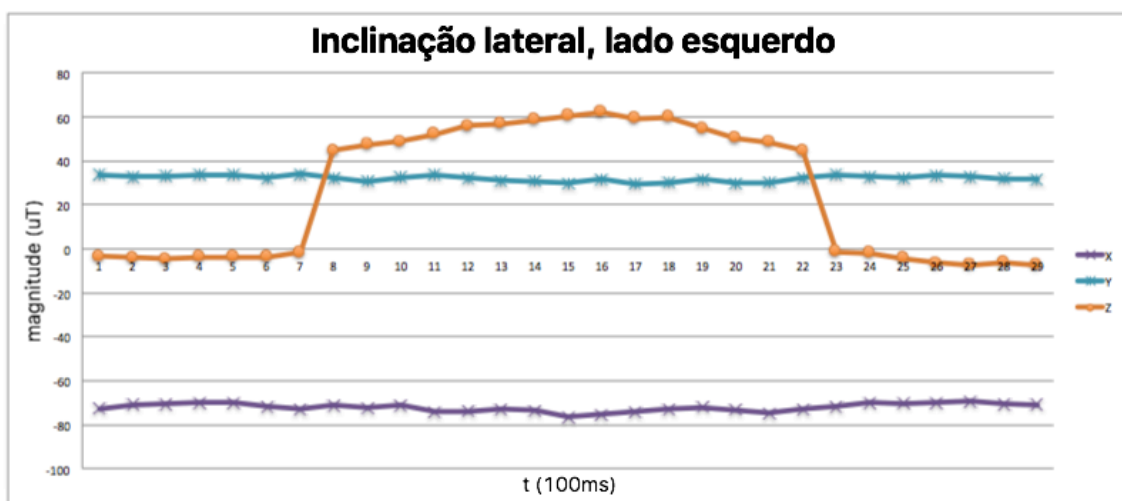


Gráfico 12 – Magnetómetro, Movimento de inclinação para o lado esquerdo

Além das medições de movimento apresentadas não terem uma correspondência clara entre o movimento efetuado e a alteração de valores num determinado eixo, quando realizadas outras medições aos mesmos tipos de movimento, estas não apresentam alterações de valor com os mesmos padrões acima apresentados. Tal alteração de medições deverá estar associada ao facto deste tipo de medição ser influenciado pelo meio envolvente. Qualquer alteração ao meio onde o sensor atue, ou seja, qualquer alteração aos componentes magnetizados que estejam no raio de ação do sensor, irá influenciar qualquer medição que este efetue.

5.4.6. Estudo do giroscópio MEMS

Nesta secção é estudado o último sensor do SensorTag apresentado neste capítulo. O estudo é feito à resposta de medição que o sensor apresenta aos movimentos definidos para comandar a cadeira de rodas.

O envio dos valores de rotação é feito em intervalos de tempo de 100ms, o que corresponde a uma receção de 10 amostras de valor por segundo.

Uma vez colocado o SensorTag na cabeça do utilizador, como ilustrado na figura 43 (boné colocado na cabeça com a pala para a frente), os movimentos de comando são detetados segundo os eixos referenciais apresentados na figura 44.



Figura 43 – Colocação do SensorTag na cabeça do utilizador através de um boné

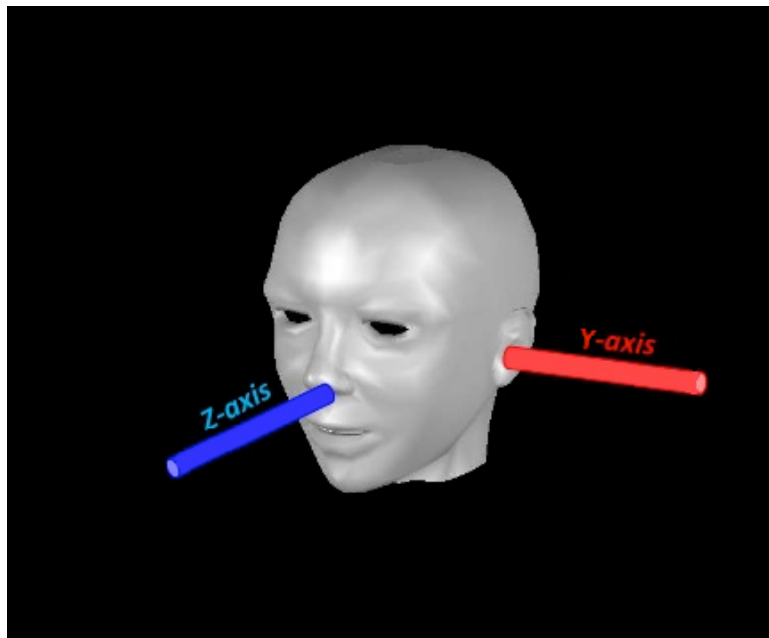


Figura 44 – Eixos referenciais do Giroscópio do SensorTag

Efetuada o movimento de flexão, o sensor mede uma velocidade angular, segundo o eixo Z, representada graficamente por uma parábola com a concavidade para cima, seguida de outra com a concavidade para baixo. A medição de um movimento deste tipo é apresentada no gráfico 13.

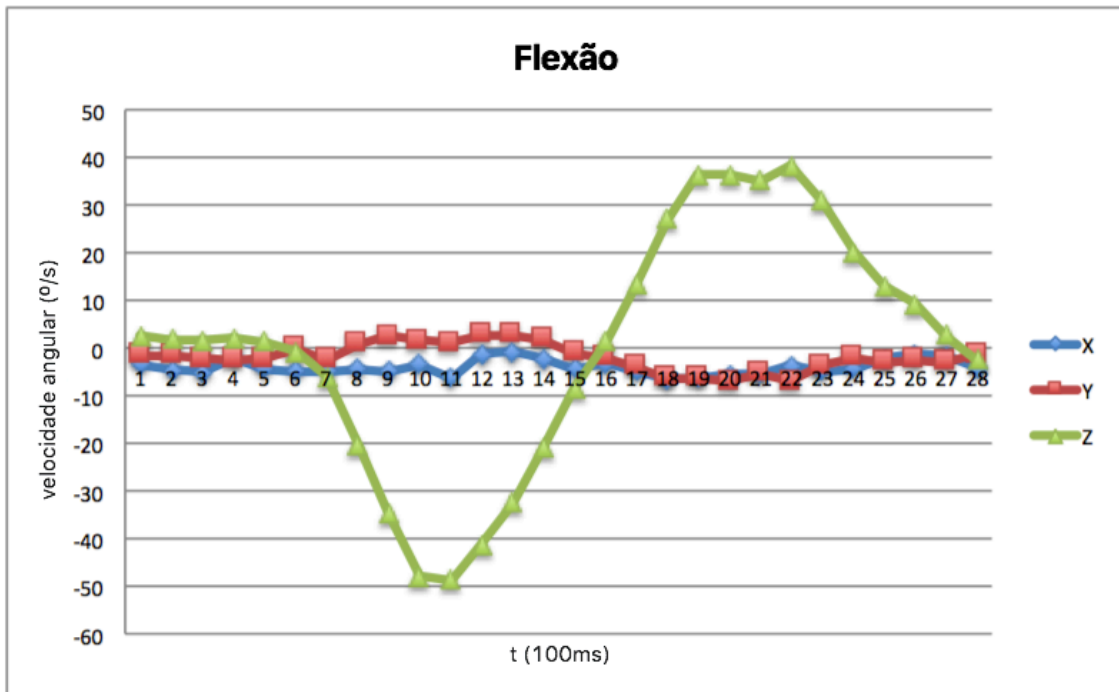


Gráfico 13 – Giroscópio, Movimento de flexão

Efetuada o movimento de extensão, o sensor mede uma velocidade angular, segundo o eixo Z, representada graficamente por uma parábola com a concavidade para baixo, seguida de outra com a concavidade para cima. A medição de um movimento deste tipo é apresentada no gráfico 14.

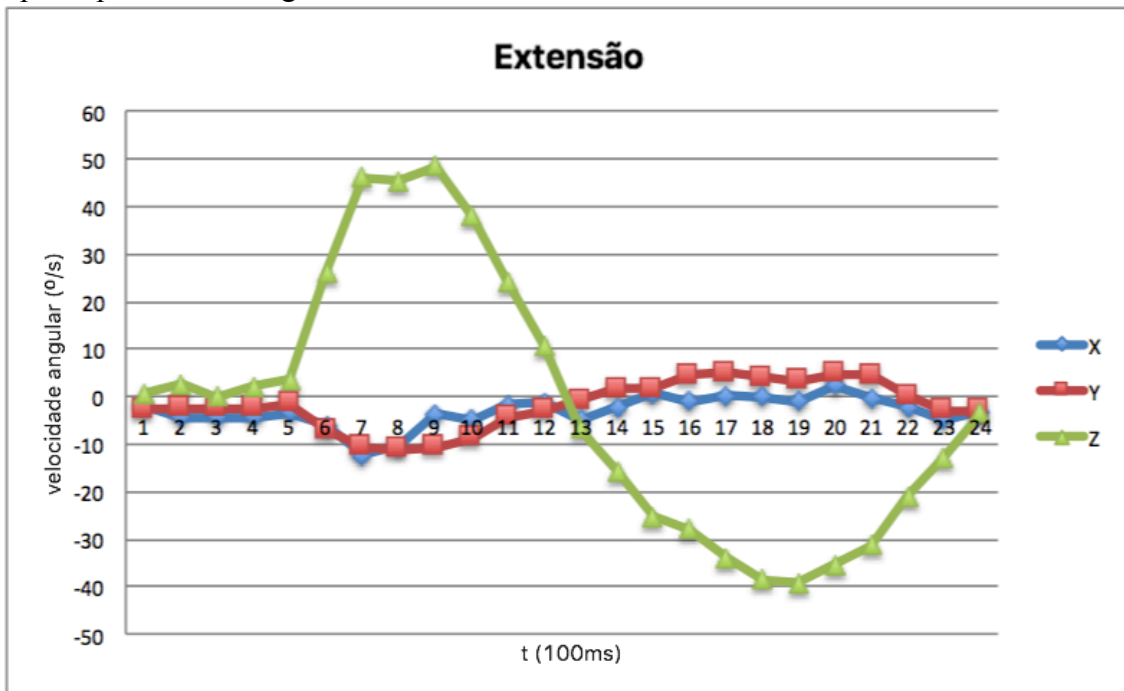


Gráfico 14 – Giroscópio, Movimento de extensão

Efetutando o movimento de inclinação para o lado direito, o sensor mede uma velocidade angular, segundo o eixo Y, representada graficamente por uma parábola com a concavidade para baixo, seguida de outra com a concavidade para cima. A medição de um movimento deste tipo é apresentada no gráfico 15.

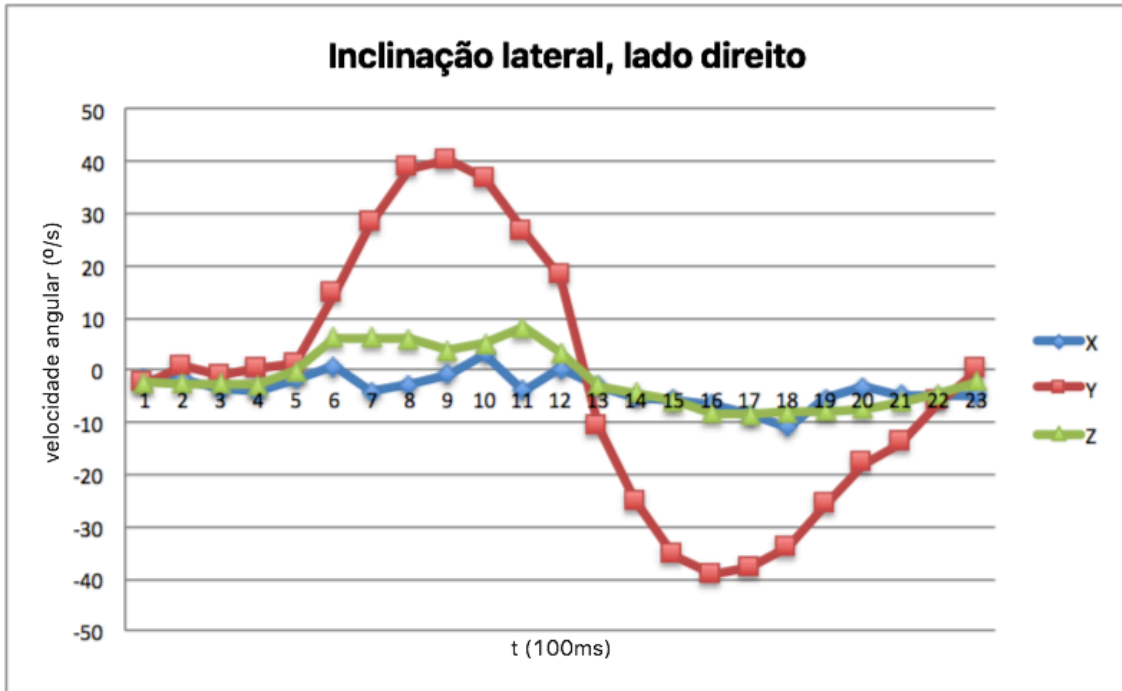


Gráfico 15 – Giroscópio, Movimento de inclinação para o lado direito

Efetutando o movimento de inclinação para o lado esquerdo, o sensor mede uma velocidade angular, segundo o eixo Y, representada graficamente por uma parábola com a concavidade para cima, seguida de outra com a concavidade para baixo. A medição de um movimento deste tipo é apresentada no gráfico 16.

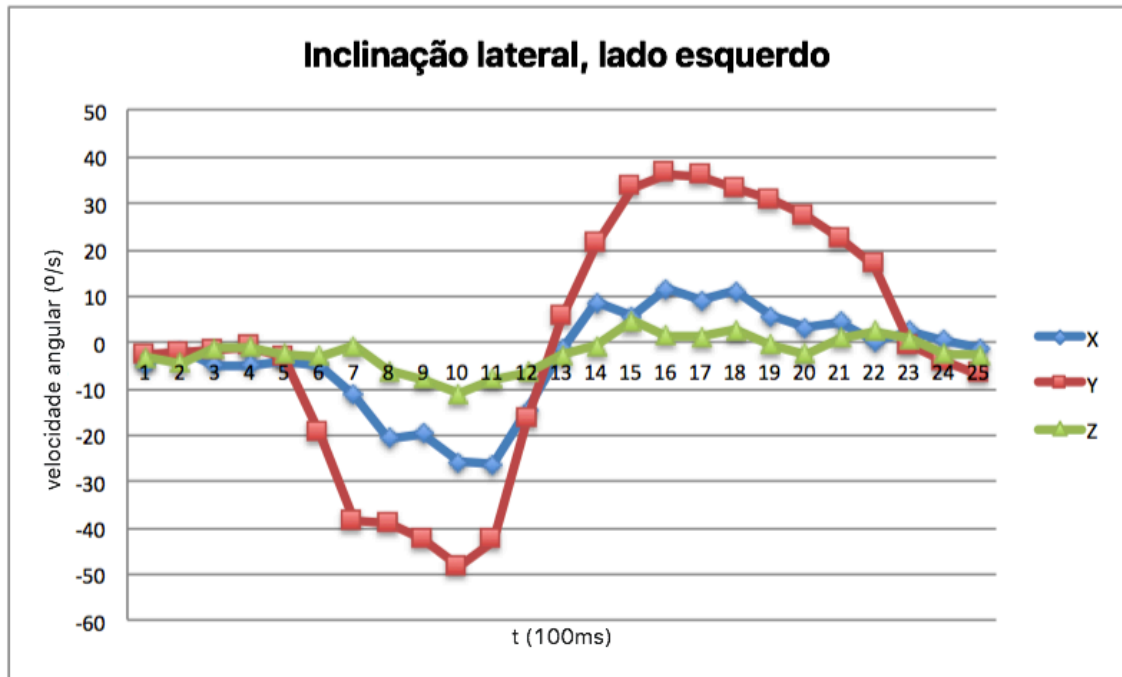


Gráfico 16 – Giroscópio, Movimento de inclinação para o lado esquerdo

Através dos gráficos apresentados anteriormente, consegue-se identificar com clareza a medição dos movimentos segundo os eixos de referência do giroscópio.

Com este sensor, além de se conseguir identificar uma alteração de valores correspondente ao início do movimento efetuado, alteração da posição da cabeça face à posição vertical, também se consegue identificar uma alteração de valores correspondente ao término do movimento de comando, ou seja, ao retorno da posição da cabeça à posição vertical.

As duas alterações da posição da cabeça do utilizador são representadas graficamente pelas parábolas descritas nas duas páginas anteriores. A título de exemplo: no gráfico 13, a parábola com concavidade para cima, representa a inclinação da cabeça do utilizador para a frente; a parábola com concavidade para baixo, representa o retorno da posição da cabeça à posição vertical.

Através da análise aos gráficos do giroscópio, consegue-se obter outro dado importante para a implementação do sistema poRsCHE. O intervalo de tempo que ocorre entre o início das primeiras parábolas e o fim das segundas, corresponde a aproximadamente 2 segundos (20 amostras de valor). Com este dado pode-se afirmar que o tempo que uma pessoa demora a retornar a posição da cabeça à posição vertical, depois de a ter inclinado para a frente/trás, é aproximadamente 2 segundos; valor que será tido em conta para aguardar a confirmação dos movimentos de comando: flexão e extensão²¹. Uma vez que estes valores surgem de testes feitos com uma pessoa sem limitações físicas, esta avaliação irá ser repetida na validação do sistema (testes realizados pelo público alvo).

²¹ Confirmação feita através do retornar da posição da cabeça à posição vertical.

5.4.7. Seleção do sensor de movimento

Uma vez estudada a medição dos movimentos de comando por parte dos três sensores integrados no SensorTag, é altura de seleccionar aquele que melhor responde aos movimentos utilizados no sistema poRsCHE.

Como identificado na secção de estudo, o magnetómetro não apresenta uma boa resposta à medição dos movimentos definidos para comandar a condução da cadeira de rodas. Não só, não se identifica uma correspondência entre os tipos de movimentos e os eixos referenciais do sensor, como também não se identifica um padrão fixo em diferentes medições do mesmo tipo de movimento.

Relativamente ao estudo do acelerómetro, a situação descrita para o magnetómetro já não ocorre. Através do acelerómetro consegue-se identificar tanto uma correspondência entre os tipos de movimentos e os eixos referenciais do sensor, como se consegue também identificar um padrão fixo em diferentes medições realizadas ao mesmo tipo de movimento.

Não obstante, o acelerómetro apresentar uma medição de movimento orientada segundo os eixos referenciais, identifica-se que a medição dos movimentos de comando por parte deste sensor não é a mais adequada. A baixa aceleração aplicada aos movimentos de cabeça condiciona a percepção dos movimentos efetuados para identificar os comandos de condução da cadeira de rodas.

Como referido na secção de estudo do acelerómetro, a medição dos movimentos de comando apenas apresenta um diferencial de valores de 0,4 da unidade de medida. Este diferencial de valores é identificado como sendo baixo para uma boa identificação do comando de condução da cadeira.

Por último, o giroscópio não apresenta nenhuma das limitações identificadas anteriormente. Além da medição dos movimentos de comando ser orientada segundo um dos três eixos referenciais, apresenta também diferenciais de valores bastante superiores aos medidos pelo acelerómetro, o que facilita e aperfeiçoa a identificação do comando.

Através do giroscópio, além de se identificar com maior clareza qual o movimento de comando efetuado, também se conseguem identificar os dois movimentos que formam o movimento de comando. A título de exemplo: no gráfico 13, através da parábola com concavidade para cima consegue-se identificar a inclinação da cabeça do utilizador para a frente e através da parábola com concavidade para baixo, consegue-se identificar o retorno da posição da cabeça à posição vertical.

Pelas razões acima apresentadas, a seleção do sensor de movimento para o sistema recai sobre o giroscópio. Este sensor não só apresenta valores de medição bastante explícitos para a identificação dos movimentos de comando efetuados, como mede o movimento de retorno da posição da cabeça à posição vertical. Identificação de movimento que permite identificar, caso este não aconteça, o desmaio/indisposição do utilizador.

Capítulo 6. *Software*

Concluída a implementação *Hardware* do sistema poRsCHE, inicia-se a implementação *Software*.

O *software* do sistema poRsCHE irá ser formado por dois programas: uma aplicação móvel, para auxílio à configuração do sistema e envio do alerta de emergência, e um programa que irá ser executado na PD RaspberryPi, para comandar a condução da cadeira de rodas elétrica.

A apresentação deste capítulo é dividida por três secções. Na primeira secção apresenta-se a seleção da plataforma móvel, sistema operativo através do qual será executada a aplicação móvel. Na segunda secção é apresentada a implementação das ligações *wireless* BLE entre os três dispositivos do sistema (SensorTag, *SmartPhone* e RaspberryPi). Na última secção apresenta-se o funcionamento do sistema poRsCHE: como a aplicação móvel é utilizada para configurar o sistema, como este opera para comandar a cadeira de rodas e como é despoletado o alerta de emergência médica.

6.1. Seleção da plataforma móvel

O atual mercado de equipamentos de redes móveis é liderado por três grandes marcas no que concerne aos sistemas operativos utilizados: Android, iOS e Windows.

Embora os sistemas operativos iOS e Windows tenham expressão no mercado das telecomunicações, esta não é comparável com a do Android.

O sistema operativo Android é o grande líder e tem vindo a reforçar liderança nos últimos anos.

Sistema Operativo	4Q15 Unidades vendidas (unidades de milhar)	4Q15 Quota de mercado	4Q14 Unidades vendidas (unidades de milhar)	4Q14 Quota de mercado	4Q14/4Q15 Crescimento
Android	325,394.4	80.7%	279,057.5	76.0%	4.7%
iOS	71,525.9	17.7%	74,831.7	20.4%	-2.7%
Windows	4,395.0	1.1%	10,424.5	2.8%	-1.7%
Blackberry	906.9	0.2%	1,733.9	0.5%	-0.3%
Others	887.3	0.2%	1,286.9	0.4%	-0.2%
Total	403,109.4	100.0%	367,334.4	100.0%	---

Tabela 8 – Venda de *SmartPhones* (unidades e quotas de mercado) a nível Mundial, por Sistema Operativo utilizado, no quarto quadrimestre de 2014 e 2015 [49]

Através do estudo realizado pela Gartner²², que se apresenta na tabela 8, é expressiva não só a liderança mencionada, como o reforço dessa mesma liderança de 2014 para 2015, tendo sido o único sistema operativo a apresentar um crescimento de vendas. Segundo os dados apresentados, do número de equipamentos vendidos em 2014, 76% foram equipados com o sistema operativo Android. Já em 2015, ano transato, a percentagem de equipamentos móveis vendidos com este sistema operativo subiu de 76% para 80% das vendas mundiais [49].

Face aos factos apresentados e tendo como objetivo, a utilização do sistema poRsCHE por um maior número de indivíduos, a aplicação móvel poRsCHE é implementada através da plataforma de desenvolvimento Android.

²² Empresa de consultoria fundada em 1979 por Gideon Gartner. Consultoria feita na área IT (*Information Technology*) em mais de 85 países pelo mundo.

6.2. Ligações *wireless* BLE

O sistema poRsCHE é implementado com três dispositivos BLE que estabelecem entre si três ligações e uma comunicação *wireless*.

Antes da condução da cadeira de rodas elétrica, para configuração do sistema, são estabelecidas duas ligações:

1. Ligação estabelecida entre o *SmartPhone* e o *SensorTag* para calibrar os movimentos de comando.
2. Ligação estabelecida entre o *SmartPhone* e a PD *RaspberryPi* para comunicação dos valores de calibração.

Durante a condução da cadeira de rodas, é efetuada a terceira ligação:

3. Ligação estabelecida entre a PD *RaspberryPi* e o *SensorTag* para interpretação dos movimentos de cabeça do utilizador de modo a comandar a cadeira de rodas.

Após a condução da cadeira, caso o programa implementado na PD *RaspberryPi* interprete uma indisposição do utilizador, o *SensorTag* comunica esta situação à aplicação *Android*:

4. Comunicação entre o *SensorTag* e o *SmartPhone* para sinalizar o alarme de emergência médica.

A primeira e terceira ligação, ligações entre os dispositivos *Central*, *SmartPhone* e PD *RaspberryPi*, e o dispositivo *Peripheral*, *SensorTag*, (topologia *Connected*) é realizada através do serviço *Movement Service*. Serviço BLE do *SensorTag* que disponibiliza aos dispositivos *Central*, os valores medidos pelos seus sensores de movimento. Através da figura 45 e 46 ilustram-se as duas ligações.

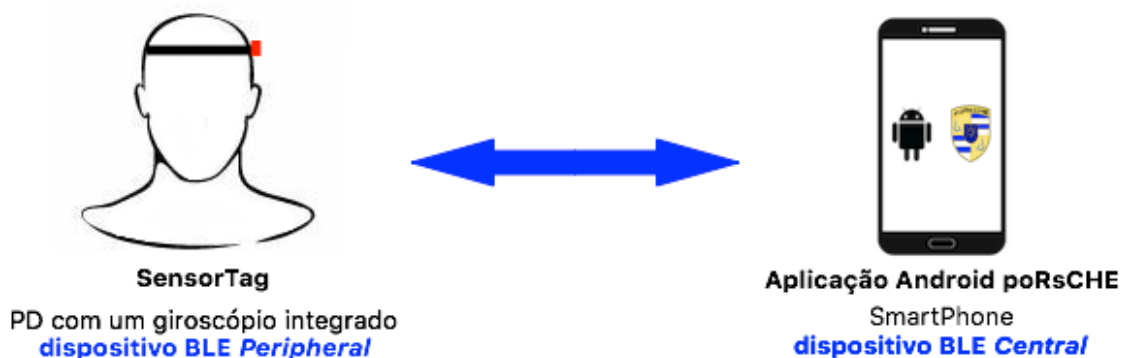


Figura 45 – 1ª ligação BLE estabelecida pelo sistema

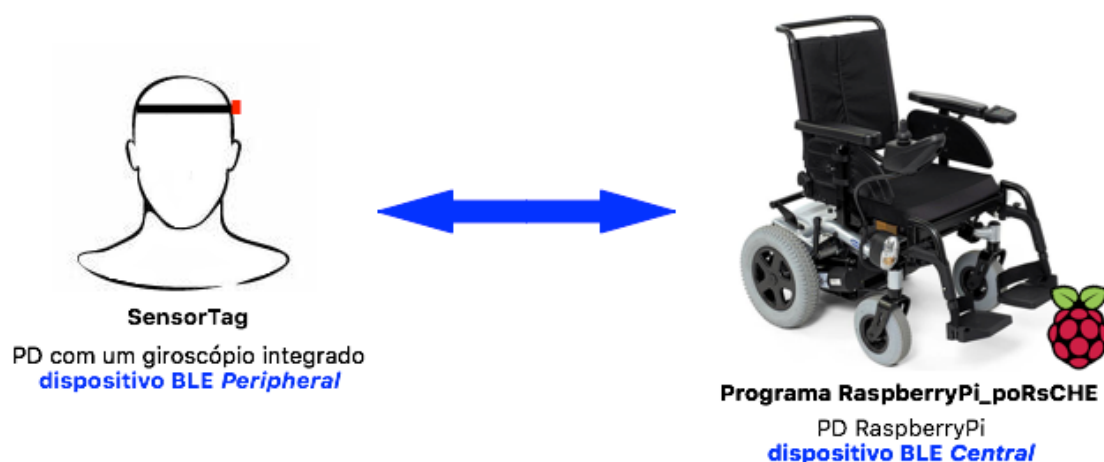


Figura 46 – 3ª ligação BLE estabelecida pelo sistema

O serviço *Movement Service* além de disponibilizar os valores de movimento medidos pelos sensores do dispositivo, disponibiliza também duas características que permitem uma configuração da apresentação destes valores. Através da tabela 9 apresenta-se a estrutura do serviço BLE.

Tipo	UUID	Permissões	Descrição
GATT Primary Service Declaration	f000aa80-0451-4000-b000-000000000000	Leitura	<i>Propertie</i> – apresentação de informação sobre o serviço <i>Movement Service</i>
GATT Characteristic Declaration	12:39:00:00:00:00: 00:00:00:00:b0:00:40:51:04:81:aa:00:f0	Leitura	<i>Propertie</i> – apresentação de informação sobre a característica <i>Movement Data</i>
<i>Movement Data</i>	f000aa81-0451-4000-b000-000000000000	Leitura e Notificações	<i>Value</i> – valor da característica <i>Movement Data</i>
<i>Client Characteristic Configuration</i>	00002902-0000-1000-8000-00805f9b34fb	Leitura e Escrita	<i>Descriptor</i> – elemento para ativação/desativação de notificações de alterações ao valor da característica
GATT Characteristic Declaration	0a:3c:00:00:00:00: 00:00:00:00:b0:00:40:51:04:82:aa:00:f0	Leitura	<i>Propertie</i> – apresentação de informação sobre a característica <i>Movement Config</i>
<i>Movement Config</i>	f000aa82-0451-4000-b000-000000000000	Leitura e Escrita	<i>Value</i> – valor da característica <i>Movement Config</i>
GATT Characteristic Declaration	0a:3e:00:00:00:00: 00:00:00:00:b0:00:40:51:04:83:aa:00:f0	Leitura	<i>Propertie</i> – apresentação de informação sobre a característica <i>Movement Period</i>
<i>Movement Period</i>	f000aa83-0451-4000-b000-000000000000	Leitura e Escrita	<i>Value</i> – valor da característica <i>Movement Period</i>

Tabela 9 – Serviço BLE *Movement Service* [27]

Para que o *software* poRsCHE (aplicação Android e programa RaspberryPi_poRsCHE) tenha acesso aos valores do giroscópio de forma a melhor compreender os movimentos de cabeça do utilizador, este, depois de se conectar com o SensorTag, executa o seguinte procedimento:

1. ativação dos eixos do giroscópio – escrita na Característica BLE *Movement Config* o valor “00000011”;
2. configuração do período de receção de notificações a alterações de valor da característica *Movement Data* (característica onde o SensorTag apresenta os valores medidos pelo giroscópio) para 100ms – escrita na Característica BLE *Movement Period* o valor “1010” (configuração para o período mínimo = 100ms);
3. ativação das notificações de alteração de valor da característica *Movement Data* – escrita no *Descriptor* BLE *Client Characteristic Configuration* o valor “01:00”;
4. leitura do valor de movimento medido pelo giroscópio – leitura do valor da Característica BLE *Movement Data* (leitura feita sempre a seguir à receção de uma notificação de alteração de valor).

O procedimento apresentado além de configurar o serviço BLE para apenas receber os valores do giroscópio, reduzindo assim a transmissão de dados para o essencial ao sistema (poupança de energia consequente de uma redução dos dados transmitidos), configura também o período de notificações para o valor mínimo, 100ms.

Para que o sistema consiga a melhor compreensão possível dos movimentos de cabeça efetuados pelo utilizador, é crucial uma leitura de medições com a maior frequência possível. A configuração do período de notificações para 100ms (10 leituras de medição por segundo) torna a compreensão dos movimentos efetuados bastante razoável.

No capítulo anterior, capítulo 5., foi referido que a versão do SensorTag adotada para a implementação do sistema foi a versão CC2650. Referiu-se que para além da primeira versão, CC2541, apresentar piores consumos energéticos, apresentava também uma falha de *firmware*. A falha de *firmware* detetada prendeu-se com a configuração apresentada no parágrafo anterior. Embora tivesse sido feita uma configuração do serviço BLE idêntica à apresentada anteriormente, definindo o período de notificações para 100ms, o SensorTag CC2541 apenas notificava o *software* poRsCHE de 1000 em 1000ms; falha que, face aos argumentos apresentados, justificou o abandono desta.

Para a segunda ligação BLE, ligação estabelecida entre a aplicação Android e a PD RaspberryPi (ligação ilustrada através da figura 47), o sistema implementa um serviço próprio. O serviço implementado denomina-se *Commands Calibration* e é organizado segundo a estrutura apresentada na tabela 10.

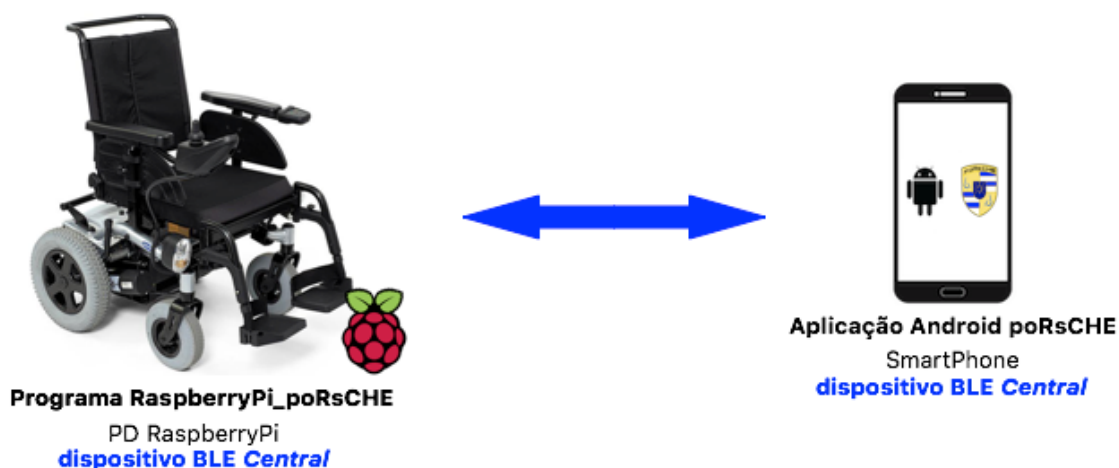


Figura 47 – 2ª ligação BLE estabelecida pelo sistema

Tipo	UUID	Permissões	Descrição
<i>GATT Primary Service Declaration</i>	f00000870-0aaa-aa00-000a-bfffffff	Leitura	<i>Propertie</i> – apresentação de informação sobre o serviço <i>Commands Calibration</i>
<i>Forward</i>	f00000871-0aaa-aa00-000a-bfffffff	Escrita	<i>Value</i> – valor da característica <i>Forward Data</i>
<i>Stop</i>	f00000872-0aaa-aa00-000a-bfffffff	Escrita	<i>Value</i> – valor da característica <i>Stop Data</i>
<i>Right</i>	f00000873-0aaa-aa00-000a-bfffffff	Escrita	<i>Value</i> – valor da característica <i>Right Data</i>
<i>Left</i>	f00000874-0aaa-aa00-000a-bfffffff	Escrita	<i>Value</i> – valor da característica <i>Left Data</i>
<i>Boolean Calibration Success Data</i>	f00000875-0aaa-aa00-000a-bfffffff	Leitura	<i>Value</i> – valor da característica <i>Boolean Calibration Success Data</i>

Tabela 10 – Serviço BLE *Commands Calibration*

Para a identificação dos valores de amplitude que o utilizador do sistema apresenta aos movimentos de comando, o serviço BLE é implementado com quatro características de escrita, uma por cada movimento de comando. Em cada uma destas características, a aplicação móvel escreve os valores de maior amplitude que o utilizador apresenta ao movimento. A título de exemplo: no movimento de flexão (comando *Forward*), o utilizador efetua dois movimentos; primeiro inclina a cabeça em direção ao seu peito e em seguida, retorna a posição da cabeça à posição vertical. Destes dois movimentos, resultam as duas parábolas ilustradas no gráfico 17. Os valores de maior amplitude, e que serão utilizados como valores de calibração, correspondem aos picos das parábolas e são representados no gráfico pelas duas circunferências a vermelho.

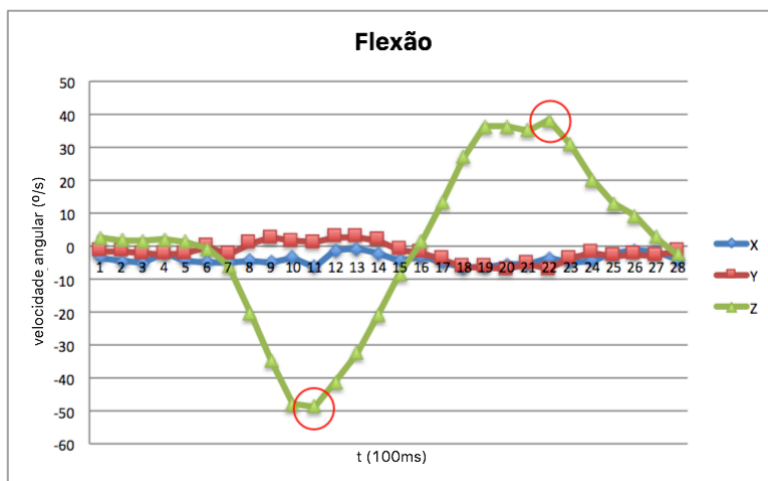


Gráfico 17 – Valores de calibração (amplitudes máximas)

Uma vez realizada a escrita dos valores de calibração nas características BLE, a aplicação lê o valor da Característica BLE *Boolean Calibration Success Data*. Valor que irá ser diferente consoante a receção ou não de todos os valores de calibração.

Se o o programa RaspberryPi_poRsCHE identificar uma correta escrita de todos os valores de calibração nas características BLE de acesso à escrita, este coloca o valor da Característica BLE *Boolean Calibration Success Data* a “1”; caso contrário, coloca o valor a “0”. Através da leitura desta variável, a aplicação móvel tem *feedback* da comunicação BLE, repetindo a transmissão dos dados caso se justifique.

A implementação de uma característica de leitura para garantir a transmissão completa dos valores de calibração deveu-se à deteção de uma falha nas comunicações BLE. Nos primeiros testes de comunicação entre a aplicação móvel Android e a PD RaspberryPi, identificou-se que, quando os dois equipamentos não se conheciam (estabelecimento de uma primeira ligação), o tempo de acesso à escrita para transmissão dos valores de calibração expirava. Com a implementação desta característica de leitura, garante-se que a aplicação Android apenas termina a execução de transmissão de dados quando a PD RaspberryPi possui todos os dados necessários.

Por último, caso o programa RaspberryPi_poRsCHE identifique uma indisposição do utilizador, é acionada a comunicação BLE entre o SensorTag e a aplicação Android (topologia *Broadcast*).

Assim que o programa RaspberryPi_poRsCHE identifique uma indisposição do utilizador, este desconecta-se do SensorTag. Ao ser desconectado da PD RaspberryPi, o dispositivo BLE *Peripheral*, para que outros dispositivos BLE *Central* tenham conhecimento da sua atividade, inicia a transmissão de pacotes de sinalização. Estando a aplicação Android à procura de dispositivos BLE *Peripheral*, esta é avisada do estado disponível do SensorTag, e interpretando-o como o sinal de emergência poRsCHE, despoleta o envio da SMS de alerta de emergência médica²³. Através da figura 48 ilustra-se a comunicação BLE apresentada.

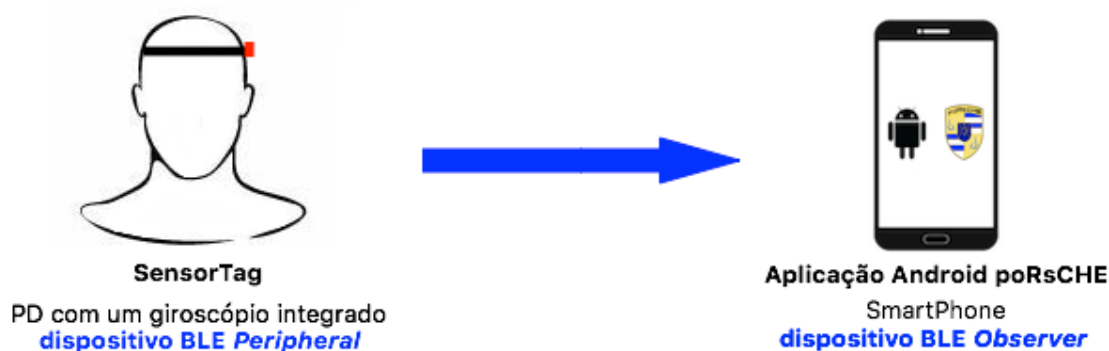


Figura 48 – Comunicação BLE alarme de emergência

²³ Para que a aplicação Android não gaste os recursos energéticos do *SmartPhone* em demasia, a procura dos dispositivos BLE é feita em intervalos de cinco segundos espaçados por intervalos de igual período.

6.3. Funcionamento do sistema poRsCHE

O funcionamento do sistema embebido poRsCHE é dividido em três períodos: pré-condução da cadeira de rodas, condução e pós-condução.

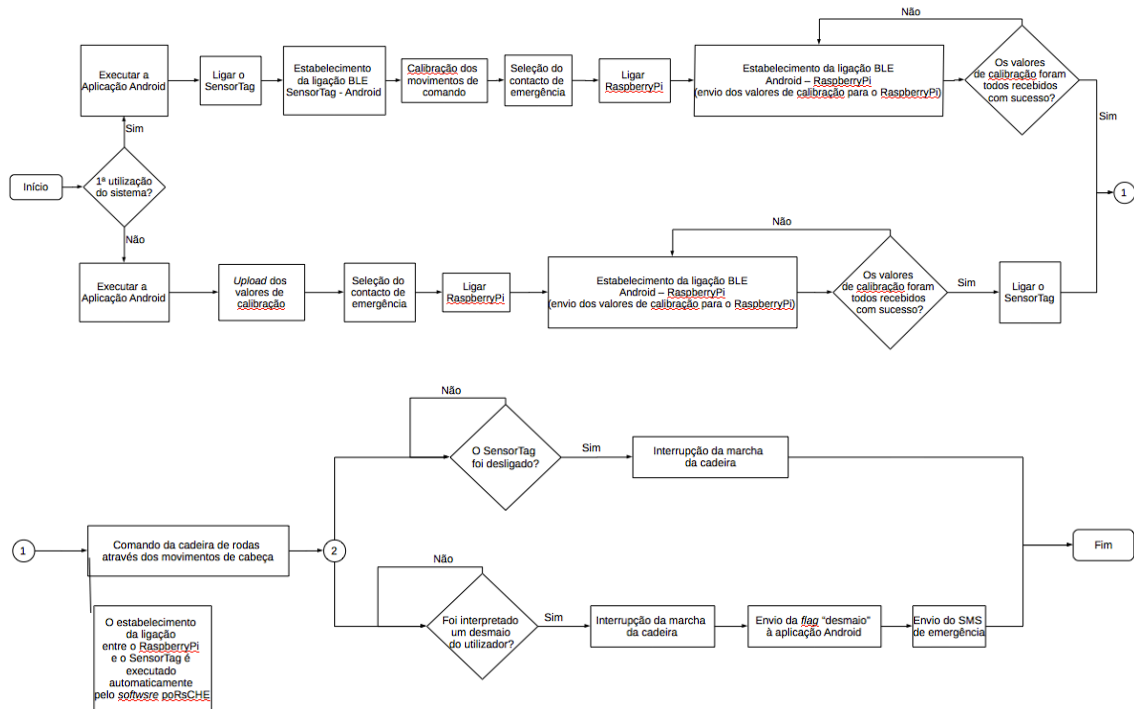


Figura 49 – Fluxograma do sistema poRsCHE

A figura 49 ilustra o fluxograma do funcionamento geral do sistema poRsCHE. Até ao nó 1 é representado o primeiro período, período de configuração do sistema. Entre os nós 1 e 2 é representado o período de funcionamento onde o utilizador consegue comandar a cadeira de rodas através dos seus movimentos de cabeça (o processador do sistema interpreta os valores de velocidade angular recebidos do SensorTag para comandar a cadeira). A partir do nó 2 é representado o período onde o sistema termina a sua execução por ordem do cuidador do utilizador (desligar do SensorTag) ou por interpretação de um desmaio do utilizador (interpretação de um movimento de desmaio seguido de um envio da SMS de emergência).

6.3.1. Pré-condução da cadeira de rodas

Antes do utilizador do sistema poRsCHE poder comandar a condução da cadeira de rodas, é necessário efetuar a configuração do sistema.

Como referido anteriormente, no capítulo 3., esta configuração é feita com recurso a uma aplicação móvel. Dadas as limitações físicas do utilizador do sistema, esta operação, pré-condução da cadeira, necessita incondicionalmente de auxílio de uma segunda pessoa.

Uma vez instalada a aplicação Android no *SmartPhone* do utilizador, este, com auxílio da segunda pessoa, terá que realizar as seguintes operações:

1. Calibração dos movimentos de comando.
 - a. Numa primeira utilização, a aplicação guia o utilizador para a Atividade²⁴ *Calibration*, onde através da conexão com o SensorTag, são detetadas as amplitudes máximas que este apresenta aos movimentos de comando.

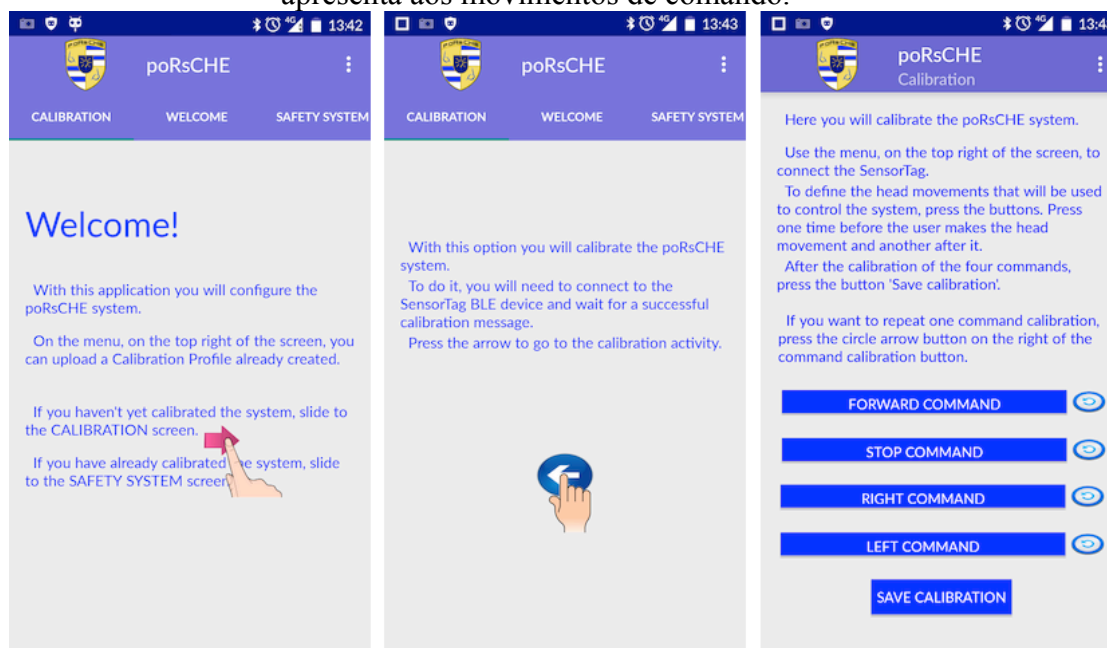


Figura 50 – Calibração dos movimentos de comando através da aplicação Android

Para executar esta operação é indicado ao utilizador que execute o seguinte procedimento:

- i. através do menu, apresentado no canto superior direito do ecrã, conectar o SensorTag. A conexão do equipamento é dividida em duas fases: *Scan*, descoberta da presença do equipamento na periferia do *SmartPhone*; e *Connection*, conexão com o equipamento (conexão realizada ao ser selecionado o nome do equipamento que será apresentado por cima da opção *Scan*);
- ii. realizar os quatro movimentos de comando. A realização de cada um dos movimentos será feita entre uma primeira seleção do botão que indica o movimento de comando e uma segunda seleção do mesmo (para repetir a calibração de um comando, é indicado ao utilizador que apague a primeira calibração efetuada através do botão ilustrado

²⁴ Elemento isolado da atividade da aplicação Android que é executado segundo instruções inseridas na interface gráfica.

com uma seta circular, disposto ao lado do botão referente ao comando calibrado);

- iii. guardar a calibração com o nome do utilizador. A calibração efetuada é guardada na memória do *SmartPhone* para posteriores utilizações (apenas será possível guardar os valores de calibração após a calibração dos quatro movimentos de comando; se algum dos quatro comandos não for calibrado, a aplicação informa o utilizador para que efetue a calibração do comando em falta).
- b. Em utilizações posteriores à primeira, o utilizador pode realizar o *upload*²⁵ dos seus valores de calibração utilizando o menu do *layout*²⁶ “Welcome!”. Menu que apresenta os perfis de calibração criados e uma opção de eliminação.

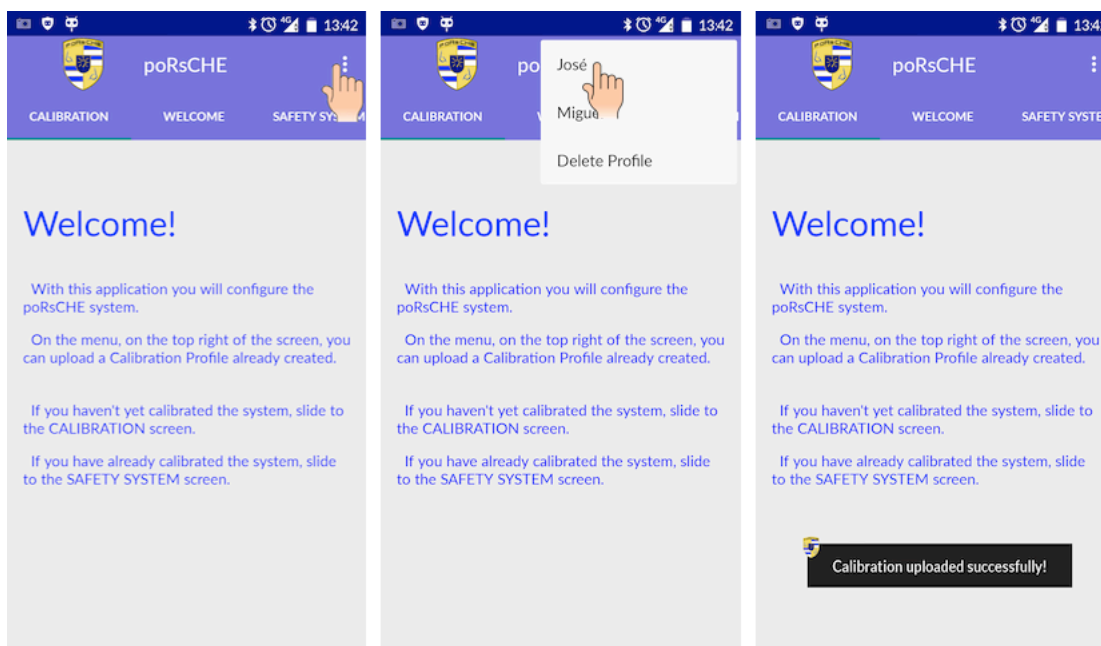


Figura 51 – Aplicação Android - *upload* dos valores de calibração

²⁵ Transmissão de dados efetuada da memória do *SmartPhone* para a aplicação Android.

²⁶ Interface gráfica associada a uma das Atividades da aplicação Android.

2. Seleção do contacto de emergência. Concluída a calibração dos movimentos de comando, a aplicação guia o utilizador para a Atividade *Emergency Contact*.

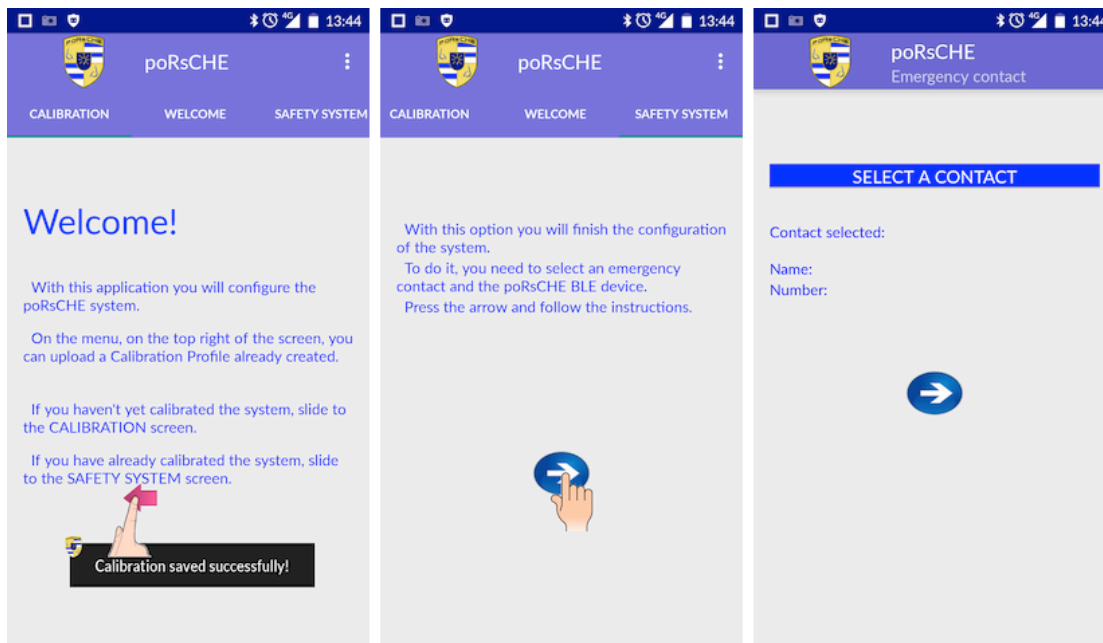


Figura 52 – Aplicação Android – seleção do contacto de emergência

Uma vez no *layout* da Atividade *Emergency contact*, o utilizador dispõe do botão “SELECT A CONTACT” para aceder à lista telefónica do *SmartPhone* e seleccionar o contacto que irá receber, caso aconteça algo que se justifique, a SMS de alerta de emergência médica. O contacto pode ser de um telefone móvel ou fixo. Caso o contacto esteja mal registado, a aplicação identifica a situação e avisa o utilizador, de modo a que este selecione um contacto válido.

- Envio dos valores de calibração ao processador do sistema (PD RaspberryPi). Selecionado o contacto de emergência, o utilizador é guiado para a última Atividade da aplicação.

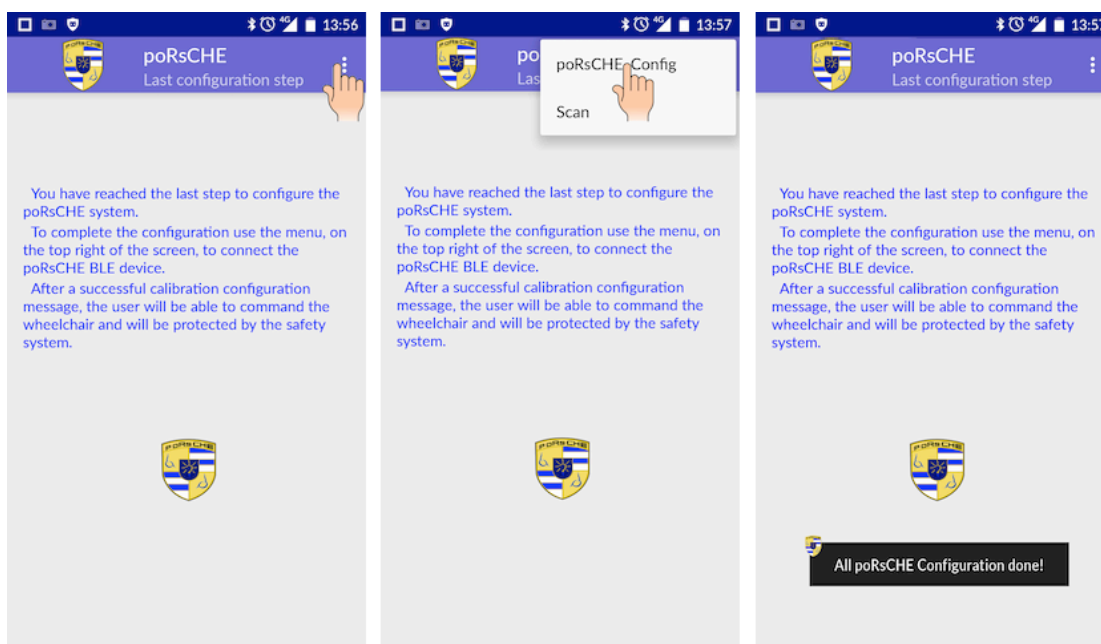


Figura 53 – Aplicação Android - comunicação à PD RaspberryPi dos valores de calibração

Através da Atividade *Last configuration step*, o utilizador conclui a configuração do sistema. Conectando o dispositivo RaspberryPi BLE poRsCHE_Config, de igual modo como conectou o SensorTag, despoleta o envio dos valores de calibração a este²⁷. Após o envio de todos os valores de calibração, a aplicação apresenta uma mensagem a avisar o utilizador que toda a configuração do sistema foi realizada com sucesso.

Para que a aplicação possa despoletar o envio da SMS de alerta de emergência, assim se justifique, o *SmartPhone* deve ser colocado na cadeira de rodas e a atividade desta deve permanecer na Atividade *Last configuration step*.

²⁷ A conexão da aplicação Android ao dispositivo BLE RaspberryPi é subsequente à ligação deste.

6.3.2. Condução da cadeira de rodas

Uma vez configurado o sistema, o utilizador pode, através dos seus movimentos de cabeça, comandar autonomamente a cadeira de rodas.

Comando efetuado segundo dois estados:

- *Standby* (liberdade de movimentos – estado onde o utilizador movimenta a cabeça sem afetar o comando de condução);
- *Driving* (estado inicial do sistema – interpretação dos movimentos para comando da condução).

E seis subestados do estado *Driving*:

- *Stop* (subestado inicial – cadeirada parada),
- *1st gear* (deslocação em frente numa velocidade reduzida),
- *2nd gear* (deslocação em frente numa velocidade moderada),
- *Backward* (deslocação em marcha-atrás),
- *Turning Right* (rotação para a direita),
- *Turning Left* (rotação para a esquerda).

Após deteção do SensorTag, o programa RaspberryPi_poRsCHE²⁸ conecta-se a este e configura o serviço que lhe fornece os valores de movimento medidos pelo giroscópio.

Uma vez configurado o serviço *Movement Service*, o programa interpreta os movimentos de comando segundo a análise apresentada nos gráficos 18, 19, 20, 21, 22 e 23.

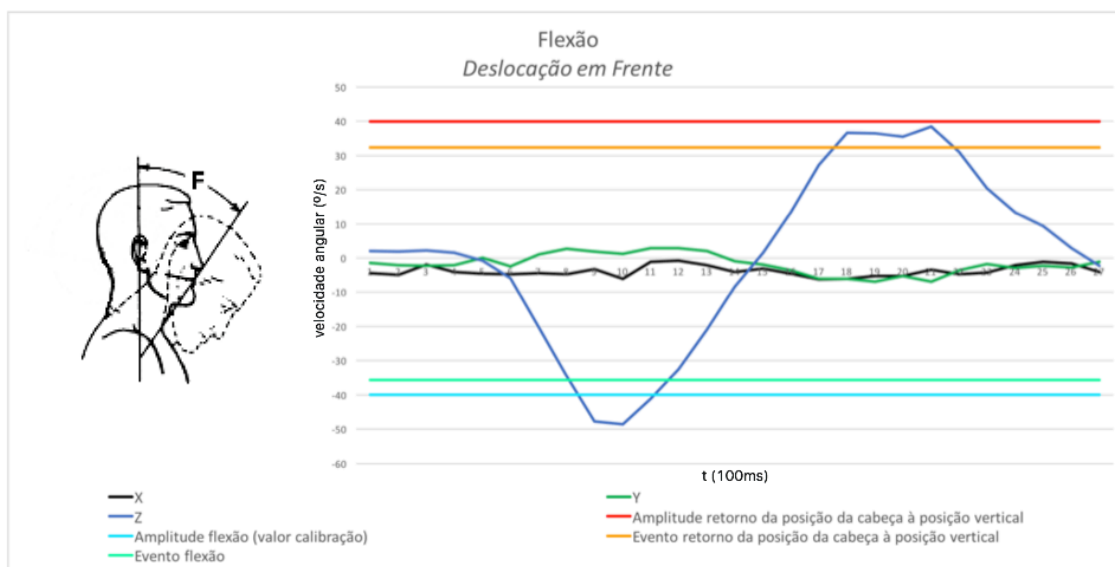


Gráfico 18 – Identificação do Movimento de comando Deslocação em frente

Através do gráfico 18, apresenta-se: os valores de movimento medido no eixo Z do giroscópio (eixo referencial utilizado para identificação da inclinação da cabeça nos movimentos anterior/posterior), os valores de calibração obtidos através da aplicação móvel (amplitudes máximas dos dois movimentos que formam o movimento de comando), valores de interpretação de evento (valores de calibração menos desvio padrão

²⁸ Programa executado no processador do sistema (PD RaspberryPi) para controlar a deslocação da cadeira.

calculado²⁹) e os valores de movimento medido nos dois outros eixos referenciais do giroscópio.

Os valores de calibração apresentados em todos os gráficos são apenas exemplificativos, não sendo originários de nenhuma calibração efetuada por um elemento do público alvo.

Uma vez detetado o evento flexão (recepção de um valor igual ou superior ao calculado como evento), o programa vai aguardar até três segundos (30 amostras de valor) para o utilizador confirmar o comando através do retorno da posição da cabeça à posição vertical. Se este segundo evento se realizar, o sistema dá ordem aos motores da cadeira para que esta se desloque em frente (alteração do estado do sistema para *1st gear*, caso o subestado presente seja *Stop* ou para *2nd gear*, caso o subestado presente seja *1st gear*); caso contrário, não se realizando o segundo evento dentro do intervalo de tempo de três segundos, o programa interpreta que o utilizador desmaiou/sentiu-se indisposto, interrompe a deslocação da cadeira e dispara o alarme de emergência médica.

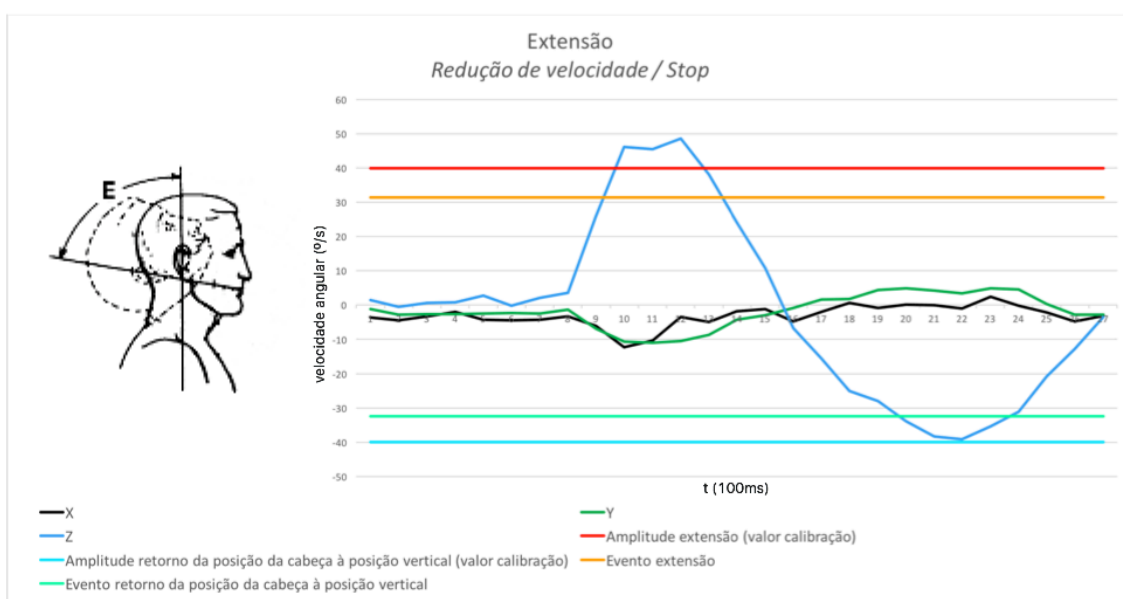


Gráfico 19 – Identificação do Movimento de comando Redução de velocidade / Stop

A análise apresentada através do gráfico 19, é idêntica à apresentada no gráfico 18. Caso o programa detete o evento extensão, aguarda três segundos para identificar o evento retorno da posição da cabeça à posição vertical. Se o segundo evento ocorrer, o sistema dá ordem aos motores para interromperem o seu funcionamento (alteração do subestado do sistema para *Stop*, caso o presente seja *1st gear*) ou para redução da velocidade (alteração do subestado do sistema para *1st gear*, caso o presente seja *2nd gear*).

²⁹ Para o cálculo do desvio padrão dos valores de amplitude máxima aos movimentos de comando, foram efetuadas dez calibrações por um indivíduo sem limitações. A seleção de um indivíduo com total mobilidade foi feita por se entender que este apresenta maiores variações ao movimento de cabeça.

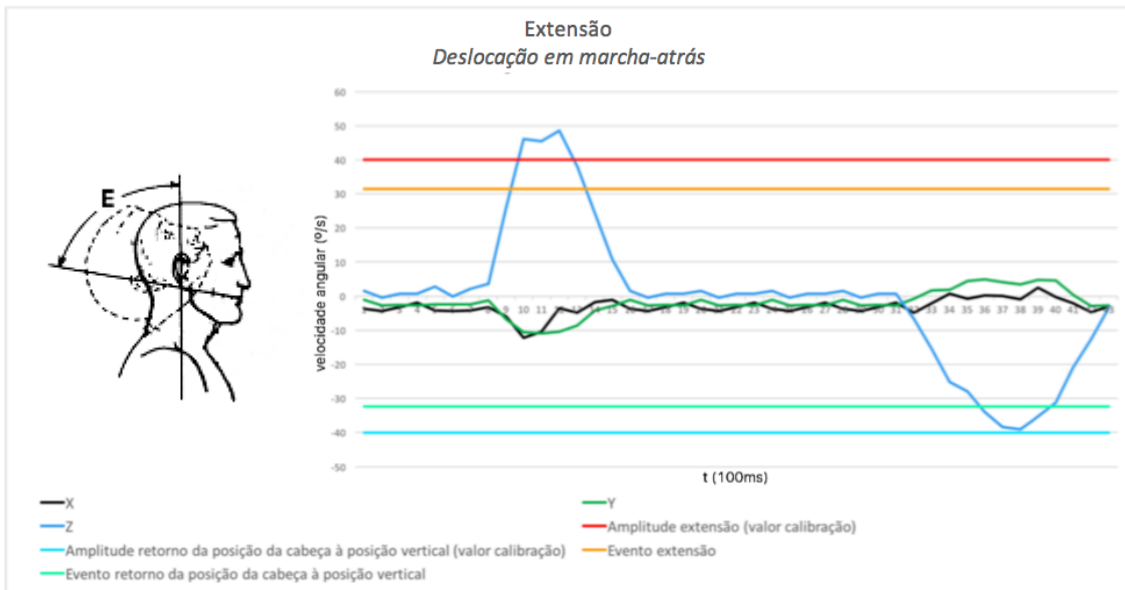


Gráfico 20 – Identificação do Movimento de comando Deslocação em marcha-atrás

Através do gráfico 20 apresenta-se a interpretação do comando para colocar a cadeira a deslocar-se em marcha-atrás. Estando o sistema no estado *Driving* e no subestado *Stop*, após deteção do evento extensão, o programa verifica durante um segundo e meio (quinze amostras de valor)³⁰ se o utilizador mantém a cabeça em extensão (não retorno da posição da cabeça à posição vertical). Se a condição se confirmar, o sistema dá ordem aos motores da cadeira para que esta se desloque em marcha-atrás (alteração do subestado do sistema para *Backward*). Uma vez detetado o evento de retorno da posição da cabeça à posição vertical, o sistema dá ordem aos motores para interromperem o seu funcionamento (alteração do subestado do sistema para *Stop*).

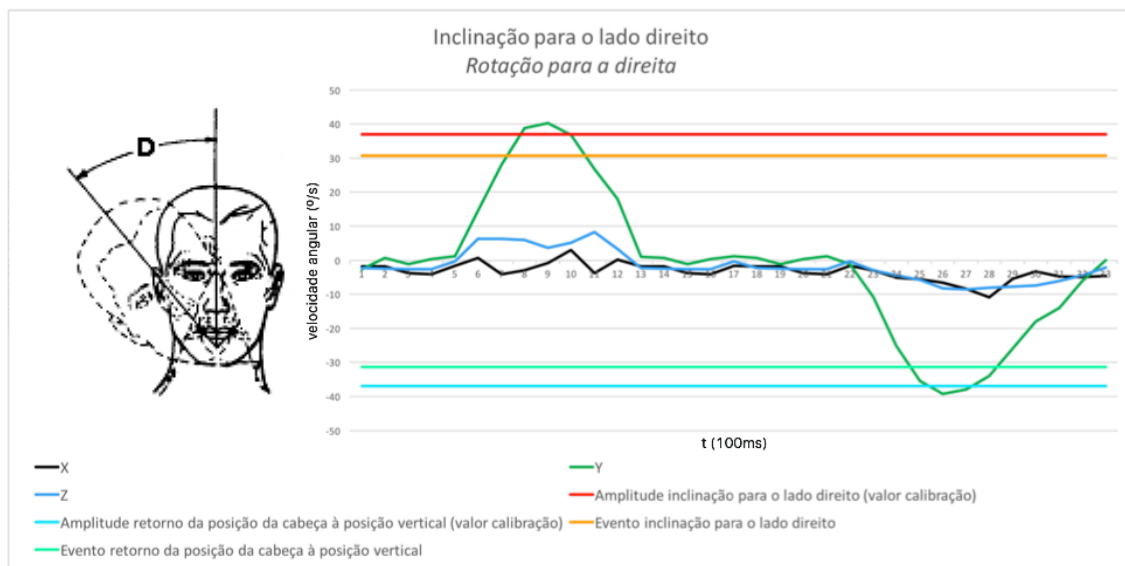


Gráfico 21 – Identificação do Movimento de comando Rotação para a direita

Através do gráfico 21, apresenta-se: os valores de movimento medido no eixo Y do giroscópio (eixo referencial utilizado para identificação da inclinação da cabeça nos movimentos laterais), os valores de calibração obtidos através da aplicação móvel (amplitudes máximas dos dois movimentos que formam o movimento de comando),

³⁰ Período de tempo testado e ajustado com o público alvo por forma a garantir a melhor relação eficiência/ergonomia.

valores de interpretação de evento (valores de calibração menos desvio padrão calculado) e os valores de movimento medido nos dois outros eixos referenciais do giroscópio.

Uma vez detetado o evento inclinação para o lado direito (receção de um valor igual ou superior ao calculado como evento), o programa vai dar ordem de comando aos motores da cadeira para a rodarem para o lado direito (subestado *Turning Right*). Uma vez com a cadeira em rotação, o programa vai aguardar cinco segundos (tempo que o protótipo demora a realizar uma rotação de 360°) para que o utilizador retome a posição da cabeça à posição vertical (evento que interrompe a rotação da cadeira – passagem do sistema ao subestado anterior, *Stop/1st gear/2nd gear*). Se o segundo evento for detetado, o sistema passa ao subestado anterior e o utilizador pode inserir um novo comando; se o evento não for detetado, o programa interpreta que o utilizador desmaiou/sentiu-se indisposto, interrompe a deslocação da cadeira e dispara o alarme de emergência médica.

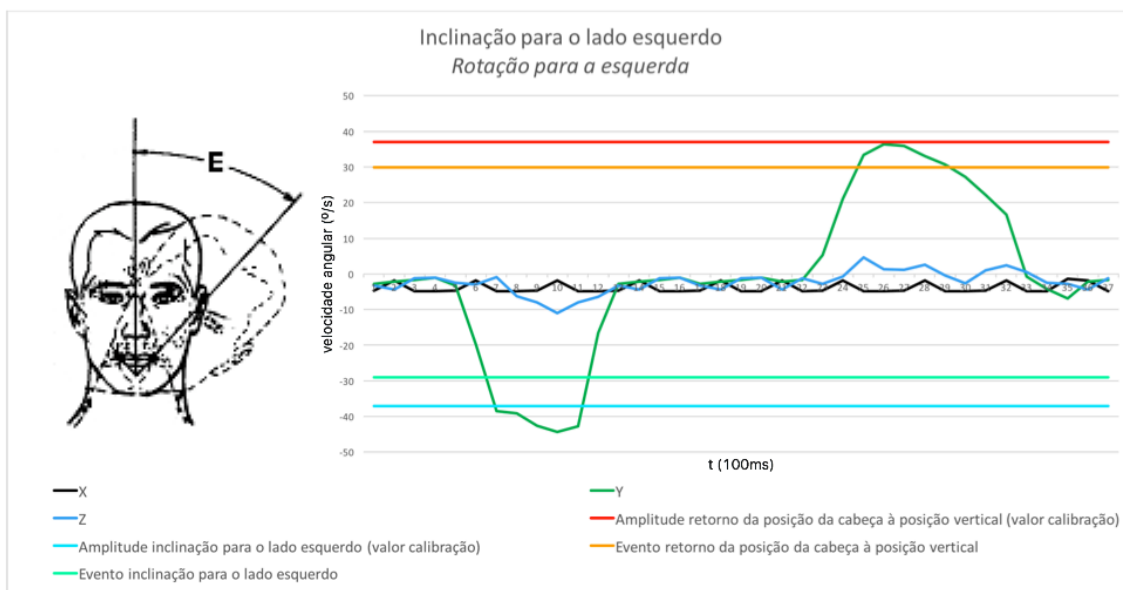


Gráfico 22 – Identificação do Movimento de comando Rotação para a esquerda

A análise apresentada através do gráfico 22, é igual à apresentada no gráfico 21, alterando-se apenas o lado para o qual a cadeira poRsCHE irá rodar.

Estando o sistema no estado *Driving* e no subestado *Stop*, e querendo o utilizador realizar outra tarefa que não a condução da cadeira de rodas, este pode colocar o sistema em *Standby* (liberdade de movimentos). Para alteração do estado do sistema, o utilizador tem que realizar uma sequência de dois movimentos de extensão. Esta sequência é representada através do gráfico 23.

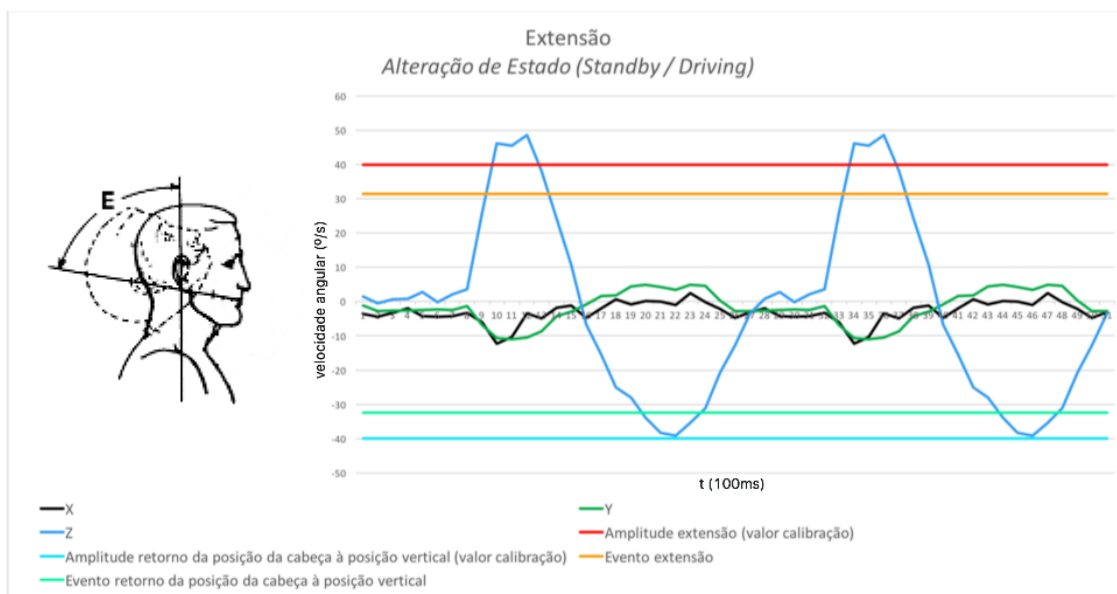


Gráfico 23 – Identificação do Movimento de comando Alteração de Estado

A sequência apresentada tem que ser realizada num intervalo até seis segundos para que o sistema interprete o movimento de comando. Caso o utilizador não realize os dois movimentos de extensão no intervalo de tempo definido, o sistema permanece no estado atual.

Caso o utilizador execute um movimento que não alcance os valores para identificação de evento, o sistema identifica a situação para que o movimento contrário não seja mal interpretado. Um exemplo desta situação é apresentado através do gráfico 24.

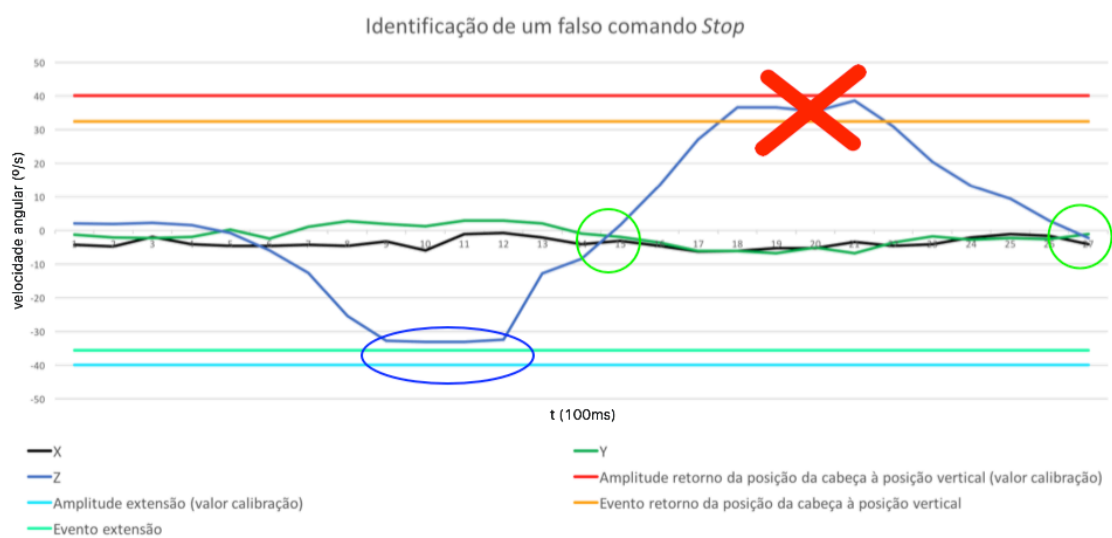


Gráfico 24 – Identificação de um falso comando

Identificando um pico de amplitude que não alcance os valores de evento (valores representados através da elipse azul), o sistema descarta o evento sinalizado pela cruz vermelha (evento originado pelo retorno da cabeça à posição vertical). Para voltar a considerar a deteção de eventos (não descartar mais eventos), o sistema conta o número de passagens das medições efetuadas pelo referencial de origem (valores representados através das circunferências a verde). A seguir a uma segunda passagem pelo referencial (estabilização da cabeça), o sistema volta a tentar identificar eventos de comando.

Paralelamente à interpretação e execução dos comandos de condução inseridos através dos movimentos de cabeça do utilizador, o sistema efetua a interpretação da informação proveniente do sinal do sonar. Estando a cadeira em deslocação e uma vez detetado um obstáculo no seu caminho, a uma distância igual ou inferior a metade do comprimento da cadeira (no caso do protótipo 15cms), o programa imobiliza-a (passagem do subestado do sistema ao subestado *Stop*) e apenas executa ordem de comando aos movimentos de comando laterais ou de marcha-atrás. Após o obstáculo ser contornado, o sistema passa a aceitar novamente todos os movimentos de comando de condução.

6.3.3. Pós-condução da cadeira de rodas

Após o utilizador concluir a sua deslocação, ou após disparo do alarme de emergência médica, a execução do programa RaspberryPi_poRsCHE é terminada.

Para terminar a execução do programa após conclusão da deslocação, o cuidador do utilizador desliga o equipamento SensorTag. Ao detetar o pressionar do botão *Power* do SensorTag³¹ e consequente término da ligação BLE, o programa termina a sua execução e desliga o equipamento presente na cadeira.

Caso o alarme de emergência seja disparado, o programa termina a sua execução, deixando o SensorTag no estado disponível para estabelecimento de ligação com outro dispositivo (transmissão de pacotes de sinalização). Ao detetar um pacote de sinalização do SensorTag, a aplicação Android despoleta o envio da SMS de alerta de emergência.

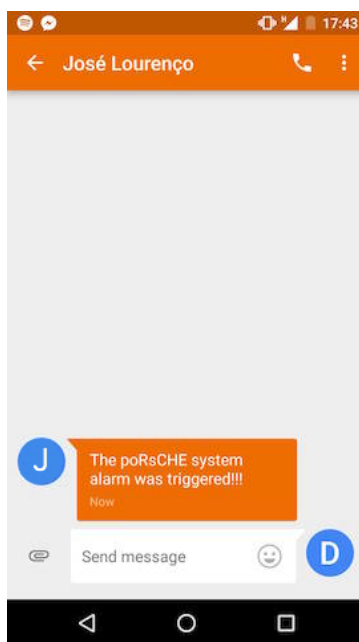


Figura 54 – Receção da SMS de alerta de emergência no *SmartPhone* do cuidador

³¹ A análise do estado do botão *Power* do SensorTag é feita através da receção de notificações por parte do dispositivo. Notificações ativas através da escrita do valor “01:00” no *descriptor BLE Client Characteristic Configuration* da característica *Key Press State*.

Capítulo 7. Conclusão

Após alguns contratempos e aprendizagens, que se enunciam a seguir, o sistema proposto é dado por concluído.

No decorrer da implementação *hardware*, o projeto confrontou-se com três contratempos.

A primeira versão do SensorTag, versão atual à altura da seleção da PD, apresentou-se com uma falha de *firmware* que viria a condicionar a eficiência do sistema. Ao não se conseguir uma receção de medições de velocidade angular superiores a uma amostra por segundo, o sistema não se apresentava rápido o suficiente para interpretar, em tempo real, os movimentos de comando. Após deteção da falha apresentada, de alguns contactos infortúnios com a Texas Instruments, e uma vez lançada a segunda versão do equipamento, esta foi abandonada.

Abandonada a primeira versão do SensorTag, e adquirindo a atual (SensorTag CC2650), o sistema deparou-se com um segundo contratempo. Embora o fabricante já tivesse lançado algumas aplicações móveis para teste do equipamento e publicado algumas informações sobre o funcionamento deste; não tinha ainda publicado informação necessária para que se conseguisse integrar este num sistema com outros equipamentos. Aquando da aquisição desta versão, a Texas Instruments não tinha publicado informação sobre como descodificar os dados enviados pelo SensorTag; informação apenas disponibilizada em inícios de setembro de 2015.

Ultrapassados os contratempos com o equipamento SensorTag, o projeto deparou-se com um terceiro contratempo. O módulo rádio inicialmente selecionado para estabelecer a ligação BLE entre o SensorTag e o processador do sistema, Microchip RN4020, não tinha acesso a todos os serviços do equipamento (nomeadamente ao serviço *MovementService*, serviço utilizado para aceder às medições de velocidade angular). Após vários contactos com o fabricante, que não foram suficientes para resolver a falha de *firmware* encontrada, e contando com três meses de atraso no desenvolvimento do projeto, foi tomada a decisão de se abandonar o módulo rádio da Microchip e de se optar por uma opção mais *friendly user*, integração de uma *pen*, com um módulo rádio BLE, numa PD RaspberryPi (opção que como referido anteriormente se mostrou ser uma mais valia por apresentar uma maior capacidade de processamento).

Do desenvolvimento da presente dissertação, não ficam apenas três contratempos por referir; também algumas aprendizagens, que vieram aperfeiçoar o sistema, são consideradas importantes de registo.

Antes de o sistema poRsCHE ser apresentado e testado no CMRA, este não era implementado com filtragem de dados; o protótipo não se deslocava em marcha-atrás; apenas existia um estado de funcionamento (*Driving*); e não tinha sido implementada uma interface gráfica.

Numa fase pré-testes, o tratamento dos valores de velocidade angular era feito sem qualquer tipo de filtragem. Entendia-se que não havendo medições que pudessem levantar dúvidas na interpretação do movimento; um processamento cru dos dados recebidos (processamento dos dados sem filtragem) seria favorável para alcançar uma resposta, ao movimento efetuado, o mais próximo possível do tempo real. Realizados os testes com o público alvo, apreendeu-se que a abordagem adotada não estava errada. O tempo de resposta que o sistema apresentava para colocar a cadeira em movimento, era da ordem dos duzentos milissegundos; valor bastante positivo. No entanto, com o decorrer dos testes, identificaram-se situações que necessitavam de um processamento mais rigoroso. Após um espirro de um dos pacientes, espirro que colocou o protótipo em

marcha, identificou-se que movimentos repentinos (movimentos de comando efetuados num período igual ou inferior a 1,2 segundos) teriam que ser descartados da interpretação dos movimentos como movimentos de comando (implementação de um filtro passa baixo que descarta movimentos com período $\leq 1200\text{ms}$).

Dos objetivos iniciais para o sistema, apenas estavam definidos três sentidos de deslocação para o protótipo: marcha em frente, rotação para a esquerda e rotação para a direita. Entendia-se que face às limitações físicas do público alvo, uma deslocação em marcha-atrás seria uma deslocação de difícil execução. Interpretou-se que, existindo a capacidade de rotação de 180° da cadeira de rodas, esta poderia ser conduzida em todas as direções possíveis. Uma vez apresentado o projeto aos profissionais do CMRA (engenheiros biomédicos e terapeutas ocupacionais), foi detetado que a não existência de uma deslocação em marcha-atrás era uma falha crucial para a deslocação do público alvo. Não obstante uma rotação de 180° permitir uma deslocação em todas as direções possíveis, os espaços onde o público alvo se movimenta (nomeadamente espaços interiores – ligações entre divisões e divisões pequenas como as casas de banho) não permitem que a rotação da cadeira seja feita com ângulo de rotação necessário para o utilizador se movimentar em todas as direções.

O objetivo inicial do projeto era através de uma interpretação de movimentos de cabeça, substituir os atuais comandos especiais para pessoas tetraplégicas comandarem uma cadeira de rodas. Numa fase inicial, pré-contacto com o público alvo, o sistema interpretava interruptamente todos os movimentos efetuados por forma a identificar movimentos de comando. Após contacto com os pacientes do CMRA, foram implementados dois estados de funcionamento: *Driving* (interpretação dos movimentos de comando para condução da cadeira) e *Standby* (liberdade de movimentos, estado onde o utilizador pode efetuar outras ações que não a condução da cadeira). Com esta implementação, o sistema não só permite que o utilizador seja capaz de conduzir a sua cadeira, como de realizar outras ações entre deslocações.

Com a implementação dos dois estados referidos, foi implementada uma interface gráfica com base em LEDs. Esta interface gráfica não só informa o utilizador do estado atual do sistema, como o avisa da disponibilidade de inserção de um novo comando (o sistema informa através da luz de uma LED que já processou com sucesso o movimento efetuado e que já está disponível a aceitar um novo movimento de comando). A interface gráfica implementada, sendo o mais simples possível, proporciona uma rápida interpretação que não condiciona a capacidade de o utilizador conduzir a cadeira.

O resultado deste projeto foi alvo de publicação de um artigo científico [50] apresentado no Simpósio de Informática INForum 2016 (ver anexo).

Neste capítulo apresentam-se os resultados obtidos nos testes efetuados pelo público alvo, as ilações retiradas após conclusão do sistema e objetivos futuros para aperfeiçoamento do mesmo.

7.1. Validação do Sistema

Uma vez concluída a implementação *Hardware e Software*, realizaram-se testes para detetar possíveis falhas e para se efetuar os ajustes que se consideraram necessários para aperfeiçoar o funcionamento do sistema.

Os testes efetuados, calibrações (obtenção de valores de movimentos de comando), testes de comando e condução, foram realizados por três elementos do público alvo. Indivíduos tetraplégicos (lesões vertebro-medulares altas, inexistência de movimentação motora a baixo dos ombros), maiores de idade e pacientes do CMRA.

Através das calibrações efetuadas pelos três utilizadores, foi confirmado que um intervalo de tempo de três segundos é suficiente para que o sistema interprete eficazmente os movimentos de cabeça como ordens de comando (leitura do retorno da posição da cabeça à posição vertical).

O primeiro utilizador (paciente do sexo masculino na casa dos sessenta anos de idade) apresenta movimentos de cabeça muito limitados e por esse motivo não é capaz de comandar o sistema poRsCHE. Nos testes realizados foram efetuados vários ajustes aos valores definidos como evento (aumento/diminuição do desvio padrão calculado para obter maior/menor sensibilidade à deteção dos movimentos de comando). No entanto as limitações físicas do utilizador impediram que este controlasse eficientemente o sistema. O utilizador apenas conseguiu dar algumas ordens de comando através dos movimentos de flexão e extensão; não só apresenta movimentos de cabeça limitados (valores de calibração para os movimentos de extensão e flexão na ordem dos 20°/s e valores de calibração para os movimentos laterais na ordem dos 10°/s), como também apresenta pouca destreza para efetuar as sequências de movimento necessárias para uma condução contínua do protótipo.

O segundo utilizador (paciente do sexo masculino na casa dos quarenta anos de idade) embora apresentando movimentos de cabeça limitados, não tem qualquer tipo de dificuldade em comandar eficazmente o protótipo durante longos períodos de tempo. Apresenta valores de calibração na ordem dos 50°/s para o movimento de extensão, 30°/s para o movimento de flexão e 40°/s para os movimentos laterais. Não só consegue acionar eficazmente todos os movimentos de condução do protótipo, como consegue alterar o estado do sistema (*Standby/Driving*) e despoletar o envio do alarme de emergência (rotação 360° do protótipo e movimento de flexão superior a três segundos).

O terceiro utilizador (paciente do sexo feminino na casa dos trinta anos de idade) embora apresentando movimentos de cabeça limitados, não tem qualquer tipo de dificuldade em comandar eficazmente o protótipo durante longos períodos de tempo. Apresenta valores de calibração idênticos aos do segundo utilizador e, como este, consegue utilizar o sistema na sua plenitude (condução, alteração de estado e disparo do alarme de emergência).

No fim dos testes realizados, foram efetuadas entrevistas de modo a auscultar o *feedback* de cada utilizador³².

Sendo atuais utilizadores de sistemas como os apresentados no capítulo 2., estrutura com um *joystick* a ser controlado pelo queixo (comando mentoniano) e estrutura de *switches* instalados nas laterais e na retaguarda da cabeça, os pacientes afirmam sentir uma maior satisfação no comando de uma cadeira de rodas através do sistema poRsCHE;

³² Para a recolha dos dados gerais do utilizador (sexo, idade e tipo de limitação física) e de uma captura de imagem e som para testemunho do *feedback* ao sistema, foi cumprido um processo protocolar de modo às três entidades (autor da dissertação, CMRA e paciente) estarem de acordo com as informações a serem trocadas e publicadas.

ao nível ergonómico (movimentos de comando de curta duração que reduzem consideravelmente o stress muscular do pescoço) e ao nível estético (impacto visual bastante menor do que o apresentado pelos equipamentos dos atuais sistemas existentes no mercado). Além de manifestarem satisfação pelas características do sistema ao nível ergonómico, estético e do cuidado presente com o alerta de emergência, os pacientes demonstraram-se com vontade de continuarem a sua participação no projeto de modo a este evoluir para um patamar de produto final.

7.2. Ilacões

O principal objetivo da presente dissertação foi alcançado. Foi implementado com sucesso um sistema embebido *wireless* ergonómico, não intrusivo e com recurso a tecnologias de consumo energético reduzido, que permite que pessoas, com pouca ou nenhuma mobilidade dos membros superiores, possam comandar uma cadeira de rodas elétrica.

Os movimentos de comando implementados agradam a utilizadores e profissionais de saúde não só pela intuitiva utilização como pela melhoria ergonómica face aos implementados nos sistemas atualmente comercializados. O cuidado tido na implementação dos movimentos de comando para que o utilizador não tenha a cabeça, durante mais do que alguns segundos, numa posição que não a vertical, foi referenciado por ambas as partes como uma característica muito positiva. Como referido na secção anterior, após os testes realizados, os utilizadores não deixaram de evidenciar a redução do stress muscular sentido após condução, quando comparado com os seus presentes sistemas.

A reduzida dimensão do equipamento utilizado pelo utilizador, bem como a sua estética, foram alvo de reforço positivo tanto pelos utilizadores como pelos profissionais de saúde. O impacto visual, bastante reduzido quando comparado com os equipamentos dos atuais sistemas, foi alvo de elogio por parte dos utilizadores. Os utilizadores que testaram o sistema demonstraram bastante agrado por poderem controlar uma cadeira de rodas através de um equipamento que passa despercebido aos olhares da sociedade. Entende-se que esta seja uma característica de elevada importância uma vez que melhora consideravelmente a autoestima do utilizador (não se sentindo alvo de olhares, o utilizador passa a ter uma autoestima mais elevada).

Após vários testes ao sistema, utilização do equipamento principal do sistema durante um intervalo de tempo superior a sete meses, a bateria que alimenta o SensorTag apresenta a mesma tensão elétrica do que quando adquirida. Considera-se que, assim como publicitado pelo fabricante, o equipamento que interpreta os movimentos de comando é capaz de ser utilizado durante longos períodos de tempo, períodos superiores a doze meses.

Embora um dos três utilizadores não tenha tido capacidade para comandar o sistema na sua plenitude, considera-se que este esteja implementado para ser comandado por grande parte do público alvo. O segundo e o terceiro utilizadores, embora apresentando limitações na movimentação do pescoço (sequelas dos acidentes que sofreram; lesões vertebro-medulares altas), foram capazes de comandar o sistema na sua plenitude e durante longos períodos de tempo.

Além de elogiarem o sistema poRsCHE pelo seu reduzido impacto visual e pela sua implementação *wireless*, os pacientes e profissionais de saúde do CMRA, elogiaram a implementação da funcionalidade extra que dispara um alarme de emergência após identificação de um desmaio do utilizador. Foi referido pelos profissionais de saúde que uma funcionalidade como esta é bastante importante para que estes pacientes possam

ganhar independência sem, no entanto, perderem cuidados de atenção por parte dos seus familiares e/ou profissionais de saúde.

Não obstante o sistema ainda não estar preparado para comandar uma cadeira de rodas elétrica³³, conseguiu-se uma implementação total a nível de *hardware* sem se alcançar o montante de 80€ para aquisição de todos os equipamentos principais (SensorTag, fita desportiva, RaspberryPi, pen Asus BT-400, sonar e LEDs). Valor baixo quando comparado com os atuais valores para os comandos especiais de cadeiras de rodas elétricas; valores superiores a 700€ para comandos à base de *switches* e superiores a 1600€ para comandos mentonianos [51].

7.3. Trabalho futuro

Após algumas reuniões com os pacientes e profissionais do CMRA, algumas ideias de implementação ficaram propostas para trabalho futuro:

- implementação de um protótipo à escala real (testar o sistema em ambientes de maior dificuldade; espaços públicos e espaços interiores de habitação);
- integração e implementação de um sensor de inclinação para evitar que a cadeira se desloque para locais de perigo de queda;
- implementação de mais do que um contacto de emergência médica (caso um dos contactos não esteja disponível, o alarme não fica perdido; o segundo contacto recebe de igual modo a mensagem de alerta de emergência);
- implementação do envio de coordenadas geográficas na mensagem de alerta de emergência por modo a que os cuidadores do utilizador possam saber onde este se encontra;
- implementação de uma funcionalidade para desligar/ligar o alarme de emergência (funcionalidade que permite ao utilizador conduzir a cadeira de rodas sem disparar o alarme; rotação superior a 360°);
- implementação de outros movimentos de comando para controlar outras funcionalidades da cadeira, como a basculação.

Nesta fase inicial de protótipo, não foram consideradas questões de segurança (confidencialidade dos dados trocados) também porque não existem dados de índole confidencial que sejam trocados; no entanto, para precaver questões de segurança do utilizador, é uma questão a ser avaliada nomeadamente na versão 5.0 do Bluetooth.

Em implementações futuras pode ser considerada a utilização da versão 3 do RaspberryPi; versão que além de ter maior capacidade de processamento, já incorpora um módulo rádio BLE.

³³ Para uma implementação final do sistema, onde este conseguirá comandar uma cadeira de rodas elétrica, é necessário ter acesso ao *software* da cadeira de rodas por forma a se conseguir implementar uma interface que ligue este ao *software* implementado no sistema poRsCHE.

Referências

- [1] Prémio Melo e Castro – Investigação no âmbito das lesões vertebro–medulares, http://www.scml.pt/pt-PT/areas_de_investigacao/lesoes_vertebro_medulares/informacao_geral/
- [2] Catalogo de equipamentos para cadeiras de rodas elétricas da marca Stealth, https://www.stealthproducts.com/pdfs/2014/cat_Product--Catalog.pdf
- [3] Rechy-Ramirez, E. J., HU, H., Mcdonald-Maier, K.: Head movements based control of an intelligent wheelchair in an indoor environment. Em: Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics, Guangzhou, China, 11-14 dezembro. Publicação IEEE 2012 (2012)
- [4] Silva, K. L., Luz, C. M., Coelho, M. F. G., Pareja, V. H. D.: Protótipo de uma Cadeira de Rodas controlada por Movimentos da Cabeça. Projeto Académico, Universidade Federal de Uberlândia, Faculdade de Engenharia Elétrica, Uberlândia, Brasil, (2013)
- [5] Kaur, S., Vashist, C.: Automation of Wheel Chair Using Mems Accelerometer (Adxl330). Em: Advance in Electronic and Electric Engineering, ISSN 2231-1297, vol. 3, nº 2, p.227-232 (2013)
- [6] Christensen, H. V., Garcia, J. C.: Infrared Non-Contact Head Sensor, for Control of Wheelchair Movements. Em: Proceedings of the 8th European Conference for the Advancement of Assistive Technology in Europe. p.336–340 (2005)
- [7] Kumar, N., V.: Two Dimension Head Movements Based Smart Wheelchair Using Accelerometer: Head Motion Can Control Wheel Chair. Em: International Journal of Scientific Engineering and Research (IJSER). julho 2014, ISSN 2347-3878, vol. 2, questão 7, p.9-11 (2014)
- [8] Chen, S.-H., Chen, Y.-L., Kuo, T.-S., Chu, C.-Y., Hung, C.-N.: M3S-based Electrical Wheelchair with Head-controlled Device. Em: 28th IEEE EMBS Annual International Conference, New York City, USA, 30 agosto – 3 setembro 2006, p.4917–4920. Publicação IEEE 2006 (2006)
- [9] Dokic, B., Pajkanovic, A.: Wheelchair Control by Head Motion. Em: Serbian Journal of Electrical Engineering. fevereiro 2013, vol. 10, nº 1, p.135-151 (2013)
- [10] Fusco, D. A.: Acionamento de uma Cadeira de Rodas através de um Acelerômetro Bi-Axial como Inclinômetro. Projeto de Diplomação, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil (2010)
- [11] Brito, P., J., I., Sivarasu, S., Mathew, L.: Automation of Wheel Chair using Ultrasonics and Body Kinematics. Em: National Conference on Computational Instrumentation CSIO, Chandigarh, INDIA, 19-20 março 2010, p.37-43. Publicação CSIO 2010 (2010)
- [12] Reis, L. P., Braga, R. A. M., Sousa, M., Moreira, A. P.: IntellWheels MMI: A Flexible Interface for an Intelligent Wheelchair. Projeto Académico, Universidade do Porto, Porto, Portugal (2006)
- [13] Braga, R. A. M., Petry, M., Reis, L. P., Moreira, A. P.: IntellWheels: Modular development platform for intelligent wheelchairs. Em: Journal of Rehabilitation Research & Development. vol. 48, nº 9 (2011)
- [14] Braga, R. A. M., Petry, M., Moreira, A. P., Reis, L. P.: INTELLWHEELS - A Development Platform for Intelligent Wheelchairs for Disabled People. Em: International Conference on Informatics in Control, p.115-121, 2008. Publicação ICINCO 2008 (2008)

- [15] Saad, C., Mostafa, B., Cheikh, E. A., Abderrahmane, H.: Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Applications. Em: International Journal of Advanced Computer Science and Applications, vol. 5, nº 4, p.76-85 (2014)
- [16] Lee, J.-S., SU, Y.-W., Shen, C.-C.: A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee and Wi-Fi. Em: The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Taipei, Taiwan, 5-8 novembro 2007, p.46-51 (2007)
- [17] Dementyev, A., Hodges, S., Taylor, S., Smith, J.: Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario. Em: Microsoft Research, Cambridge, United Kingdom <http://research.microsoft.com/pubs/192688/IWS%202013%20wireless%20power%20consumption.pdf> (2013)
- [18] Siekkinen, M., Hienkari, M., Nurminen, J. K., Nieminen, J.: How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4. Em: WCNC 2012 Workshop on Internet of Things Enabling Technologies, Embracing Machine-to-Machine Communications and Beyond. setembro 2012, p.232–237. Publicação IEEE 2012 (2012)
- [19] Omre, A. H.: Bluetooth Low Energy: Wireless Connectivity for Medical Monitoring. Em: Journal of Diabetes Science and Technology. março 2010, vol. 4, nº 2, p.457-463 (2010)
- [20] Townsend, K., Cufí, C., Davidson, A. & R.: *Getting Started with Bluetooth Low Energy – tools and techniques for low-power networking*. Primeira edição. California: O'Reilly Media. 2014. ISBN 978-1-491-94951-1
- [21] Heydon, R.: *Bluetooth Low Energy – The Developer's Handbook*. Terceira edição. Crawfordsville, Indiana: Prentice Hall. 2014. ISBN 978-0-13-288836-3
- [22] Get Connected with the Future Connectivity Solutions – Taking the World Wireless, http://www1.futureelectronics.com/Mailing/FCS/PublicationDownload/FCS_GetConnected.pdf
- [23] Gomez, C., Oller, J., Paradells, J.: Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Em: Sensors – Open Access Journal. 2012, vol. 12, nº 9, p.11734-11753, <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3478807/>
- [24] Bluetooth Low Energy GATT – Technology Overview, <https://developer.bluetooth.org/TechnologyOverview/Pages/GATT.aspx>
- [25] DigiKey ELECTRONICS – Texas Instruments Bluetooth Smart, http://dkc1.digikey.com/us/en/TOD/Texas_Instruments/BluetoothSmart/BluetoothSmart.html
- [26] Bluetooth SIG – Bluetooth Smart (Low Energy) Technology, <https://developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx>
- [27] Texas Instruments - SensorTag CC2650, http://www.ti.com/ww/en/wireless_connectivity/sensortag2015/tearDown.html
- [28] InvenSense – MPU-9250, <http://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/>
- [29] IMUDUINO, <http://femto.io/products/imduino>
- [30] Texas Instruments - SensorTag CC2541, <http://www.ti.com/tool/cc2541dk-sensor>
- [31] Fita desportiva K-Open, <https://www.sportzone.pt/desporto/mais-desportos/tenis/acessorios>
- [32] Arduino, <https://www.arduino.cc/>

- [33] RaspberryPi, <https://www.raspberrypi.org/>
- [34] PTROBOTICS – Arduino Uno, <http://www.ptrobotics.com/plataformaarduino-e-modelos-alternativos-equivalentes/3987-genuino-uno-r3.html>
- [35] PTROBOTICS – RaspberryPi Model B+, <http://www.ptrobotics.com/raspberrypi/2863-raspberrypi-model-b-plus.html>
- [36] ASUS – Adaptador USB Bluetooth 4.0, <https://www.asus.com/pt/Networking/USBBT400/>
- [37] PCDiga – Asus BT400, <https://www.pcdiga.com/2/14703/Adaptador-Bluetooth-4-0-Asus-USB-BT400-90IG0070-BW0600>
- [38] MICROCHIP RN4020, <http://www.microchip.com/wwwproducts/en/RN4020>
- [39] Futaba S3104, <http://www.servodatabase.com/servo/futaba/s3104>
- [40] Sonar HC-SR04, <http://www.micropik.com/PDF/HCSR04.pdf>
- [41] Texas Instruments – LM7805C, <http://www.ti.com/product/LM7805C>
- [42] RaspberryPi – Power SUPPLY, <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>
- [43] RaspberryPi – GPIO, <https://www.raspberrypi.org/documentation/usage/gpio/>
- [44] Yurish, S. Y.: *Sensors and Biosensors, MEMS Technologies and its Applications*. Segunda edição. Barcelona: International Frequency Sensor Association Publishing. 2013. ISBN 978-84-616-4154-3
- [45] Nihtianov, S., Luque, A.: *Smart sensors and MEMS – Intelligent devices and microsystems for industrial applications*. Primeira edição. Philadelphia: Woodhead Publishing. 2014. ISBN 978-0-85709-929-7
- [46] InvenSense – Movement Sensors, <https://www.youtube.com/watch?v=C7JQ7Rpwn2k>
- [47] HowToMechatronics – MEMS sensors, <http://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>
- [48] SensorWiki – Gyroscope, <http://www.sensorwiki.org/doku.php/sensors/gyroscope>
- [49] PCWorld – Android leads in Gartner's smartphone sales report, <http://www.pcworld.com/article/3035100/phones/android-leads-and-windows-phone-fades-in-gartners-smartphone-sales-report.html>
- [50] Lourenço, J., Almeida, A. P.: poRsCHE: Remote Control via Head movEs. Em: INForum – Simpósio de Informática, Lisboa, Portugal, 8 e 9 novembro 2016, artigo nº 10 (2016)
- [51] Catalogo de equipamentos para cadeiras de rodas elétricas da marca Invacare, http://www.invacare.pt/sites/pt/files/product_documents/d006f17266d125283acc10c40248be3f30ce8f19_d12084144f0a3c88562da96d861c4b95.pdf

Anexo

poRsCHE: Remote Control via Head movEs

José Lourenço¹, Alexandre Passos Almeida²

¹Departamento de Ciências e Tecnologias da Informação,
ISCTE-IUL, Lisboa, Portugal, jappl@iscte-iul.pt

²Instituto de Telecomunicações and Instituto Universitário de Lisboa
(ISCTE-IUL), Lisboa, Portugal

Abstract. We present an ergonomic, nonintrusive and low power Embedded System which allows tetraplegic people to control an electric wheelchair by using their head movements. These command movements are implemented by taking into account people's physical limitations. The main sensing device is installed in a low weight board attached to the user head via an elastic band. Wireless communication is implemented with Bluetooth 4.0 (Bluetooth Low Energy) which assures an ergonomic (non-existence of cables) and a low power battery operated system. A smartphone connection provides extra functionality which triggers an emergency contact call after identifying a user fainting condition.

Keywords: Tetraplegia; Remote Control; Head Moves; Wheelchair; Low Energy; SensorTag; Sensor System; Faint Alarm.

1 Introduction

Accidents and diseases which limits people's mobility occur and are diagnosed every day. The standard electric wheelchairs give some autonomy to people who lost their legs mobility. However, they are not prepared for people who also have reduced mobility in their arms.

Despite all kinds of accidents which cause paraplegia or tetraplegia, there are also some diseases and pathological conditions [1] which reduce people's mobility. Some are as follows:

- lupus;
- multiple sclerosis;
- changes in vascular function, which may lead to stroke;
- tumours;
- infections;
- deformities;
- degenerative or compressive processes.

Standard wheelchairs are moved by electric motors and controlled by joysticks. This implementation is effective for people who lost legs mobility but it is not effective for people who have also lost mobility in their arms. People with reduced arms mobility are not able to manage the joystick in a way to efficiently control a wheelchair.

To surpass the above limitation, we propose a wireless head movement system. By using a microcontroller with a gyroscope sensor installed on the user's head, the system interprets the user's head movements and sends appropriate information to the electric motors so the wheelchair can move as the user wants to.

Some studies, to be referred in the next section, have already been done but for some reasons, either cost and/or ergonomics, they have not been widely adopted.

To implement a better system some questions, at the qualitative level, need to be addressed:

- Which wireless protocol is the best to ensure low energy battery consumption?
- Which movement sensor equipment is the best to detect the user's head movements?
- How to install the movement sensor equipment in the user's head?
- Which are the most efficient and ergonomic movements for users to control the wheelchair?
- How to implement an emergency contact?

To answer the questions above and further try to define quantitative metrics, we studied previous work in this area, conducted our own experiments and in addition talked to the target audience at a rehabilitation centre. We conducted meetings and interviews with professionals and users of *Centro de Medicina de Reabilitação de Alcoitão* (CMRA), one of the best institutions on the Rehabilitation Medicine area.

The rest of the paper is organized as follows: section 2 presents some related work, the choice of the wireless protocol, the choice of the movement sensor equipment and the definition of the implementation goals; section 3 explains the hardware architecture and the software design of the poRsCHE system; section 4 provides some experimental tests and result discussion; and section 5 presents the final conclusions and discuss future work.

2 State of the Art

In the current wheelchair market, the special command equipments built to help tetraplegic people are either bulky or non-ergonomic. As depicted in figure 1, they can be big structures of switches installed on the sides of the user's head and/or big structures with a joystick to be managed by the user's mouth [2].



Fig. 1. Special command equipments used in the current wheelchair market [2]

In the recent years, some scientific projects have been done to try to surpass these limitations. Some have implemented systems controlled by head movements [3-10], some by eye movements [11] and others have been implemented using more than one type of command control (Multimodal Interfaces – movement sensors installed on the user’s head, webcam to recognize facial expressions, microphone to capture voice commands, etc.) [12-14].

In general, all project’s authors claim that the systems were successful. However, some limitations are easily identified.

In Multimodal Interfaces projects, the voice commands were not implemented with success. As said by the authors, the noise environment did not allow a 100% interpretation of the control commands [3-4], [12]. In project [5], the equipment to detect the head motion is relatively expensive. In project [11], the use of a shoulder sensor to detect a faint of the user is considered intrusive. And in projects [3-4], [8], [10-12], [14], the amount of equipment and/or the use of cables to connect the system components is considered non-ergonomic.

3 Available Technology

3.1 Wireless Protocol

To evaluate which wireless protocol is the best to ensure efficiency and low power consumption, eight protocols were considered: UWB (Ultra-Wide Band), Wi-Max, Wi-Fi, Bluetooth, ZigBee, GSM/GPRS, BLE (Bluetooth Low Energy) and ANT.

Comparing the first six protocols, we agree that Bluetooth is the most energy efficient [15-16]. For the same amount of exchanged data, Bluetooth spends less energy than the others. In figure 2 we illustrate the energy efficiency difference between the protocols.

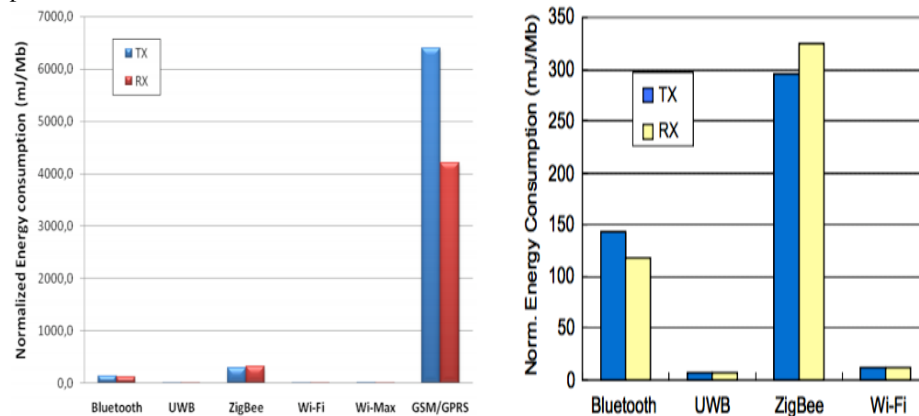


Fig. 2. Energy efficiency presented by the protocols Bluetooth, UWB, ZigBee, Wi-Fi, Wi-Max and GSM/GPRS [15-16]

The two remaining protocols, BLE and ANT, were also addressed [17-18]. It is clear that the best protocol for low payload and short-range communications is the BLE. This protocol not only presents less energy consumption transmitting and receiving data but also presents a protocol architecture that puts the equipments in a standby mode when

they do not have data to share. In figure 3 we illustrate the reduced power used from the BLE taking into account different standby mode periods.

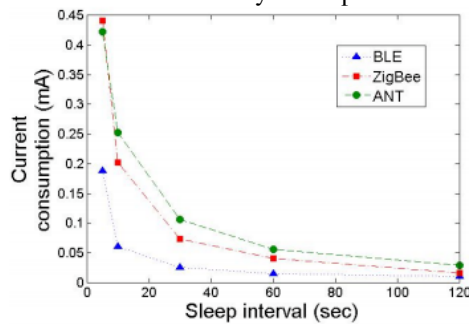


Fig. 3. Current consumption of the BLE protocol with different standby mode periods [17]

Based on our initial evaluation, we chose BLE as the protocol to be used in our system. This protocol besides being the best to ensure efficiency and low energy consumption, gives to the system a direct link to mobile communications. Connecting sensors to equipments like tablets and smartphones via BLE, the system acquires a direct link to mobile and/or internet communications. Through this communication link, the system becomes able to trigger a fast and efficient alarm [19], which in the case of the poRsCHE system takes the form of triggering a faint alarm by sending an SMS from a smartphone (as explained in section 4.2).

3.2 Movement sensor equipment

In the current electronic market place only two equipments are built with the desired features: a BLE radio module and a MEMS gyroscope sensor¹.



Fig. 4. Movement Sensor Equipments (IMUduino and SensorTag) [20-21]

The first equipment, presented in figure 4, is named IMUduino, has a BLE radio module from Nordic (nRF8001), a MEMS (Gyroscope/Accelerometer) sensor from InvenSense (MPU6050) and is very small (39,8mm x 15,72mm) and light (2,7grams) [20].

The second one is named SensorTag, has a BLE radio module (TI CC2650), a MEMS (Gyroscope/Accelerometer/Magnetometer) sensor from InvenSense (MPU9250), a protective cover made from rubber, reduced dimensions (5cm x 6,7cm), a battery module powered by a cell coin (CR2032) and it is also very light (30grams) [21].

Taking into account the manufacturer reputation, the existence of a battery module and of a protective cover (cover that makes easier the equipment installation in a piece

¹ Micro-Electro-Mechanical Gyroscope – nano-scale gyroscope sensor made from silicon technology.

of clothing or object used in the user's head; like a hair band, a cap or glasses), we chose this solution (SensorTag) for the sensor module of our system. This equipment not only gives to the system a better hardware implementation but also a reduced price when compared to the other alternative.

3.3 Implementation Goals

In our project we identified some crucial features to the implementation of the poRsCHE system:

- The system processor (equipment that interprets the user's head movements and drives the wheelchair) is a microcontroller. The processing capacity of a microcontroller is considered enough, and its dimensions and power consumptions are considered better than the ones from a personal computer.
- A wireless system is considered beneficial to the system ergonomics. The existence of cables in the wheelchair architecture is considered negative because it reduces user's mobility.
- To be intuitive, the head movement commands are implemented as the joystick commands. To drive the wheelchair forward, the user will have to lean his head toward the chest; to rotate the wheelchair, the user will have to lean his head to the desired side; to stop the wheelchair the user will have to lean his head backward.
- To provide user personalization, the system has a calibration configuration that defines the maximum user's head inclination. The inclination values configured will be used to interpret the user's head movements to command the wheelchair drive.
- A safety function to interpret a faint or an indisposition of the user is considered important to this type of system. A system that offers some autonomy to tetraplegic people must have functions that pay attention to his health and welfare.
- To add an extra degree of safety, an inner system that detects obstacles is implemented. To avoid collisions, the system has sonars that will detect fix or mobile obstacles.

All points presented in this section will be further described in section 4.

4 Implementation

4.1 Hardware Architecture Design

The poRsCHE system is implemented with the following equipment set: one SensorTag (movement sensor to be installed on user's head), one smartphone (user's smartphone with the poRsCHE mobile application installed) and one microcontroller board (microcontroller which interprets the user's head movements to command the wheelchair drive as the user wants to).

The first component was presented in the previous section. It has the gyroscope sensor to detect the user's head movements and, in this first implementation of the system, it is installed on the user's head with a hair band. The proposed implementation is illustrated in figure 5.



Fig. 5. SensorTag attached to a hair band

The second component is the user's own smartphone. Taking advantage of the large use of smartphones, the poRsCHE system is implemented with a mobile application that offers an easy GUI (Graphical User Interface) to calibrate the head movement commands and provides communication (Short Message Service communication protocol) to trigger a faint alarm. In this first implementation of the system, the mobile application is implemented on the Android platform.

The third component is a microcontroller that operates as the poRsCHE system processor (interpretation of the user's head movements and control of the wheelchair motors). For these functions the equipment chosen is the RaspberryPi model B+ with a BLE radio module from ASUS (ASUS BT400) as illustrated in figure 6.

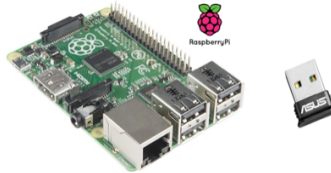


Fig. 6. System processor: RaspberryPi model B+ with the BLE radio module ASUS BT400

To test the system, without danger to the user, a mock-up wheelchair prototype is built with a PVC plate, two driving wheels powered by two servos, two free rotating wheels, one sonar to detect the obstacles and three LEDs to signal the system feedback info (one LED to signal the standby state, another to signal the driving state and a last one to signal the availability to receive a new command). The proposed implementation is illustrated in figure 7.

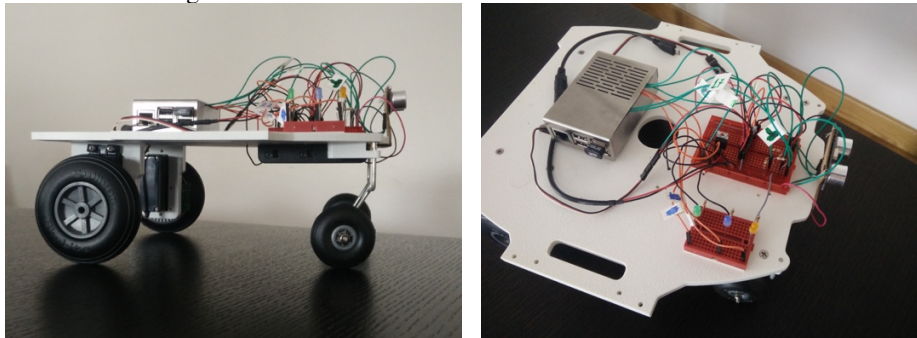


Fig. 7. poRsCHE wheelchair prototype

4.2 Software Design

The poRsCHE system has implemented three operational periods (setup, driving and powered off). During the driving state the system analyses the user's head movements comprising four types (forward, backwards, right and left). We first describe these head movements and later we talk about the operational periods.

Movement Commands

To assure a commitment between an efficient control of the wheelchair and ergonomics head movements, four head movement commands are implemented.

To drive the wheelchair forward, the user has to lean his head toward the chest and return it to the natural rest position (vertical position). This head movement command is illustrated in figure 8 by (left) a body scheme and (right) a data plot from the gyroscope sensor.

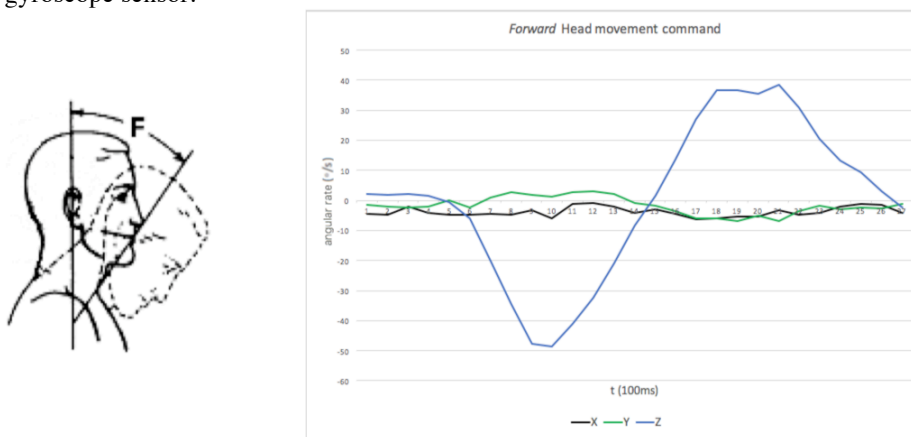


Fig. 8. Forward head movement command

The first parabola represents the lean of the head to the chest and the second one its return to the natural position. If the wheelchair is stopped, it will move forward in a slower speed (first gear); if the wheelchair is moving forward, it will move faster (second gear).

To stop or reduce the speed of the wheelchair, the user has to lean his head to the back and return it to the natural position. This head movement command is illustrated in figure 9.

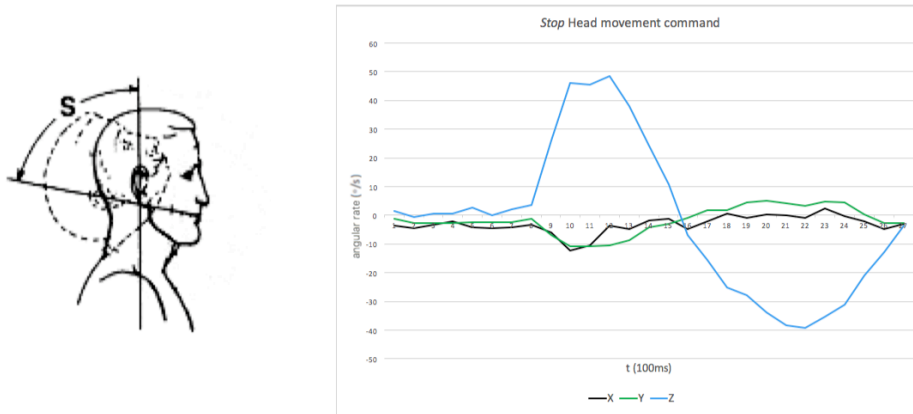


Fig. 9. Stop head movement command

The first parabola represents the lean of the head to the back and the second one its return to the natural position. If the wheelchair is in a second gear, it will pass to the first one; if the wheelchair is moving in the first gear, it will be stopped. If the wheelchair is stopped and if the user does this movement two times consecutively, the system goes into a standby mode (free movements – the user can move his head without sending commands to the processor); to return to the driving mode, the user has to do the same two movements.

To rotate the wheelchair, the user has to lean his head to the desired side until it is in the intended position. When the wheelchair is in the intended position, the user returns his head to the natural position. This head movement command is illustrated in figure 10.

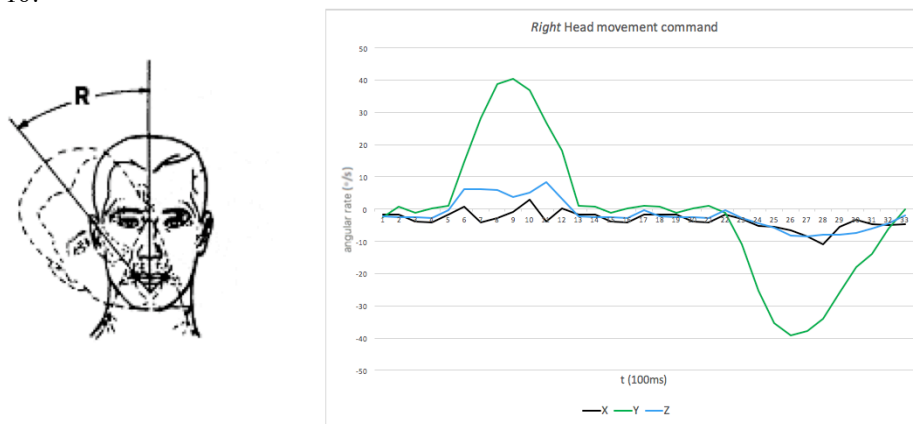


Fig. 10. Right head movement command

The first parabola represents the lean of the head to the right shoulder and the second one its return to the natural position. The left command has the same logic. The graphic interpretation is in the same axis but the values are reversed (this first parabola is negative and the second one is positive).

Once powered on

Once powered on, the system has three operational periods. A setup period, managed by the mobile application; a driving period, managed by the user's head movements; and a powered off period, triggered by a power-off command done by the user's helper (second person) or by a user's faint (faint condition detected by the system).

Before the user is able to drive the wheelchair, he has to configure the system, with the help of a second person. The system configuration phase is divided in three parts.

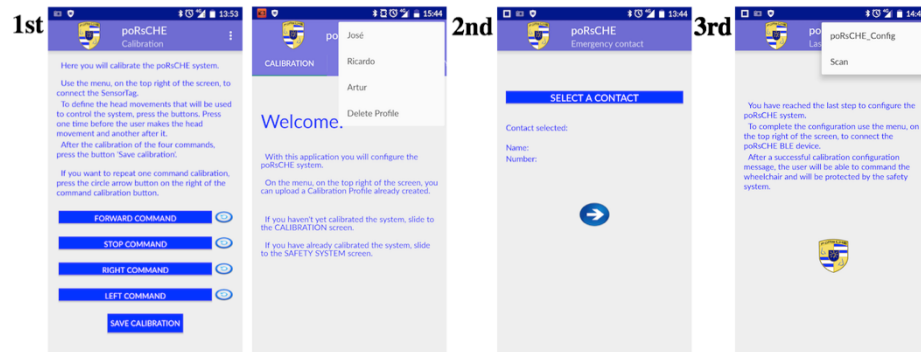


Fig. 11. poRsCHE Android application

First, the user has to calibrate the movement commands. The calibration of the commands is done following the instructions presented on a proper GUI (connection of the SensorTag to the smartphone and calibration procedure by head movements to identify the maximum amplitude of the movements). The referred GUI is illustrated in figure 11.

After a first calibration of the system, the user can upload his calibration values by selecting his name on the menu presented in the upper right corner of the Welcome layout (second layout illustrated in figure 11).

Second, the user's helper has to select an emergency contact, the contact which will receive a SMS text if the user faints or is indisposed (third layout illustrated in figure 11).

Third, the user's helper has to send the calibration values to the wheelchair equipment (microcontroller RaspberryPi). This communication is done just by selecting the wheelchair poRsCHE equipment name "poRsCHE_Config" (fourth layout illustrated in figure 11).

After completed the poRsCHE system calibration, the user is able to control the wheelchair drive by his head movements.

To interpret the user's head movements, the system processor does the following interpretation:

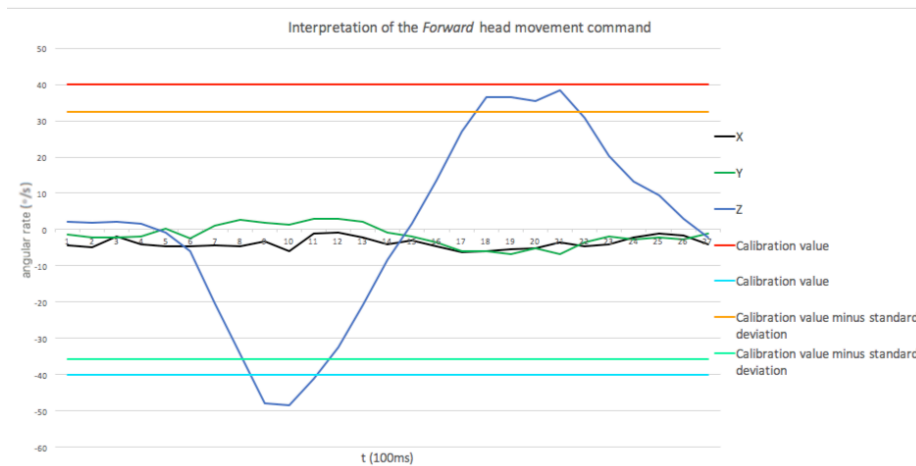


Fig. 12. Interpretation of a head movement command

- Forward/Stop movement command – If the user returns his head to the natural position in a period of time up to three seconds, the command is triggered. This happens when the sensor measurement value surpasses the lower or higher bands that were defined by the calibration procedure (maximum movement amplitude minus standard deviation²). Else, the faint alarm is triggered.
- Right/Left movement command – If the user returns his head to the natural position in a period of time smaller than the necessary to rotate the wheelchair 360°, the wheelchair rotates normally. Else, if the wheelchair rotates a full 360°, the faint alarm is triggered.

After the wheelchair drive, two situations can occur. If the user’s helper presses the power on/off SensorTag button, the whole system is powered off. If the system interprets a user’s faint, the wheelchair stops, the system is powered off and the sending of the poRsCHE alarm SMS is triggered.

4 Experiments and Results

The poRsCHE system was tested by three tetraplegic patients, all with spinal cord injuries. The first one had very short range head movements, and for that reason was not capable to drive the prototype successfully. The other two, even with reduced head movements, were capable to drive the prototype successfully, with all head movement commands.

Looking into the calibration values of the users, we easily identified the limitations of the first user. While the second user presented calibration values around 50°/s for the stop command, 30°/s for the forward command and 40°/s for the sides commands, the first user only presented calibration values around: 20°/s for the stop command, 10°/s for the forward command and 10°/s for the sides commands.

² The standard deviation was calculated from ten calibrations made by a non-tetraplegic person. A non-tetraplegic person has an extended range of head travel movements.

To evaluate the degree of satisfaction from the target users we conducted in person interviews to find out the merits and demerits of our system.

One of the users who tested the prototype successfully, did, not only, drive the prototype in the four directions but also changed the operation state (driving state and standby state) and triggered the faint alarm successfully. Previously this user was using a joystick manipulated via his mouth to control the wheelchair. After using our prototype, the user manifested a much higher degree of satisfaction.

The other test user mentioned the lower weight of our head piece which was crucial to reduce the stress on the neck muscles that she was feeling after extended periods of usage of her previous control equipment. Moreover, she also referred that the visual appearance of the head piece greatly reduced the psychological impact that she felt from contacting outside observers.

5 Conclusions and Future Work

The prototype system was well accepted by the health professionals and patients of CMRA. They not only congratulated the ergonomics of the wireless head movement system, but also the inclusion of the faint alarm via SMS, which assure safety considerations to the user.

Relating future work we are considering:

- an implementation of more than one emergency contact (if one phone is not working properly, another one will be contacted);
- the implementation of a function to power on/off the faint alarm system (feature to give to the user the possibility to drive the wheelchair without triggering the faint alarm, rotation higher than 360°);
- the implementation of head movement commands to control other wheelchair functions like basculation and reverse gear mode.

References

1. Melo e Castro Award, Research within the spinal cord injuries, http://www.scml.pt/pt-PT/areas_de_investigacao/lesoes_vertebro_medulares/informacao_geral/
2. Catalog of wheelchair products from the brand Stealth, https://www.stealthproducts.com/pdfs/2014/cat_Product--Catalog.pdf
3. Rechy-Ramirez, E. J., HU, H., Mcdonald-Maier, K.: Head movements based control of an intelligent wheelchair in an indoor environment. In: Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics, Guangzhou, China, 11-14 December. Publication IEEE 2012 (2012)
4. Silva, K. L., Luz, C. M., Coelho, M. F. G., Pareja, V. H. D.: Protótipo de uma Cadeira de Rodas controlada por Movimentos da Cabeça. Academic Project, Federal University of Uberlândia, Electricity Engineer College, Uberlândia, Brasil (2013)
5. Kaur, S., Vashist, C.: Automation of Wheel Chair Using Mems Accelerometer (Adx1330). In: Advance in Electronic and Electric Engineering, ISSN 2231-1297, vol. 3, n° 2, pp. 227-232 (2013)
6. Christensen, H. V., Garcia, J. C.: Infrared Non-Contact Head Sensor, for Control of Wheelchair Movements. In: Proceedings of the 8th European Conference for the Advancement of Assistive Technology in Europe. pp. 336–340 (2005)

7. Kumar, N., V.: Two Dimension Head Movements Based Smart Wheelchair Using Accelerometer: Head Motion Can Control Wheel Chair. In: International Journal of Scientific Engineering and Research (IJSER). July 2014, ISSN 2347-3878, vol. 2, issue 7, pp. 9-11 (2014)
8. Chen, S.-H., Chen, Y.-L., Kuo, T.-S., Chu, C.-Y., Hung, C.-N.: M3S-based Electrical Wheelchair with Head-controlled Device. In: 28th IEEE EMBS Annual International Conference, New York City, USA, 30 Aug.–3 Sept. 2006 pp. 4917–4920. Publication IEEE 2006 (2006)
9. Dokic, B., Pajkanovic, A.: Wheelchair Control by Head Motion. In: Serbian Journal of Electrical Engineering. February 2013, vol. 10, n° 1, pp. 135-151 (2013)
10. Fusco, D. A.: Acionamento de uma Cadeira de Rodas através de um Acelerômetro Bi-Axial como Inclinômetro. Graduation Project, Federal University of Rio Grande do Sul, Porto Alegre, Brasil (2010)
11. Brito, P., J., I., Sivarasu, S., Mathew, L.: Automation of Wheel Chair using Ultrasonics and Body Kinematics. In: National Conference on Computational Instrumentation CSIO, Chandigarh, INDIA, 19-20 March 2010, pp. 37-43. Publication CSIO 2010 (2010)
12. Reis, L. P., Braga, R. A. M., Sousa, M., Moreira, A. P.: IntellWheels MMI: A Flexible Interface for an Intelligent Wheelchair. Academic Project, Porto University, Porto, Portugal (2006)
13. Braga, R. A. M., Petry, M., Reis, L. P., Moreira, A. P.: IntellWheels: Modular development platform for intelligent wheelchairs. In: Journal of Rehabilitation Research & Development. vol. 48, n° 9 (2011)
14. Braga, R. A. M., Petry, M., Moreira, A. P., Reis, L. P.: INTELLWHEELS - A Development Platform for Intelligent Wheelchairs for Disabled People. In: International Conference on Informatics in Control, pp. 115-121, 2008. Publication ICINCO 2008 (2008)
15. Saad, C., Mostafa, B., Cheikh, E. A., Abderrahmane, H.: Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Applications. In: International Journal of Advanced Computer Science and Applications, vol. 5, n° 4, p. 76-85 (2014)
16. Lee, J.-S., SU, Y.-W., Shen, C.-C.: A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee and Wi-Fi. In: The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Taipei, Taiwan, 5-8 November 2007, p.46-51 (2007)
17. Dementyev, A., Hodges, S., Taylor, S., Smith, J.: Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario. In: Microsoft Research, Cambridge, United Kingdom <http://research.microsoft.com/pubs/192688/TWS%202013%20wireless%20power%20consumption.pdf> (2013)
18. Siekkinen, M., Hienkari, M., Nurminen, J. K., Nieminen, J.: How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4. In: WCNC 2012 Workshop on Internet of Things Enabling Technologies, Embracing Machine-to-Machine Communications and Beyond. September 2012 pp. 232–237. Publication IEEE 2012 (2012)
19. Omre, A. H.: Bluetooth Low Energy: Wireless Connectivity for Medical Monitoring. In: Journal of Diabetes Science and Technology. March 2010, vol. 4, n°2, p.457-463 (2010)
20. IMUDUINO, movement sensor equipment, <http://femto.io/products/imduino>
21. SensorTag CC2650, movement sensor equipment, http://www.ti.com/ww/en/wireless_connectivity/sensortag2015/teardown.html