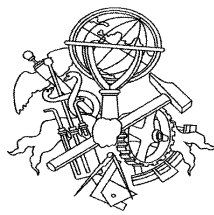


INPERLYS – INDEPENDENT PERSONAL LOCATION SYSTEM

Hélder António Santos Ferreira



Mestrado em Engenharia Electrotécnica e de Computadores

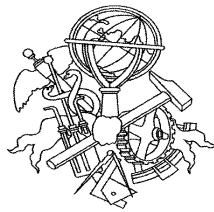
Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2010

Candidato: Helder António Santos Ferreira, 1030333@isep.ipp.pt
Orientação científica: Lino Figueiredo, lbf@isep.ipp.pt e António Meireles,
atmeireles@googlemail.com



Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Automação e Sistemas
Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia do Porto
18 de Novembro de 2010

Resumo

O desenvolvimento de sistemas de localização pedestre com recurso a técnicas de *dead reckoning* tem mostrado ser uma área em expansão no mundo académico e não só. Existem algumas soluções criadas, no entanto, nem todas as soluções serão facilmente implementadas no mercado, quer seja pelo *hardware* caro, ou pelo sistema em si, que é desenvolvido tendo em conta um cenário em particular.

INPERLYS é um sistema que visa apresentar uma solução de localização pedestre, independentemente do cenário, utilizando recursos que poderão ser facilmente usados. Trata-se de um sistema que utiliza uma técnica de *dead reckonig* para dar a localização do utilizador. Em cenários *outdoor*, um receptor GPS fornece a posição do utilizador, fornecendo uma posição absoluta ao sistema. Quando não é possível utilizar o GPS, recorre-se a um sensor MEMS e a uma bússola para se obter posições relativas à última posição válida do GPS. Para interligar todos os sensores foi utilizado o protocolo de comunicações sem fios ZigBee™. A escolha recaiu neste protocolo devido a factores como os seus baixos consumos e o seu baixo custo. Assim o sistema torna-se de uso fácil e confortável para o utilizador, ao contrário de sistemas similares desenvolvidos, que utilizam cabos para interligarem os diferentes componentes do sistema. O sensor MEMS do tipo acelerómetro tem a função de ler a aceleração horizontal, ao nível do pé. Esta aceleração será usada por um algoritmo de reconhecimento do padrão das acelerações para se detectar os passos dados. Após a detecção do passo, a aceleração máxima registada nesse passo é fornecida ao coordenador, para se obter o deslocamento efectuado.

Foram efectuados alguns testes para se perceber a eficiência do INPERLYS. Os testes decorreram num percurso plano, efectuados a uma velocidade normal e com passadas normais.

Verificou-se que, neste momento, o desempenho do sistema poderá ser melhorado, quer seja a nível de gestão das comunicações, quer a nível do reconhecimento do padrão da aceleração horizontal, essencial para se detectar os passos. No entanto o sistema é capaz de fornecer a posição através do GPS, quando é possível a sua utilização, e é capaz de fornecer a orientação do movimento.

Palavras-Chave

Localização Pedestre, *dead reckoning*, MEMS, GPS, ZigBee™, reconhecimento de padrão da aceleração.

Abstract

The development of pedestrian localization systems using the dead reckoning technique has been growing not only in academic world, but also in the industrial one. There are some solutions that can be found, however, not all of them are easy to be implemented due to factors like expensive hardware, or the system architecture which was developed to work in a particular scenario.

INPERLYS is a pedestrian location system intended to give a solution for pedestrian location, regardless of the surrounding environment, using resources that can be handled easily. It implements a dead reckoning technique to achieve the user localization. In outdoor environments, it uses the GPS to give the localization, providing an absolute coordinate to the system. When it is not possible to establish communication between GPS receiver and satellites, the system uses a MEMS sensor to estimate the distance traveled and a digital compass for orientation. To connect the sensors and the GPS, it uses the wireless ZigBee™ network protocol. This protocol was chosen because it has low power consumption and low cost. The system uses a wireless network because it is comfortable to use with clothes, unlike many similar systems, that use cables to connect all modules of the system. The MEMS accelerometer sensor has the purpose of measuring the horizontal acceleration, at foot level. This acceleration will be used by an algorithm developed to detect steps by recognizing particular patterns in the measured acceleration. When a step is detected, it will be sent to the network coordinator the maximum measured step acceleration, in order to obtain the distance traveled.

The system was evaluated in order to understand its reliability. The test consisted in user walking in a flat path, at constant velocity.

As a result from the test, I realize that the system performance can be improved. The acceleration messages routing management can be enhanced and there are some steps that are not detected. However, the system is able to give the user's position, whenever it's possible to use GPS and it is also able to provide the user's movement orientation.

Keywords

Pedestrian Localization, dead reckoning, MEMS, GPS, ZigBee™, acceleration pattern recognition.

Índice

RESUMO	III
ABSTRACT	VII
ÍNDICE	X
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABELAS	XVII
ACRÓNIMOS.....	XIX
1. INTRODUÇÃO	23
1.1. CONTEXTUALIZAÇÃO	23
1.2. OBJECTIVOS.....	24
1.3. CALENDARIZAÇÃO	24
1.4. ORGANIZAÇÃO DO RELATÓRIO	25
2. ANÁLISE DO PASSO HUMANO	27
2.1. ANÁLISE DAS ACELERAÇÕES QUE OCORREM DURANTE O PASSO	27
2.2. MÉTODOS PARA DETECÇÃO, MEDIÇÃO E ANÁLISE DO PASSO	30
3. SISTEMAS DE LOCALIZAÇÃO PEDESTRE.....	33
3.1. INTRODUÇÃO	33
3.2. SISTEMA SATÉLITE <i>GLOBAL POSITIONING SYSTEM</i>	33
3.3. SISTEMAS DE LOCALIZAÇÃO PEDESTRE DESENVOLVIDOS	40
4. SENSORES PARA SISTEMAS DE LOCALIZAÇÃO PEDESTRE.....	51
4.1. MEMS.....	51
4.2. BÚSSOLA	53
5. REDE ZIGBEE™	55
5.1. INTRODUÇÃO	55
5.2. NORMA 802.15.4	56
5.3. CAMADA FÍSICA DA NORMA 802.15.4	57
5.4. CAMADA MAC DA NORMA 802.15.4	58
5.5. CAMADA DE REDE NWK.....	60
5.6. CAMADA DE APLICAÇÃO APL.....	62
5.7. ESTRUTURA DAS TRAMAS DO ZIGBEE™	64
6. INPERLYS	67
6.1. DESCRIÇÃO DO SISTEMA.....	67

6.2.	ARQUITECTURA	68
6.3.	DESCRIÇÃO DO <i>HARDWARE</i>	70
6.4.	DESCRIÇÃO DO <i>SOFTWARE</i>	80
6.5.	ANÁLISE DE RESULTADOS	94
7.	CONCLUSÕES E FUTUROS DESENVOLVIMENTOS	99
	REFERÊNCIAS DOCUMENTAIS	101

Índice de Figuras

Figura 1 Ciclo do passo humano [10].....	28
Figura 2 Forças que actuam na perna quando esta se encontra em movimento [10]	28
Figura 3 Padrão horizontal da aceleração [10].....	29
Figura 4 Padrão vertical da aceleração [10].....	29
Figura 5 Forças repulsivas que actuam na perna [10]	29
Figura 6 Padrão da aceleração vertical durante um passo [10]	30
Figura 7 Padrão da aceleração horizontal durante um passo [10].....	30
Figura 8 Componentes da velocidade do pé no plano 3D [18].....	31
Figura 9 Ponto em que é aplicada a técnica ZUPT [19].....	32
Figura 10 Cenário de localização. No lado direito é visível o ponto onde as 3 circunferências coincidem [20]	35
Figura 11 Problemática da distribuição da constelação de satélites e a sua influência na localização [20].....	36
Figura 12 Figura geométrica Elipsóide Biaxial [20].....	37
Figura 13 Plano Equatorial e Meridiano [20]	38
Figura 14 Exemplo de trama GPS MNEA 0183 do tipo GGA	39
Figura 15 Esquema geral de suporte ao AGPS [21]	40
Figura 16 Disposição dos sensores no corpo do utilizador [16]	41
Figura 17 Posição do calcanhar quando o outro pé está totalmente em contacto com o chão [16]	42
Figura 18 Arquitectura do PeNa.....	42
Figura 19 Distâncias máxima e mínima calculadas pelo SiLMU [15]	43
Figura 20 Sistema Inercial montado no sapato do utilizador [19]	44
Figura 21 Arquitectura do NavShoe [22]	45
Figura 22 Disposição dos sensores que compõe o sistema no corpo do utilizador [2]	46
Figura 23 Arquitectura do algoritmo de fusão sensorial [2].....	48
Figura 24 Princípio de funcionamento de um acelerómetro capacitivo [26].....	53
Figura 25 Placas fixas e móveis de um acelerómetro capacitivo [23].	53
Figura 26 Distribuição dos protocolos <i>wireless</i> de acordo com o seu alcance [28].	55
Figura 27 Arquitectura ZigBee™ [36].....	57
Figura 28 Composição da trama MAC [38]	59
Figura 29 Troca de informação entre coordenador e dispositivo da rede [38].....	60
Figura 30 Tipologia Estrela [39].....	61
Figura 31 Tipologia do tipo Árvore [39].....	61

Figura 32 Tipologia do tipo Malha [39]	62
Figura 33 Arquitectura da <i>stack</i> da Microchip [®] para ZigBee [™] [39].....	63
Figura 34 Exemplo de interacção entre dispositivos usando um <i>Profile</i> [39].....	64
Figura 35 Estrutura das tramas do protocolo ZigBee [™]	65
Figura 36 Sistema INPERLYS	68
Figura 37 Arquitectura do sistema INPERLYS	69
Figura 38 Exemplo de módulo ZigBee [™] [41].....	70
Figura 39 PIC18LF4620 [42]	71
Figura 40 Antena do Kit PICDEM [™] Z [41].....	72
Figura 41 Transreceptor utilizado na antena do <i>kit</i> [43].....	73
Figura 42 Bússola Digital CMPS03 [44]	74
Figura 43 Ligação Eléctrica do MEMS [45].....	75
Figura 44 <i>Pins</i> GPS [46].....	77
Figura 45 Módulo Bússola digital	79
Figura 46 Módulo GPS.....	79
Figura 47 Módulo MEMS	80
Figura 48 Rotina Leitura trama GGA do GPS.....	86
Figura 49 Descrição pontos chave na análise ao padrão da aceleração horizontal	88
Figura 50 Rotina reconhecimento do padrão.....	89
Figura 51 Aceleração impressa no <i>hyperterminal</i> após recepção do coordenador	90
Figura 52 Orientação impressa no <i>hyperterminal</i> após recepção do coordenador	91
Figura 53 Fluxograma da leitura do módulo da bússola e envio de dados para o coordenador	91
Figura 54 Sistema de referência para <i>clocks</i>	93

Índice de Tabelas

Tabela 1 Planeamento de execução das tarefas	25
Tabela 2 Significado de alguns dados da trama GPS NMEA 0183 do tipo GGA.....	39
Tabela 3 Descrição <i>pins</i> bússola digital	74
Tabela 4 Descrição <i>pins</i> MEMS [45]	76
Tabela 5 Descrição <i>pins</i> GPS [46]	78

Acrónimos

3GPP	-	<i>3rd Generation Partnership Project</i>
ACK	-	<i>Acknowledgment</i>
APL	-	<i>Application Layer</i>
APO	-	<i>Application Objects</i>
APS	-	<i>Application Support Sub Layer</i>
ASCII	-	<i>American Standard Code for Information Interchange</i>
BPSK	-	<i>Binary Phase Shift Keying</i>
CAP	-	<i>Contention Access Period</i>
CFP	-	<i>Contention Free Period</i>
CMOS	-	<i>Complementary Metal-Oxide Semiconductor</i>
CRs	-	<i>Cognitive Radios</i>
CRC	-	<i>Cyclic Redundancy Check</i>
CSMA-CA	-	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CTRS	-	<i>Conventional Terrestrial Reference System</i>
DR	-	<i>Dead Reckoning</i>
DRC	-	<i>Dimmer Remote Control</i>
DSSS	-	<i>Direct Sequence Spread Spectrum</i>
EEPROM	-	<i>Electrically Erasable Programmable Read-Only Memory</i>

FCS	- <i>Frame Check Sequence</i>
FFD	- <i>Full Function Device</i>
GGA	- <i>Global Position System Fix</i>
GSM	- <i>Global System for Mobile communications</i>
GPRS	- <i>General Packet Radio System</i>
GPS	- <i>Global Positioning System</i>
GTS	- <i>Guaranteed Time Slot</i>
I2C	- <i>Inter-Integrated Circuit</i>
ICSP	- <i>In-circuit serial programming</i>
IMU	- <i>Inertial Measurement Unit</i>
IEEE	- <i>Institute of Electrical and Electronics Engineers</i>
MAC	- <i>The Medium Access Control Layer</i>
MEMS	- <i>Micro Electrical Mechanical Systems</i>
NAM	- <i>National American Datum</i>
NMEA	- <i>National Marine Electronics Association</i>
NWK	- <i>Network Layer</i>
O - QPSK	- <i>Offset Quadrature Phase Shift Keying</i>
PAN	- <i>Personal Area Network</i>
PDA	- <i>Personal Digital Assistant</i>
PHY	- <i>Physical Layer</i>
PWM	- <i>Pulse Width Modulation</i>

RFD	-	<i>Reduced functionality device</i>
SPI	-	<i>Serial Peripheral Interface</i>
TOF	-	<i>Time of Flight</i>
TOW	-	<i>Time of Week</i>
TTF	-	<i>Time To First Fix</i>
TTL	-	<i>Transistor-Transistor Logic</i>
UART	-	<i>Universal Asynchronous Receiver/Transmitter</i>
UGAD	-	<i>Universal Geographical Area Description</i>
UMTS	-	<i>Universal Mobile Telecommunications System</i>
UWB	-	<i>Ultra-Wide Band</i>
WGS	-	<i>World Geodesic System</i>
WIMAX	-	<i>World Wide Interoperability for Microwave Access</i>
WLAN	-	<i>Wireless Local Area Network</i>
WMAN	-	<i>Wireless Metropolitan Area Network</i>
WPAN	-	<i>Wireless Personal Area Network</i>
WRAN	-	<i>Wireless Regional Area Network</i>
WWAN	-	<i>Wireless Wide Area Network</i>
ZCP		<i>ZigBee Compliant Platform</i>
ZDO	-	<i>ZigBee Device Objects</i>
ZUPT	-	<i>Zero velocity Updates</i>

1. INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO

Quando o *Global Positioning System* (GPS), desenvolvido pelo Departamento Norte Americano de Defesa, foi disponibilizado para uso civil, abriu-se um novo horizonte para o desenvolvimento de sistemas de localização. No entanto, é difícil obter uma informação precisa, sendo por vezes até impossível obter qualquer informação, do GPS quando não existe um sinal directo entre o receptor GPS e os satélites [1] [2] [3]. Esta dificuldade pode existir, por exemplo, em locais com densa florestação, em túneis, quando estamos rodeados por edifícios altos ou quando nos encontramos no interior de um edifício.

Foram desenvolvidos alguns sistemas que permitem obter a localização dentro de um edifício, como por exemplo RADAR [4], LANDMARC [5] ou SpotON [6]. No entanto este tipo de sistemas são centralizados, obrigam, uns mais do que outros, a uma pré instalação no local onde se pretende localizar o utilizador. Este facto é uma grande desvantagem se pensarmos em casos, por exemplo, em que uma equipa de bombeiros necessita de entrar num edifício, à partida desconhecido. Aqui é impraticável ser implementado um sistema centralizado para dar suporte à localização da equipa. Além da centralização, outra desvantagem desta abordagem é o facto de não abranger todos os cenários que são hostis ao sinal GPS.

Uma possível solução para esta dificuldade pode passar por implementar o método *Dead Reckoning* (DR) [1] [2] [3]. Este método consiste em somar coordenadas relativas à coordenada absoluta. A coordenada absoluta seria dada, por exemplo, pelo último sinal credível do GPS e as coordenadas relativas seriam dadas por sensores que permitissem obter a

distância percorrida no ambiente hostil ao sinal GPS. Para obter as coordenadas relativas pode recorrer-se, por exemplo, a sensores *Micro Electrical Mechanical Systems* (MEMS), nomeadamente acelerómetros, giroscópios e sensores de pressão [7] [8] [9] [10]. Bússolas digitais também são uma solução [11] [12] [13] quando se quer, além da estimativa da distância percorrida, a orientação.

1.2. OBJECTIVOS

Pretende-se com este trabalho desenvolver um sistema de localização pedestre. Este sistema deverá ser independente de qualquer fonte externa e deverá permitir obter a localização do utilizador em qualquer zona. O sistema deverá ser implementado utilizando *hardware* de baixo custo, de baixo consumo e de fácil utilização.

Dada a complexidade inerente a este objectivo, sentiu-se a necessidade de o subdividir em múltiplas tarefas de realização mais simples tais como:

- Estudo do estado de arte dos sistemas de localização pedestre. Perceber que tipo de tecnologias estão associadas a este tipo de sistemas e a forma como são utilizadas.
- Projectar a arquitectura do sistema. Após ter uma percepção das necessidades associadas aos sistemas de localização pedestre, é necessário idealizar uma arquitectura que permita obter o objectivo principal, o de localizar o utilizador.
- Implementação do sistema. Desenvolver todo o *hardware* e *software* necessários para que o sistema se torne uma realidade.
- Validação do sistema. Submeter o sistema ao um conjunto de testes para permitir saber qual o seu comportamento e que alterações deverão ser eventualmente efectuadas.

1.3. CALENDARIZAÇÃO

Para que os objectivos fossem cumpridos, foi elaborado o planeamento descrito na Tabela 1.

Tabela 1 Planeamento de execução das tarefas

TAREFA	DURAÇÃO	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL	
Estado de arte	2 MESES	█									
Estudo sensores	1 MÊS	█									
Estudo rede ZigBee	2 MESES		█								
Implementação	5 MESES			█							
Testes	2 MESES								█		
Rectificações	1 MÊS								█		
Conclusões	1 MÊS									█	
Relatório	8 MESES		█								

1.4. ORGANIZAÇÃO DO RELATÓRIO

Após este capítulo introdutório, o relatório obedece à seguinte estrutura: no Capítulo 2 é efectuada uma abordagem teórica ao passo humano, sendo analisadas as acelerações que sofre o pé do ser humano enquanto se encontra em movimento, sendo também referidas algumas técnicas que permitem analisar as acelerações sentidas no passo humano, análise que poderá ser útil à qualificação e quantificação do passo. No Capítulo 3 são abordados sistemas de localização, começando desde logo pelo sistema de localização por ventura mais utilizado, o GPS. São abordados alguns detalhes particulares deste sistema. Neste capítulo são ainda descritos alguns sistemas de localização pedestres que recorrem a métodos típicos da técnica DR. No Capítulo 4 são abordados sensores utilizados em sistemas de localização pedestre, em particular o princípio de funcionamento, as vantagens e desvantagem do tipo de sensores que foram utilizados no trabalho. O Capítulo 5 é dedicado à rede utilizada no trabalho, a rede ZigBeeTM. Começa por fazer uma abordagem teórica, situando a rede ZigBeeTM no panorama geral das redes *wireless*. De seguida é efectuada uma descrição da norma sobre a qual se assenta esta rede e das restantes camadas destinadas ao utilizador final que utilizam esta rede, sendo utilizada para o efeito a *stack* da Microchip[©], que foi de resto utilizada neste trabalho. O Capítulo 6 descreve o trabalho desenvolvido, analisando o seu princípio de funcionamento, o seu *hardware*, o seu *software* e discutindo os resultados obtidos. O relatório termina com o Capítulo 7, onde são retiradas as conclusões e projectadas previsões de futuros desenvolvimentos.

2. ANÁLISE DO PASSO HUMANO

No Capítulo 2 é efectuada uma abordagem teórica ao passo humano, sendo analisadas as acelerações que sofre o pé do ser humano enquanto se encontra em movimento, sendo também referidas algumas técnicas que permitem analisar as acelerações sentidas no passo humano, análise que poderá ser útil à qualificação do passo.

2.1. ANÁLISE DAS ACELERAÇÕES QUE OCORREM DURANTE O PASSO

A perna de um ser humano encontrar-se ao longo do tempo, em uma de duas fases: em repouso (pé em contacto com o chão) ou em movimento. Em movimento, a perna do ser humano encontra-se desfasada do centro de gravidade do corpo a maior parte do tempo em que decorre o passo [10]. A Figura 1 retrata essa situação, designando os seus autores de *swing phase* a fase de movimento, em que a perna está, numa primeira fase atrás do centro de gravidade (*1st swing phase*, começa na altura em que a ponta do pé deixa de estar em contacto com o chão) passando de seguida para a frente do centro de gravidade (*2nd swing phase*, culmina esta fase de movimento quando o calcanhar entra em contacto com o chão).

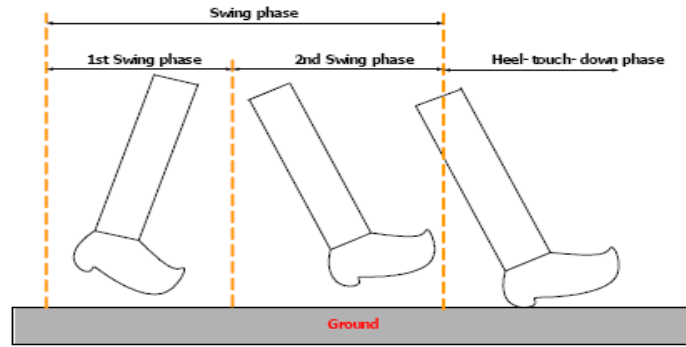


Figura 1 Ciclo do passo humano [10]

Quando a perna está em movimento, a aceleração é composta de acelerações verticais h e horizontais a [10] ilustradas na Figura 2.

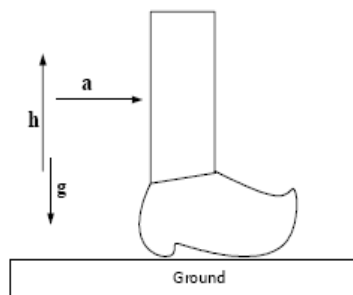


Figura 2 Forças que actuam na perna quando esta se encontra em movimento [10]

Estas acelerações regem-se pelas Fórmulas 1 e 2, onde g expressa a aceleração da gravidade:

$$A(t)_{horizontal} = (h - g) \sin \theta(t) + a \cos \theta(t) \quad (1)$$

$$A(t)_{vertical} = (h - g) \cos \theta(t) - a \sin \theta(t) \quad (2)$$

Ao processar as Fórmulas 1 e 2 para vários instantes de tempo t , assumindo valores de θ e h , podemos obter os padrões das velocidades. As Figuras 3 e 4 demonstram os padrões.

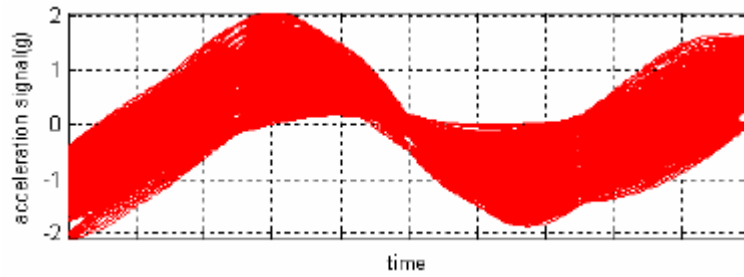


Figura 3 Padrão horizontal da aceleração [10]

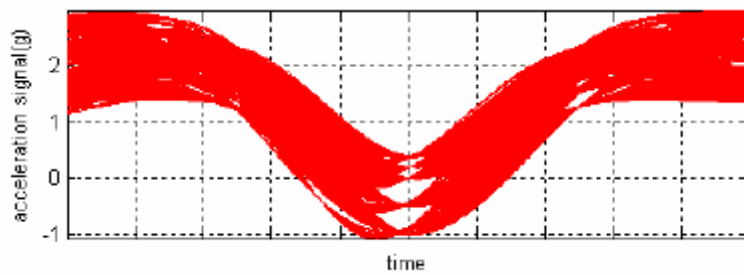


Figura 4 Padrão vertical da aceleração [10]

A seguir à fase em que a perna se encontra em movimento, segue-se o contacto com o chão. Nesta fase, a perna sofre forças repulsivas, como ilustra a Figura 5.

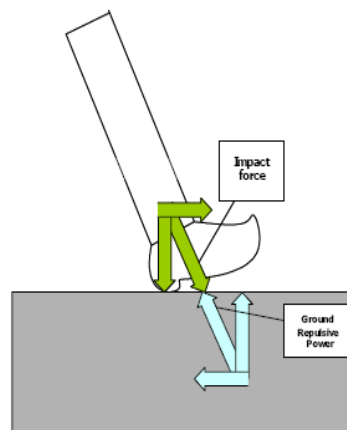


Figura 5 Forças repulsivas que actuam na perna [10]

Combinado a fase repouso com o movimento da perna, obtemos os padrões de referência da aceleração representados na Figura 6 e 7.

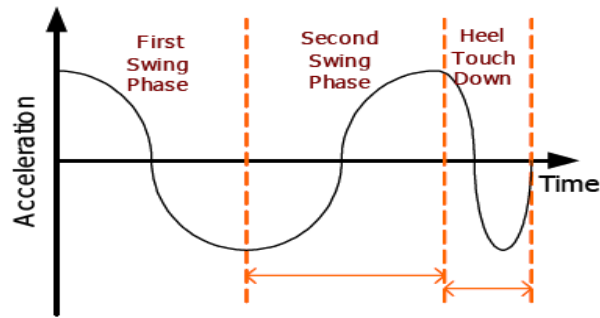


Figura 6 Padrão da aceleração vertical durante um passo [10]

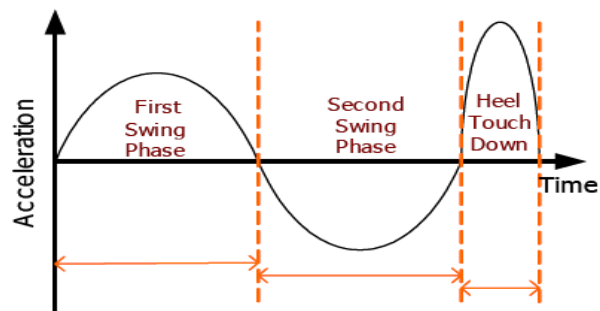


Figura 7 Padrão da aceleração horizontal durante um passo [10]

Estes padrões ilustrados são apenas gerais, podendo ser usados como referência. Como é óbvio o padrão, assim como as suas amplitudes, variam de pessoa para pessoa e de acordo com a velocidade de deslocamento. O padrão dos passos de um idoso não é seguramente igual ao padrão dos passos de um jovem a correr...

2.2. MÉTODOS PARA DETECÇÃO, MEDIÇÃO E ANÁLISE DO PASSO

Como foi referido no Capítulo 1, uma técnica muito utilizada neste tipo de sistemas é a técnica de *dead reckoning* (DR), que consiste na soma de coordenadas relativas a uma coordenada absoluta para obter a distância percorrida. Como obter esta distância? Uma solução simples passaria por estipular um valor constante que reflectisse o tamanho do passo e multiplicar este pelo número de passos contabilizados. O passo poderia ser detectado, por exemplo, através de um algoritmo que lesse os picos de aceleração, considerando os padrões de aceleração representados nas Figuras 6 e 7 [10]. No entanto o tamanho do passo varia de pessoa para pessoa, e o mesmo utilizador poderá apresentar alterações no tamanho do seu passo ao longo do percurso, devido a fadiga ou piso irregular. Torna-se assim necessário obter

o tamanho do passo utilizando um método que reflecta a evolução das acelerações sentidas pelo pé ao longo do percurso.

Para obter este tamanho podemos recorrer a *hardware* adicional, como é sugerido em [15]. Aqui é utilizado o sistema SiLMU, que tenta através da medida *Time of Flight* (TOF) de ultras sons, calcular a distância entre tornozelos. Outro sistema é apresentado em [16] e é composto por um detector de electromagnetismo que tenta obter uma relação entre 2 sensores electromagnéticos, cada um colocado em cada perna do utilizador, para obter o tamanho do passo.

A Analog Devices[©] apresenta um método empírico [17], que recorre à Fórmula 3:

$$tamanho = \sqrt[4]{(A_{m\acute{a}x} - A_{m\acute{i}n})} \times n \times K \quad (3)$$

$A_{m\acute{a}x}$ e $A_{m\acute{i}n}$ são as acelerações máximas e mínimas detectadas num passo, no eixo Z. O número de passos é dado por n e k é a constante para a conversão na unidade pretendida.

Outro tipo de abordagem para obter a distância percorrida é analisando a velocidade horizontal da perna ou pé, quando se encontra em movimento, recorrendo para isso a acelerómetros e giroscópios colocados no pé [18], como indicado na Figura 8. O cálculo da distância passa por saber a velocidade horizontal dos passos, através da integração da aceleração dos mesmos.

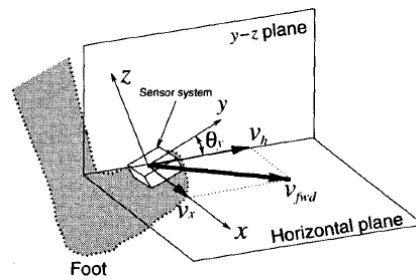


Figura 8 Componentes da velocidade do pé no plano 3D [18]

A velocidade horizontal do pé é dada pela Fórmula 4:

$$v_{fd(t)} = \sqrt{(\{v_{yz}(t)\}^2 + \{v_x(t)\}^2)} \quad (4)$$

em que

$$v_{yz}(t) = \int_0^t (a_{yz}(\tau) d\tau) \quad (5)$$

(6)

$$v_x(t) = \int_0^t (a_x(\tau) d\tau)$$

onde a_x é dado pelo acelerómetro e $a_{yz} = a_y \cos(\theta_y) - a_z \sin(\theta_y)$.

Para minimizar a acumulação de erros característico dos MEMS ao longo do tempo, pode ser utilizada uma técnica chamada *Zero Velocity Updates* (ZUPT) [19]. Numa abordagem simplista, a técnica anula a acumulação de erros, efectuando novas leituras sempre que seja detectado que o pé está em contacto com o chão, em repouso. Deste modo o erro introduzido, não reflecte a soma dos erros das leituras anteriores. Considere o seguinte exemplo ilustrado na Figura 9:

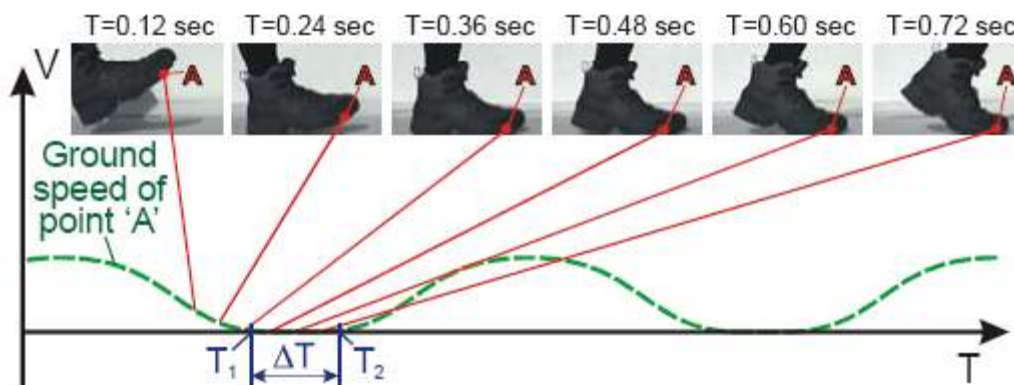


Figura 9 Ponto em que é aplicada a técnica ZUPT [19]

Colocando o MEMS no ponto A, para o intervalo de tempo Δt [0.48;0.72], a aceleração detectada deverá ser zero no eixo X e Y, e próximo de 1 g no eixo Z, considerando que o pé não desliza. Todas as amostras compreendidas no intervalo de tempo Δt deverão ser zero para X e Y. Senão for esse o caso, o valor medido pode ser considerado um erro e usado para eventuais ajustes, por exemplo, utilizado como *offset* para posterior subtração da aceleração medida para esse eixo.

3. SISTEMAS DE LOCALIZAÇÃO PEDESTRE

3.1. INTRODUÇÃO

Neste capítulo são abordados sistemas de localização, começando desde logo pelo sistema de localização por ventura mais utilizado, o GPS. São abordados alguns detalhes particulares deste sistema. Neste Capítulo são ainda descritos alguns sistemas de localização pedestres que recorrem a métodos típicos da técnica DR.

3.2. SISTEMA SATÉLITE *GLOBAL POSITIONING SYSTEM*

3.2.1. INTRODUÇÃO

O *Global Positioning System* (GPS) é composto por uma rede de satélites que tem como objectivo fornecer informação contínua sobre o posicionamento de um determinado objecto. Foi pensado inicialmente para ser um sistema de defesa militar, desenvolvido pelo Departamento de Defesa Norte-Americano. Mais tarde, o seu campo de utilização foi alargado ao uso civil, permitindo surgir assim um conjunto de aplicações nas mais diversas áreas, desde a localização de tráfego terrestre, marítimo ou aeronáutico, sistemas de monitorização na área de engenharia civil (monitorização de edifícios), agricultura...

A rede é composta actualmente por 32 satélites, e é disposta de modo a que existam pelo menos 4 satélites em cada um dos 6 planos orbitais, para garantir uma cobertura contínua a nível mundial [20].

O sistema GPS é dividido em 3 segmentos: espacial, controlo e utilizador [20]. O segmento espacial consiste na rede de satélites responsável, entre outras tarefas, de enviar uma mensagem com a localização dos satélites em função do tempo. O segmento de controlo consiste numa rede espalhada a nível mundial de estações de monitorização. Este segmento tem como principal tarefa monitorizar cada satélite para que seja possível, além de determinar a posição de cada um, controlar a integridade de todo o sistema. Por último, o segmento do utilizador é composto por um conjunto de elementos que permitem receber a informação do segmento espacial e torná-la útil e acessível ao utilizador ou à aplicação final. Os elementos que compõem este segmento são o receptor e respectiva antena de sinais GPS e um *software* que traduza os elementos que compõe a trama de informação, por exemplo, num ponto apresentado no mapa do computador de bordo do nosso carro.

3.2.2. PRINCÍPIO DE FUNCIONAMENTO

Cada satélite transmite um sinal de rádio frequência que é composto por 2 portadoras (duas ondas sinusoidais), moduladas por 2 códigos digitais e a mensagem de navegação. As 2 portadoras são iguais, independentemente do satélite que as transmite. Já os códigos que as modulam variam de satélite para satélite. Estes códigos são vectores binários e são designados de *coarse acquisition (C/A code)* e *precision (P code)* e são gerados através de algoritmos matemáticos. A mensagem de navegação contém, entre outras informações, a posição dos satélites, em função do tempo [20].

Quando um receptor de sinais GPS é ligado, irá começar à procura do sinal de rádio frequência transmitido pelos satélites. Após o recepcionar, o sinal é decodificado de maneira a obter a posição e distância dos satélites, através de um *software* introduzido no receptor.

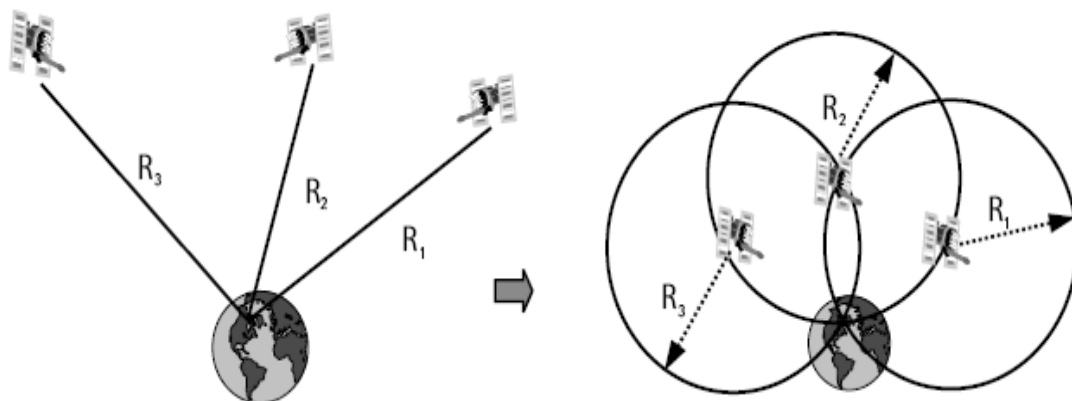


Figura 10 Cenário de localização. No lado direito é visível o ponto onde as 3 circunferências coincidem [20]

Através das distâncias entre o receptor e os satélites é possível determinar a posição do receptor, estará na intercepção das esferas dadas pelos sinais dos satélites com os quais o receptor tem contacto, como demonstra a Figura 10 [20]. O tempo que leva desde a procura de sinal até à obtenção da localização é conhecido como *Time To First Fix* (TTFF).

3.2.3. CAUSAS DE ERROS

Os erros que afectam os sinais GPS podem ter origem em 3 grupos: no satélite, no receptor ou na propagação do sinal. Um satélite pode originar erros devido, por exemplo, ao seu relógio interno, ou devido à sua órbita. Originalmente existia um erro muito grande associado ao satélite chamado selectividade. Tratava-se de um sistema de segurança implementado pelo Departamento de Defesa Norte Americano para impedir que utilizadores não autorizados utilizassem o sistema GPS de uma forma eficiente. No entanto, esta medida foi levantada em 2000, não sendo este factor hoje uma causa de erro. Os erros associados ao receptor envolvem fenómenos como *multipath* (o sinal reflecte em vários objectos, seguindo caminhos diferentes, sendo o sinal principal recepcionado de forma distorcida pelos “novos” sinais originados pelas reflexões), relógio interno do receptor ou ruído provocado pela construção (electrónica) do receptor. Quanto à propagação do sinal, este ao viajar entre o satélite e o receptor tem de passar pela camada ionosférica e troposférica da atmosfera, sofrendo uma refacção [20].

De salientar que a estes factores podemos adicionar ainda a posição dos satélites que fornecem o sinal ao receptor. Quanto mais e melhor distribuídos estiverem, melhor será a obtenção da posição. Observe o seguinte exemplo, ilustrado na Figura 11:

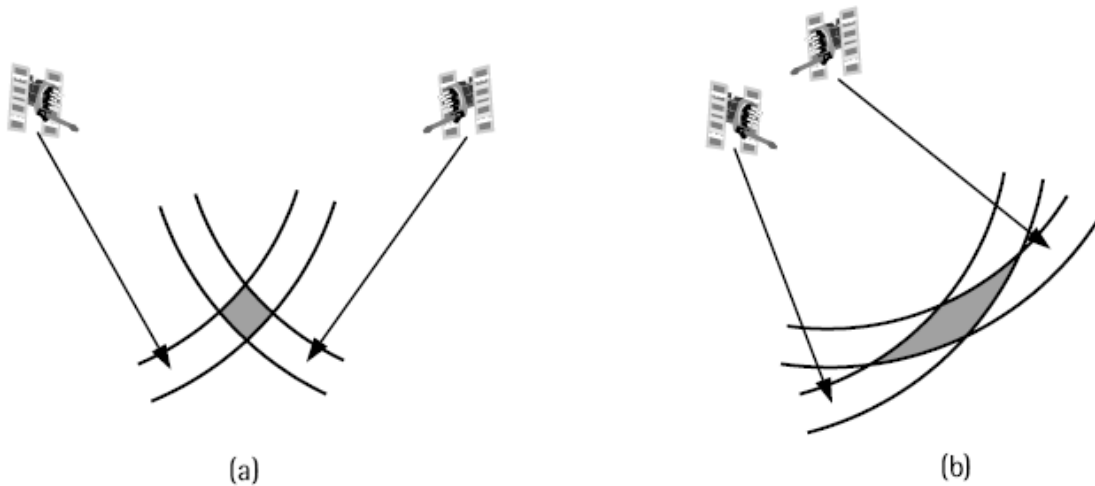


Figura 11 Problemática da distribuição da constelação de satélites e a sua influência na localização [20].

Na Figura 11, o melhor cenário para uma localização mais apurada é o cenário (a). A área de intercepção das esferas dos 2 satélites é menor, logo menor será a área de incerteza associada à localização.

3.2.4. SISTEMA DE COORDENADAS GEODÉSICAS

Um sistema de coordenadas é definido como um conjunto de regras utilizadas para definir as coordenadas de um ponto, sendo normalmente constituído pela origem das coordenadas assim como um conjunto de eixos de referência com uma orientação conhecida [20].

Dada a irregularidade da superfície terrestre, foi necessário obter uma referência para achar as coordenadas geodésicas (ou geográficas) de um determinado ponto. Havia que se encontrar um compromisso entre precisão e simplicidade de cálculos. A melhor referência para aproximar a superfície terrestre encontrada, sem cálculos matemáticos demasiado pesados, foi a elipsóide [20].

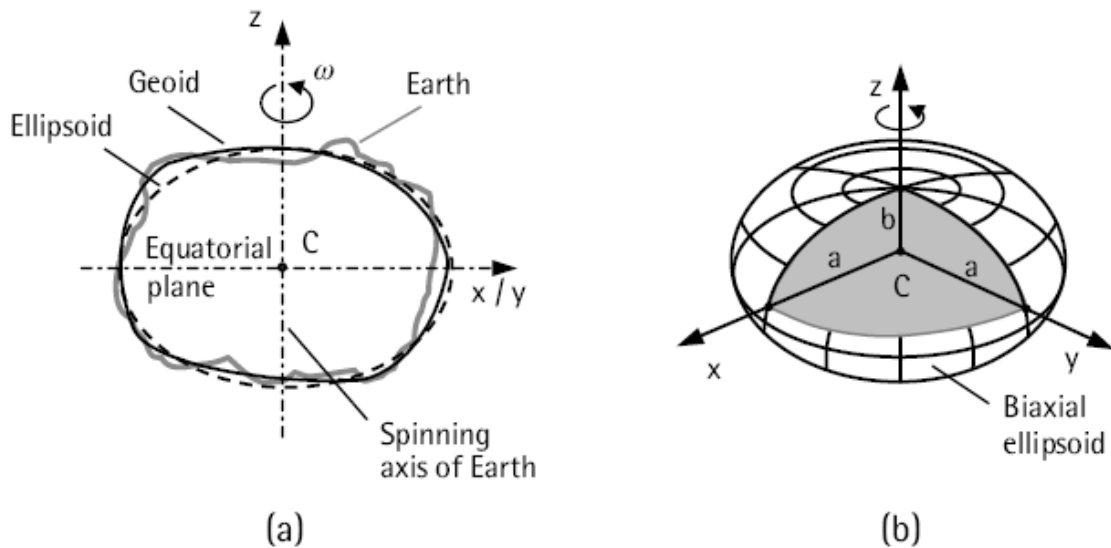


Figura 12 Figura geométrica Elipsóide Biaxial [20].

Esta aproximação da superfície terrestre à fórmula matemática da elipse é conhecida como coordenada geodésica. A Figura 12 a) mostra a geóide, resultado da aproximação da superfície terrestre ao modelo dado pela elipsóide. Uma coordenada geodésica é então uma referência matemática com um centro conhecido e uma orientação. Por exemplo, na Figura 12 b), o ponto C é uma coordenada geodésica geocêntrica, ou seja, situada no centro da terra. Torna-se ainda necessário, para caracterizar a coordenada geodésica, além da aproximação da superfície terrestre à elipse que já foi referida, classificar a orientação dos eixos e da origem. Para isso são utilizados 2 planos, o meridiano que abrange os pólos norte e sul (eixo Z) e o plano equatorial da elipsóide como ilustra Figura 13.

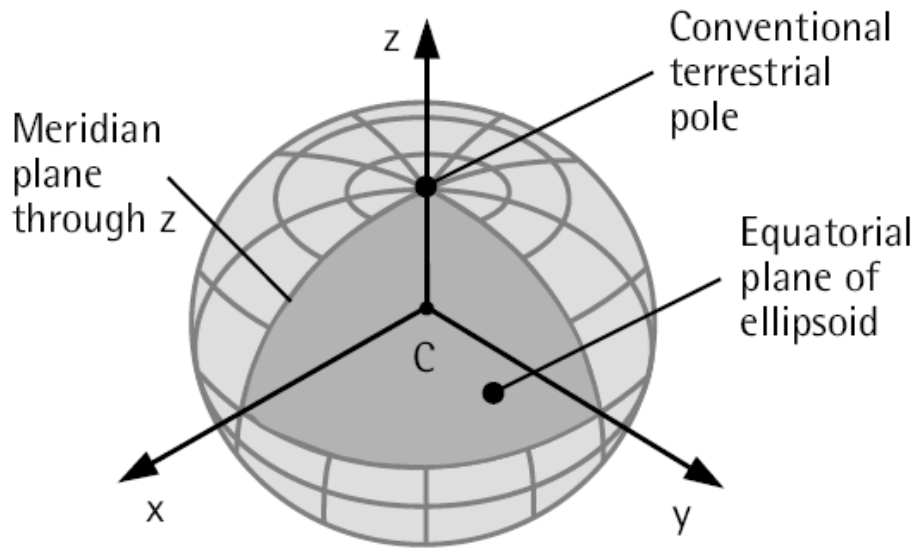


Figura 13 Plano Equatorial e Meridiano [20]

A coordenada geodésica é constituída por 8 parâmetros: 2 parâmetros para definir a dimensão da elipsóide de referência, 3 parâmetros para definir a posição da origem e 3 parâmetros para definir a orientação dos eixos em relação à terra. Num sistema GPS com uma localização de 3 dimensões (3D) e que utilize o sistema de coordenadas geodésicas, os pontos da coordenada são disponibilizados através da latitude (θ), longitude (λ) e a altura acima da superfície de referência (h). Através de processos matemáticos, é possível passar de coordenadas geodésicas para coordenadas cartesianas ou para um sistema de coordenadas de grelha rectangular (utilizada em mapas) [20].

Um exemplo de sistema de coordenadas geodésicas é o *World Geodetic System* de 1984 (WGS 84), que é o sistema oficial de coordenadas do GPS. Este utiliza o sistema *Conventional Terrestrial Reference System* (CTRS), que é um sistema 3D geocêntrico, combinado com uma elipsóide de referência. Outro exemplo de um sistema de coordenadas geodésicas é o *National American Datum* de 1983 (NAM 83), utilizado pela América do Norte para posicionamento espacial [20].

3.2.5. FORMATO NORMALIZADO DE DADOS GPS NMEA 0183

De forma a facilitar a interacção entre vários dispositivos de diferentes marcas, foram normalizados alguns formatos de tramas de dados GPS. Neste ponto abordarei o formato criado pela *National Marine Electronics Association* (NMEA). A razão pela qual foco este

formato em particular prende-se apenas com o facto de ser este o formato utilizado no trabalho realizado.

NMEA 0183 foi adoptado em 1983 como interface entre equipamentos electrónicos na marinha. É constituído por tramas de dados no formato *American Standard Code for Information Interchange* (ASCII), transmitidos a uma taxa de 4800 bits por segundo. Os dados são enviados em formas de mensagens, começadas com o carácter '\$' e terminando com o carácter '<CR> <LF>'. Após o carácter '\$' segue-se o campo de endereço, composto por 5 caracteres, em que os 2 primeiros identificam o emissor e os restantes o tipo de dados e o formato dos sucessivos campos. O último campo em qualquer mensagem é sempre o campo *checksum* de controlo de dados que é antecipado pelo carácter '*'. O tamanho máximo total de caracteres em cada mensagem NMEA 0183 é 82, ou seja, entre o início e fim da trama, podem ser colocados 79 caracteres.

O formato NMEA 0183 dedica alguns tipos de mensagem para o GPS, entre eles o *Global Positioning System Fix data* (GGA).

\$GPGGA,111055.999,4110.7106,N,00836.5536,W,1,03,1.5,-0.9,M,0.0,0000*3A

Figura 14 Exemplo de trama GPS NMEA 0183 do tipo GGA

Na Figura 14 temos um exemplo de uma mensagem GPS utilizando o tipo de trama de dados fixos GGA. Na Tabela 1 é indicado o significado de alguns campos.

Tabela 2 Significado de alguns dados da trama GPS NMEA 0183 do tipo GGA

Símbolo	Significado
\$	Início da trama.
GP	Identificador Emissor. GP indica que emissor é o sistema GPS.
GGA	Identificador do tipo de trama. Dados do tipo fixo neste caso.
,	Delimitador dos sucessivos campos
111055.999	Tempo de registo da posição no formato horas, minutos, segundos, décimo de segundo
4110.7106	Latitude em graus, minutos e décimo de segundo
N	Orientação, Norte ou Sul.
00836.5536	Longitude em graus, minutos e décimo de segundo
W	Orientação, Este ou Oeste

3.2.6. ASSISTED GPS

O *Assisted* GPS (AGPS) consegue ser mais eficiente que o tradicional GPS, pois recorre a uma infraestrutura de comunicações para obter uma previsão da localização. Para que seja possível o AGP funcionar, além do sistema que constitui o GPS convencional é ainda necessário a existência de estações com servidores AGPS e infraestruturas de comunicações, normalmente utilizadas nas telecomunicações.

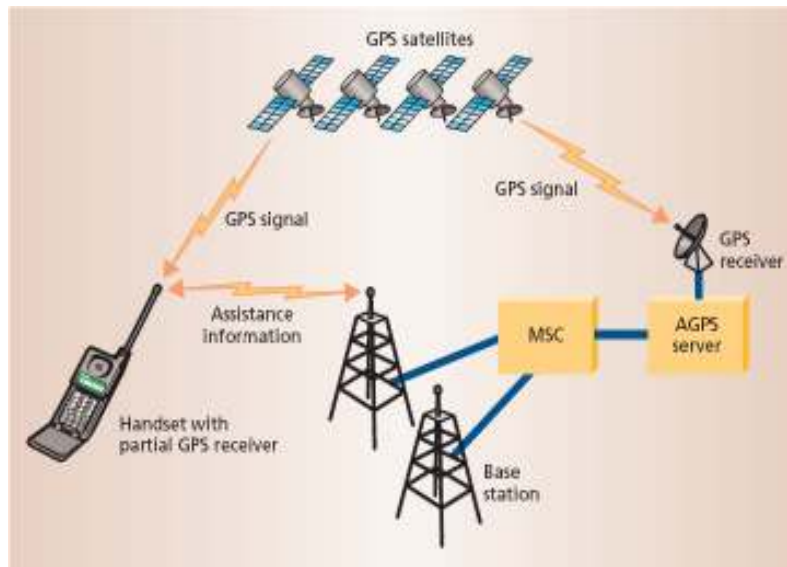


Figura 15 Esquema geral de suporte ao AGPS [21]

Consideremos o exemplo de um *Personal Digital Assistant* (PDA) que contém AGPS, ilustrado na Figura 15. A infraestrutura de comunicações em que o PDA está inserido consegue prever o sinal GPS que este irá receber e consegue, através da rede de telecomunicações, enviar esse sinal de uma forma mais rápida, incurtando o tempo de determinação da posição, TTFF. Este encurtamento também se deve ao facto de não estar relegado apenas para o PDA os cálculos na determinação da posição, estes cálculos são partilhados com o servidor de AGPS [21].

3.3. SISTEMAS DE LOCALIZAÇÃO PEDESTRE DESENVOLVIDOS

Akihiro Hamaguchi *et al* desenvolveu um sistema de localização pedestre recorrendo a sensores colocados no corpo do utilizador [16], ilustrado na Figura 16. A posição do utilizador é estimada tendo em conta o tamanho dos seus passos.

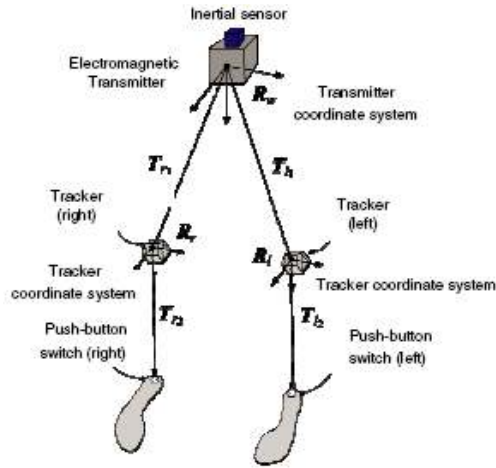


Figura 16 Disposição dos sensores no corpo do utilizador [16]

O sistema é composto por sensores de pressão, que são colocados ao nível do calcanhar do utilizador, um sensor inercial, colocado ao nível da cintura e sensores electromagnéticos, colocados em cada perna, abaixo dos joelhos.

Para determinar a posição, a distância entre os sensores electromagnéticos é adicionada à última posição conhecida. Esta distância está relacionada directamente com o tamanho do passo. Este tamanho é calculado recorrendo à Fórmula 7,

$$p_{heel} = R_w \left((T_{f1} + R_f T_{f2}) - (T_{b1} + R_b T_{b2}) \right) \quad (7)$$

onde p_{heel} é o vector do passo, R_w é a orientação absoluta da anca (dada pelo sensor inercial), T_{f1} e T_{b1} é a posição da perna, R_f e R_b é a pose da perna e T_{f2} e T_{b2} é o tamanho da perna. A distância é então dada pela Fórmula 8:

$$Distância = \sum_{i=1}^n p_{heel_i} \quad (8)$$

Idealmente para este método, ambos os calcanhares deveriam estar no chão ao mesmo tempo para actualização da posição. No entanto, o movimento normal dos passos indica que quando o calcanhar do pé que está a entrar em contacto com o chão, o calcanhar do outro pé já está levantado, como indica a Figura 17.

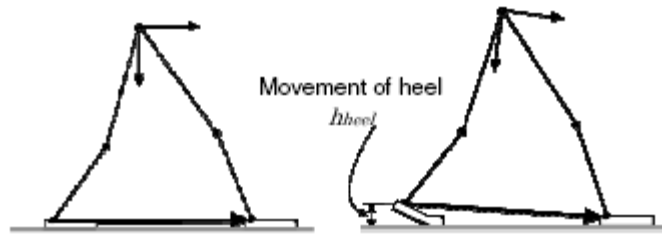


Figura 17 Posição do calcanhar quando o outro pé está totalmente em contacto com o chão [16]

Para minimizar este efeito que iria introduzir um erro acumulativo, foi verificada a altura da posição do calcanhar e a fórmula do vector do passo foi compensada de acordo com a Fórmula 9:

$$p_{heel} = R_w \left((T_{f1} + R_f T_{f2}) - (T_{b1} + R_b T_{b2}) \right) + d_{heel} \quad (9)$$

No entanto se o utilizador estiver a correr, situação em que pode haver instantes em que ambos os calcanhares não estão em contacto com o chão, o sistema falha. O facto de a posição (distância e orientação) ser actualizada quando pelo menos um pé se encontra no chão (o que se traduz neste sistema a um calcanhar em contacto com o seu sensor), significa que sempre que o utilizador estiver a subir escadas, o pé deverá estar totalmente em contacto com o degrau, o que nem sempre acontece, estando apenas a ponta do pé em contacto com o mesmo.

Personal Navigation (PeNa) [15] é um sistema de localização de pedestres em ambientes *indoor*. O sistema é independente de fontes externas, utilizando a técnica de DR para obter a estimativa da localização. É composto por um computador portátil, um sistema de medição do tamanho do passo SiLMU já descrito no ponto 2 do 2º capítulo, um giroscópio de fibra óptica e uma bússola. Na Figura 18 podemos ver a arquitectura deste sistema.

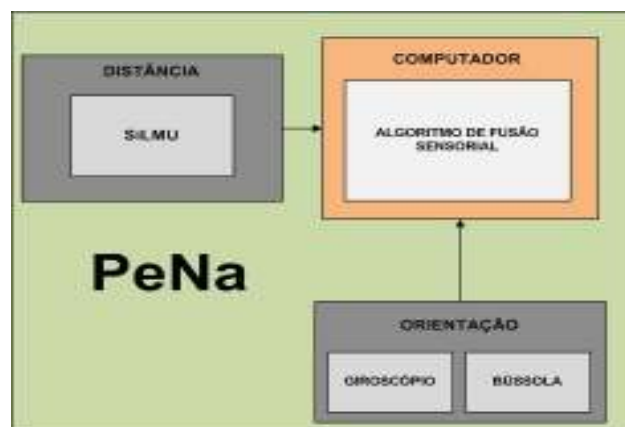


Figura 18 Arquitectura do PeNa

O giroscópio tem como objectivo obter a orientação do utilizador. Como não existe compensação entre eixos para reduzir o erro associado ao balanceamento do sensor causado pelo movimento, é utilizada uma bússola para efeitos de compensação. A estimativa da orientação surge através da fusão entre os dados do giroscópio e os dados da bússola por meio de um filtro *Kalman*.

O sistema SiLMU mede as distâncias entre tornozelos através do tempo de resposta de sinais ultra-sons, sendo colocado o emissor num pé e receptor no outro. O algoritmo, executado no computador e desenvolvido no *software* MATLAB[®] detecta mínimos e máximos, no padrão composto pelos sinais das distâncias entre tornozelos, enviados pelo SiLMU.

A Fórmula 10 permite prever o tamanho do passo é:

$$tamanho = \sqrt{(máximo^2 - mínimo^2)} \quad (10)$$

A Figura 19 demonstra as distâncias utilizadas na Fórmula 10, obtidas através do sistema SiLMU.



Figura 19 Distâncias máxima e mínima calculadas pelo SiLMU [15]

Para testes realizados, percorrendo uma distância de 55m em linha recta, o SiLMU demonstrou ter um erro médio de 0.7m em relação á distância correcta, para um universo de 10 amostras.

Não obstante de ser ainda um sistema em desenvolvimento o PeNa recorre a diverso *hardware* (tem um peso de 14 kg sem contar com o computador), estando este *hardware*

ligado entre si através de cabos. Tudo isto em conjunto não torna o sistema muito confortável para o utilizador. De salientar ainda que o sistema está previsto apenas para cenários *indoor*.

Lauro Ojeda e Johann Borenstein desenvolveram um sistema de localização pedestre (PDR) [19]. O sistema é composto por um conjunto de sensores inerciais, colocados ao nível do pé do utilizador, como ilustra a Figura 20. O sistema fornece a posição linear (orientação e velocidade de deslocamento), assim como altitude, podendo dar deste modo uma localização 3D. Os sensores inerciais estão ligados a um computador através de um cabo de comunicação série RS 232. O computador recebe os dados e corre um algoritmo para efectuar a estimativa da posição. Para eliminar a acumulação de erros o sistema implementa a técnica ZUPT.



Figura 20 Sistema Inercial montado no sapato do utilizador [19]

Para fazer a estimativa da posição em 3D, foi desenvolvido um algoritmo que recorre a equações de representação numérica do tipo *Quaternion*. q é um vector que define a orientação usando 4 parâmetros, a , b , c e d . q propaga-se em função das acelerações angulares w_b de acordo com a Fórmula 11:

$$\dot{q} = \frac{q \cdot p}{2} \quad (11)$$

onde $p = [0, w_b]$ e $w_b = [w_x, w_y, w_z]$. Assim que a orientação esteja calculada, a aceleração do corpo a_b pode ser calculada utilizando as Fórmulas 12 e 13:

$$a_n = q \cdot a_b \cdot q^* \quad (12)$$

$$q^* = (a - b - c - d) \quad (13)$$

q^* é o complexo conjugado de q .

A velocidade de deslocamento pode ser calculada integrando a aceleração, eliminando a componente local de gravidade g_l , utilizando a Fórmula 14:

$$v_n = \int \dot{v}_n dt = \int (a_n + g_l) dt \quad (14)$$

Finalmente pode-se obter a posição integrando v_n , resultando na Fórmula 15:

$$p_n = \int v_n dt \quad (15)$$

O sistema foi validado utilizando dois testes; um 2D e outro 3D. O primeiro teste consistiu em efectuar um percurso que tinha a forma geométrica de um rectângulo, tendo sido percorrido em cada volta uma distância total de 521m, obtendo-se um erro relativo médio de 0.74m quando o percurso era feito no sentido dos ponteiros do relógio e um erro relativo de 0.35m quando o teste era efectuado no sentido contrário. No segundo teste, efectuou-se 1 volta num percurso com uma distância de 324m e, obteve-se um erro 1.1m para a direcção horizontal, tendo sido observado um erro de 4m na direcção vertical.

O *NavShoe* [22] é um sistema de localização pedestre que recorre a uma unidade de sensores inerciais, colocada no sapato do utilizador, para obter a aceleração e direcção de deslocamento quando o sinal GPS não está disponível, sendo o sinal GPS utilizado para a localização em ambientes *outdoor*. Os dados são lidos e armazenados em memória, sendo posteriormente processados por um algoritmo no computador. A sua arquitectura está ilustrada na Figura 21.

É utilizada a técnica ZUPT referenciada no ponto 2 do 2º Capítulo para evitar a acumulação de erros. O *software* do sistema detecta a fase em que o pé está em repouso e aplica a respectiva correcção aplicando as leituras efectuadas nesta fase num filtro digital *Kalman*.

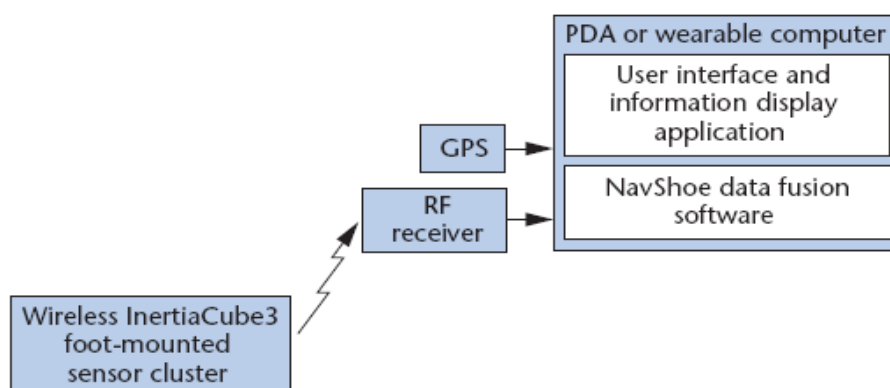


Figura 21 Arquitectura do NavShoe [22]

O sistema inercial é composto por 3 acelerómetros, 3 magnetómetros e 3 giroscópios.

Os resultados de testes *indoor* revelam que, a distância estimada desviou-se 0.3% para uma distância total percorrida de 118.5 metros.

Valérie Renaudin *et al* desenvolveu um sistema híbrido de localização que recorre a MEMS e ao sistema AGPS [2]. O sistema é composto por alguns sensores espalhados ao longo do corpo do utilizador. Estes sensores têm o objectivo, não só de analisar os passos dados pelo utilizador, como também a sua postura. Esta última análise pode ser interessante para monitorização da segurança do utilizador. Por exemplo um bombeiro que esteja a ser monitorizado enquanto desempenha a sua função, se for detectado que a sua postura seja tal como a de deitado, e/ou se a sua postura não se alterar durante um período seguido de tempo, pode ser indício que algo de errado se passa com esse bombeiro. Na Figura 22 pode-se observar o conjunto de sensores que compõe o sistema.



Figura 22 Disposição dos sensores que compõe o sistema no corpo do utilizador [2]

Ao fim de cada passo, a distância percorrida é calculada. Para calcular o tamanho do passo, é utilizada a rotação angular α medida num determinado instante da perna, utilizando a Fórmula 16.

$$s = L \times \sqrt{2 \times (1 - \cos(\alpha))} \quad (16)$$

L é o valor do tamanho da perna do utilizador, resultante da multiplicação do tamanho deste por um factor proveniente de cálculos estatísticos sobre dados fisiológicos. Através do tamanho do passo é ainda possível saber a velocidade horizontal, usando a Fórmula 17:

$$s = v \times \Delta t \times \tan(\gamma) \quad (17)$$

O tempo do ciclo do passo Δt pode tomar valores entre 1 e 2.5s, γ é uma constante que traduz o declive do chão. Este último valor é ignorado (os autores partem do princípio que o chão *indoor* e *outdoor* é geralmente plano) o que resulta na simplificação da Fórmula 17, obtendo a Fórmula 18:

$$s \approx v \times \Delta t \quad (18)$$

O facto de se poder saber a velocidade é uma vantagem, porque permite um cálculo da localização final mais apurado, uma vez que a velocidade do utilizador pode ser diferente em diferentes períodos do tempo, o que dá a capacidade do sistema reagir mais rapidamente e actualizar a localização.

As diferenças na altitude do utilizador são detectadas e é possível saber se o utilizador está a subir ou a descer escadas. A altitude actual, em metros, é dada pela Fórmula 19.

$$h_{k+1} = h_k \pm n \times 0.2 \quad (19)$$

O valor 0.2 vem da suposição dos autores de que cada degrau tem 20 cm de altura e n é o número de degraus detectados pelo algoritmo que analisa os passos do utilizador. O facto de a mudança de altitude do utilizador ser fornecida pela análise dos seus passos e não directamente através dos MEMS, poderá trazer problemas, como por exemplo, se for utilizado um elevador para mudar de piso, a mudança de altitude poder ser vista pelo sistema como um erro e ser descartada.

Os dados fornecidos pelos MEMS e pelas medidas AGPS são recebidos por um algoritmo que utiliza a técnica de filtragem *Kalman* para estimar a localização.

Relativamente às medidas AGPS, foi adoptada a norma, o 3rd *Generation Partnership Project* (3GPP), que prevê especificações para sistemas móveis que suportam os protocolos *Global System for Mobile Communications* (GSM) e *Universal Mobile Telecommunications System* (UMTS). A composição da trama de dados obedece ao formato previsto na especificação *Universal Geographical Area Description* (UGAD).

O formato da trama de dados do sistema AGPS inclui um campo que se refere ao *Time Of Week* (TOW), que indica o tempo do sistema GPS. O sistema de sensores MEMS tem o seu próprio relógio de tempo, pelo que um cuidado a ter quando se utiliza estes 2 sistemas em

conjunto é o de garantir um processo de sincronização entre os mesmos. A Figura 23 demonstra a arquitectura do algoritmo híbrido.

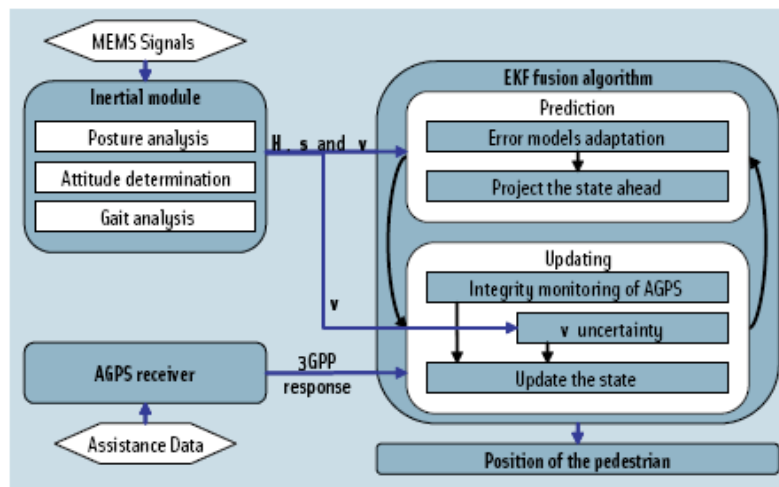


Figura 23 Arquitectura do algoritmo de fusão sensorial [2]

Os dados fornecidos pelos MEMS e pelas medidas AGPS são recebidos por um algoritmo que utiliza a técnica de filtragem *Kalman* para estimar a localização.

O vector da trajectória do utilizador é dada segundo as Fórmulas 20 e 21:

$$\dot{E} = \lambda \times v \times \sin(H + \delta H) \quad (20)$$

$$\dot{N} = \lambda \times v \times \cos(H + \delta H) \quad (21)$$

onde λ corresponde a um factor que influencia as medidas de velocidade horizontal v obtida pelos MEMS. δH diz respeito a perturbações relacionadas com as leituras dos MEMS.

O sinal *indoor* fornecido pelo AGPS sofre influências devido às atenuações impostas, por exemplo das paredes do edifício, ou devido ao fenómeno *multipath*. Isto causa incerteza quanto à integridade do sinal recebido. Para assegurar um bom funcionamento do filtro *Kalman*, foi utilizado um método que passa pela utilização de uma matriz que indica através de cálculos estatísticos (co-variância) o comportamento dos sinais *indoor* AGPS. Assim sempre que o algoritmo híbrido recebe uma medida AGPS vai associar esta a uma determinada incerteza, através da matriz.

As especificidades do caminhar humano podem ajudar a avaliar a integridade das medidas AGPS. Os sensores MEMS gravam as alterações nas cadências registadas nos passos do utilizador. Estas alterações deverão corroborar as variações detectadas nos dados AGPS ao longo do tempo.

Para testar o sistema, os autores fizeram 2 conjuntos de testes. Num dos testes, *outdoor*, o sistema calcula a posição utilizando os dados dos MEMS e AGPS, obtendo-se um desvio médio de 0.54m entre a posição calculada e a posição real. O segundo teste consistia num percurso misto, *indoor* e *outdoor*. A distância total percorrida no segundo teste foi de 370m. Para a parte *outdoor* do percurso misto, os autores referem um desvio médio de 0.99m da trajectória real. Quanto á parte *indoor* deste segundo teste, os autores referem apenas que o desvio aumenta, não referindo o seu valor, indicando também que assim que há condições para o sinal AGPS ser utilizado, o algoritmo de fusão sensorial é capaz de corrigir os erros acumulados.

4. SENSORES PARA SISTEMAS DE LOCALIZAÇÃO PEDESTRE

4.1. MEMS

MEMS são dispositivos compostos por elementos mecânicos e eléctricos, como sensores, actuadores, motores ou válvulas, produzidos à escala microscópica, entre μm e mm , montados numa pastilha de silício, através de processos de fabrico micro electrónico [23] [24]. Estes sensores por vezes são agrupados e constituem dispositivos designados *Inertial Measurement Unit* (IMU). Um dispositivo MEMS não se resume a uma aplicação ou a um sistema, assim como o seu processo de fabrico não é único, podendo um sensor MEMS ser constituído por centenas de elementos ou apenas por alguns. Exemplos de processos de fabrico utilizados para produzir MEMS são *Bulk micromachining*, *Wafet-to-Wafer bonding*, *High aspect radio micromachining* ou *Surface micromachining* [23] [24].

Existe um conjunto de aplicações tão distintas em que estes dispositivos são empregues tais como elementos de segurança passiva (detecção da pressão dos pneus) e activa (sistema *airbag*) em automóveis, utilização em portáteis (sistema protecção do disco quando detectada variação abrupta da velocidade vertical) ou em aplicações de entretenimento, utilizado em

consolas de jogos (colocado nos comandos para produzir um efeito no jogo após uma acção do jogador) e telemóveis (por exemplo rodar o ecrã de acordo com a orientação). No mundo particular dos sistemas de localização, são elementos fundamentais para sistemas que utilizam a técnica de DR como foi verificado no Capítulo 3, oferecendo ao sistema a capacidade de localização independente do meio em que esteja inserido.

4.1.1. ACELERÓMETRO

Acelerómetros são dispositivos que usufruem da tecnologia MEMS para detectar a aceleração a que está sujeito um determinado objecto.

Podemos qualificar os acelerómetros quanto ao princípio de funcionamento, com propriedades *piezo* resistivas, capacitivas, ressonantes ou térmicas, e relativamente aos eixos em que medem a aceleração, 1 eixo, 2 eixos ou nos 3 eixos. Os acelerómetros capacitivos são os que apresentam maior sensibilidade, são menos influenciados pelo ruído e temperatura, apresentam menores derrapagens e tem uma estrutura mais simples [25].

A aceleração detectada pelos MEMS é utilizada para calcular a distância percorrida. Através da sua integração pode ser qualificado o passo [10]. A colocação destes dispositivos pode variar, mas normalmente é colocado ao nível da cintura, pernas [2], podendo ainda ser colocado no pé (sapato) [19] [22]. A colocação do acelerómetro no corpo é muito importante pois as suas medidas vão ser influenciadas pelo ângulo a que este se encontra a cada momento, em relação a uma referência, ângulo este que pode variar ao longo do percurso, devido aos movimentos do corpo.

Um acelerómetro capacitivo é constituído por uma placa com a forma de um pente, colocada sobre a substância de silício, e por âncoras fixas. A disposição das âncoras fixas e da placa móvel é tal que faz com que as hastes com as quais as âncoras e a placa móvel são compostas fiquem paralelas, formando o conjunto de hastes, um condensador.

A substância de silício, também conhecida por *proof mass*, é deformada pelas forças de aceleração, provocando o movimento da placa. A placa ao mover-se vai provocar alteração da distância entre as hastes das âncoras fixas e as suas hastes, o que, conseqüentemente, altera a capacidade do condensador. A Figura 24 demonstra esta situação.

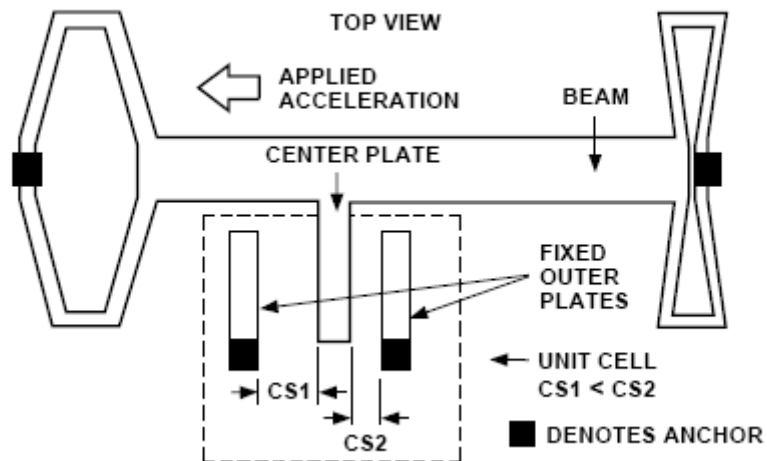


Figura 24 Princípio de funcionamento de um acelerómetro capacitivo [26].

A variação da alteração da capacidade é medida por um circuito que, posteriormente, vai traduzir a tensão medida em aceleração.

A Figura 25 mostra uma fotografia real do conjunto das placas fixas e móveis de um acelerómetro capacitivo.

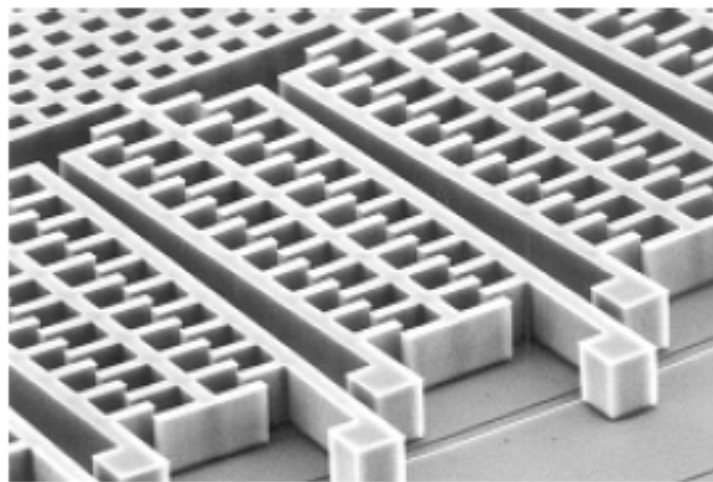


Figura 25 Placas fixas e móveis de um acelerómetro capacitivo [23].

4.2. BÚSSOLA

Uma bússola é utilizada em vários sistemas quando é necessário saber, além da posição, a orientação, ou seja, para onde está direccionado o utilizador.

O campo magnético terrestre tem uma intensidade de 50 a 60 μT . A direcção horizontal do campo magnético terrestre está sempre a apontar para norte, facto que é utilizado para a

determinação da orientação. Para uma medida eficaz, a bússola deverá estar num plano horizontal, caso contrário, a suas medidas são afectadas por erros. Os sensores mais utilizados nas bússolas digitais são baseados no efeito magneto resistivo, no efeito de *fluxgate* ou no efeito magneto indutivo pois são capazes de detectar campos magnéticos fracos, como é o campo magnético da terra. Quanto a sensores de efeito de *Hall*, dado que o campo magnético terrestre é de pouca intensidade, não são muito utilizados devido à sua pouca sensibilidade. Relativamente aos outros tipos de sensores magnéticos, apesar de ter uma maior sensibilidade, requerem alguma energia para o seu uso, não podendo assim ser uma hipótese valida para sistemas de baixa potência e, no caso dos sensores de efeito magneto resistivo, não são totalmente compatíveis com a tecnologia *Complementary Metal-Oxide Semiconductor* (CMOS) [27].

Relativamente à posição, é muito difícil manter a bússola na sua posição ideal devido ao movimento natural do utilizador quando se desloca, por exemplo. Assim deverá existir alguma compensação da sua inclinação. Alguns métodos foram desenvolvidos para determinar esta inclinação, utilizando sensores como giroscópios [15].

5. REDE ZIGBEE™

5.1. INTRODUÇÃO

O *Institute of Electrical and Electronic Engineers* (IEEE) agrupa os protocolos para redes *wireless* em camadas, de acordo com o seu alcance. A Figura 26 distingue essas camadas.

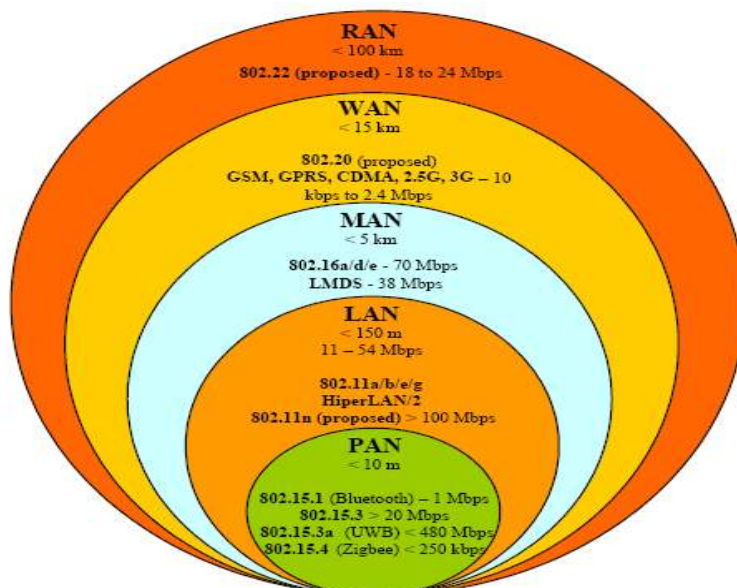


Figura 26 Distribuição dos protocolos *wireless* de acordo com o seu alcance [28].

Na camada *Wireless Wide Area Network* (WWAN) encontram-se protocolos *wireless* de maior alcance, como *Global System for Mobile Communications* (GSM) e *General Packet*

Radio System (GPRS). [29] A norma IEEE 802.22 insere-se nesta camada. Esta norma especifica as camadas física (PHY) e de acesso ao meio (MAC) para utilização em redes *Wireless Regional Area Network* (WRAN) baseadas em *Cognitive Radios* (CRs), aproveitando o espaço não utilizado nos espectros das transmissões de televisão. Tem um alcance típico de 30 km mas poderá chegar aos 100 km, com uma taxa de transmissão de 18 Mbps para um canal com frequência de 6 Mhz [28].

Wireless Metropolitan Area Network (WMAN) é uma camada para redes *wireless* cujo alcance não é tão elevado quanto as redes da WWAN. Esta camada é essencialmente conhecida pela norma IEEE 802.16 *Worldwide Interoperability for Microwave Access* (WiMAX™). WiMAX™ tem um alcance máximo de 50 km e uma taxa de transmissão na ordem dos 100 Mbps. Pode ser utilizado como *backbone* para ligar redes previstas na norma 802.11 [30].

No parágrafo anterior referi a norma 802.11, em que se baseia o protocolo Wi-Fi®. Este insere-se na camada *Wireless Local Area Network* (WLAN). Wi-Fi™ permite, uma taxa de transmissão superior a 100 Mbps que poderão chegar aos 600 Mbps (802.11n) [31] [32], com um alcance máximo de 450 m [33]. É normalmente utilizada para ligar vários computadores numa rede local *wireless*, mas já é possível encontrar aplicações deste protocolo em MP3 ou telemóveis.

A última camada é a *Wireless Personal Area Network* (WPAN). Esta camada agrupa os protocolos de curta distância, como Bluetooth™, ZigBee™ e UWB™, pelo que esta camada é utilizada em aplicações com médio débito na transferência de dados como transferência de dados entre dispositivos *Personal Digital Assistant* (PDA), troca de ficheiros entre computadores ou em aplicações com baixo consumo e baixa transferência de dados, como uma rede de sensores [34].

5.2. NORMA 802.15.4

O IEEE divide, na sua norma 802.11, as redes de comunicações sem fios em algumas categorias, entre elas a *Wireless Personal Area Network* (WPAN), onde estão inseridos os padrões de rede sem fios de pequeno alcance. Existem vários padrões e tecnologias que suportam este tipo de redes, como a norma IEEE 802.15.1 conhecida como Bluetooth™, a norma IEEE 802.11 WI-FI™ ou a norma IEEE 802.15.4 sobre a qual foi desenvolvido o protocolo ZigBee™.

Nesta norma estão previstas as características da camada de controlo de acesso ao meio (MAC) e a camada física (PHY), para darem suporte a redes de baixo consumo [35]. As restantes camadas, como a camada da aplicação, é da responsabilidade do utilizador final que desenvolve um produto para este tipo de rede, como mostra a Figura 27.

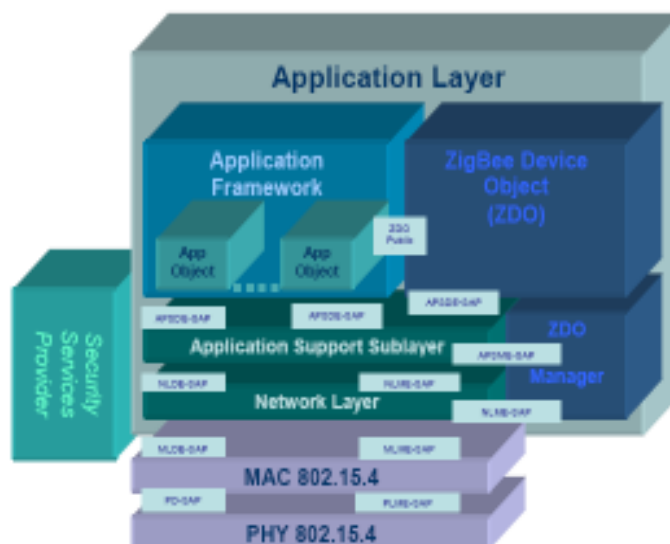


Figura 27 Arquitectura ZigBee™ [36].

As características desta norma possibilitam criar tipologias de rede, como estrela, ou do tipo *peer to peer*. A norma prevê ainda 2 tipos de equipamentos, *Reduced Function Devices* (RFD) e *Fully Function Devices* (FFD). Estes últimos têm a capacidade de comunicar com qualquer dispositivo presente na rede, enquanto os RFD's conseguem apenas comunicar com os FFD's. Numa rede terá que existir, pelo menos um FFD, designado como o coordenador dessa rede, podendo a restante rede ser constituída por mais FFD's, apenas por RFD's ou uma mistura dos dois tipos de dispositivos.

5.3. CAMADA FÍSICA DA NORMA 802.15.4

Esta camada refere como é que os dispositivos podem comunicar entre si através de um canal. Fornece funcionalidades para a escolha do canal de comunicação, envio e recepção de dados, detecção de energia no canal, previsão da qualidade do canal de transmissão.

Existem 3 frequências que podem ser utilizadas, sendo a capacidade de transmissão directamente influenciada pela escolha da frequência. Como opções temos a frequência de 2450 MHz (250 Kb/s – 16 canais) com cobertura a nível mundial, 914 MHz (40 Kb/s – 10

canais) com cobertura na América do Norte e por último, na Europa, a frequência de 869 MHz (20 Kb/s – 1 canal) [35].

A modulação do sinal também depende da frequência. Na frequência de 2.45 GHz, o sinal é modulado recorrendo ao método *Offset Quadrature Phase Shift Keying* (O - QPSK). Para as frequências de 914 MHz e 868 MHz a modulação é do tipo *Binary Phase Shift Keying* (BPSK) [35].

Todas as redes utilizam o mesmo modo de acesso ao meio, *Direct Sequence Spread Spectrum* (DSSS).

5.4. CAMADA MAC DA NORMA 802.15.4

Esta camada especifica quando os dispositivos podem aceder ao meio. Fornece funcionalidades que permite a geração de *beacons*, sincronização, formação de *Personal Areas Network* (PAN), controlar o acesso ao meio através do método *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA), fornecer e manter o *Guaranteed Time Slot* (GTS) e validar as mensagens trocadas entre dispositivos. Estas funcionalidades estão presentes nos equipamentos do tipo FFD's, como os coordenadores da PAN [37].

As redes podem ser do tipo *Beacon* ou *Non Beacon*, sendo esta a característica que permite ao ZigBee™, usufruir de uma das suas vantagens, a poupança de energia dos dispositivos na rede. Quando a rede é do tipo *Non Beacon*, os seus dispositivos estão constantemente ligados, podendo deste modo comunicar entre eles a qualquer altura. Aqui a sincronização deverá ser providenciada por níveis superiores da camada. Se a rede for do tipo *Beacon*, o coordenador da rede envia periodicamente *beacons*, que os dispositivos utilizam para sincronização e para saberem quando podem transmitir e receber mensagens. O envio de um *beacon* determina o início do envio da trama MAC, com a qual todos os dispositivos da rede devem estar sincronizados.

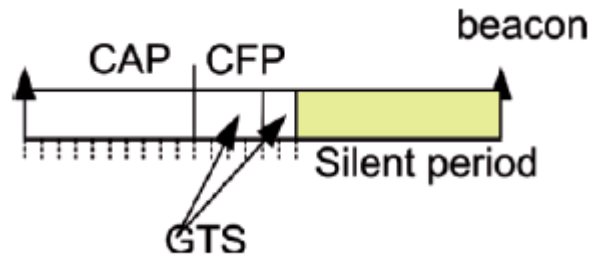


Figura 28 Composição da trama MAC [38]

Na Figura 28 vemos a composição da trama MAC, trama esta totalmente configurável. A observação desta figura permite distinguir claramente o período activo, utilizado para troca de informação, e o período inactivo, em que os dispositivos estão em modo de poupança de energia, não existindo assim troca de informação. A parte dedicada à troca de informação é dividida em 16 partes de igual tamanho, existindo 2 períodos dominantes, *Contention Access Period* (CAP) e *Contention Free Period* (CFP), que por sua vez está subdividido em períodos GTS.

O período CAP serve para os dispositivos acederem ao meio, utilizando CSMA-CA. Este período deverá ser constituído por 9 partes, mas pode ser constituído pela totalidade das 16. No período CFP, o dispositivo tem acesso exclusivo ao canal e troca informação com o coordenador, sendo esta comunicação *half-duplex*. Este período está localizado no fim da parte activa da trama, e pode ter no máximo 7 das 16 partes que compõe a parte activa da trama. Se o coordenador quiser enviar uma mensagem para um dispositivo, guarda essa mensagem e indica, quando envia o *beacon* que tem uma mensagem para enviar. O dispositivo ao receber o *beacon*, no período CAP, pede a mensagem ao coordenador.

O tamanho máximo da trama MAC é de 127 bytes, dos quais 16 bits são designados para controlo da composição da mesma, utilizando o método *Cyclic Redundancy Check* (CRC). Para controlo da comunicação e por forma de saber se as tramas são ou não entregues, esta camada prevê um mecanismo de transferência de tramas que utiliza *acknowledgements*, ou seja confirmação das tramas recebidas por parte do dispositivo receptor. Este tipo de confirmação é opcional entre transferências do receptor para o seu coordenador, mas são obrigatórias em transferências do coordenador para o receptor. A Figura 29 ilustra esta situação.

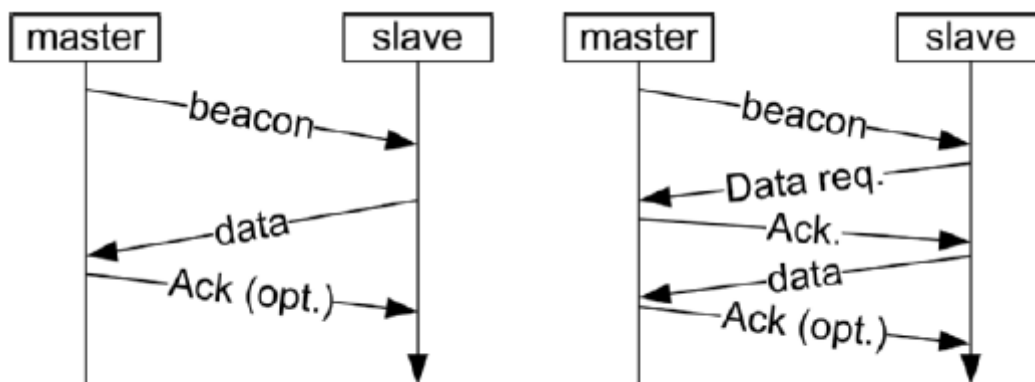


Figura 29 Troca de informação entre coordenador e dispositivo da rede [38].

Para uma transmissão do tipo *non beacon*, a sequência nos 2 casos era a mesma mas, claro, sem o envio do *beacon* [37].

As camadas abordadas a seguir já não fazem parte da norma 802.15.4, sendo da responsabilidade da entidade que desenvolve a aplicação. Nesta descrição podem ser utilizados termos associados à *stack* criada pela Microchip[©] (membro da ZigBee[™] Alliance), para ZigBee[™]. Foi esta a *stack* utilizada no trabalho.

5.5. CAMADA DE REDE NWK

Esta camada é responsável por organizar e encaminhar as tramas pelos vários pontos da rede até ao seu destinatário.

Numa rede ZigBee[™], quanto ao seu papel na rede, poderão existir 3 tipos de dispositivos, os designados *end device*, normalmente RFD (mas também poderão ser FFD), *router*, obrigatoriamente FFD, e o coordenador, um por rede, também obrigatoriamente FFD, responsável por toda a rede. Normalmente os *end device* são, por exemplo módulos ligados aos sensores que se controlam na rede. A utilização de *routers* nas redes prende-se pelo facto de muitas vezes a estrutura da rede não ser tão básica como a tipologia estrela ou ser necessário estender o seu alcance. Já foi dito que a camada MAC suporta tipologias como estrela ou *peer to peer*, pelo que caso se pretendam outras tipologias que não estas, são necessários *routers* para ligar os vários ramos ao coordenador da rede.

A camada de rede do ZigBee[™], é mais abrangente quanto ao tipo de configuração possível numa rede. Para além da tipologia estrela, podemos ter redes em Malha (conhecidas pelo

seu termo anglo saxónico *mesh*) ou redes do tipo *Árvore* (termo anglo saxónico *cluster tree*).

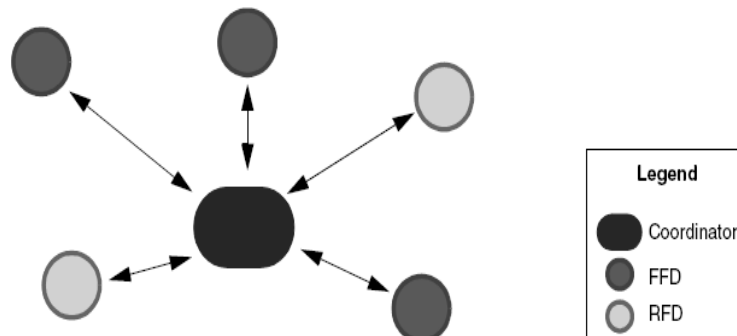


Figura 30 Tipologia Estrela [39]

Na tipologia do tipo Estrela, exemplificada na Figura 30, existe um coordenador e um ou mais *end devices*. Estes últimos apenas podem comunicar com o coordenador. Se existir a necessidade de um *end device* comunicar com outro, a informação passará em primeiro lugar pelo coordenador, que por sua vez, a entregará ao seu destinatário.

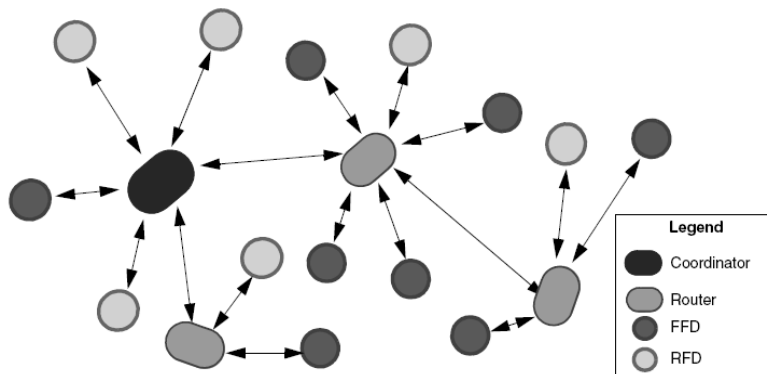


Figura 31 Tipologia do tipo Árvore [39]

A tipologia em *Árvore* e em *Malha* são compostas, para além do coordenador e dos *end devices*, por *routers*. Estas tipologias são mais adequadas quando existe a necessidade de alargar o alcance da rede ou aumentar o número de nós da mesma, uma vez que apenas pode existir um coordenador na rede, ficando assim cada *router* responsável por gerir o seu nó, comunicando este com o coordenador sempre que é necessário um dispositivo do seu nó comunicar com o coordenador ou enviar uma mensagem para outro nó. A Figura 31 ilustra uma tipologia do tipo *Árvore*.

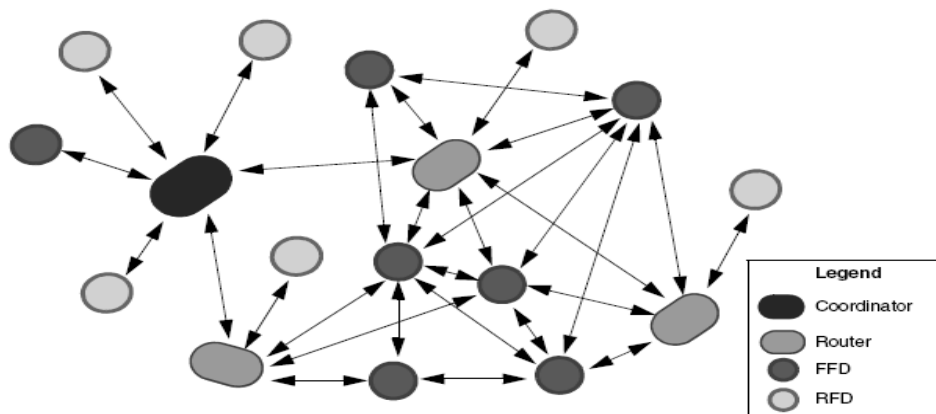


Figura 32 Tipologia do tipo Malha [39]

A diferença entre a tipologia *Árvore* e a tipologia em *Malha*, exemplificada na Figura 32, é que nesta última, os dispositivos do tipo *FFD* podem trocar mensagens directamente entre si. Como em todas outras tipologias, dispositivos *RFD* continuam sem poder comunicar directamente, sendo necessário a informação passar pelo *router* (e eventualmente coordenador) na tipologia *Árvore* ou *Malha*, ou pelo coordenador, na tipologia *Estrela*, antes de chegar ao destinatário.

Quando um dispositivo pretende associar-se a uma rede já existente, a camada de rede inicia um procedimento de busca de redes que se encontram ao alcance do dispositivo. Após tomar conhecimento das redes vizinhas, e após as camadas superiores terem decidido qual a rede a que se deve juntar, selecciona o seu parente (coordenador ou *router*) com o qual vai estabelecer as trocas de informação, iniciando de seguida a camada *MAC* um procedimento de ligação à rede.

5.6. CAMADA DE APLICAÇÃO APL

Esta camada faz a gestão e dá suporte às aplicações presentes na rede ZigBee™. Contém a *Application Framework*, para suporte ao desenvolvimento de aplicações, *ZigBee Device Objects* (*ZDO*) e *Application Support Sub Layer* (*APS*).

Os dispositivos inseridos numa rede ZigBee™ obedecem a um determinado *Profile*. Um *Profile* descreve o ambiente da aplicação, ou seja, contém um conjunto pré definido de mensagens, contém os procedimentos e interfaces de um determinado dispositivo, define também o formato dos atributos. Atributos são blocos de dados trocados entre dispositivos através de comandos. Um conjunto de atributos designa-se *cluster*. Os dispositivos são assim definidos pelos seus *Profiles* e implementados na rede como *Application Objects*

(APO). A cada APO é atribuído um único número *endpoint* que o liga à restante *stack* ZigBee™. Um *endpoint* suporta um ou mais *clusters* e é-lhe atribuído um endereço (entre 1 a 240) que é adicionado ao endereço da rede de 16 bits e ao endereço MAC de 64 bits. Deste modo, as APO's podem comunicar através dos seus *endpoints*, utilizando *clusters*.

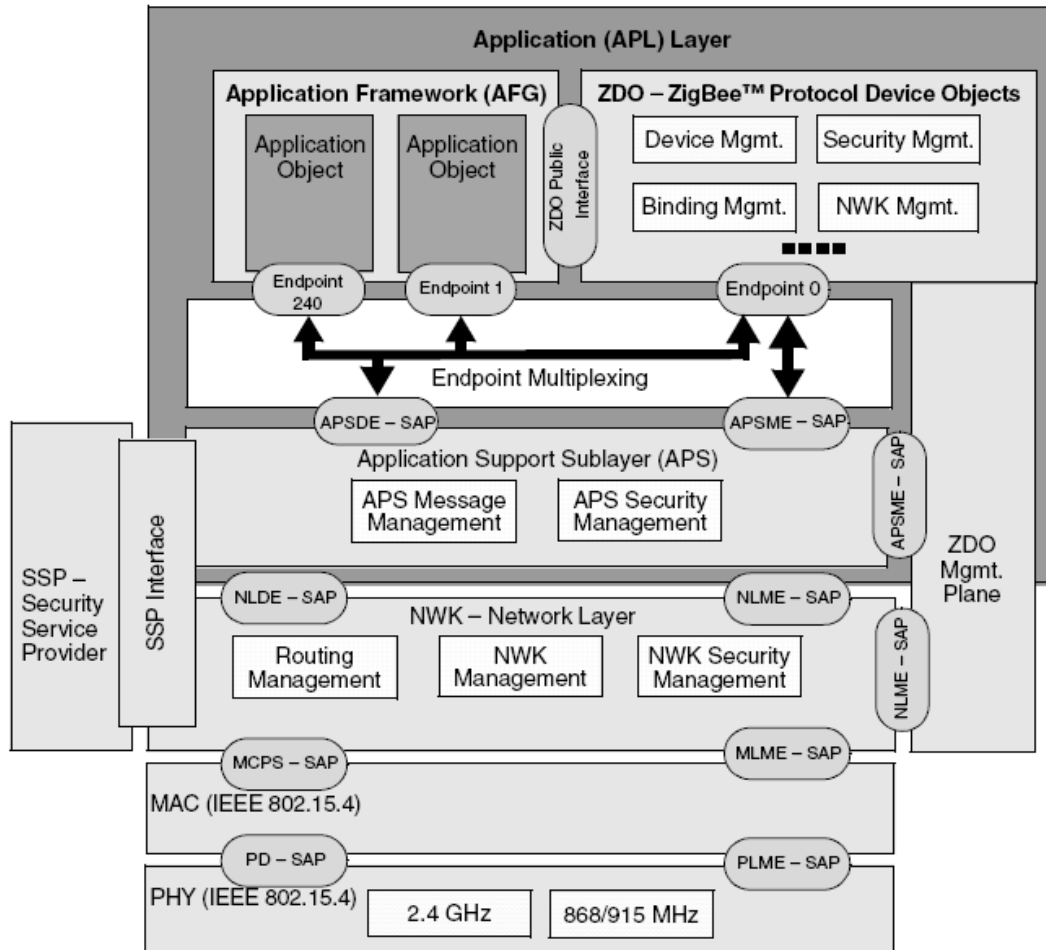


Figura 33 Arquitectura da *stack* da Microchip® para ZigBee™ [39]

O *endpoint* 0 está atribuído ao ZDO. Através do ZDO, a camada de aplicação pode comunicar com as outras camadas, podendo assim proceder à sua inicialização e configuração. Para que tal seja possível é utilizada a APS, que liga os *endpoints* à camada de rede. O ZDO fornece também aos *endpoints* o processo designado de “*binding*”. *Binding* é a capacidade de relacionar 2 dispositivos diferentes mas compatíveis (por exemplo interruptor e lâmpada). A Figura 33 exemplifica a *stack* utilizada no trabalho, e a Figura 34 mostra um exemplo de uso de um *Profile*, diferenciando as várias camadas que o compõe, e mostrando como elas interagem entre si.

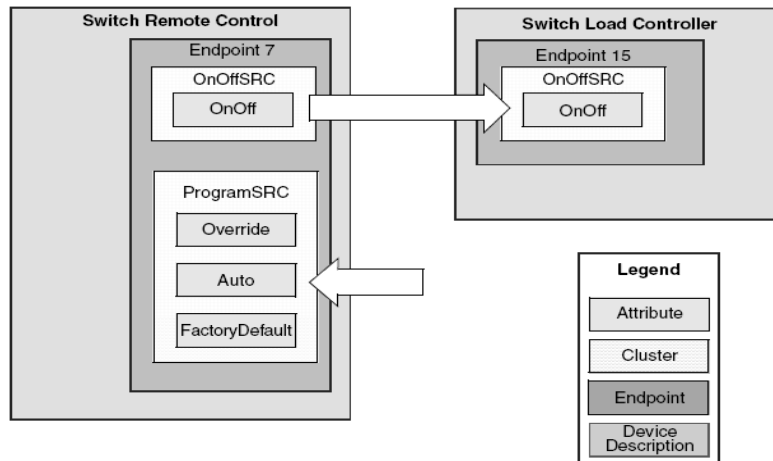


Figura 34 Exemplo de interação entre dispositivos usando um *Profile* [39]

O *Profile* define a identificação dos atributos e dos *clusters*, assim como o formato de cada atributo. Por exemplo, no *Profile Lighting* usado para controlo doméstico, o *cluster* OnOffDRC do dispositivo *Dimmer Remote Control* (DRC) contém um atributo, OnOff, que tem de ser do tipo *unsigned* de 8 bits, com o valor 0xFF significando “On” e, 0x00 significando “Off” e 0xF0 com significado “*toggled output*”. O *Profile* indica também que *clusters* são obrigatórios e os que são opcionais para cada dispositivo.

O *Profile* pode assim, uma vez definido, ser utilizado em qualquer aplicação, permitindo dispositivos de diferentes marcas entenderem-se numa aplicação.

5.7. ESTRUTURA DAS TRAMAS DO ZIGBEE™

A norma 802.15.4 prevê 4 tipos de tramas: de dados, de confirmação (*ack*), comando MAC e *beacon*.

Todas estas tramas, na sua estrutura, têm em comum o facto de começarem com um cabeçalho de sincronização, com 5 *bytes* de comprimento, sendo seguido pelo cabeçalho da camada física PHY, com 1 *byte* de comprimento.

Após estes 2 cabeçalhos, a estrutura do cabeçalho MAC vai variar, dependendo do tipo de trama. Apesar da estrutura do cabeçalho MAC variar devido aos 4 diferentes tipos de trama possíveis, o seu tamanho é sempre 127 *bytes*, o que somando com os 6 *bytes* dos outros 2 campos da camada física, resulta num tamanho total de 133 *bytes* da trama do protocolo ZigBee™.

Quanto à estrutura da trama de dados (*Data frame*), é composta por um campo *data payload* de 102 bytes de tamanho máximo, que é preenchido pela sub camada APS. Esta trama é numerada para garantir a sequência correcta de envio, existindo para este efeito um campo designado de *Sequence Number*, cujo tamanho é de 1 byte. Existe ainda um campo designado de *Frame Check*, com comprimento de 2 bytes para garantir que as tramas são recebidas sem erros, um campo com informações sobre endereço do emissor/receptor, cujo tamanho máximo é de 20 bytes. A informação neste campo é preenchida pela camada de rede NWK. Existe ainda o campo *Frame Control* com 2 bytes.

A trama do tipo *acknowledgment (ack)* é utilizada pelo protocolo sempre que é utilizada confirmação pelo dispositivo. É composta pelos campos *Frame Check*, *Sequence Number* e *Frame Control*.

Relativamente à trama *MAC command*, é utilizada para controlo e configuração de nós remotos. A sua estrutura é igual à trama de dados, diferenciando-se desta pelo facto de não ter um campo *APS payload*, mas sim um campo *Command type*. Quanto ao tamanhos dos campos, são os mesmos que a trama de dados.

Finalmente, o tipo de trama *beacon* que é utilizada para sincronismo pelos nós quando saem do modo de poupança de energia. A Figura 35 ilustra os 4 tipos de mensagens.

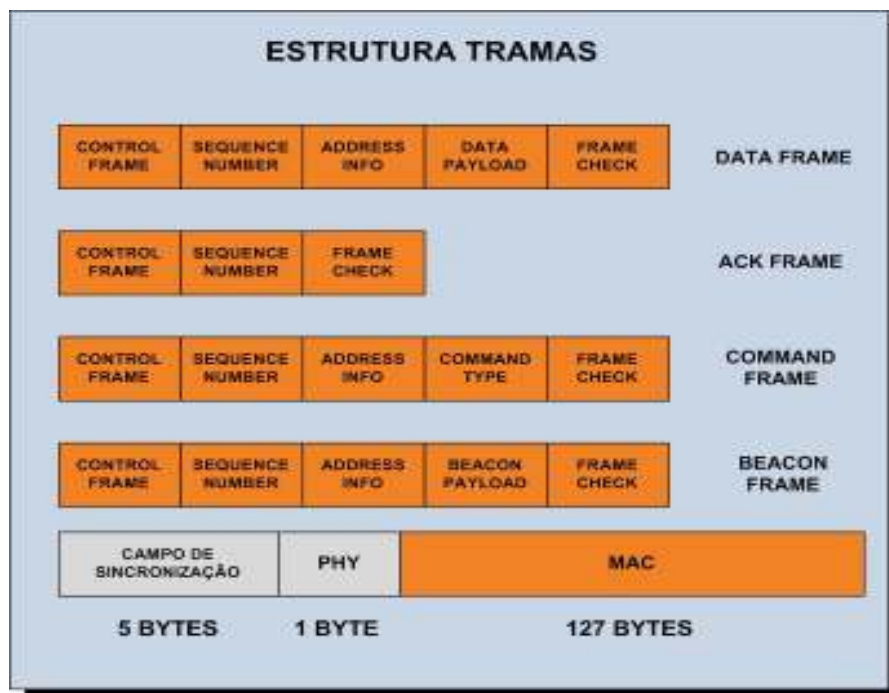


Figura 35 Estrutura das tramas do protocolo ZigBee™

6. INPERLYS

Este Capítulo é dedicado à descrição do sistema desenvolvido. Começa-se por fazer uma descrição geral do sistema e da sua arquitectura. O *hardware* utilizado é descrito, assim como o *software* desenvolvido, dando especial relevo às rotinas que assumem papéis centrais no sistema. O Capítulo 6 encerra com a análise dos resultados obtidos nos testes efectuados ao sistema.

6.1. DESCRIÇÃO DO SISTEMA

Em zonas *outdoor*, onde existe uma linha de contacto directo com os satélites GPS, é relativamente fácil obter a posição do pedestre, com boa fiabilidade e continuidade. No entanto, o mesmo não se pode dizer quando o pedestre se encontra em locais que cujas características são altamente degradantes e prejudiciais ao sinal GPS, como florestas com densa vegetação, zona rodeada de edifícios altos ou mesmo dentro de edifícios (zonas *indoor*). INPERLYS [40] (*INdependent PERSONal Location sYstem*) é um sistema de localização pedestre capaz de monitorizar os movimentos de um pedestre em qualquer sítio.

INPERLYS é um sistema baseado na integração de sinais fornecidos pelo *Assisted GPS* (A-GPS) e dados fornecidos por um sensor MEMS (acelerómetro) e uma bússola digital. Com esta integração será possível utilizar a técnica *Pedestrian Dead Reckoning* (PDR),

onde a posição absoluta é fornecida pelo A-GPS, à qual serão somadas as posições relativas fornecidas pelos sensores. O A-GPS utiliza a disseminação de dados efectuada por uma rede de telecomunicações para melhorar os dados recebidos.

Quando não se verifica a possibilidade de se obter o sinal A-GPS, serão utilizados os dados fornecidos pelos sensores acima indicados, que estarão colocados nas roupas do utilizador. Estes sensores têm a capacidade de poder fornecer uma estimativa da posição de uma forma totalmente independente de fontes externas. É visível na Figura 36 os módulos de sensores responsáveis pelo fornecimento de coordenadas relativas, assim como o módulo GPS, responsável pelas coordenadas absolutas.

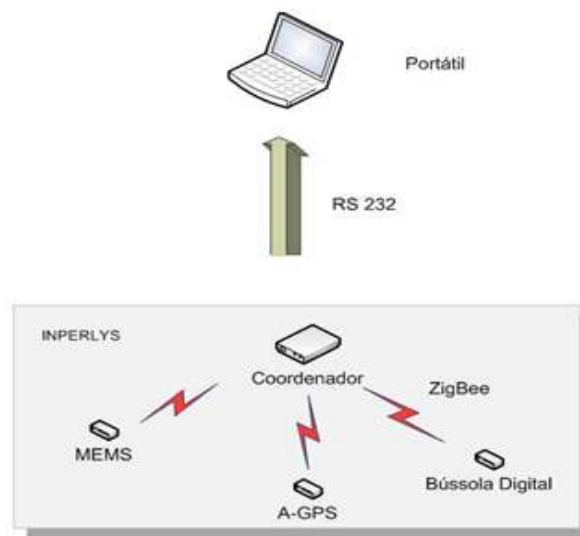


Figura 36 Sistema INPERLYS

Todos estes elementos estão ligados a uma rede *wireless* ZigBee™, para uso mais confortável do sistema. A rede é composta por um módulo ZigBee™ FFD coordenador e 3 módulos RFD. O coordenador, além dos serviços associados à criação e gestão da rede, tem o objectivo de receber os valores fornecidos pelos 3 módulos e fornecê-los, através de uma ligação RS-232, ao computador que os exhibe no *Hyperterminal*.

6.2. ARQUITECTURA

A arquitectura do INPERLYS pode ser dividida em 3 áreas: área sensorial de recolha de dados, área de gestão e tratamento de informação e área aplicacional.

A área sensorial tem como objectivo principal a recolha da informação que irá traduzir a monitorização do pedestre, quer seja através dos dados fornecidos através do A-GPS, quer seja através das acelerações medidas pelo MEMS e orientação fornecida pela bússola digital. Podemos subdividir esta área em 2 novas áreas; o módulo que é composto pelo A-GPS (monitorização *outdoor*) que será utilizado sempre que seja possível, ou seja, sempre que a localização do pedestre permita a recepção de informação credível por parte do GPS e os módulos compostos pelo MEMS (análise do passo) e pela bússola digital (orientação do movimento), que serão a alternativa ao A-GPS.

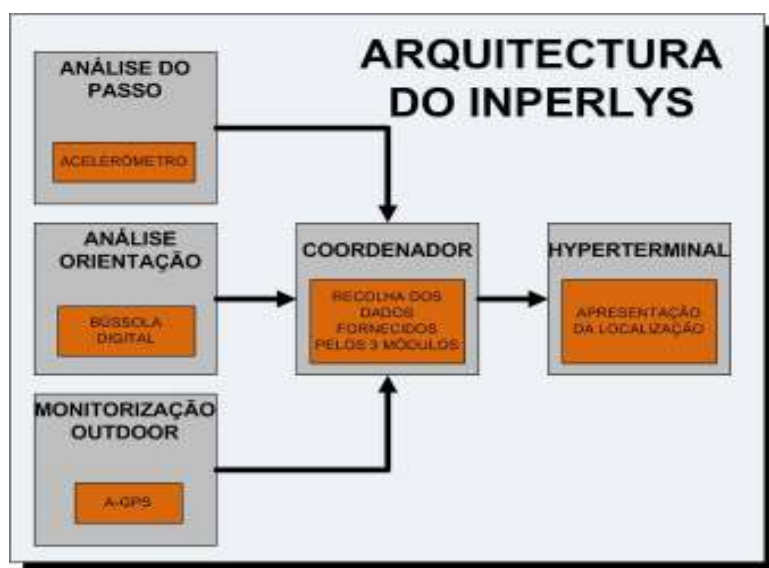


Figura 37 Arquitectura do sistema INPERLYS

A área de gestão e tratamento de informação é composta pelo módulo coordenador da rede ZigBee™. Esta área tem como objectivo, numa primeira fase, criar condições para que todos os módulos estejam ligados entre si, criando para isso uma PAN, sob a qual terá o poder de gestão inerentes à rede ZigBee™. Além da rede, este módulo faz também a recolha dos dados provenientes da área sensorial, preparando-os para fornecer à área aplicacional. Por último, o coordenador é também responsável por garantir a sincronia dos dados dos diferentes módulos.

Quanto à área aplicacional tem como principal objectivo, após receber os dados da área de tratamento e gestão de informação, apresentar a informação de uma forma clara e útil ao utilizador final, para que o objectivo principal do sistema, o de monitorizar os movimentos do pedestre, seja finalmente conseguido. A Figura 37 apresenta a arquitectura do INPERLYS, evidenciando a interacção entre as diferentes áreas do sistema.

6.3. DESCRIÇÃO DO *HARDWARE*

6.3.1. *KIT ZIGBEE™*

O *kit* de demonstração PICDEM™ Z da Microchip® fornece uma plataforma de desenvolvimento e demonstração de aplicações baseadas na tecnologia ZigBee™. O *kit* é composto por dois módulos ZigBee™. Cada módulo contém uma *motherboard* e uma antena. Na Figura 38 pode-se observar um módulo do *kit*. Além de poder demonstrar as aplicações base que já vem pré-programadas, o *kit* PICDEM™ Z fornece o código fonte da *stack* da Microchip® para a rede ZigBee™, podendo assim este ser adaptado a aplicações particulares.



Figura 38 Exemplo de módulo ZigBee™ [41]

Este *kit* é composto por:

- Duas placas de demonstração (*motherboards*) PICDEM™ Z
- Duas cartas (antenas) PICDEM™ Z
- Analisador de rede Zena™
- Aplicações de demonstração e o código fonte da *stack*

A *motherboard* utiliza um micro controlador PIC18LF4620, que já vem pré-programado, (com código para coordenador ou RFD), para realizar todas as operações necessárias à implementação de uma rede ZigBee™. A *motherboard* inclui componentes como um

sensor de temperatura, dois leds, dois botões de entrada, um interface RS-232 e um interface RJ-11. Poderá ser alimentada usando uma bateria de 9v ou através de uma fonte de alimentação externa.

A programação do micro controlador poderá ser efectuada usando uma entrada RJ-11. De realçar que a entrada RJ-11 permite efectuar *debugging* e a programação utilizando o dispositivo MPLAB ICD2 e que também suporta ICSP (*in-circuit serial programming*).

O micro controlador que se encontra na *motherboard*, é um PIC da família 18, o PIC18LF4620, ilustrado na Figura 39.

Sendo uma das vantagens da rede ZigBee™ o baixo consumo de energia, é muito importante que o micro controlador tenha um consumo baixo de energia. Uma das principais características deste micro é a *Nanowatt technology*, que consiste num método de alternância entre *clocks* com o objectivo de minimizar o consumo energético. O micro controlador pode utilizar o oscilador de *clock* principal ou usar o oscilador do *Timer1*. Quando o micro controlador pretende funcionar a velocidade máxima utiliza o *clock* principal mas quando o micro controlador tiver pouca actividade e por isso não necessitar de funcionar a uma velocidade tão elevada utiliza o *clock* fornecido pelo *Timer1*, funcionando a uma frequência mais baixa. Isto implica um decréscimo de consumo de energia que poderá chegar aos 90% durante a execução.

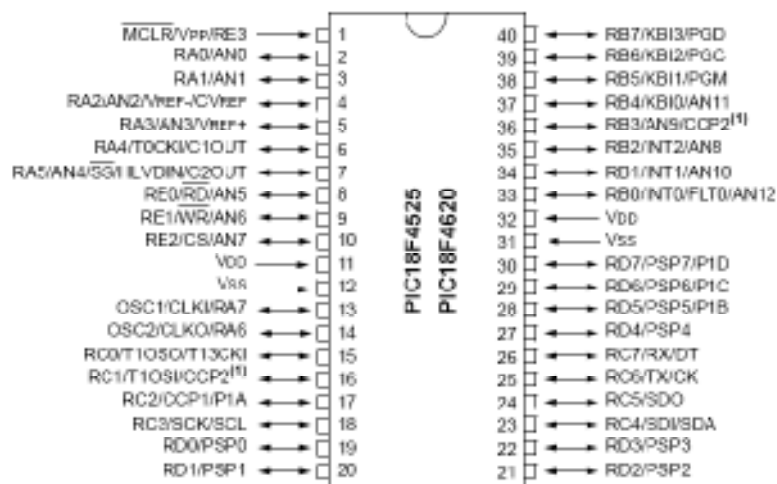


Figura 39 PIC18LF4620 [42]

Segue-se uma listagem das características mais importantes deste micro controlador:

- Interface USART endereçável com função auto *wake-up* através de *start-bit* e auto-deteção da taxa de transmissão
- Suporta protocolos SPI e I2C

- Múltiplas opções em termos de tipo de oscilador, incluindo um interno
- Conversor A/D de 10 bits integrado
- *Watchdog timer*
- 4 *timers*
- 20 fontes de interrupção
- 5 portos bidireccionais
- Memória *flash* programável de 64KBytes
- 3986 *bytes* de SRAM de memória de dados
- 1024 *bytes* de EEPROM para memória de dados
- 3 modos de consumo de energia: *Run*, *Idle* e *Sleep*
- Consumo de 11 μA (*Run*), 2.5 μA (*Idle*) e 100 nA (*Sleep*)

A antena PICDEM™ Z 2.4 GHz RF foi desenhada com o objecto de ser uma solução para o protocolo ZigBee™ para a frequência de banda de 2.4GHz. Junto com a *motherboard* formam um nó da rede ZigBee™. Um exemplo de uma antena é ilustrado na Figura 40.

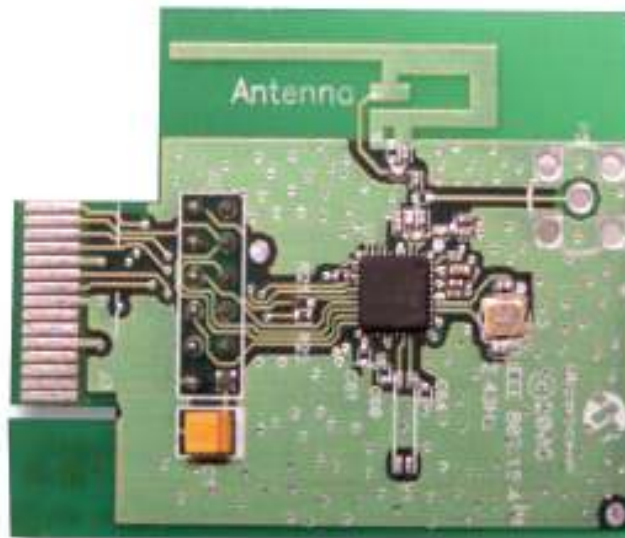


Figura 40 Antena do Kit PICDEM™ Z [41]

Usa um transceptor MRF24J40 RF 2.4GHz, ilustrado na Figura 41, fabricado pela Microchip®, compatível com a norma IEEE 802.15.4. para uma frequência de banda de 2.4GHz. É compatível com outros protocolos, como o MiWi™.

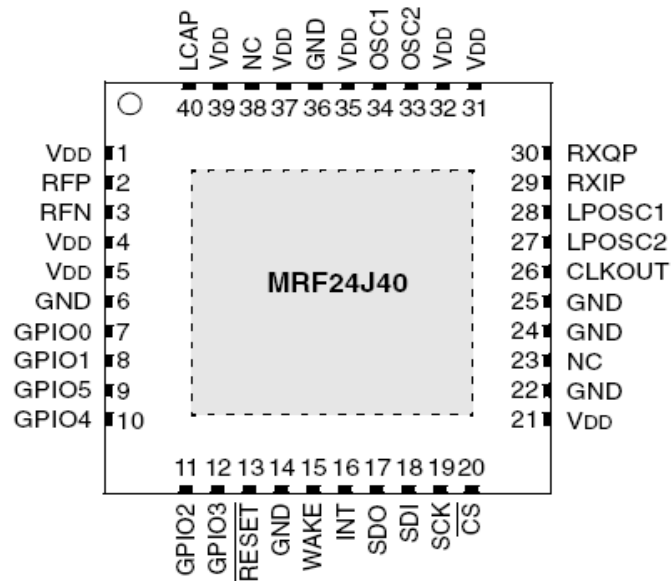


Figura 41 Transreceptor utilizado na antena do *kit* [43]

As suas principais características são:

- Preparado para trabalhar com osciladores de 20 MHz e 32.768 KHz
- Comunicação através protocolo SPI (4 ligações)
- Modo de poupança de energia (modo *Sleep*)
- Consumos na ordem dos 18 mA (modo recepção) e 22 mA (modo transmissão)

6.3.2. BÚSSOLA DIGITAL

O módulo da bússola digital usada no trabalho, ilustrada na Figura 42, é a CMPS03. Este módulo foi concebido com o propósito de auxiliar na navegação no mundo da robótica. Utiliza os sensores de campos magnéticos KMZ51 da Philips, capazes de detectar o campo magnético terrestre.

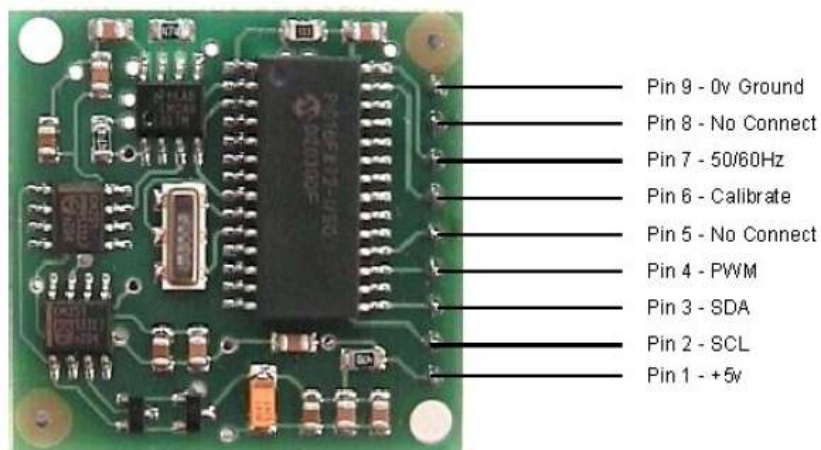


Figura 42 Bússola Digital CMPS03 [44]

A orientação é dada através de um único número. O valor deste número pode variar entre 0 e 255 (1 *byte*), se a opção for ler do registo 1 ou 0 e 359.9 (2 *bytes*) se a opção for ler dos registos 2 e 3. Esta leitura pode ser efectuada de 2 formas: através do protocolo de comunicações série I2C ou através de *Pulse Width Modulation* (PWM). Apresenta um consumo nominal de 15 mA para uma tensão de entrada de 5V. A Tabela 3 contém a descrição dos *pins* da bússola.

Tabela 3 Descrição *pins* bússola digital

Pin	Descrição
1	Alimentação
2	<i>Serial Clock</i> (SCL) para protocolo I2C
3	<i>Serial Data</i> (SDA) utilizado no protocolo I2C.
4	PWM. Alternativa ao I2C para obter orientação
5	Sem ligação
6	<i>Pin</i> utilizado para calibragem da bússola
7	Pin para selecção da frequência de operação da bússola
8	Sem ligação
9	Massa

6.3.3. ACELERÓMETRO

O acelerómetro digital utilizado no trabalho é o modelo LIS3LV0DL da STMicroelectronics[®]. É um sensor com uma saída digital da aceleração dos 3 eixos. Comunica com o exterior através do protocolo de comunicações série I2C ou SPI. A Figura 43 mostra as ligações eléctricas essenciais para o sensor funcionar.

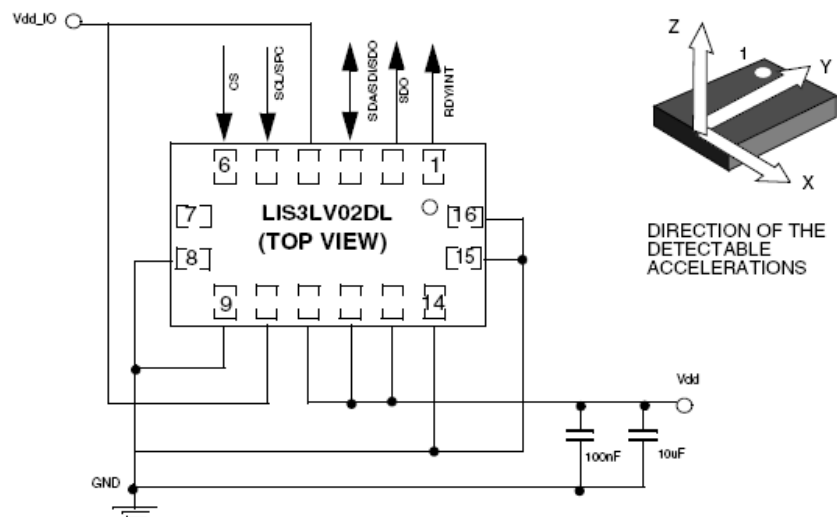


Figura 43 Ligação Eléctrica do MEMS [45]

As suas características são:

- Interfaces digitais para protocolos I2C e SPI
- Possibilidade de opção de representação de dados com 12 ou 16 bits
- Funcionalidade de activação de interrupção sempre que se detecte movimento ou se atinja um determinado valor de aceleração.
- Possibilita mais que uma frequência de leitura da aceleração
- Tensão de alimentação de 3.3V, com consumo nominal de 0.65 mA

A Tabela 4 descreve os *pins* do MEMS.

Tabela 4 Descrição *pins* MEMS [45]

<i>Pin</i>	Nome	Descrição
1	RDY/INT	Utilizado quando se pretende que origine uma interrupção quando tem dados prontos, ou quando a aceleração atinge um determinado valor
2	SDO	<i>Serial Data Output</i> (SDO). Utilizado no protocolo SPI
3	SDA/ SDI/SDO	<i>Serial Data</i> (SDA) utilizado no protocolo I2C. <i>Serial Data Input</i> (SDI). Utilizado no protocolo SPI. SDO, utilizado na versão 3 fios para dados do protocolo SPI.
4	Vdd_I/O	Alimentação
5	SCL/SPC	<i>Serial Clock</i> (SCL) para protocolo I2C. <i>Serial Port Clock</i> (SPC) para protocolo SPI
6	-	Massa.
7	NC	Não está internamente ligado
8	CK	Entrada para <i>Clock</i> externo
9	GND	Massa
10	Reservado	<i>Pin</i> Reservado
11	Vdd	Alimentação
12	Reservado	Deverá ser ligado a Vdd
13	Vdd	Alimentação
14	GND	Massa
15	Reservado	Ligar à massa ou deixar desconectado
16	GND	Massa

6.3.4. GPS

O módulo receptor GPS utilizado no trabalho é o SPK-GPS-GS405, ilustrado na Figura 44. Este módulo é baseado no *chipset* SiRF Star III e inclui uma antena omnidireccional que permite um aumento da precisão e velocidade na recepção do sinal. Apresenta um consumo de 0.75mA para uma tensão de 3.3V. Comunica com o exterior através de um interface UART *full duplex*, com nível de tensão *Transistor-Transistor Logic* (TTL). A

taxa de transmissão é de 4800 bps e utiliza o tipo de trama NMEA 0183. Leitura da posição à frequência de 1 Hz.

As suas principais características são:

- *Chipset SiRF Star III*
- Antena omnidirecional
- Suporta característica AGPS
- Capacidade de captação de 20 satélites em simultâneo
- Protocolo NMEA 0183
- Sistema de coordenadas WGS 84

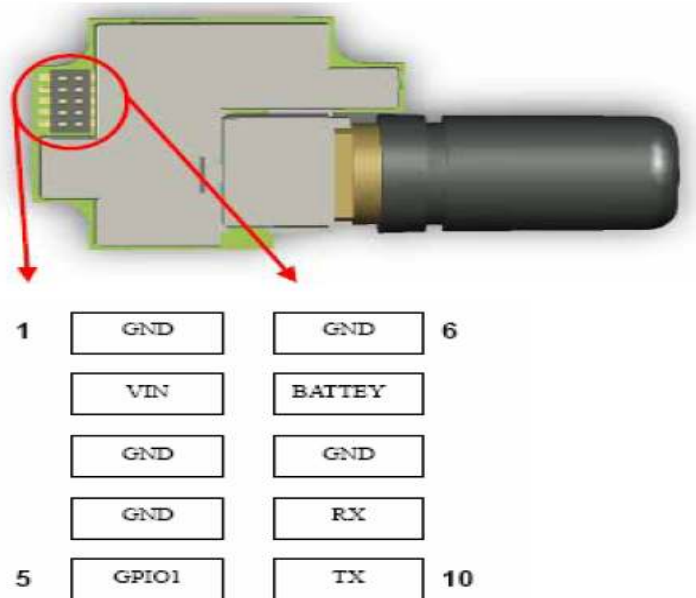


Figura 44 *Pins* GPS [46]

A tabela 5 contém a descrição dos *pins* do GPS.

Tabela 5 Descrição *pins* GPS [46]

Pin	Entrada/Saída	Descrição
1	-	Massa.
2	Entrada	Tensão de alimentação
3	-	Massa.
4	-	Massa.
5	Saída	<i>Pin</i> para funções especiais, como ligação a led.
6	-	Massa.
7	Bateria	Tensão de <i>backup</i> de alimentação à <i>Static Random Access Memory</i> (SRAM) e ao <i>Real Time Clock</i> (RTC), sempre que a tensão de alimentação principal for desligada. Não é obrigatório uma bateria de <i>backup</i> para o módulo funcionar, mas o seu desempenho será menor na primeira aquisição de dados, sempre que o módulo é ligado.
8	-	Massa.
9	Entrada	<i>Pin</i> de Recepção. Utilizado para efectuar eventuais configurações.
10	Saída	<i>Pin</i> de transmissão de dados.

6.3.5. MÓDULO BÚSSOLA DIGITAL

Este módulo é responsável por fornecer a orientação ao sistema na ausência de sinal GPS. Foi adicionado ao módulo do *kit* a bússola digital. A bússola está a comunicar com o micro controlador utilizando o protocolo I2C. Para que isso fosse possível, foi ligada aos *pins* do micro controlador RD2 e RD1. RD1 está configurado como saída do micro controlador e é utilizado para enviar impulsos de *clock* à bússola. Sempre que necessita de enviar dados para o micro controlador ou receber dados deste, a bússola utiliza o *pin* RD2. A bússola é alimentada pela *motherboard* do *kit* de demonstração. A Figura 45 mostra o módulo da bússola utilizado no sistema.



Figura 45 Módulo Bússola digital

6.3.6. MÓDULO GPS

Este módulo tem a tarefa de fornecer a posição absoluta ao sistema. Sempre que haja condições de recepção do sinal GPS, serão utilizados os dados fornecidos por este módulo. O GPS foi montado no módulo do *kit ZigBee™* e comunica com este através da porta série RS 232, a uma taxa de transmissão de 4800 bps. A comunicação é efectuada apenas num sentido, do GPS para o *kit*, uma vez que não é necessário efectuar qualquer tipo de configuração no GPS. O módulo do *kit ZigBee™* é responsável pela alimentação do GPS. A Figura 46 contém o módulo GPS utilizado no sistema.



Figura 46 Módulo GPS

6.3.7. MÓDULO MEMS

Este módulo, ilustrado na Figura 47, é responsável pela detecção dos passos dados durante o percurso. Ao módulo original do *kit* de demonstração foi adicionado o acelerómetro. O MEMS está a comunicar com o micro controlador utilizando o protocolo I2C. Para que isso fosse possível, foi ligado aos *pins* do micro controlador RD2 e RD1. RD1 está configurado como saída do micro controlador e é utilizado para enviar impulsos de *clock* ao MEMS. O MEMS sempre que necessita de enviar dados para o micro controlador ou receber dados deste, utiliza o *pin* RD2. O MEMS é alimentado pela *motherboard* do *kit* de demonstração.



Figura 47 Módulo MEMS

6.4. DESCRIÇÃO DO SOFTWARE

Como já foi indicado no ponto 6.2., o coordenador tem a função de criar a rede, recolher os dados enviados pelos módulos e validá-los. No ponto 6.4.2. é indicado de que forma a *stack* gera a formação de rede, sendo focadas as primitivas envolvidas. É também discutido no ponto 6.4.5. a forma como o coordenador recebe as tramas dos diferentes módulos e por fim, no que ao coordenador diz respeito, no ponto 6.4.9. é analisada a forma como o coordenador executa a sincronização dos dados recebidos dos diferentes *end devices*. Quanto aos *end devices*, o ponto 6.4.3. é dedicado à descrição como a *stack* gere a entrada em rede de um *end device* e no ponto 6.4.4. é referida a forma como são enviadas as tramas de um *end device* para o coordenador. Os pontos 6.4.6., 6.4.7 e 6.4.8. são detalhadas as rotinas de leitura dos diferentes dados dos diferentes *end devices*, como e quando é que

estas são executadas. Antes de tudo isto, no ponto 6.4.1. é efectuada uma descrição da estrutura base da *stack* executada em cada módulo, seja no coordenador ou num *end device*.

6.4.1. ESTRUTURA BASE DA *STACK* ZIGBEE™ DA MICROCHIP®

O funcionamento da *stack* é efectuado dentro de um ciclo infinito, mas antes de entrar nesse ciclo é necessário executar algumas funções e inicializações para que seja possível o normal funcionamento da *stack*. A começar pela inicialização da primitiva. Uma primitiva é uma instrução que é dada à *stack* para esta executar. Cada processo na *stack*, seja formação rede, procura rede, envio de dados, recepção de dados, processos relacionados com acesso ao meio ou nível físico tem uma primitiva associada. A execução de uma primitiva leva muitas vezes à execução de outras primitivas de resposta, seguindo a *stack* desta forma um caminho até à satisfação do pedido original que despoletou a primitiva original.

A primeira primitiva a ser executada pela *stack* é a primitiva `NO_PRIMITIVE`. Esta primitiva será sempre invocada quando a *stack* termina a tarefa de uma primitiva anterior, pois é esta primitiva vai permitir a execução de funcionalidades introduzidas pelo utilizador final do *kit* de demonstração. Para além da inicialização da primitiva são executadas 3 funções. A função `ConsoleInit` contém as configurações da porta série, necessárias para se poder utilizar o *hyperterminal*. A função `HardwareInit` inicializa todo o *hardware* necessário ao funcionamento do *kit*, definição de portos do micro, definição do módulo de comunicação série para comunicar com a antena, ... A função `ZigBeeInit` inicializa a *stack*.

```
void main
{
    currentPrimitive = NO_PRIMITIVE;

    ConsoleInit();
    HardwareInit();
    ZigBeeInit();

    (...)

    While(1)
    {
        ZigBeeTasks(&currentPrimitive);
        ProcessZigBeePrimitives();
        ProcessNONZigBeeTasks();
    }
}
```

```
}
```

Após as definições iniciais, essenciais ao funcionamento da *stack* e do *kit*, o código entra num ciclo infinito, só interrompido se for efectuado *reset* ao micro. A função `ZigBeeTasks` vai dar início ao funcionamento da *stack*, recebendo como parâmetro `NO_PRIMITIVE`, sendo as várias primitivas processadas utilizando a função `ProcessZigBeePrimitives`. A função `ProcessNONZigBeeTasks` como o próprio nome indica, serve para processar tudo o que não esteja relacionado com tarefas relacionadas com a rede. Será no corpo desta função que poderão ser depositadas rotinas associadas a tarefas de leituras de sensores, por exemplo. Esta função será sempre desempenhada quando não há mais nada para fazer, no que diz respeito a processos relacionados com a rede. Neste cenário a *stack* invoca esta função de 18 em 18ms. Se eventualmente também não houver nada para fazer nesta função, um *end device* poderá entrar em modo *sleep* para poupar energia.

6.4.2. FORMAÇÃO DE REDE

No início do código, a primitiva é inicializada com o parâmetro `NO_PRIMITIVE`. Quando é esta a primitiva a ser executada, é verificado se já existe uma rede formada, no caso do coordenador. Se já existe rede formada, a *stack* fica à espera de um pedido por parte do utilizador, através do *hyperterminal*, que faça despoletar outra primitiva e executar funções adicionadas à *stack*, enquanto não houver nova primitiva a executar (por exemplo, no trabalho funções como a leitura da bússola ou leitura da aceleração são efectuadas aqui). Caso não haja rede formada, irá ser iniciado o processo de formação de rede. A *stack* sabe se a rede está formada ou não através da *flag* `ZigBeeStatus.flags.bits.bNetworkFormed`. A primitiva deixa de ser `NO_PRIMITIVE` para passar a ser `NLME_NETWORK_FORMATION_request`.

```
case NO_PRIMITIVE:
if (!ZigBeeStatus.flags.bits.bNetworkFormed)
{
    (...)

    currentPrimitive = NLME_NETWORK_FORMATION_request;
}
break;
```

No fim do processo de formação da rede a primitiva a ser executada é a `NLME_NETWORK_FORMATION_confirm`. A formação da rede é confirmada pelo estado da *flag* `params.NLME_NETWORK_FORMATION_confirm.Status`:

```
case NLME_NETWORK_FORMATION_confirm:
if (!params.NLME_NETWORK_FORMATION_confirm.Status)
{
    ConsolePutROMString( (ROM char *)"PAN started
successfully " );
}
break;
```

6.4.3. ENTRADA NA REDE

Assim como no código do coordenador, para um *end device*, a primitiva é inicializada com o parâmetro `NO_PRIMITIVE`, no entanto como se trata de um *end device*, a *stack* em vez de verificar se existe rede formada vai querer saber se o dispositivo já está inserido ou não em alguma rede. A *flag* que permite esta informação é a *flag* `ZigBeeStatus.flags.bits.bNetworkJoined`. Caso não esteja inserido numa rede, a primitiva passa a ser `NLME_JOIN_request`:

```
if (!ZigBeeStatus.flags.bits.bNetworkJoined)
{
    if (!ZigBeeStatus.flags.bits.bTryingToJoinNetwork)
    {
        (...)
        currentPrimitive = NLME_JOIN_request;
    }
}
```

No final do processo, a *stack* fica a saber se o *end device* entrou com sucesso na rede ou não através da *flag* `params.NLME_JOIN_confirm.Status`. Em caso negativo inicia o processo novamente. Em caso positivo, invoca a primitiva `APSDE_DATA_request` para anunciar ao coordenador que entrou na rede.

```
case NLME_JOIN_confirm:
if (!params.NLME_JOIN_confirm.Status)
{
```

```

    ConsolePutROMString( (ROM char *)"Join successful!\r\n"
        (...
    currentPrimitive = APSDE_DATA_request;
}
break;

```

6.4.4. ENVIO DE TRAMAS PARA COORDENADOR

A primitiva invocada sempre que se pretende enviar dados na *stack*, é a primitiva `APSDE_DATA_request`. Foi criada uma função chamada `SendData` que mais não faz senão colocar a informação no *buffer*, inicializar os parâmetros necessários à execução da primitiva, e inicializar `currentPrimitive` com `APSDE_DATA_request`.

```

void SendData()
{
    TxBuffer[TxData++] = 0x00;
    TxBuffer[TxData++] = Hour;
    TxBuffer[TxData++] = Minutes;
    TxBuffer[TxData++] = Seconds;
    TxBuffer[TxData++] = MiliSeconds;

    (...

    params.APSDE_DATA_request.SrcEndpoint    = 1;
    params.APSDE_DATA_request.DstEndpoint    = 240;

    (...
    ZigBeeBlockTx();
    currentPrimitive = APSDE_DATA_request;
}

```

`TxBuffer` é o *buffer* que contém os dados a transmitir. Os parâmetros para que a primitiva seja executada, como o *end point* de origem e destino, são inicializados antes da *stack* executar a função `ZigBeeBlockTx`. Esta função bloqueia o *buffer* de transmissão da porta série.

6.4.5. RECEPÇÃO DE DADOS PELO COORDENADOR

Sempre que o coordenador recebe dados dos *end devices* a primitiva que a *stack* vai executar é a `APSDE_DATA_indication`.

```

Case APSDE_DATA_indication:
{
    switch (params.APSDE_DATA_indication.DstEndpoint)
    {
        (...

```

```

Case EP_ZDO:
    Break;

default:
    {
        (...)
    }
    Break;
}
}
Break;

```

O coordenador verifica qual o *endpoint* de destino da informação. Caso não seja nenhum em especial como o ZDO por exemplo, existe a opção *default* que lida com toda a informação destinada a um *endpoint* para o qual não esteja especificado num procedimento específico. É aqui que o coordenador está a receber os dados recebidos dos 3 módulos. É verificado apenas o endereço de origem da trama através do parâmetro `params.APSDE_DATA_indication.SrcAddress.ShortAddr` para o coordenador saber o que fazer com a informação recebida e a que diz respeito.

6.4.6. LEITURA E ENVIO DE DADOS GPS

Ao contrário dos outros módulos, o módulo da bússola não processa a informação recebida do GPS na função destinada a executar funcionalidades adicionadas pelo utilizador à *stack* do *kit* de demonstração. Neste caso os dados recebidos são processados na rotina dedicada às interrupções geradas. Como o GPS comunica com o micro através da porta série, é utilizada a interrupção gerada pelo receptor da porta série para analisar carácter a carácter da mensagem NMEA 0183 recepcionada. A Figura 48 contém o fluxograma da rotina de leitura e envio dos dados GPS.

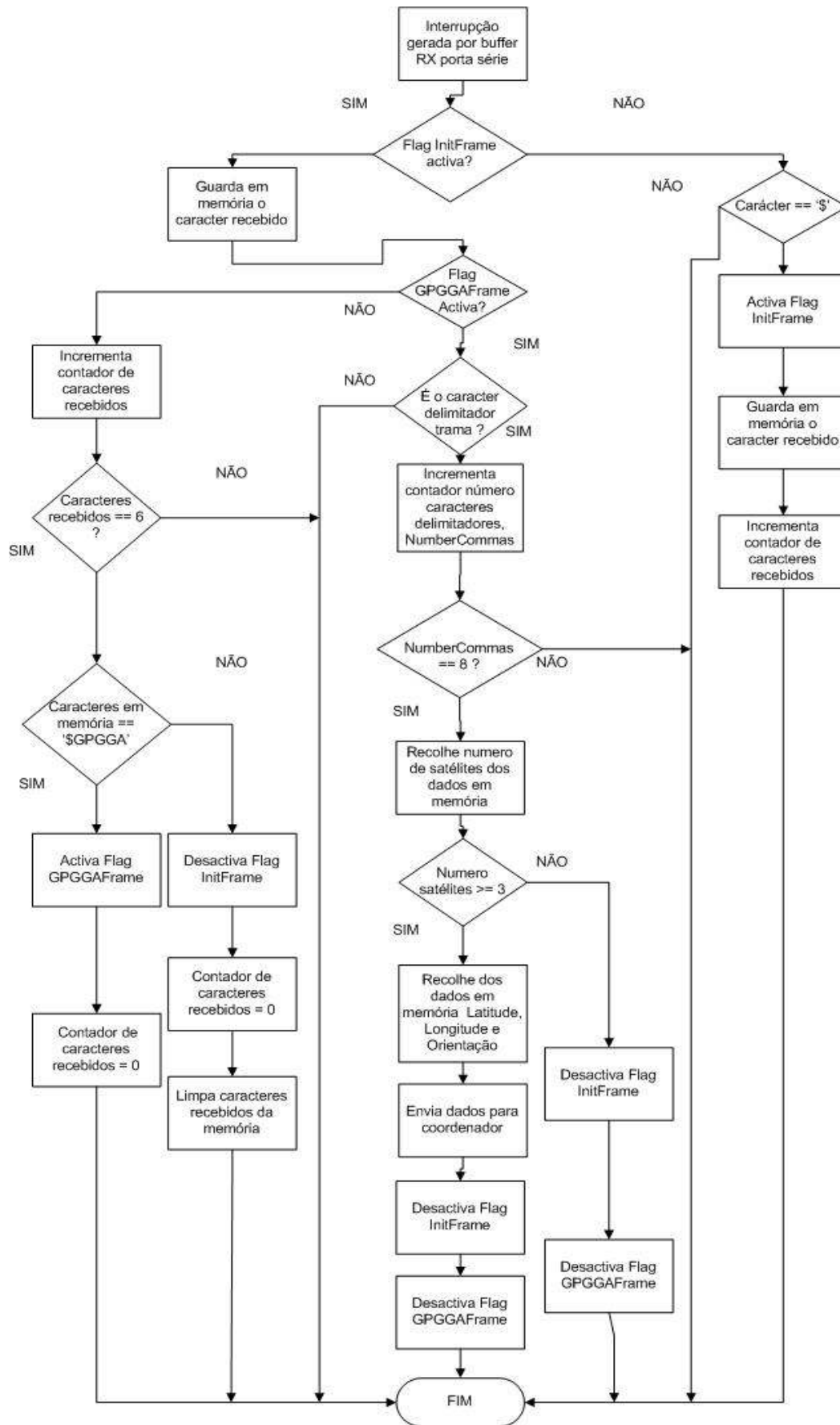


Figura 48 Rotina Leitura trama GGA do GPS

Para controlo desta rotina, existem algumas *flags* declaradas a nível global no código, como `NumberCommas`, `InitFrame`, `GPGGAFrame`.

Assim que o *buffer* de recepção da porta série está cheio, a interrupção é activa e a rotina de análise dos dados recebidos é usada. Começa-se por verificar se o carácter recebido é o indicador de trama (carácter \$) ou se `InitFrame` está activa. Em caso negativo ignora-se a restante informação recebida. Caso se trate do carácter indicador de início de trama, a *flag* `InitFrame` é activa e os próximos 5 caracteres que entretanto serão recebidos serão guardados. Se estes caracteres indicarem que se trata de uma trama GGA, a *flag* `GPGGAFrame` é activa, e como os restantes caracteres são de interesse, são armazenados após a sua recepção. Os dados são guardados no vector `DataFrame`. Caso não se trate de uma trama GGA, os dados são irrelevantes e `InitFrame` é desactiva, sendo os dados entretanto ignorados até se detectar novamente o início de trama (\$). À medida que se vão recebendo dados, com `InitFrame` e `GPGGAFrame` activas, a variável que contém o número de caracteres delimitadores de campos na trama GGA recepcionados, `NumberCommas`, será incrementada se o carácter recebido for 0x2C. Quando o seu valor for 8, indica que as informações que queremos (número de satélites em contacto, latitude, longitude e orientação) já foram recepcionadas. Caso o número de satélites seja igual ou superior a 3, são extraídas as informações da latitude, longitude e orientação, sendo enviadas para o coordenador. Em caso negativo, ignora-se a informação recebida, as *flags* são desactivas e todo o processo volta ao início. O facto de ter variáveis globais permite que a rotina não bloqueie até receber o próximo carácter, podendo assim a *stack* seguir o seu fluxo normal após atendimento da rotina de interrupção. Sempre que se receber um carácter novo e a interrupção for gerada, a rotina já sabe, graças às *flags* se esse carácter é de interesse ou não e se o deve guardar ou não.

6.4.7. LEITURA E ENVIO DE DADOS MEMS

Antes do módulo que contém o MEMS tentar entrar na rede, recolhe 10 amostras da aceleração horizontal com o pé estacionário, efectua a sua média e armazena este valor. Este passo é necessário devido à posição do MEMS no sapato não ser horizontal. Após achar o valor de aceleração correspondente à fase estacionária, o módulo vai à procura de uma rede para se poder registar.

Para a detecção do passo foi desenvolvido um algoritmo de reconhecimento do padrão da aceleração horizontal. Considere a Figura 49 que representa um esboço da onda da aceleração que se obtém durante os passos. Subdivide-se o período da onda em 2 fases: Fase1, na parte ascendente da onda e Fase 2 na parte descendente. Verifica-se que as fases começam (Fase 1) ou acabam (Fase 2) na parte estacionária (zona com aceleração próxima de zero correspondente ao pé totalmente em contacto com o chão), verifica-se também que cada fase passa por zero no seu movimento ascendente ou descendente e a transição entre fases corresponde aos picos da aceleração máxima e mínima.

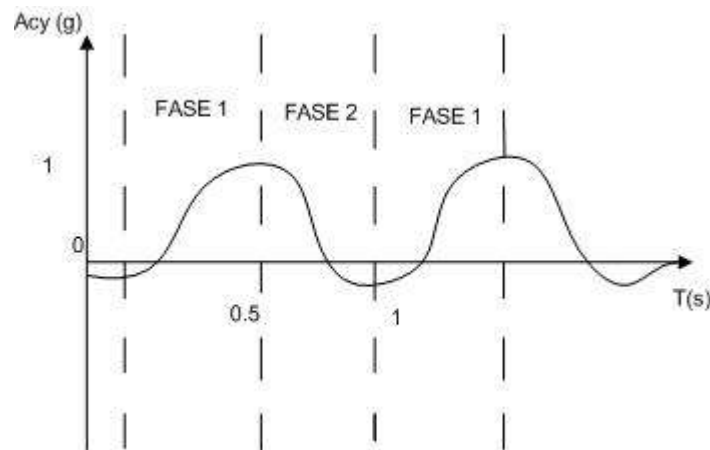


Figura 49 Descrição pontos chave na análise ao padrão da aceleração horizontal

Deste modo se o algoritmo conseguir detectar estas transições e, por consequência ser capaz de identificar a Fase 1 e a Fase 2, o passo pode ser detectado. A Figura 50 contém o fluxograma da rotina de análise ao padrão da aceleração horizontal

O algoritmo de análise do padrão da aceleração, ilustrado na Figura 50, implementa um método de janela deslizante, que guarda 5 amostras da aceleração. Na primeira vez que é executado, efectua 5 leituras da aceleração e guarda-as em memória. Analisa estas 5 amostras e verifica se detecta algum dos pontos chave da análise ao padrão da aceleração (detecção da fase estacionária, passagem por zero, picos máximo ou mínimo da aceleração). Cada um destes pontos chave tem uma *flag* que é activa quando se verifica a condição correspondente na análise. A próxima vez e seguintes que a rotina for executada, lê apenas 4 amostras que são guardadas em memória juntamente com a última amostra da leitura anterior, isto para que haja um efeito de continuidade na análise, caso contrário poderia haver pontos chave, como a passagem por zero ou a passagem por um pico

positivo ou negativo que poderiam não ser detectados, se estes se verificassem no momento de transição para leitura de 5 novas amostras.

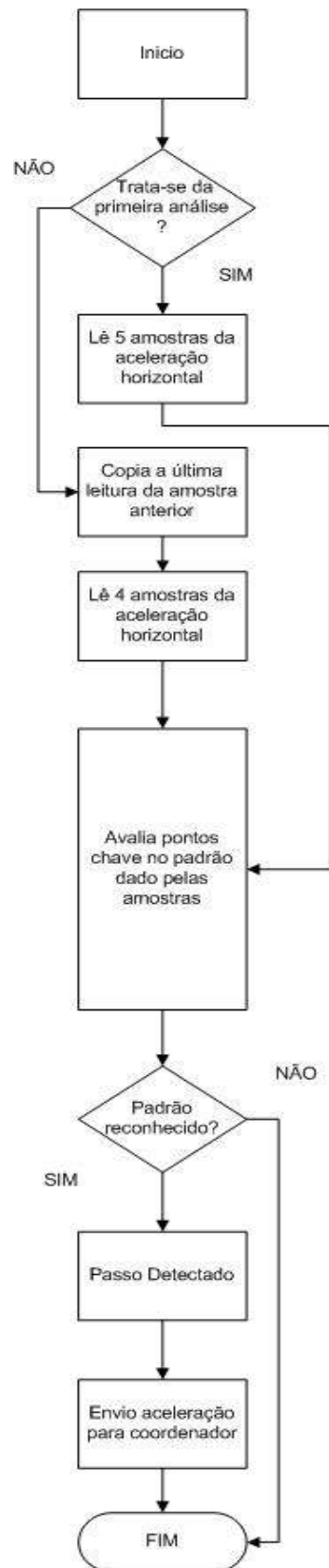


Figura 50 Rotina reconhecimento do padrão

Assim que as 2 fases forem detectadas, é enviado para o coordenador da rede o valor da aceleração máxima sentida nesse passo, assim como o tempo em que foi detectado o passo e o número de passos detectados até ao momento. Para esta rotina ser executada é necessário que o módulo já tenha entrado com sucesso na rede. Após a entrada na rede, a rotina é executada sempre que a *stack* confirma que não há primitivas relacionadas com o ZigBee™ para realizar. Esta análise por parte da *stack* demora 18ms. Uma vez chamada a rotina, é efectuada uma amostra de 226 em 226ms, fazendo com que a rotina demore a ser executada 1.130s na 1ª vez que é chamada e 904ms nas restantes.

O envio só é efectuado quando se verifica que a última trama solicitada para envio tenha sido entregue ou não com sucesso.

```
void ProcessNONZigBeeTasks(void)
{
    if(Is2read)
    {
        AxisAnalyse();
    }
}
```

A *flag* *Is2read* é accionada quando o módulo entra na rede. Após esse momento, as acelerações são analisadas através da função *AxisAnalyse*.

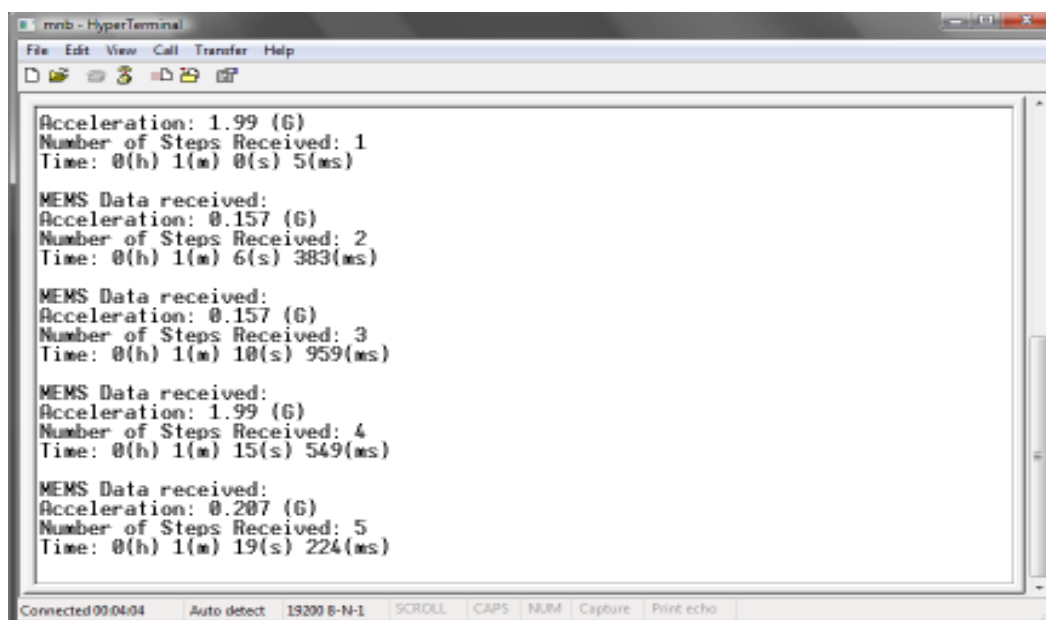
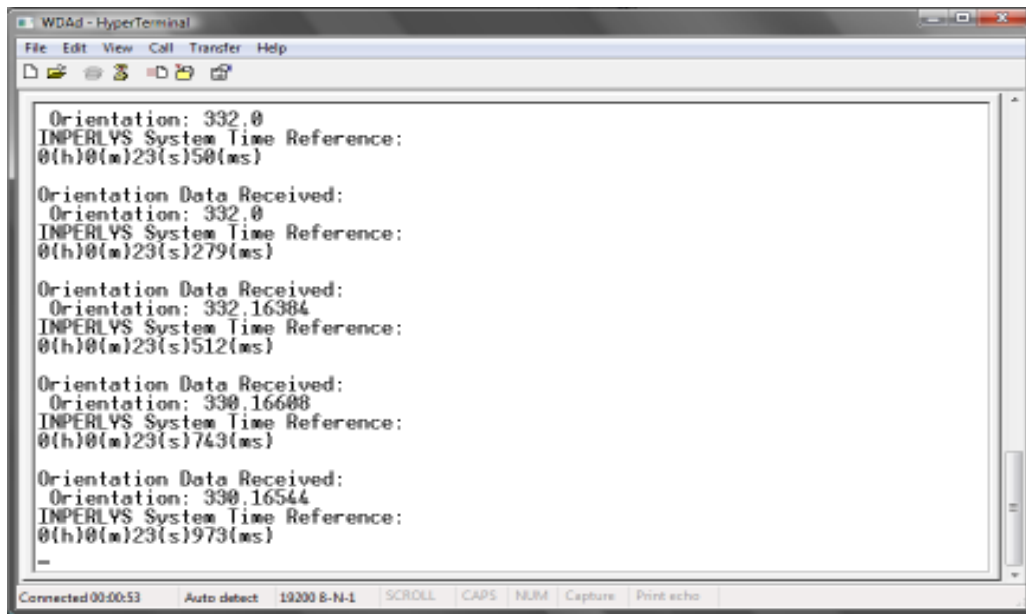


Figura 51 Aceleração impressa no *hyperterminal* após recepção do coordenador

A Figura 51 mostra os dados impressos no *hyperterminal* após o coordenador receber a informação enviada pelo módulo MEMS.

6.4.8. LEITURA E ENVIO DE DADOS BÚSSOLA

De início o módulo irá efectuar uma busca por redes às quais se possa ligar, como acontece com os outros módulos. Após entrada na rede a rotina de leitura da bússola é executada sempre que haja a confirmação por parte da *stack* de que não há primitivas a serem executadas, neste caso de 18 em 18ms e demora 143ms a ser executada.



```
WDAd - HyperTerminal
File Edit View Call Transfer Help
Orientation: 332,0
INPERLYS System Time Reference:
0(h)0(m)23(s)50(ms)
Orientation Data Received:
Orientation: 332,0
INPERLYS System Time Reference:
0(h)0(m)23(s)279(ms)
Orientation Data Received:
Orientation: 332,16384
INPERLYS System Time Reference:
0(h)0(m)23(s)512(ms)
Orientation Data Received:
Orientation: 330,16608
INPERLYS System Time Reference:
0(h)0(m)23(s)743(ms)
Orientation Data Received:
Orientation: 330,16544
INPERLYS System Time Reference:
0(h)0(m)23(s)973(ms)
--
Connected 00:00:53 Auto detect 19200 B-N-1 SCROLL CAPS NUM Capture Print echo
```

Figura 52 Orientação impressa no *hyperterminal* após recepção do coordenador

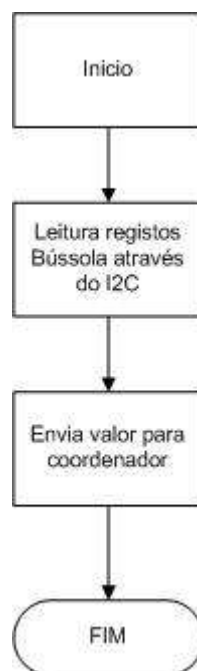


Figura 53 Fluxograma da leitura do módulo da bússola e envio de dados para o coordenador

```

void ProcessNONZigBeeTasks(void)
{
    if(Is2read)
    {
        Is2read = 0;
        ReadCompass();
        SendData();
    }
}

```

A *flag* `Is2read` é activa quando o módulo entra na rede, e quando se recebe a confirmação, positiva ou negativa, de envio da trama. É desactiva antes de se executar a função de leitura da bússola `ReadCompass` para garantir que não efectuadas novas leituras antes do envio da última trama. A Figura 52 mostra os dados impressos no *hyperterminal* pelo coordenador, após este receber dados do módulo da bússola. Na Figura 53 está representado o fluxograma de leitura dos dados da bússola e seu envio para o coordenador.

6.4.9. SINCRONIZAÇÃO DE DADOS

Para o sistema efectuar a fusão dos dados provenientes dos diferentes módulos é fundamental saber quando é que foi lido um determinado dado, saber por exemplo, se a orientação que recebe agora foi medida a instantes atrás e pode ser relacionada com o passo que acabou de receber ou se de facto a orientação já tem 10 segundos de atraso e o passo foi detectado apenas há 2 segundos atrás pelo que deverá ser descartada. Para isso é necessário criar um mecanismo que permita obter a sincronia de dados.

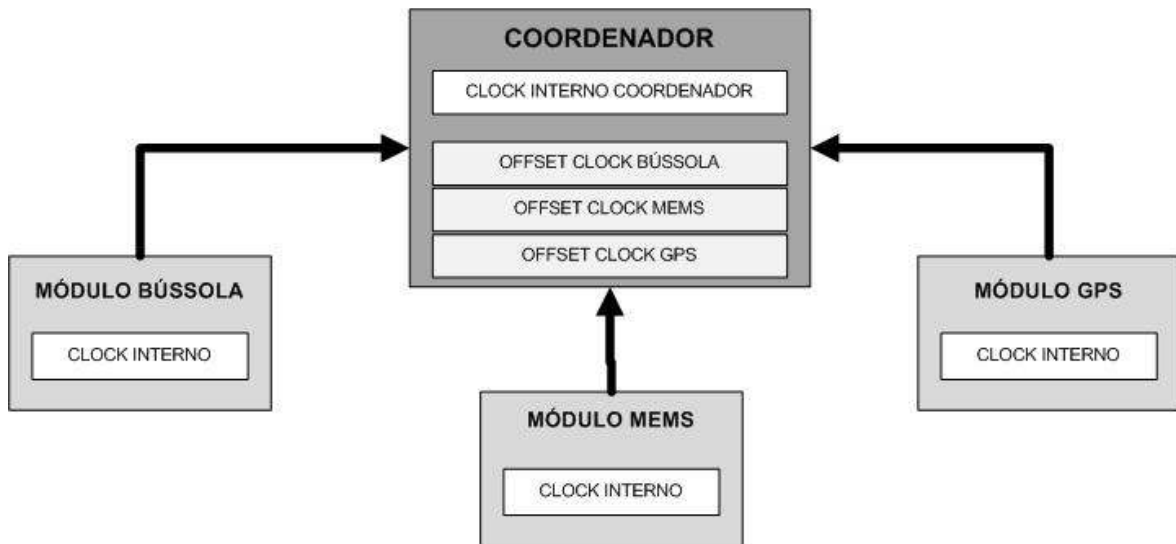


Figura 54 Sistema de referência para clocks

A Figura 54 mostra o sistema desenvolvido. O coordenador, após criar a rede, dá início ao seu *clock*. Cada um dos *end devices*, assim que entram na rede, iniciam os respectivos *clocks*. Cada *end device* tem conhecimento apenas do seu relógio. O coordenador ao receber a informação de que um determinado *end device* entrou na rede, registra esse momento e guarda como *offset* para futuras recepções de dados desse *end device*. Por exemplo, supondo que o coordenador tinha criado a rede há 00(h):00(m):12(s):689(ms). Neste momento entra na rede o módulo da bússola. Será este o tempo de *offset* para a bússola, o *clock* do módulo da bússola, como entrou neste instante na rede, será 0 (na realidade existe o tempo de atraso associado ao envio de informação mas esse valor para já é ignorado). Um *end device* quando envia uma trama de dados, além da informação dada pelo sensor (orientação, aceleração...) enviará também o seu *clock* interno no momento em que a trama foi gerada. O coordenador ao receber a trama, vai somar ao tempo do *clock* do respectivo *end device*, o seu *offset* (no caso do exemplo dado, somaria 12s e 689ms) e registra este valor, com o respectivo dado associado, orientação, aceleração, latitude... Este mecanismo torna ainda possível saber se uma trama chega fora da ordem em que deveria chegar.

Quando o *end device* entra na rede, o valor de *offset* associado a si é registado:

```

If (params.NLME_JOIN_indication.ExtendedAddress.v[7] ==
0x04)
{
    (...)
}

```

```
MEMSHourOffset = Hour;
MEMSMinuteOffset = Minute;
MEMSSecondOffset = Second;
MEMSMiliSecondOffset = MiliSecond;
}
```

O valor `MiliSecondTemp` guardará o valor final de referência para o campo dos milésimos de segundo. É necessário prever se a soma excede os valores máximos possíveis e rectificar se necessário o valor final.

```
MiliSecondTemp = OrientationMiliSecond +
OrientationMiliSecondOffset;

if (MiliSecondTemp > 999)
{
    MiliSecondTemp = MiliSecondTemp - 1000;
    SecondTemp = SecondTemp + 1;
}

(...)
```

Esta rotina repete-se para os restantes campos do sistema temporal, segundo, minuto e hora. Neste último caso não existe teste de valor máximo, uma vez que o sistema temporal de referência não conta dias:

```
HourTemp = OrientationHour + OrientationHourOffset;
```

6.5. ANÁLISE DE RESULTADOS

Para validar o sistema foram efectuados alguns testes. Os resultados dos testes permitem indicar que o módulo da bússola tem uma boa resposta às alterações de orientação provocadas pelo utilizador, no que diz respeito ao envio de informação. Quanto ao módulo GPS revelou ser eficiente quando usado em zonas outdoor com pouca ou nenhuma obstrução aos sinais GPS. Relativamente ao módulo do sensor MEMS é o que apresenta mais dificuldades. O algoritmo de análise da aceleração dificulta o processo de gestão de envio e confirmação de entrega da trama de informação. Este algoritmo falha a detecção de alguns passos.

Quanto ao envio de informação, os testes revelaram uma eficácia de 100% na entrega de informação às tramas solicitadas para envio, em tempos aceitáveis para o módulo da bússola digital para o módulo GPS. Quanto ao módulo MEMS, apesar dos atrasos no envio de informação já referenciados, a taxa de sucesso de entrega registada também foi de 100%. Nos pontos seguintes serão detalhados os testes e resultados efectuados a cada módulo.

6.5.1. MÓDULO BÚSSOLA DIGITAL

Os testes efectuados ao módulo da bússola digital consistiram no seguinte: numa 1ª fase, rodou-se a bússola em movimentos no sentido dos ponteiros de relógio e em movimentos contrários ao sentido de ponteiros de relógio. Verificou-se que o valor obtido da orientação (de 0 a 359.9) aumentava ou decrescia de acordo com o movimento de rotação. De seguida testou-se o módulo da bússola com os restantes módulos em rede, em ambiente *outdoor* e em ambiente *indoor*. Para tal foram efectuados 10 testes em carga em cada ambiente, com o módulo da bússola constantemente a debitar dados. Nos primeiros 5 testes em cada ambiente, após o coordenador ter dado entrada na rede, foi dada ordem ao módulo da bússola para entrar na rede. Os restantes módulos deram entrada a seguir. Nos restantes 5 testes, o módulo da bússola foi o último a dar entrada. A ordem de entrada da bússola na rede serviu para verificar como a *stack* lida com os processos de entrada e manutenção do módulo na rede quando estão a ser enviados dados. Cada teste teve uma duração de 5 minutos.

Nos 10 testes realizados *indoor*, obteve-se uma taxa de sucesso de 100% de entrega para as tramas solicitadas para os 2 conjuntos de 5 testes, obtendo-se igual resultado para os testes *outdoor*.

O tempo que decorre entre o envio da trama e a confirmação positiva ou negativa do seu envio é em média de 143ms. O tempo que demora o coordenador a receber a informação, até à impressão de dados no *hyperterminal* é em média de 234ms. Os valores são os mesmos para cenários com e sem outros módulos em rede.

6.5.2. MÓDULO MEMS

Foram realizados testes num percurso em linha recta, a uma velocidade constante com um tipo de passada normal.

Quando o algoritmo de detecção de passos foi testado num projecto à parte, sem a integração com a *stack* do ZigBee™, os resultados eram satisfatórios, em 10 passos, o algoritmo detectava 8 ou 9 passos. No entanto ao integrar o algoritmo com a *stack* a eficácia na detecção diminuiu bastante. Nos mesmos 10 passos o algoritmo detecta uma média de 5 a 6 passos. Acredito que a razão da diminuição da eficácia deve-se ao facto do algoritmo, neste momento não estar a contar os passos que se dá entre o envio e a confirmação positiva ou negativa de sucesso de envio. O tempo de envio médio registado para 6 tentativas, desde o pedido de

envio de tramas do módulo MEMS até a confirmação positiva ou negativa de envio, com o módulo MEMS sozinho na rede é de 2.686s, enquanto que o tempo de envio médio registado para 6 tentativas desde o pedido de envio de tramas do módulo MEMS até a confirmação positiva ou negativa de envio com outros módulos na rede foi de 2.798s. Ou seja neste intervalo de tempo uma pessoa pode dar entre 1 a 2 passos nas mesmas condições em que o teste foi realizado. Foi efectuada uma tentativa de enviar dados sempre que fossem detectados passos, independentemente de ter recebido a confirmação ou não de envio da última trama, para evitar assim a perda de passos. No entanto verificou-se que esta solução degradava muito o funcionamento da rede, havendo em média cerca de 5 a 6 segundos de atraso entre a detecção do passo e o coordenador a receber a informação, pelo que se abandonou esta alternativa. A solução para o problema de perda de passos poderá passar pelo armazenamento dos valores de aceleração em memória e assim que existirem condições de envio de trama (ou seja ter recebido confirmação de entrega ou não da última trama) envia esses dados, passando nesta caso uma trama poder enviar mais que uma aceleração com o respectivo tempo em que foi registada.

Entretanto foi efectuado um teste para confirmar a origem do elevado tempo decorrido entre o envio de dados e a confirmação de sucesso ou não dos mesmos. O teste consistia em enviar uma trama de dados pré definida. Verificou-se que o tempo decorrido entre o envio da trama e recepção de confirmação de sucesso de entrega ou não, é apenas de 45ms. Como cada amostra decorre de 220 em 220ms, sendo que o tempo total da rotina de análise da aceleração demora cerca de 904ms (1.130 na primeira execução), torna-se evidente que esta rotina é o factor que contribui para um tempo tão elevado.

6.5.3. MÓDULO GPS

O teste ao módulo GPS consistiu em tentar utilizá-lo em ambiente *indoor*, *outdoor* com obstruções ao sinal GPS, como árvores, edifícios e *outdoor* sem obstruções. Uma vez que não existe nenhuma bateria de *backup* a alimentar a SRAM do GPS, confirmou-se que sempre que o GPS é ligado, demora cerca de 40 segundos a obter a posição, mesmo num cenário sem qualquer obstrução. Confirmou-se também a cadência de fornecimento de dados de 1s. Verificou-se que, em ambientes *indoor* o GPS não capta sinal dos satélites, como já se esperava, mesmo em zonas com possível contacto com satélites, como perto da janela. Em ambientes *outdoor*, confirmou-se o estabelecimento de ligação com satélites, mesmo em zonas que poderiam apresentar um obstáculo ao sinal GPS, como zonas com alguma

florestação. A taxa de transmissão em ambientes outdoor foi de 100% para 5 testes realizados em cada um dos 3 cenários. Cada teste teve a duração de 5 minutos.

7. CONCLUSÕES E FUTUROS DESENVOLVIMENTOS

O sistema INPERLYS, neste momento, é um sistema independente de fontes externas, capaz de fazer recolha dos dados dos vários módulos que o compõe.

No que diz respeito à orientação, está totalmente operacional, o módulo é capaz de fornecer a orientação, mesmo em ambientes que poderiam mostrar-se adversos à captação do campo terrestre por parte da bússola, como no interior de edifícios. O módulo do GPS também está totalmente operacional, fornece os elementos necessários quando está em contacto directo com os satélites, mesmo em zonas com alguma obstrução.

Quanto à análise do passo detectado e estimativa da distância percorrida, o elemento necessário a este cálculo, a aceleração do passo, já é fornecido ao sistema. No que diz respeito ao algoritmo de obtenção do passo, o aumento de eficácia do algoritmo pode passar pela solução já avançada na análise de resultados, ou seja, implementar um algoritmo que armazene os valores de aceleração em memória e assim que existirem condições de envio de trama, envia os dados armazenados até esse momento. Outra solução possível será enriquecer o algoritmo de análise do passo com dados da aceleração de outros acelerómetros, em posições diferentes do corpo, para evitar falsas detecções de

passos. Por exemplo, ler a aceleração vertical da perna, ao nível da tíbia, juntamente com a aceleração ao nível do peito. Isto evita que, por exemplo, se o utilizador estiver simplesmente a baloiçar o pé, mas estiver sentado, o sistema contabilize passos e envie acelerações que se iriam traduzir em distância supostamente percorrida, quando na realidade o utilizador está sentado.

Esta aceleração ao nível do peito pode ainda dar outras funcionalidades ao sistema, como a de monitorização da posição do utilizador, podendo dar um sinal de alerta caso o utilizador se mantenha imóvel por um período de tempo pré definido. Esta valência, a de monitorização não só da distância percorrida, mas como também da posição do utilizador abre novos horizontes para a aplicação do sistema.

Outro desenvolvimento a ser considerado no futuro é a introdução da detecção de deslocamento vertical, ou seja dar ao sistema ferramentas que permitam obter a altitude do utilizador. Esta análise poderá ser feita através de mais *hardware*, como sensores para detectar as alterações de pressão atmosférica sentidas quando o utilizador sobe ou desce ou, poderá passar novamente pela análise do padrão das acelerações sentidas. A componente aplicacional do INPERLYS resume-se neste momento ao envio dos dados para a aplicação *hyperterminal*. No futuro será esta componente responsável pela integração da aceleração e consequente estimativa da distância percorrida, devido aos fracos recursos do micro controlador para este tipo de cálculo.

Referências Documentais

- [1] Gérard Lachapelle – “*Pedestrian navigation with high sensitivity GPS receivers and MEMS*”. Springer – Pers Ubiquit Comput. Volume 11, Páginas 481-488, 2007.
- [2] Valéri Renaudin, Okan Yalak and Philip Tomé – “*Hybridization of MEMS and assisted GPS for Pedestrian Navigation*”. InSide GNSS, Janeiro/Fevereiro, 2007.
- [3] Stéphane Beauregard and Harald Haas – “*Pedestrian Dead Reckoning: a Basis of Personal Positioning*”. Proceedings of the 3rd Workshop on Positioning, Navigation and Communication, 2006.
- [4] Paramvir Bahl e Venkata N. Padmanabhan – “*RADAR: An In Building RF-based User Location and Tracking System*”. 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Volume 2, Páginas 775-784. Israel, 2000.
- [5] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau e Abhishek P. Patil – “*LANDMARC: Indoor Location Sensing Using Active RFID*”. Proceedings of the first IEEE International Conference on Pervasive Computing and Communications, 2003.
- [6] Jeffrey Hightower, Gaetano Borriello e Roy Want – “*SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength*”, 2000
- [7] Lei Fang, Panos F. Antsaklis, Luis Montestruque, M. Brett McMickell, Michael Lemmon, Yashan Sun, Hui Fang, Ioannis Koutroulis, Martin Haenggi, Min Xie and Xiaojuan Xie – “*Design of a wireless Assisted Pedestrian Dead Reckoning System – The NavMote Experience*”. IEEE transactions on Instrumentation and Measurement, 2005.

- [8] Wei Chen, Zhongqian Fu, Ruizhi Chen, Yuwei Chen, Octavien Andrei, Tuomo Kröger and Jianyu Wang – “*An Integrated GPS and Multi-sensor Pedestrian Positioning System for 3D Urban Navigation*”. Urban Remote Sensing Joint Event, 2009.
- [9] Valéri Renaudin, Okan Yalak, Philip Tomé, Bertrand Merminod – “*Indoor Navigation of Emergency Agents*”. Science - European Journal of Navigation, Volume 5, N° 3, Julho 2007.
- [10] Jeong Won Kim, Han Jin Jang, Dong-Hwan Hwang and Chansik Park – “*A Step, Stride and Heading Determination for the Pedestrian Navigation System*”. Journal of the Global Positioning Systems, Volume 3, N° 1-2, Páginas 273-279, 2004.
- [11] Stéphane Beauregard, Widyawan and Martin Klepal – “*Indoor PDR Performance Enhancement using Minimal Information and Particle Filters*”. Position, Location and Navigation Symposium, IEEE/ION, 2008.
- [12] Quentin Ladetto and Bertrand Merminod – “*In Step with INS*”. GPS World, Outubro 2002.
- [13] Quentin Ladetto, Vincent Gabaglio and Bertrand Merminod – “*Two different approaches for Augment GPS Pedestrian Navigation*”. International Symposium on Location Based Services for Cellular Users. Locellus, 2001.
- [14] S. Godha, M. G. Petovello and G. Lachapelle – “*Performance analysis of MEMS/HSGPS/Magnetic Sensor Integrated System in Urban Canyons*”. ION GNSS, Long Beach, California. Setembro, 2005.
- [15] Jari Saarinen, Jussi Suomela, Seppo Heikkilä, Mikko Elomaa e Aarne Halme – “*Personal Navigation System*”. IEEE International Conference on Intelligent Robots and Systems, Sendai, Japan. 28 de Setembro – 2 de Outubro, 2004.
- [16] Akihiro Hamaguchi, Masayuki Kanbara and Naokazu Yokoya – “*User Localization using wearable Electromagnetic Tracker and Orientation Sensor*”. 10th IEEE International Symposium on Wearable Computers, 2006.

- [17] Harvey Weinberg – “*Using the ADXL202 in Pedometer and Personal Navigation Systems*”. Application Note AN602, Analogue Device.
- [18] Koichi Sagawa and Yutaka Satoh – “*Non-restricted measurement of walking distance*”. IEEE International Conference on Systems, Man and Cybernetics. 2000.
- [19] Lauro Ojeda e Johann Borenstein – “*Personal Dead-reckoning System for GPS-denied Environments*”. IEEE international Conference on Safety, Security and Rescue Robots (SSRR2007). Roma, Itália, 27-29 de Setembro de 2007.
- [20] Ahmed El-Rabbany – “*Introduction to GPS the Global Positioning System*”. Artech House mobile communications series. 2002.
- [21] Goran M. Djuknic and Robert E. Richton – “*Geolocation and Assisted GPS*”. IEEE Journal Computer Communications, Volume 34, Páginas 123-125. February 2001.
- [22] Eric Foxlin – “*Pedestrian Tracking with Shoe-Mounted Inertial Sensors*”, IEEE Computer Graphics and Applications, Volume 25, Páginas 38-46. 2005.
- [23] Mohamed Gad-el-Hak – “*The MEMS handbook*”. Editora CRC Press LLC, 2002.
- [24] Kaigham J. Gabriel – “*Microelectromechanical Systems (MEMS) Tutorial*”. IEEE International Test Conference. Washington DC, USA. 1998.
- [25] Inzarulfaisham Abd. Rahim, Muhamad Azman Miskam, Othman Sidek, Shahril Azwan Zaharudin, Mohammad Zaidi Zainol e Shukri Korakkottil Kunhi Mohd – “*Development of a Vibration Measuring Unit Using a Microelectromechanical System Accelerometer for Machine Condition Monitoring*”. European Journal of Scientific Research. Volume 35, Nº 1, Páginas 150-158. 2009.
- [26] Analog Devices – “*ADXL05 Datasheet*”.1996.
- [27] Robert Racz, Christian Schott e Samuel Hubber – “*Electronic Compass Sensor*”. IEEE Sensors, Páginas 1446-1449, Vol.3. 2004.
- [28] Carlos Cordeiro, Kiran Challapali e Dagnachew Birru – “*IEEE 802.22: An Introduction to the first Standard based on Cognitive Radios*”. Journal od Communications, Volume 1, nº1. Abril de 2006.

- [29] Qian Zhang, Chuanxiong Guo, Zihua Guo e Wenwu Zhu – “*Efficient Mobility Management for Vertical Handoff between WWAN and WLAN*”. IEEE Communications Magazine. Novembro de 2003.
- [30] Zakhia Abichar, Yanlin Peng e J. Morris Chang – “*WiMAX: The Emerge of Wireless Broadband*”. IT Professional, páginas 44-48, Agosto de 2006.
- [31] Guido R. Hiertz, Sebastien Max, Rui Zhao, Dee Denteneer e Lars Berlemann – “*Principles of IEEE 802.11s*”. Computer Communications and Networks. Honolulu, Hawaii. Agosto de 2007.
- [32] Josip Lörincz e Dinko Begušić – “*Physical Layer Analysis of Emerging IEEE 802.11n WLAN Standard*”. International Conference Advanced Communication Technology, 2006.
- [33] Lieven De Strycker– “*Wireless Networks and Applications*”, Semana Internacional ISEP. 2008
- [34] Jin - Shyan Lee, Yu - Wei Su e Chung - Chou Shen – “*A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee e Wi-Fi*”. Industrial Electronics Society, páginas 46-51. Taipé, Tailândia. Novembro de 2007.
- [35] Norma IEEE - “*IEEE Std 802.15.4a™*”, 2007
- [36] E. Dalila Pinedo-Frausto, J. António Garcia-Macias – “*An Experimental Analysis of ZigBee Networks*”. Local Computer Networks, páginas 723-729. Montreal. Outubro de 2008.
- [37] Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta e Y. Fun Hu - “*Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards*. Computer Communications in Wired/Wireless Internet Communications, Volume 30, páginas 1655-1695. Maio de 2007.
- [38] Houda Labiod, Hossan Afifi e Constantino De Santis – “*Wi-Fi™, Bluetooth™, ZigBee™ and WiMax™*”. Editora Springer. 2007.
- [39] David Flowers, Kim Otten, Yifeng Yang e Nilesh Rajbharti – “*Microchip Stack for the ZigBee™ Protocol*”. 2007.

- [40] H. Ferreira, L. Figueiredo e A. Meireles – “*INPERLYS – Independent Personal Location System*”. Ambient Intelligent Fórum 2009 (AMIF2009). República Checa, Setembro 2009.
- [41] Microchip – “*PICDEM™ Z Demonstration Kit User’s Guide*”. 2007.
- [42] Microchip – “*28/40/44-Pin Enhanced Flash Microcontrollers with 10-bit A/D and nano Watt Technology*”. PIC18LF4620 Datasheet. 2008.
- [43] Microchip – “*IEEE 802.15.4 2.4 GHz RF Transceiver*”. MRF24J40 Datasheet. 2008.
- [44] CMPS03 Datasheet.
- [45] STMicroelectronics – “*LIS3LV02DL Datasheet*”. Revisão 2. 2008.
- [46] SPK Electronics Co – “*Specification Smart Antenna SPK-GPS-GS405*”. Versão 1.0. 2006.

