



Departamento
de Engenharia Electrotécnica

**FERRAMENTA PARA O DESENVOLVIMENTO DE SISTEMA DE ANÁLISE E MONITORIZAÇÃO
PARA A MODALIDADE DA CANOAGEM DE VELOCIDADE**

Dissertação apresentada para a obtenção do grau de Mestre em Engenharia Electrotécnica

Autor

Ana Claudia Sousa Alves

Orientador

Professor Doutor José Pedro M. N. Amaro

Instituto Superior de Engenharia de Coimbra

Coimbra, Junho, 2016

AGRADECIMENTOS

Agradeço à minha mãe, ao meu pai, ao meu avô e à minha irmã Joana.

Um especial e maior agradecimento a ti Lara. Com a promessa de que a partir de agora o tempo é nosso e para as nossas brincadeiras.

Pelo companheirismo nesta viagem um obrigada ao Pedro Amaro.

Este documento não está escrito de acordo com o Novo Acordo Ortográfico.

RESUMO

É um desafio determinar a pagaiada perfeita e perceber os dados adquiridos. O processo torna-se mais complexo se a informação for adquirida no ambiente natural da prática desportiva. A análise dos parâmetros da canoagem começou por ser feita em ambientes controlados no entanto, a monitorização de atletas na água, facilitada pelo progresso da microelectrónica, tem conduzido a estudos mais realistas e viáveis.

O propósito deste trabalho é apresentar uma ferramenta para desenvolver um sistema de aquisição e avaliação do desempenho de um atleta de canoagem. Mantendo este propósito em mente e, de acordo com uma rede sensorial sem fios, foi pensada uma arquitectura que permite a aquisição de dados da pagaia, do barco e do finca-pés. Na rede definida para adquirir e transferir os dados dos sensores em tempo real é necessário um protocolo de baixo alcance, o Bluetooth™ Low Energy (BLE). Por outro lado, a comunicação a longas distâncias serve para levar os pacotes de dados desde o barco até à margem do rio, onde se encontra o dispositivo central de análise, o receptor.

Neste trabalho, é apresentada a arquitectura e o conceito dos módulos da ferramenta para o sistema de monitorização. São igualmente demonstrados os resultados adquiridos em provas de conceito assim como um estudo e análise do protocolo BLE. O trabalho é finalizado com a apresentação de uma aplicação C# que ilustra o conceito e a implementação do protocolo BLE.

Palavras-Chave: canoagem, monitorização desportiva, acelerómetro, giroscópio, magnetómetro, sensor de pressão, rede sensorial sem fios, comunicação a longa distância, Bluetooth™ Low Energy, SensorTag, C#

ABSTRACT

It is challenging to determine the perfect paddling technique and the clarification of the acquire data is slightly difficult. It becomes more complex if the information is attained in the natural training environment, since it introduces a number of considerations that must be added. Monitoring athletes in water, leads to a better and more real understanding and has been facilitated by some progresses in microelectronics.

The purpose of this paper is to present a tool to develop a system of acquisition and evaluation of the performance of a canoeing athlete. Keeping this in mind and using a functional architecture based on Wireless Sensor Network (WSN) technology, data can be acquired from the paddle, boat and foot stands. The WSN nodes entail short range connectivity that is reliable for on-boat sensor fusion. The Bluetooth™ Low Energy protocol is needed to acquire and transfer data in real time. On the other hand, long range communications are used to bring data packets from the boat to the margin of the river where is the central device analysis, the receptor.

This work presents the architecture and the concept of the tool modules for the monitoring system. The results acquired in proofs of concept as well as a study and analysis of BLE protocol are also demonstrated. The work is concluded with a presentation of a C # application that illustrates the concept and implementation of the BLE protocol.

Key words: canoeing, sports monitoring, accelerometer, gyroscope, magnetometer, pressure sensor, WSN, Long range communication, Bluetooth™ Low Energy, SensorTag, C#

ÍNDICE

AGRADECIMENTOS	3
RESUMO	5
ABSTRACT	6
ÍNDICE DE FIGURAS	9
ÍNDICE DE QUADROS	12
SIMBOLOGIA E ABREVIATURAS	12
SIMBOLOGIA	12
ABREVIATURAS	13
CAPÍTULO 1 - INTRODUÇÃO	15
1.1. UMA VIAGEM NO TEMPO	15
1.1.1. O impacto da tecnologia no desempenho desportivo	15
1.2. FORMULAÇÃO DO PROBLEMA	17
1.3. MOTIVAÇÃO	18
1.4. ESTRUTURA DA DISSERTAÇÃO	19
CAPÍTULO 2 - ESTADO DA ARTE	21
CAPÍTULO 3 – A CANOAGEM E O SISTEMA PROPOSTO	25
3.1. SPRINT – VELOCIDADE, FORÇA E TÉCNICA	25
3.2. ESPECIFICAÇÕES DO SISTEMA	28
CAPÍTULO 4 - PROTOCOLOS DE COMUNICAÇÃO	31
4.1. RÁDIO FREQUÊNCIA, ZIGBEE™	31
4.2. BLUETOOTH™ LOW ENERGY	33
4.3. ZIGBEE ou BLUETOOTH™ LOW ENERGY?	44
CAPÍTULO 5 - ARQUITECTURA PROPOSTA PARA O SISTEMA DE MONITORIZAÇÃO	47
5.1. ARQUITECTURA PROPOSTA	47
5.2. VERSÃO 1.0.	48
5.2.1. Módulo da Pagaia	48
5.2.2. Módulo do barco	52
5.2.3. Comunicação RF para curtas distâncias e comunicação a longas distâncias	54
5.2.4. Resultados da implementação da versão 1.0.	56
5.2.5. Módulo do finca-pés	63
5.3. VERSÃO 2.0.	64
CAPÍTULO 6 – O SENSORTAG E O BLUETOOTH LOW ENERGY	67
6.1. BTOOL	67

6.2. EVENTOS E COMANDOS HCI ESPECÍFICOS DA TEXAS INSTRUMENTS™	68
6.3. PERCEBER O BLE ATRAVÉS DO SENSORTAG E DA BTOOL	72
CAPÍTULO 7 - PLATAFORMA DE COMUNICAÇÃO E VISUALIZAÇÃO DE DADOS	79
7.1. A APLICAÇÃO	79
7.2 O FUNCIONAMENTO	82
7.2.1. Configuração.....	82
7.2.2. Aquisição de dados	86
CAPÍTULO 8 - CONCLUSÕES	91
8.1. CONCLUSÕES.....	91
8.2. SUGESTÕES PARA TRABALHO FUTURO	92
REFERÊNCIAS BIBLIOGRÁFICAS	93

ÍNDICE DE FIGURAS

Figura 1: Análise de um ciclo de uma pagaia. Análise das sub-fases: 1 - momento de entrada na água; 2 – momento de arrastamento da pagaia dentro de água, 3 – momento de saída da pagaia da água.	26
Figura 2: Principais forças existentes numa pagaia.	27
Figura 3: Principais factores que influenciam o resultado de uma competição.	28
Figura 4: Arquitectura do protocolo de comunicação ZigBee™.	33
Figura 5: Arquitectura de um dispositivo <i>low energy</i> . [34]	35
Figura 6: Frequency Hopping. [35]	36
Figura 7: Comportamento de um advertiser no Protocolo BLE.[35]	36
Figura 8: Scanner e advertiser. [35]	37
Figura 9: Diagrama de estados para obter uma rede BLE. [35]	37
Figura 10: Arquitectura de ligação do sistema proposto no BLE.	38
Figura 11: Cadeia de acontecimentos quando o intervalo de ligação e o tempo de latência do periférico são diminuídos/aumentados.	39
Figura 12: Formato de pacote de um comando HCI. [36]	40
Figura 13: Formato de pacote de um evento HCI do Bluetooth™ (em cima); Formato de pacote de um evento HCI do BLE (em baixo). [36]	40
Figura 14: Tabela de atributos do SensorTag com identificação dos atributos e propriedades. [40]	42
Figura 15: Tabela de atributos do SensorTag com identificação das características que definem um determinado serviço. [40]	42
Figura 16: Tabela de atributos do SensorTag com identificação dos elementos que constituem uma característica, o seu valor e descrição. [40]	43
Figura 17: Hierarquização dos dados e dos atributos do serviço acelerómetro.	43
Figura 18: Arquitectura em rede proposta para a ferramenta em estudo.	47
Figura 19: Versão 1.0. com os respectivos módulos sensoriais. Observação: o módulo do finca-pés ainda não tinha sido desenvolvido, contendo a Figura uma representação simbólica de um sensor de força.	48
Figura 20: Aquisição de dados, SMB380. [23]	49
Figura 21: Protótipo do módulo da pagaia desenvolvido.	50
Figura 22: Diagrama de blocos que traduz o módulo da pagaia.	50
Figura 23: Modelo de Comunicação SPI. [38]	51
Figura 24: Máquina de estados do MSP430 para o módulo da pagaia.	52
Figura 25: Protótipo desenvolvido para o módulo do barco.	53
Figura 26: Diagrama de blocos que traduz o módulo do barco.	53
Figura 27: Máquina de estados do MSP430 para o módulo do barco.	54
Figura 28: Diagrama de blocos que traduz o módulo de recepção de dados na margem.	55
Figura 29: Máquina de estados do MSP430 para o módulo de recepção de dados na margem.	56
Figura 30: Testes preliminares à versão 1.0. realizados no rio Mondego.	57

Figura 31: Representação gráfica de x, y e z adquiridos no momento da pagaiada. Observação: imagens guardadas em ambiente de rio através de imagens retiradas ao ecrã do computador.	57
Figura 32: Contagem do número de pagaiadas: 1) Criança de 8 anos, 11 pagaiadas; 2) Adulto 15 segundos, 9 pagaiadas; 3) Adulto 30 segundos, 27 pagaiadas; 4) Série de 200m de um adulto, 33 pagaiadas.	58
Figura 33: Análise da informação adquirida de uma série de pagaiadas. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.	59
Figura 34: Parâmetros de um ciclo de remadas. [42]	59
Figura 35: Teste de 223m de distância entre o dispositivo emissor e receptor.	60
Figura 36: Amostra dos dados recebidos no módulo receptor no teste a 223m do módulo emissor. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.	60
Figura 37: Teste de 212m de distância entre o dispositivo emissor e receptor.	61
Figura 38: Amostra dos dados recebidos no módulo receptor no teste a 212m do módulo emissor. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.	61
Figura 39: Teste de 366m seguidos de 628m.	62
Figura 40: Amostra dos dados recebidos no teste às distâncias de 366m+628m. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.	62
Figura 41: Representação da solução idealizada para os sensores no finca-pés. 1,2,3 e 4 correspondem aos sensores.	63
Figura 42: Estrutura desenvolvida para o teste de conceito do sensor no finca-pés.	64
Figura 43: CC2541 SensorTag e respectivo <i>hardware</i> . [27]	65
Figura 44: Configuração <i>SoC</i> do SensorTag. [34]	65
Figura 45: Interface da Btool.	68
Figura 46: Configuração <i>Dual IC over HCI</i> entre a Btool e a dongle. [34]	69
Figura 47: Envio de comandos e pacotes HCI na comunicação entre a dongle e a Btool. [38]	69
Figura 48: Configuração em rede do envio de comandos e eventos HCI na rede estabelecida entre a Btool, a CC2540 Dongle e o CC2541 SensorTag. [38].	70
Figura 49: Comparação entre a estrutura de um comando HCI definido pelas especificações BLE e pelo fornecedor. [38].	70
Figura 50: Estrutura de um comando HCI definido para a Texas Instruments™. [38]	70
Figura 51: Comparação entre a estrutura de um evento HCI definido pelas especificações BLE e pelo fornecedor. [38]	71
Figura 52: Grupos de eventos HCI definidos pelo fornecedor, Texas Instruments™. [38]	71
Figura 53: Primeiros dados trocados quando a CC2540 USB dongle é ligada à Btool.	72
Figura 54: Pacotes de dados trocados no processo de <i>Scan</i> .	73
Figura 55: Pacotes de dados trocados no processo de <i>Establish</i> .	74
Figura 56: Modo de activação do serviço do acelerómetro.	74
Figura 57: Eventos e comandos HCI para activação do serviço do acelerómetro.	75
Figura 58: Modo de activação das notificações do acelerómetro.	76
Figura 59: Eventos HCI das notificações do acelerómetro.	76
Figura 60: Registo dos valores de aceleração dos três eixos.	76
Figura 61: Eventos e comandos HCI para terminar a ligação entre dispositivos.	77
Figura 62: Estrutura de um pacote de dados.	77

Figura 63: Estrutura do pacote de mensagens com os valores de aceleração dos eixos X,Y e Z.	78
Figura 64: Aplicação em C# com os dados do acelerómetro, do giroscópio e do magnetómetro do SensorTag a serem adquiridos em tempo real.	80
Figura 65: Imagem representativa do separador “Master”.	81
Figura 66: Imagem representativa do separador “Setup”.	81
Figura 67: Definição do comando GAP DeviceInit na plataforma C#. [38]	82
Figura 68: Definição do evento GAP DeviceInitDone na plataforma C#. [38]	82
Figura 69: Mensagem gerada na aplicação C# no momento de execução do evento GAP DeviceInitDone.	83
Figura 70: Definição do comando GAP GetParameter na plataforma C#. [38]	83
Figura 71: Definição do comando GAP DeviceDiscoveryRequest na plataforma C#. [38]	84
Figura 72: Definição do evento GAP DeviceInformation na plataforma C#. [38]	84
Figura 73: Mensagem gerada na aplicação C# no momento de execução do evento GAP DeviceInformation.	84
Figura 74: Definição do evento GAP DeviceDiscovery na plataforma C#. [38]	84
Figura 75: Mensagem gerada na aplicação C# no momento de execução do evento GAP DiscoveryDeviceDone.	85
Figura 76: Definição dos comandos GAP DeviceDiscoveryCancel e GAP EstablishLinkRequest na plataforma C#. [38]	85
Figura 77: Definição do evento GAP LinkEstablished na plataforma C#. [38]	86
Figura 78: Mensagem gerada na aplicação C# no momento de execução do evento GAP EstablishLink.	86
Figura 79: Formato e definição do comando GATT WriteLongCharValue.	87
Figura 80: Definição do atributo de activação do acelerómetro.	87
Figura 81: Definição do atributo de activação das notificações do acelerómetro.	87
Figura 82: Gráfico de visualização dos valores do acelerómetro, com e sem componente gravítica, na aplicação C#.	88
Figura 83: Definição dos atributos, relativos ao serviço do magnetómetro, que activam o sensor e as notificações do sensor. [40]	88
Figura 84: Definição dos atributos, relativos ao serviço do giroscópio, que activam o sensor e as notificações do sensor. [40]	89
Figura 85: Gráficos de visualização dos valores magnetómetro (imagem superior) e do giroscópio (imagem inferior) na aplicação C#.	89
Figura 86: Definição do comando GAP TerminateLinkRequest na plataforma C#. [38]	90

ÍNDICE DE QUADROS

Quadro 1: Consumo de corrente em *sleep mode* e no modo activo nos sensores do SensorTag. 61

SIMBOLOGIA E ABREVIATURAS

SIMBOLOGIA

V - Tensão

N - Newton

A - Ampere

nA - nanoAmpere

μ A - microAmpere

Ω - Ohm

ms - milissegundo

s - segundo

mm - milímetro

cm - centímetro

km - quilómetro

m - metro

bps - bits por segundo

kbps - quilobit por segundo

kb - quiloByte

GHz - GigaHertz

MHz - MegaHertz

Mbps - Mega bits por segundo

Mbit - Megabit

dB - decibel

dBm - decibel miliwatt

mW - miliwatt

ABREVIATURAS

ADC - Analog Digital Converter

API - Application Programming Interface

BLE - Bluetooth™ Low Energy

BT - Bluetooth™

CLK - Serial Clock

COM - Communication Port

CPU - Central Processing Unit

CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance

CSRK - Connection Signature Resolving Key

EEPROM - Electrically Erasable Programmable Read-Only Memory

FIC - Federação Internacional de Canoagem

FIFO - First In First Out

FPGA - Field Programmable Array

GPS - Global Positioning System

HW - Hardware

IC - Integrated Circuit

IDE - Integrated Development Environment

IEEE - Institute of Electrical and Electronics Engineers

IMU - Inertial Motion Unit

IRK - Identify Resolving Key

ISM RADIO BAND - Industrial, Scientific and Medical radio band

I/O - Input/Output

I²C - Inter Integrated Circuit

LED - Light Emitting Diode

LSB - Least Significant Bit

MOSI - Master Output, Slave Input

MISO - Master Input, Slave Output

MSB - Most Significant Bit

MPU - Microprocessor Unit
OTA - Over the Air
RAM - Random Access Memory
RF - Rádio Frequência
Rx ou Rxd - Receiver mode
SPI - Serial Peripheral Interface
SoC - System On Chip
SW - Software
SIG - Special Interest Group
Tx ou Txd - Transmitter mode
UART - Universal Synchronous Receiver/Transmitter
UUID - Universal Unique Identifier
USB - Universal Serial Bus
3D - Três Dimensões

CAPÍTULO 1 - INTRODUÇÃO

1.1. UMA VIAGEM NO TEMPO

A natureza é competitiva. Da mesma forma, o ser humano é naturalmente competitivo. Este não é um facto recente ou inesperado, mas mais do que isso. É uma verdade praticamente impossível de destronar, sustentada por séculos de história. Deixemo-nos viajar por lendas, mitos e narrativas que rezam outros tempos, outras verdades e doutrinas. Para os mais crentes são a imagem fiel de pura valentia e sabedoria dos nossos distantes antepassados. Para os mais cépticos não passam de fábulas para enaltecer a força do homem, encobrir o desconhecido e esconder o medo. Certamente que nem tudo será verdade, nem tudo será mentira e o exagero que narra alguns factos encobre a falta deles noutros. Idealismos à parte, e cada um com os seus, porque no fundo falamos de décadas de viagem que mais do que nos separar, nos unem.

É com um regresso à Grécia antiga, numa distância temporal que remonta aos tempos “antes de Cristo”, que revivemos a origem do evento desportivo, que nesta era moderna mais testemunha a competitividade do ser humano, glorificando o desporto em praticamente todas as suas modalidades. Se é uma verdade que já não se compete por amor e razão a um só Zeus, é também verdade que entre épocas, guerras, políticas, protestos, crenças e religiões, o espírito olímpico insistiu em prevalecer. Este, que nasceu para ser considerado um instrumento de aproximação entre os povos, homenageia nos nossos tempos a dedicação e o desempenho de um atleta que sonha pelo erguer de uma bandeira. A entrega que não se mede e a esperança que não se calcula são sentimentos que para um atleta têm tanto de certo como de ambíguo. São certos quando o ambicionado pódio se conquista com certeza e por outro lado ingratos quando o que distingue a memória de um vencedor da de um participante se resume a meros números, ainda que depois de uma vírgula. Tão ou mais importante do que fazer e fazer bem, é conseguir fazê-lo no tempo certo. É esta a noção de tempo de um atleta: demorar menos possível para fazê-lo melhor do que qualquer um. Em tanta ambição é o rigor, o método e a perseverança que distinguem um atleta exímio. É o treino quem determina a técnica, a técnica quem compromete o tempo e é o tempo quem decide uma competição e que dá alma ao sonho.

1.1.1. O impacto da tecnologia no desempenho desportivo

Num reino onde a diversidade é animalésca, a competitividade do Homem assemelha-se à de outras espécies. No entanto, não há quem mais impere pela sabedoria. O Homem tem o poder de conceber, questionar e analisar. Desenhar métodos, criar ferramentas e desenvolver soluções. De forma a responder às suas necessidades ele complica para depois descomplicar. Em busca das melhores respostas às mais complexas perguntas surge quem melhor representa e simboliza anos de evolução, inovação e transformação, a tecnologia. Esta, que tem como definição teórica, “conjunto de equipamentos técnicos e procedimentos recentes que permitem o tratamento e a difusão de informação de forma mais rápida e eficiente” no Dicionário Básico da Língua

Portuguesa da Porto Editora, resume-se de forma descomplicada e prática a tudo aquilo que surgiu para facilitar o nosso dia-a-dia.

Entre os séculos XVI e XVIII, transformações colossais estabeleceram uma nova percepção do mundo, que ainda pulsa nos dias de hoje. Os míticos Jogos Olímpicos do Verão de 86 inspiraram a mudança e aspiraram pela inovação. Desde então, um progresso emergente apoiado por avanços tecnológicos e pelo crescente acesso à informação melhorou a imagem do desporto.

A introdução da fibra e materiais fibrosos no desporto é um exemplo claro de transformação em praticamente qualquer modalidade desportiva seja pela inclusão da mesma em equipamentos desportivos e de monitorização ou no vestuário de um atleta. Para uma noção mais razoável do impacto desta mudança, basta pensar no record de 1972 de Eddie Merckx que foi considerado o maior esforço humano possível devido ao impacto que uma bicicleta, sem um desenho de produto (*design*) aerodinâmico e materiais leves na sua constituição, teve na prestação do atleta. Por outro lado, e bem mais recente, Biederman em 2009 abriu as primeiras páginas de todos os jornais com a sua participação naqueles que foram considerados como os “the plastic games”, no Campeonato Mundial de Desporto Aquático em Roma. Biederman, nos primeiros três de oito dias de prova superou 15 records mundiais por ter utilizado um fato de fibras especialmente desenhado para conferir ao atleta flutuabilidade, estabilidade, velocidade e resistência.

No entanto, o percurso evolutivo de praticamente todas as modalidades desportivas, para além de ter sido largamente influenciado pela tecnologia, teve também intervenção da mudança do estilo de vida e maturação dos atletas, de outros métodos de treino adquiridos, e do acesso a estudos e informação que resultaram em novas técnicas e na obtenção de melhores resultados. Um exemplo disso encontra-se no lançamento do dardo onde, para além de outras mudanças, houve a necessidade de deslocar quatro centímetros para a frente o centro de massa do dardo, uma vez que as distâncias de lançamento alcançadas pelos atletas tendiam a superar o comprimento dos estádios. Curiosamente no atletismo um adolescente de 15 anos atinge em 100 metros de corrida de velocidade os 10,27 segundos que foram, em 1896, mérito de uma medalha de ouro.

Assim, desafiando éticas e limites, a tecnologia aliada à própria evolução natural da espécie, provou desempenhar um papel fundamental no desporto sobretudo na melhoria do desempenho de atletas de alto rendimento. Em [1] determina-se que a influência da tecnologia para que atletas tenham superado records mundiais é de 2%. Numa atmosfera limpa e justa que deve ser uma competição desportiva, a igualdade entre todos os atletas é fundamental. Desta forma, o que distingue um atleta com ou sem medalha é a resistência, coordenação e técnicas adquiridas no momento de treino. Por ser a chave do sucesso, é no treino que reside a maior aposta de qualquer atleta. Beneficiando desse facto, mais do que nunca, a tecnologia aproveitou para entre acordo e desacordo de morais e regras, vingar no mercado. Enganam-se aqueles que pensam que este é o seu auge de ascensão, porque no desporto só agora começou a emergir.

1.2. FORMULAÇÃO DO PROBLEMA

Gadget é um conceito associado à tecnologia moderna que entrou no nosso quotidiano e que, mais do que nenhum outro, faz jus à frase publicitária que Fernando Pessoa fez em 1927 para a Coca-Cola “Primeiro estranha-se e depois entranha-se”. Se no início tudo o que era considerado um *gadget* era estranho à maioria, hoje a realidade é outra. Apesar de se associar cada vez mais ao engenhoso do que propriamente ao útil, um *gadget* surgiu com o intuito de simplificar o nosso dia-a-dia.

O desporto não é excepção. Actualmente já se perdeu a noção à quantidade de equipamentos existentes no mercado que, de forma inteligente e eficiente, são capazes de monitorizar e analisar rendimentos físicos, adquirindo informação, melhorando técnicas e corrigindo falhas. Estes dispositivos entraram no mercado do desporto não para serem vencidos, mas para vencer e tornar vencedores quem a eles se aliar. A personalização a cada atleta, a adaptação a cada treino, a criação de objectivos e o acompanhamento pessoal, aliados à leveza e portabilidade são apenas algumas das características que lhe atribuem valor e conferem interesse. Num mercado incipiente como este, mas com um futuro promissor, surgem ainda as tecnologias *wearable*, tecnologias electrónicas ou computadorizadas que podem ser incorporadas em peças de roupa e acessórios. Estas, mais do que adeptas do conforto, primam pelo *design*. Se estes aparelhos, que tornaram a tecnologia pessoal, provaram ser um parceiro útil para qualquer amante do desporto, para um atleta profissional e de alto rendimento são ferramentas imprescindíveis no seu treino. Assim, tanto o que se preocupa com o seu bem-estar, como aquele que faz disso o seu dia-a-dia, é acompanhado por dispositivos que gravam e preenchem a sua actividade com dados e informação multimédia que podem, em tempo real ou no final do treino, ser analisados e partilhados em comunidades e redes sociais.

No entanto, numa fase onde estas soluções são cada vez mais valorizadas, nem todas as modalidades desportivas foram merecedoras da mesma atenção, havendo neste enredo modalidades protagonistas e outras secundárias. A distinção é óbvia. Com um analogismo a uma balança facilmente se percebe o quão desequilibrado foi o interesse no desenvolvimento de soluções direccionadas a determinadas modalidades desportivas. Assim, se de um lado temos o mediático e enriquecido mundo do futebol ou do golf, do lado oposto, desequilibrando a balança, temos a modéstia e des vaidade da canoagem ou do remo.

A ferramenta, apresentada nesta dissertação, pretende o desenvolvimento de um sistema de monitorização para colmatar a falha nos dispositivos e métodos de aquisição de dados e avaliação de rendimento em atletas de modalidades náuticas não monitorizadas, nomeadamente na canoagem. Para os atletas desta modalidade informação relativa ao seu exercício físico e prestação, em momento de treino ou competição, pode ser importante para trabalhar o sucesso e corrigir falhas. A ferramenta proposta é baseada numa arquitectura em rede, pela implementação de sensores inerciais na pagaia e no barco, bem como sensores de força no finca-pés, criando três módulos sensoriais distintos mas com informação que se complementa. Com esta ferramenta pretende-se adquirir e fundir dados, dos três módulos sensoriais, em tempo real e devolver os dados adquiridos, depois de processados, numa aplicação C# desenvolvida

para visualizar a informação recolhida. A aquisição e o tratamento dos dados provenientes dos módulos sensoriais é possível pela implementação de protocolos de comunicação na rede.

1.3. MOTIVAÇÃO

A pergunta que se impõe é porquê a canoagem? A resposta mais adequada é que não foi um projecto ou uma ideia que foi de encontro a esta modalidade desportiva, mas sim a canoagem que procurou uma solução para algumas das suas necessidades. Pela proximidade com atletas, federados e não federados, e treinadores de dois dos mais emblemáticos clubes de canoagem nacionais, o Clube Fluvial de Ponte de Lima e o Clube Náutico de Coimbra, foi possível perceber as necessidades tecnológicas para a canoagem que é reconhecidamente uma modalidade com poucas soluções comerciais de monitorização do rendimento do desportista. Motivo pelo qual actualmente os parâmetros de treino dos atletas, amadores ou profissionais, são na sua maioria avaliados de forma visual pelos seus treinadores ou auxiliares, que recorrem a conceitos teóricos e à sua experiência pessoal, ou ainda pelo próprio atleta em treinos individuais. Tudo se resume assim, a uma observação que se espera ser exímia mas que na prática é influenciável por condições atmosféricas, por distâncias e até pelo cansaço e monotonia. Por ser um método falível e subjectivo há uma tentativa e um esforço grandes por parte dos atletas em obter, de forma automática e inteligente, parâmetros do seu treino. O contacto directo com esta realidade, junto de atletas e dos seus treinadores, permitiu concluir que nenhum dos produtos actualmente existentes no mercado consegue assegurar a completa funcionalidade para a canoagem, ou por ser reaproveitado de outras modalidades ou porque de facto não dá ao atleta toda a informação que o mesmo precisa de ter. Por este motivo é frequente que atletas integrem mais do que um dispositivo para obterem informação relevante.

De forma a conferir ao atleta maior autonomia e desempenho e ao treinador um complemento à sua análise foi pensado no CanADev como uma ferramenta para o desenvolvimento de um sistema de monitorização da modalidade de canoagem de velocidade. O CanADev, a ferramenta apresentada nesta dissertação, consiste num estudo à forma como analisar e recolher parâmetros quantitativos e qualitativos da prestação de um atleta, em tempo real, com recurso à electrónica e a protocolos de comunicação. Assim, nesta dissertação, é apresentada uma forma possível de adquirir dados de vários sensores, implementados numa arquitectura em rede e baseados em tecnologia sem fios, recorrendo a protocolos de comunicação. A forma descrita pode ser utilizada num sistema de monitorização da canoagem de velocidade. A prova conceito que sustenta a aquisição de dados, dos módulos sensoriais da ferramenta apresentada, permitiu a incidência do estudo nos protocolos de comunicação, nomeadamente no BluetoothTM Low Energy.

1.4. ESTRUTURA DA DISSERTAÇÃO

Depois de introduzida e feita uma abordagem ao Estado da Arte segue-se o Capítulo 3 que descreve a modalidade desportiva para a qual esta ferramenta foi pensada: A canoagem. Neste Capítulo é apresentada de forma generalizada a ferramenta proposta, a sua arquitectura e características. Além disso, é feita uma breve referência às especificações de um sistema de monitorização com o qual a ferramenta é compatível, podendo contribuir para o seu desenvolvimento. O Capítulo 4 é relativo aos protocolos de comunicação. É feita uma breve apresentação do ZigBee™ por ter sido uma possibilidade de protocolo a implementar, na arquitectura definida, e uma explicação do Bluetooth™ Low Energy, como sendo o protocolo eleito. O Capítulo 5 descreve duas versões da ferramenta utilizadas para testar e validar a aquisição e encaminhamento de dados pela rede e módulos que dela fazem parte. É associado à versão 1.0. que surge uma apresentação detalhada dos módulos implementados na arquitectura em rede, o módulo da pagaia, do barco e do finca-pés. Além disso, é esta versão e os resultados com ela obtidos, que justificam a possibilidade de, com a ferramenta proposta, adquirir dados dos módulos sensoriais. Esta versão, que pode ser considerada como uma prova de conceito, desencadeou o início do estudo do protocolo de comunicação BLE. A versão 1.0. permitiu identificar a necessidade de implementação de um protocolo mais adequado na rede. Por esse motivo, surge a versão 2.0. como um estudo ao BLE e à integração do BLE numa rede com módulos sensoriais. A versão 2.0. introduz os Capítulos seguintes desta dissertação. No Capítulo 6 é feito o estudo detalhado do Bluetooth™ Low Energy, especificando como é estabelecida a comunicação entre dispositivos e a aquisição de dados de módulos sensoriais BLE. Este Capítulo tornou-se essencial para sustentar o Capítulo seguinte. É no Capítulo 7 que é feita a descrição de uma plataforma de comunicação e visualização de dados, desenvolvida em C# a partir de código fornecido pela Texas Instruments™, para validar a aquisição de dados de módulos sensoriais e a implementação do Bluetooth 4.0.

CAPÍTULO 2 - ESTADO DA ARTE

Procurando pelos trabalhos propostos na literatura e na indústria, que podem ser relacionados com a canoagem e respeitantes à monitorização de atletas, da sua técnica e do seu rendimento, é possível distinguir duas vertentes. Se por um lado existem estudos e soluções comerciais que apostam no processamento de imagem e análise de vídeo, por outro lado, o estudo desta modalidade desportiva assenta igualmente em dispositivos e sistemas electrónicos, comerciais e não comerciais, baseados em sensores e sistemas embebidos. As soluções inseridas em qualquer uma das vertentes mencionadas anteriormente, para além de monitorizarem parâmetros análogos, têm uma característica em comum: destinam-se a analisar o rendimento e prestação do atleta no treino fora de água. Todos os estudos que se revêm nesta particularidade carecem de rigor na medida em que o ambiente e as condições em que foram desenvolvidos e testados são controlados e por isso diferentes do ambiente e condições reais. Por outro lado, em laboratório, por mais que sejam simuladas a influência de condicionantes externos na execução do atleta, como o exemplo do atrito do ar, da água e as próprias oscilações do barco, qualquer simulação está longe de inserir o atleta num ambiente semelhante ao estar dentro de água.

Segundo [2-10] para que os resultados de um sistema de monitorização por vídeo sejam detalhados e viáveis, o sistema deve contar com o maior número possível de câmaras estrategicamente colocadas. Em estudos diferentes e espaçados no tempo, foram feitas diversas experiências para a análise desta modalidade desportiva com um número variado de câmaras. As referências [2] e [3] são exemplos de soluções onde foi envolvida uma câmara de vídeo. Em [2] é descrito um sistema simplificado que pretende analisar a técnica da pagaia através de alguns parâmetros medidos como o ângulo da pagaia no momento de entrada na água, a força da pagaia e o seu tempo dentro de água. O estudo mencionado em [3] também se foca na análise da pagaia e da sua cinemática, desta vez, fora de água com o auxílio de um ergómetro. Se em [2] a informação é retirada pela análise detalhada de imagem, em [3] os dados são transferidos para o computador através de ligações por cabo e processados num programa desenvolvido em MatlabTM. As soluções a que [4] e [5] se referem indicam a utilização de duas câmaras para a análise do comportamento da pagaia, à semelhança dos estudos referidos anteriormente, e para a avaliação do movimento do corpo do atleta. Apesar de recorrerem a duas câmaras de vídeo, a forma como estas foram posicionadas em ambos os testes foi diferente. Em [4] foram colocadas para dar uma visão frontal e lateral da pagaia e do atleta e em [5] foram estruturadas para que o atleta se posicionasse a meio do sistema de vídeo para obter a visão das duas laterais. O processamento de imagem recorre a algoritmos e à aplicação de transformadas matemáticas com o objectivo de sincronizar as imagens adquiridas por ambas as câmaras no mesmo instante de tempo. Em [6], o sistema desenvolvido inclui a utilização de quatro câmaras de vídeo digitais assim como um sistema de vídeo a três dimensões. Desta forma, foi possível fazer uma análise quantitativa da técnica da pagaia pela prévia caracterização de movimentos típicos de uma pagaia, o movimento aéreo e o de arrastamento dentro de água, e a respectiva comparação com os resultados obtidos. Um sistema de telemetria, isto é, que não recorre a cabos para a transferência dos dados monitorizados para uma plataforma de análise, é referido

em [7]. Este foi o estudo onde foram obtidos os melhores resultados pela utilização de seis câmaras de vídeo que permitiram fazer uma correlação entre a morfologia do atleta e o seu desempenho. Em [8], [9] e [10] são descritas soluções comerciais baseadas em análise de imagem e processamento de vídeo onde são feitas a reconstrução do movimento da pagaia e a comparação de parâmetros analisados em ambientes simulados e controlados dentro de água com os adquiridos por um ergómetro.

Mais do que as análises de vídeo, que implicam a preparação de infraestruturas e aquisição de equipamentos que podem tornar uma solução e um estudo demasiado dispendiosos, os dispositivos e equipamentos baseados em sensores inerciais e em sistemas embebidos são os mais comumente utilizados e desenvolvidos na monitorização desportiva. Motivo pelo qual, a maior parte das soluções existentes no mercado para a monitorização e aquisição de dados na canoagem, têm por base a utilização destes sensores. Os sensores garantem portabilidade, miniaturização e ao mesmo tempo rigor. Em [11], [12] e [13] são apresentados trabalhos de investigação que pretendem obter informação relevante sobre uma pagaiada pela utilização de acelerómetros, giroscópios e magnetómetros. Dados como a força da pagaiada, a fase, a sua duração e o seu tempo total, que inclui o tempo em que a pagaia está dentro e fora de água, são essenciais para definir a frequência de um movimento repetido, caracterizado como sendo a cadência da pagaiada. É com base na cadência da pagaiada que são efectuados treinos de resistência e técnica, na medida em que, quanto maior for o número de pagaiadas idealmente executadas no menor espaço de tempo, maior é a velocidade adquirida pelo conjunto atleta e barco. Através de um sensor de três eixos, um acelerómetro, de um microprocessador de baixo consumo de 16 MHz e um GPS (Global Positioning System) é gerada uma plataforma de aquisição de dados, proposta em [11]. Em tempo real e com recurso a uma porta série, dados de velocidade, posição e aceleração são adquiridos e enviados para um computador para serem processados. Embora seja possível, com este sistema, obter informação de forma visual em tempo real, o sistema é, na melhor opinião do autor desta dissertação, limitativo pela utilização de cabos, que compromete a normal execução da técnica do atleta, e pela quantidade de informação que é fornecida. Em [13] é proposto um sistema de medição e estimativa da técnica da pagaiada. Este sistema, desenvolvido na Faculdade de Ciências do Desporto da Universidade de Coimbra, inclui sensores inerciais, uma bateria para garantir a portabilidade do sistema e um cartão micro SD que permite que os dados monitorizados sejam guardados. Devido a esta solução de recolha de dados, limitada pelo espaço existente no cartão e pela preocupação em garantir espaço suficiente para armazenar os dados do treino, é utilizado um cabo *USB* (Universal Serial Bus) que permite depois da prática desportiva processar e analisar os dados no computador. A recorrer igualmente a um cartão de memória para guardar os dados adquiridos surge o trabalho [14] que refere uma solução em que é utilizado um acelerómetro colocado no casco do barco e a informação recolhida é analisada, tal como em [13] pós treino. Para a monitorização da duração de uma pagaiada, da sua respectiva força, do tempo de imersão em água após cada impulso e da simetria entre pagaiadas, à direita e à esquerda, foi desenvolvida a solução [12], com três unidades independentes de aquisição de dados, denominadas por nós. Cada nó, um na parte mais alta do casco do barco e dois nas pás da pagaia respectivamente, são constituídos por sensores de força. Para complementar o sistema foram

incorporados os mesmos sensores numa versão *wearable* nos cotovelos do atleta. Os dados são sincronizados recorrendo ao protocolo de comunicação BluetoothTM e a análise dos mesmos é feita remotamente numa plataforma desenvolvida em LabViewTM. Os autores deste estudo apresentaram, após a análise dos resultados conseguidos, algumas preocupações relativas à impermeabilidade dos sensores colocados nas pás da pagaia que por ficarem completamente submersos em água, tiveram que ser envolvidos e protegidos com uma estrutura robusta que poderá ter afectado as próprias medições. Além disso, os mesmos revelam ainda relutâncias no protocolo utilizado que limita a distância entre o sistema de aquisição de dados e o de processamento. O método de análise discutido em [15] representa, à semelhança do anteriormente referido, uma rede sem fios, baseada em quatro nós que incluem sensores de força, dois nós idênticos em cada pá da pagaia e outros dois colocados no local onde o atleta apoia os pés. O método inclui ainda um nó central que permite que os dados sejam sincronizados e enviados para uma unidade de processamento por BluetoothTM (BT). À semelhança daquilo que foi apontado e comentado em relação a [12], as mesmas observações são válidas para este trabalho. Existem ainda dispositivos e estudos baseados em GPS para a aquisição de dados, [1]. O sistema desenvolvido em [1] faz o cruzamento de informação recebida pelo GPS com a informação adquirida por três nós individuais colocados nas pás da pagaia e no barco. O sistema pretendido recorre a sensores piezoeléctricos resistivos para calcular as forças na pagaia e a acelerómetros e giroscópios para obter velocidades e posições. Os dados são processados recorrendo a uma solução comercial nomeadamente uma plataforma MulleTM, que é um sistema embebido de baixo consumo com um microprocessador e com um módulo BluetoothTM. A referência [16] é relativa à solução comercial que mais se direcciona à modalidade, sendo por isso em termos comerciais a mais completa. Esta inclui uma bateria, um acelerómetro de três eixos, um microprocessador e a transmissão é feita recorrendo ao protocolo ANT+. O dispositivo desenvolvido, que não é mais do que um medidor de cadência da pagaiada, é colocado no centro de uma pagaia como se de uma mola se tratasse. Este dispositivo permite a integração com equipamentos comerciais de outras marcas como por exemplo cintas de medição de frequência cardíaca, e ainda com dispositivos de treino que incluam GPS. No entanto, a integração e a recolha de dados só são possíveis se houver um receptor em todos os dispositivos envolvidos que utilize o mesmo protocolo de comunicação. Apesar de este produto ser no mercado o mais vocacionado para a canoagem não é utilizado pela maioria dos atletas. Se por um lado a única informação fornecida, a menos que seja complementado com outros equipamentos, é a de cadência, que é informação insuficiente a um atleta, por outro é financeiramente pouco viável à maioria dos atletas quando adquirida individualmente, e muito menos viável quando complementada com outros dispositivos. Desta forma, o atleta que necessita de informação acaba por se restringir aos relógios de pulso, direccionados a outras modalidades desportivas, mas que ainda assim adquirem, em termos quantitativos, mais informação comparativamente ao produto referido em [16]. Por sua vez os relógios de pulso, para além de estarem otimizados para outras modalidades condicionam o rendimento e execução do exercício ao atleta. Sendo também limitativos em termos de visualização da informação recolhida, o tamanho e apresentação dos dados são reduzidos e necessitam da complementaridade de acessórios para obter toda a informação necessária a um canoísta.

Seja pelos condicionantes à execução de movimentos do atleta, seja pela limitação de informação adquirida e fornecida, pelos limites impostos no processamento de dados, que na sua maioria é pós treino, ou ainda pela quantidade de equipamentos que tem que ser utilizados e montados, qualquer uma das soluções apresentadas provou ser pouco viável para aquele atleta, a quem o que mais lhe interessa é obter dados objectivos do seu treino.

CAPÍTULO 3 – A CANOAGEM E O SISTEMA PROPOSTO

A canoagem é uma modalidade desportiva que apesar de habitualmente associada à competição de velocidade em águas calmas, sprint, à agilidade em turbulência, slalom, pode ser distinguida em outros tipos como estilo livre, onde se avaliam as manobras com a embarcação em águas agitadas, longas distâncias, entre outros. Embora estas e outras disciplinas sejam reconhecidas pela Federação Internacional de Canoagem, FIC, é o sprint, aceite nos Jogos Olímpicos desde 1936, que detém um maior número de praticantes e apoiantes.

Este Capítulo faz uma breve introdução ao sprint por ser na canoagem a modalidade estudada para a ferramenta apresentada nesta dissertação. Desta forma, este Capítulo prevê uma familiarização com conceitos e conhecimentos técnicos da modalidade. Perceber as fases e sub-fases de uma pagaiada bem como as forças envolvidas é necessário para a compreensão do esforço exigido na canoagem. Além disso, é importante ter conhecimento dos factores que influenciam o resultado da prestação de um atleta no ambiente em que se insere. A interiorização destes conceitos permite especificar um possível sistema de monitorização com o qual a ferramenta, e arquitectura para ela prevista, é compatível podendo inclusive ser utilizada no seu desenvolvimento.

3.1. SPRINT – VELOCIDADE, FORÇA E TÉCNICA

A velocidade, a força e a técnica são características que para além de descreverem esta disciplina na canoagem, distinguem os atletas que a praticam. O sprint é realizado em águas calmas, em canais de água artificiais ou naturais, em pistas com normalmente 2 km de comprimento para competições que se podem realizar às distâncias de 200, 500 e 1000 metros. Mais recentemente, para além do habitual barco que tão facilmente reconhecemos pela sua elegância e rapidez, foi autorizada pela FIC a prática de canoagem em canoas. As embarcações, tanto no barco como na canoa, podem conter 1, 2 ou 4 atletas sendo reconhecidas na modalidade como K1, K2, e K4 no barco, e C1, C2 e C4 na canoa.

A técnica da pagaiada consiste numa sequência de movimentos repetidos e sincronizados caracterizada pela execução de duas fases sequenciais: a fase de voo da pagaia e a fase em que esta é submersa na água. A fase de voo da pagaia ocorre sempre que a pagaia sai da água após a sua submersão. Esta fase corresponde ao deslocamento da pagaia no ar para que pagaiadas à esquerda e à direita ocorram alternadamente. Na submersão da pagaia distinguem-se três sub-fases igualmente sequenciais que correspondem ao momento em que a pagaia rompe a linha de água, ao de arrastamento da pagaia dentro de água e ao momento de corte e saída da linha de água. Desde que a pagaia sai de água, terminando a fase de submersão de uma das pás, é iniciada a fase aérea de deslocamento da pagaia para que a pá oposta seja agora submergida.

A Figura 1, mostra a análise de um ciclo completo, duas pagaiadas consecutivas à direita e à esquerda, caracterizado pelas suas fases e sub-fases.

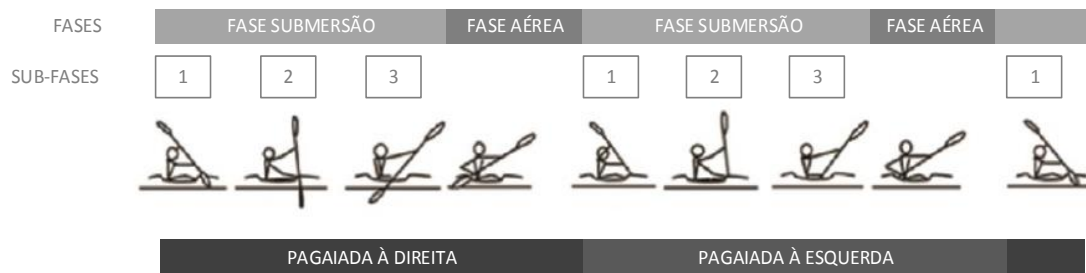


Figura 1: Análise de um ciclo de uma pagaiada. Análise das sub-fases: 1 - momento de entrada na água; 2 – momento de arrastamento da pagaia dentro de água, 3 – momento de saída da pagaia da água.

O tempo individual de cada uma das fases quando somado corresponde ao tempo de execução de uma pagaiada. Quanto menor for o tempo em que a fase aérea decorre, menor é o tempo entre fases de submersão e por isso, maior é a frequência ou cadência da pagaiada. A fase aérea é também considerada como a fase de recuperação e a fase de submersão como a fase activa. Um dos parâmetros que distingue a eficiência de um atleta de elite é o tempo de ambas as fases. Estima-se que um atleta de elite tenha longas fases activas, ou seja longos períodos de aplicação de forças máximas, e curtas fases de recuperação, para executar o maior número de pagaiadas possível. Para um atleta de alto rendimento é crucial, numa boa prestação, executar não menos do que 120 a 130 pagaiadas por minuto, [17].

O movimento de uma pagaiada é caracterizado por forças entre os elementos atleta, barco e pagaia. A Figura 2, mostra uma representação das principais forças existentes numa pagaiada. Para além das forças indicadas existem outras que tem uma menor influência na estabilidade e velocidade do barco, como a força exercida no banco onde o atleta se senta.

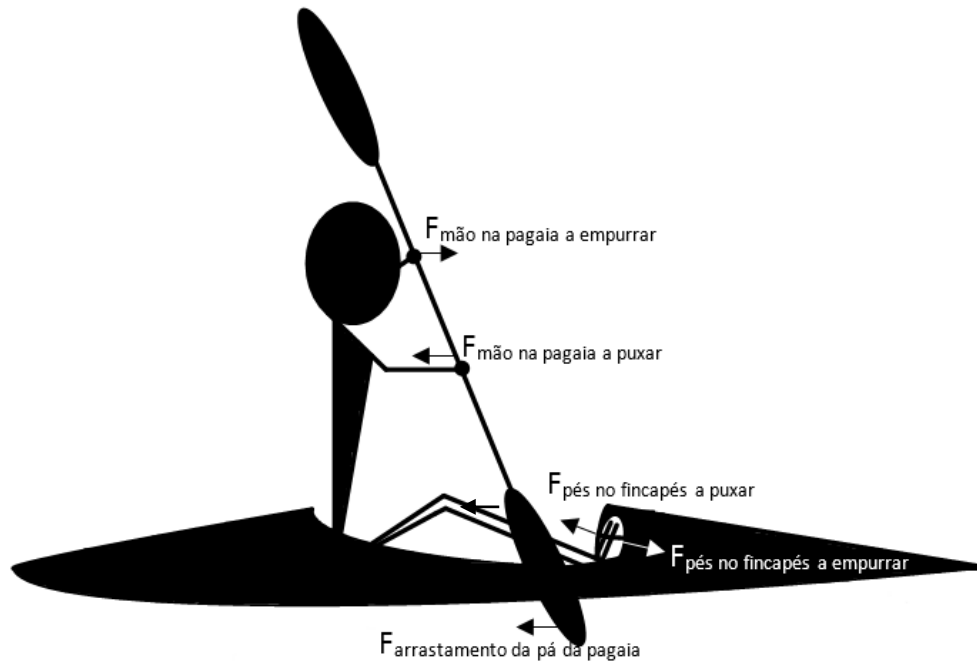


Figura 2: Principais forças existentes numa pagaçada.

A grande maioria dos estudos existentes foca-se naquela que é quantitativamente mais significativa, a força da pagaçada. Não existe uma regra linear que auxilie um atleta na escolha de uma pagaia, mas existem factores que a caracterizam. No entanto, no momento de escolha não é o tamanho, o peso e o factor de forma que mais importam mas a variável conforto que, embora subjectiva, é essencial. A força que o atleta exerce sobre a pagaia varia de acordo com o conforto que o mesmo sente em relação à distância de posicionamento das mãos, do comprimento e peso da pagaia, e do seu valor e direcção [18]. Uma abordagem a alguns estudos mencionados em [17] que medem as forças exercidas por atletas de alto rendimento em diferentes situações permitem resumir que a força média no momento de largada é de aproximadamente 400N. Para uma regata de 500 metros foram aferidos valores de força na ordem dos 250 a 300N e numa competição de 1000 metros o valor médio da força exercida por um atleta é de 250N. Por outro lado e relativamente à força exercida no finca-pés em [19] é referido que a força, que é diferente nos dois pés, pode atingir valores até 480N.

A técnica de uma pagaçada, que é complexa e exigente, implica uma actividade dinâmica de vários membros e partes do corpo como os braços, ombros, tronco e pernas. A acção que os pés exercem no finca-pés é essencial para a rotação do tronco e movimento pélvico, impulsionando e facilitando o movimento dos braços e ombros. O estudo da técnica mais eficiente que permite uma pagaçada óptima tem sido largamente desenvolvido e pode ser encontrado em trabalhos anteriores [19], [20], [21] e [22]. Para além do tempo de uma pagaçada, da cadência e das forças envolvidas são vários os factores que nesta modalidade desportiva decidem um resultado. A Figura 3, apresenta alguns dos factores mais relevantes que influenciam a prestação de um atleta numa competição.

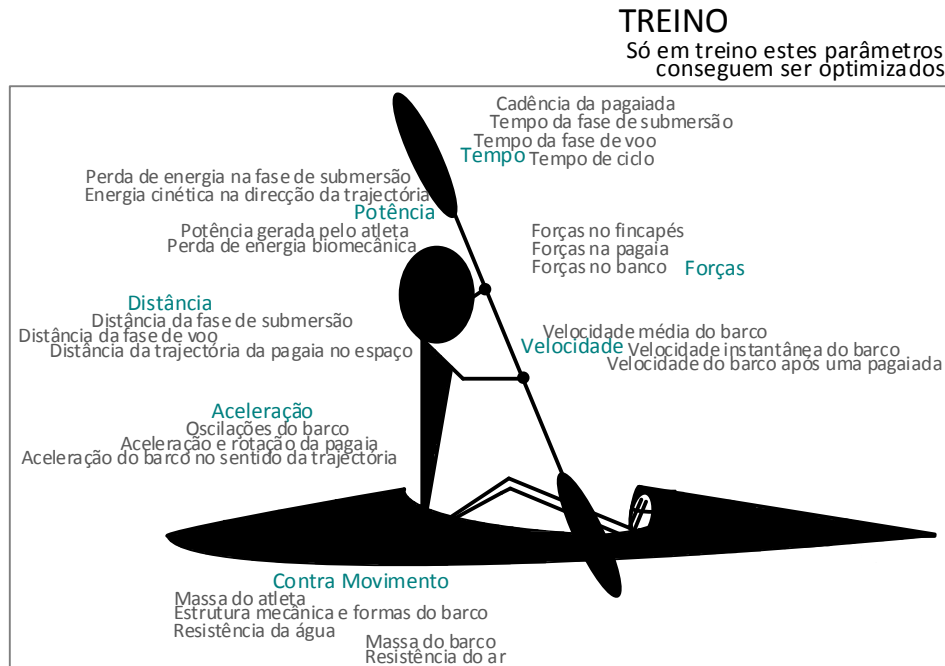


Figura 3: Principais factores que influenciam o resultado de uma competição.

É importante perceber o conceito e técnica da canoagem, os factores que influenciam o resultado e os requisitos e parâmetros que os atletas e treinadores consideram relevantes para idealizar e especificar um sistema de monitorização e aquisição de dados para esta modalidade.

3.2. ESPECIFICAÇÕES DO SISTEMA

- Ferramenta de auxílio para que atletas de forma individual e autónoma monitorizem o próprio treino;
- Ferramenta de auxílio para os treinadores obterem e analisarem dados objectivos de vários atletas em simultâneo;
- Ferramenta que permite a optimização do barco a cada atleta pelo fornecimento de informação que pode, em termos mecânicos e de construção, melhorar o rendimento do próprio barco;
- Mecanismo que auxilie o treinador a perceber a melhor disposição dos atletas numa tripulação, de forma a optimizar os resultados;
- Sistema sem fios, leve, portátil e não intrusivo;
- Sistema que facilmente se adapta ao atleta e a diferentes pagaias e barcos que o mesmo possa ter;
- Medições em tempo real e dentro de água;
- Sistema de transmissão a longas distâncias para que o factor distância não seja limitativo;
- Possibilidade de guardar os dados em situação de inviabilidade do módulo de transmissão a longa distância;

- Sistema funcional em momento de treino e competição, sem estar dependente de uma interface de visualização para poder ser utilizado pelo atleta em momento de prova;
- Capaz de fornecer ao atleta um alerta sonoro da cadência de pagaiada;
- Capaz de adquirir dados como força de pagaiada individualmente, isto é, à direita e à esquerda;
- Possibilite a avaliação da força feita individualmente em cada pé para perceber dessa forma o sincronismo do movimento dos pés com o movimento da pagaiada e a força de impulso;
- Permita a avaliação da forma como a pagaia entra e sai da água, bem como o seu movimento de inclinação e torção;
- Permita concluir o comportamento da pagaia dentro de água e o tempo de imersão e de cada sub-fase;
- Permita calibrar o equilíbrio da pagaia e adquirir dados de velocidade, aceleração e oscilações do barco;
- Facilite a observação do movimento corporal do atleta, avaliando posturas e corrigindo falhas;
- Facilmente extensível para a inclusão, incorporação e complementaridade de outros dispositivos em versões futuras;

As especificações apresentadas dizem respeito a uma proposta de sistema de monitorização da canoagem de velocidade com a qual a ferramenta proposta nesta dissertação é compatível. O estudo apresentado, de um método possível de aquisição e tratamento de dados da prestação de um atleta desta modalidade com recurso a uma rede de sensores sem fios e protocolos de comunicação, pode contribuir para o desenvolvimento do sistema especificado.

Pela certeza que o sistema mais adequado para monitorizar um canoísta passa pela menor influência possível com a execução de movimentos e da própria prática desportiva, foi estudada uma arquitectura, para a ferramenta apresentada, que facilmente se insere e adapta ao atleta e ao barco. Sem comprometer a prática do exercício, em treino ou em competição, é possível com uma arquitectura baseada numa rede sensorial sem fios e constituída por três módulos (módulo da pagaia, módulo do barco e módulo do finca-pés) adquirir dados específicos e individualizados do atleta. A base da arquitectura em rede projectada é um protocolo de comunicação e por isso o estudo e validação dos protocolos de comunicação possíveis de implementar na arquitectura proposta são discutidos no Capítulo seguinte.

CAPÍTULO 4 - PROTOCOLOS DE COMUNICAÇÃO

Uma conversa entre duas pessoas é baseada na utilização de um protocolo de comunicação e é certamente um dos mais antigos. Tal como em qualquer protocolo, numa conversa há um emissor, um receptor e troca de informação entre ambos. Mantendo esta analogia e de acordo com a experiência de qualquer um de nós, numa conversa pode haver perda de informação entre os intervenientes. As ondas sonoras que se propagam no ar quando falamos, não são eficazes numa situação de ruído ou de longas distâncias sendo por isso necessária uma grande quantidade de energia para propagar uma onda sonora nessas condições. Por outro lado, uma onda electromagnética no mesmo contexto consegue ser eficaz devido à forma estruturada como a gama de frequências se encontra dividida.

Quando se pretende adquirir dados em simultâneo de três módulos distintos e individualizados é importante que os três módulos se harmonizem para que troquem informação entre eles. Um protocolo de comunicação serve para gerir e organizar os três módulos e tal como se de uma conversa se tratasse definir quais dos módulos o emissor e o receptor. O protocolo de comunicação é por isso a base da arquitectura do sistema proposto, permite o sincronismo e impede a perda de informação.

Neste Capítulo serão abordados dois protocolos de comunicação: o ZigBee™ e o Bluetooth™ Low Energy. Estes foram os dois protocolos considerados para estabelecer a rede sensorial na arquitectura proposta.

4.1. RÁDIO FREQUÊNCIA, ZIGBEE™

O ZigBee™ é um protocolo de comunicação que pretende associar a transmissão de dados sem fios a um reduzido consumo energético e a uma elevada fiabilidade. Este protocolo opera nas bandas de frequência não licenciadas de 2.4GHz, 900MHz e 868MHz sob a especificação 802.15.4. do IEEE (Institute of Electrical and Electronics Engineers). Embora se tenham como referência as bandas de frequência autorizadas a este protocolo, a saída de dados será menor do que a taxa máxima especificada, devido a latências e à sobrecarga de processamento de dados. As aplicações pensadas para ambientes fechados estão mais limitadas no que diz respeito às distâncias máximas para a comunicação que podem variar entre 10 a 20 metros. Numa aplicação de exterior a distância máxima para comunicação é de 1500 metros medidos em linha recta a uma potência que geralmente assume valores não superiores a 20dB (100mW) [28].

Pensado para operar numa rede, o ZigBee™ apresenta características dinâmicas de reconfiguração que permitem que sejam assumidas tipologias em malha de múltiplos nós com capacidade de se autodescobrir para garantir uma maior estabilidade da rede perante mudanças de condições ou falhas em nós individuais. Em qualquer rede deste protocolo há um coordenador que assume um papel de controlador central e que comunica directamente com o *end device*, o dispositivo periférico. É o coordenador o responsável pela inicialização,

distribuição de endereços, manutenção da rede e reconhecimento de todos os nós. O coordenador é um dispositivo FFD, *Full Function Device*, pela sua complexidade e Hardware (*HW*) específico que permite a implementação do protocolo. Um coordenador é por isso implementado em microcontroladores com não menos do que 32KB de memória de programa para poderem ser definidas as configurações da rede. O *end device*, por sua vez, corresponde ao dispositivo mais simples da rede e que consome menos energia, normalmente corresponde a sensores ou actuadores. Um *end device* é um RFD, *Reduced Function Device*, e por não serem necessários demasiados recursos em termos de protocolo pode ser implementado em microcontroladores com uma memória de programa próxima de 6KB [28]. No ZigBee™ existem dois estados distintos de operação. Qualquer dispositivo aliado a este protocolo pode operar em modo activo, no momento de envio e recepção de dados, ou num modo de baixo consumo. Quando não está a ser utilizado, um dispositivo que entra em baixo consumo, poupa recursos e prolonga a autonomia da bateria. Assim, falamos em transmissão com Rádio Farol, *Beaconing*, quando os dispositivos definidos como encaminhadores de dados (*routers*) transmitem durante períodos de tempo definidos, de forma a confirmarem a sua presença na rede, alertas de sinalização. Neste modo os restantes nós da rede, por terem sido configurados para os tempos em que é feita a sinalização do *router*, fazem coincidir o seu estado activo com o momento de sinalização, permanecendo em modo de baixo consumo no tempo restante. Por outro lado, sem transmissão Rádio Farol, no modo *Non-Beaconing*, a maior parte dos nós da rede permanecem activos.

A arquitectura deste protocolo de comunicação é baseada em quatro camadas físicas, Figura 4. A camada mais baixa da arquitectura, camada **PHY**, é a responsável pela transmissão e recepção de mensagens através de um dos 16 canais disponíveis de Rádio Frequência (RF). Esta camada é a responsável pela gestão da ligação e da sua qualidade, pela detecção e análise dos níveis de energia, pela selecção do canal para que a comunicação se estabeleça e ainda pela recepção e transmissão de pacotes através do meio físico. Segue-se a camada **MAC**, **MAC Layer**, que é a responsável pela sincronização do envio e recepção de pacotes de dados e da estrutura dos mesmos. É na **MAC Layer** que é feita a prevenção de colisões recorrendo a mecanismos de *CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance* ou seja, é feito um teste inicial à portadora de onda do canal de 2.4GHz antes de qualquer transmissão e para evitar colisões de pacotes é dado um tempo aleatório entre transmissões. As camadas referidas anteriormente são definidas pelas normas padrão IEEE 802.15.4, ao contrário das camadas seguintes, **Network and Security** e **Application**, que são definidas pela norma ZigBee™. A principal função da camada **Network and Security** é garantir a correcto funcionamento da camada inferior, a camada **MAC**, e a correcta integração desta com a camada superior, a camada das aplicações. A camada **Network and Security** está encarregue de estabelecer a ligação com novos dispositivos e criar novas redes. Assim, esta camada define o encaminhamento na rede e o emparelhamento com novos nós garantindo a segurança na troca de informação e na configuração dos dispositivos. Para gerir e assegurar o suporte das diversas aplicações é utilizada a camada **Application**. Esta camada contém três subcamadas: a Application Support Sublayer (APS), o ZigBee™ Device Object (ZDO) e a Application Framework (AF), [29] e [30].

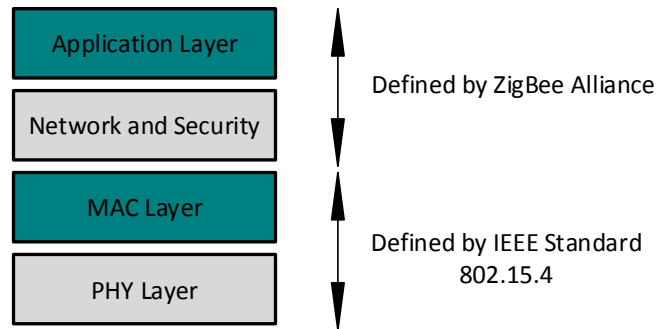


Figura 4: Arquitectura do protocolo de comunicação ZigBee™.

A descoberta de dispositivos pode acontecer de dois modos dependendo do conhecimento ou não do endereço de rede. Se este for conhecido a associação é directa por outro lado, se este não for conhecido é enviado um *broadcast*, ou seja um pedido que se difunde, através dos coordenadores e *routers* para todos os *end devices* da rede para que respondam com o respectivo endereço. À medida que a rede vai sendo descoberta e definida, os coordenadores da rede preenchem tabelas de correspondência onde associam a informação recebida aos respectivos nós da rede. Depois de ter sido feita a associação com os endereços da rede, o endereçamento pode ser indirecto, quando recorre a todos os intervenientes da rede (desde o coordenador ao *end device* específico), ou directo.

4.2. BLUETOOTH™ LOW ENERGY

Em 1994 surgiu, simbolicamente associada a um rei Dinamarquês reconhecido pela união de povos, uma nova tecnologia que se propunha, em analogismo ao rei, unir diferentes dispositivos e sectores industriais. O Bluetooth™ começou por distinguir-se pela sua rapidez e hoje, com a sua versão Low Energy, é reconhecido pelo baixo consumo.

O BLE é um sistema desenvolvido para aplicações específicas de comunicações que impliquem, não só a transmissão de pacotes de dados com um tamanho reduzido a cada instante, como também um baixo consumo de energia durante cada transmissão. Este, que corresponde a um sistema de tecnologia sem fios que a versão 4.0 do Bluetooth™ padrão permite, distingue-se do clássico pelo seu baixo consumo e excelentes prestações a um tamanho reduzido. Assim, permite uma excelente integração e cooperação com dispositivos electrónicos. A comunicação com outros dispositivos, sem ter que haver um envolvimento do utilizador, já que dinamicamente qualquer aparelho envia constantemente informações para estabelecer a comunicação com outros, aliada à não limitação do número de equipamentos ligados, torna o BLE ideal para aplicações de monitorização desportiva como a proposta. Em qualquer aplicação, a capacidade de transmissão depende essencialmente dos seguintes factores: modo e ambiente de operação, o *design* da antena, a orientação e encapsulamento do dispositivo e a potência de transmissão. No BLE a potência de transmissão pode ser configurada entre -30 e 0dBm, [31], e quanto maior o valor configurado, melhor a transmissão conseguida. No entanto, compromete o tempo de vida da bateria já que os consumos energéticos do dispositivo serão

mais elevados. Assim, havendo a possibilidade de configurar a potência para um alcance de até 30 metros em linha recta, deve haver um compromisso consciente para não afectar o tempo de vida do dispositivo e a recepção dos dados no receptor final. A transmissão de dados no BLE pode ser feita a uma velocidade máxima de 1 Mbit por segundo com uma taxa de transmissão de até 0.2 Mbits por segundo.

Os dispositivos que geralmente são utilizados para aplicações onde o baixo consumo de energia é crucial e dependente de pequenas baterias, são dispositivos *low energy*. Estes actuam em modo único e devem conseguir gerir de forma inteligente o baixo consumo de energia. Contrariamente, um computador ou um telemóvel são exemplos de dispositivos que funcionam em modo duplo. Uma vez inseridos na versão 4.0 podem comunicar tanto com dispositivos *low energy* como com dispositivos ligados à *stack*, ou seja ao módulo, do clássico Bluetooth™, considerados de BR/EDR (Basic Rate/Enhanced Data Rate), [32]. A diferença mais relevante entre o clássico BT e o BLE é essencialmente o baixo consumo energético do BLE que, segundo a especificação, necessita de só 10% do que necessita o BT para funcionar [33]. O BLE comparativamente ao BT foi desenvolvido para minimizar a necessidade de computação nos dispositivos. Assim, enquanto no BT se distinguem 28 tipos de pacotes diferentes, no BLE existem apenas dois formatos de pacotes. Por outro lado, o BLE tem 8 tipos de mensagens para controlar o momento de ligação, contrariamente o BT tem 75 mensagens distintas [33]. A Figura 5 representa a arquitectura configurada num dispositivo *low energy*. Esta corresponde à configuração mais comum e simples, sendo utilizada na maioria das aplicações e projectos BLE.

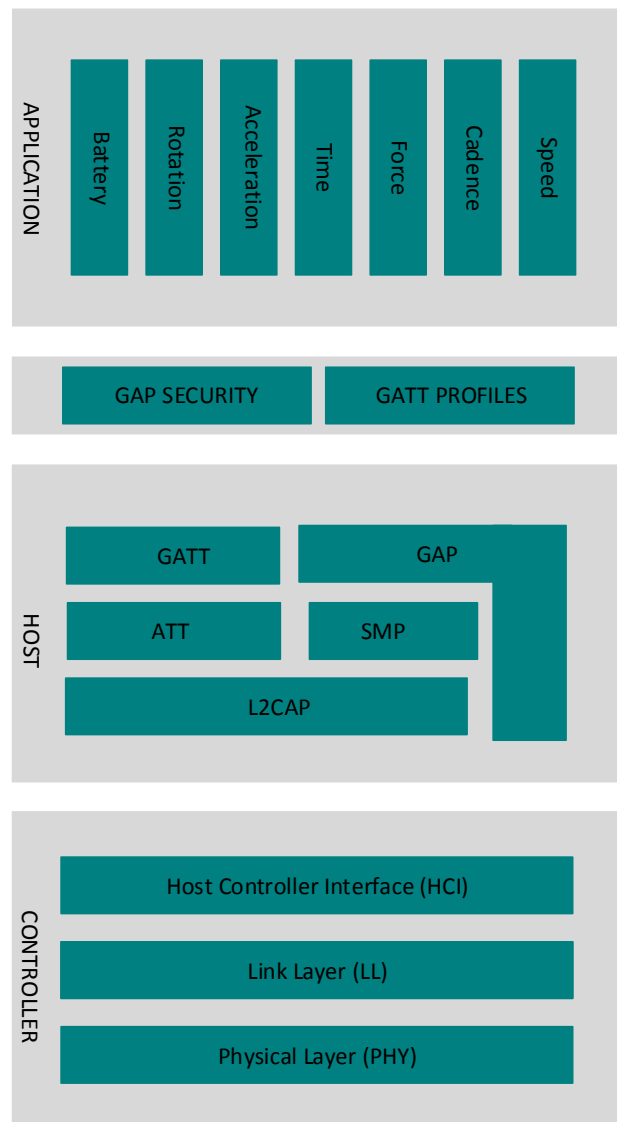


Figura 5: Arquitectura de um dispositivo *low energy*. [34]

A camada física, **PHY Layer**, corresponde ao canal de rádio frequências onde opera o BLE, 2.54GHz, a banda ISM (Industrial, Scientific and Medical radio band) reservada a equipamentos industriais, científicos e médicos. Esta é uma frequência não licenciada para aplicações e partilhada com o *Wi-fiTM* e as comunicações por microondas. Na banda de frequências admitida o BLE ocupa, para tráfego de dados, 40 canais de 2MHz de largura de banda [32]. O BLE ocupa 3 dos 40 canais para descobrir outros dispositivos e estabelecer novas ligações contrariamente ao clássico BT que recorre para o mesmo efeito a 32 canais. No BLE os restantes 37 canais são utilizados para a comunicação bidireccional entre dispositivos com ligação estabelecida. A duplicação da largura de banda, em relação à largura de banda do BT, deve-se ao aumento do índice de modulação do sinal que permitiu no BLE reduzir a potência de transmissão e aumentar a distância de alcance dos dispositivos. Além disso, o tamanho dos pacotes de dados foi reduzido para praticamente metade, do BT para o *Low Energy*, e no BLE é de 64 bits. Os módulos BluetoothTM evitam a interferência de outros sinais fazendo aquilo que se considera o *hopping* ou o “salto” para novas frequências depois de transmitir ou receber um

pacote de dados. Este processo, de saltar de portadora através de vários canais de frequência para evitar distúrbios no sinal, é conhecido como salto em frequência ou *frequency hopping*, [35]. O BLE tem vantagens no *frequency hopping* em relação a outros protocolos por ter a capacidade de alterar rapidamente de portadora, facilitado por pacotes de dados de tamanho reduzido, e por excluir do processo canais já ocupados, Figura 6.

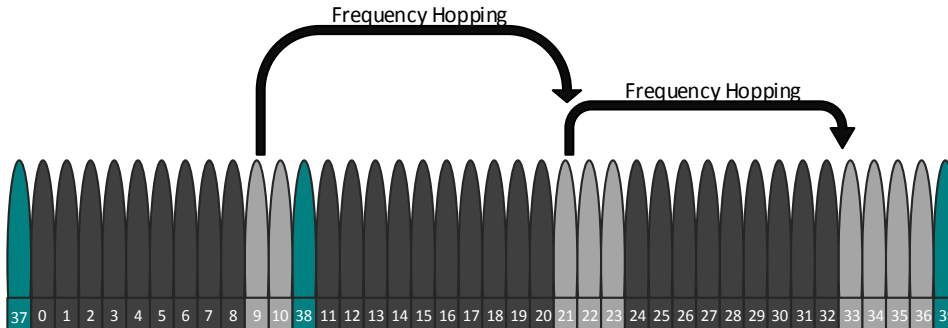


Figura 6: Frequency Hopping. [35]

Durante a comunicação entre dois dispositivos é utilizada a estrutura do *frequency hopping*. Assim, os dispositivos recebem e enviam dados num canal específico e havendo a necessidade de trocaram de portadora, para evitar os distúrbios anteriormente referidos, voltam a encontrar-se num outro canal. Mesmo que não exista troca de dados entre os dispositivos, uma vez emparelhados, os canais de *frequency hopping* são mantidos para permitir que a qualquer momento a troca de pacotes seja restabelecida.

A camada **Link**, **Link Layer (LL)**, controla os estados de rádio frequência que o dispositivo pode assumir. À semelhança de outros protocolos de comunicação, os dispositivos adoptam papéis distintos criando assim as condições para que uma ligação se estabeleça. Antes que qualquer ligação se inicie é necessário que dois dispositivos se descubram e demonstrem disponibilidade para comunicar. Em regras definidas, enquanto um dispositivo transmite de forma contínua e activa informação (*advertiser*), em pacotes de dados de 31 bytes, demonstrando a sua intenção de emparelhamento, o outro escuta e espera passivamente por quem manifeste a sua presença (*scanner*). A Figura 7 demonstra a execução de eventos de um *advertiser*.

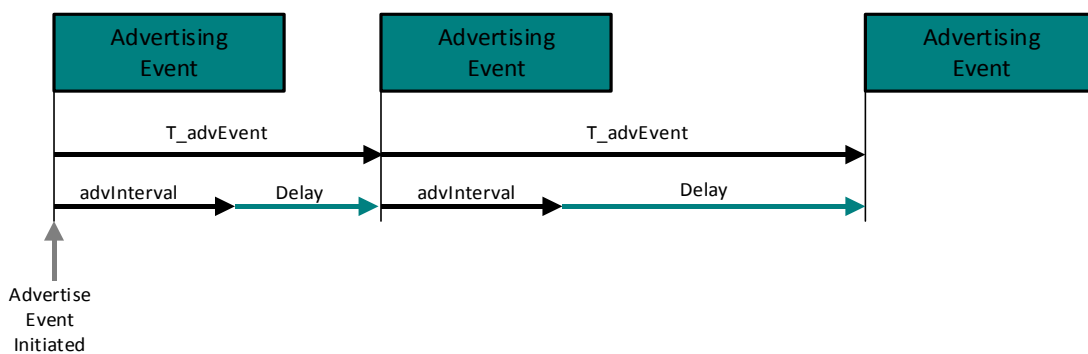


Figura 7: Comportamento de um advertiser no Protocolo BLE.[35]

Para um *advertiser* o tempo mínimo entre dois eventos corresponde ao intervalo de *advertising* que pode variar entre os 20ms e os 10.24s. A camada **Link** é responsável por, de forma aleatória, criar atrasos (*delays*) antes do próximo evento do *advertiser*, que podem ser no máximo de 10ms, de forma a impedir a concorrência de eventos de vários dispositivos. O tempo de um evento de *advertising* resulta do tempo efectivo de *advertising* e do *delay* estabelecido pela **LL**, [36].

Quando os dois se encontram, um *scanner* e um *advertiser*, e perante o interesse de comunicarem um com o outro, é enviado um pedido de ligação pelo scanner que passa a assumir o papel de iniciador, Figura 8. Segundo o protocolo se não houver necessidade de confirmação da ligação, o tempo de ligação entre um *scanner* e um *advertiser* é de aproximadamente 3ms, [36].

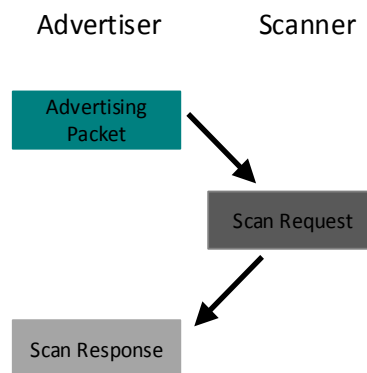


Figura 8: Scanner e advertiser. [35]

O iniciador assume, no momento da ligação, o papel de mestre e o dispositivo que aceitar o pedido de ligação torna-se escravo. O pedido enviado contém um conjunto de parâmetros para o dispositivo escravo sobre os canais para a ligação e o tempo requerido para a estabelecer. A Figura 9 apresenta um diagrama de estados possíveis que os dispositivos podem assumir até formarem entre si uma rede.

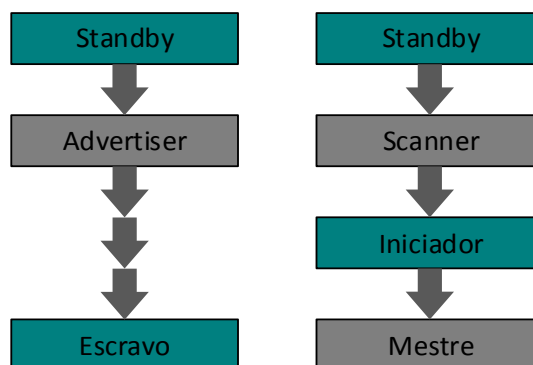


Figura 9: Diagrama de estados para obter uma rede BLE. [35]

A Figura 10 ilustra, na arquitectura em rede pensada com os três módulos sensoriais, a aplicação directa do protocolo antes e depois das comunicações serem estabelecidas.

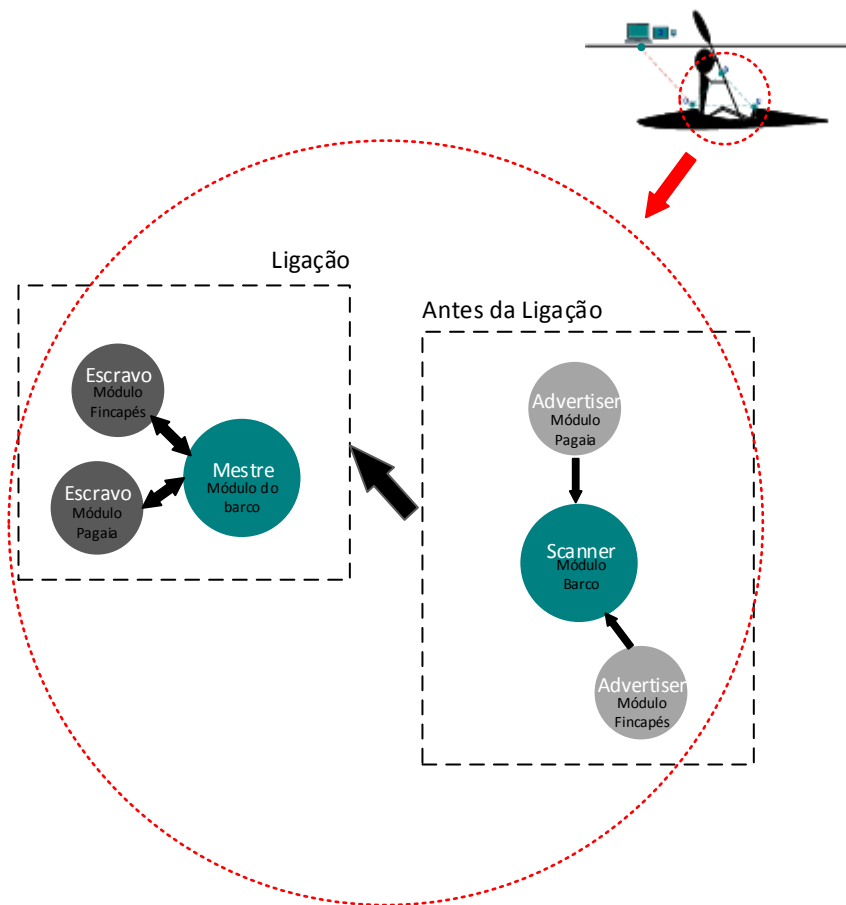


Figura 10: Arquitectura de ligação do sistema proposto no BLE.

A comunicação é estabelecida a partir do momento em que os primeiros dados são trocados em intervalos de tempo específicos, chamados de intervalos de ligação, múltiplos de 1.25ms e que devem variar entre os 7.5ms e os 4s de acordo com as Especificações do Bluetooth™ (BT Core Specifications), [37]. Dependendo da aplicação, o tempo requerido para os intervalos de ligação pode variar, o que traz vantagens e desvantagens associadas ao protocolo. Intervalos de ligação longos permitem uma redução nos consumos de energia, uma vez que o dispositivo está na maior parte do tempo em *standby* ou seja, em espera. Neste período, mesmo que exista a necessidade de trocar pacotes de dados, o modo de *standby* não pode ser interrompido e por isso é necessário aguardar por uma próxima ligação. Por outro lado, intervalos de ligação curtos aumentam a frequência de eventos de ligação. No entanto, é exigido um maior consumo de corrente. Se o dispositivo periférico não responder aos pacotes de dados enviados pelo dispositivo central no tempo máximo de supervisão da ligação, a ligação é perdida. Este tempo, que é considerado o *timeout* de supervisão da ligação, corresponde assim ao tempo máximo entre duas ligações estabelecidas com sucesso e é contabilizado em múltiplos de 10ms, tendo como valor mínimo 100ms e máximo 32s [33]. O *timeout* deve ser superior ao intervalo máximo de tempo entre eventos de ligação ou ao tempo de ligação efectiva. O tempo de latência do dispositivo periférico consiste no número máximo de eventos de ligação que o dispositivo pode

não considerar e varia entre 0 e 499. No entanto, existe uma limitação e não pode ser superior a 32s que corresponde ao *timeout* entre duas ligações.

O tempo de uma ligação efectiva é traduzido pela fórmula:

$$\begin{aligned} & \textit{Tempo de uma ligação efectiva} \\ & = (\textit{Intervalo de ligação}) \times (1 + (\textit{Tempo de latência do periférico})) \quad (4.1) \end{aligned}$$

Em 4.1. é considerado o tempo de latência máximo, isto é, que o periférico ignorou todos os eventos de ligação possíveis [33]. As referências a intervalo de ligação e tempo de latência, implicam o conhecimento do seguinte, Figura 11:

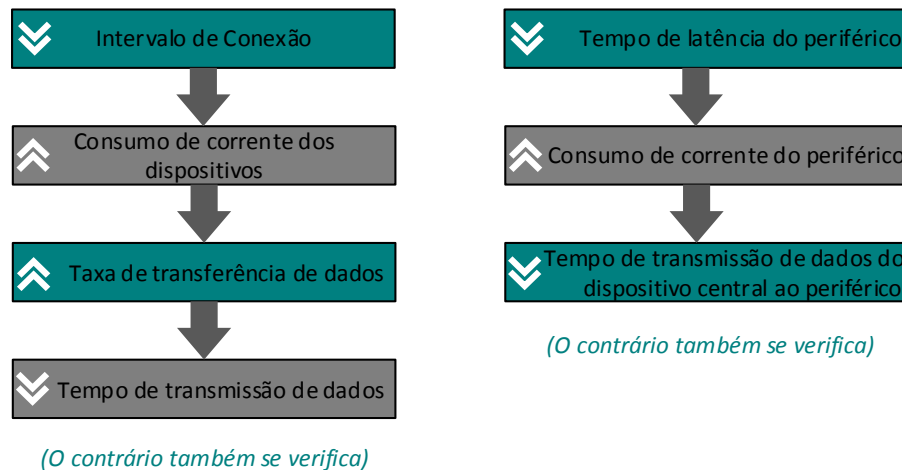


Figura 11: Cadeia de acontecimentos quando o intervalo de ligação e o tempo de latência do periférico são diminuídos/aumentados.

Sempre que se estabelece uma comunicação entre dispositivos, há informação que é guardada, como encriptações e autenticações, para salvaguardar e facilitar futuras ligações. Este processo é referido como *bondig process* ou seja permite o estabelecimento de ligações ao nível lógico.

A comunicação entre as camadas mais altas da *stack* do protocolo BT, no *Host*, e as camadas mais baixas, no *Controller*, é gerida pela camada **HCI**. A **HCI** é uma interface Bluetooth™ padrão preparada para enviar comandos depois da recepção de eventos, e para enviar e receber dados. Esta comunicação pode ser implementada numa interface de *software API* (Application Programming Interface) ou através de interfaces *HW* como *UART* (Universal Synchronous Receiver/Transmitter), *USB* ou *SPI* (Serial Peripheral Interface), [38]. O BLE reutiliza a especificação da camada **HCI** do Bluetooth™ e expande a mesma a comandos *Low Energy* [39]. Assim, um dispositivo que opera em modo único, só de acordo com o BLE, tem ao dispor uma lista de comandos HCI específicos do BLE. Por outro lado um dispositivo que funcione em modo duplo e aceite o BT e o BLE, pode implementar comandos de ambas as versões do protocolo. Um pacote de dados de um **comando HCI** tem um código próprio de 16 bit que o identifica, o *OpCode*. O *OpCode* é seguido de um parâmetro de 8 bit que especifica o tamanho total dos parâmetros que compõe o pacote, Figura 12.

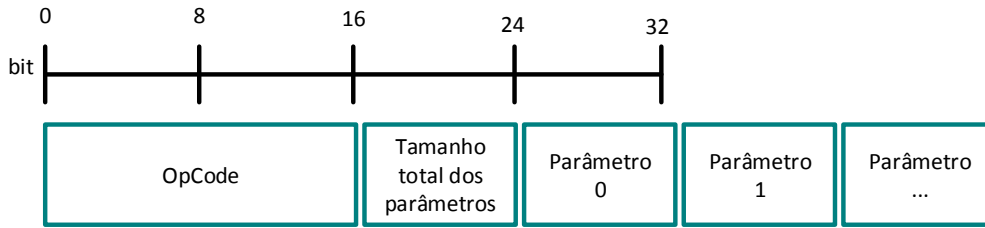


Figura 12: Formato de pacote de um comando HCI. [36]

Um pacote de um **evento HCI** tem um código de 8 bit que o identifica e que é único para eventos BLE [34]. Desta forma, é possível distinguir quando um evento HCI pertence ao BT ou quando é específico *Low Energy*. Se for dedicado ao BLE, para além do código do evento que define o evento como pertencente ao BLE, existe um código de um sub evento que identifica o exacto evento *Low Energy*, Figura 13. No protocolo é possível distinguir quando um comando ou evento é específico *Low Energy* pelo prefixo LE nas suas descrições.

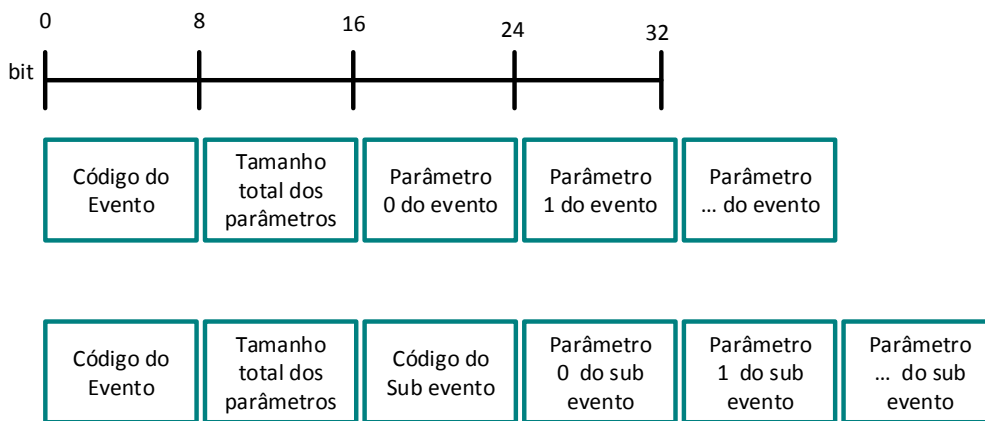


Figura 13: Formato de pacote de um evento HCI do Bluetooth™ (em cima); Formato de pacote de um evento HCI do BLE (em baixo). [36]

Numa aplicação, o funcionamento mais habitual é com um fluxo de dados do *Host* para o *Controller*. É o *Host* quem controla o canal de dados (*buffer*) do *Controller* para evitar colisões e o *overflow*, ou sobrecarga, do *buffer* com outros dados que possam coexistir.

Para além dos comandos e dos eventos referenciados na **HCI** definidos pela especificação BT, existe a possibilidade de utilizar na **HCI** comandos e eventos específicos e determinados pelos fornecedores como forma de integrar o dispositivo ou o produto por eles desenvolvido no protocolo. Assim, é permitido no BLE que os fornecedores possam definir os seus próprios perfis em casos de uso não abrangidos pela especificação. Uma analogia pode ser feita com a Texas Instruments™ que como fornecedora desenvolveu um conjunto de comandos e eventos HCI específicos para o funcionamento do SensorTag que é um módulo de desenvolvimento e de aplicação do BLE. Os comandos e eventos HCI serão abordados em pormenor nos Capítulos 6 e 7. Uma vez que nestes Capítulos é feita uma análise dos comandos e eventos HCI utilizados no BLE para a comunicação entre dispositivos. O estudo descrito nestes Capítulos é baseado no SensorTag e por isso, este módulo será novamente referido.

A **L2CAP** garante o serviço de encapsulamento de dados das camadas superiores. Já a camada **SM, SM Layer**, define e controla a segurança dos métodos de emparelhamento e é responsável pela distribuição de chaves de protecção. A camada **Generic Access Profile, GAP**, permite definir as funções, procedimentos e modos que possibilitam aos dispositivos a transmissão de dados e estabelecer e gerir ligações. Para além de estar directamente associada com a ligação a dispositivos e respectivos modos de funcionamento e perfis por eles assumidos, é a responsável pelas funcionalidades de segurança a cada ligação [34].

O BLE utiliza uma arquitectura estruturada em serviços e baseada num protocolo de atributos, o protocolo ATT. A comunicação entre dispositivos é gerida no *Host* pela camada **GATT, Generic Attribute Profile**. Nesta camada é aplicado o modelo de cliente e servidor, que não está relacionado com o papel que o dispositivo pode assumir na **LL**. Um dispositivo que na **LL** seja considerado mestre, ou central, pode ser cliente ou servidor GATT e o mesmo acontece para um dispositivo que na **LL** seja considerado escravo. Os papéis estabelecidos de acordo com a camada a **GATT** não são vinculativos mas uma forma estruturada de trocar dados entre dispositivos. Um servidor GATT guarda serviços estruturados em atributos e aceita, pelo protocolo ATT, comandos e pedidos do cliente GATT para aceder aos mesmos. Assim, um dispositivo para ser considerado servidor GATT recebe pedidos do cliente GATT e partilha como resposta, pelo protocolo ATT, dados organizados em atributos que podem ter um tamanho variável entre 1 e 23 bytes. Um dispositivo é cliente GATT quando envia pedidos, lê e escreve atributos de um servidor GATT [34]. Os atributos correspondem a dados endereçáveis no servidor GATT relativos ao próprio atributo como a sua estrutura, agrupamento e valor em si. Cada atributo é identificado por um número, reconhecido na norma como uma *Handle*, que é o valor pelo qual um atributo pode ser endereçável sempre que o cliente lhe quiser aceder. Por esse motivo, a *Handle* é única e atribuída de forma definida pelo servidor. Um atributo, tem ainda um tipo que identifica, de acordo com a norma, os dados que representa e é caracterizado por um *UUID* (Universal Unique Identifier) que é gerido pelo Grupo que zela pelos interesses do Bluetooth™, o SIG (Special Interest Group). Um *UUID*, que pode ser no BLE de 16 bit ou 32 bit, contém informação relativa ao tipo de dados associados aos atributos e é essencial para perceber o número de bytes de cada valor de um atributo [33]. O *UUID* não é único num dispositivo e pode haver mais do que um atributo com o mesmo *UUID*. A permissão de um atributo diz respeito à forma de acesso do cliente, em modo leitura e/ou escrita a um atributo do servidor. Um atributo contém ainda um valor que corresponde ao conteúdo de dados do próprio. Não existem restrições quanto ao tipo de dados que este pode ter no seu valor mas existe uma limitação de 23 bytes no seu tamanho [39]. A Figura 14 representa a tabela de atributos do SensorTag. Na figura é possível distinguir uma *Handle*, *UUID*, o valor do atributo e a sua permissão.

handle (hex)	handle (dec)	Type (hex)	Type (#DEFINE)	Hex / Text Value (default)	GATT Server Permissions	Notes
0x2B	43	0x2800	GATT_PRIMARY_SERVICE_UUID	0xAA10 (ACCELEROMETER_SERV_UUID)	GATT_PERMIT_READ	Start of Sensor Profile Accelerometer Service
0x2C	44	0x2803	GATT_CHARACTER_UUID	12 (properties: read/notify) 2D 00 (handle: 0x002D) 11 AA (UUID: 0xAA11)	GATT_PERMIT_READ	
0x2D	45	0xAA11	ACCELEROMETER_DATA_UUID	00:00:00 (3 bytes)	GATT_PERMIT_READ	X : Y : Z Coordinates
0x2E	46	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)	GATT_PERMIT_READ GATT_PERMIT_WRITE	Write "01:00" to enable notifications, "00:00" to disable
0x2F	47	0x2901	GATT_CHAR_USER_DESC_UUID	"Accel. Data" (14 bytes)	GATT_PERMIT_READ	
0x30	48	0x2803	GATT_CHARACTER_UUID	0A (properties: read/write) 31 00 (handle: 0x0031) 12 AA (UUID: 0xAA12)	GATT_PERMIT_READ	
0x31	49	0xAA12	ACCELEROMETER_CONF_UUID	1 (1 byte)	GATT_PERMIT_READ GATT_PERMIT_WRITE	Write "01" to start Sensor and Measurements, "00" to put to sleep
0x32	50	0x2901	GATT_CHAR_USER_DESC_UUID	"Accel. Conf." (15 bytes)	GATT_PERMIT_READ	
0x33	51	0x2803	GATT_CHARACTER_UUID	0A (properties: read/write) 34 00 (handle: 0x0034) 13 AA (UUID: 0xAA13)	GATT_PERMIT_READ	
0x34	52	0xAA13	ACCELEROMETER_PERI_UUID	1 (1 byte)	GATT_PERMIT_READ GATT_PERMIT_WRITE	Period = [Input*10] ms, default 1000 ms, lower limit 100 ms
0x35	53	0x2901	GATT_CHAR_USER_DESC_UUID	"Acc. Period" (12 bytes)	GATT_PERMIT_READ	

Figura 14: Tabela de atributos do SensorTag com identificação dos atributos e propriedades.

[40]

Um grupo de atributos define uma característica e um conjunto de características define um serviço. O primeiro atributo de cada serviço corresponde à sua declaração. Um serviço é composto por características e cada característica contém um valor e uma descrição. É no valor que estão os dados actuais da característica que podem ser lidos ou escritos, e na descrição, que é opcional, onde se encontra informação adicional sobre o valor. As Figura 15 e 16 ilustram, na tabela de atributos do SensorTag, o conceito de característica e a forma como é constituída.

handle (hex)	handle (dec)	Type (hex)	Type (#DEFINE)	Hex / Text Value (default)	GATT Server Permissions	Notes
0x2B	43	0x2800	GATT_PRIMARY_SERVICE_UUID	0xAA10 (ACCELEROMETER_SERV_UUID)	GATT_PERMIT_READ	Start of Sensor Profile Accelerometer Service
0x2C	44	0x2803	GATT_CHARACTER_UUID	12 (properties: read/notify) 2D 00 (handle: 0x002D) 11 AA (UUID: 0xAA11)	GATT_PERMIT_READ	Característica 0
0x2D	45	0xAA11	ACCELEROMETER_DATA_UUID	00:00:00 (3 bytes)	GATT_PERMIT_READ	
0x2E	46	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)	GATT_PERMIT_READ GATT_PERMIT_WRITE	Característica 1
0x2F	47	0x2901	GATT_CHAR_USER_DESC_UUID	"Accel. Data" (14 bytes)	GATT_PERMIT_READ	
0x30	48	0x2803	GATT_CHARACTER_UUID	0A (properties: read/write) 31 00 (handle: 0x0031) 12 AA (UUID: 0xAA12)	GATT_PERMIT_READ	Característica 1
0x31	49	0xAA12	ACCELEROMETER_CONF_UUID	1 (1 byte)	GATT_PERMIT_READ GATT_PERMIT_WRITE	
0x32	50	0x2901	GATT_CHAR_USER_DESC_UUID	"Accel. Conf." (15 bytes)	GATT_PERMIT_READ	Característica 1
0x33	51	0x2803	GATT_CHARACTER_UUID	0A (properties: read/write) 34 00 (handle: 0x0034) 13 AA (UUID: 0xAA13)	GATT_PERMIT_READ	
0x34	52	0xAA13	ACCELEROMETER_PERI_UUID	1 (1 byte)	GATT_PERMIT_READ GATT_PERMIT_WRITE	Característica 1
0x35	53	0x2901	GATT_CHAR_USER_DESC_UUID	"Acc. Period" (12 bytes)	GATT_PERMIT_READ	

Figura 15: Tabela de atributos do SensorTag com identificação das características que definem um determinado serviço. [40]

Sensor Tag Application: Complete Attribute Table						
TI Base UUID: F0000000-0451-4000-B000-00000000xxxx. 128-bit UUIDs are typed 'bold'						
handle (/hexL)	handle (/hex)	Type (/hexL)	Type (#DEFINE)	Hex / Text Value (default)	GATT Server Permissions	Notes
0x2B	43	0x2800	GATT_PRIMARY_SERVICE_UUID	0xAA10 (ACCELEROMETER_SERV_UUID)	GATT_PERMIT_READ	Start of Sensor Profile Accelerometer Service
0x2C	44	0x2803	GATT_CHARACTER_UUID	12 (properties: read/notify) 2D 00 (handle: 0x002D) 11 AA (UUID: 0xAA11)	GATT_PERMIT_READ	
0x2D	45	0xAA11	ACCELEROMETER_DATA_UUID	00:00:00 (3 bytes)	GATT_PERMIT_READ	X : Y : Z Coordinates
0x2E	46	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)	GATT_PERMIT_READ GATT_PERMIT_WRITE	Write "01:00" to enable notifications, "00:00" to disable
0x2F	47	0x2901	GATT_CHAR_USER_DESC_UUID	"Accel. Data" (14 bytes)	GATT_PERMIT_READ	

→ Declaração do serviço
 → Declaração da característica
 → Característica
 → Valor da característica
 → Descrição da característica

Figura 16: Tabela de atributos do SensorTag com identificação dos elementos que constituem uma característica, o seu valor e descrição. [40]

Fazendo uma analogia, no módulo da pagaia, o acelerómetro corresponde a um serviço. Os dados por ele adquiridos, a sua própria configuração e o período com que é definido são exemplos de características deste serviço. Cada uma das características é definida por um valor e pode ou não ter associada uma descrição. Quando um cliente GATT pretende aceder a um atributo do servidor é enviado um pedido GATT, um comando, que é respondido por um evento GATT. Aceder a um atributo implica para ao cliente GATT conhecer a *Handle* característica do atributo através da qual é feito o endereçamento. Na Figura 17 está representada a hierarquia de dados e atributos do serviço acelerómetro de acordo com a descrição anterior.

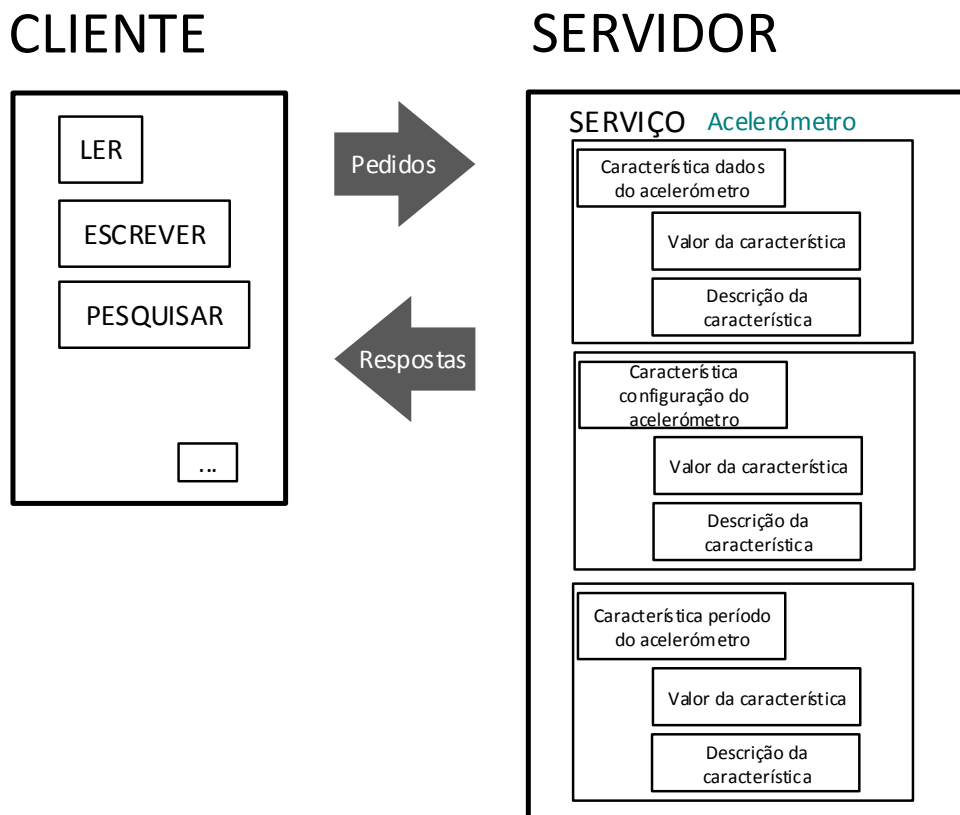


Figura 17: Hierarquização dos dados e dos atributos do serviço acelerómetro.

4.3. ZIGBEE ou BLUETOOTH™ LOW ENERGY?

Independentemente do protocolo a considerar, qualquer arquitectura de uma rede de sensores sem fios consiste tipicamente numa baixa taxa de transmissão de dados, a consumos reduzidos de energia, por nós com capacidade sensorial que transmitem periodicamente os dados para um nó central. No estudo proposto, para a ferramenta apresentada, optou-se pela utilização do BLE. Esta escolha foi justificada por diversos factores, entre eles o baixo consumo energético do protocolo. Em [41] faz-se um paralelismo entre os consumos energéticos de ambos os protocolos. Para que pudessem ser retiradas conclusões, o estudo mencionado refere as condições necessárias a estabelecer para uma configuração semelhante em ambos. Assim, foi estabelecido para cada transmissão de 0dBm um pacote de dados de 8 bytes. Optou-se por desabilitar a encriptação nas análises realizadas e foi utilizada uma fonte de alimentação de 3.3V. Os testes foram feitos em ambiente de interior a uma distância entre o mestre e o escravo de 30cm. A análise dos resultados permitiu concluir que um dispositivo BLE, no modo de poupança de energia, consome $0.78\mu\text{A}$ enquanto um dispositivo ZigBee™ consome, nas mesmas condições, três vezes mais. No modo activo, e para as mesmas circunstâncias de funcionamento, o consumo para o BLE é de 4.5mA e o do ZigBee™ de 9.3mA. A eficiência de uma ligação entre o mestre e o escravo é um dos processos que mais pode comprometer o consumo de energia numa aplicação. Uma ligação é reestabelecida sempre que dois ou mais nós perdem a ligação no final de uma transmissão e está directamente relacionada com o *duty cycle* da comunicação. Este é definido pela relação entre o tempo médio no modo activo e o tempo total de um ciclo. Um protocolo com um *duty cycle* maior significa que demora mais tempo a restabelecer uma ligação. Para reestabelecer a ligação entre o mestre e o escravo o BLE necessita de 1.15s e o ZigBee™ de 0.25s. Estes valores tão díspares são explicados pelo facto do BLE entre envios de pacotes entrar em modo de baixo consumo por mais tempo melhorando o seu ciclo de trabalho e consumo. Assim, o estudo determina que mais importante do que o consumo de corrente na eficiência de um protocolo, é o tempo que um dispositivo se consegue manter em baixo consumo energético. No estudo descrito, entre o envio de pacotes, o BLE mantém-se nesse modo por mais tempo. É importante referir que o presente estudo não deve ser generalizado na medida em que o consumo de corrente dos dois protocolos pode variar de acordo com as condições em que forem realizados os testes, desde o ambiente, interior ou exterior, à variação do tamanho do pacote de dados, passando pelos parâmetros de configuração do nó central e das distâncias mantidas entre o transmissor e o receptor.

Para além de ser energeticamente mais eficiente, houve outras características importantes do BLE a serem tidas em conta na escolha do protocolo a implementar. Com o Bluetooth™ é possível ter várias redes com um número ilimitado de nós a coexistirem num mesmo local sem troca de dados e informação. Esta particularidade é relevante para que numa competição de canoagem possam ser adquiridos dados de vários atletas, da mesma ou de outras tripulações, em simultâneo. Além disso, o BLE foi concebido para ser aceite pela maioria dos dispositivos electrónicos modernos. Uma vez que os dispositivos terminais utilizam BT é com facilidade

que se pode estabelecer a ligação e enviar dados para a um *tablet*, um computador ou um telemóvel.

CAPÍTULO 5 - ARQUITECTURA PROPOSTA PARA O SISTEMA DE MONITORIZAÇÃO

Depois de definida a arquitectura, os módulos de sensores que dela fazem parte e o protocolo de comunicação que a estrutura, estão reunidas as condições para descrever em concreto a ferramenta em estudo.

O Capítulo que se segue, começa por demonstrar a arquitectura proposta para a rede de sensores e aborda posteriormente duas versões da ferramenta. A primeira versão é relativa à implementação de uma rede sensorial e é abordada neste Capítulo a respectiva prova de conceito e resultados associados. A segunda versão descrita foi pensada para o estudo da implementação do BLE na rede.

5.1. ARQUITECTURA PROPOSTA

Uma possibilidade para a rede sensorial sem fios é a apresentada na Figura 18. A rede é constituída por módulos de sensores colocados nos locais onde é essencial adquirir a informação, nomeadamente na pagaia, no barco e no finca-pés. Cada um destes módulos, que é um nó na rede estabelecida, comunica em rádio frequência por BLE e os dados adquiridos por cada módulo individualmente são agregados num nó central, o módulo do barco. De forma a garantir a chegada dos dados, reunidos no módulo central, à margem foi implementado um protocolo de longas distâncias. Por sua vez, na margem, o BLE volta a ser necessário para que os dados sejam visualizados em dispositivos terminais como um *tablet* ou um telefone.

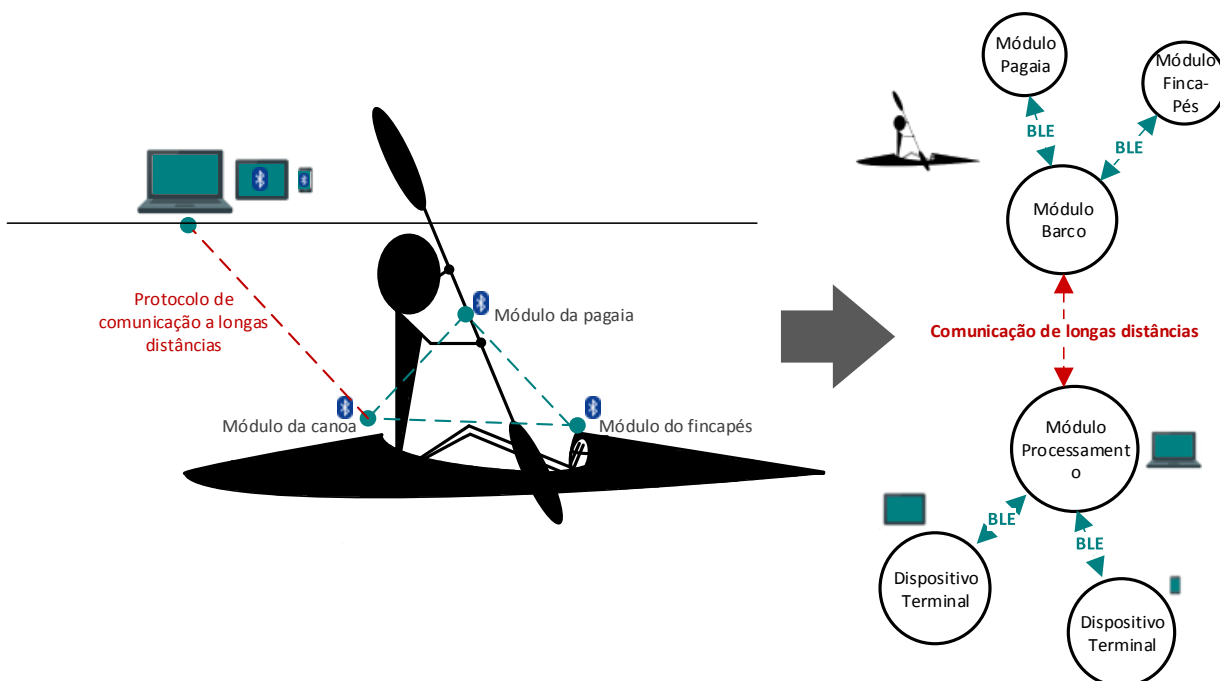


Figura 18: Arquitectura em rede proposta para a ferramenta em estudo.

5.2. VERSÃO 1.0.

Na versão 1.0. foram desenvolvidos os protótipos do módulo da pagaia e do barco para testar a aquisição de dados e a recepção de informação na margem. Na Figura 19 é apresentada a versão 1.0. da ferramenta.



Figura 19: Versão 1.0. com os respectivos módulos sensoriais. Observação: o módulo do finca-pés ainda não tinha sido desenvolvido, contendo a Figura uma representação simbólica de um sensor de força.

5.2.1. Módulo da Pagaia

O módulo da pagaia é o módulo de maior complexidade electrónica. Este módulo foi desenvolvido para adquirir dados objectivos da pagaia como a cadência, a aceleração e a força, e para distinguir e analisar os diferentes momentos de uma pagaiada.

Optou-se pela utilização da *launchpad* da Texas InstrumentsTM, uma TI MSP430G2553, por facilitar a ligação *USB* e por conter uma ferramenta de emulação e depuração de erros, associada à interface com o utilizador, e uma ferramenta de desenvolvimento e programação IDE (Integrated Drive Electronics), o IAR Embedded WorkbenchTM. A *launchpad* utiliza um MPU (Microprocessor Unit) de baixo consumo da Texas InstrumentsTM, nomeadamente um MSP430. Um MPU consiste numa unidade de processamento computacional caracterizado por ter um baixo factor de forma e consumo devido à reduzida velocidade de processamento, ao baixo nível de programação e ainda ao número limitado de interfaces físicas. Geralmente um MPU é utilizado, pelas características que apresenta, em aplicações limitadas e com baixo processamento de dados. Para processar informação de maior complexidade opta-se por uma de duas soluções: pela utilização de uma outra unidade de processamento, como uma FPGA (Field Programmable Array) que suporta a implementação de circuitos lógicos relativamente grandes e complexos, ou por uma situação repartida em que parte do processamento fica a cargo do MPU e o mais exigente do lado da FPGA. A *launchpad* de 16 bit possui 8 canais

conversores analógicos para digital (ADC - Analog Digital Converter) de 10 bit e outros componentes electrónicos como LEDs (Light Emitting Diodes), botões e pinos I/O prontos a serem utilizados. Para alimentar a *launchpad* e o circuito periférico foram utilizadas duas baterias de 1.5V.

Para a aquisição de dados foi utilizado um SMB380, um acelerómetro de três eixos que permite a medição da aceleração em eixos perpendiculares. O acelerómetro de baixo consumo, que pode ser configurado para operar com gama de $\pm 2g$, $\pm 4g$ ou $\pm 8g$, fornece à saída um sinal de 10 bit e tem uma largura de banda que varia entre os 25 e 1500Hz. Tal como referido anteriormente, de acordo com o momento de execução de uma pagaiada podem ser atingidos valores de força variáveis entre 250 e 400N. Desta forma, por serem esperados valores de aceleração elevados, o acelerómetro para o módulo da pagaia foi configurado para o seu máximo, $\pm 8g$. O acelerómetro escolhido tem três modos de funcionamento úteis para garantir consumos controlados de energia. No modo normal de operação, que está idealizado para consumir $200\mu A$ de corrente, dados do estado do acelerómetro, do controlo e da memória EEPROM (Electrically Erasable Programmable Read-Only Memory) podem ser lidos e alterados. No modo de baixo consumo, de $1\mu A$, não é possível realizar qualquer comunicação com o sensor, incluindo comandos de leitura e escrita, e por isso não é adquirido qualquer valor de aceleração. O acelerómetro pode ser ainda configurado para o modo “wake-up” que consiste na possibilidade de, com uma interrupção, activar o acelerómetro perante um determinado valor de aceleração previamente definido. Assim, o acelerómetro mantém-se em modo de baixo consumo e, de acordo com um período definido pelo utilizador, avalia os dados de aceleração. Desta forma, é possível alterar o modo de actividade do sistema [23]. A Figura 20, mostra como é gerida a aquisição e estruturado o pacote de dados de aceleração. Quando são adquiridos três novos valores para x,y e z é gerada uma interrupção, ou seja, é feita uma actualização aos valores de aceleração da última série lida. A interrupção ocorre, como é possível verificar pela Figura 20, no final da leitura do valor proveniente do eixo Z. Qualquer interrupção gerada por engano, sempre que a leitura de valores de aceleração estiver em execução, é apagada através de um *reset*. Assim, é garantida a sequência de três valores de aceleração correspondentes ao mesmo movimento, sem que haja distúrbios no pacote de dados enviado e recebido.

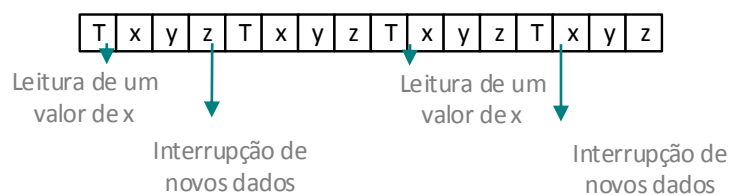


Figura 20: Aquisição de dados, SMB380. [23]

A comunicação RF para curtas distâncias por ser comum a outros módulos será abordada individualmente neste Capítulo.

Idealmente, este módulo e mais tarde na versão final do sistema, será colocado no interior do tubo de uma pagaia para que não interfira na prática do atleta, utilizando para esse efeito um encaixe central que as próprias pagaias têm para facilitar o seu transporte. No entanto, esta ou outra perspectiva ponderada de *design* só serão tidas em conta depois da estabilização do

desenvolvimento do próprio módulo uma vez que, para efeitos de teste, é vantajoso ter o módulo acessível. Um protótipo do módulo da pagaia é representado na Figura 21.

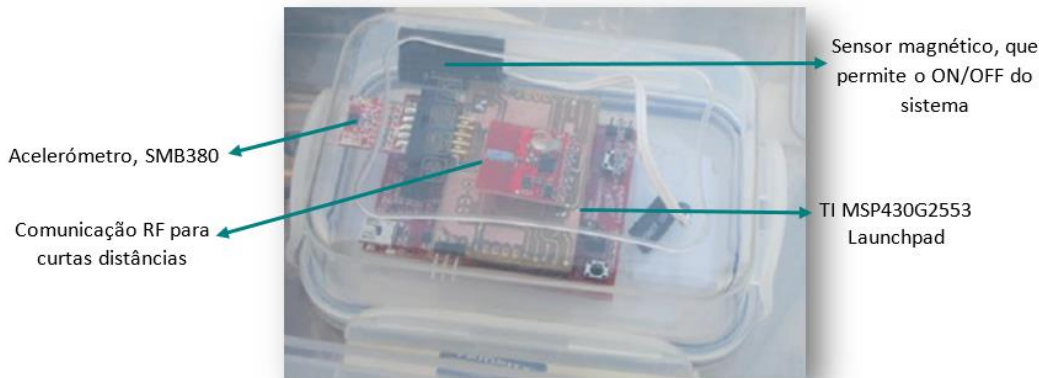


Figura 21: Protótipo do módulo da pagaia desenvolvido.

O módulo da pagaia pode ser traduzido no diagrama de blocos ilustrado na Figura 22. O sensor magnético foi incluído no módulo para facilitar o atleta ligar e desligar o sistema em momento de testes dentro de água. Sendo esta uma forma prática de o fazer mantendo a estanquicidade da caixa que envolve os sensores. De acordo com o diagrama apresentado, na Figura 22, o MSP430 acede ao acelerómetro e ao módulo RF de curtas distâncias através da UART 1 configurada como SPI. O protocolo SPI é um protocolo de comunicação série entre microcontroladores e periféricos. No SPI os dados são enviados e recebidos em canais distintos, modo *full-duplex*, e são definidos para a comunicação quatro sinais lógicos. O Chip Select (CS) e o clock (CLK) são canais lógicos de controlo e o MOSI (Master Output, Slave Input) e MISO (Master Input, Slave Output) canais lógicos de dados, Figura 23. A comunicação SPI suporta velocidades superiores a 10 Mbps e para que pacotes de dados (tipicamente pacotes de 8 bits) sejam trocados entre o microcontrolador e o periférico é necessário definir no início da comunicação o modo de operação, a fase e polaridade do relógio, a ordem de transmissão e a frequência do relógio. Na definição do modo de comunicação o microcontrolador assume o papel de mestre e o periférico de escravo.

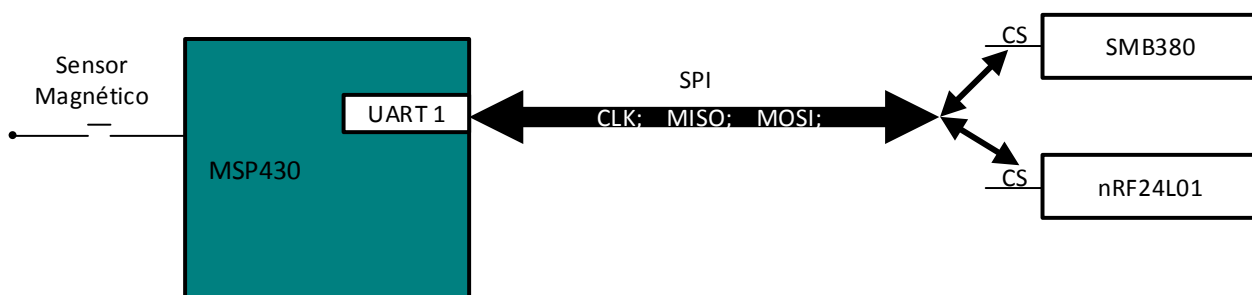


Figura 22: Diagrama de blocos que traduz o módulo da pagaia.

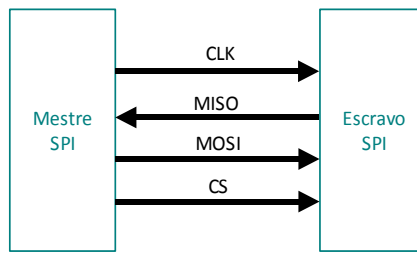


Figura 23: Modelo de Comunicação SPI. [38]

Para o módulo da pagaia, a fase do relógio foi definida com a colocação de dados no primeiro flanco do relógio e a leitura de dados no segundo flanco. A polaridade do *clock* é com o zero activo e a frequência do *clock* é a máxima permitida pelo protocolo, 4 MHz. O MSP liga-se ao SM380 e ao nRF24L01 separadamente pela linha de controlo digital, CS, que altera o estado do módulo. Assim, para que o MSP430 se ligue a um dos módulos, o outro deve estar desligado. O Chip Select, tanto no acelerómetro como no módulo RF, está ligado aos pinos I/O do circuito integrado dos dispositivos e por isso activar e desactivar o CS está directamente associado com o ligar/desligar do dispositivo.

A TI MSP430G2553, como referido anteriormente, contém uma interface *USB* que permitiu com o IAR Embedded WorkbenchTM a programação do circuito integrado MSP430. Assim, com esta ferramenta foi definido o modo de funcionamento do microcontrolador de acordo com a máquina de estados ilustrada na Figura 24. O estado “Intro” corresponde ao primeiro estado da máquina de estados. Neste estado é feita a configuração do MPU, da UART, da interface SPI e dos temporizadores (*timers*). A configuração dos periféricos, do acelerómetro e do módulo de comunicação a curtas distâncias, é feita no estado seguinte “ConfigPeriféricos”. A transição do estado “Intro” para o estado “ConfigPeriféricos” é feita, como todas as outras transições de estados, através de interrupções do *timer*. Desta forma, o microcontrolador entra no estado “sleep” em função uma interrupção do temporizador. A leitura do acelerómetro é periódica e é em função do tempo definido no *timer*. Os valores lidos são escritos no módulo RF de curtas distâncias para serem enviados para o módulo do barco. O MSP430 entra no estado “sleep” onde permanece até nova interrupção do *timer*.

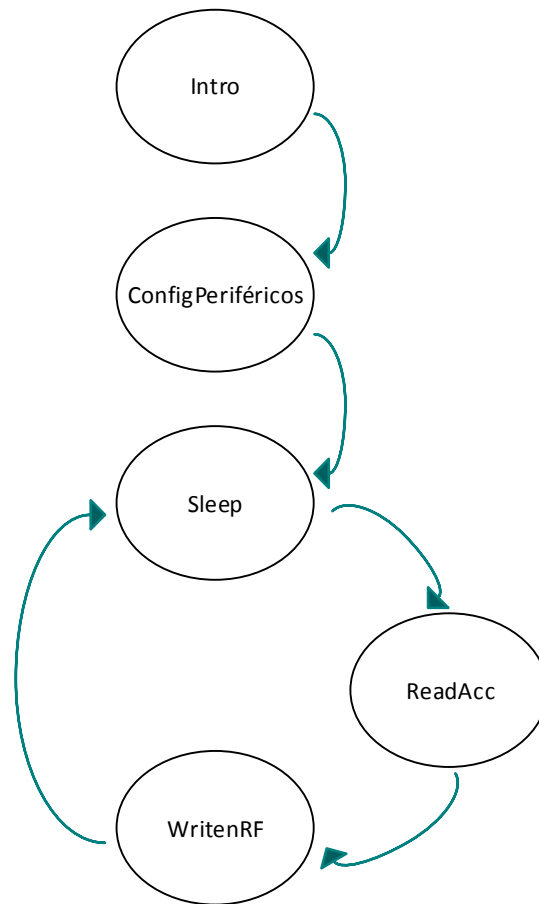


Figura 24: Máquina de estados do MSP430 para o módulo da pagaia.

5.2.2. Módulo do barco

O módulo do barco é aquele que permite adquirir valores relativos à oscilação e velocidade do barco em relação à água. Assim, para este efeito, o módulo contém um SMB380, associado a um TI MSP430G2553 e uma bateria para alimentação do módulo. Este módulo corresponde ao nó central, que incluído na rede de sensores, recolhe os dados provenientes dos restantes módulos e os envia para a margem para serem analisados e processados em tempo real. Por este motivo e por funcionar como um *gateway* (ponte de ligação) de comunicações, este módulo deve assumir os dois tipos de comunicação, a curta e a longa distância. A curta distância, através de um módulo RF, para recolher dados dos restantes módulos da rede, e a longa distância para enviar os dados recolhidos para a margem. A Figura 25 apresenta o protótipo desenvolvido para o módulo do barco.

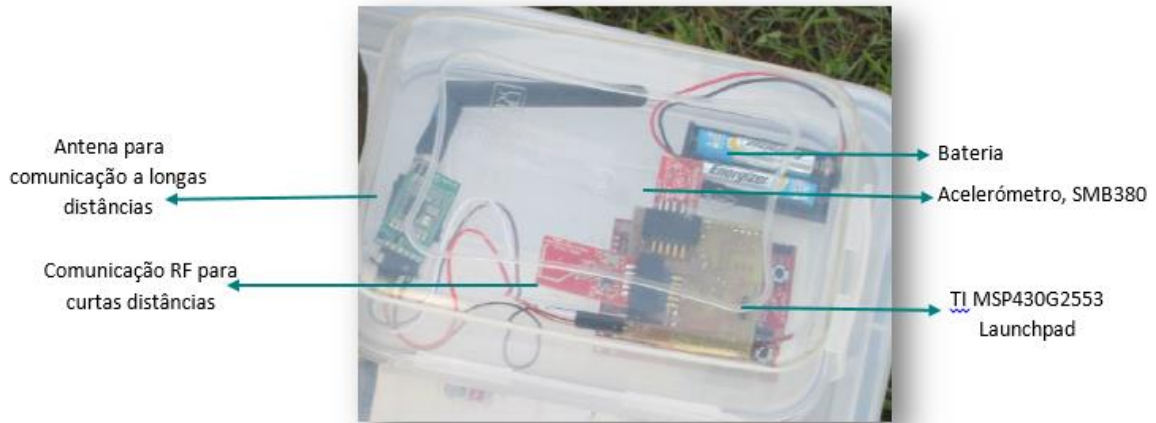


Figura 25: Protótipo desenvolvido para o módulo do barco.

O acelerómetro utilizado neste módulo é semelhante, em *HW* e funcionamento, ao utilizado no módulo da pagaia e por isso não será repetida informação. Pelo mesmo motivo, ser comum ao módulo da pagaia, a comunicação a curtas e longas distâncias será abordada da secção 5.2.3 deste Capítulo.

O módulo do barco é representado pelo diagrama de blocos da Figura 26. O MSP430 deste módulo está configurado para 2 UARTs, uma configurada como SPI e outra para o estado “WriteUART” através dos pinos de transmissão, Txd, e recepção de dados, Rxd. A ligação ao SMB380 e ao nRF24L01 através da interface SPI foi configurada, no MSP430, da mesma forma que no módulo da pagaia.

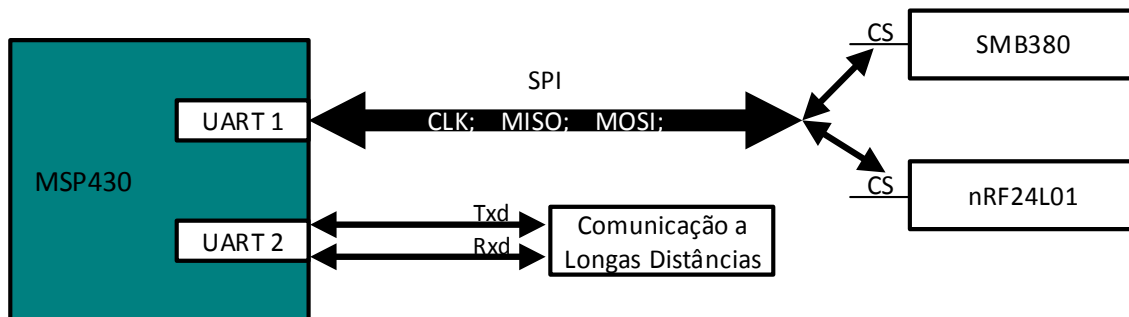


Figura 26: Diagrama de blocos que traduz o módulo do barco.

A Figura 27 ilustra a máquina de estados do módulo do barco. Os estados “Intro” e “ConfigPeriféricos” são semelhantes aos estados “Intro” e “ConfigPeriféricos” da máquina de estados do módulo da pagaia. Depois de configurados os periféricos, o microcontrolador entra “sleep” até que uma interrupção do *timer* seja gerada. Em função do tempo definido no temporizador, ocorre uma interrupção que permite a leitura dos dados do módulo de curtas distâncias relativos ao módulo da pagaia. É com uma nova interrupção que são lidos os valores de aceleração do acelerómetro do barco. Os dados recebidos pelo módulo da pagaia juntamente com os dados lidos do acelerómetro do barco são enviados para o módulo de recepção de dados, na margem do rio, através do estado “WriteUART”.

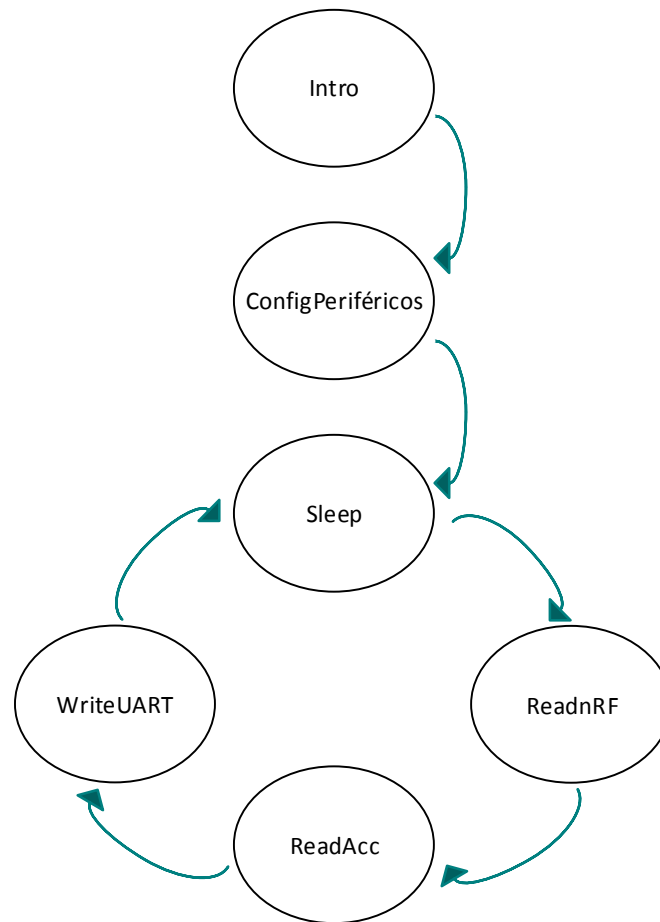


Figura 27: Máquina de estados do MSP430 para o módulo do barco.

5.2.3. Comunicação RF para curtas distâncias e comunicação a longas distâncias

Para a comunicação a curtas distâncias, do módulo do barco e da pagaia, foi utilizado um circuito RF, nomeadamente um nRF24L01. O nRF24L01 é um dispositivo de transmissão a 2.4GHz especialmente desenhado para aplicações sem fios de baixo consumo. O nRF foi configurado para operar à largura de banda ISM que inclui as frequências de 2.4 a 2.4835GHz. Associado ao circuito existe um protocolo de comunicação, Enhanced ShockBurst™, que se baseia na comunicação por pacotes a partir de FIFOs internos, [25]. Desta forma é assegurado um fluxo de dados entre o sistema de rádio e o sistema de processamento onde, o primeiro dado a entrar no sistema rádio é o primeiro a sair para o sistema de processamento (FIFO - First In First Out). A velocidade de transmissão de dados suportada pelo nRF24L01 é de 2Mbps e pode ser verificada nos 126 canais de RF que o circuito contém. Para manter a razoabilidade e impedir problemas de compatibilidade o circuito em questão é, à semelhança de outros módulos adquiridos, da Texas Instruments™. Em *standby* consome 22µA e desligado 900nA [25]. Quando o modo RX (Receiver mode ou modo de recepção) está activo o circuito funciona como um receptor de pacotes que coloca os dados recebidos no seu registo de recepção. É feita uma procura constante por pacotes válidos e sempre que um pacote válido é encontrado é feito um RX FIFO. O nRF24L01 mantém-se no modo RX até que o MPU o configure para o modo

standby ou para desligar. No modo RX é activo um detector de sinal, *CSMA/CA*, que detecta quando um sinal RF está na gama de frequências configurada e evita colisões nas comunicações. No modo TX (Transmitter modo ou modo de transmissão) o nRF24L01 funciona como um transmissor de pacotes e permanece neste modo até que um pacote de dados seja transmitido. Se o TX FIFO estiver vazio, o módulo permanece neste modo até que o pacote seja preenchido com informação. De outra forma, se estiver completo, o nRF24L01 passa para o modo *standby*. É importante que cada transmissão não demore mais do que 4ms [25].

A comunicação a longas distâncias é importante para que possa haver a transmissão dos dados adquiridos em tempo real, pelos módulos de aquisição de dados, para unidades de processamento. Para estabelecer esta comunicação foi utilizado um módulo comercial de longas distâncias funcional a 433MHz com um adaptador *USB* para interface série. Este módulo de baixo consumo está especificado para operar de 3.3V a 5.5V, a uma potência de transmissão de 1200bps a mais de 1000 metros.

Em situações de treino e competição a distância máxima que o atleta se afasta da margem é, tendo em conta uma pista comum da modalidade, de 2km. A total liberdade e despreocupação do atleta na prática desportiva são importantes. Para não comprometer esses factores nem o processamento dos dados, foi desenvolvido um sistema com arquitectura própria para a comunicação a longas distâncias. O ambiente onde a canoagem é praticada, pelas características que apresenta, é só por si limitativo. Assim, estabelecendo um compromisso entre a quantidade de dados recebidos, a distância pretendida, a taxa de transmissão e a frequência de transmissão, foi desenvolvido um sistema que suporta um bom funcionamento para distâncias até aos 5 km.

O módulo de recepção de dados na margem para comunicações a longas distâncias apresenta o diagrama de blocos da Figura 28. Este módulo é composto pela UART 1 e a UART 2. A primeira UART serve para que os dados recebidos pelo módulo sejam processados no computador. A segunda UART é configurada para a recepção dos dados enviados pelo módulo do barco.

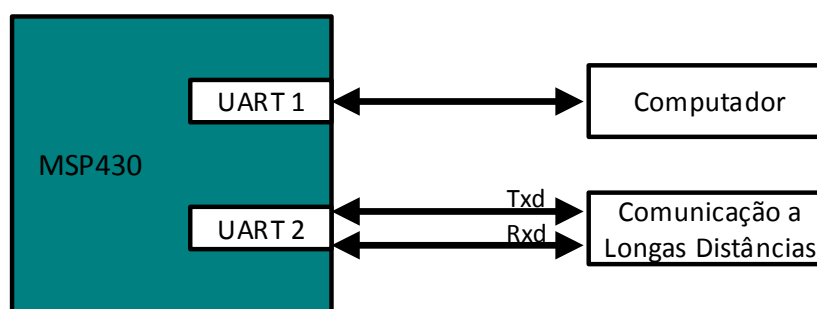


Figura 28: Diagrama de blocos que traduz o módulo de recepção de dados na margem.

A Figura 29 ilustra a máquina de estados do microprocessador, MSP430, para o módulo de recepção de dados na margem. Os estados “Intro” e “ConfigPeriféricos” são semelhantes aos anteriormente referidos para o módulo da pagaia e do barco. Quando uma interrupção do *timer* acontece, o microcontrolador, transita para o estado “sleep”. Perante um nova interrupção, de acordo com um tempo previamente configurado, avança para o estado onde lê os dados

recebidos do módulo do barco, num módulo nrF24L01. A escrita desses dados na UART 2, com o estado “WriteUART”, ocorre de acordo com uma nova interrupção gerada.

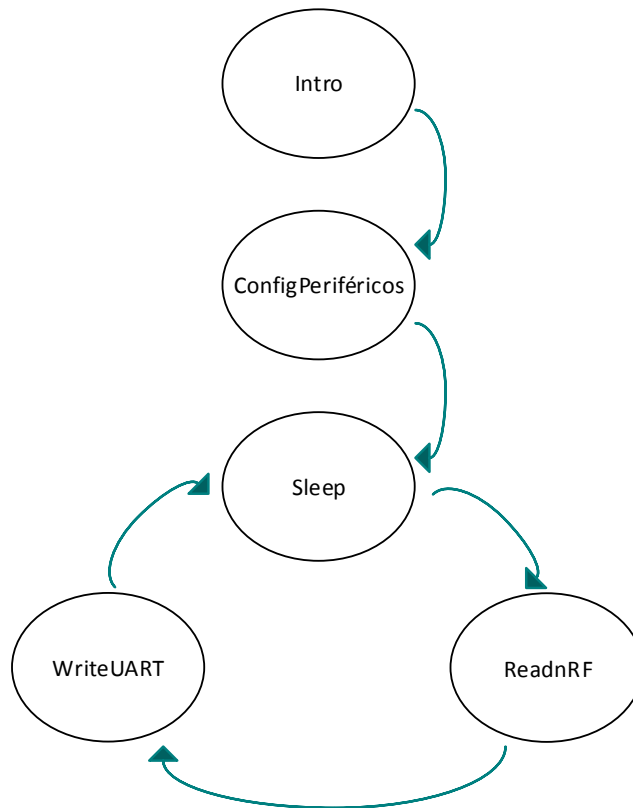


Figura 29: Máquina de estados do MSP430 para o módulo de recepção de dados na margem.

5.2.4. Resultados da implementação da versão 1.0.

Os resultados apresentados dizem respeito à implementação da versão 1.0. decorrente da integração, em rede, do módulo da pagaia com o módulo do barco complementados por comunicações a curtas e a longas distâncias de acordo com as descrições anteriores. Uma vez que o módulo do fincapés está em desenvolvimento, não foi incluído nesta prova de conceito.

Os primeiros dados recebidos na margem resultaram de um teste, para a aquisição de valores de aceleração da pagaia, onde só o circuito de comunicação RF para curtas distâncias estava implementado. A informação recebida foi analisada e processada em MatlabTM que é uma poderosa ferramenta matemática, interactiva e de elevado desempenho reconhecida pela facilidade em construir gráficos e resolver cálculos complexos. Para que os dados adquiridos pudessem ser analisados e traduzidos graficamente foi desenvolvido um código em MatlabTM que permitia fazer uma análise aos valores recebidos pela porta série do computador e identificar no pacote de dados os valores de x,y e z do acelerómetro da pagaia a cada instante de tempo. A informação era depois traduzida num gráfico deslizante para que os valores pudessem ser analisados sem acumularem na janela de visualização. A Figura 30 representa um dos primeiros testes realizados no exterior com um atleta do Clube Fluvial de Coimbra e a Figura 31 uma análise preliminar dos dados recebidos.

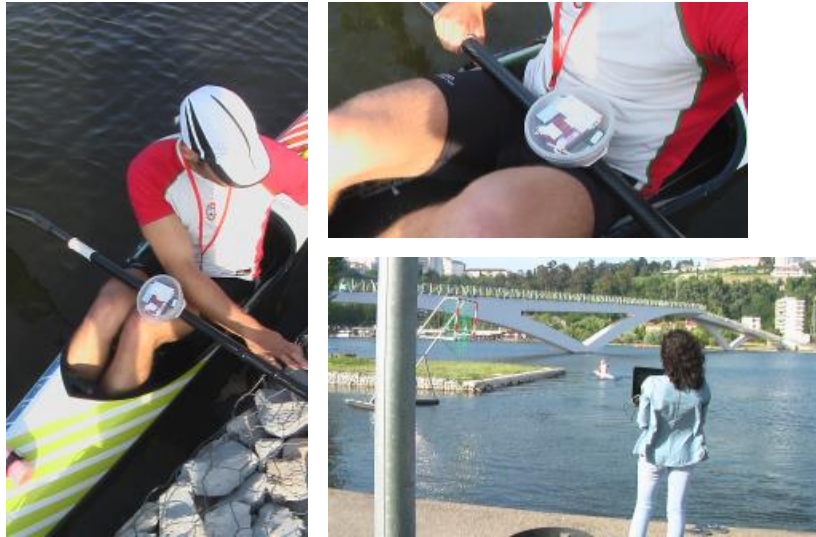


Figura 30: Testes preliminares à versão 1.0. realizados no rio Mondego.

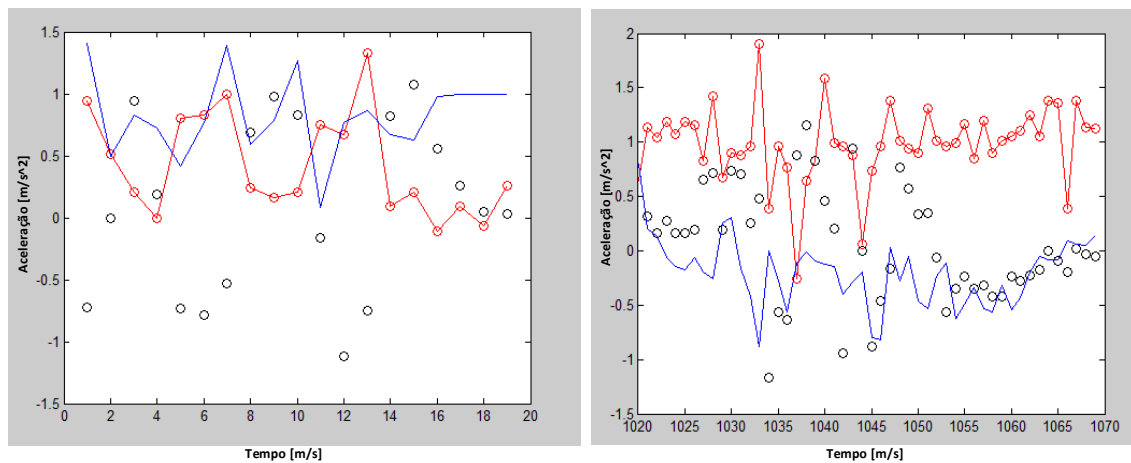


Figura 31: Representação gráfica de x, y e z adquiridos no momento da pagaiada. Observação: imagens guardadas em ambiente de rio através de imagens retiradas ao ecrã do computador.

Na Figura 31 os gráficos à direita e à esquerda correspondem a pagaiadas de atletas diferentes e em momentos distintos. Apesar de ainda não serem reconhecidas pagaiadas sabemos que o gráfico à esquerda corresponde a um início de pagaiada e o da direita a um conjunto de pagaiadas sucessivas. Além disso, pode concluir-se, pelos valores de aceleração obtidos, que o atleta do gráfico da direita tem uma pagaiada mais forte com valores de aceleração superiores. A distância máxima conseguida entre o atleta e o sistema receptor, sem comprometer a recepção dos dados, foi de 10 metros. Para distâncias superiores foram verificadas latências e perda de informação.

Por haver limitações no que à distância diz respeito foi implementada a comunicação a longas distâncias que permitiu fazer a contagem de pagaiadas recorrendo para isso aos dados recolhidos pelo eixo y do acelerómetro da pagaia. Para validar os dados recebidos foram feitos testes, em laboratório, a adultos e crianças em intervalos de tempo variáveis. Analisando a Figura 32 conseguem ser verificadas as diferenças esperadas entre a prática de uma criança e

de um adulto. Além disso é notória a diferença de rendimento e eficiência de pagaiada entre o caso de estudo 2) e 4). Enquanto em 2) o atleta pagaiava de forma descontraída e ocasional, em 4) estava preparado para uma prática mais exigente o que resultou em pagaiadas exímias ao mesmo ritmo e semelhantes entre si, atingindo uma cadência perfeita.

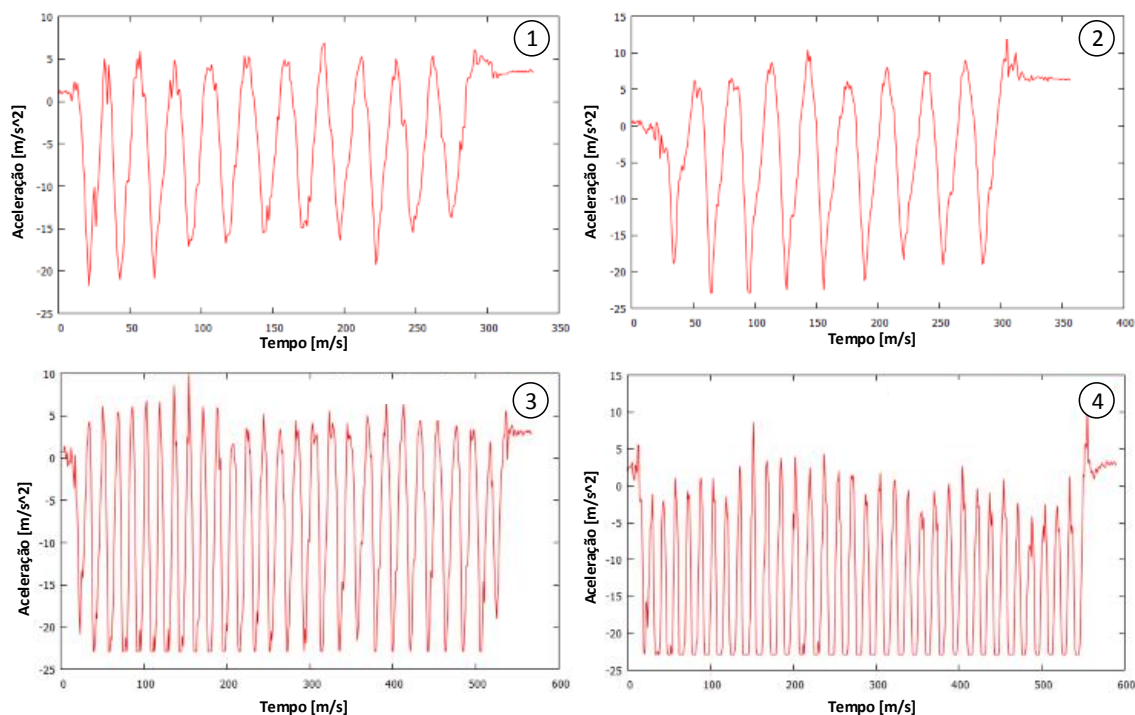


Figura 32: Contagem do número de pagaiadas: 1) Criança de 8 anos, 11 pagaiadas; 2) Adulto 15 segundos, 9 pagaiadas; 3) Adulto 30 segundos, 27 pagaiadas; 4) Série de 200m de um adulto, 33 pagaiadas.

A análise feita na Figura 33 pode ser comparada a análises de estudos anteriores [42]. O dispositivo desenvolvido em [42] é baseado num giroscópio e num acelerómetro, ambos de três eixos, e os dados adquiridos pelo sistema, que é controlado por um microcontrolador 8051, são guardados num cartão SD. Os dados são, depois de adquiridos e guardados no cartão, analisados numa aplicação desenvolvida para o efeito. A Figura 34 mostra os parâmetros adquiridos pelo dispositivo do estudo e permite que uma comparação possa ser feita com os parâmetros obtidos pela versão 1.0. da ferramenta, Figura 33.

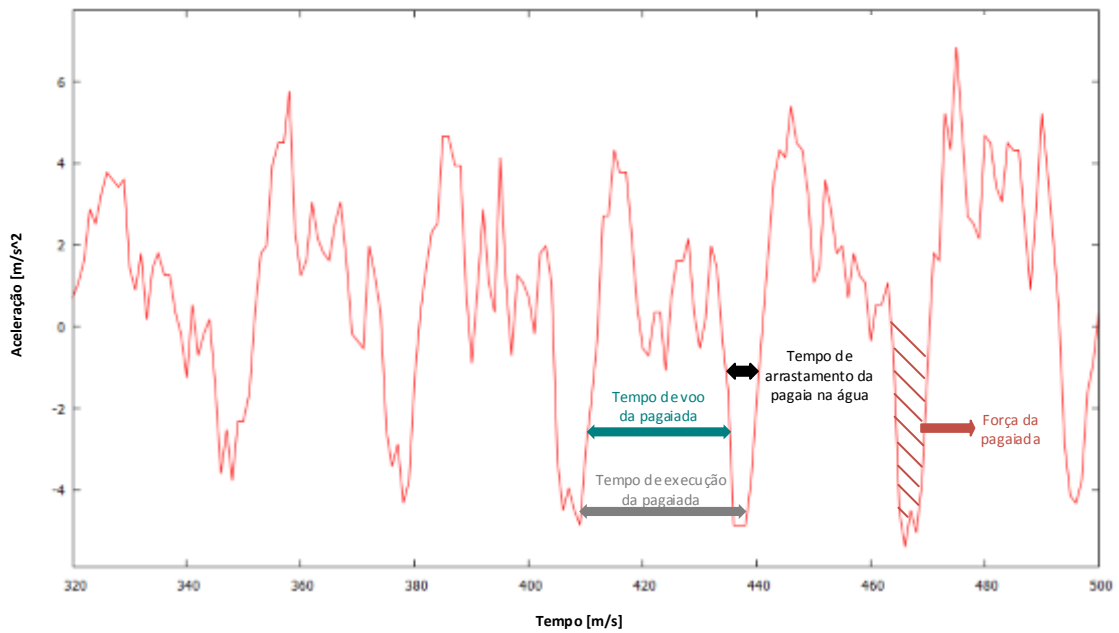


Figura 33: Análise da informação adquirida de uma série de remadas. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.

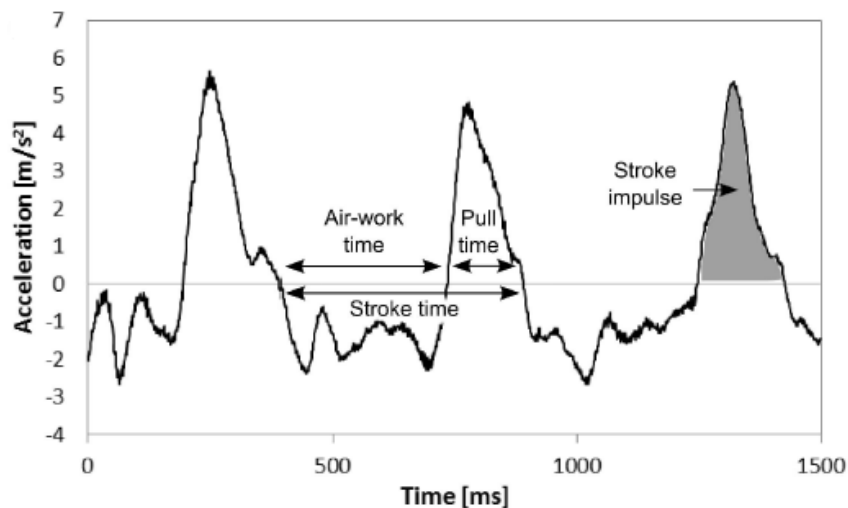


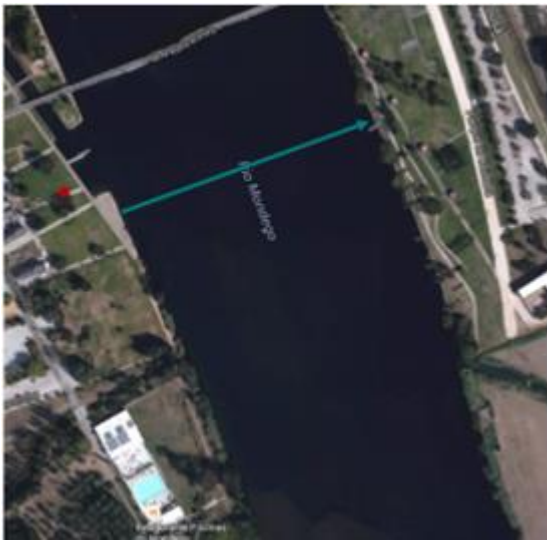
Figura 34: Parâmetros de um ciclo de remadas. [42]

Ao comparar as Figuras 33 e 34 nota-se que a contagem e identificação das remadas é feita em quadrantes diferentes, isto é, na Figura 33 é feita para valores de y negativos e na Figura 34 para valores de y positivos. Esta diferenciação é justificada pela orientação do acelerómetro nas duas situações. No entanto, é uma forma de validação dos resultados adquiridos.

As Figuras 32 e 33 correspondem a imagens conseguidas com o GnuPlot™, um *software* (SW) que gera gráficos a partir de ficheiros com extensão txt. Os valores adquiridos de x, y e z passaram a ser guardados em ficheiros .txt para poderem ser analisados depois da prática e dos testes realizados. Percebeu-se nesta fase que o Matlab™ foi útil para analisar os dados e a forma

de serem processados no entanto, e no ambiente em que são efectuados os testes, demonstrou ser uma ferramenta demasiado pesada e com dificuldades em reproduzir graficamente, em tempo real, as aquisições. O Gnuplot™, comparativamente ao Matlab™, é uma ferramenta mais básica e apesar de não funcionar em tempo real foi suficiente para o tipo de análises que no momento se pretendiam.

Para testar a distância máxima possível de distanciamento entre o módulo receptor e o conjunto atleta-barco, sem a perda de informação adquirida no módulo receptor, foram realizados testes no exterior. Os testes em ambiente de rio permitiram avaliar diferentes posições do módulo receptor e distâncias percorridas pelo atleta. Nas Figuras 35, 37 e 39 a marca a vermelho corresponde à localização do elemento receptor e as setas a verde ao trajecto percorrido pelo atleta.



≈223m

● Situado em cima de uma elevação de terreno a 25m da margem

Figura 35: Teste de 223m de distância entre o dispositivo emissor e receptor.

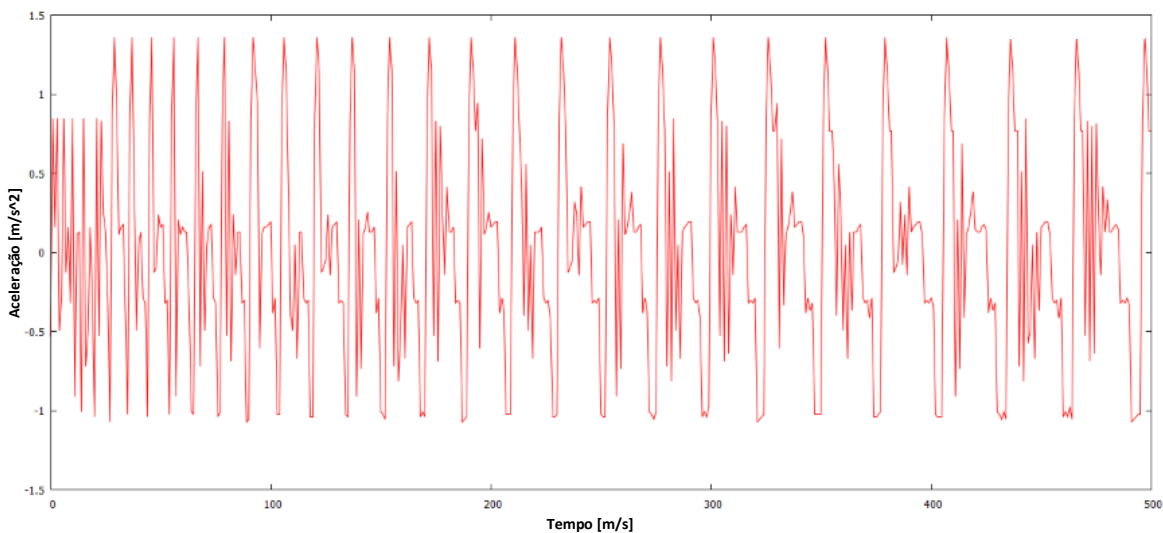


Figura 36: Amostra dos dados recebidos no módulo receptor no teste a 223m do módulo emissor. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.



≈212m



Figura 37: Teste de 212m de distância entre o dispositivo emissor e receptor.

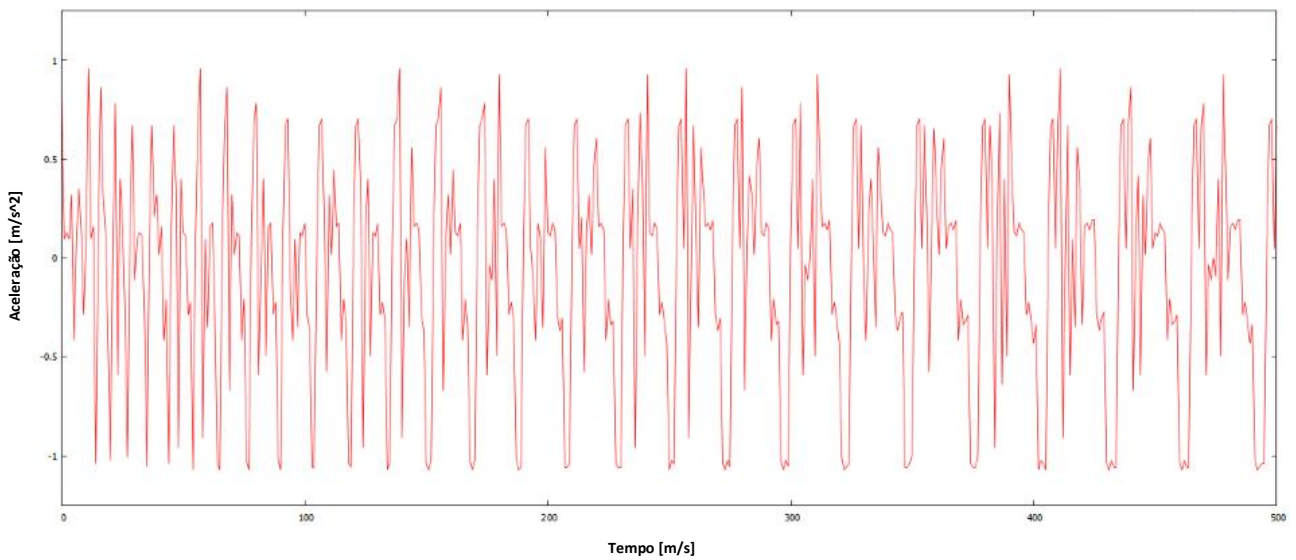
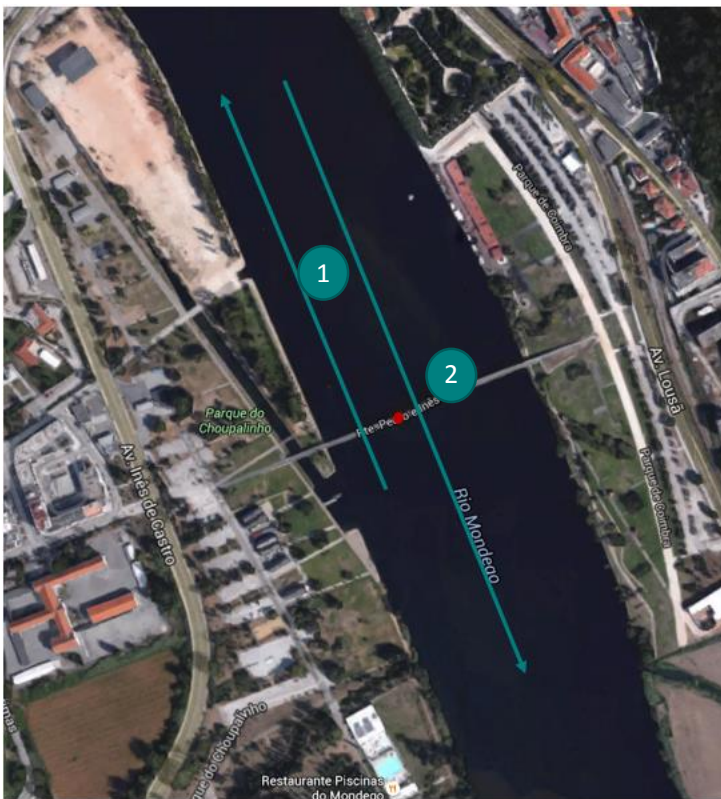


Figura 38: Amostra dos dados recebidos no módulo receptor no teste a 212m do módulo emissor. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.



- 1 ≈366m
- 2 ≈628m
- Situado na ponte pedonal

Figura 39: Teste de 366m seguidos de 628m.

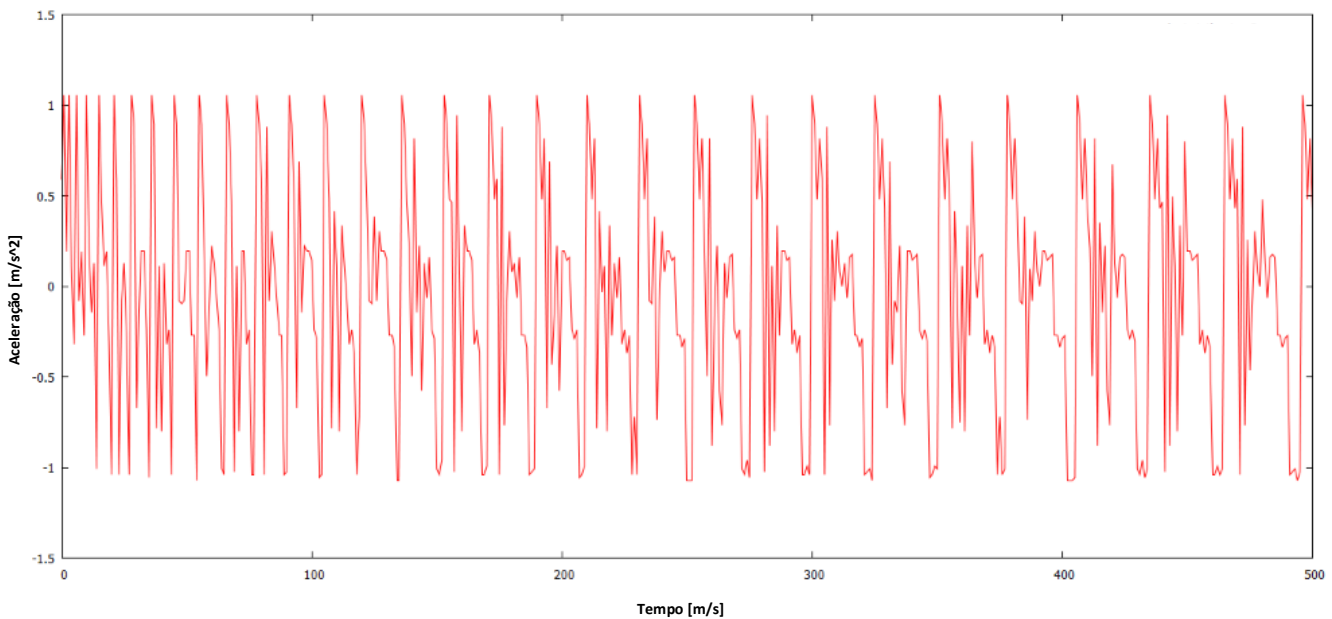


Figura 40: Amostra dos dados recebidos no teste às distâncias de 366m+628m. Os dados correspondem aos dados do eixo y do acelerómetro da pagaia.

O objectivo da versão 1.0. era realizar uma prova de conceito com o intuito de entender se era possível adquirir dados perceptíveis sobre o comportamento da pagaia e se, com o módulo de comunicações a longas distâncias implementado, os dados eram recebidos na margem para serem processados e analisados. Os resultados obtidos demonstram que os dois objectivos da

prova de conceito foram conseguidos. A rede desenvolvida com o módulo da pagaia e do barco, versão 1.0., permite não só contar e identificar pagaiadas, como também receber na margem, no elemento receptor da comunicação a longas distâncias, os dados adquiridos.

5.2.5. Módulo do finca-pés

A medição das forças exercidas pelos pés dos atletas é possível pela inclusão de sensores de força no finca-pés. A referência que é feita nesta secção ao módulo do finca-pés descreve o ponto em que o protótipo se encontra em termos mecânicos.

Os sensores escolhidos para esta solução correspondem a sensores de pressão Flexiforce da Tekscan. Foram utilizados sensores do modelo A201 de 197mm de comprimento com uma gama de medição até aos 440N. A implementação destes sensores no finca-pés, em termos mecânicos e electrónicos, necessitou de algum tempo de estudo e análise pelas preocupações e implicações do meio em que se inserem. Os sensores não serão submersos em água mas, estando abaixo da cota de água, poderão ser sujeitos a um ambiente húmido e por esse motivo são requeridas soluções de segurança e estanquicidade apropriadas, sem afectar a própria aquisição dos dados. A prototipagem deste módulo contou com a ajuda e transferência tecnológica do Departamento de Engenharia Mecânica e Desgaste de Materiais do Instituto Pedro Nunes.

O protótipo, ainda em fase de desenvolvimento, foi pensado de forma a otimizar a medição das forças dos pés de forma individual e de acordo com o movimento que é exercido. Uma vez que a força aplicada no pé não é uniforme, sendo aplicada mais força nas plantas laterais e no calcanhar do pé, foi pensada uma solução tal como a representada na Figura 41. Desta forma, e de acordo com a melhor opinião dos atletas da modalidade, a solução pensada prevê o desenvolvimento de um tapete com sensores, a fixar na superfície do finca-pés onde os atletas colocam os pés, Figura 41.

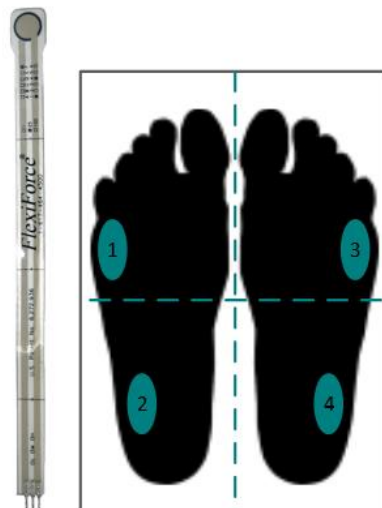


Figura 41: Representação da solução idealizada para os sensores no finca-pés. 1,2,3 e 4 correspondem aos sensores.

A solução apresentada pelo Instituto Pedro Nunes corresponde a uma prova de conceito para avaliar o envolvimento e a sensibilidade do sensor de pressão quando incluído em silicone. A Figura 42 demonstra o protótipo desenvolvido para o teste de conceito.

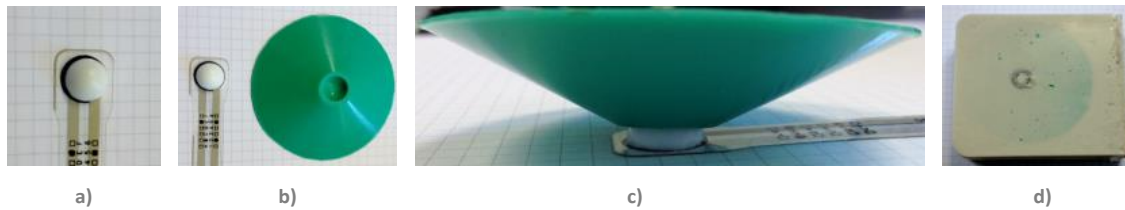


Figura 42: Estrutura desenvolvida para o teste de conceito do sensor no finca-pés.

Como a área de aquisição de valores no sensor de pressão tem dimensões reduzidas, que limitam o sistema desenvolvido para a medição das forças exercidas pelos pés do atleta, foram desenvolvidas duas estruturas que permitem aumentar a área de superfície em que são aplicadas as forças, b). As duas estruturas juntas, c), foram envolvidas em silicone através de um molde resultando em d).

5.3. VERSÃO 2.0.

A versão 1.0. e os resultados com ela obtidos demonstraram a necessidade de implementar na rede um protocolo de comunicação mais apropriado, o BLE. A versão 2.0. surge como um estudo ao BLE e para a sua integração na rede. Para perceber o protocolo de baixo consumo na sua integridade, avaliar e validar a sua implementação foi utilizado o CC2541 SensorTag. O SensorTag é uma solução de teste e validação oferecida pela Texas InstrumentsTM de baixo consumo que, para além de ter um acelerómetro, tem um giroscópio, um magnetómetro, um sensor de temperatura, humidade e de pressão. Este dispositivo é alimentado por uma bateria e é baseado no protocolo BLE. Por poder ser usado como um modelo de referência e uma plataforma de desenvolvimento com código aberto e disponível, o SensorTag é uma opção viável para validar conceitos e sistemas. Além disso, as características de *HW* do SensorTag são semelhantes às idealizadas para uma versão final do módulo da pagaia. A Figura 43 mostra o CC2541 SensorTag e o seu *HW*.

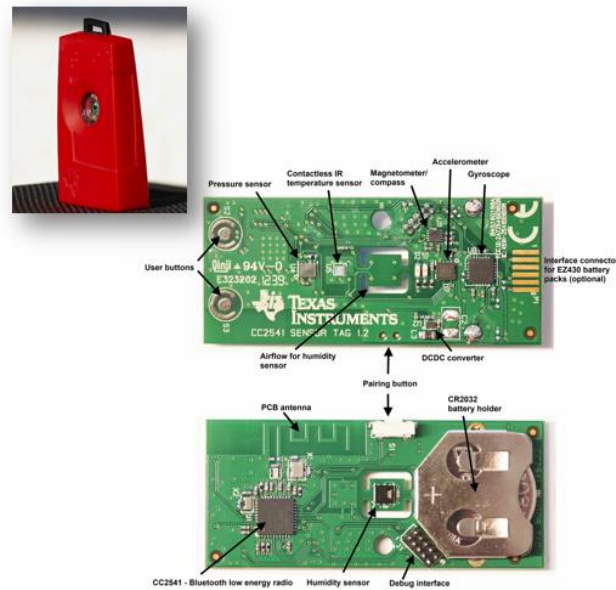


Figura 43: CC2541 SensorTag e respectivo *hardware*. [27]

O SensorTag é implementado com um microcontrolador CC2541 Bluetooth™ Low Energy Radio System. Este microcontrolador é uma solução válida para comunicações que utilizem o protocolo BLE. O CC2541 tem uma memória flash programável de 256KB, uma memória RAM (Random Access Memory) de 8KB e suporta taxas de transmissão de dados de 250kbps como valor mínimo e 2Mbps como máximo [27]. O microcontrolador foi desenvolvido para ter consumos otimizados e por isso é robusto em redes de nós sem fios onde sejam prioritários tempos baixos de transição entre os vários modos de operação. O CC2541 tem uma configuração *System On Chip* (SoC) ou seja o integrado inclui circuitos digitais, analógicos e de RF. No SensorTag a Aplicação, o *Host* e o *Controller* são executados num único IC e por isso não é necessária uma camada de comunicação (*UART*, *USB*, *SPI* ou outra). Esta é a configuração de *HW* mais utilizada em dispositivos BLE quando se pretende manter a complexidade e o custo do circuito impresso baixos [39]. A Figura 44 apresenta a configuração *SoC* do SensorTag.

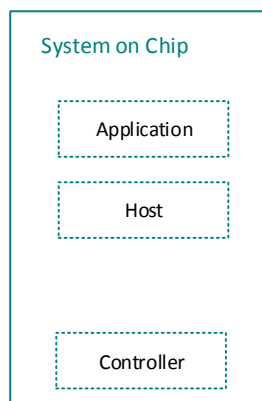


Figura 44: Configuração *SoC* do SensorTag. [34]

No SensorTag os sensores usam uma interface I²C (Inter Integrated Circuit) e, apesar de utilizarem diferentes sinais de activação, estão ligados ao mesmo barramento. Porque é

importante manter baixos consumos de corrente, os sensores mantêm-se em modo de baixo consumo entre aquisições [27]. O Quadro 1 permite analisar o consumo de corrente de cada sensor de acordo com dois modos de operação, modo activo e em baixo consumo.

Quadro 1: Consumo de corrente em *sleep mode* e no modo activo nos sensores do SensorTag.

[27]

Sensor	Active Mode [μ A]	Sleep mode [μ A]
Giroscópio	7000	5
Acelerómetro	135	0.9
Magnetómetro	350	3
Humidade	300	0.15
Pressão	5	0.1
Temperatura IV	240	0.1

Os dados recebidos pelos sensores do SensorTag, o acelerómetro, o giroscópio e o magnetómetro, permitiram perceber a importância destes sensores na análise do movimento e comportamento da pagaia. Desta forma, a inclusão de um *IMU*, Inertial Motion Unit, no módulo da pagaia tornou-se uma possibilidade. Os módulos inerciais combinam o baixo custo, e tamanho reduzido, com a capacidade de adquirir dados que permitem perceber no espaço, sem a existência de um elemento de referência, o comportamento de um objecto [41]. Um *IMU* é um dispositivo electrónico que combina as potencialidades inerciais de um acelerómetro, de um giroscópio e de um magnetómetro, todos sensores de três eixos. Desta forma, com o acelerómetro é detectada a aceleração aplicada sobre o sensor mais a aceleração da gravidade. O acelerómetro permite perceber a orientação do sensor em relação ao plano horizontal quando o objecto não está a sofrer nenhum tipo de aceleração. O giroscópio detecta as velocidades angulares a que o sensor está sujeito. Mesmo que o sensor esteja a sofrer algum tipo de aceleração é possível calcular continuamente, através de um giroscópio, a velocidade angular a que o sensor está sujeito. O magnetómetro embora não seja um sensor inercial, fazendo parte de um *IMU*, permite obter a orientação em relação ao norte magnético. Apesar de serem os três sensores de três eixos, cada sensor é orientado de maneira diferente e por isso se podem definir o *roll*, *pitch* e *yaw*. O *roll*, *pitch* e *yaw* correspondem aos ângulos de Euler que surgem associados a rotações de três dimensões, [41]. No fundo estes ângulos correspondem a uma combinação de um conjunto de rotações em 3D que pode ser representada por uma única rotação em torno de um vector apropriado. Qualquer rotação 3D pode ser dividida em 3 rotações em torno do sistema de eixos de coordenadas fixas X,Y e Z. Desta forma, o *IMU* seria relevante na percepção, a três dimensões, do comportamento da pagaia no espaço. Com um único integrado é possível obter dados de aceleração e estimar a posição no espaço de um objecto.

CAPÍTULO 6 – O SENSORTAG E O BLUETOOTH LOW ENERGY

Perceber o Bluetooth Low Energy, como é iniciado e estabelecido, em detalhe é importante para o poder aplicar na rede sensorial definida e por se ter como objectivo desenvolver uma aplicação própria para o sistema (apresentada no Capítulo 7).

O Capítulo que se segue, baseado na versão 2.0. da ferramenta, começa por fazer referência à Btool, recurso disponibilizado pela Texas InstrumentsTM para testar aplicações Bluetooth 4.0., e descreve o protocolo BLE de acordo com o funcionamento do SensorTag. Por ser uma ferramenta documentada, com código e *SW* disponível, e pela semelhança de *HW* entre o SensorTag e o módulo da pagaia (numa versão final), o SensorTag constituiu uma mais-valia no estudo aprofundado do BLE.

6.1. BTOOL

Para perceber o funcionamento do protocolo recorreu-se à Btool, da Texas InstrumentsTM, que é uma ferramenta de desenvolvimento auxiliar do SensorTag. Começar a comunicar com o SensorTag implica ter uma interface *USB* CC2540, uma *dongle*, ligada a uma porta *USB* de um computador. A única diferença entre o microcontrolador CC2540 da *dongle* e o CC2541 do SensorTag é o facto de o CC2540 estar preparado para uma interface *USB* enquanto o CC2541, para uma interface *I²C* e estar optimizado para baixos consumos [27]. A *dongle* funciona nesta comunicação como o dispositivo central e o SensorTag assume-se como dispositivo periférico da ligação. Quem inicia o processo de descoberta, estabelece a comunicação e desencadeia a activação de serviços, funcionando como um cliente GATT é a *dongle*. Por outro lado, o servidor GATT é SensorTag que contém a tabela de atributos dos serviços nele disponíveis. A forma mais eficiente, em termos energéticos, de obter dados de um sensor com o BLE é numa primeira instância activar a notificação do sensor e em seguida o sensor. Este processo é gerido pela camada GATT que através do protocolo ATT permite que atributos do servidor GATT (o SensorTag) sejam endereçáveis pelo cliente GATT (a *dongle*). Quando a *dongle* recebe a notificação, o sensor do SensorTag pode ser desligando e o que se mantém activo é o serviço de notificações. O SensorTag tem intervalos de notificação cujo valor por defeito é de 100ms.

Para se iniciar uma ligação recorrendo à Btool é necessário numa primeira fase definir a porta COM (Communication Port), a *baudrate* (taxa de transferência de dados) e outras configurações da *dongle*. Quando isto é feito, há uma comunicação que é estabelecida entre a *dongle* e a Btool e a informação que é configurada é mostrada na Btool numa janela relativa à “Informação dos dispositivos”. É o processo de descoberta de dispositivos, que o dispositivo central inicia, o primeiro a acontecer no protocolo. Este processo é activado pelo botão “Scan” da aplicação. Em seguida, perante um endereço de um dispositivo disponível para comunicação, é estabelecida com o botão “Establish” a ligação. Este dispositivo pode ser, por exemplo, o SensorTag. Nesta fase, depois de estabelecida a comunicação entre os dois dispositivos, é possível distinguir o endereço e *Handle* entre a *dongle* e o dispositivo de ligação. A *Handle*

atribuída corresponde a uma identificação que a Btool confere aos dispositivos para distinguir no protocolo se o dispositivo é a *dongle*, 0XFFFE, ou o SensorTag, 0x0000. Normalmente ao dispositivo central é sempre atribuída por defeito a *Handle* 0XFFFE. À medida que a troca de pacotes de dados acontece é feito, em paralelo, o registo das mensagens que o protocolo gera. As mensagens registadas na aplicação, numa janela apropriada para o efeito, permitem descrever e analisar o protocolo em pormenor. A descrição anterior, que pode ser acompanhada pela Figura 45, permite compreender a interface Btool, [43].

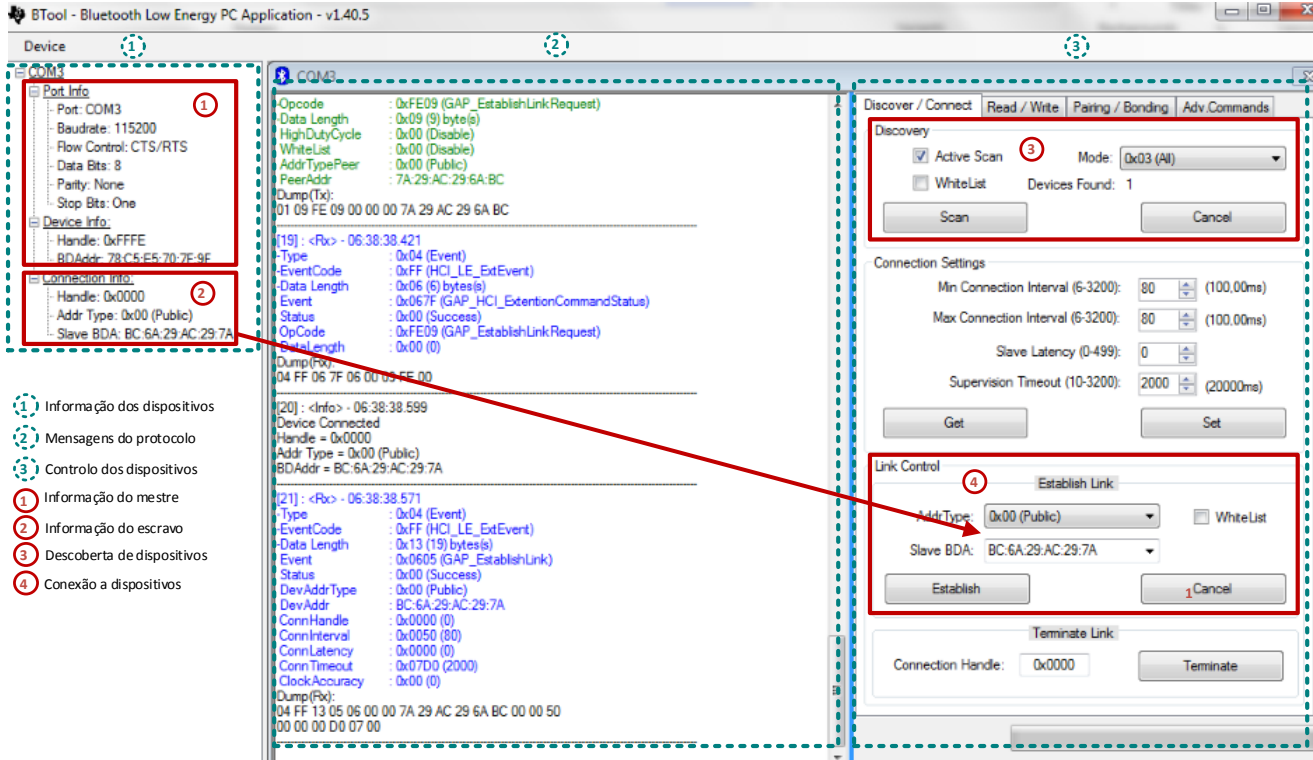


Figura 45: Interface da Btool.

6.2. EVENTOS E COMANDOS HCI ESPECÍFICOS DA TEXAS INSTRUMENTS™

Depois de explicado o funcionamento base da Btool importa estabelecer a ligação entre a CC2540 *USB dongle* e a Btool. Na execução desta ligação pode observar-se o envio de eventos e comandos HCI. De acordo com o descrito anteriormente no Capítulo 4, na referência ao protocolo BLE e interface HCI, o envio de comandos e recepção de eventos pode ser implementado numa comunicação *USB*. Assim, no processo de estabelecer a comunicação entre a *dongle* e a Btool, são trocados comandos HCI pela interface *USB* da forma representada na Figura 46. A configuração ilustrada na Figura 46 é *Dual IC over HCI* ou seja existem na configuração dois circuitos integrados e um deles tem associada uma potente unidade de processamento central, *CPU- Central Processing Unit*, que executa *astack* do protocolo. Na configuração apresentada o *Host* é a Btool, alocada no computador, e o *Controller* a *dongle* CC2540.

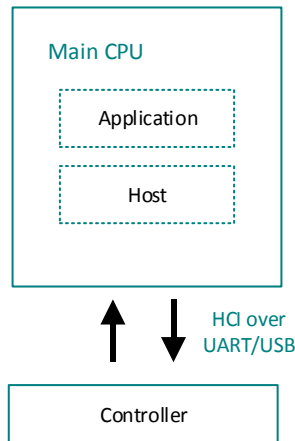


Figura 46: Configuração *Dual IC over HCI* entre a Btool e a dongle. [34]

Nesta comunicação é a Btool, o cliente, que envia comandos HCI através de pedidos que são respondidos com eventos por parte do servidor, a *dongle*, Figura 47.

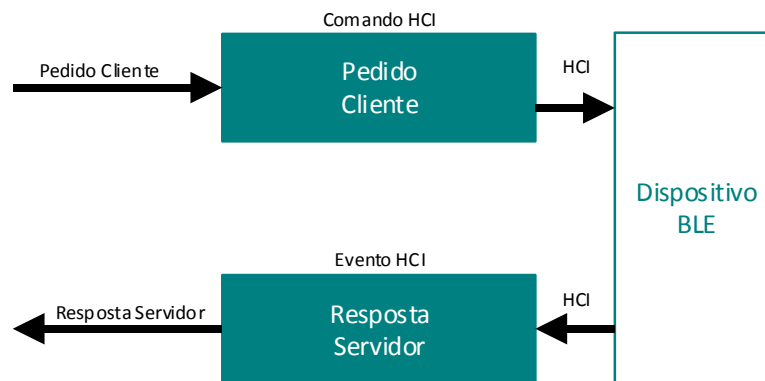


Figura 47: Envio de comandos e pacotes HCI na comunicação entre a dongle e a Btool. [38]

Quando é estabelecida a ligação entre a *dongle* e o SensorTag voltam a ser trocados comandos e eventos HCI. Desta forma, se começou por responder com eventos a comandos enviados pela Btool, no estabelecimento da ligação com o SensorTag é a *dongle* quem envia comandos HCI que espera serem respondidos por eventos do servidor. Esta configuração é a ilustrada na Figura 48. A análise da Figura 48 permite identificar dois dispositivos BLE, a *dongle* e o SensorTag, que comunicam *Over The Air (OTA)* ou seja por uma ligação sem fios. A Figura 48 corresponde a uma imagem generalizada e o SensorTag apesar de ser um dispositivo BLE, não gere os comandos e eventos HCI como representado na Figura 48 mas internamente de acordo com o ilustrado na Figura 44.

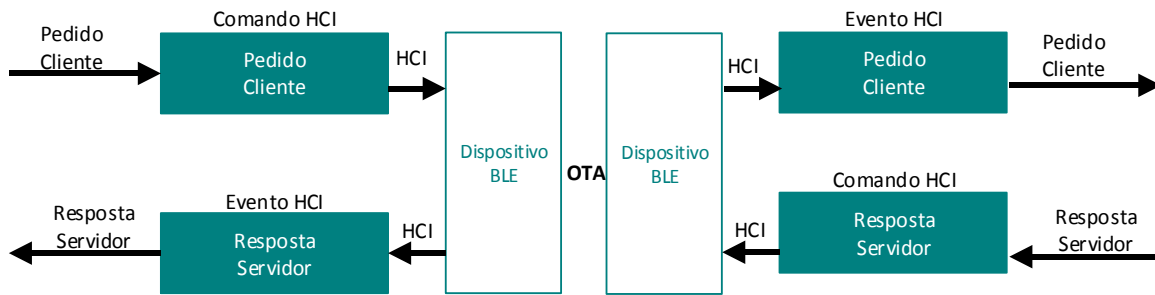


Figura 48: Configuração em rede do envio de comandos e eventos HCI na rede estabelecida entre a Btool, a CC2540 Dongle e o CC2541 SensorTag. [38].

Como mencionado no Capítulo 4 cabe ao fornecedor, em caso de necessidade, especificar os seus próprios eventos e comandos HCI usando para isso *OpCodes* e *Event Codes* próprios para os fornecedores. A Texas Instruments™ dividiu os 10 bits disponíveis do *OpCode* específico em duas partes. Os 3 MSB (Most Significant Bit) são utilizados para definir o subgrupo do comando e os restantes 7, os LSB (Least Significant Bit), o comando em si [38], Figura 49.

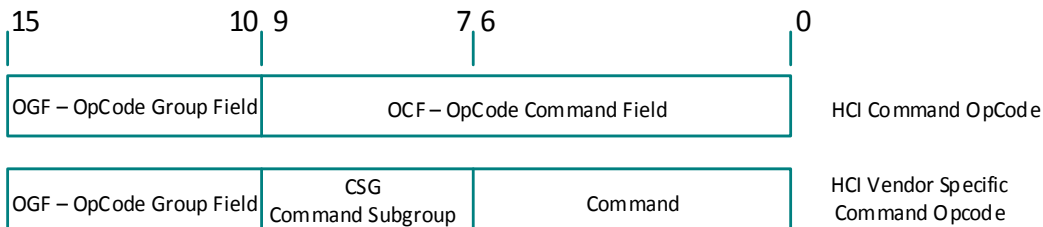


Figura 49: Comparação entre a estrutura de um comando HCI definido pelas especificações BLE e pelo fornecedor. [38].

Os comandos HCI definidos com o valor 63 no OGF, *OpCode Group Field* são comandos específicos do fornecedor. Os subgrupos de comandos, que distinguem a camada no BLE a que cada comando HCI pertence, estão apresentados na Figura 50.

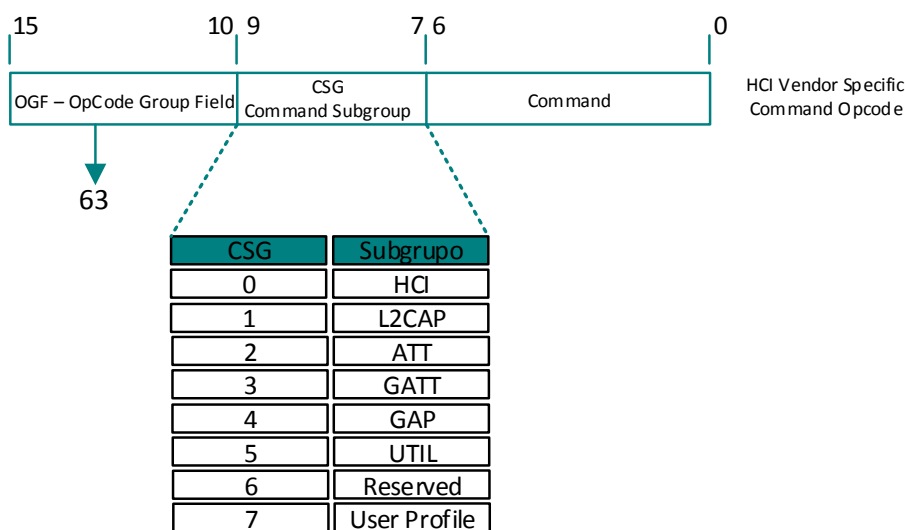


Figura 50: Estrutura de um comando HCI definido para a Texas Instruments™. [38]

Para além de comandos específicos, a Texas Instruments™, definiu eventos HCI próprios, Figura 51. Na Figura 51 é feita uma comparação entre um evento HCI BLE e um evento HCI específico da Texas Instruments™ explicando a repartição do pacote de dados.

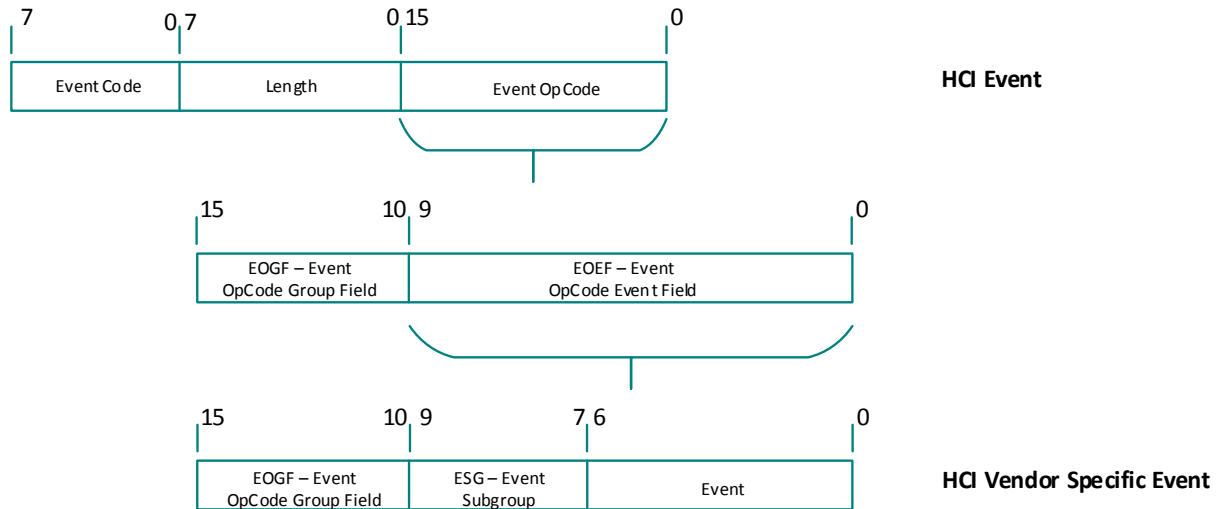


Figura 51: Comparação entre a estrutura de um evento HCI definido pelas especificações BLE e pelo fornecedor. [38]

Quando um *Event Code* assume o valor 255 significa que é específico do fornecedor. A Texas Instruments™ dividiu o *Event OpCode* em *Event OpCode Group Field* e *Event Field*. Os subgrupos dos eventos indicam o grupo a que os eventos se referem, Figura 52.

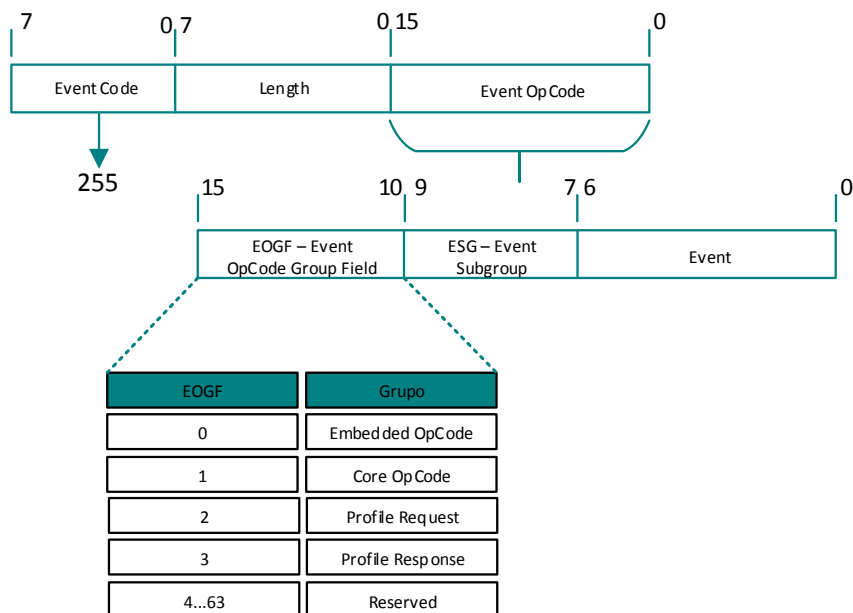


Figura 52: Grupos de eventos HCI definidos pelo fornecedor, Texas Instruments™. [38]

6.3. PERCEBER O BLE ATRAVÉS DO SENSORTAG E DA BTOOL

Aliando os tópicos anteriores referidos à Btool e aos eventos e comandos HCI específicos da Texas Instruments™, é possível descrever o protocolo e o processo de ligação entre dispositivos através das Figuras [55-55].

A primeira transferência de dados entre a *dongle* e a Btool é traduzida na Figura 53. É a Btool quem inicia o processo, como *Host*, e faz um pedido à *dongle* pelo seu endereço e informação para estabelecer a comunicação. Este pedido é através de um comando HCI **GAP_DeviceInit** que é associado à camada **GAP** por ser nesta camada onde é feita a definição dos perfis e modos de actuação dos dispositivos durante a comunicação. Depois de receber o endereço da *dongle*, através de eventos HCI, a Btool solicita parâmetros necessários à configuração da comunicação entre ambos. Com o comando HCI **GAP_GetParam** são pedidos os intervalos máximo e mínimo de ligação, o tempo de latência e o *timeout*, tempo máximo admitido para a ligação. Estes parâmetros devem ser definidos pelos dispositivos antes que a ligação aconteça. Quando a *dongle* responde, é estabelecida a ligação entre ambos.

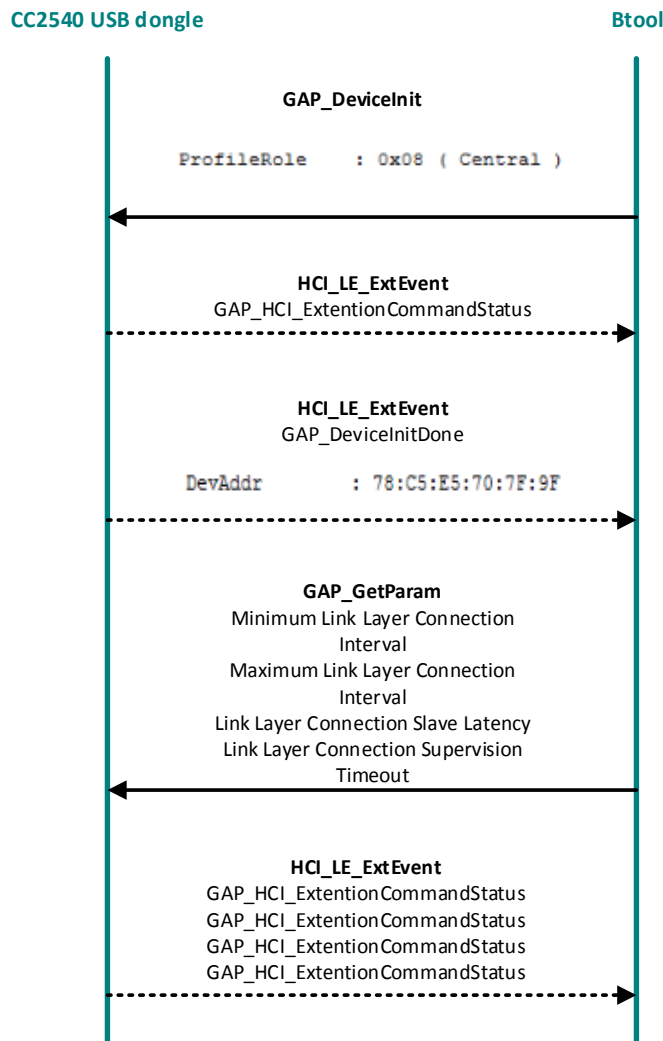


Figura 53: Primeiros dados trocados quando a CC2540 USB dongle é ligada à Btool.

Depois de estar configurada e devidamente ligada a *dongle* pode iniciar o *scan* por dispositivos dispostos a ligarem-se. Para isso a *dongle* envia o comando HCI **GAP_DeviceDiscoveryRequest** OTA. Se o SensorTag estiver em modo activo e disponível para a ligação é possível reconhecer a sua existência por ser encontrado na Btool o seu endereço. A disponibilização deste endereço traduz a resposta do Sensortag, ao comando enviado pela *dongle*, através de um evento HCI com o seu endereço. A Figura 54 ilustra o processo de *scan* que decorre quando o botão “Scan” da Btool é pressionado.

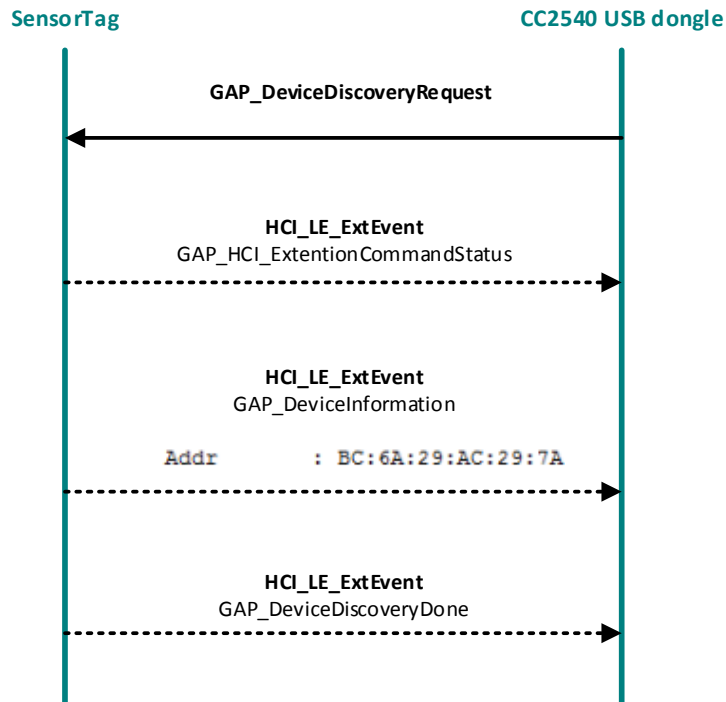


Figura 54: Pacotes de dados trocados no processo de *Scan*.

Depois do SensorTag disponibilizar o seu endereço, a comunicação entre ele e a *dongle* pode ser verdadeiramente estabelecida. A Figura 55, traduz o fluxo de dados que ocorre a partir do momento em que é pressionado o botão “Establish” na Btool. É com o comando HCI **GAP_EstablishLinkRequest** que a CC2540 *dongle* pede ao SensorTag para que estabeleçam a comunicação. Em resposta ao pedido o SensorTag, que já se tinha mostrado disponível para a ligação, envia um evento HCI com a informação que confirma que a mesma está estabelecida.

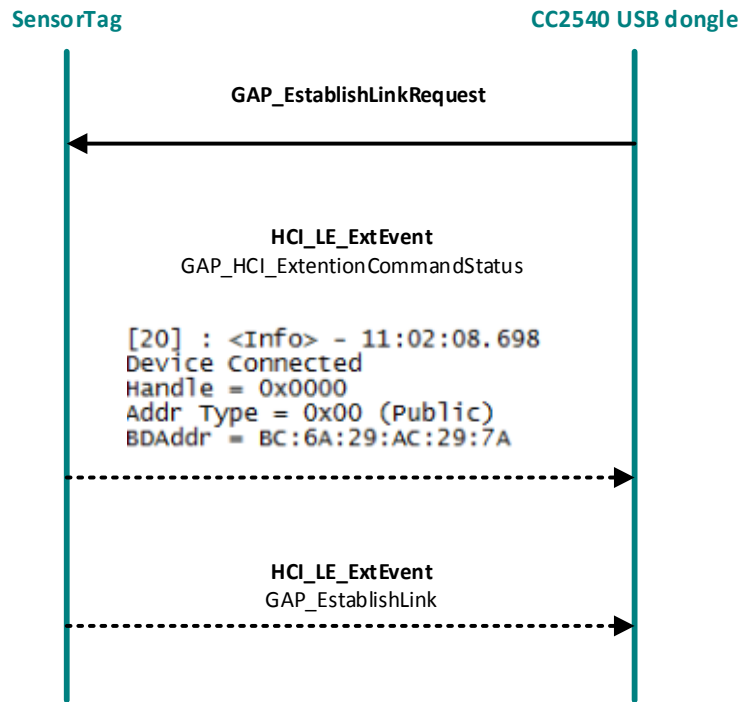


Figura 55: Pacotes de dados trocados no processo de *Establish*.

É importante referir que nas Figuras [55-55], à excepção dos eventos **GAP_HCI_ExtensionCommandStatus** que são comuns ao BT e BLE, os restantes eventos e comandos são específicos da Texas Instruments™.

Neste momento existe uma ligação estabelecida entre a *dongle* e o SensorTag. O acesso aos dados disponíveis nos sensores do SensorTag é possível pela utilização do serviço do respectivo sensor. A descrição deste processo é feita com o exemplo da recolha de dados do serviço do acelerómetro. Antes de pedir ao servidor GATT por valores de aceleração é preciso aceder no servidor ao atributo que activa o acelerómetro e permite a sua configuração. Para isso é necessário endereçar à *Handle*, do respectivo atributo (de valor hexadecimal 0x0034), e com permissão de escrita, deve ser colocada a “01”. A Figura 56 ilustra a interface de configuração da Btool para activação do acelerómetro.

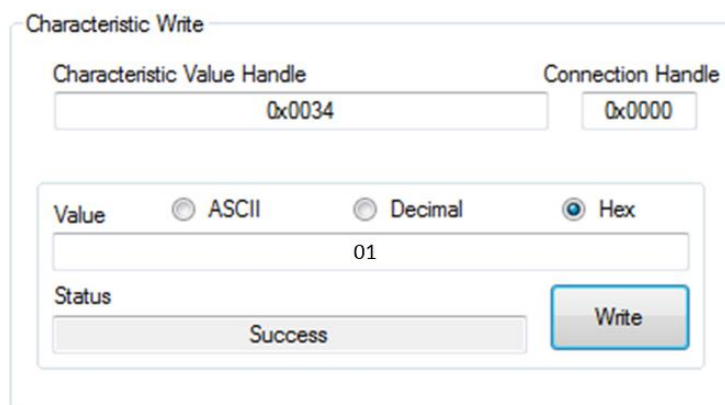


Figura 56: Modo de activação do serviço do acelerómetro.

A sequência e troca de comandos/eventos da Figura 57 ilustra o processo protocolar que ocorre na activação do acelerómetro do SensorTag. A *dongle* como cliente GATT pede ao SensorTag, o servidor GATT, para aceder ao atributo endereçado na *Handle* 0x0034 através de um comando HCI **GAP_LinkParameterUpdate**. Em resposta o SensorTag envia um evento HCI que confirma a autorização para que seja feita uma mudança de estado, um *update*, ao valor do atributo em causa. A *dongle* através de um comando **GATT_WriteCharValue** escreve o valor “01” no valor da *Handle* característica. O comando enviado é um comando HCI associado à camada GATT por ser a camada que gere, com base no protocolo ATT, como é feita a descoberta e o acesso, por modo de leitura ou escrita, a atributos. O SensorTag em resposta envia um evento HCI do tipo **ATT_WriteRsp** que confirma o sucesso da escrita.

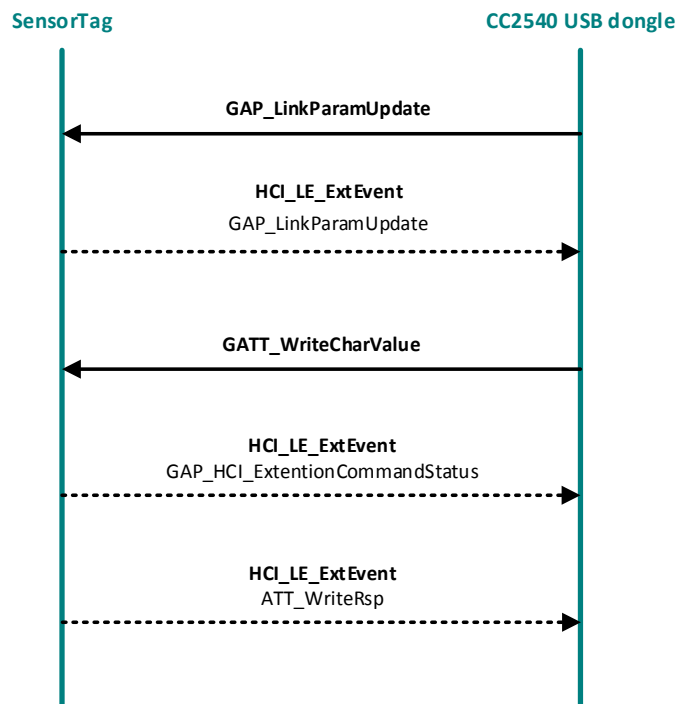


Figura 57: Eventos e comandos HCI para activação do serviço do acelerómetro.

Para obter dados do acelerómetro a *dongle*, deve enviar um pedido de leitura dos valores do atributo que guarda, para o serviço do acelerómetro, os dados de aceleração. Os valores de aceleração são recebidos através de notificações, **ATT_HandleValueNotification**, que foram activadas com a escrita do valor “01:00” para a *Handle* que, no serviço do acelerómetro, está associada ao atributo das notificações, Figura 58.

Characteristic Write

Characteristic Value Handle: 0x0031 Connection Handle: 0x0000

Value: 01:00 ASCII Decimal Hex

Status: Success Write

Figura 58: Modo de activação das notificações do acelerómetro.

O processo ilustrado na Figura 58, pode ser traduzido na sequência de troca de eventos e comandos HCI, da figura seguinte. A Figura 59 representa a sequência de notificações que são enviadas, sob eventos HCI, do SensorTag para a *dongle* com os valores de aceleração medidos.

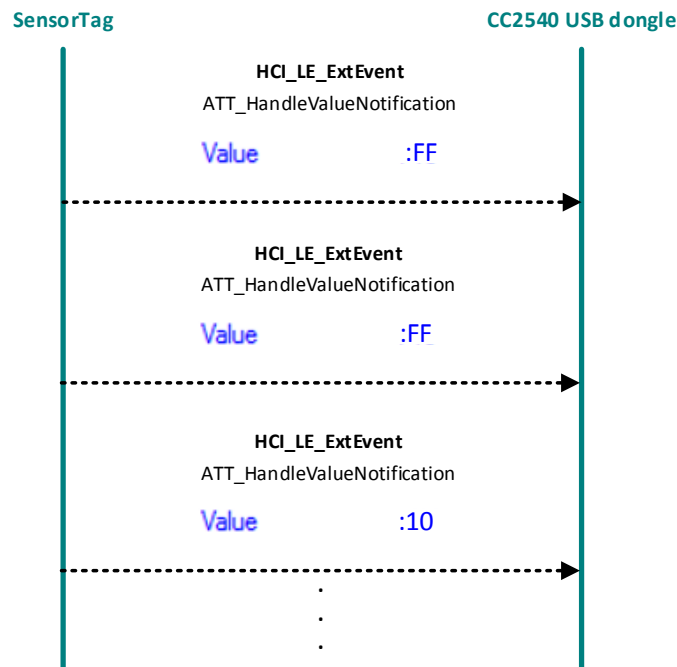


Figura 59: Eventos HCI das notificações do acelerómetro.

A Figura 60 ilustra os valores de aceleração dos três eixos do acelerómetro recebidos na Btool a cada 100ms, tempo definido por defeito no SensorTag entre notificações. O valor recebido “FF:FF:10” corresponde aos 3 bytes de aceleração das coordenadas X:Y:Z. A *Handle* 0x0030 é a *Handle* que define a característica **Accelerometer_Data_UUID**.

ConHnd	Handle	Uuid	Uuid Description	Value
0x0000	0x0030			FF:FF:10
			+	

Figura 60: Registo dos valores de aceleração dos três eixos.

Terminar a ligação com o SensorTag consiste na execução do comando **GAP_TerminateLinkRequest**, específico do fornecedor, Figura 61.

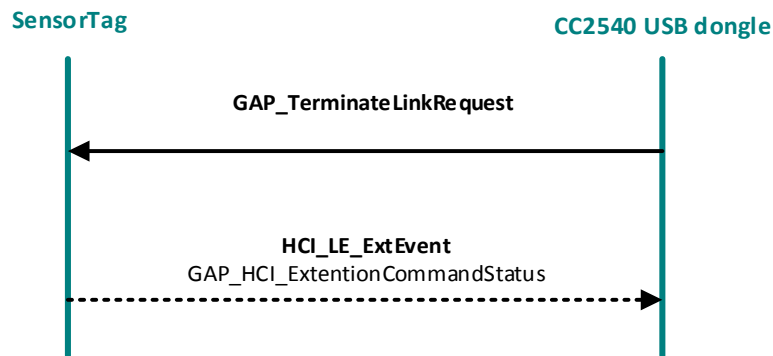


Figura 61: Eventos e comandos HCI para terminar a ligação entre dispositivos.

No decorrer de todo o processo a comunicação entre os dispositivos é interpretada e descrita em mensagens pela Btool. Dependendo do evento ou comando HCI e do momento ou o propósito com que são trocados, o conteúdo do pacote de dados apresenta informação diferente. Com a mensagem gerada pela Btool da Figura 62 é possível perceber como é formado um pacote de dados pelo BLE.

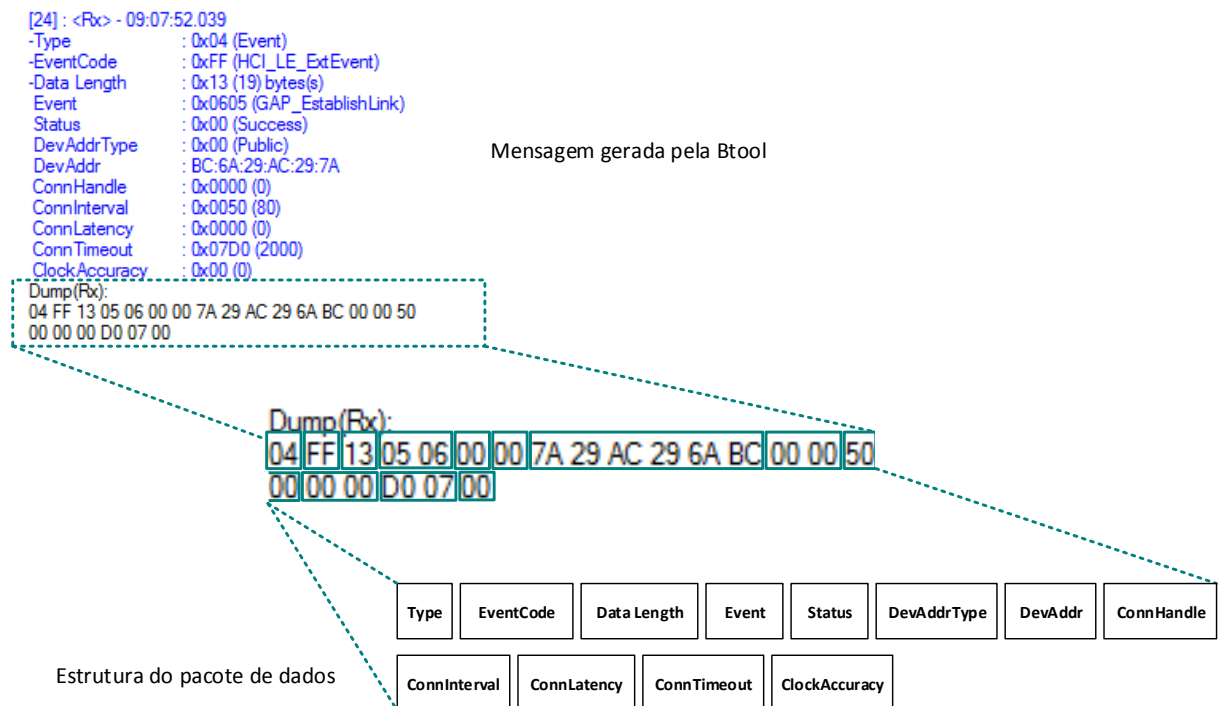


Figura 62: Estrutura de um pacote de dados.

As mensagens da Btool que traduzem as notificações enviadas pelo SensorTag com os valores de aceleração permitem identificar, no pacote de dados BLE, que os últimos 3 bytes do pacote dizem respeito aos valores do acelerómetro, Figura 63.

```
[2272] : <Rx> - 02:59:41.247
Type           : 0x04 (Event)
EventCode      : 0xFF (HCI_LE_ExtEvent)
Data Length    : 0x09 (9) bytes(s)
Event          : 0x051B (ATT_HandleValueNotification)
Status         : 0x00 (Success)
ConnHandle     : 0x0000 (0)
PduLen        : 0x03 (3)
Handle        : 0x0042 (66)
Value          : FF:FF:10
Dump(Rx):
04 FF 09 1B 05 00 00 00 03 42 00 FF FF 10
```

Figura 63: Estrutura do pacote de mensagens com os valores de aceleração dos eixos X,Y e Z.

Perceber o funcionamento pormenorizado do protocolo foi importante para o desenvolvimento de uma plataforma própria de comunicação e visualização de dados em C#, que será abordada no Capítulo seguinte.

CAPÍTULO 7 - PLATAFORMA DE COMUNICAÇÃO E VISUALIZAÇÃO DE DADOS

A plataforma de comunicação e visualização de dados surgiu da necessidade de ter na margem do rio uma ferramenta própria que permitisse ver e processar em tempo real, em simultâneo e de forma gráfica o comportamento de todos os sensores da rede em qualquer um dos nós. Pela compatibilidade, segurança e robustez da linguagem, a aplicação, que partiu de um código fornecido pela Texas InstrumentsTM, foi desenvolvida em C#.

Por já ser conhecido o SensorTag e o seu funcionamento no protocolo BLE, pelo estudo apresentado no Capítulo anterior, a plataforma foi otimizada e testada para a recepção de dados dos três sensores do SensorTag. Se a Btool permitiu conhecer melhor o protocolo BLE, a plataforma C#, recorrendo ao SensorTag, ilustra o conceito. Neste Capítulo, que descreve na plataforma C# como é implementado o BLE, desde que dois dispositivos comunicam até ao momento de aquisição de dados, é esperada uma associação directa com o referido anteriormente no Capítulo 6.

7.1. A APLICAÇÃO

A aplicação desenvolvida, partiu de um código fornecido pela Texas InstrumentsTM, e foi otimizada para ilustrar em diferentes separadores os dados de cada um dos módulos. Assim, existe um separador para o módulo da pagaia, outro para o módulo do finca-pés e outro para o barco. Para testar a aplicação e o protocolo na aplicação, foi utilizado, como já referido, o SensorTag. Desta forma, a aplicação obtém os dados em tempo real do SensorTag que são visualizados no separador destinado ao módulo da pagaia pelas semelhanças de *HW* que existem entre o SensorTag e este módulo numa versão final. Desta forma, os dados adquiridos em tempo real do acelerómetro, do magnetómetro e do giroscópio do SensorTag são registados de forma gráfica na aplicação. A Figura 64 ilustra a aplicação C#.

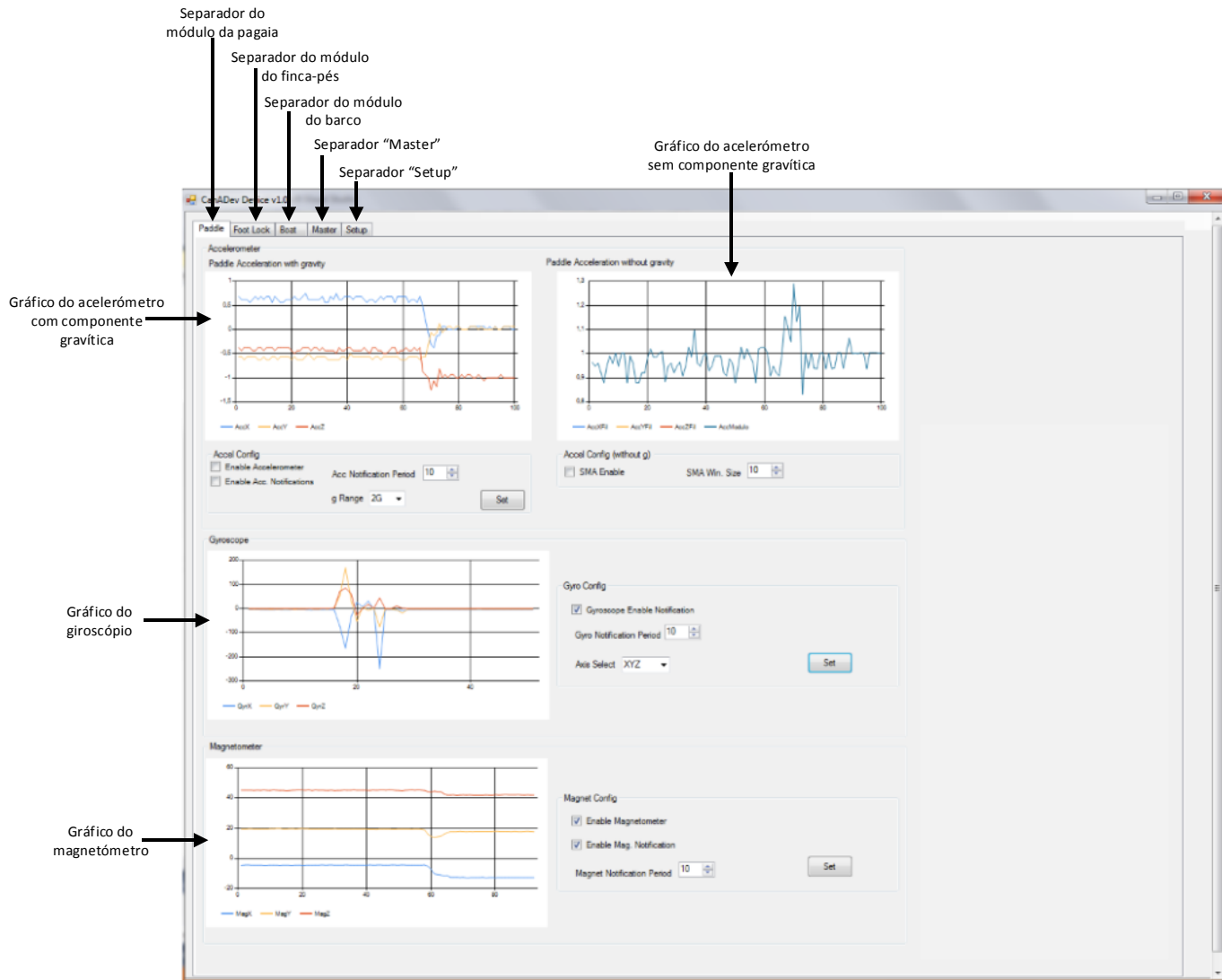


Figura 64: Aplicação em C# com os dados do acelerómetro, do giroscópio e do magnetómetro do SensorTag a serem adquiridos em tempo real.

Para além dos separadores destinados ao registo de informação de cada módulo existem os separadores “Master” e “Setup” essenciais à percepção do protocolo e funcionamento da aplicação. No separador “Master” é feito o registo dos *UIDs* da porta *USB* que estão a ser requisitados pelo protocolo à medida que a comunicação é estabelecida, Figura 65. O separador “Setup” tem a mesma funcionalidade da interface inicial da Btool demonstrada na Figura 45. É neste separador que é garantida a ligação da *dongle* ao computador, que é feita a procura por dispositivos periféricos de ligação, através do botão “Scan” e estabelecida e terminada a comunicação entre dispositivos pelos botões “Connect” e “Terminate”, respectivamente. Este separador, à semelhança da Btool, tem um espaço para o registo das mensagens do protocolo. No entanto, na aplicação existe a opção de não fazer este registo para que esta seja mais rápida a executar, Figura 66.

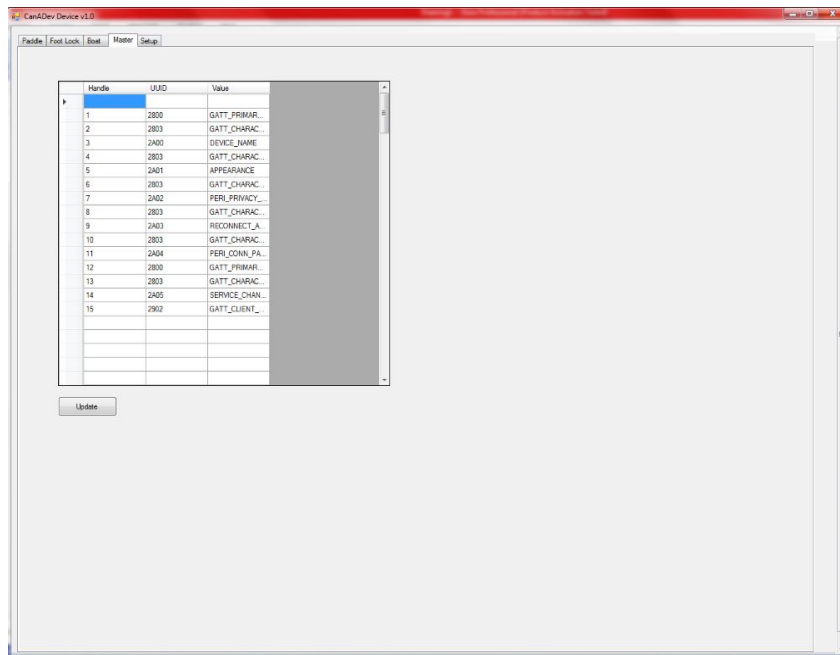


Figura 65: Imagem representativa do separador “Master”.

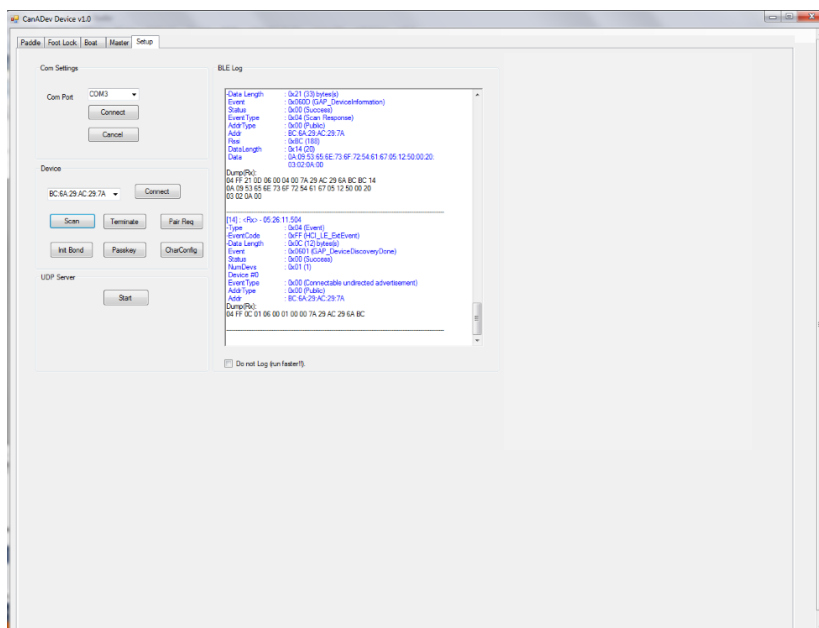


Figura 66: Imagem representativa do separador “Setup”.

7.2 O FUNCIONAMENTO

7.2.1. Configuração

A configuração e estabelecimento da ligação, tanto entre a *dongle* e a plataforma C# como entre a *dongle* e o SensorTag, são feitos no separador “Setup” da aplicação.

No momento de ligação entre a *dongle* e a plataforma C# é feita a abertura e configuração da porta série para a *baudrate* de 57600, 8 *databits*, com 1 *stopbit* e sem paridade. O primeiro comando HCI a ser trocado entre a plataforma C# e a *dongle* é o **GAP DeviceInit**. A Figura 67 apresenta a forma como deve ser estruturado na aplicação o comando **GAP DeviceInit**. Os parâmetros de configuração deste comando, de *OpCode* 0xFE00, definem o dispositivo como central e 3 o número de pacotes de *advertisement* que o dispositivo pode receber. De acordo com o número de pacotes de *advertisement* definidos o dispositivo aloca espaço de memória no canal de transmissão e recepção de dados. O IRK (Identify Resolving Key) e o CSRK (Connection Signature Resolving Key) são chaves de privacidade e segurança do dispositivo, de 16 bytes, geradas pelo GAP.

Command	OpCode	Command Parameters	Return Param.
GAP_DeviceInit	0xFE00	ProfileRole:GAP_Profile_Central MaxScanResponses:0x03, value buffer space default IRK:16 bytes address CSRK:16 bytes address SignCounter:Default counter	Status

Figura 67: Definição do comando GAP DeviceInit na plataforma C#. [38]

Este comando gera um evento do tipo **HCI Ext Command Status**, com o parâmetro *status* que indica o estado do comando enviado. O parâmetro *status* assume “0x00” em caso de sucesso ou “0x02” se o parâmetro for inválido. Para além do evento **HCI Ext Command Status**, a *dongle* responde com um pacote estruturado do tipo **GAP DeviceInitDone** que confirma que o dispositivo terminou o processo de inicialização. A Figura 68 ilustra a forma estruturada de um evento **GAP DeviceInitDone**. Este evento, de *OpCode* 0x0600, contém informação relativa à *dongle* nomeadamente o endereço, o tamanho e número de pacotes de dados e os 16 bytes IRK e CSRK.

Event	OpCode	Event Parameters
GAP_DeviceInitDone	0x0600	Status DevAddr DataPktLen NumDataPkts IRK CSRK

Figura 68: Definição do evento GAP DeviceInitDone na plataforma C#. [38]

A Figura 69 ilustra a mensagem recebida na aplicação no envio do evento **GAP DeviceInitDone**.

```

-Type           : 0x04 (Event) ←0x04 Indica o tipo de pacote de dados HCI (0x04 = Evento)
-EventCode      : 0xFF (HCI_LE_ExtEvent)
-Data Length    : 0x2C (44) bytes(s) ←Tamanho do pacote de dados HCI
  Event         : 0x0600 (GAP_DeviceInitDone)←Evento HCI do tipo GAP_DeviceInitDone
  Status        : 0x00 (success)
  DevAddr       : 78:C5:E5:70:7F:9F←Endereço da dongle
  DataPktLen    : 0x001B (27) ←Tamanho e número de pacotes de dados
  NumDataPkts   : 0x04 (4)
  IRK           : 56:CD:11:3C:C5:0F:2C:90:DA:DD:E9:99:26:8D:0A:18←IRK e CSRK
  CSRK          : 07:F1:C0:03:AC:90:A2:7D:92:3B:DD:2F:29:9A:EA:9D
    
```

Figura 69: Mensagem gerada na aplicação C# no momento de execução do evento GAP DeviceInitDone.

Antes que a comunicação aconteça devem ser definidos parâmetros que configuram a ligação entre os dispositivos. Estes parâmetros correspondem aos intervalos máximo e mínimo de ligação, ao tempo de latência e ao *timeout*, tempo máximo admitido para a ligação. De forma a ler estes parâmetros GAP, a aplicação C# envia o comando HCI **GAP GetParameter** configurado de acordo com o ilustrado na Figura 70. Este comando, de *OpCode* “0xFE31”, identifica através de um valor, o ParamID, o parâmetro que pretende ler. O intervalo máximo e mínimo da ligação assumem o ParamID 21 e 22, respectivamente. Por outro lado o tempo de latência é identificado com o ParamID 26, e o *timeout* 25. Quando este comando é recebido, a *dongle*, envia um evento **HCI Ext Command Status** com o estado do comando enviado (0x00 em caso de sucesso e 0x02 se o ParamID for inválido) e o valor lido do parâmetro.

Command	OpCode	Command Parameters	Return Param.
GAP_GetParam	0xFE31	ParamID	Status ParamValue

Figura 70: Definição do comando GAP GetParameter na plataforma C#. [38]

A partir do momento em que é configurada e estabelecida a ligação da *dongle* com a ferramenta C# pode ser iniciada a procura de dispositivos com os quais a *dongle* se possa ligar. Este processo é iniciado com o botão “Scan”. Quando este botão é pressionado é enviado pela *dongle* o comando HCI **GAP DeviceDiscoveryRequest**. O **GAP DeviceDiscoveryRequest**, de *OpCode* 0xFE04, é o comando enviado para iniciar a procura, ou o *scan*, de pacotes de *advertising*. O comando é definido na aplicação de acordo com a Figura 71. No **GAP DeviceDiscoveryRequest** o *scan* foi configurado para procurar por pacotes de *advertising* em todos os dispositivos mantendo desligado o “ActiveScan”. A “WhiteList” é uma lista de dispositivos centrais com os quais um periférico pode aceitar pedidos de ligação. Durante o *scan* a “WhiteList” não é utilizada o que significa que a procura é feita por todos os dispositivos disponíveis para a ligação independentemente de estarem ou não listados na “WhiteList”.

Command	OpCode	Command Parameters	Return Param.
GAP_DeviceDiscoveryRequest	0xFE04	Mode:Scan for all devices ActiveScan:Turn off active scan WhiteList:Don't use the whitelist during scan	Status

Figura 71: Definição do comando GAP DeviceDiscoveryRequest na plataforma C#. [38]

Este comando gera um evento do tipo **HCI Ext Command Status**, com o parâmetro *status* que pode assumir o valor “0x00” em caso de sucesso, “0x11” se o *scan* não for possível ou “0x12” se o *scan* estiver configurado de forma imprópria.

O SensorTag, em resposta ao comando **GAP DeviceDiscoveryRequest**, envia os eventos **HCI GAP DeviceInformation** e **GAP DiscoveryDeviceDone**. A forma como o evento **GAP DeviceInformation** se estrutura é ilustrada na Figura 72.

Event	OpCode	Event Parameters
GAP_DeviceInformation	0x060D	Status EventTypes AddrType Addr Rssi DataLen DataField

Figura 72: Definição do evento GAP DeviceInformation na plataforma C#. [38]

Este evento, de *OpCode* 0x060D, gera a mensagem de dados apresentada na Figura 73.

```

msg -Type          : 0x04 (Event) ← 0x04 = Evento
-EventCode         : 0xFF (HCI_LE_ExtEvent)
-Data Length       : 0x21 (33) bytes(s) ← Tamanho do pacote de dados do evento
Event              : 0x060D (GAP_DeviceInformation) ← Evento HCI do tipo GAP_DeviceInformation
Status             : 0x00 (Success)
EventType          : 0x04 (Scan Response) ← Tipo de evento definido
AddrType           : 0x00 (Public)
Addr               : BC:6A:29:AC:29:7A ← Endereço do SensorTag
Rssi               : 0xBD (189)
DataLength         : 0x14 (20) ← Tamanho do pacote de dados do parâmetro
Data               : 0A:09:53:6E:73:6F:72:54:61:67:05:12:50:00:20:
                   03:02:0A:00
    
```

Figura 73: Mensagem gerada na aplicação C# no momento de execução do evento GAP DeviceInformation.

A Figura 74 ilustra o evento de *OpCode* “0x0601”, o **GAP DiscoveryDeviceDone**, que indica que o processo de *scan* foi terminado. O parâmetro “NumDevs” corresponde ao número de dispositivos detectados no momento do *scan* e o endereço corresponde ao endereço do dispositivo encontrado para estabelecer uma possível ligação.

Event	OpCode	Event Parameters
GAP_DeviceDiscovery	0x0601	Status NumDevs EventType AddrType Addr

Figura 74: Definição do evento GAP DeviceDiscovery na plataforma C#. [38]

O pacote de dados que é gerado no envio do evento **GAP DiscoveryDeviceDone** é ilustrado na Figura 75.

```

device discovery done -Type: 0x04 (Event) ←0x04 Indica o tipo de pacote de dados HCI
-EventCode           : 0xFF (HCI_LE_ExtEvent) ←Tamanho do pacote de dados do evento HCI
-Data Length         : 0x04 (4) bytes(s)
Event                : 0x0601 (GAP_DeviceDiscoveryDone) ←Evento HCI do tipo GAP_DeviceDiscoveryDone
Status               : 0x00 (Success)
NumDevs              : 0x01 (1) ← Número de dispositivos encontrados no scan
EventType            : 0x04 (Scan Response)
AddrType             : 0x00 (Public)
Addr                 : BC:6A:29:AC:29:7A
    
```

Figura 75: Mensagem gerada na aplicação C# no momento de execução do evento GAP DiscoveryDeviceDone.

Depois de encontrado um endereço válido para a ligação, que neste caso é o endereço do SensorTag, é estabelecida a ligação com o botão “Connect”. Quando este botão é pressionado na aplicação, a *dongle* termina a procura de dispositivos com o comando **GAP DeviceDiscoveryCancel**. Este comando é definido pelo *OpCode* “0XFE05”, na sua estrutura não é constituído por parâmetros e gera, à semelhança de outros comandos um evento do tipo **HCI Ext Command Status**, com o parâmetro *status* que indica o estado do comando. Para estabelecer a comunicação com o SensorTag é enviado pela *dongle* o comando **GAP EstablishLinkRequest** definido na Figura 76. O *OpCode* deste comando é “0XFE09” e os parâmetros foram configurados de forma a desabilitar o Duty Cycle na comunicação, não utilizar a “WhiteList” e estabelecer a ligação com um dispositivo periférico de endereço público de 6 bytes.

Command	OpCode	Command Parameters	Return Param.
GAP_EstablishLinkRequest	0xFE09	HighDutyCycle:Disable Duty Cycle WhiteList:Don't use the white list AddrTypePeer:Public address type PeerAddr:6 bytes address	Status

Figura 76: Definição dos comandos GAP DeviceDiscoveryCancel e GAP EstablishLinkRequest na plataforma C#. [38]

O comando referido na Figura 76 gera um evento do tipo **HCI Ext Command Status**, com o estado do comando que pode ser “0x00” em caso de sucesso, “0x10” se o dispositivo não estiver pronto para executar a acção por permanecer em modo *scan* ou “0x12” se o comando for incorrecto. O SensorTag envia o evento **GAP EstablishLink** em resposta ao comando enviado pela *dongle*, **GAP EstablishLinkRequest**. O evento definido com o *OpCode* “0x0605” tem como parâmetros os apresentados na Figura 77. A Figura 78 é relativa à mensagem criada na ferramenta C# no momento em que o evento, **GAP EstablishLink**, é enviado pelo SensorTag.

Event	OpCode	Event Parameters
GAP_LinkEstablished	0x0605	Status, Status, DevAddrType, DevAddr, ConnHandle, ConnInterval, ConnLatency, ConnTimeout, ClockAccuracy

Figura 77: Definição do evento GAP LinkEstablished na plataforma C#. [38]

```

establish link -Type: 0x04 (Event) ← 0x04 = Evento
-EventCode    : 0xFF (HCI_LE_ExtEvent)
-Data Length  : 0x13 (19) bytes(s) ← Tamanho do pacote de dados do evento HCI
Event        : 0x0605 (GAP_EstablishLink) ← Evento HCI do tipo GAP_EstablishLink
Status       : 0x00 (Success)
DevAddrType  : 0x00 (Public)
DevAddr      : BC:6A:29:AC:29:7A ← Endereço do SensorTag
ConnHandle   : 0x0000 (0)
ConnInterval : 0x0050 (80)
ConnLatency  : 0x0000 (0)
ConnTimeout  : 0x07D0 (2000)
ClockAccuracy: 0x00 (0)

```

Parâmetros que configuram a ligação entre a *dongle* e o SensorTag

Figura 78: Mensagem gerada na aplicação C# no momento de execução do evento GAP EstablishLink.

Analisando a Figura 78, o parâmetro “ConnHandle” corresponde à *Handle* da ligação. Por outro lado, o parâmetro “ConnInterval” é o intervalo de tempo entre cada ligação e o “ConnTimeout” o tempo de *Timeout* entre duas ligações bem sucedidas. O tempo de latência foi definido a 0. De acordo com a equação 6.1, vem que:

$$\text{Intervalo de conexão} = (80 \times 1,25) \text{ms} = 0,1 \text{ s} \quad 7.2.$$

$$\text{Timeout da conexão} = (2000 \times 1,25) \text{ms} = 2,5 \text{ s} \quad 7.3.$$

Depois de estabelecida a ligação lógica ao dispositivo central, a *dongle*, é possível obter dados dos serviços disponibilizados pelo dispositivo periférico, o SensorTag, de acordo com as definições do BLE.

7.2.2. Aquisição de dados

A aquisição e visualização de dados são feitas através dos separadores relativos aos módulos. Da mesma forma que foi feito no Capítulo anterior, com recurso à Btool, vai ser descrita a activação do serviço do acelerómetro através da aplicação em C#. Os dados adquiridos correspondem aos valores do acelerómetro, do magnetómetro e do giroscópio no SensorTag. Tal como referido anteriormente, devido às semelhanças de *HW* entre o SensorTag e o módulo da pagaia (numa versão final do módulo), os dados adquiridos pelo SensorTag são disponibilizados no separador da aplicação relativo à informação do módulo da pagaia.

A ligação do serviço do acelerómetro implica a activação da característica, ou atributo, que configura o acelerómetro, seguida da activação das notificações do sensor. As duas são desencadeadas pelo envio de um pedido **GATT WriteLongCharValue** pela *dongle*. O acesso, leitura e escrita de dados em características específicas de um serviço do servidor é gerido pela camada **GATT**. A troca de dados, relativos a serviços, entre os dispositivos, é dependente da camada **GATT** que, baseada no protocolo ATT, define a forma como a troca de informação pode acontecer. A *dongle*, cliente GATT, pretende escrever o valor de uma característica no SensorTag, servidor GATT, de forma a activar um serviço e as respectivas notificações. Para o efeito, a *dongle* deve aceder à *Handle* que determina o campo do servidor que guarda a respectiva característica. Esta informação está descrita na tabela de atributos do SensorTag onde são declarados os serviços, as características e os valores das características. O comando **GATT**

WriteLongCharValue apresenta o formato representado na Figura 79. Este comando serve para escrever um valor característico numa *Handle*, neste caso a *Handle* que se associa à activação do serviço do acelerómetro.

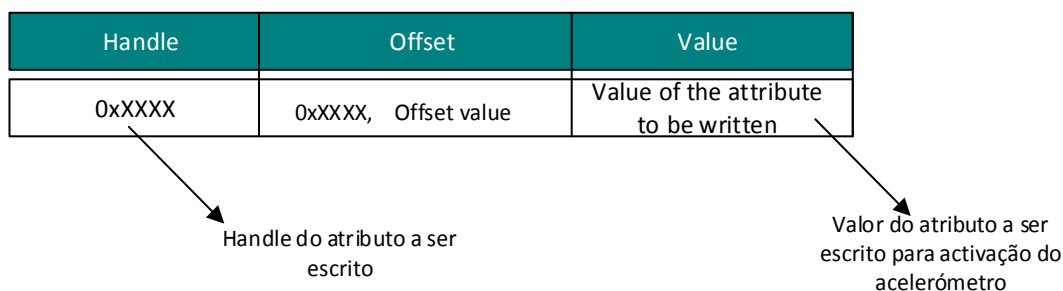


Figura 79: Formato e definição do comando GATT WriteLongCharValue.

Recorrendo à tabela de atributos do SensorTag, para a activação do serviço do acelerómetro, é possível perceber os valores que o comando **GATT WriteLongCharValue** deve conter na sua estrutura. A Figura 80 representa a forma como o atributo de configuração do acelerómetro pode ser endereçado e activado. O endereço do atributo corresponde à sua *Handle* característica, “0x034”, e de acordo com a descrição com o valor “01”, “02” ou “03” é activo o acelerómetro e definida a sua componente gravítica. Uma vez que o pretendido é activar o serviço, o GATT tem que autorizar permissões de escrita no atributo do servidor.

Handle (hex)	Handle (dec)	Type (hex)	Type (text)	Hex value	Gatt Server Permissions	Description/Value (text)
0x034	52	0xAA12	Accelerometer Config	00	RW	Write "01" to select range 2G, "02" for 4G, "03" for 8G, "00" disable sensor

Figura 80: Definição do atributo de activação do acelerómetro.

Assim, o comando **GATT WriteLongCharValue** deve ser definido para na *Handle* “0x034” escrever o valor “03” para que o acelerómetro seja activo com uma componente gravítica de 8G.

Depois deste comando, o SensorTag responde com um **ATT WriteRsp** que é um evento de resposta e confirmação a um comando GATT. O **ATT Write Rsp** indica se o comando **GATT WriteLongCharValue** foi executado com sucesso.

Para receber valores de aceleração dos três eixos do sensor é necessário activar as notificações do acelerómetro. A Figura 81 ilustra a tabela de atributos do SensorTag para o atributo relativo às notificações do acelerómetro. A tabela permite definir que o comando **GATT WriteLongCharValue** deve conter como *Handle* o valor “0x031” e escrever o valor “01:00” para que as notificações do acelerómetro sejam activas.

Handle (hex)	Handle (dec)	Type (hex)	Type (text)	Hex value	Gatt Server Permissions	Description/Value (text)
0x031	49	0x2902	Client Characteristic	00:00	RW	Write "01:00" to enable notifications, "00:00" to disable

Figura 81: Definição do atributo de activação das notificações do acelerómetro.

Os valores de aceleração são dados através de eventos **ATT_HandleValueNotification** enviados pelo SensorTag à *dongle*. Um **ATT_HandleValueNotification** é utilizado quando o servidor é configurado para notificar o cliente com valores característicos a qualquer momento e sem receber a confirmação de que as notificações estão a ser recebidas com sucesso.

Aos pacotes de dados recebidos na aplicação com valores de aceleração, anteriormente ilustrados na Figura 63, é feito o processamento que individualiza os dados por eixo. A Figura 82 ilustra a visualização de dados relativos ao acelerómetro, com e sem componente gravítica, na aplicação C#.

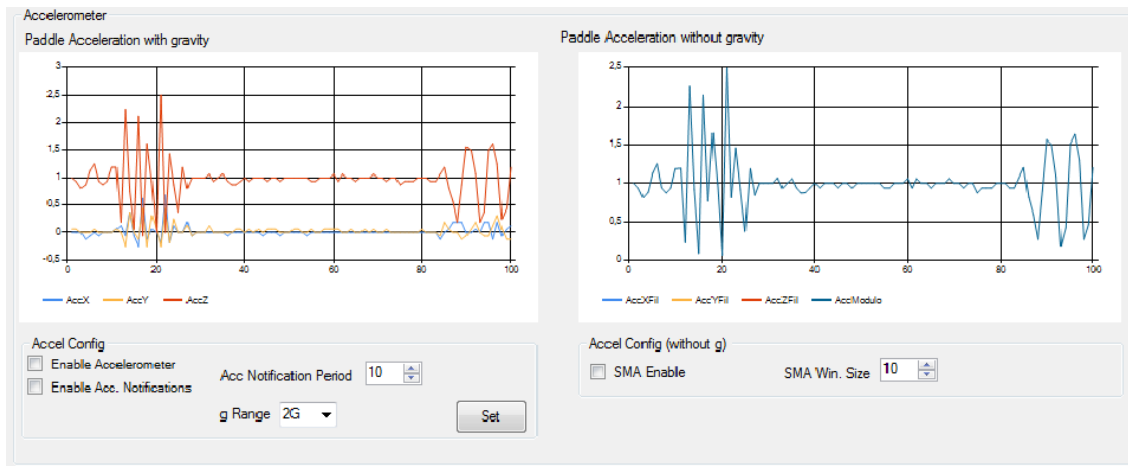


Figura 82: Gráfico de visualização dos valores do acelerómetro, com e sem componente gravítica, na aplicação C#.

Um processo semelhante ao descrito, para activação do acelerómetro e das suas notificações, foi feito para receber dados do giroscópio e do magnetómetro. As Figura 83 e 84 ilustram a tabela de atributos do SensorTag com os atributos relativos à activação do magnetómetro e do giroscópio e respectivas notificações.

Handle (hex)	Handle (dec)	Type (hex)	Type (text)	Hex value	Gatt Server Permissions	Description/Value (text)
0x4A	74	0xAA32	Magnetometer Config	00	RW	Write "01" to start measurements, "00" to put to sleep

Handle (hex)	Handle (dec)	Type (hex)	Type (text)	Hex value	Gatt Server Permissions	Description/Value (text)
0x047	71	0x2902	Client Characteristic	00:00	RW	Write "01:00" to enable notifications, "00:00" to disable

Figura 83: Definição dos atributos, relativos ao serviço do magnetómetro, que activam o sensor e as notificações do sensor. [40]

O comando **GATT WriteLongCharValue** deve ser definido para na *Handle* "0x4A" escrever o valor "01" para que o magnetómetro seja activo. Para activar as notificações do sensor para a *Handle* "0x047" foi escrito o valor "01:00"

Para activar o giroscópio, de acordo com a Figura 84, o comando **GATT WriteLongCharValue** deve escrever para a *Handle* “0X64” o valor “7”. As notificações são activas com o valor “01:00” na *Handle* “0x061”.

Handle (hex)	Handle (dec)	Type (hex)	Type (text)	Hex value	Gatt Server Permissions	Description/Value (text)
0x061	97	0x2902	Client Characteristic	00:00	RW	Write “01:00” to enable notifications, “00:00” to disable
0x64	100	0xAA52	Gyroscope Config	00	RW	Write 0 to turn off gyroscope, 1 to enable X axis only, 2 to enable Y axis only, 3 = X and Y, 4 =Z only, 5 = X and Z, 6 = Y and Z, 7 = X, Y and Z

Figura 84: Definição dos atributos, relativos ao serviço do giroscópio, que activam o sensor e as notificações do sensor. [40]

A Figura 85 ilustra os gráficos do magnetómetro e do giroscópio, na aplicação C#, no momento de aquisição de valores dos sensores.

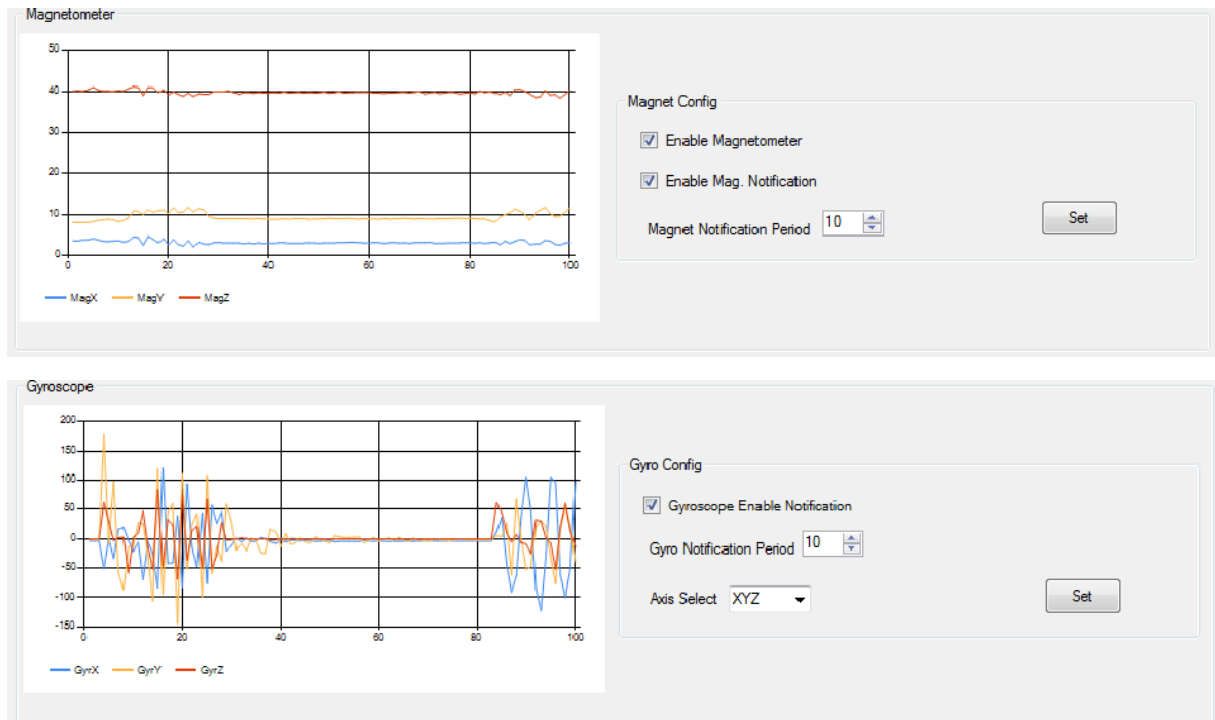


Figura 85: Gráficos de visualização dos valores magnetómetro (imagem superior) e do giroscópio (imagem inferior) na aplicação C#.

A ligação entre a *dongle* e o *SensorTag* pode ser terminada a qualquer momento com o envio do comando **GAP TerminateLinkRequest**. A Figura 86 ilustra a estrutura de um comando **GAP TerminateLinkRequest**, de OpCode“0XFE0A”. Este comando contém o parâmetro “connHandle” que permite, de acordo com a sua configuração, terminar a ligação com uma *Handle*. Para terminar a ligação a uma *Handle* específica de um atributo, o valor de configuração varia entre 0 e 0XFFFD. Por outro lado, para terminar o “Establish Link Request”

o comando deve ser configurado para a *Handle* “0XFFFE” e para terminar todas as ligações “0XFFFF”.

Command	OpCode	Command Parameters	Return Param.
GAP_TerminateLinkRequest	0xFE0A	connHandle	status

Figura 86: Definição do comando GAP TerminateLinkRequest na plataforma C#. [38]

Este comando gera um evento do tipo **HCI Ext Command Status**, com o parâmetro *status*.

CAPÍTULO 8 - CONCLUSÕES

8.1. CONCLUSÕES

A aplicação de redes de sensores sem fios é cada vez mais comum e abrangente. No desporto, o comportamento destas redes é promissor quando, sem comprometer a prática do exercício e o conforto do atleta, se integram para adquirir dados em momento de treino ou competição. O acesso a esta informação e controlo da técnica são a tendência para um atleta, amador ou profissional, com a ambição de se superar constantemente.

Esta dissertação tinha como principal objectivo a apresentação de uma ferramenta para o desenvolvimento de um sistema de monitorização desportiva, nomeadamente para a modalidade da canoagem de velocidade. A ferramenta descrita é baseada numa rede sem fios com três nós que correspondem a módulos sensoriais. Cada um destes módulos diz respeito a uma componente crucial da modalidade. Adquirir e interpretar de forma individual os dados relativos à pagaia, ao barco e ao finca-pés pode ser importante para um atleta melhorar e corrigir a técnica. A ferramenta apresentada consiste num estudo de um possível modo de aquisição de dados na canoagem recorrendo a uma rede sensorial sem fios e protocolos de comunicação. Na arquitectura estudada os módulos inerciais, do barco e da pagaia, e o módulo do finca-pés (com sensores de força) adquirem informação de forma individualizada mas que se complementa a cada instante de tempo. Na rede estabelecida é o módulo do barco quem assume o papel de dispositivo central e reúne os dados adquiridos pela pagaia e o finca-pés, e os envia para a margem para um módulo receptor. Distinguem-se assim dois protocolos em toda a rede: o BLE para a troca de dados entre os módulos do finca-pés, do barco e da pagaia, e entre o módulo receptor da margem e os dispositivos terminais; e a comunicação a longas distâncias para que os dados recolhidos pelo módulo do barco cheguem ao elemento receptor na margem. Esta é a descrição da arquitectura da ferramenta proposta no entanto, a evolução natural do trabalho fez com que as necessidades mudassem e o plano inicialmente delineado para este estudo também. Para provar a arquitectura pensada para a ferramenta foi feita uma prova de conceito que se considerou como a versão 1.0. da ferramenta. Com a versão 1.0. foi implementada uma rede entre o módulo da pagaia e o módulo do barco. O módulo do finca-pés não foi incluído na rede para obtenção de resultados por, até à data da versão 1.0., não estar concluído. Na rede estabelecida entre os dois módulos, da pagaia e do barco, optou-se por comunicação a curtas distâncias, com um módulo básico de RF e para receber os dados na margem por comunicação a longas distâncias. Os resultados obtidos desta prova conceito permitiram perceber que: era possível adquirir dados relevantes de pagaiadas, a comunicação estabelecida para longas distâncias era suficiente e eficaz e que a comunicação RF implementada para curtas distâncias não era apropriada. A versão 1.0. permitiu testar a rede e validar a aquisição de dados e o protocolo de comunicações a longas distâncias. Por ser necessária a implementação de um protocolo de comunicação a curtas distâncias e de baixo consumo, foi iniciado o estudo do BLE, o protocolo considerado como mais adequado à ferramenta. O SensorTag intervém no

trabalho como uma ferramenta auxiliar por dois motivos. Se por um lado se assemelha ao *HW* do módulo da pagaia, numa versão final, por outro permite a compreensão de forma clara e em pormenor do BLE. O facto de com o SensorTag poder ser utilizada a Btool consistiu numa vantagem para a optimização e implementação do BLE na aplicação C#, optimizada para a ferramenta a partir de código fornecido pela Texas InstrumentsTM. Esta dissertação foca a análise do protocolo de comunicação BLE essencial na optimização e direccionamento da aplicação C# às necessidades e características da ferramenta. Esta aplicação permitiu validar a aquisição, processamento e visualização de forma gráfica dos dados de um módulo sensorial BLE. É importante ter em mente que, apesar de o estudo ser feito com o SensorTag, pode ser substituído por um módulo BLE proprietário.

Em conclusão, apesar de alguns elementos não terem sido concluídos ou implementados, dada a extensão do trabalho, o estudo apresentado nesta dissertação demonstra que os conceitos defendidos por esta ferramenta são válidos e que a mesma, quando considerada como um método de aquisição de dados da canoagem de velocidade, pode contribuir para o desenvolvimento de um sistema de monitorização da modalidade.

8.2. SUGESTÕES PARA TRABALHO FUTURO

Como trabalho futuro propõe-se a integração de um *IMU* no módulo sensorial da pagaia e a conclusão e incorporação do módulo do finca-pés na rede. Porque o estudo do BLE, foi o foco desta dissertação, a implementação deste protocolo entre os módulos da pagaia, do barco e do finca-pés, é uma tarefa plausível de integrar na ferramenta até aqui desenvolvida. A nível aplicacional, a recepção de dados do módulo do barco e do finca-pés é importante para completar o seu funcionamento e objectivo. A melhoria, da interface com o utilizador, da aplicação é igualmente uma sugestão para trabalho futuro a considerar.

A longo prazo, dada a complexidade e extensão do trabalho envolvido, é sugerido o desenvolvimento do sistema, tal como especificado, com a integração da ferramenta discutida nesta dissertação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] D. Sturm, "Wireless Multi-Sensor Feedback Systems for Sports, Doctoral Thesis," KTH Technology and Health, Stockholm, 2012.
- [2] R. S. B. R. Jacob S. Michael, "Determinants of kayak paddling performance," *Sports Biomechanics*, pp. 167-179, 2009.
- [3] B. D. N. M. Neil Fleming, "A biomechanical assessment of ergometer task specificity in elite flatwater," Physiology Department, Anatomy Department, Trinity College Dublin, Ireland, 2012.
- [4] R. H. S. Selina J. Kendal, "The technique of elite flatwater kayak paddlers using the wing paddle," *International Journal of Sport Biomechanics*, 8,, p. 233–250, 1992.
- [5] D. R. R. S. B. K. John Baker, "A three dimensional analysis of male and female elite sprint kayak paddlers," *17 International Symposium on Biomechanics in Sports* , pp. 53-56, 1999.
- [6] C. R. S. J. López López, "Quantitative analysis of kayak paddling technique: definition of an optimal stroke profile," em *Revista Andaluza de Medicina del Deporte*, 2011, pp. 91-95.
- [7] B. E. T. A. A. L. Kuan Ong, "Performance tolerance and boat set-up in elite sprint Kayaking," School of Human Movement and Exercise Science , University of Western Australia; Western Australia Institute of Sport, Australia, 2010.
- [8] L. M. Francesca Di Puccio, "Kayak rowing: kinematic simulation of different techniques," *Journal of Biomechanics, 16th ESB Congress, Human Motion*, pp. Poster P-52, 2008.
- [9] K. S. M. M. K. FUNATO, "Development of Paddling Tank Equipped with Circulating Water Channel (CWC);," *Journal of Biomechanics*, Vols. %1 de %2XXI ISB Congress, Poster Sessions, 2007.
- [10] F. C. M. Begon, "A kayak ergometer using a sliding trolley to reproduce accurate on-water mechanical conditions," *Journal of Biomechanics*, vol. XXI ISB Congress, 2007.
- [11] N. D. T. R. Daniel A. James, "An Accelerometer Based Sensor Platform for Insitu Elite Athlete," *Sensors, 2004. Proceedings of IEEE*, vol. 3, pp. 1373 - 1376, 2004.
- [12] R. F. A. B. J. B. I. Helmer, "Instrumentation of a kayak paddle to investigate blade/water interactions," *Procedia Engineering, 13:501–506*, Vols. %1 de %25th Asia-Pacific Congress on Sports Technology (APCST), 2011.
- [13] N. V. R. S. F. C. J. P. V.-B. M. V. B. Gomes, "Analysis of the on-water paddling force profile of an elite kayaker," em *29th Int. Symp. Biomechanics in Sports*, 259-262, p. 2011.
- [14] L. E. H. T. W. P. K. F. Michael G. Robinson, "Accelerometry Measurements of Sprint kayaks: The coaches' New Tool," *International Journal of Coaching Science*, Vols. %1 de %25, No.1, pp. 3-25, 2011.
- [15] K. Y. M. E. Dennis Sturm, "A wireless, unobtrusive Kayak Sensor Network enabling Feedback Solutions," *2010 International Conference on Body Sensor Networks*, pp. 159-163, 2010.

- [16] D. C. R. Hayden Croft, "Developing and applying a tri-axial accelerometer sensor for," *6th Asia-Pacific Congress on Sports Technology (APCST)*, pp. 16-21, 2013.
- [17] H. D. B. Pereira, "Relação entre a força máxima/força explosiva e a performance anaeróbia em canoagem," Universidade de Lisboa, Faculdade de Motricidade Humana, Lisboa, 2011.
- [18] Y. T., U. M., I. K., C. M., I. M., N. F. e Y. T., "Significance of the contribution of aerobic and anaerobic components to several distance running performances in female athletes.," *European Journal Of Applied Physiology & Occupational Physiology*, pp. 249-253, 1990.
- [19] M. Sousa, "Projecto e concepção de um dispositivo experimental para medição e caracterização dos esforços aplicados no finca-pés de um caiaque de pista," Porto, 2012.
- [20] J. T. K. W. C. Byrnes, "Aerobic and Anaerobic Contributions During Simulated Canoe/Kayak Sprint Events," *Medicine and Science in Sports and Exercise*, 1997.
- [21] T. K. Isaka T., "Effects of off- and pre-season training on aerobic and anaerobic power of kayak paddlers.," *Medicine and Science in Sports and Exercise*, 1997.
- [22] D. G. J. Paul B. Laursen, "Optimising Training Programmes and Maximising Performance in Highly Trained Endurance Athletes," *Sports Medicine*, pp. 53-73, 2002.
- [23] Z. Alliance, "Zigbee Specification," 2008.
- [24] J. K. Jay Hendrix, "ZigBee : Overview," 2009.
- [25] M. A. A. K. Anneleen Van Nieuwenhuysse, "On the use of the ZigBee protocol for Wireless Sensor Networks," Polytechnic Institute of Porto, School of Engineering, 2006.
- [26] V. Samosuyev, "Bluetooth Low Energy Compared to Zigbee and Bluetooth Classic Bachelor's Thesis," 2010.
- [27] Blackberry, "BlackBerry Device Support Forum - Bluetooth LE primer for developers," Blackberry, [Online]. Available: <https://supportforums.blackberry.com/t5/Native-Development-Knowledge/BlackBerry-10-Bluetooth-LE-primer-for-developers/ta-p/2287377>. [Acedido em 2015].
- [28] R. M. Pedro, "Bluetooth v4.0: a futura solução de baixo consumo," 2011.
- [29] T. Instruments, "Texas Instruments CC2540/41 Bluetooth Low Energy Software Developer's Guide v1.3.2," Texas Instruments , 2010-2013.
- [30] T. Instruments, "LPRF San Diego Bluetooth Low Energy Deep Dive," Texas Instruments, 2011.
- [31] N. Gupta, *Inside Bluetooth Low Energy*, Artech House, 2013.
- [32] N. Semicondutor, "Creating Bluetooth Low Energy Applications Using nRF51822, Application Note v1.0".
- [33] T. Instruments, "TI BLE Vendor Specific HCI Reference Guide Version 1.4," Texas Instruments.
- [34] K. Townsend, C. Cufí, Akiba e R. Davidson, *Getting Started with Bluetooth Low Energy*, 2015.
- [35] T. Instruments, "SensorTag attribute table," Texas Instruments.

- [36] A. Dementyev, S. Hodges, S. Taylor e J. Smith, "Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario," *Wireless Symposium (IWS), 2013 IEEE International*, 2013.
- [37] B. Sensortec, "SMB380 Triaxial acceleration sensor Data Sheet(preliminary)," 2007.
- [38] M. António, "Bluetooth Low Energy para a monitorização da postura no ciclismo," Universidade do Minho, 2014.
- [39] N. Semiconductor, "nRF24L01+ Preliminary Product Specification v1.0".
- [40] G. Vadai, Z. Gingl, R. Mingesz e G. Makan, ""Performance estimation of kayak paddlers based on fluctuation analysis of movement signals"," em *Noise and Fluctuations (ICNF), 2013 22nd International Conference on*, 2013, pp. 1-4.
- [41] T. Instruments, "SensorTag User Guide," Texas Instruments, [Online]. Available: http://processors.wiki.ti.com/index.php/SensorTag_User_Guide. [Acedido em 2015].
- [42] S. Patrão e A. J. Pedro, "Ferramenta de teste e verificação para sensores inerciais e algoritmos de posicionamento," *ICE UBI*, 2015.
- [43] T. Instruments, "Bluetooth Low Energy CC2540/41 Mini Development Kit User's Guide," Texas Instruments.
- [44] A. F. J. B. I. B. R. J. N. Helmer, "Instrumentation of a kayak paddle to investigate blade/water," *5th Asia-Pacific Congress on Sports Technology*, vol. 13, p. 501–506, 2011.
- [45] B. J. Sperlich J., "Biomechanical testing in elite canoeing," *Scientific Proceedings of the XX International Symposium on Biomechanics in Sport*, pp. 44-47, 2000.
- [46] J. S. S. R. M. R. K. B. Michael, "Determinants of kayak paddling," *Sports Biomechanics*, vol. 8, nº 2, pp. 167-179, 2009.
- [47] H. K. E. S. M. a. V. J. Mononen, "Paddle force characteristics during 200m kayaking," *In International Congress on Applied Research in Sport*, pp. 151-154, 1994.
- [48] N. Q. M. a. S. S. Petrone, "A load acquisition device for the paddling action on olympic kayak," *Experimental mechanics, advances in design, testing and analysis: proceedings of XI ICEM*, vol. 2, pp. 817-822, 1998.
- [49] Tekscan, "FlexiForce Standard Model A201 datasheet".
- [50] M. W. D. Center, "D3DXMatrixRotationYawPitchRoll function," Microsoft, [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/bb205361\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb205361(v=vs.85).aspx). [Acedido em 2015].

