

WSNs Pervasivas Aplicadas a Sistemas de Apoio à Vida

Por

Nuno André Dias Branco

Orientador: Doutor Carlos Manuel José Alves Serôdio

Co-orientador: Doutor Pedro Miguel Mestre Alves da Silva

Dissertação submetida à

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

para obtenção do grau de

MESTRE

em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no

DR – I série – A, Decreto-Lei n.º 74/2006 de 24 de Março e com as alterações introduzidas pelos

Decretos-Leis n.º 107/2008, de 25 de Junho, e 230/2009, de 14 de Setembro, e demais legislação

aplicável e no Regulamento de Ciclo de Estudos Conducente ao Grau de Mestre da

Universidade de Trás-os-Montes e Alto Douro

DR, 2.ª série – n.º 149/2011 de 4 de Agosto

WSNs Pervasivas Aplicadas a Sistemas de Apoio à Vida

Por

Nuno André Dias Branco

Orientador: Doutor Carlos Manuel José Alves Serôdio

Co-orientador: Doutor Pedro Miguel Mestre Alves da Silva

Dissertação submetida à

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

para obtenção do grau de

MESTRE

em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no

DR – I série – A, Decreto-Lei n.º 74/2006 de 24 de Março e com as alterações introduzidas pelos

Decretos-Leis n.º 107/2008, de 25 de Junho, e 230/2009, de 14 de Setembro, e demais legislação

aplicável e no Regulamento de Ciclo de Estudos Conducente ao Grau de Mestre da

Universidade de Trás-os-Montes e Alto Douro

DR, 2.ª série – n.º 149/2011 de 4 de Agosto

Orientação Científica :

Doutor Carlos Manuel José Alves Serôdio

Professor Associado com Agregação do
Departamento de Engenharias da Escola de Ciências e Tecnologia da
Universidade de Trás-os-Montes e Alto Douro

Doutor Pedro Miguel Mestre Alves da Silva

Professor Auxiliar do
Departamento de Engenharias da Escola de Ciências e Tecnologia da
Universidade de Trás-os-Montes e Alto Douro

”Se puderes, ajuda os outros; se não o puderes fazer, ao menos não lhes faças mal.”

Dalai Lama (1935 – presente)

À minha família

WSNs Pervasivas Aplicadas a Sistemas de Apoio à Vida

Nuno André Dias Branco

Submetido na Universidade de Trás-os-Montes e Alto Douro
para o preenchimento dos requisitos parciais para obtenção do grau de
Mestre em Engenharia Electrotécnica e de Computadores

Resumo — A crise demográfica e económica em que actualmente a Europa e por conseguinte Portugal, obrigam a que o serviço nacional de saúde tenha de trabalhar com menos pessoal médico para uma maior quantidade de utentes em cuidados continuados. Os meios tecnológicos que hoje existem permitem oferecer uma melhor e maior quantidade de serviços, entre os quais de localização e monitorização em tempo real de utentes em regime hospitalar, oferecendo assim tanto a utentes e auxiliares uma melhor qualidade de vida.

Esta dissertação descreve o desenvolvimento de um sistema inteligente para a localização e monitorização, dentro de edifícios, de pacientes em regime hospitalar. A solução apresentada contém um *software* que serve de interface para os auxiliares de saúde e sistema sem fios que permite a localização de um paciente. Os auxiliares podem em tempo real saber informações como localização, bem como verificar a condição física dos utentes através de sensores e podem controlar os movimentos dos pacientes em cadeiras de rodas. O sistema móvel recolhe os dados e envia através do rede sem fios, ficando o *software* responsável de emitir alertas.

Palavras Chave: WLAN, CAN, IEEE 802.15.4, Localização *indoor*, Rede de sensores, Monitorização remota.

Pervasive WSNs for Life Support Systems

Nuno Nuno André Dias Branco

Submitted to the University of Trás-os-Montes and Alto Douro
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computers

Abstract — Demographic and economic crisis in Europe and consequently in Portugal, requires National Health Service to do its task with less medical staff, for a greater number of patients in continuous health care. The technology that exists today allows to provide more and better services, including localization and real-time monitoring of hospitalized patients, thus offering to patients and assistants a better quality of life.

This thesis describes the development of an intelligent system for indoor location and monitoring of hospitalized patients. The presented solution contains a software that provides an interface to the medical staff, and a wireless system that allows the localization of a patient. Medical staff can know in real time information about the location, and check the physical condition of patients, through sensors. They also can also control the movements of patients in wheelchairs. The mobile system collects data and sends it through a wireless network, the software has the task to launch warnings.

Key Words: WLAN, CAN, IEEE 802.15.4, *Indoor location, Sensor network, Remote monitoring.*

Agradecimentos

Os meus agradecimentos ao Magnífico Reitor da Universidade de Trás-os-Montes e Alto Douro, Professor Doutor António Fontainhas Fernandes, bem como ao Presidente da Escola de ciências e Tecnologia, Professor Doutor José Boaventura Ribeiro da Cunha, assim como ao Director do Mestrado em Engenharia Electrotécnica e de Computadores da Universidade de Trás-os-Montes e Alto Douro, Professor Doutor Paulo Moura Oliveira.

Ao Professor Doutor Carlos Serôdio, Professor Associado com Agregação do departamento de Engenharias da Universidade de Trás-os-Montes e Alto Douro, orientador deste trabalho, pela sua motivação, pelas suas sugestões, ideias inovadoras e orientações.

Ao Professor Doutor Pedro Mestre, Professor Auxiliar do departamento de Engenharias da Universidade de Trás-os-Montes e Alto Douro, na qualidade de co-orientador deste trabalho, pela sua motivação, pelas suas sugestões, ideias inovadoras e orientações.

À minha família, pelo incentivo e motivação que me deram durante todo este tempo. A todos os meus colegas do curso de Engenharia Electrotécnica e de Computadores da Universidade de Trás-os-Montes e Alto Douro pela sua amizade e simpatia.

A todos, um sincero obrigado!

UTAD,
Vila Real, 15 de Outubro de 2013

Nuno André Dias Branco

Índice geral

Resumo	ix
<i>Abstract</i>	xi
Agradecimentos	xiii
Índice de tabelas	xvii
Índice de figuras	xix
Acrónimos e abreviaturas	xxi
1 Introdução	1
1.1 Motivação	2
1.2 Objectivos	3
1.3 Organização da Dissertação	4
2 Sistemas de Localização <i>Indoor</i>	5
2.1 Técnicas de Localização	5
2.2 Projectos Existentes	7
3 Enquadramento Tecnológico	13
3.1 Redes sem Fios	13
3.1.1 Redes Locais (WLAN) - IEEE 802.11	13
3.1.2 Redes de Pessoais (WPAN) - IEEE 802.15.4 e <i>Bluetooth</i>	14

3.1.3	GSM	20
3.2	CAN	27
3.2.1	Modelo OSI e CAN	27
3.2.2	Estrutura do CAN	27
3.2.2.1	Estrutura do CAN 2.0A	28
3.2.2.2	Estrutura do CAN 2.0B	28
3.2.3	Camada Física	29
3.2.3.1	CAN <i>bit</i>	29
3.2.3.2	Temporização de um <i>bit</i>	29
3.2.3.3	Topologia do Barramento	32
3.2.4	Camada Ligação de Dados	33
3.2.4.1	Tipos de Tramas	33
3.2.4.2	Detecção de Erros	38
3.2.4.3	Gestão dos Erros	40
3.2.5	Camada Aplicação	42
4	Concepção e Implementação	45
4.1	Concepção	45
4.2	Implementação	48
4.2.1	Rede híbrida Xbee e CAN	48
4.2.2	<i>Bluetooth</i> e Aplicação para o Equipamento Móvel	50
4.2.3	GSM	51
4.2.4	Aplicação	52
4.2.5	<i>Hardware</i>	53
5	Testes e Resultados	57
5.1	Testes Individuais	57
5.1.1	Aplicação	57
5.1.2	CAN e GSM	61
5.1.3	Xbee	61
5.2	Cenário de Localização Hospitalar	66
6	Conclusões e Trabalho Futuro	69
	Referências bibliográficas	71
A	Anexo A1 - Esquemático da Estação Móvel	75
B	Anexo A2 - Esquemático da Estação Base	79

Índice de tabelas

3.1	Características do barramento.	33
5.1	Potências medidas.	67
5.2	Pontos extremos.	68

Índice de figuras

2.1	Componentes do sistema Horus (Youssef et al., 2003; Youssef and Agrawala, 2004)	7
2.2	Arquitetura do TraceMe (Silva et al., 2009a)	8
2.3	Modelo sensível ao contexto (Serôdio et al., 2010)	9
2.4	Diagrama de Blocos da Framework (Pahlavan et al., 2002)	10
2.5	Indicações em tempo real no PDA (Chang et al., 2008)	11
2.6	Distribuição dos nós de sensores na área de teste (Yan et al., 2008) .	12
3.1	Modelo OSI para IEEE 802.11	14
3.2	Comparação Modelo OSI e a <i>Stack</i> do IEEE 802.15.4	15
3.3	Formato dos pacotes transmitidos na camada física	16
3.4	Stack de protocolos do <i>Bluetooth</i> (Fonte: Bluetooth.org)	18
3.5	Arquitetura da rede GSM	21
3.6	Esquema do NSS (Hightower and Borriello, 2002)	22
3.7	Esquema do BSS (Hightower and Borriello, 2002)	23
3.8	Esquema do NMS e da rede GSM (Hightower and Borriello, 2002) . .	25
3.9	Diagrama de camadas do GSM	26
3.10	Formato da trama de dados	34

3.11	Diagrama de estado de erro	41
4.1	Esquema do sistema paciente	46
4.2	Esquema do sistema assistente	47
4.3	Infraestrutura	47
4.4	Esquema da rede	49
4.5	Aplicação: janela inicial	53
4.6	Aplicação: janela adição de pacientes	53
4.7	Visualização do esquemático (esquerda) e foto (direita), do nó de rede	54
4.8	Nó de rede com o módulo Xbee	55
4.9	Visualização do esquemático (esquerda) e foto (direita), da estação base	55
4.10	Visualização do esquemático (esquerda), e foto do terminal móvel (direita)	56
5.1	Aplicação: janela adição de pacientes preenchida	58
5.2	Aplicação: janela principal	59
5.3	Aplicação: janela de alarmes	59
5.4	Aplicação: alarmes listados	60
5.5	Aplicação: selecção de alarme	60
5.6	Aplicação: resolução de alarmes	61
5.7	Teste da comunicação CAN	62
5.8	Mensagem recebida no telemóvel	62
5.9	X-CTU software de configuração dos módulos Xbee	63
5.10	Mensagem transmitida/recebida pelos módulos Xbee	65
5.11	Cenário hospitalar	66
5.12	Projecto do cenário	67
A.1	Esquemático da Estação Móvel (Microcontrolador e driver CAN) . . .	75
A.2	Esquemático da Estação Móvel (Módulo Xbee)	76
A.3	Esquemático da Estação Móvel (Carregador de bateria e regulador) .	77
B.1	Esquemático da Estação Base	79

Acrónimos e abreviaturas

Lista de acrónimos e de abreviaturas

Sigla	Expansão
ACK	<i>Acknowledge</i>
AJAX	<i>Asynchronous JavaScript and XML</i>
BS	<i>Base Station</i>
CAN	<i>Controller Area Network</i>
CAN_H	<i>CAN_High</i>
CAN_L	<i>CAN_Low</i>
CiA	<i>CAN -in Automation</i>
CRC	<i>Cyclic Redundancy Check</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
DIP	<i>Digital Image Processing</i>
DLC	<i>Data Length Code</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
EOF	<i>End of Frame</i>
FDD	<i>Frequency Division Duplex</i>

Sigla	Expansão
FDMA	<i>Frequency-Division Multiple</i>
FHSS	<i>Frequency-Hopping Spread Spectrum</i>
G-FSK	<i>Gaussian Frequensy Shift Keying</i>
GSM	<i>Global System for Mobile Communications</i>
HCI	<i>Host Controller Interface</i>
ID	<i>Identifier</i>
IDE	<i>Identifier Extension</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IFS	<i>Inter Frame Space</i>
IrDA	<i>Infrared Data Association</i>
ISM	<i>Industrial, Scientific and Medical</i>
L2CAP	<i>Logical Link Control and Adaptation</i>
LAN	<i>Local Area Network</i>
LBS	<i>Location Based Service</i>
MAC	<i>Medium Access Control</i>
MAC	<i>Medium Access Control</i>
MAN	<i>Metropolitan Area Network</i>
MS	<i>Mobile Station</i>
MSC	<i>Mobile Switching Centre</i>
NRZ	<i>Non-Return to Zero</i>
NSS	<i>Network Switching Subsystem</i>
OSI	<i>Open Systems Interconnection</i>
OSI	<i>Open Systems Interconnection</i>
PC	<i>Personal Computer</i>
QoS	<i>Quality of Service</i>
RF	<i>Radio Frequency</i>
RFCOMM	<i>Radio Frequency COMM</i>

Sigla	Expansão
RFID	<i>Radio Frequency Identification</i>
RJW	<i>Resynchronization Jump Width</i>
RSS	<i>Received Signal Strength</i>
RSSI	<i>Received Signal Strength Indication</i>
RTR	<i>Remote Transmission Request</i>
SDP	<i>Service Discovery Protocol</i>
SOF	<i>Start of Frame</i>
SRR	<i>Substitute Remote Request</i>
T_q	<i>Time Quantum</i>
TDD	<i>Time Division Duplex</i>
TDMA	<i>Time Division Multiple Access</i>
TTL	<i>Transistor-Transistor Logic</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
WEP	<i>Wired Equivalent Privacy</i>
WPA	<i>Wi-Fi Protected Access</i>
WPAN	<i>Wireless Personal Area Network</i>
WSN	<i>Wireless Sensor Network</i>



Introdução

Com o avanço efectuado nas áreas da electrónica e geo-referenciação, muitos têm sido os projectos e serviços propostos baseados na localização.

Na sociedade moderna, muitos são os exemplos onde os serviços baseados na localização são utilizados, desde a publicidade contextualizada, à automação industrial e passando pelos sistemas de alarme e de saúde.

Desde à muito que existe uma incorporação de técnicas e tecnologias nos hospitais, a fim de satisfazerem a necessidade de melhores cuidados de saúde, bem como de auxiliar os técnicos nas suas tarefas. Os hospitais não são espaços fechados e a existência de espaços de lazer e de convívio em conjunto com a liberdade de movimentos dos pacientes, proporciona uma dificuldade acrescida aos técnicos de saúde, que é o de saber aonde se encontra determinado paciente. Este problema incentiva a criação de aplicações de cariz hospitalar baseados nos serviços LBS (*Localization Based Service*) a fim de poderem oferecer melhor informação e rapidez no apoio prestado pelos auxiliares.

1.1 Motivação

As novas exigências da sociedade de informação, associada aos avanços nas tecnologias de rede sem fios bem como novos paradigmas de computação permitiram o desenvolvimento de infra-estruturas sem fios para uso em localização. As aplicações baseadas em LBS podem ser aplicadas numa ampla gama de serviços de rastreamento *indoor*, *outdoor*, sistemas de alerta, automação, bem como em aplicações de apoio à vida e de saúde.

O Mundo e a assim sendo a Europa, já estão e irão enfrentar um grande desafio, devido à mudança demográfica. As populações nos países industrializados estão a envelhecer, ainda que com algumas diferenças de região para região. São inúmeros os factores que contribuíram para este facto, desde o despovoamento, redução da natalidade e o principal o aumento da esperança média de vida.

Este desafio provocado pela mudança demográfica na nossa sociedade, tem vindo a aumentar e assim continuará, por conseguinte assiste-se não só a um aumento das despesas dos regimes sociais, seja devido às pensões de reforma como também devido as despesas dos sistemas de saúde. Esta população devido à idade e ao facto de estar mais debilitada, perde muitas vezes a sua independência e necessita de apoio e acompanhamento no dia-a-dia, é por isso que existe uma necessidade dos governos e dos profissionais de saúde assegurarem a toda a população idosa o máximo de independência, bem como que ela se sinta activa e permaneça em boas condições de saúde (Costa, 2005; Eurostat, 2008).

Nos últimos anos o número de internamentos nos hospitais tem vindo a aumentar e o número de mortes de pessoas internadas também tem subido, tendo também em conta que redução nos orçamentos hospitalares, iremos ver uma cada vez maior deficiência no número dos assistentes de saúde. As pessoas internadas nos hospitais estão, devido à condição associada, debilitadas, logo existe sempre a possibilidade de recaídas, infecções hospitalares, existe assim uma possibilidade de que os meios tecnológicos possam ajudar e apoiar, tanto a população que necessite de cuidados de saúde bem como os assistentes de saúde. Trazendo deste modo melhor qualidade

de vida aos idosos e toda a população que venha a usufruir destes meios e manter um maior equilíbrio entre o número de auxiliares e a população que necessita de cuidados (Marques, 2011).

1.2 Objectivos

Pretende-se com este trabalho desenvolver uma aplicação médica que irá fornecer um serviço de alerta e localização dentro de portas de doentes internados nos serviços hospitalares. Os objectivos podem ser divididos em dois tipos, direccionados aos pacientes e os direccionados aos assistentes e instituições.

A aplicação irá ser constituída por uma rede de nós IEEE 802.15.4 interligados por uma rede CAN, por um terminal móvel com capacidade sensorial e de controlo, e ainda um terminal fixo com possibilidade de comunicação por *Bluetooth* e GSM. O terminal móvel fará o processamento da sua localização através da potência do sinal dos nós da rede e saberá a informação dos sensores acoplados, enviando quando necessário a informação ao terminal fixo. O terminal fixo por sua vez, irá gerir os utentes e os alarmes dos mesmos.

Os utilizadores irão usufruir de uma aplicação que irá não só alertar os auxiliares médicos se algo de anormal acontecer, mas também fornecer a sua localização, para uma mais rápida assistência. Isto fará com que a liberdade de movimentos seja maior (dentro do "hospital") dando-lhe maior independência.

No caso dos assistentes terão uma ferramenta que os auxilia e lhes dará mais informação acerca dos doentes que necessitam de maior vigilância, por isso mesmo liberta os assistentes para que possam ocupar-se de mais doentes, sem perder a qualidade dos serviços prestados, pois podem estar a monitorizar mais pacientes ao mesmo tempo.

1.3 Organização da Dissertação

Esta dissertação está estruturada em cinco capítulos. O presente capítulo é feita a introdução e é exposta a motivação e objectivos do trabalho.

O segundo capítulo descreve as diferentes técnicas de localização. De seguida são apresentados vários sistemas de localização em ambiente *indoor*, mostrando os principais avanços nesta área.

O enquadramento tecnológico é feito no capítulo 3 e são descritos os meios tecnológicos necessários para a realização deste sistema.

No capítulo 4, é mostrado a concepção do sistema e onde é feita a descrição da utilidade de cada bloco, descreve-se ainda a implementação do sistema e todos os componentes que o constituem.

Os testes e resultados da implementação do sistema estão descritos no capítulo 5, finalizando-se com o capítulo da conclusão e trabalho futuro.

2

Sistemas de Localização *Indoor*

Neste capítulo são descritas as diferentes técnicas de localização existentes com o potencial de aplicação ao sistema proposto. Finalizando o capítulo com a apresentação de vários projectos, encontrados na literatura, que se enquadram nesta área.

2.1 Técnicas de Localização

Os dados necessários para a obtenção da localização de um determinado objecto podem ser conseguidos de diferentes formas. Os vários métodos existentes para a localização estão divididos em três grandes grupos e estes divididos em vários subgrupos, como se pode ver em seguida:

Triangulação: Este método utiliza as propriedades geométricas dos triângulos para obter a localização de um objecto. A Triangulação está dividida em duas subcategorias, a lateração e a angulação. Sendo que lateração é o termo usado para medições de distâncias enquanto angulação é usado para os ângulos (W.C. Xuesong and Ming, 2008).

- **Lateração:** A lateração calcula a posição de um objecto através da distância entre vários pontos de referência cuja posição é conhecida.
- **Angulação:** Na angulação a posição é calculada com o ângulo de incidência entre o objecto e os pontos de referência com posição conhecida.

Análise de Padrões: Neste método são utilizados técnicas de análise de padrões, para o cálculo da localização ([Hightower and Borriello, 2001](#)).

- ***Fingerprinting*:** É criada uma imagem electromagnética de um local podendo-se obter a localização.
- **Análise de Imagens:** Através de várias imagens de um local e recorrendo a algoritmos de PDI (Processamento Digital de Imagem), faz-se a localização do objecto.

Proximidade: Esta técnica determina o grau de proximidade de um objecto por descobrir ou identificar em relação a ao ponto de referência mais próximo e que é conhecido. Existem três subgrupos para a localização através desta técnica, são elas ([Hightower and Borriello, 2001](#); [Gu et al., 2009](#)):

- **Contacto Físico:** Este é o método mais básico, detecta a proximidade do objecto através do toque em diversos tipos de sensores.
- **Célula de Origem:** É obtida a localização sabendo a que célula o objecto pertence, isto é conseguido através da monitorização da qualidade do sinal de vários pontos de acesso.
- **Cartões de Identificação Automáticos:** A implementação desta técnica exige que vários terminais vão registando o uso do cartão de identificação e assim poder-se estimar a localização.

2.2 Projectos Existentes

Sistema Horus

O sistema Horus, ver Figura 2.1, oferece um agrupamento de técnicas para estimativa da localização, que utiliza um método probabilístico. O sistema foi implementado num contexto de redes sem fio IEEE 802.11. Cada candidata a coordenada de localização, é considerada como classe ou categoria. A fim de minimizar os erros da distância, a localização é escolhida enquanto a sua probabilidade for a maior. Os resultados obtidos por esta técnica, mostraram uma precisão de mais do que 90% dentro de 2,1 metros, aumentando o número de amostras do local de estudo, podem-se obter melhorias (Youssef et al., 2003; Youssef and Agrawala, 2004).

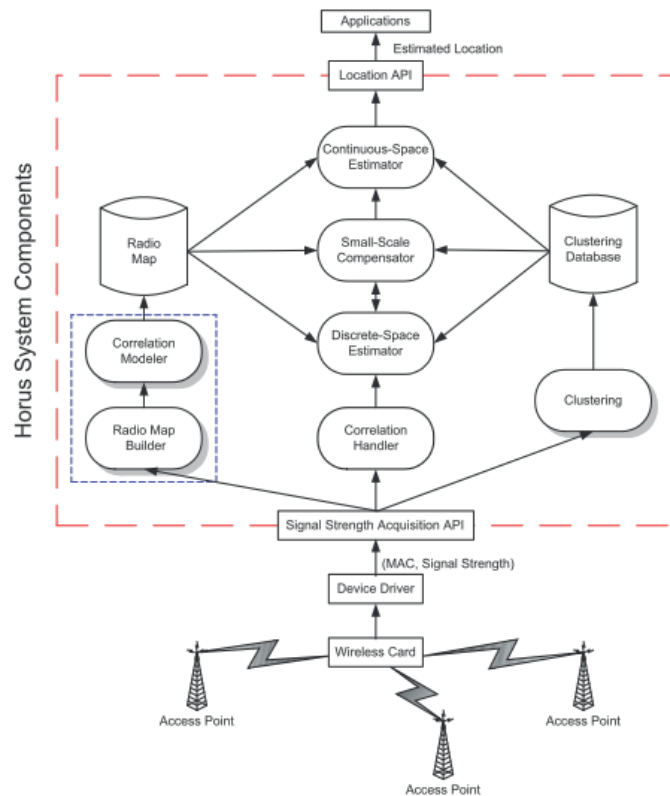


Figura 2.1 – Componentes do sistema Horus (Youssef et al., 2003; Youssef and Agrawala, 2004)

TraceMe - *Indoor Real-Time Location System*

O TraceMe é um sistema de localização de tempo real, em ambientes *indoor* utilizando a tecnologia RFID, JDBC, JSP e *Reverse AJAX*. Para além da localização é capaz de fornecer controlo de acesso a pessoas e objectos, ainda gera alarmes devido a falhas de comunicação, violação de acessos ou violação do cartão. Cada utilizador ou objecto está associado a uma *tag* que utiliza comunicações de curto e longo alcance que funcionam a frequências diferentes, em conjunto com dois algoritmos, para fazer a localização dos mesmos. Este sistema foi usado em ambiente hospitalar, a sua arquitectura pode ser observada na Figura 2.2 (Silva et al., 2009a).

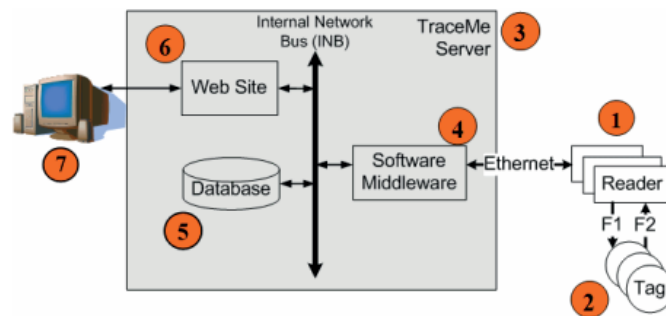


Figura 2.2 – Arquitectura do TraceMe (Silva et al., 2009a)

Pervasive WSN based solutions applied to Health and Life-support Systems

Descreve uma solução LBS para a saúde e suporte à vida, sensível ao contexto, ver Figura 2.3, com base na integração e fusão de sensores e tecnologias sem fios. Foi proposto a integração dispositivos e protocolos IEEE 802.15.4, RFID, *Bluetooth*, 3G e CAN para fornecer sinais biométricos e alarmes. A infraestrutura combina nós fixos com nós moveis, sendo os moveis o alvo de localização. Um algoritmo utiliza RSS, comparando-o com sinais previamente medidos, e combina-o com os dados das tags RFID. Os sinais de alarme e as ligações de comunicação são sensíveis ao contexto no ambiente físico. Como o sistema é sensível ao contexto foram analisadas

várias soluções de localização *indoor*, baseados em modelos de propagação e uma nova abordagem pelo paradigma de "proximidade" e redes oportunistas. Foram realizados vários testes e os resultados demonstram que o sistema é viável, adaptável a diferentes cenários e compatível com aplicações de cuidado de saúde (Serôdio et al., 2010).

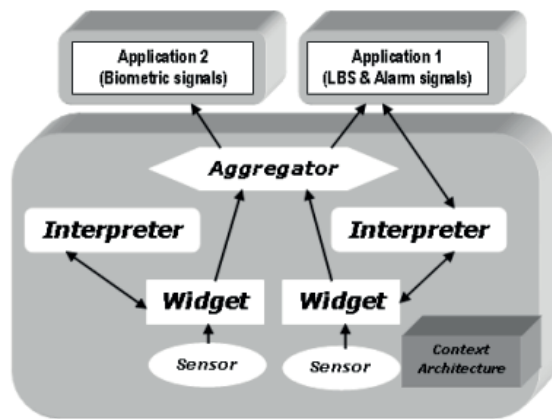


Figura 2.3 – Modelo sensível ao contexto (Serôdio et al., 2010)

A Multi-Technology Framework for LBS using Fingerprinting

Devido aos desafios da computação Ubíqua, a estimativa da localização automática de um dispositivo móvel em redes sem fios é cada vez mais importante. Dada a complexidade dos ambientes *indoor*, devido a factores como a interferência multi-caminho, reflexão e refacções é utilizado o método de análise por *fingerprinting*. Um modelo com várias tecnologias e para várias plataformas foi apresentado. Depois deste modelo uma *Framework* para LBS foi implementada e o diagrama está representado na Figura 2.4. Ele pode ser usado para análise da performance do cálculo estimativo da localização, tendo sido desenvolvido em vários níveis de abstracção. Este modelo é destinado a ambientes *indoor*, tendo sido apresentados e testados vários cenários analisados pelo método *fingerprinting* e através de vários algoritmos de localização. A *Framework* foi testada usando os padrões IEEE 802.11 e IEEE 802.15.4 com a estimativa da localização em tempo real, tendo uma precisão 1 metro, tendo alcançado uma taxa de erro de 10 % (Silva et al., 2009b).

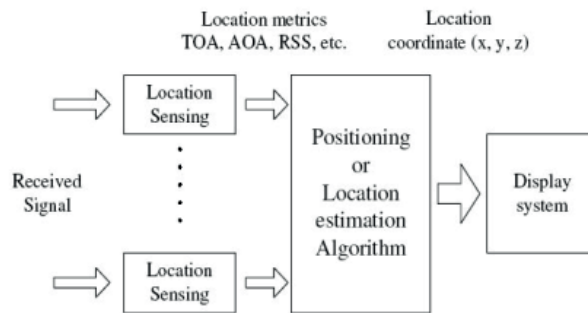


Figura 2.4 – Diagrama de Blocos da Framework (Pahlavan et al., 2002)

A Novel Indoor Wayfinding System Based on Passive RFID for Individuals with Cognitive Impairments

É proposto neste sistema, que tem o objectivo de aumentar a área de trabalho e a independência de pessoas com deficiências cognitivas, uma forma de localização e auxílio à orientação. Para isso baseou-se em tags RFID, com coordenadas, que são colocadas nos locais onde os pacientes têm de tomar decisões quanto ao caminho que vão escolher seguir, como virar à esquerda ou à direita. Os pacientes têm de transportar um PDA (ver Figura 2.5), com leitor RFID incorporado, de modo a ler as tags passivas, e assim fornecer indicações em tempo real, de modo a orientar os pacientes. Em conjunto com as indicações também é apresentado a foto do local. Os locais visitados são monitorizados e guardados e em caso de anomalias são emitidos alertas. O protótipo pode ter problemas com leitor de *tags*, devido ao efeito sombra; e à escalabilidade do sistema ser limitada, porque necessita da associação com a *tag* local. No entanto pode auxiliar os pacientes, num ambiente hospitalar, a encontrar um determinado local. (Chang et al., 2008).

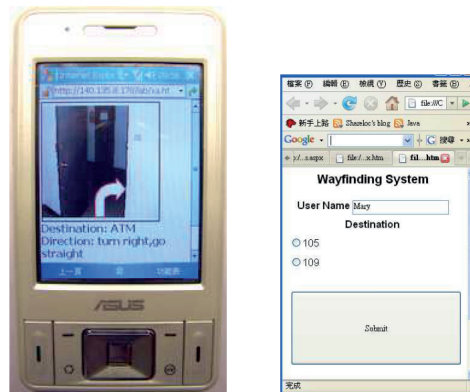


Figura 2.5 – Indicações em tempo real no PDA (Chang et al., 2008)

Location-based Services for Elderly and Disabled People

O ZUPS é um sistema de localização baseado nas tecnologias ZigBee e ultra-som, que fornece uma cobertura multi-célula com a possibilidade de ser alargada. Para além da tecnologia de ultra-som, o ZigBee foi escolhido por causa de interoperabilidade com outros aplicativos de inteligência ambiente, como por exemplo a domótica. O sistema utiliza as duas tecnologias para medir a distância entre os dispositivos móveis (*tags*) e as *beacons* com a localização conhecida; no entanto, ele utiliza os métodos de localização por proximidade e multi-lateration simultaneamente. A coexistência de ZigBee e ultra-som, diminui a necessidade, na infra-estrutura, dos sistemas de ultra-som a serem instalados. Embora seja o alerta, o seu principal objectivo, caso ocorra um alarme; o sistema pode fornecer outras funções, tais como a orientação e treino de orientação espacial dentro de grandes edifícios para idosos e pessoas com deficiência. (Marco et al., 2008).

An Experimental Study on Home-Wireless Passive Positioning

Neste estudo é proposto um técnica chamada posicionamento passivo, que faz uso das pessoas como um objecto de espalhamento de rádio e o efeito na intensidade do sinal recebido (RSS) dos nós de sensores fixos, podemos observar a distribuição dos nós na Figura 2.6. A localização do objecto móvel é estimada pela alteração do RSS

de um conjunto de nós. Se alguns dos parâmetros do RSS, tais como amplitude, média local ou variação local das ligações mudar mais do que um determinado limiar, o objecto ou a pessoa podem ser detectados, e a posição pode ser grosseiramente determinada de acordo com a variável RSS e pelo cruzamento das correlações. Estes tipos de sistema são mais aconselhados quando a informação de presença é suficiente, pois a obtenção da posição exacta em ambientes ruidosos é extremamente difícil e propenso a erros. (Yan et al., 2008).

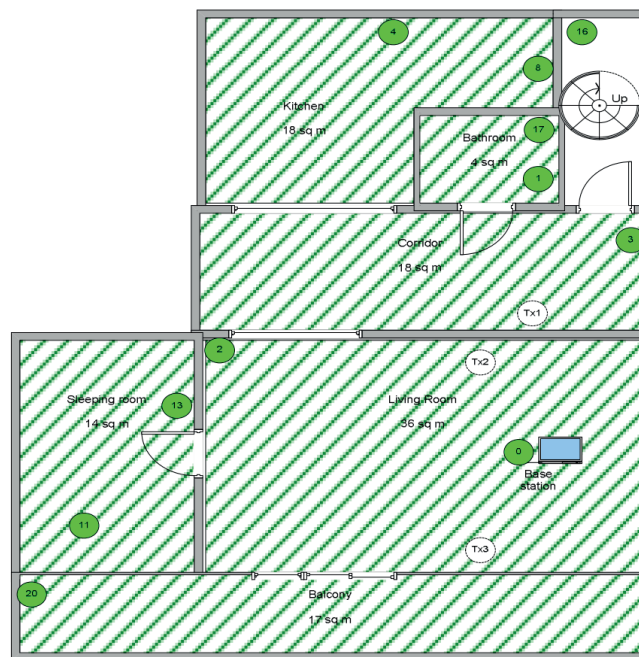


Figura 2.6 – Distribuição dos nós de sensores na área de teste (Yan et al., 2008)



Enquadramento Tecnológico

Neste capítulo são descritos os protocolos e as tecnologias utilizadas para a realização deste tipo de sistemas de localização, monitorização sensorial e controlo remoto. E de que modo se inserem no sistema proposto.

3.1 Redes sem Fios

De entre os padrões definidos pelo IEEE (*Institute of Electrical and Electronics Engineers*) para os protocolos de comunicação em rede, está o grupo 802, que é um padrão do IEEE para as redes locais (LAN) e metropolitanas (MAN). Este grupo contém ainda outros subgrupos que especificam as redes que utilizam comunicação sem fio (*wireless*) (IEEE, 2001).

3.1.1 Redes Locais (WLAN) - IEEE 802.11

A norma IEEE 802.11 foi a primeira norma para as redes sem fios adoptada pelo IEEE. Nela são definidas a camada de acesso ao meio (MAC - *Medium Access Control*) e também a camada física (PHY - *Physical Layer*) do modelo OSI. O objectivo

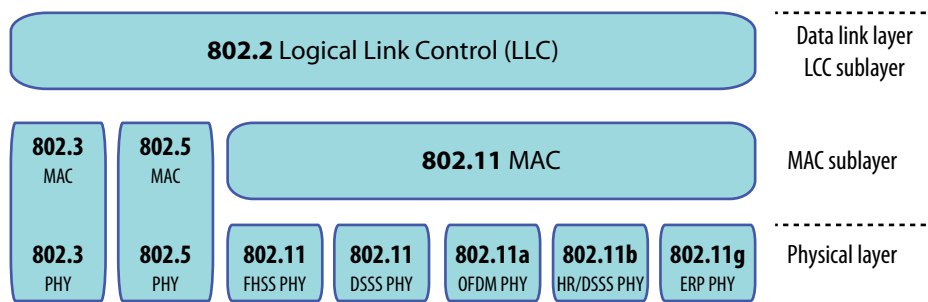


Figura 3.1 – Modelo OSI para IEEE 802.11

desta tecnologia foi acabar com a dependência do uso de cablagem de cobre em redes de computadores e na transmissão de dados de alta velocidade, utilizando como meio de propagação ar livre (IEEE, 2007). Na Figura 3.1 pode-se observar a relação entre que o modelo OSI e a norma IEEE 802.11.

Camada Física: O padrão IEEE 802.11 define na camada mais baixa várias implementações diferentes, as mais importantes são as IEEE 802.11a, IEEE 802.11b e IEEE 802.11g. O protocolo IEEE 802.11a opera na faixa dos 5GHz, com uma taxa de transmissão máxima de 54Mbit/s, o IEEE 802.11b opera na faixa dos 2.4GHz, com uma taxa de transmissão de 11Mbit/s e por ultimo o IEEE 802.11g usa a faixa dos 2.4GHz e uma taxa de transmissão de 54Mbit/s (IEEE, 2007).

Subcamada MAC: É responsabilidade desta camada garantir que os dados cheguem de forma legível às camadas superiores e também pelo acesso ao meio, neste caso utilizando a técnica CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), para tornar o meio mais seguro utiliza algoritmos de segurança WEP (*Wired Equivalent Privacy*) ou WPA (*Wi-Fi Protected Access*)(IEEE, 2007).

3.1.2 Redes de Pessoais (WPAN) - IEEE 802.15.4 e *Bluetooth*

- IEEE 802.15.4

O standard IEEE 802.15.4 especifica as duas camadas inferiores do modelo OSI,

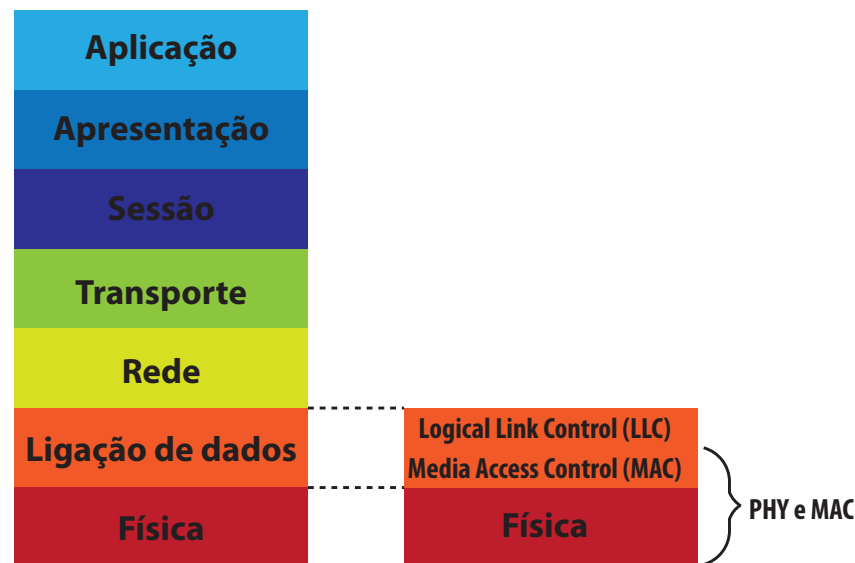


Figura 3.2 – Comparação Modelo OSI e a *Stack* do IEEE 802.15.4

a camada (PHY- *Physical Layer*) e camada de Ligação de dados constituída pela subcamada Controlo de acesso ao meio (MAC - *Medium Access Control*), como se pode ver pela Figura 3.2 (IEEE, 2003).

Este *standard* é utilizado em aplicações que necessitam de transmissão sem fios, de baixa taxa de transmissão de dados e um baixo consumo energético, podendo ser aplicados em vários tipos de cenários, como o industrial, agrícola, transportes, residências e os que fornecem assistência à saúde.

O IEEE 802.15.4 opera na banda ISM (*Industrial, Scientific and Medical*), é um protocolo mais simples que outros *standards*, para transferência de dados para as redes sem fios.

O acesso ao meio é feito através da técnica CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) e *time slotting opcional*. Possui ainda reconhecimento de mensagem e uma estrutura para sinalização, chamada *beacon*.

O *standard* IEEE 802.15.4 pode operar em três bandas diferentes: 868MHz, 915MHz e 2.4GHz. A operar a 868Mhz tem uma taxa máxima de transferência de 20Kbit/s, a 915MHz terá uma taxa de 40Kbit/s e a 2.4GHz terá a taxa máxima de 250Kbit/s.

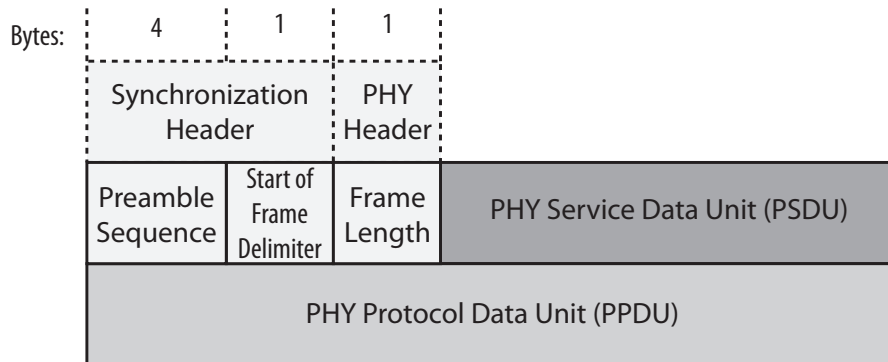


Figura 3.3 – Formato dos pacotes transmitidos na camada física

Camada Física: Nesta camada são definidas as características físicas da comunicação sem fios, a forma como irá transmitir e receber os dados através de ondas electromagnéticas. A técnica de modulação dos sinais é denominada por DSSS (*Direct Sequence Spread Spectrum*). Esta técnica exige menor potência, o que permite ter dispositivos de rede com baixa complexidade e custo. De acordo com o padrão a estrutura do pacote a ser transferido desta camada PPDU terá o formato da Figura 3.3. É então constituído por um cabeçalho de sincronização (SHR); um cabeçalho de camada física, que contém o tamanho da *frame* e por último uma variável (PSDU).

Subcamada MAC: Nesta subcamada é feita a interface entre a camada física e a de aplicação. É responsável pela prevenção de colisão através da técnica CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Existem dois tipos de configuração, a *Beacon* e *nonBeacon*. Em *nonBeacon*, os dispositivos depois de associados à rede podem estabelecer comunicação um com o outro a qualquer momento. Numa rede configurada como *Beacon*, é o coordenador de rede que transmite periodicamente uma *frame, Beacon*, para os dispositivos sincronizarem e determina o envio das mensagens.

• **BLUETOOTH**

O *Bluetooth* é uma tecnologia para comunicação sem fios entre dispositivos de diferentes fabricantes, normalizada pelo padrão IEEE 802.15.1. Esta tecnologia, está

em constante evolução muito devido ao seu grande êxito. Entre as principais características está o seu curto alcance (máximo 100m), baixa taxa de transmissão de dados (1-3 Mbit/s), baixo consumo de energia e de baixo custo. Devido ao seu principal objectivo, as suas características técnicas são fornecer aos utilizadores a transmissão de voz e dados em tempo real entre dispositivos, substituído o uso de cablagem e o IrDA (*Infrared Data Association*) (IEEE, 2002).

Quanto ao tipo de utilização o *Bluetooth* foi dividido em três classes, com diferentes alcances e potências:

Classe 1: alcance máximo de 100 metros e uma potência máxima 100 mW;

Classe 2: alcance máximo de 10 metros e uma potência máxima 10 mW;

Classe 3: alcance máximo de 1 metros e uma potência máxima 1 mW.

A arquitectura do protocolo consiste numa série de camadas que constituem uma *stack* (pilha). Na Figura 3.4 podemos ver as quatro camadas mais baixas do protocolo.

Os protocolos do núcleo do sistema são o protocolo Rádio Frequência (RF), *Link Controller* (LC), *Link Manager* (LM), *Logical Link Control and Adaptation* (L2CAP), todos eles estão completamente definidos em partes subsequentes da especificação *Bluetooth*. Para além destes, o *Service Discovery Protocol* (SDP) é o protocolo da camada de serviço necessária para todas as aplicações *Bluetooth*.

Camada Rádio: corresponde à camada física do modelo OSI, especifica a banda de frequências, a obtenção do canal, as características do transmissor e a potência e modulação do sinal. Operando na banda ISM (*Industrial, Scientific and Medical*) sendo constituído por 79 canais RF com espaçamento de 1MHz. A modulação do sinal é a G-FSK (*Gaussian Frequency Shift Keying*), com FHSS (*Frequency-Hopping Spread Spectrum*) como método de transmissão do sinal e para fornecer *full-duplex* usa TDD (*Time Division Duplex*).

Camada Baseband: define o acesso ao meio e os procedimentos da camada física

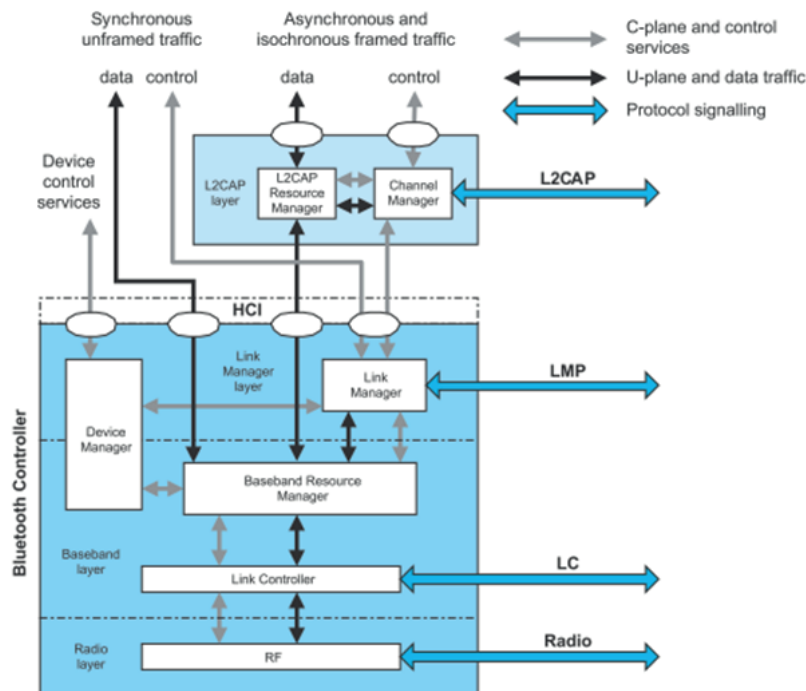


Figura 3.4 – Stack de protocolos do *Bluetooth* (Fonte: Bluetooth.org)

entre diferentes dispositivos. Esta camada é responsável por todas as operações de dados digitais e controla toda a operação rádio, de modo a estabelecer uma ligação RF com outros dispositivos *Bluetooth*.

Camada *Link Manager*: essencialmente, esta camada é responsável pelos aspectos de segurança, como autenticação e encriptação, a configuração e gestão das ligações entre dispositivos. Através do protocolo de ligação (LMP), o gestor de ligação do *Bluetooth* comunica com o gestor de ligação de outros dispositivos.

HCI: A *Host Controller Interface* não é uma camada, pois serve só de interface comum entre as camadas superiores e as inferiores orientadas ao *Hardware*.

Camada L2CAP: O *Logical Link Control and Adaptation Protocol* é um protocolo de ligação de dados simples que adapta os protocolos da camada superior às camadas inferiores, fornecendo tanto serviços *connectionless* como serviços orientados à ligação e sinalização de canais. Faz a multiplexagem entre diferentes protocolos

das camadas superiores, permitindo que partilhem ligações com as camadas inferiores. Este protocolo é responsável pela segmentação e reassemblagem que permite a transferência de grandes pacotes para o suporte das camadas inferiores. Faz a gestão de grupos, providenciando uma transmissão única para um grupo de outros dispositivos *Bluetooth*. Lida, ainda, com requisitos de qualidade de serviço, ou seja, a especificação de QoS por canal.

Existem ainda as camadas superiores que apresentam diversos perfis, que podem ser escolhidos consoante o uso que é pretendido.

SDP: O *Service Discovery Protocol* é um protocolo que segue um modelo Cliente/Servidor e fornece um meio para as aplicações descobrirem os serviços disponíveis.

TCS or TCP: O *telephony control protocol specification or telephony control protocol* é responsável pelo controlo das chamadas, gestão de grupo e funciona como *gateway* entre telemóveis e telefones fixos.

RFCOMM: É um protocolo de transporte simples e fiável que permite a emulação da porta série, o RFCOMM conecta-se com as camadas inferiores através da camada L2CAP. Suporta múltiplas portas virtuais sobre um único canal físico entre dois dispositivos (60 conexões simultâneas entre dois dispositivos *Bluetooth*).

HID: A *Human Interface Device* fornece suporte para dispositivos como teclados, ratos, *joysticks*. Suporta ainda *Plug and Play*.

OBEX: O *Object Exchange* é um protocolo de comunicação que permite a troca de objectos.

BNEP: O *Bluetooth Network Encapsulation Protocol* é usado para transportar protocolos de rede comuns sobre *Bluetooth*, tais como o IPv4 e o IPv6. O formato do pacote é baseado no *EthernetII/DIX Framing*, conforme definido pelo IEEE 802.3 (funciona directamente sobre L2CAP). O BNEP é usado nas redes de área pessoal (PAN).

3.1.3 GSM

Uma chamada entre duas pessoas, é o serviço básico das redes móveis (GSM), mas para fornecer este serviço a rede tem de ser capaz de configurar e manter a chamada, envolve uma série de tarefas, como identificar a chamada, determinar a localização, encaminhar a chamada e assegurar que a chamada não cai a meio da conversa. E no fim cobrar pelo serviço que usou.

Numa rede de telefones fixos, o processo de fornecer e gerir as chamadas é mais simples, porque os telefones estão ligados por fio à rede e a sua localização permanece inalterada. Numa rede móvel, uma chamada é uma tarefa muito mais complexa, como a possibilidade dos utilizadores moverem-se livremente numa rede sem fios, desde que fiquem dentro da área de cobertura. Na prática, a rede tem de encontrar soluções para vários problemas antes de qualquer chamada seja feita com sucesso. O subscritor do serviço tem de ser encontrado e identificado para fornecer-lhe os serviços pedidos. A forma a entender-se como os subscritores são servidos, é necessário identificar as principais interfaces, subsistemas e elementos presentes na rede GSM, bem como as funções que desempenham.

Um dos objectivos da especificação GSM é definir várias interfaces abertas, de modo a que, qualquer que seja o fornecedor dos elementos de rede, eles vão operar entre si. Quando uma interface é aberta, define estritamente o que acontece através dela, e por sua vez também define o tipo de acções, procedimentos e funções que têm de ser implementadas entre as interfaces.

A especificação GSM define duas interfaces abertas dentro das redes GSM. A primeira é entre a MS (*Mobile Station*) e a BS (*Base Station*). Esta interface tem o nome de "*air interface*" ou seja a interface de rádio. É fácil de entender porque a interface é aberta, como temos dispositivos móveis de diferentes marcas que têm de ser capazes de comunicar com todos os diferentes fornecedores de redes GSM. A segunda interface encontra-se entre a MSC (*Mobile Switching Centre*) e a (*Base Station Controller*). Esta interface tem o nome de "*A-interface*". O sistema inclui mais duas interfaces, mas elas não são totalmente abertas ([Hightower and Borriello](#),

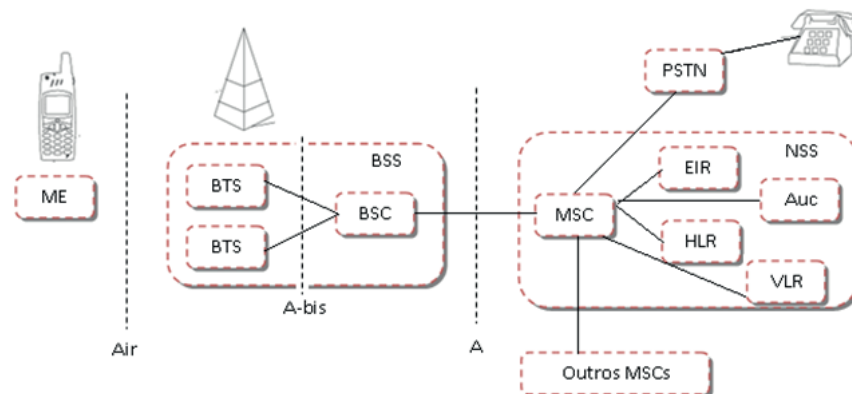


Figura 3.5 – Arquitectura da rede GSM

2002).

A rede GSM é descentralizada e a sua implementação pode ser dividida em três subsistemas, ver Figura 3.5:

- **NSS** (*Network Switching Subsystem*)
- **BSS** (*Base Station Subsystem*)
- **NMS** (*Network Management Subsystem*)

A rede actual precisa, de modo a estabelecer uma chamada, da NSS e da BSS. A BSS é responsável pelo controlo rádio e cada chamada liga-se através da BSS. A NSS toma conta das funções de controlo da chamada. As chamadas são sempre ligadas ao, e através do, NSS (Schiler, 2003).

Mobile Station

A MS é uma combinação entre um equipamento terminal e os dados do subscritor. O terminal chamado de ME (Mobile Equipment), e os dados subscritor armazenados num módulo separado chamado SIM (Subscriber Identify Module).

Temos então, $ME + SIM = MS$. O SIM é um dispositivo de memória montado num pequeno cartão, guardando informação específica do utilizador. O cartão SIM

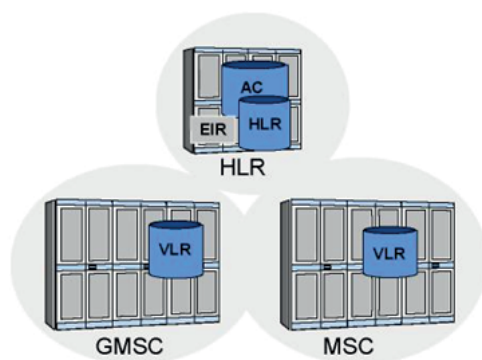


Figura 3.6 – Esquema do NSS (Hightower and Borriello, 2002)

pode ser inserido/retirado de qualquer ME, e para além de conter o número de identificação do utilizador e a lista de redes disponíveis, contém ferramentas para a autenticação e de encriptação [ETSI \(1996\)](#).

Os subsistemas da rede GSM e os diferentes elementos constituintes, são:

Network Switching Subsystem

O NSS (*Network Switching Subsystem*) contém os seguintes elementos de rede: MSC, VLR, HLR, AC e EIR, ver [Figura 3.6 \(Schiler, 2003\)](#).

As funções do Subsistema NSS, são:

- **Controlo das chamadas:** Identifica o subscritor, estabelece a chamada e liberta a ligação após cessar a chamada.
- **Tarifação:** Este recolhe a informação de cobrança acerca das chamadas (a quem ligou, o tempo, tipo de serviço utilizado) e transfere a informação para o centro de facturação.
- **Gestão de mobilidade:** Este mantém a informação acerca da localização do subscritor.
- **Sinalização:** Aplica-se à interface entre BSS e PSTN.

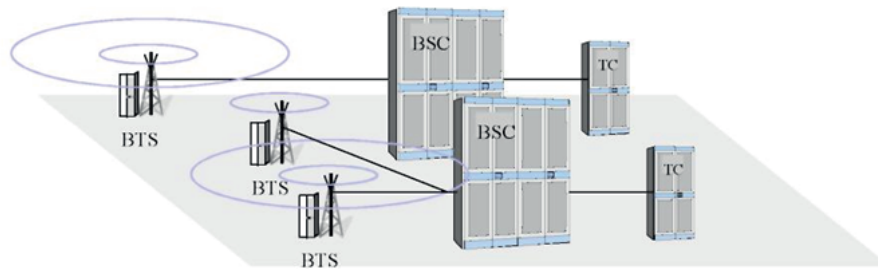


Figura 3.7 – Esquema do BSS (Hightower and Borriello, 2002)

- **Tratamento dos dados do subscritor:** O armazenamento dos dados permanentes é feito no HLR e o armazenamento dos dados temporários de dados relevantes é feito no VLR.

Base Station Subsystem

O BSS (*Base Station Subsystem*) é responsável pela gestão da rede rádio, e é controlado pelo MSC. Normalmente, um MSC contém vários BSSs. O BSS em si pode cobrir uma área geográfica consideravelmente grande constituído por muitas células (uma célula refere-se à área coberta por uma ou mais frequências). O BSS é composto pelos seguintes elementos, e aonde têm as ligações observadas na Figura 3.7 (Schiler, 2003):

- **BSC** (*Base Station Controller*)
- **BTS** (*Base Transceiver Station*)
- **TC** (Transcoder)

Algumas das tarefas mais importantes do BSS:

- **Controlo dos recursos Rádio:** Na rede GSM, o BSS é o elemento que trata dos recursos de rádio, ou seja, a alocação do canal de rádio e a qualidade da ligação.

- **Sincronização:** O BSS usa sincronização hierárquico, o que quer dizer que o MSC sincroniza o BSC e o BSC sincroniza o BTS associados a ele. A sincronização é um assunto crítico numa rede GSM devido à natureza da informação transferida. Se o sincronismo não funcionar correctamente, chamadas podem ser desligadas ou a qualidade pode ser prejudicada. Em última instância a chamada pode nem ocorrer.
- **Sinalização *Interface Air* e *A*:** A fim de estabelecer uma chamada, a MS deve conectar-se através do BSS.
- **Estabelecer uma ligação entre a MS e o NSS:** O BSS está localizado entre duas interfaces, a air interface e a A interface. A MS deve ter feito ligação através destas duas interfaces antes de conseguir estabelecer a chamada. De um modo geral, essa ligação pode ser apenas de sinalização ou de tráfego.
- **Gestão da mobilidade e *transcoding* da voz:** BSS gere a mobilidade e abrange vários casos de *handovers*.

Network Management Subsystem

O NMS (*Network Management Subsystem*) tem o objectivo de monitorizar várias funções e elementos da rede, como se observa na Figura 3.8.

As estações de trabalho dos operadores estão ligadas à base de dados e aos servidores através de uma LAN (*Local Area Network*). A base de dados guarda as informações de acerca da rede. Os servidores tomam conta da comunicação entre o NMS e os equipamentos da rede GSM. Estas comunicações são realizadas através de uma DCN (*Data Communications Network*), que se liga ao NMS via router. O DCN é normalmente implementado usando X.25 *Packet Switching Network* (Schiler, 2003; Rappaport, 1996).

As funções do NMS podem ser divididas em três categorias:

- Gestão de falhas;

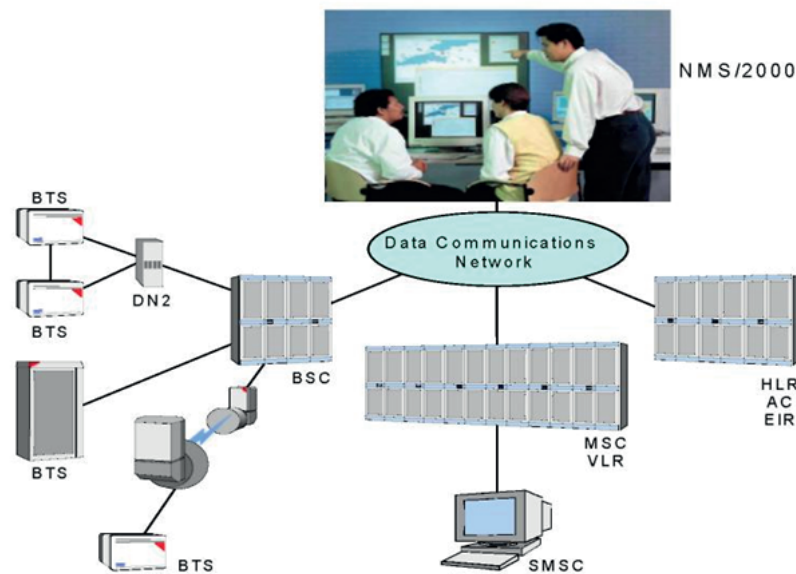


Figura 3.8 – Esquema do NMS e da rede GSM (Hightower and Borriello, 2002)

- Gestão de configuração;
- Gestão de desempenho.

Estas funções abrangem a totalidade dos elementos da rede GSM, começando pelos individuais BTSs até aos MSCs e HLRs.

Gestão de falhas: O objectivo da gestão de falhas é garantir o bom funcionamento da rede e a rápida correcção dos erros que são detectados. Fornece aos operadores da rede a informação acerca do estado dos alarmes e mantém um histórico deles. Os alarmes são armazenados numa base de dados da NMS e esta base de dados pode ser pesquisada com critérios definidos pelo operador da rede.

Gestão de Configuração: O objectivo da gestão de configuração é manter actualizado as informações acerca das operações e o estado das configurações dos elementos de rede. Funções específicas de configuração incluem-se a configuração da rede rádio, gestão de *software* e *hardware* dos elementos de rede, sincronização do relógio e operações de segurança.

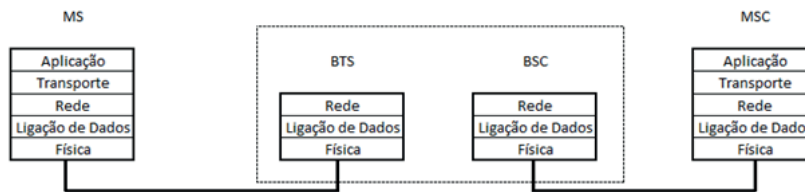


Figura 3.9 – Diagrama de camadas do GSM

Gestão de desempenho: Na gestão de desempenho, a NMS recolhe os dados de cada elemento da rede e armazena na base de dados. Com base nesses dados, o operador da rede é capaz de comparar o desempenho real da rede com o planeado.

A rede GSM uma pilha de 5 camadas, ver Figura 3.9, cada camada tem funções específicas:

- **Camada da Aplicação:** É formada pelos serviços e aplicações de alto nível como por exemplo a caixa de entrada e de saída de mensagens de texto, registo de chamadas não atendidas entre outras.
- **Camada de Transporte:** Tem a responsabilidade de resolver os problemas de congestionamento e controlo de fluxo.
- **Camada de Rede:** A sua função é o endereçamento, o encaminhamento dos pacotes, efectuar o *handover* e a localização recorrendo a técnicas de Célula de Origem.
- **Camada de Ligação de Dados:** Tem as funções de autenticação, acesso ao meio e multiplexagem.
- **Camada Física:** A rede GSM opera em duas bandas: a de 900MHz e a de 1800MHz, utilizando como método de transmissão do sinal uma combinação entre TDMA (*Time Division Multiple Access*) e FDMA (*Frequency-Division Multiple Access*). A comunicação em modo *full-duplex* é feita usando a técnica FDD (*Frequency Division Duplex*). A técnica de modulação usada é a MSK (*Minimum-shift keying*).

3.2 CAN

O CAN (*Controller Area Network*) é um protocolo de comunicação série, que suporta eficientemente controlo distribuído em tempo real com um grande nível de segurança. Este protocolo foi um dos primeiros e mais duradouros a ser usado nos automóveis, aonde podem existir várias redes CAN em funcionamento, com uma velocidade máxima de 1Mbit/s. As redes CAN nos automóveis servem para a gestão de todo o controlo electrónico do veículo, tais como a subida e descida dos vidros, o estado dos travões, gestão do motor, etc. A característica principal do CAN é a sua segurança e alta fiabilidade, por isso é que é utilizada no ramo automóvel.

3.2.1 Modelo OSI e CAN

O protocolo CAN apenas define três das camadas do modelo OSI, são elas a camada física, camada ligação de dados e a camada aplicação. Todas as outras camadas não são implementadas pelo protocolo, a camada aplicação inicialmente também não fazia parte, mas devido a solicitações da indústria esta viria a ser adicionada. Várias empresas desenvolveram *software* para a camada de aplicação, mas qualquer programador poderá criar a parte de aplicação, visto que as duas camadas inferiores são já implementadas nos microcontroladores (Corrigan, 2008).

3.2.2 Estrutura do CAN

A estrutura do CAN é composta por três camadas, anteriormente descritas. A camada Física, é a camada que define como o *hardware* deve converter as mensagens em sinais eléctricos e vice-versa, especifica as características físicas e eléctricas do barramento. A camada de Ligação de Dados, é a camada encarregue de entender o formato das mensagens. É aqui que as tramas são criadas e configuradas para poderem ser transmitidas, bem como descodificar as tramas recebidas. O protocolo CAN inicialmente era designado por CAN 2.0A, teve várias actualizações, até ao

CAN 2.0B, não tendo sido alterada a sua estrutura em comparação como o modelo OSI CiA (1991a,b).

3.2.2.1 Estrutura do CAN 2.0A

A estrutura protocolar da versão 2.0A é constituída pelas camadas: Física; Ligação de Dados e Aplicação. A camada Física tem como características: o nível de sinal; representação de *bits*; transmissão ao meio e sincronização de *bits*. Na camada de Ligação de Dados encontram-se duas sub-camadas, que são o Objecto e a Transferência. Os objectivos da sub-camada Transferência são: temporização e transferência de *bits*; sinalização e detecção de erros; validação e formatação das mensagens; confirmação da integridade das mensagens; detecção de falhas, mas limitado na correcção e verificação do barramento. O Objecto tem os objectivos de: saber qual a mensagem a transmitir; decidir quais as mensagens recebidas, que estão a ser utilizadas; ligar as camadas inferiores, relativas ao hardware, à camada de aplicação e verificação, processamento e filtragem das mensagens. A última camada, de Aplicação não tem funções atribuídas na especificação do protocolo CiA (1991a,b).

3.2.2.2 Estrutura do CAN 2.0B

O protocolo CAN 2.0B, sofreu algumas alterações em relação à versão inicial. E essas alterações vão cingir-se às duas camadas inferiores. Na camada Física surgem assim três sub-camadas, sendo estas o PS (*Physical Signalling*), PMA (*Physical Medium Attach*) e MDI (*Medium Dependent Interface*). As funções destas camadas são:

- **PLS:** A codificação/descodificação e temporização dos *bits* e a sincronização;
- **PMA:** Características do *transceiver*;
- **MDI:** Características do meio de transmissão (cabos, fichas, etc).

A camada Ligação de Dados, é constituída pelas sub-camadas LLC e MAC. As características da LLC são a de filtrar a aceitação de mensagens, notificação de *overload* e gestão de recuperação de mensagens ou de situações de erro. Já a sub-camada MAC tem como características o encapsulamento/desencapsulamento dos dados nas mensagens, codificação das mensagens (*stuffing/destuffing*), gestão de acesso ao meio, detecção e sinalização de erros, confirmação da integridade das mensagens e serialização/desserialização da informação [CiA \(1991a,b\)](#).

3.2.3 Camada Física

A camada Física é responsável pela transmissão física dos dados (*bits*) entre os diferentes nós da rede.

3.2.3.1 CAN *bit*

A codificação utilizada num barramento CAN é *Non Return to Zero* (NRZ). O barramento tem três tipos de estados lógicos, dominante, recessivo e *idle*, sendo que este último é recessivo. O problema neste tipo de codificação, é que a transmissão de diferentes *bits* não implica a mudança de estado do barramento, assim numa sequência de *bits* no mesmo estado lógico os nós podem ter dificuldade de sincronização. Para evitar este problema, recorre-se ao *bit stuffing*, onde só são permitidos cinco *bits* consecutivos no mesmo estado lógico. Se for necessário transmitir mais do que cinco, são introduzidos *stuff bits* de polaridade inversa (pela camada Ligação de dados). Todos os nós receptores farão o oposto para que a mensagem seja bem recebida ([Bosch, 2010](#)).

3.2.3.2 Temporização de um *bit*

O tempo de *bit*, é o tempo que ele se encontra no barramento, logo é o período de tempo que vai desde que o bit é transmitido até que é recebido. As funções como a sincronização dos nós, compensação de atrasos e a posição da amostragem, são

definidos pela entidade *bit timing programmable logic*. Cada nó contém o seu relógio interno, cujos pulsos são utilizados para obter o tempo nominal de *bit*. O tempo nominal de *bit* é formado por quatro segmentos, e cada segmento é formado por múltiplos *Time Quantum* (T_q). O *Time Quantum* é a mais pequena unidade de tempo fixo, sendo que cada *bit* é formado por 8 a 25 *Time Quanta* (Bosch, 2010).

A estrutura do tempo nominal de bit é a seguinte:

- **Sincronização:** é utilizado na sincronização dos nós e tem a duração de $1T_q$;
- **Tempo de Propagação:** fornece o tempo de propagação no barramento e pode ir até aos $8T_q$;
- **Segmento "Phase Buffer 1":** serve, junto com o segmento 2, para compensar os erros de fase, podendo, por ressincronização, se encurtado ou estendido. Pode ter até $8T_q$;
- **Segmento "Phase Buffer 2":** o valor do segmento 2 é o máximo entre o segmento 1 e o tempo de processamento da informação.

O ponto de amostragem fica no fim do segmento 1, e é onde é lido o nível do *bit* (recessivo ou dominante). Para compensar os erros de fase, os segmentos 1 e 2 podem ser encurtado ou estendido, e ao número de T_q s que se adiciona ou retira tem o nome de *Resynchronization Jump Whith* (RJW).

Sincronização

O segmento Sincronização existente no tempo nominal de *bit*, é usado para fazer a sincronização dos nós. A transição do sinal e dá-se neste segmento e caso exista um desvio, os segmentos 1 e 2 serão ajustados. Existem dois tipos de sincronização, a *Hard Synchronization* e a *Soft Synchronization*.

- **Hard Synchronization:** Este ocorre na primeira passagem de recessivo para dominante (nível lógico 1 para 0), quando o barramento está em *idle*, o que indica um *Start-of-Frame* (SOF);
- **Soft Synchronization:** Ocorre quando o nível passa de recessivo para dominante, durante a recepção das mensagens, e em opção, de dominante para recessivo quando as velocidades são baixas. Enquanto o nó CAN está a receber uma mensagem, o *Soft Synchronization* pode diminuir ou aumentar o tempo de *bit*. Quando o transmissor é lento o segmento 1 é aumentado de RJW, quando é rápido o segmento 2 é diminuído.

Se a passagem do estado recessivo para o dominante for detectado, quando o tempo de *bit* estiver no segmento 1, vai ser adicionado RJW ao segmento 1, e interpreta-se que o transmissor é lento. Ao invés, de acontecer no segmento 1, se acontecer no segmento 2, interpreta-se que o transmissor é rápido e é retirado o RJW ao segmento 2, para que o tempo de *bit* comece mais cedo (Voss, 2005; Bosch, 1991).

Propagação

O segmento Propagação é utilizado para compensar os atrasos que possam ocorrer e é composto por 1 a $8T_q$. Este segmento deve ter o dobro do tamanho do atraso, para compensar o atraso do transmissor para o receptor e vice-versa (Voss, 2005).

Segmentos 1 e 2

Os segmentos são utilizados para compensar variações e erros da fase. Podem ter até $8T_q$. O segmento 1 é usado para compensar o erro da fase e o seu comprimento pode ser aumentado durante a resincronização, o segmento 2 usando na mesma situação, mas o seu comprimento pode ser diminuído (Voss, 2005).

3.2.3.3 Topologia do Barramento

O barramento é constituído por um fio bi-filar simples (CAN_H e CAN_L) e nas suas extremidades têm uma resistência de terminação de 120Ω , para anular as reflexões eléctricas que possam acontecer. O barramento CAN por ser diferencial faz com que haja menos interferências electromagnéticas (Voss, 2005; Bosch, 1991).

Sinal no Barramento

O CAN utiliza uma técnica de sinalização por tensão diferencial, significa que o nível de sinal é determinado pela diferença de tensão entre o CAN_H e o CAN_L. Quando o nível do sinal é recessivo, o CAN_H não pode ter uma tensão maior que 0.5V em relação ao CAN_L. No nível dominante a diferença de tensão tem de ser de pelo menos 0.9V, o valor nominal em CAN_H é de 3.5V e de 1.5V para CAN_L. A diferença de tensão, em condições normais, será de 2V e de 0V, para o estado dominante e recessivo, respectivamente. A vantagem da transmissão diferencial é o facto de estar menos susceptível a interferências, pois caso existam as duas ligações do CAN irão sofrer de igual modo, permanecendo assim inalterada a diferença de tensão (Voss, 2005).

Ligação dos nós

A interface do controlador CAN e o barramento é feita através de um *transceiver*, que tem a função de codificar os sinais do controlador para o sinal diferencial. O *transceiver* tem então a função de converter os níveis *standards*, na tensão diferencial do barramento (Voss, 2005).

Comprimento e velocidade do barramento

A velocidade de transmissão e o comprimento do barramento estão intrinsecamente ligados, pois os processos de arbitragem necessitam que o sinal de propague até o nó remoto e regresse antes de ser amostrada. Caso se necessite de mais distância pode-se fazer uso de repetidores, que reforçam o sinal. A tabela seguinte 3.1, mostra possíveis secções de cabo e tipos de configuração de rede seleccionados (Voss, 2005).

Tabela 3.1 – Características do barramento.

Velocidade	Tipo de cabo	Resis./Comp. do cabo	Terminal	Comprimento
50 kb/s a 1 km	0.75 ...0.8 mm ² (AWG18)	70mΩ	150 ... 300Ω	600 .. 1000 m
100 kb/s a 500 m	0.5 ... 0.6 mm ² (AWG20)	<60mΩ	150 ... 300Ω	300 ... 600 m
500 kb/s a 100 m	0.34 ...0.6 mm ² (AWG22, AWG20)	<40mΩ	127Ω	40 ... 300 m
1000 kb/s a 40 m	0.25 ...0.34 mm ² (AWG23, AWG22)	<26mΩ	124Ω	0 ... 40 m

3.2.4 Camada Ligação de Dados

As mensagens transmitidas na rede CAN são enviadas por *broadcast*, logo todos os nós recebem as mensagens. Isto não significa que o nó reaja a qualquer mensagem recebida. As mensagens contém um identificador de mensagem (ID), que é único em toda a rede. Este identificador define o significado da mensagem bem como a sua prioridade, o que é importante em caso de competição pelo barramento.

3.2.4.1 Tipos de Tramas

Existem quatro tipo diferentes de tramas, que efectuam a transmissão das mensagens:

- Trama de dados;
- Trama de pedido remoto;
- Trama de erro;
- Trama de sobrecarga.

Os dois primeiros tipos de trama, transmitem as mensagens quando o sistema está normal, os outros dois enviam mensagens quando o sistema está num estado de erro ou de possível erro (Voss, 2005; Metrôlho, 1999).

Trama de dados

Para o CAN 2.0A e para CAN 2.0B o formato da trama de dados é ligeiramente diferente, mas contém o mesmos campos. Os campos são: Início da Trama (SOF - *Start of Frame*); Campo de Arbitragem (*Arbitration field*); Campo de Controlo (*Control Field*); Campo de Dados (*Data Field*); Campo de Verificação Cíclica de Erro (CRC - *Cyclic Redundancy Check*); Campo de ACK (*Acknowledge Field*) e Fim da Trama (EOF - *End of Frame*), ver Figura 3.10. Entre as frame existe ainda o Espaço Entre Tramas (IFS - *InterFrame Space*) (Voss, 2005; Metrôlho, 1999).

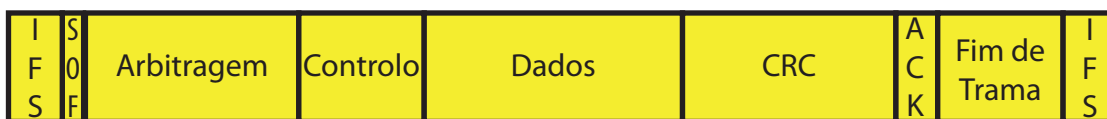


Figura 3.10 – Formato da trama de dados

- Campo Início da Trama: Este campo marca o inicio da trama de dados e de pedido remoto. Consiste em apenas 1 *bit* dominante.
- Campo de Arbitragem – CAN 2.0A: Este campo é composto pelo Identificador e pelo *bit* RTR (*Remote Transmission Request Bit*). O Identificador tem o tamanho de 11 *bits*, sendo que os *bits* do 10 ao 4 são os mais significativos e os

outros os menos significativos. O *bit* RTR distingue se a trama é de dados (RTR = 0) ou pedido remoto (RTR = 1).

- Campo de Arbitragem – CAN 2.0B: O campo arbitragem do CAN 2.0B é semelhante ao anterior, sendo formado pelo Identificador de 29 *bits*, 1 bit SRR (*Substitute Remote Request*), 1 bit IDE (*Identifier Extension*) e 1 *bit* RTR. O *bit* IDE, distingue se a trama é *standard*(CAN 2.0A) (IDE = 0) ou *extended*(CAN 2.0B) (IDE = 1). O *bit* SRR, é recessivo.
- Campo de Controlo: Este campo é formado por 6 *bits*, os dois primeiros são dominantes e os outros quatro definem o comprimento do campo de dados (DLC - *Data Length Code*). Se for uma trama de pedido remoto este campo indicará a quantidade de dados pedida.

O primeiro *bit* correspondem a IDE/r1, que para a trama do tipo *standard* será usada como IDE, para a trama do tipo *extended* corresponde a r1, pois o IDE já existe no campo arbitragem. Significa que o CAN 2.0A, fica com apenas um *bit* para uma futura extensão (r2).

- Campo de Dados: Este campo contém os dados a serem transmitidos através da trama de dados e pode transportar 0 a 8 *bytes* de dados.
- Campo de Verificação Cíclica de Erro: O CRC é formado por 15 *bits* seguidos de 1 *bit* recessivo (CRC *Delimiter bit*), esta sequência de *bits* resulta do código *hamming* de distância mínima de seis, que permite que sejam detectados no máximo 15 erros, ou a correcção 1 a 5 erros espalhados aleatoriamente pelos *bits* protegidos. Os *bits* protegidos são os contidos nos campos Início da Trama, Campo de Arbitragem, Campo de Controlo e Campo de Dados.
- Campo de ACK: Este campo é formado por dois *bits*, o *Acknowledgement Slot* e *Acknowledgement Delimiter*. Este campo é usado para que o transmissor saiba que pelo menos um nó da rede recebeu a mensagem. Aquando da transmissão ambos os *bits* são recessivos, no envio pelo receptor o *bit* ACK Slot é dominante.

- Campo de Fim da Trama: O campo Fim da Trama é constituído por 7 *bits* recessivos, e tem a função de assegurar que as *flags* erro permanecem na mensagem, recolhendo-as.

Trama de Pedido Remoto

Esta trama é muito semelhante à trama de dados, havendo apenas duas diferenças, a primeira é que não existe o campo destinado aos dados, a segunda diferença está no bit RTR que nesta trama é enviado como recessivo, ao contrário da trama de dados. Em condições normais o transmissor envia a trama de dados, mas por vezes o receptor precisa de receber dados e envia a trama de Pedido Remoto (Voss, 2005; Metrôlho, 1999).

Trama de Erro

Quando ocorre um erro na transmissão ou recepção de uma trama de dados ou de pedido remoto, é assinalado na trama de erro, o nó detector envia a trama de erros a todos os outros nós da rede. Poderão existir nós que não detectem o erro, mas irão detectar o erro de *stuffing* introduzido pela *flag* de erro. A trama é constituída por dois campos, a *flag* de erro (*error flag*) e o delimitador de erro (*error delimiter*). A *flag* de erro, sinaliza a existência de um erro. E existem dois tipos de erro o activo e o passivo. Se o nó tiver a transmitir quando ocorre o erro, interrompe-se a transmissão com uma *flag* de erro activa, que corresponde a uma sequência de 6 *bits* dominantes. Como a sequência viola a regra de *bit stuffing*, que não permite 6 *bits* consecutivos de mesmo valor lógico, os outros nós irão detectar que houve um erro, e também eles gerar uma trama de erro. O delimitador de erro, conclui a trama de erro e após esta o nó que foi interrompido tentará reenviar a mensagem, e o barramento regressa ao normal.

Se o erro acontecer durante a recepção, transmite-se uma *flag* de erro passiva, esta

flag é formada por 6 *bits* recessivos consecutivos, pelo que consiste num total de 14 *bits* recessivos. A menos que o nó transmissor detecte o erro, a trama de um erro passivo não reproduz nenhum efeito na rede, quando gerada pelo receptor. Se o nó transmissor gerar um erro passivo, os outros nós também irão gerar tramas de erro, devido ao *bit stuffing*. O delimitador de erro é constituído 8 *bits* recessivos e permite aos nós retomarem a transmissão. Por último existe o espaço entre tramas, formado por 3 *bits* recessivos (Voss, 2005; Metrôlho, 1999).

Trama de Sobrecarga

A trama de sobrecarga em tudo semelhante à trama de erro, contém dois campos, a *flag* de sobrecarga (*overload flag*) e o delimitador de sobrecarga (*overload delimiter*), com funções idênticas. Existem dois tipos de condições que provocam a transmissão da *flag* de sobrecarga:

- O receptor necessita de um certo tempo de atraso para aceitar a próxima trama de dados ou de pedido remoto. Assim o início da trama de sobrecarga só é permitido no primeiro *bit* que antecipa a fase de intermissão (*intermission*) no período entre tramas (*InterFrame Space*);
- Aquando da detecção de um *bit* dominante durante a fase de intermissão. Neste caso a trama de sobrecarga começara após a detecção do *bit* dominante.

A *flag* de sobrecarga é constituída por 6 *bits* dominantes. Quando existe a sobreposição de *flags* de sobrecarga, a fase de intermissão é destruída, o que faz com que todos os nós detectem a sobrecarga. O delimitador é constituído por 8 *bits* recessivos. Após a transmissão de uma *flag* de sobrecarga, e assim que detectem um *bit* recessivo os nós transmitem em conjunto mais 7 *bits* recessivos, completando os 8 *bits* do delimitador (Voss, 2005; Metrôlho, 1999).

Espaço entre tramas

O espaço entre tramas tem como função a separação de duas tramas, de dados ou de pedido remoto, consecutivas. É constituída por 3 *bits* obrigatoriamente recessivos, designado por intermissão, durante os quais não pode haver nenhuma trama, a não ser que seja trama de erro. Existe ainda um campo designado de transmissão suspensa (*Suspend Transmission*) que surge aquando do erro passivo, consiste em 8 *bits* recessivos após o campo intermissão (Voss, 2005; Metrôlho, 1999).

3.2.4.2 Detecção de Erros

Na rede CAN a consistência dos dados é uma das prioridades, e é conseguida através de conceitos como o *multicast* e *error handling* (suporte de erros). O suporte de erros estão definidos diversos tipos de erros bem como mecanismos de detecção:

- Erro de ACK (*Acknowledge error*);
- Erro de CRC (*Cyclic Redundancy Check*);
- Erro de formato (*Form Check*);
- Erro de *bit* (*Bit Check*);
- Erro de *bit stuffing*.

Detecção do erro de ACK

Na recepção de uma mensagem, o nó receptor deve confirmar que recebeu a mensagem, colocando o *bit* ACK como dominante. Em caso do transmissor não receber a confirmação, gera-se um erro de ACK.

Detecção do erro de CRC

O protocolo CAN implementa o CRC para garantir a integridade dos dados, com este método temos uma alta probabilidade de detectar erros. O seu funcionamento passa pelo transmissor calcular uma soma, através dos *bits* que vão desde o SOF até o fim do campo de dados. O resultado é enviada na trama, no campo CRC, o nó receptor calcula também o CRC e compara com o recebido. Caso não corresponda gera-se um erro de CRC, é enviada a trama de erro e a mensagem inicial é retransmitida.

Detecção do erro de formato

Existem vários campos, cujo nível é estático e recessivo, são eles o delimitador CRC, delimitador ACK, fim da trama e intermissão das tramas de dados e de pedido remoto. Quando surge em algum destes campos *bits* dominantes, é gerado um erro de formato. Mas existem excepções a considerar:

- Quando o nó receptor detecta um *bit* dominante no último *bit* do campo EOF;
- Quando um qualquer nó detectar um *bit* dominante no último *bit* do campo delimitador de erro;
- E quando qualquer nó detecta um *bit* dominante no último *bit* da camada delimitador de sobrecarga.

Detecção do erro de *bit*

O nó após transmitir uma sequência de *bits* para o barramento, verifica simultaneamente o seu estado. Em caso de detectar um estado diferente da sequência enviada, dá-se um erro de *bit*. O erro de *bit*, não só permite detectar o erro, como saber o local onde aconteceu. Neste tipo de erro também existem algumas excepções:

- Quando é detectado um *bit* dominante no barramento, na transmissão de um *bit* recessivo durante o campo de arbitragem;
- Quando é detectado um *bit* dominante no barramento, na transmissão de um *bit* recessivo durante o campo de ACK;
- Quando é detectado um *bit* dominante no barramento, na transmissão de um trama de erro passivo, sequência de 6 *bits* recessivos;
- E quando o nó esteja a transmitir o campo de arbitragem.

Detecção do erro de *bit stuffing*

Devido à utilização da codificação NRZ, por parte da rede CAN, não existe um estado de repouso, quando é enviado um sequência de *bits* do mesmo nível. Então dá-se uso ao *bit stuffing*, que a cada 5 *bits* do mesmo estado envia um inverso. O transmissor, que tem a responsabilidade de codificar a mensagem, tem de acrescentar um *bit* dominante após 5 *bits* consecutivos recessivos e vice-versa. O receptor, quando recebe a mensagem, deve tirar os *bits* em excesso. Existem várias excepções que têm um valor lógico fixo e que assim não violam a regra. Aquando do *bit stuffing*, a trama de erro é enviada, havendo um retransmissão da mensagem.

3.2.4.3 Gestão dos Erros

Na rede CAN é utilizado o mecanismo *Fault Confinement*, para gerir os erros. Em cada nó existem dois contadores de erro (TEC - Contador de erros de transmissão, e REC - Contador de erros de recepção). Existem três estados possíveis em que o nó pode estar, como se observa na Figura 3.11, mediante os valores dos seus contadores de erro, são eles:

- Estado activo ao erro;
- Estado passivo ao erro;

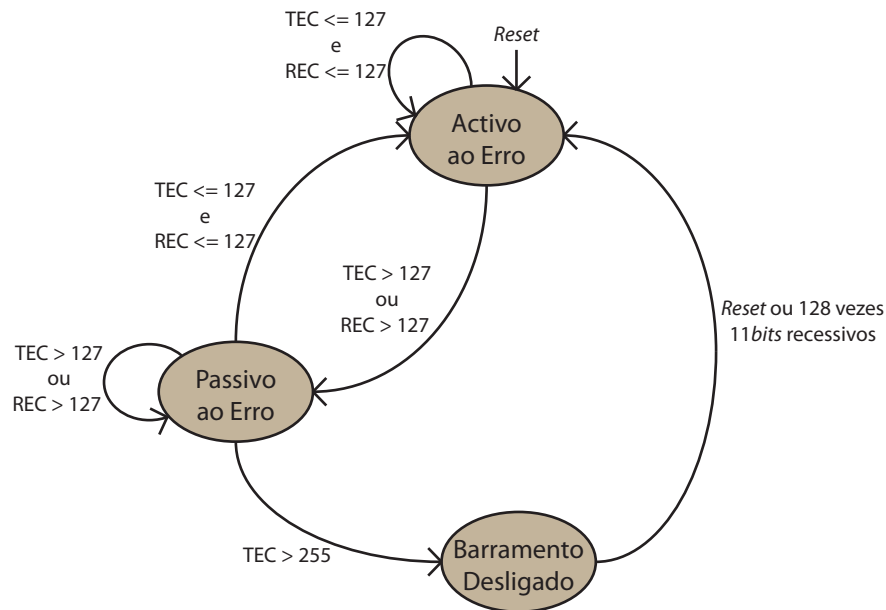


Figura 3.11 – Diagrama de estado de erro

- Barramento desligado (*Bus off*).

Depois de um *Reset* o nó fica no estado activo ao erro, neste estado o nó participa nos procedimentos de comunicação na rede e envia uma *flag* sempre que detecta um erro, os contadores de erro vão sendo actualizados. Se durante a transmissão e recepção, pelo menos um nó detectar um ou mais erros, a transmissão é interrompida e sinalizado o erro, os contadores de erro vão ser incrementados. Se a mensagem for bem transmitida, os contadores são decrementados.

No estado activo ambos os contadores têm valor iguais ou inferiores a 127, se um dos contadores tiver mais de 127, o nó passa para estado passivo ao erro. No estado passivo ao erro o nó pode comunicar, mas caso detecte erros irá sinalizar a *flag* passiva de erros, a qual não provocará distúrbios na rede. Para além disso, este nó terá de aguardar mais 8 *bits*, do que aconteceria caso estivesse em modo activo ao erro, antes da retransmissão da mensagem sinalizada como errada. Se um nó estiver no estado passivo e os contadores decrementarem para menos de 128, o nó passa para estado activo ao erro, caso incremente para valores superiores a 255, o estado vai mudar para *bus off*. Neste estado o nó não tem influência no barramento.

Poderá regressar à actividade, após um *Reset* do sistema ou caso surja 128 vezes 11 *bits* recessivos consecutivos. Isto vai, limpar os contadores e o nó regressa ao estado activo ao erro (Voss, 2005; Pfeiffer et al., 2008).

3.2.5 Camada Aplicação

Inicialmente o protocolo CAN, apenas implementava as duas primeiras camadas do modelo OSI, mas devido às exigências da indústria em fornecer mais e melhores serviços, foram desenvolvidos pacotes software para implementar esses serviços na camada de aplicação. Várias companhias criaram as suas próprias soluções e algumas são mencionados de seguida:

- NMEA 2000: é utilizado em aplicações navais e aeronáuticas e foi desenvolvido por IMEA (*International Marine Electronics Association*);
- ISO 11783: é usado em aplicações agrícolas e foi criado por VDMA (*Verband Deutscher Maschinen - und Anlagenbau - German Engineering Federation*);
- ISO 11992: é utilizado em veículos, em camiões pesados é usado para comunicação entre o tractor e um ou vários reboques e foi projectado pela CiA (*CAN - in Automation*);
- SAE J1939: é aplicado em redes automóveis, com especial atenção para os autocarros e camiões e foi desenvolvido por SAE International (*Society of Automotive Engineers International*);
- DIN 9684: é encontrado em aplicações agrícolas, foi criado por LBS (*Landwirtschaftliches Bus-System*);
- CANKingdom: é utilizado em robots, teares, máquinas, etc, e Kvaser (Kvaser AB) foi quem a projectou;
- CANOpen: serve para diversos tipos de redes industriais, tendo sido desenvolvido pela CiA (*CAN - in Automation*);

- CAL: tem utilização em sistemas para controlo e monitorização de dispositivos industriais, e foi desenvolvido pela CiA (CAN - *in Automation*);
- SDS: é utilizado em sensores inteligentes e actuadores ligados directamente a uma rede e a *Micro Switch Honeywell* foi a empresa responsável pela sua existência;
- TT-CAN: tem uso essencialmente na indústria automóvel (controlo de sistemas do motor, chassis e transmissão, com espaço para aplicações by-wire), e foi criada pela CiA (CAN - *in Automation*);
- DeviceNet: é usado em automação industrial, e foi desenvolvido inicialmente pela *Allen-Bradley*, agora a *Rockwell Automation, Inc.*

4

Concepção e Implementação

4.1 Concepção

O objectivo é desenvolver dois tipos de sistemas, que troquem dados entre si. Um terá de ser móvel e o outro fixo, sendo que o fixo tem de encaminhar a informação de forma redundante.

Do lado do paciente existirá um dispositivo móvel com a possibilidade de adicionar actuadores e sensores, que possam monitorizar e controlar alguns movimentos do mesmo. Para o efeito, teremos um módulo inteligente para processamento dos diferentes dados, com bateria e comunicação sem fios de modo, que permita o acompanhamento do paciente sem no entanto criar limitações ou consequentemente no que se refere ao deslocamento. Utilizaremos, para o tratamento dos dados e ligação aos periféricos um microcontrolador, e na comunicação dos mesmos usaremos um módulo de comunicação IEEE 802.15.4, como se pode ver no esquema [4.1](#). A rede IEEE 802.15.4, não será utilizada exclusivamente para a troca de informação, mas será também utilizada para a localização através da potência entre os nós fixos e a estação móvel.

Será ainda necessária a utilização de um conjunto de sensores ou dispositivos para

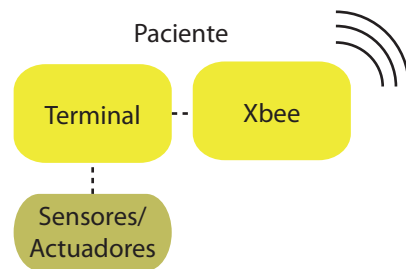


Figura 4.1 – Esquema do sistema paciente

a monitorização dos pacientes. Estes devem medir os sinais vitais, onde se podem incluir a temperatura, ritmo cardíaco, respiração e a pressão arterial, que oferecem dados muito importante para a avaliação das condições de saúde dos pacientes. A utilização de actuadores possibilita restringir os movimentos de alguns pacientes dentro do hospital, de modo que se for detectado a saída do paciente dos locais permitidos, o actuador deverá por exemplo bloquear a cadeira de rodas.

Do lado dos auxiliares de saúde teremos um terminal fixo que será constituído por um computador, que terá ligação série para a ligação com *modem* GSM, de ligação CAN à rede de nós IEEE 802.15.4 e Bluetooth, podemos observar o esquema na Figura 4.2. Como o computador não tem ligações CAN, será necessário um *driver* que faça a ligação entre a porta série do computador e a rede CAN e converta as mensagens.

Para a interface com os auxiliares ser a mais agradável terá uma aplicação em Java. A aplicação terá de permitir a gestão dos pacientes, dos alertas e manterá o historial de acontecimentos. Os auxiliares terão de ter um telemóvel, que se encontre associado ao sistema, para receber os alertas/informações por SMS ou através de Bluetooth. Ainda será implementado uma aplicação para a troca de informação por *Bluetooth*.

Temos assim uma infraestrutura combina nós fixos e móveis, que permitem a troca de dados e a localização *indoor*, e uma aplicação de monitorização e alerta que serve de interface para os auxiliares. Na Figura 4.3, podemos ver toda a infraestrutura, desde os pacientes com o dispositivo móvel, passando pelos nós fixos ligados por

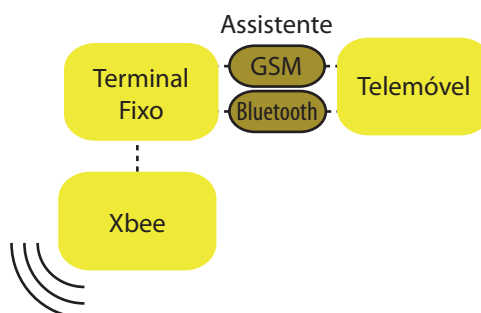


Figura 4.2 – Esquema do sistema assistente

uma rede CAN e a aplicação de monitorização.

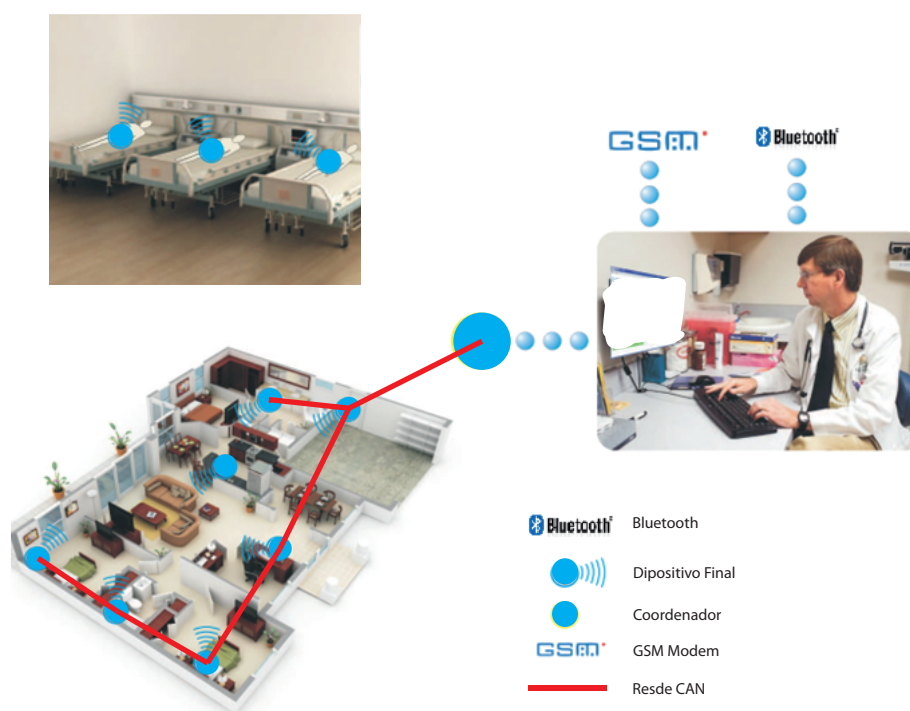


Figura 4.3 – Infraestrutura

4.2 Implementação

Na implementação são descritos os passos necessários à criação da infraestrutura, bem como as aplicações que foram usadas para a realização das diferentes tarefas. Foram efectuados os desenhos das PCBs, a montagem dos protótipos e a programação dos mesmos, bem como a construção da aplicação de assistência.

4.2.1 Rede híbrida Xbee e CAN

Para a comunicação entre os nós Xbee e o terminal fixo que processa os dados foi utilizado o protocolo CAN, por ser mais seguro e fiável do que utilizar exclusivamente a rede sem fios.

Por isso mesmo, para a implementação do sistema, e tendo em conta as especificações CAN, desenvolveu-se os nós de rede para a comunicação por este protocolo, tanto o *hardware* como o *firmware*. Como o computador, não tem ligações CAN, utilizamos um PIC, que irá fazer de *driver*, fazendo a transformação dos dados transmitidos por CAN, para RS-232 e vice-versa. Assim o primeiro dispositivo implementado é um *driver* ou estação base (EB) capaz de fazer a ponte entre o computador e a rede de nós. O *driver* é constituído por três componentes principais um PIC18F2580 para o processamento de dados, o *chip* FTDI para possibilitar o uso da ficha USB, bem como um *driver* CAN. Os nós da rede, têm a função de receber/transmitir informação por CAN e Xbee, para isso foi usado um PIC18F2580, *driver* CAN e um módulo Xbee. A estação/dispositivo móvel (EM) tem de receber/transmitir informação Xbee e CAN. A utilização do CAN no dispositivo móvel prende-se com facto de ser necessário fiabilidade, principalmente quando queremos controlar uma cadeira de rodas. A rede Xbee serve para a troca de informação, e através da potência entre antenas podemos saber a localização da estação móvel, isto com um margem de erro. A Figura 4.4 mostra todos os elementos de rede que foram acima descritos.

Para a comunicação sem fios foram criadas as duas funções mais básicas necessárias

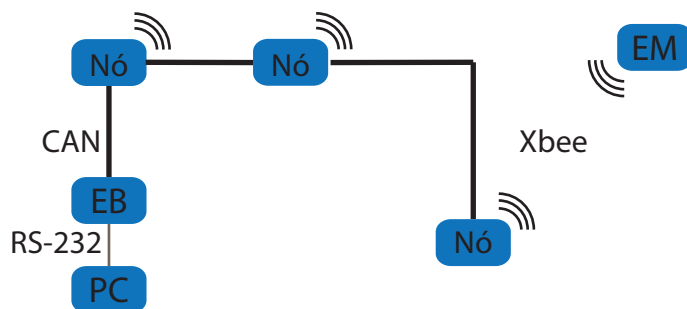


Figura 4.4 – Esquema da rede

para tal. Por isso criou-se as funções que vão comunicar com o módulo Xbee, "readFrame" e a "sendFrame", responsáveis pelo envio e recepção das frames Xbee:

Listagem 1 Funções Xbee

```

int readFrame(frame*); - Ler a frame;
void sendFrame(frame*); - Envia uma frame;

```

Foi também necessário criar uma estrutura que representa-se o tipo de frame que é enviada e recebida pelo Xbee:

Listagem 2 Frame Xbee

```

typedef struct {

    unsigned int address;
    unsigned int s_address;
    unsigned int size;
    unsigned char buffer[20];
    unsigned char rssi;
    unsigned char options;
} frame;

```

Para a comunicação CAN, entre os nós da rede, foi necessário criar também o código respectivo para os microcontroladores utilizados. Por isso mesmo foram criadas as seguintes funções:

Listagem 3 Funções CAN

```
void InitCAN(void);
unsigned char CANReceivedMessage(void);
void ReadCANMessage (unsigned int* ID, unsigned char* RTR,
unsigned char* DLC,unsigned char* dados);
void SendCANMessage (unsigned int ID, unsigned char RTR,
unsigned char DLC, unsigned char* buffer);
```

Na ligação por porta série, foi necessário implementar as rotinas seguintes:

Listagem 4 Métodos para a Porta Serial

```
void InitSerial (void);
void DisableSerialRX(void);
void EnableSerialRX(void);
void SetBaudRate (int BaudRate);
void putchar (unsigned char c);
unsigned char getch(void);
unsigned char receivedData(void);
```

4.2.2 *Bluetooth* e Aplicação para o Equipamento Móvel

O terminal móvel do lado do assistente de saúde tem duas formas de ser alertado, através de uma aplicação que utiliza um a porta serie emulada (RFCOMM) e poderá ser alertado quando necessário ou através de uma mensagem de texto (SMS). O *Bluetooth*, como já foi descrito não permite grandes distâncias entre dispositivos para que possa haver transferência de dados, a utilidade desta tecnologia de comunicação será de evitar o envio de mensagens de texto via *modem* GSM, a fim de reduzir custos de operação. Foi implementada uma aplicação do lado do telemóvel que permite trocar informação com o terminal físico, assim que ambos estivessem ao alcance e emparelhados.

4.2.3 GSM

Foi implementada na aplicação em Java no computador, os métodos para comunicação com o *modem* GSM. Para a comunicação com o modem é utilizado RS-232, por isso foi desenvolvida a classe `Serie.java`, com os seguintes métodos:

Listagem 5 Classe `Serie.java`

```
Open(String porta); - Para a abertura da porta e das Streams;  
Close(); - Para fechar a porta e as Streams;  
LerLinha(); - Ler uma linha da porta aberta;  
EscreverLinha(String linha); - Escreve uma linha pela porta;  
LerCaracter(); - Ler caracter a caracter;
```

Para interacção com o *modem* foi desenvolvida uma classe que tem a função de interpretar as respostas e enviar comandos para o mesmo. O nome da classe é `Modem.java`, e foram implementados os seguintes métodos:

Listagem 6 Classe `Modem.java`

```
Connect(); - Para a abertura da porta, para comunicacao com o  
modem. Para isso, cria um objecto da classe Serie, e utiliza  
o metodo Open();  
Disconnect(); - Fecha a porta. Utiliza para o metodo Close(),  
da classe Serie;  
SendAtCommand(String atmsg); - Envia comandos AT, aguardando  
pelos dois tipos de resposta o OK ou ERROR;  
SendSMS(String num, String sms); - Envia mensagens de texto,  
para um determinado numero.  
ReadSMS(); - Ler mensagens de texto recebidas. Para o efeito  
utiliza o metodo LerLinha() Implementada na classe Serie.
```

O `SendSMS` foi implementado da seguinte forma:

Listagem 7 Código SendSMS

```
SendAtComand("at+cmgf=1"); - O envio deste comando configura as SMS  
em modo de texto;  
Serie.EscreverLinha("AT+CMGS="+numero); - Envia-mos o numero do  
destinatario; Aguarda-se que o modem responda com os caracteres  
" >" (espaco, maior), escreve-se a mensagem de texto a enviar;  
CTRL+Z - Envia a mensagem de texto.
```

O ReadSMS foi implementado da seguinte forma:

Listagem 8 Código ReadSMS

```
SendAtComand("at+cmgf=1"); - O envio deste comando configura as SMS  
em modo de texto;  
Serie.EscreverLinha("AT+CMGL=\"all\""); - Envia-se o comando para  
abrir todas as mensagens de texto;  
Serie.LerLinha(); - Ler as mensagens devolvidas.
```

4.2.4 Aplicação

Foi necessário implementar uma interface para o pessoal médico e enfermeiros. A linguagem de programação recaiu sobre a Java, e com efeito utilizou-se a ferramenta de desenvolvimento NetBeans.

A aplicação criada tem uma janela principal, Figura 4.5, onde se observa duas áreas distintas, a da esquerda que irá manter actualizado os acontecimentos com os pacientes é semelhante a um LOG onde é guardado o histórico das acções efectuadas, a da direita faz a listagem dos pacientes activos que estão a ser monitorizados.

Para a gestão dos pacientes, será necessário premir o botão "GERIR" aonde teremos acesso a uma nova janela com onde se se pode adicionar ou modificar os pacientes, como se pode observar na Figura 4.6. A aplicação necessitará dos dados do paciente a adicionar ou modificar, a informação necessária: O MAC do dispositivo móvel do

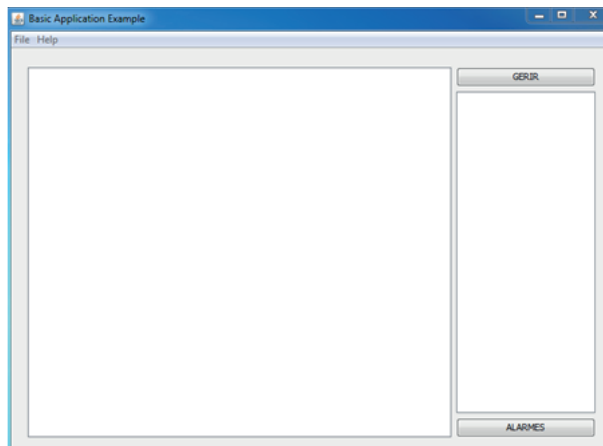


Figura 4.5 – Aplicação: janela inicial

paciente, o nome do paciente, a área aonde o paciente está permitido circular e por último o nome do enfermeiro associado ao paciente.

O botão "OK" activa um novo paciente ou actualiza um paciente existente, o botão "CANCEL" remove o paciente já existente ou cancela a criação de um paciente novo.

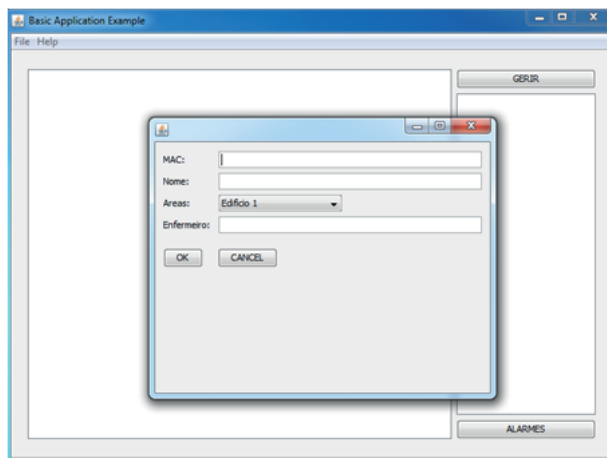


Figura 4.6 – Aplicação: janela adição de pacientes

4.2.5 *Hardware*

Foram desenhadas algumas PCBs, desde os protótipos até se chegar ao "produto final". As primeiras placas desenvolvidas foram as placas que eram mais simples,

logo sendo menos complexas permitiram a aprendizagem ao programa de desenho utilizado, deste modo foram criadas as placas associadas aos nós 4.7. As placas associadas aos nós contêm: o PIC18F2580 e o *driver* CAN PCA82C250; foram colocadas as ligações de alimentação, de programação, RS-232 e CAN. A escolha deste microprocessador deve-se a este permitir suporte CAN e Série, um requisito obrigatório no sistema a implementar. A porta Série serve para a comunicação com o módulo Xbee, ver Figura 4.8, quanto à ligação CAN permite a comunicação com a rede CAN do sistema. A placa é alimentada a 5V, bem como a placa do módulo Xbee.

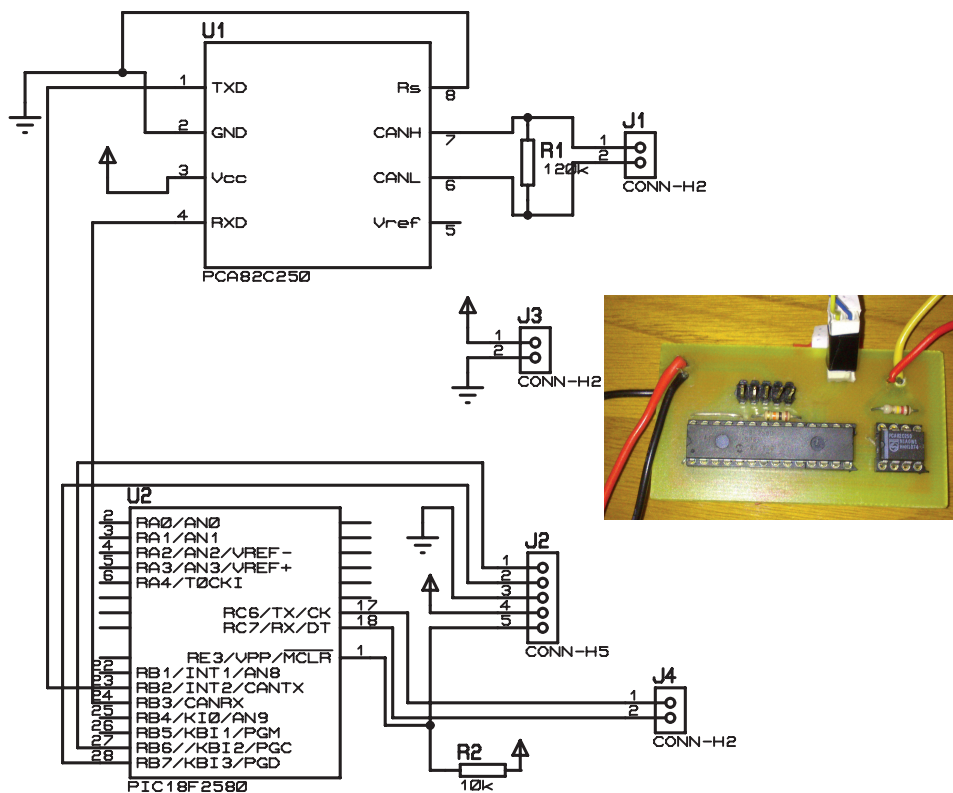


Figura 4.7 – Visualização do esquemático (esquerda) e foto (direita), do nó de rede

O dispositivo seguinte a ser desenhado foi a estação base, para que assim completasse a parte do sistema do lado dos auxiliares de saúde, e após isto desenvolveu-se o *firmware*, para a comunicação CAN. A placa desenhada na Figura 4.9, teve em conta que teria de haver comunicação entre um computador e a rede CAN, assim sendo foi utilizado um *chip* FTDI FT232BL, que serve de interface USB para UART,

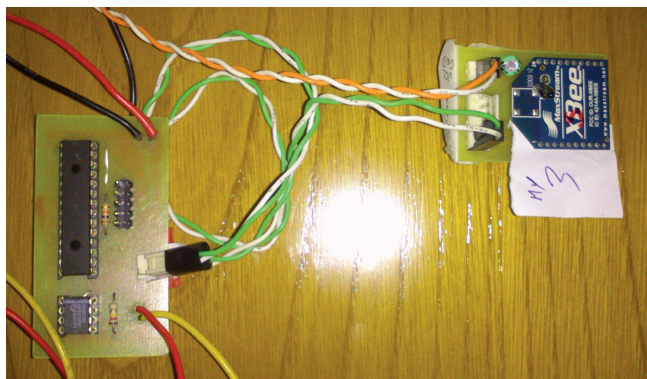


Figura 4.8 – Nó de rede com o módulo Xbee

a fim de poder comunicar com o computador através de uma porta USB, quanto ao *driver* CAN foi utilizado o MCP2551. Na PCB, foi ainda adicionado um botão de *reset*, LED de alimentação, LED TX e RX para diagnostico da comunicação série, três LEDs de *debug* e por último os pinos de programação.

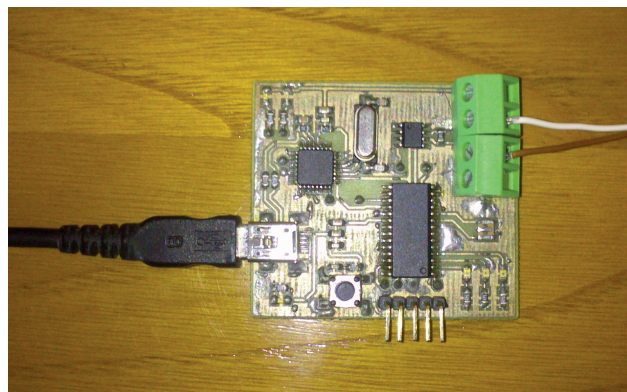
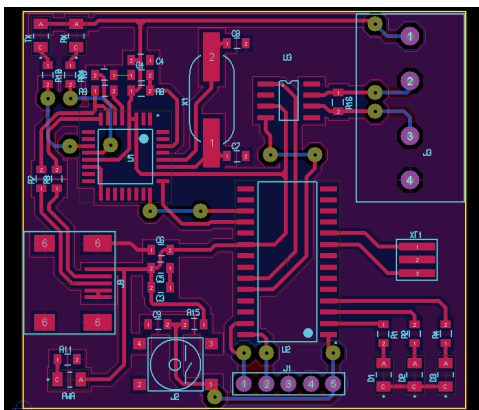


Figura 4.9 – Visualização do esquemático (esquerda) e foto (direita), da estação base

Por último foi o terminal móvel a ser desenhado, este contém um microprocessador PIC18lf2580, uma bateria de 2000mAh a 3.7V e carregador de baterias MCP73831 utilizando a porta micro USB, ainda o regulador MCP1825 com $V_{out}=3.3V$, um módulo Xbee e o *driver* CAN MCP2551, podemos ver o esquemático e a fotografia do terminal móvel na Figura 4.10. O microcontrolador escolhido para a EM, tem uma tensão de funcionamento que vai dos 2V até aos 5,5V, enquanto que a versão usada nos nós tem uma tensão de funcionamento entre os 4,2V e os 5,5V. Era

necessário que este pudesse funcionar apenas utilizando como fonte a bateria, por isso a escolha. Como as ligações da placa da estação móvel se assemelham à dos nós, a placa foi desenhada de modo a que e mediante os componentes que escolhemos soldar, podemos ter um terminal móvel ou um nó. E assim em vez de termos duas PCBs apenas temos uma, o que torna menor o custo de produção. A alimentação pode ser feita pela porta USB, pela bateria ou mesmo pela ligação CAN. As ligações existentes são os conectores CAN, do programador, micro USB, Xbee e para a bateria. À semelhança da EB também tem três LEDs para *debug*, um LED de alimentação e ao em vez dos LED TX e RX, teremos apenas o LED de estado do Xbee.

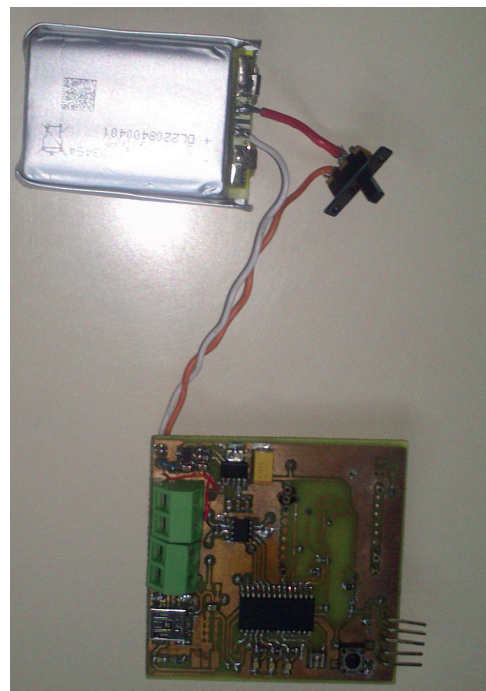
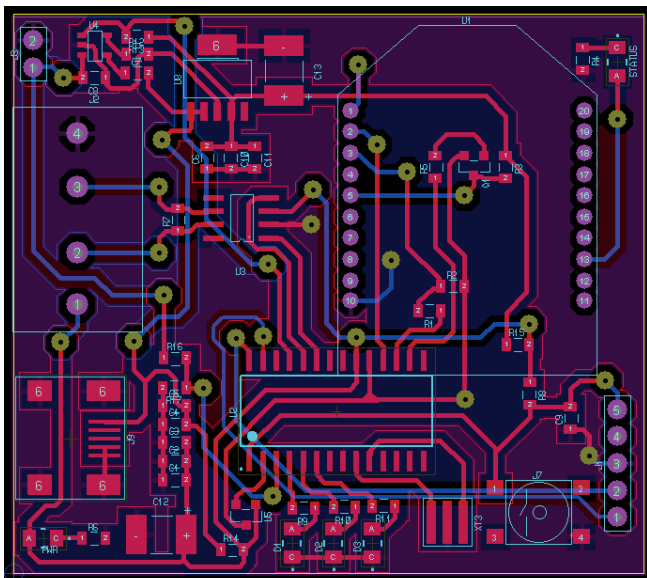


Figura 4.10 – Visualização do esquemático (esquerda), e foto do terminal móvel (direita)



Testes e Resultados

A metodologia de desenvolvimento utilizada baseia-se na prototipagem e verificações de erro, pelo que foram realizados testes durante todas as fases do projecto, de modo a verificar se cada um deles funcionava correctamente e deste modo proceder-se às actualizações necessárias.

5.1 Testes Individuais

Aquando do desenvolvimento do projecto foram sendo efectuados os testes às diferentes comunicações e aos diferentes elementos do sistema, simulando situações de funcionamento reais, para que fossem encontrados erros de *firmware* ou defeitos no *hardware*. E para que os mesmos fossem corrigidos, antes de se prosseguir para outra etapa.

5.1.1 Aplicação

Vamos criar um novo paciente, preenchendo os dados, como se pode ver na Figura, o MAC: "2", Nome: "nuno", Áreas: "Edifício 1" e Enfermeiro: "Carlos Mestre",

de seguida pressiona-se "OK", para assumir a nova entrada, como demonstrado na Figura 5.1.

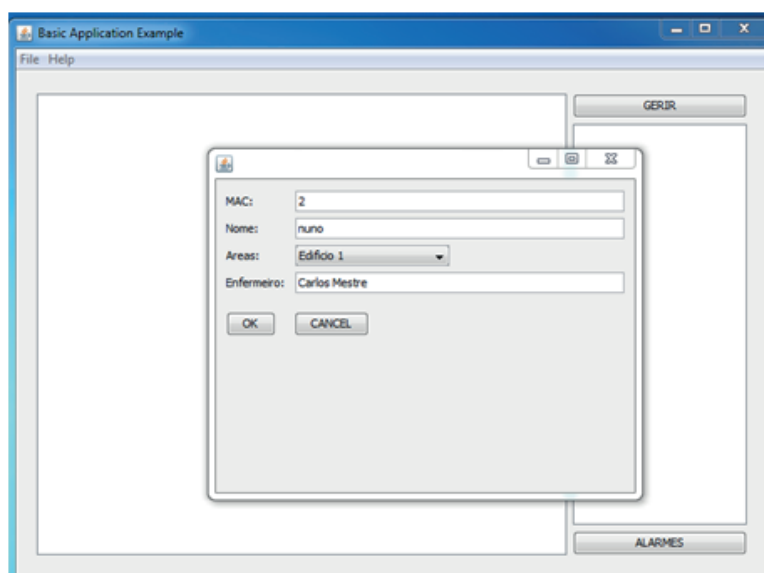


Figura 5.1 – Aplicação: janela adição de pacientes preenchida

Na janela principal a lista de pacientes é actualizada, neste caso com o nome de paciente "nuno", que tinha sido criado para o efeito, pode se observar pela Figura 5.2.

Na janela principal temos também um botão de alarmes que permite ao enfermeiro gerir os alarmes que surjam, e em tempo real. Clicando, em alarmes temos uma janela com a lista de alarmes, que na Figura 5.3 se encontra vazia.

Após terem surgido três alarmes, fazemos surgir a janela alarmes e temos os alarmes listados, como se vê na Figura 5.4.

Vamos dar como resolvido o alarme 1. Selecciona-se o alarme 1 da lista de alarmes e pressiona-se o botão "resolvido", como na Figura 5.5. Assim o alarme 1 é removido da lista de alarmes e surge a mensagem de notificação de quem foi o enfermeiro que resolveu o alarme, como se observa pela Figura 5.6.

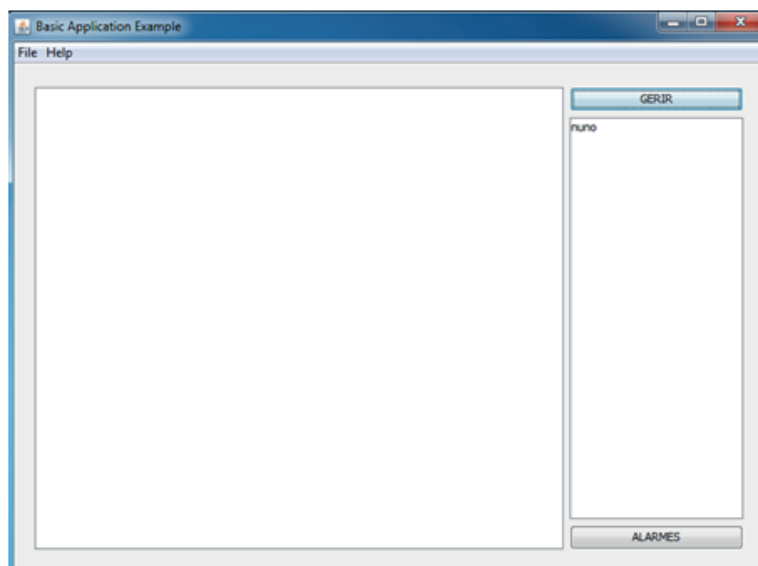


Figura 5.2 – Aplicação: janela principal

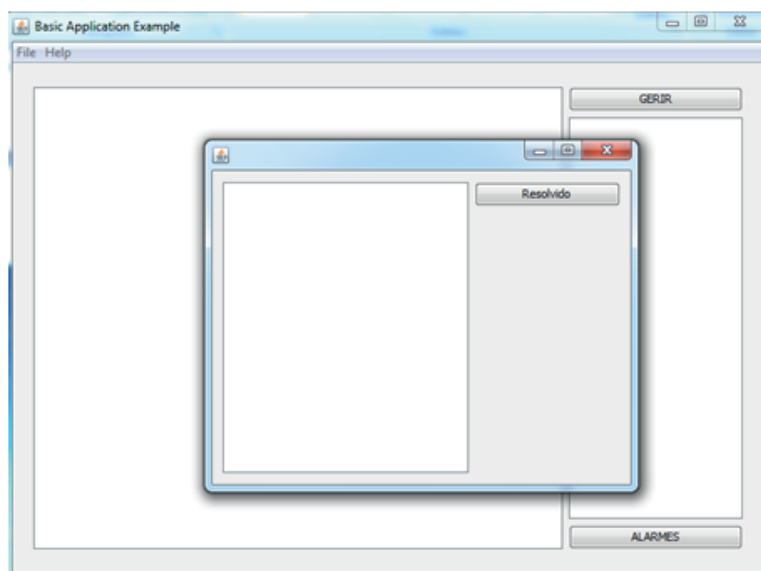


Figura 5.3 – Aplicação: janela de alarmes

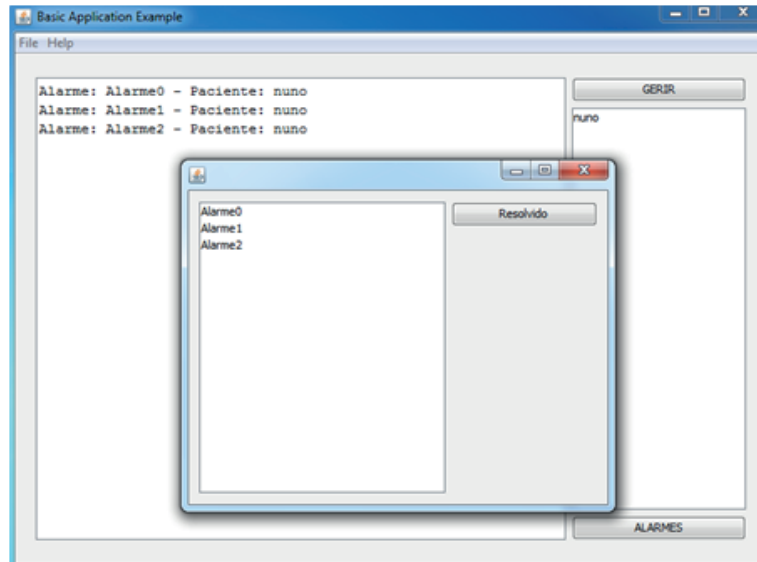


Figura 5.4 – Aplicação: alarmes listados

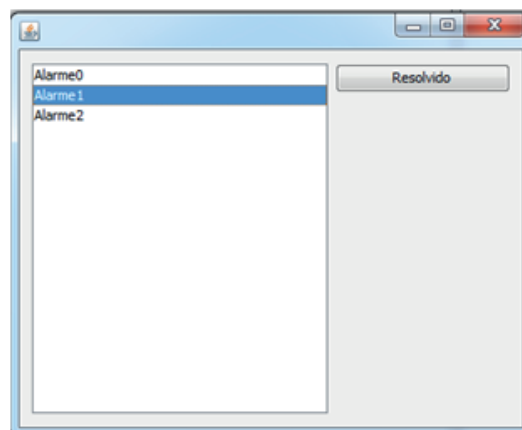


Figura 5.5 – Aplicação: selecção de alarme

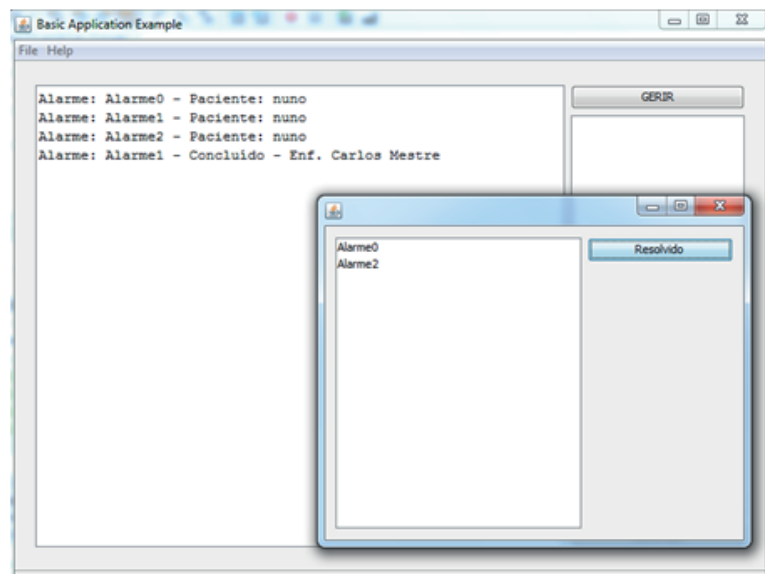


Figura 5.6 – Aplicação: resolução de alarmes

5.1.2 CAN e GSM

Foi efectuado um teste as comunicações CAN, usando os métodos implementados foi enviada uma mensagem CAN "SendCANMessage(0x222, 0,8, buffer)", como se pode observar pela Figura 5.7, fez-se um observação ao osciloscópio que nos permitiu observar a mensagem a ser enviada e o ACK do nó receptor.

Teste ao modem GSM, foi ligado o modem à porta série do computador e através do métodos desenvolvidos "Send(93*****, 'Teste)", foi enviada uma sms de teste com o conteúdo "teste". No telemóvel, podemos ver o resultado na Figura 5.8.

5.1.3 Xbee

Antes de mais foram configurados os módulos Xbee usando o *software* x-ctu, como o mesmo CANAL e PAN ID, activou-se o modo API e escolheu-se a cada módulo um endereço de rede diferente, ver Figura 5.9.

Os módulos Xbee foram configurados com os seguintes parâmetros:

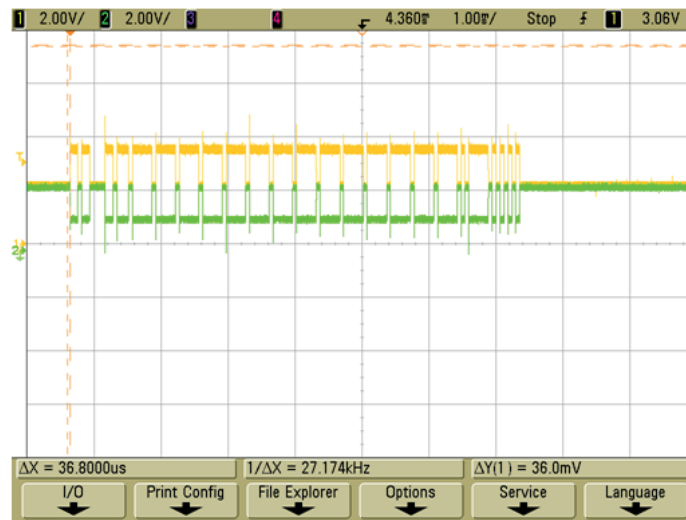


Figura 5.7 – Teste da comunicação CAN



Figura 5.8 – Mensagem recebida no telemóvel

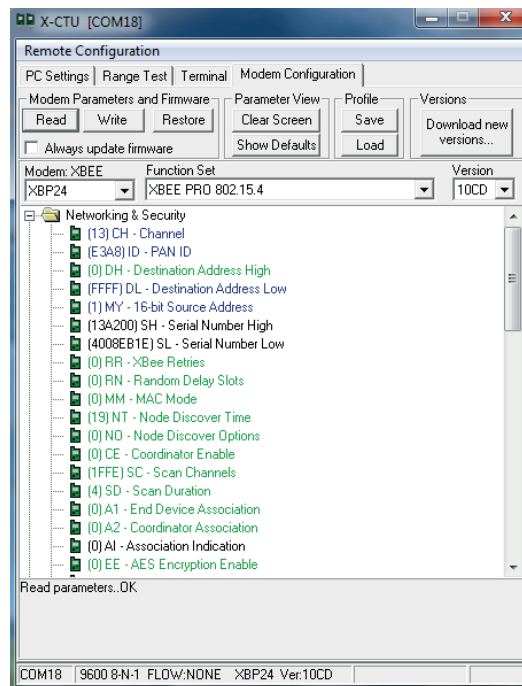


Figura 5.9 – X-CTU software de configuração dos módulos Xbee

- Canal: 13
- PAN ID: E3A8
- API Enable: 1

No teste ao funcionamento dos métodos implementados, foi enviada de forma repetida três mensagens. As mensagens enviadas continham os dados 0, 1, 2, respectivamente. E foi observado no PC, através de um conversor série, a mensagem enviada e a mensagem recebida, como podemos ver na Figura 5.10. Como no *firmware* fizemos a estrutura da *frame*, devemos ter uma mensagem que cumpra a mesma. Pegando na mensagem que está destacada na Figura 5.10 vamos ver se ela está a ser correctamente transmitida.

Mensagem enviada: "7E 00 06 01 00 FF FF 01 02 FD", quando comparada com a estrutura da *frame* de envio 9, temos que o "7E" marca o início da *frame*, o "00 06" é o comprimento da *frame* (excluído o *start delimiter* e o *checksum*), o *byte* "01" significa

que estamos a usar endereços de 16 *bits*, depois surge o *byte* "00" que é obrigatório, os "FF FF" são o endereço de destino ("FF FF" é o endereço de broadcast), depois do endereço de destino é obrigatório o *byte* "01", depois temos o *bytes* de dados, neste caso apenas transmitimos um *byte* com o valor "02" e por último o *checksum*.

Listagem 9 Estrutura da *frame* de envio

Byte_0	0x7E	-> START DELIMITER
Byte_1	0x00	-> LENGTH MSB
Byte_2	size+5	-> LENGTH LSB
Byte_3	0x01	-> 16 bit id (0x00 64bit)
Byte_4	0x00	
Byte_5	(address >>8) & 0x00FF	-> 16 bit address Destino
Byte_6	address & 0x00FF	-> "
Byte_7	0x01	
Byte_8	buffer[i]	-> Dados
Byte_9	checksum	-> Checksum

Mensagem Recebida: 7E 00 06 81 00 01 2D 02 02 4C ,quando comparada com a estrutura da *frame* de chegada [10](#), podemos retirar que o "7E" é marca o início da *frame*, o "00 06" é o comprimento da *frame* (excluído o *start delimiter* e o *checksum*), o *byte* "81" significa que estamos a usar endereços de 16 *bits*, os *Bytes* 4 e 5 são de endereço de origem "00 01", RSSI é a potência "2D", o *byte* de de opção é o "02", o *byte* de dados contém a mesma informação que foi enviada "02" e por último segue-se "4C", que corresponde ao *checksum*.

Listagem 10 Estrutura da *frame* de recepção

Byte_0	0x7E	-> START DELIMITER
Byte_1	0x00	-> LENGTH MSB
Byte_2	size+5	-> LENGTH LSB
Byte_3	0x81	-> 16 bit id (0x80 64bit)
Byte_4	(address >>8) & 0x00FF	-> 16 bit address Origem
Byte_5	address & 0x00FF	-> "
Byte_6	RSSI	-> Potencia (-dbm)
Byte_7	options	-> Opcoes
Byte_8	buffer[i]	-> Dados
Byte_9	checksum	-> Checksum

A transmissão das *frames* foi conseguida, e os métodos implementados estão a funcionar correctamente. O campo RSSI serve para saber a potência com que o sinal de transmissão chega ao nó receptor. E é através do seu valor que iremos calcular a localização da EM.

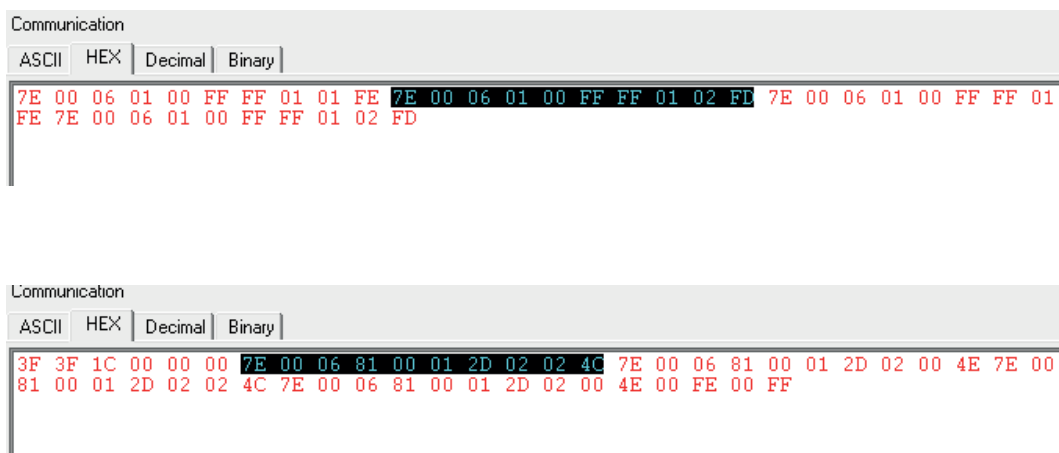


Figura 5.10 – Mensagem transmitida/recebida pelos módulos Xbee

5.2 Cenário de Localização Hospitalar

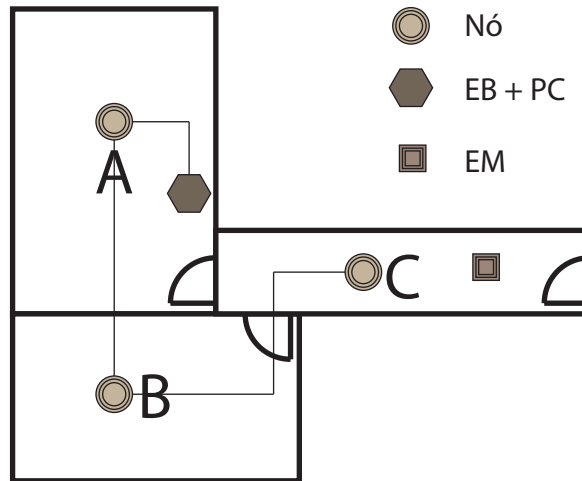


Figura 5.11 – Cenário hospitalar

Com o intuito de se simular uma localização dentro de um hospital, foram escolhidos duas salas (A e B) e um corredor (C), como se pode ver na Figura 5.11. Se fosse num hospital em vez de salas teríamos quartos e o corredor seria equivalente. Nesta simulação foram distribuídos os três nós de modo a cobrir toda a área, assim teremos os nós colocados em A, B e C. A localização da EM será feita através da técnica da Célula de Origem, isto é, a célula que tiver maior potência, é aquela onde a EM se encontra. O fim do corredor, será o limite máximo (LM) que a EM poderá percorrer, isto é, a EM só poderá percorrer as áreas A, B, C assinaladas na Figura 5.12.

Após se efectuarem várias leituras da potência nos locais assinalados, pelas letras, e calculando a média deles obteve-se a tabela 5.1. O alerta emitido após o doente sair da sua área limite, isto é, assim que ultrapassar o LM. Neste caso definimos que quando a EM tiver uma potência inferior a -57dbm da antena A, -88dbm da antena B e inferior a -57dbm da antena C, o utente saiu da área que lhe é reservada, e envia uma mensagem de alerta para a EB, que vai notificar a aplicação.

Dentro da área que os pacientes podem percorrer, não foram assinaladas perdas de frames, durante os testes. Mas de modo a termos a percepção de quais os limites de distância, que o sistema podia proporcionar, foram efectuados leituras fora da área

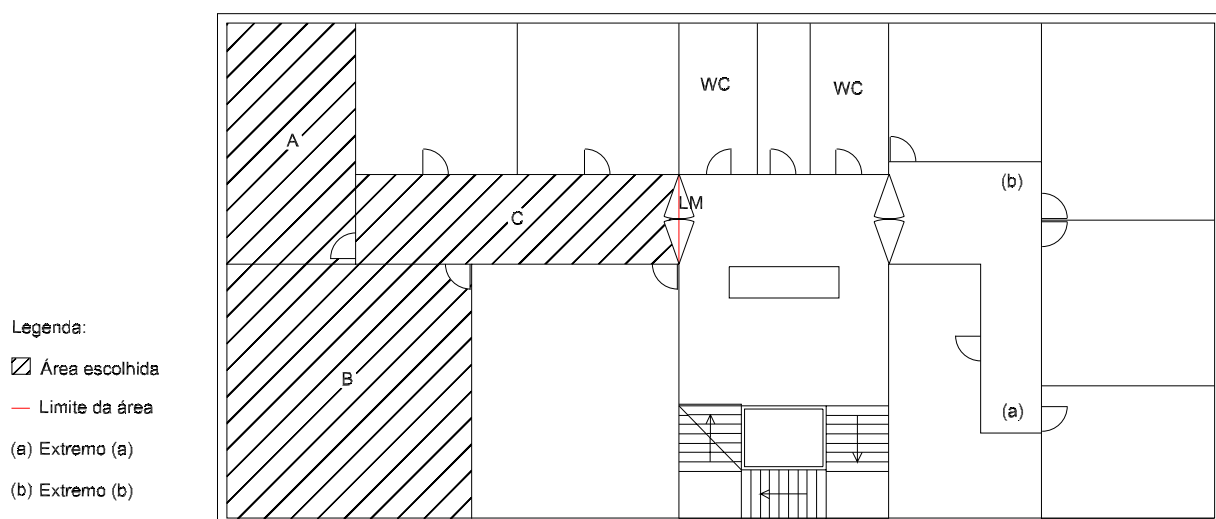


Figura 5.12 – Projecto do cenário

Tabela 5.1 – Potências medidas.

Localização da EM	Potência dos nós (dbm)		
	Nó A	Nó B	Nó C
A	-39	-60	-58
B	-46	-36	-72
C	-64	-76	-39
Limite Máximo	-75	-88	-57

escolhida assim, e no pior caso encontrado, temos os pontos (a) e (b) assinalados na Figura 5.12. Estes pontos encontram-se nas extremidades opostas do edifício, e muito devido aos obstáculos e distância chegamos a valores limite do sinal transmitido pelos nós e a algumas perdas de frames, os valores obtidos estão visíveis na tabela 5.2. O ponto (a), foi o que mais problemas com perdas de frames obteve, pois entre a sala A e o extremo (a) encontra-se o poço do elevador, várias paredes e peças metálicas interferem com o sinal rádio. Para o cálculo das perdas, foram enviados 10 frames, espaçados em 1 segundo, do transmissor. No receptor confirmava-se o número de frames que chegavam.

Tabela 5.2 – Pontos extremos.

Extremos Limite	Potência dos nós (dBm)			Perdas dos nós (%)			Distância (m)		
	Nó A	Nó B	Nó C	Nó A	Nó B	Nó C	Nó A	Nó B	Nó C
(a)	-98	-95	-82	50	30	0	24,5	21	17
(b)	-92	-95	-77	0	20	0	23	22	16

Outro teste efectuado foi ao consumo de bateria e qual a duração da mesma em utilização normal. Para isso, foram medidos os consumos, da bateria em funcionamento total 72,5mA, e com o Xbee em modo *sleep* 15,4mA. Se em metade do tempo o modulo estiver em *sleep*, teremos um consumo médio de $(72,5+15,4)/2=43,95\text{mA}$. De forma simples se dividirmos a capacidade pelo consumo médio, obteremos o número de horas que a EM funcionará, sem que seja necessário recarregar a bateria. Temos então que $2000/43,95=45\text{h}$ e 30 minutos, que a bateria demorará a descarregar.

6

Conclusões e Trabalho Futuro

Mecanismos de apoio à saúde que possam auxiliar o pessoal médico e enfermeiros, de algumas das suas funções, estão cada vez mais presentes. Este tipo de sistemas, podem fazer face à falta de algum pessoal médico e enfermeiros, que devido a cortes na saúde por parte dos governos podem vir a diminuir, para além disso faz com que a monitorização dos pacientes, seja mais continuada e/ou permanente, e remotamente podendo mesmo ser monitorizados a partir dos seus lares.

Este trabalho visa essencialmente auxiliar o pessoal médico e enfermeiros, na tarefa vigilância dos pacientes e promover uma vigilância centralizada e à distância. Foi necessário para a sua implementação, fazer-se um estudo aprofundado de várias tecnologias e rever outras, trabalhar com diversos programas e linguagens, e fazer um levantamento de quais os meios disponíveis para realizar o mesmo.

O *core* do sistema desenvolvido, os seus dispositivos necessários e as aplicações de *software* estão concluídas, poderão no entanto vir a ser alvo de *updates*. Pois como todos os sistemas, só com bastantes testes *in loco* e com a ocorrência de possíveis erros, se vão amadurecendo.

O sistema é capaz de realizar a transferência de dados de ponto a ponto e da notificação dos assistentes em tempo real, e faltará adicionar a parte sensorial ao sistema,

cujos testes foram realizados por simulação. No futuro também era importante fazer os testes em ambiente hospitalar, pois é mesmo para esse fim que se destina.

É necessário no futuro saber, quantas horas por dia a EM funcionará num cenário real, para poder-se estimar melhor a duração da bateria e proceder-se se necessário a melhorias de eficiência, pois os gastos para um dispositivo móvel são ainda bastante elevados.

Um dos trabalhos futuros seria exactamente baixar o clock do microprocessador, isto, quando o xbee está em sleep. Arranjar também soluções de modo a ter o mínimo tempo os módulos Xbee activos, pois consomem muita energia, seja à custa de algoritmos baseados na rotina, ou sensores, como aceleramentos, tudo isto promovendo a eficiência energética e prolongando a duração da bateria. Também não se devia deixar de fazer um estudo com uma área maior, pois um dos motivos para que não houvesse perdas na área seleccionada, foi o de esta ser pequena. Outro estudo seria de analisar a interferência das pessoas, no sistema, pois todos os dados foram obtidos, com a área vazia de pessoas.

Por fim, o sistema apesar de bastante completo ainda necessita de amadurecimento e de testes em diferentes ambientes, para que possam a vir a ser corrigidos os possíveis erros e para que se possa afinar e melhorar os dispositivos. Do ponto de vista tecnológico, a evolução das tecnologias e a diminuição do tamanho e do consumo dos módulos utilizados, também são fundamentais para se ter um sistema mais económico e mais eficiente.

Referências bibliográficas

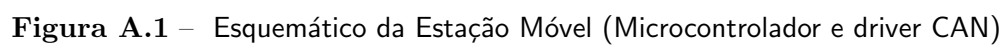
- Bosch (1991). Can specification version 2.0. Robert Bosch GmbH, Postfach 50, D-7000 Stuttgart 1. [31](#), [32](#)
- Bosch (2010). The configuration of the can bit timing. CAN in Automation. [29](#), [30](#)
- Chang, Y., Chen, C., Chou, L., and Wang, T. (2008). A novel indoor wayfinding system based on passive rfid for individuals with cognitive impairments. Second International Conference on Pervasive Computing Technologies for Healthcare. [xix](#), [10](#), [11](#)
- CiA (1991a). Can specification 2.0, part a. CAN in Automation. [28](#), [29](#)
- CiA (1991b). Can specification 2.0, part b. CAN in Automation. [28](#), [29](#)
- Corrigan, S. (2008). Controller area network physical layer requirements. Texas Instruments. [27](#)
- Costa, N. M. (2005). A evolução demográfica na europa: Uma visão dos próximos 50 anos. x colóquio ibérico de geografia. Universidade de Évora. [2](#)
- ETSI (1996). Gsm 11.14 version 5.2.0, digital cellular telecommunications system

- (phase 2+); specification of the sim application toolkit for the subscriber identity module - mobile equipment (sim - me) interface. [22](#)
- Eurostat (2008). Income and living conditions in europe. Eurostat, European Commission. [2](#)
- Gu, Y., Lo, A., and Niemegeers, I. (2009). A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, p. 16-17. [6](#)
- Hightower, J. and Borriello, G. (2001). Location sensing techniques, technical report uw cse 2001-07-30. Department of Computer Science and Engineering, University of Washington. [6](#)
- Hightower, J. and Borriello, G. (2002). Gsm architecture, training document. Finland, jan. [xix](#), [20](#), [22](#), [23](#), [25](#)
- IEEE (2001). Ieee standard 802 - local and metropolitan area networks. [13](#)
- IEEE (2002). Ieee standard 802.15.1 - wireless medium access control (mac) and physical layer (phy) specification for wireless personal area networks (lr-wpans). [17](#)
- IEEE (2003). Ieee standard 802.15.4 - wireless medium access control (mac) and physical layer (phy) specification for low-rate wireless personal area networks (lr-wpans). [15](#)
- IEEE (2007). Ieee standard 802.11 - wireless lan medium access control (mac) and physical layer (phy) specification. [14](#)
- Marco, A., Casas, R., Falco, J., Gracia, H., Artigas, J., and Roy, A. (2008). Location-based services for elderly and disabled people. *Computer Communications Volume* 31. [11](#)
- Marques, H. (2011). Internamentos públicos aumentam 32sobem 25LUSA, Lisboa, 22 set. [3](#)

- Metrôlho, J. C. M. M. (1999). Rede can para comando de actuadores em estufas agrícolas. Departamento de Electrónica Industrial, Universidade do Minho. [34](#), [36](#), [37](#), [38](#)
- Pahlavan, K., Li, X., and Makela, J. (2002). Indoor geolocation science and technology. IEEE Communications Magazine Volume 40. [xix](#), [10](#)
- Pfeiffer, O., Ayre, A., and Keydel, C. (2008). Embedded networking with can and canopen. Copperhill Media. [42](#)
- Rappaport, T. (1996). Wireless communications: Principles and practice. Prentice Hall PTR. [24](#)
- Schiler, J. (2003). Mobile communications. Pearson Education Limited. [21](#), [22](#), [23](#), [24](#)
- Serôdio, C., Mestre, P., Monteiro, J. L., and Couto, C. A. C. (2010). Pervasive wsn based solutions applied to health and life-support systems, citab-utad. University of UTAD. [xix](#), [9](#)
- Silva, P. M. M. A., Paralta, M., Caldeirinha, R., Rodrigues, J., and Serôdio, C. M. (2009a). Traceme - indoor real-time location system. CITAD-UTAD, University of UTAD. [xix](#), [8](#)
- Silva, P. M. M. A., Pinto, H., Serôdio, C. M., Monteiro, J. L., and Couto, C. A. C. (2009b). A multi-technology framework for lbs using fingerprinting. CITAD-UTAD, University of UTAD. [9](#)
- Voss, W. (2005). A comprehensible guide to controller area network. Copperhill Technologies Corporation. [31](#), [32](#), [33](#), [34](#), [36](#), [37](#), [38](#), [42](#)
- W.C. Xuesong, S. and Ming, L. (2008). Wireless sensor network for resources tracking at build construction sites. Tsinghua Science and Technology, vol. 13, no. 2, p. 78-83. [5](#)

- Yan, H., Xu, Y., Gidlund, M., and Nohr, R. (2008). An experimental study on home-wireless passive positioning. Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008). [xix](#), [12](#)
- Youssef, M., Agrawala, A., and Shankar, U. (2003). Wlan location determination via clustering and probability distributions. IEEE Int. Conf. Pervasive Comput. Commun., Mar, p 143-151. [xix](#), [7](#)
- Youssef, M. and Agrawala, A. K. (2004). Handling samples correlation in the horus system. IEEE INFOCOM 2004, Hong Kong, vol. 2, Mar., p 1023-1031. [xix](#), [7](#)

Anexo A1 - Esquemático da Estação Móvel



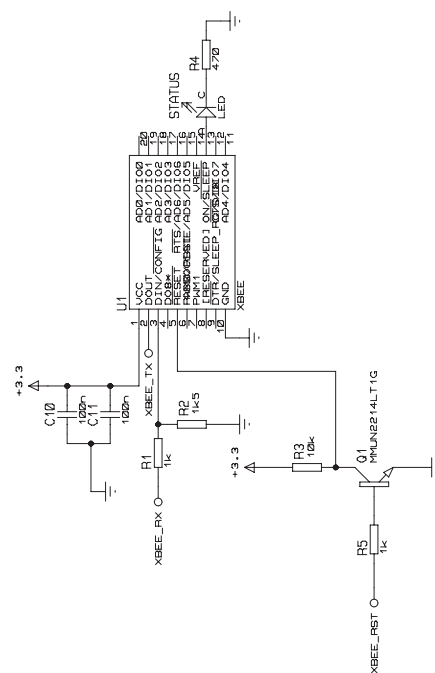


Figura A.2 – Esquemático da Estação Móvel (Módulo Xbee)





Anexo A2 - Esquemático da Estação Base

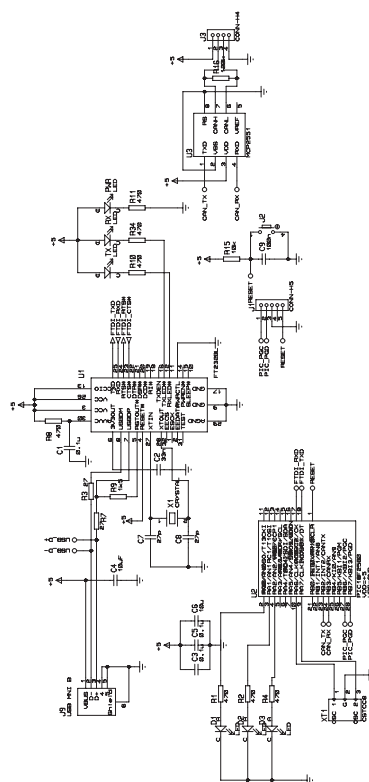


Figura B.1 – Esquemático da Estação Base

