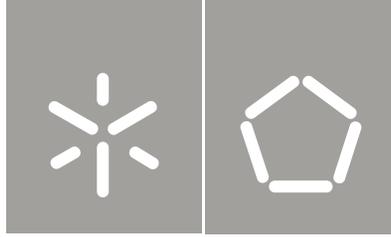




Universidade do Minho
Escola de Engenharia

Sébastien André Leroux

Convergência entre dispositivos móveis
e redes de sensores sem fios



Universidade do Minho
Escola de Engenharia

Sébastien André Leroux

Convergência entre dispositivos móveis
e redes de sensores sem fios

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efectuado sob a orientação do
Professor Doutor José Augusto Afonso

Junho de 2013

Resumo

As redes de sensores sem fios, apesar de terem um papel muito importante em áreas tão distintas como a saúde, o ambiente, a domótica, etc., possuem várias restrições que as impedem de serem mais eficientes e mais abrangentes. Esta dissertação tem como objetivo analisar a forma como algumas destas limitações podem ser ultrapassadas através do uso de dispositivos móveis, implementando módulos exemplificativos da integração destas duas tecnologias.

O primeiro módulo é um programa que permite aumentar o alcance de uma rede de sensores sem fios, aproveitando a variedade de interfaces de comunicação que um dispositivo apresenta. Este software foi implementado num smartphone HTC utilizando o sistema operativo Windows Mobile.

O segundo módulo é uma aplicação que exemplifica a complementaridade entre as redes de sensores sem fios e os dispositivos móvel. A aplicação é um sistema de deteção de quedas, em que o dispositivo móvel lê os dados provenientes da rede de sensores e aciona um alarme em caso de queda do utilizador. O dispositivo sensor utilizado é baseado num acelerómetro de três eixos. O algoritmo de deteção de quedas e respetiva interface com o utilizador foram implementados em Matlab. Os resultados obtidos nos testes efetuados sugerem que o sistema pode funcionar com poucas falhas na monitorização de pessoas em que as movimentações são limitadas, por exemplo, no caso de pessoas idosas. Já para pessoas mais jovens, o algoritmo implementado pode não ser suficiente.

O terceiro módulo é uma placa de circuito impresso que visa proporcionar uma interface de comunicação entre dispositivos de redes de sensores baseados na norma IEEE 802.15.4 e dispositivos móveis, uma vez que estes últimos não integram esta interface de raiz. Este módulo é baseado no circuito integrado CC2430, que integra o microcontrolador e o transceiver no mesmo chip. O desenvolvimento deste módulo não chegou ao fim no tempo planeado. De qualquer forma, outros módulos baseados num circuito integrado mais recente (CC2530) surgiram para colmatar a lacuna identificada no início do desenvolvimento deste módulo.

Abstract

Wireless sensor networks, despite having a very important role in areas such as health, environment, home automation, etc., have several restrictions that prevent them from being more efficient and widespread. This dissertation aims to examine how some of these limitations can be overcome through the use of mobile devices and the implementation of modules that exemplify the integration of these technologies.

The first module is a program that allows to extend the range of a wireless sensor network, taking advantage of the variety of communication interfaces that a mobile device possesses. This software has been implemented using a HTC smartphone with the Windows Mobile operating system.

The second module is an application that illustrates the complementarity between wireless sensor networks and mobile devices. This application is a fall detection system where the mobile device reads the data received from the sensor network and triggers an alarm when the user falls. The sensor device used is based on a three-axis accelerometer. The algorithm for detection of falls and respective user interface were both implemented in Matlab. The results from the tests suggest that the system can operate with few failures in the monitoring of individuals whose movements are limited, for example, in the case of elderly people. For younger people, though, the implemented algorithm may not be enough.

The third module is a printed circuit board which aims to provide a communication interface between devices of sensor networks based on IEEE 802.15.4 and mobile devices, since the latter do not support 802.15.4 communication natively. This module is based on CC2430 integrated circuit, which integrates the microcontroller and the transceiver on the same chip. The development of this module have not finished in the scheduled time. Anyway, other modules based on an newer integrated circuit (CC2530) have emerged to fill the gap identified in the early development of this module.

CONTEÚDO

1	Introdução	1
1.1	Redes de sensores sem fios	1
1.2	Vantagens da integração das duas tecnologias	2
1.3	Objetivos	3
1.4	Estrutura da dissertação	3
2	Trabalho relacionado	5
2.1	Área da saúde	6
2.1.1	Projeto CodeBlue	6
2.1.2	Projeto MobiHealth	8
2.1.3	Projeto Aid-N	11
2.1.4	Empresa eHit	12
2.2	Área da monitorização ambiental	14
2.2.1	Projeto NoiseTube	14
2.2.2	Projeto N-Smarts	15
2.2.3	Projeto SIREN	18
2.3	Outras áreas de aplicação	19
2.3.1	Multimedia Guidebook	19
2.3.2	Projeto Bikenet	21
2.3.3	Projeto Motetrack	24
3	Desenvolvimento	25
3.1	Projeto QWisNet	26
3.2	<i>Gateway</i> inter-redes	27
3.2.1	Objetivo	27
3.2.2	Arquitetura	27
3.2.3	Conceção	28
3.3	Deteção de quedas	34

3.3.1	Objetivo	34
3.3.2	Trabalho relacionado	34
3.3.3	Arquitetura	35
3.3.4	Conceção	35
3.4	Hardware de interface	39
3.4.1	Objectivo	39
3.4.2	Análise do problema e montagem electrónica	39
4	Resultados e discussão	45
4.1	<i>Gateway</i> inter-redes	45
4.2	Deteção de quedas	46
4.2.1	Cenário com ADLs	46
4.2.2	Situação de queda	49
4.2.3	Eficácia do sistema	52
4.3	Hardware de interface	54
5	Conclusão	56
	Apêndice A Montagem electrónica	62

LISTA DE FIGURAS

1.1	Cenário de monitorização normal.	2
2.1	Módulos de hardware desenvolvidos (Fonte: [1]).	7
2.2	Arquitectura do software dos nós (Fonte: [1]).	8
2.3	Programa de monitorização de pacientes (Fonte: [2]).	9
2.4	Exemplos de aplicação instaladas na MBU (Fonte: [3]).	10
2.5	Dispostivos móveis usados como MBU (Fonte: [3]).	10
2.6	Arquitectura do sistema MiTags (Fonte: [4]).	11
2.7	Arquitectura da plataforma eHit Health Gateway (Fonte: [5]).	13
2.8	Ligações entre dispositivos via eLink (Fonte: [5]).	14
2.9	<i>Screenshot</i> da aplicação do telemóvel (Fonte: [6]).	15
2.10	Mapas de poluição (Fonte: [6]).	16
2.11	Módulos desenvolvidos no âmbito do projeto N-Smarts (Fonte: [7]).	16
2.12	Módulo integrado com um telemóvel já existente (LG VX9800) (Fonte: [7]).	17
2.13	Centro de controlo da sala de teste (Fonte: [7]).	17
2.14	Visualização dos resultados via Google Earth (Fonte: [7]).	17
2.15	GUI (Graphical User Interface) disponibilizada pelo Siren (Fonte: [8]).	19
2.16	Dispositivo móvel com ponteiro associado (Fonte: [9]).	20
2.17	Menu de interacção com o utilizador (Fonte: [9]).	21
2.18	Tipos de interacção possíveis.	22
2.19	Portal Web disponibilizado aos utilizadores do projeto BikeNet (Fonte: [10]).	22
2.20	Arquitectura de software dos vários componentes do projeto BikeNet.	23
2.21	Cenário de teste do projeto MoteTrack (Fonte: [11]).	24
3.1	Estrutura lógica do <i>gateway</i>	28
3.2	Formulário de inserção de formato de mensagem.	29
3.3	Formulário de inserção de um destinatário SMS.	29
3.4	Formulário de inserção de um destinatário Wi-Fi.	30

3.5	Formulário de inserção de destinatário Bluetooth.	30
3.6	Formulário de inserção de uma regra de encaminhamento.	31
3.7	Estrutura da base de dados.	33
3.8	Arquitetura inicial do sistema de deteção de quedas.	35
3.9	Arquitetura final do sistema de deteção de quedas.	35
3.10	Parado de pé.	36
3.11	Andar.	36
3.12	Vários movimentos seguidos (entre os quais andar, sentar, levantar, abaixar, etc.).	37
3.13	Queda normal.	37
3.14	Queda de cama.	37
3.15	Interface gráfica da aplicação implementada via Matlab.	38
3.16	XBee Programmable USB Stick (Fonte: [12]).	39
3.17	Módulo CC2430EM.	40
3.18	Ligação entre o FT232 e o CC2430.	42
3.19	Representação gráfica das equações de adaptação entre os 2 módulos.	43
3.20	Divisor de tensão.	43
3.21	Representação da PCB pela ferramenta PADS Layout.	44
4.1	Visualização dos dados provenientes dos sensores (sem qualquer alteração).	46
4.2	Aplicação do filtro mediana de ordem 13.	47
4.3	Aplicação do filtro mediana de ordem 3.	47
4.4	Aplicação do filtro mediana de ordem 20.	48
4.5	Aplicação do filtro passa-alto sobre uma janela de 0.8s.	48
4.6	Aplicação do filtro passa-alto sobre uma janela de 0.1 s.	49
4.7	Aplicação do filtro passa-alto sobre uma janela de 2 s.	49
4.8	Apresentação gráfica dos 3 integrais.	50
4.9	Apresentação gráfica da soma dos 3 integrais.	50
4.10	Visualização dos dados provenientes dos sensores (sem qualquer alteração).	50
4.11	Aplicação do filtro mediana de ordem 13.	51
4.12	Aplicação do filtro passa-alto sobre uma janela de 0.8s.	51
4.13	Apresentação gráfica dos 3 integrais.	51
4.14	Apresentação gráfica da soma dos 3 integrais.	52
4.15	Apresentação gráfica da soma dos 3 integrais em situações de movimentações bruscas.	52
4.16	PCB implementada.	55
A.1	Ligações relativas ao conector USB.	62
A.2	Ligações relativas ao FT232R.	63
A.3	Ligações dos condensadores ao FT232R.	63
A.4	Divisor de tensão entre o FT232R e o CC2430.	63
A.5	Ligações relativas ao CC2430.	64
A.6	Visão global da montagem eletrónica.	64

LISTA DE TABELAS

3.1	Características DC do integrado FT232R (Fonte: [13]).	41
4.1	Máximos absolutos calculados (ordem mediana = 3).	53
4.2	Máximos absolutos calculados (ordem mediana = 5).	53
4.3	Matriz de classificação obtida nos testes de desempenho do sistema de detecção de quedas.	54

LISTA DE ACRÓNIMOS

<i>ADL</i>	Activity of Daily Living
<i>ADMR</i>	Adaptive Demand-Driven Multicast Routing
<i>API</i>	Application Programming Interface
<i>BAN</i>	Body Area Network
<i>CDMA</i>	Code Division Multiple Access
<i>ECC</i>	Elliptic Curve Cryptography
<i>ECG</i>	Electrocardiografia
<i>eLPRT</i>	Enhanced Low Power Real Time
<i>EMG</i>	Electromiografia
<i>FCT</i>	Fundação para a Ciência e a Tecnologia
<i>FIR</i>	Finite Impulse Response
<i>GPRS</i>	General Packet Radio Service
<i>GPS</i>	Global Positioning System
<i>GSM</i>	Global System for Mobile Communications
<i>GSR</i>	Galvanic Skin Response
<i>GUI</i>	Graphical User Interface
<i>HTTP</i>	HyperText Transfer Protocol
<i>HTTPS</i>	HyperText Transfer Protocol Secure
<i>IP</i>	Internet Protocol

<i>LAN</i>	Local Area Network
<i>PCB</i>	Printed Circuit Board
<i>PDA</i>	Personal Digital Assistant
<i>PIN</i>	Personal Identification Number
<i>PKI</i>	Public Key Infrastructure
<i>QWiSNet</i>	Quality of Service in Wireless Sensor and Actuator Networks
<i>RSA</i>	Rivest Shamir Adleman
<i>RSSI</i>	Received Signal Strength Indicator
<i>SIG</i>	Special Interest Group
<i>SMS</i>	Short Message Service
<i>UMTS</i>	Universal Mobile Telecommunications System
<i>USB</i>	Universal Serial Bus
<i>WBAN</i>	Wireless Body Area Network
<i>WLAN</i>	Wireless Local Area Network
<i>WSN</i>	Wireless Sensor Network

CAPÍTULO 1

INTRODUÇÃO

Imagination is more important
than knowledge. For while
knowledge defines all we currently
know and understand,
imagination points to all we
might yet discover and create.

Albert Einstein

Conteúdo

1.1	Redes de sensores sem fios	1
1.2	Vantagens da integração das duas tecnologias	2
1.3	Objetivos	3
1.4	Estrutura da dissertação	3

Este capítulo tem como objetivo introduzir conceitos e ideais basilares a esta dissertação.

Na primeira secção, é abordada a temática das redes de sensores sem fios (WSN - *Wireless Sensor Network*), explicando a sua utilidade, o seu funcionamento e salientando os seus pontos a favor e os seus pontos contra.

Na segunda secção, são explicadas as razões que explicam as vantagens da junção das tecnologias "Redes de sensores sem fios" e "Dispositivos móveis".

1.1 Redes de sensores sem fios

As redes de sensores sem fios ([14], [15]) são uma tecnologia cada vez mais presente no quotidiano, e com aplicações em vários domínios como, por exemplo, a monitorização hospitalar, a monitorização ambiental, etc. Este tipo de redes, apesar de terem várias vantagens, entre as quais a sua fácil instalação e a sua capacidade de adaptação em casos de falha de um nó, tem algumas

desvantagens. O seu alcance limitado e o facto de os nós geralmente dependerem de uma bateria para obtenção de energia são alguns dos principais pontos contra esta tecnologia. A necessidade de maximizar a autonomia da bateria obriga a que os nós dos sensores não tenham muita capacidade de processamento de dados, limitando-se na maior parte dos casos a adquirir e enviar, estando assim dependentes de uma estação base, que recebe e processa estes dados. Em situações estáticas, em que todos os elementos das redes estão confinados a um único local, não há muitos problemas, pois pode sempre existir um computador (de secretária, ou portátil), para recolher os dados. Porém, em casos em que a mobilidade é obrigatória, a recolha de dados é mais complicada.

1.2 Vantagens da integração das duas tecnologias

Hoje em dia, os dispositivos móveis apresentam capacidade de processamento cada vez mais elevada. Aliando esta capacidade de processamento às capacidades de comunicação destes dispositivos, nota-se claramente uma complementaridade entre a tecnologia dos dispositivos móveis e a tecnologia das redes de sensores sem fios. Um exemplo é apresentado na figura 1.1. Neste cenário, tem-se uma pessoa a ser monitorizada (por exemplo, a esta pessoa está ligado um nó sensor que monitoriza o eletrocardiograma) e um computador, ao qual está ligado um outro nó da rede de sensores, que recolhe os dados. Neste cenário, a pessoa, para ser monitorizada, precisa de estar ao alcance do computador. Existe uma limitação ao nível da mobilidade.

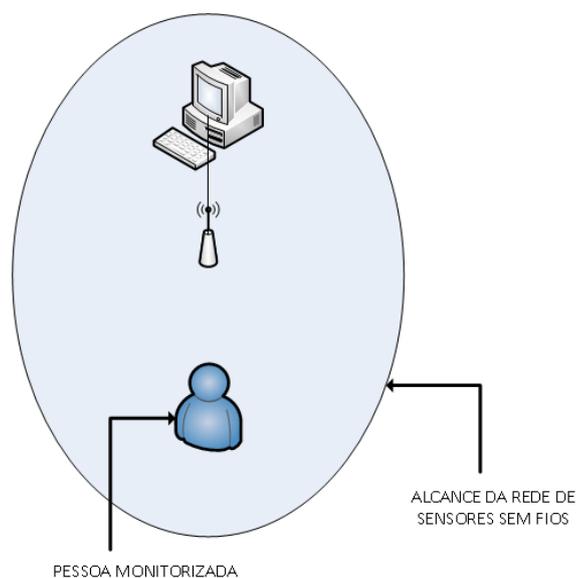


Figura 1.1: Cenário de monitorização normal.

Ao utilizar um dispositivo móvel como integrante da rede de sensores, a limitação relativa à mobilidade é ultrapassada (não é totalmente ultrapassada, pois podem existir situações em que o dispositivo móvel não tem qualquer tipo de ligação de dados disponíveis, mas, na maior parte das situações, a probabilidade de inexistência de conectividade, seja ela Wi-Fi, GSM (*Global System for Mobile Communications*), etc., é muito reduzida). Para além disso, podem-se aproveitar as capacidades do dispositivo para as mais diversas tarefas (armazenamento, apresentação de dados ao utilizador, etc.).

1.3 Objetivos

Esta dissertação tem como objetivo exemplificar a forma como duas tecnologias tão distintas como as WSN e os dispositivos móveis podem ser integrados de forma a implementar sistemas com alguma utilidade para as mais diversas áreas. Para este fim, são analisados sistemas já existentes, é apresentado um sistema genérico que pode servir de base a outros sistemas (*gateway* inter-redes), é apresentado um exemplo prático de integração das duas tecnologias (sistema de detecção de quedas), e é descrito o projeto de uma placa de circuito impresso para possibilitar a comunicação entre essas duas tecnologias (módulo de *hardware*).

1.4 Estrutura da dissertação

Esta dissertação divide-se em quatro capítulos, para além da introdução.

No próximo capítulo, são analisados alguns sistemas que aproveitam a complementaridade entre WSNs e dispositivos móveis. Tentou-se abranger o maior número de áreas distintas, de modo a ser possível perceber a potencialidade de tais sistemas.

O terceiro capítulo começa por apresentar o projeto de investigação no âmbito do qual este trabalho foi desenvolvido. A seguir, este capítulo apresenta os módulos desenvolvidos para responder aos objetivos propostos para esta dissertação. O primeiro módulo, mais genérico, exemplifica a forma como os dispositivos móveis permitem ultrapassar uma das maiores limitações das WSN, que é o curto alcance através do desenvolvimento de um *gateway* inter-redes. O segundo módulo, mais prático, apresenta um caso prático de utilização da integração entre WSNs e dispositivos móveis através do desenvolvimento de um sistema de detecção de quedas. O terceiro módulo consiste num módulo de hardware para possibilitar a comunicação entre o dispositivo móvel e a rede de sensores sem fios.

No quarto capítulo são apresentados os resultados obtidos com os três módulos desenvolvidos e a respetiva discussão.

Finalmente, o quinto capítulo apresenta as conclusões e sugestões de trabalho futuro.

CAPÍTULO 2

TRABALHO RELACIONADO

Sometimes when you innovate, you make mistakes. It is best to admit them quickly, and get on with improving your other innovations.

Steve Jobs

Conteúdo

2.1	Área da saúde	6
2.1.1	Projeto CodeBlue	6
2.1.2	Projeto MobiHealth	8
2.1.3	Projeto Aid-N	11
2.1.4	Empresa eHit	12
2.2	Área da monitorização ambiental	14
2.2.1	Projeto NoiseTube	14
2.2.2	Projeto N-Smarts	15
2.2.3	Projeto SIREN	18
2.3	Outras áreas de aplicação	19
2.3.1	Multimedia Guidebook	19
2.3.2	Projeto Bikenet	21
2.3.3	Projeto Motetrack	24

Este capítulo tem como objetivo fornecer exemplos de trabalho relacionado em que a integração das redes de sensores sem fios com os dispositivos móveis permite criar sistemas de grande utilidade em áreas tão distintas como a saúde e a monitorização ambiental, entre outras.

2.1 Área da saúde

2.1.1 Projeto CodeBlue

O projeto CodeBlue [16], desenvolvido na Universidade de Harvard, consiste na implementação de uma plataforma (*hardware* e *software*) para o uso de redes de sensores sem fios na área da saúde.

Apesar de os requisitos deste tipo de plataformas variarem muito consoante o cenário onde se encontra inserida (uma rede de sensores implementada de forma *ad-hoc* em situações de emergência tem requisitos muito diferentes de uma rede de sensores fixa num hospital, por exemplo), os autores selecionaram os seguintes requisitos como base para a implementação do projeto:

- **Wearable sensor platform:** Os dispositivos usados precisam de ser de dimensões relativamente reduzidas;
- **Fiabilidade das comunicações:** A congestão da rede ou a mobilidade não podem ter consequências negativas na fiabilidade das comunicações (perda de pacotes, etc.);
- **Receção múltipla:** Os dados de um dado paciente podem ser recebidos por mais que um destinatário (a rede tem de suportar funções de *multicast*);
- **Mobilidade dos dispositivos:** A rede tem de ser dinâmica, procurando sempre encontrar rotas mais favoráveis aquando da movimentação de médicos, pacientes ou outra pessoa com ligação à rede;
- **Segurança das comunicações:** A lei americana obriga que os dispositivos médicos cumpram os requisitos especificados no HIPAA (*Health Insurance Portability and Accountability Act*) de 1996.

Neste projeto, foram desenvolvidos vários tipos de sensores (oxímetro, ECG e sensor de movimento), para serem utilizados juntamente com as plataformas comerciais Mica2, MicaZ e Telos. Uma vez que os investigadores do projeto consideram que essas plataformas funcionam bem em cenários de investigação, mas têm muitas desvantagens em cenários reais (dimensões dos módulos por causa da bateria, etc.), também foi desenvolvida uma plataforma de suporte aos sensores mais leve, de modo a ser utilizada em módulos de monitorização de pacientes de um modo não-invasivo.

O oxímetro de pulso (ver Figura 2.1a) utiliza o BCI Medical Micro-Power Pulse Oximeter, da empresa Smiths Medical PM, por ser o módulo de dimensões mais reduzidas e com consumo mais baixo encontrado.

O sensor de ECG (eletrocardiograma) (ver Figura 2.1b) baseia-se no amplificador de instrumentação INA321, sendo necessário ligar 3 fios ao corpo do paciente: uma para referência e dois para medir a atividade cardíaca.

O sensor de movimento (ver Figura 2.1c) baseia-se em três “sub-sensores”:

- Um acelerómetro de 3 eixos (LIS3L02AQ da empresa STMicroelectronics);
- Um giróscopo de 1 eixo (ADXRS300 da Analog Devices);
- Uma sensor EMG (eletromiograma) (para monitorização da atividade elétrica dos músculos).

A plataforma de suporte aos sensores (ver Figura 2.1d), à qual foi dado o nome Pluto, baseia-se na plataforma Telos Rev B, e inclui um micro-controlador MSP430 da Texas Instruments e um módulo CC2420 para ligação rádio. As dimensões da plataforma correspondem a 70% da plataforma Telos original, tendo sido diminuído o seu peso em metade, muito devido ao uso de uma bateria recarregável via USB (*Universal Serial Bus*), em vez das pilhas usadas pela plataforma Telos.

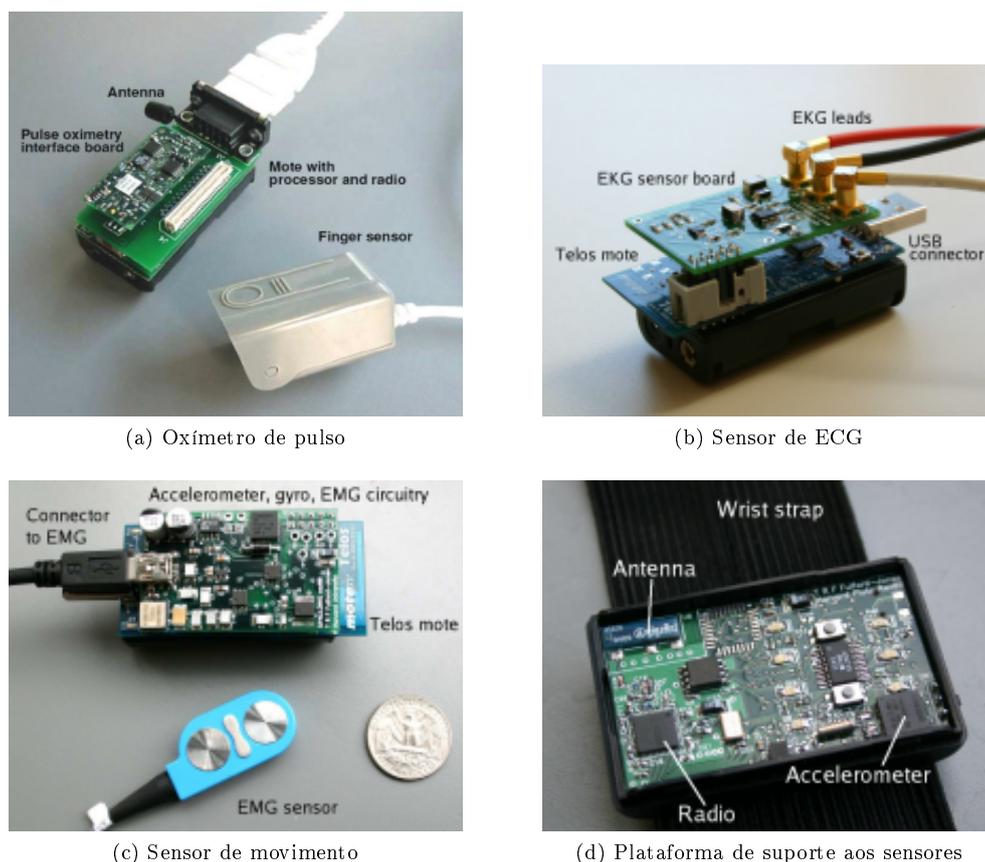


Figura 2.1: Módulos de hardware desenvolvidos (Fonte: [1]).

Para a fiabilidade das comunicações, os autores utilizam transmissões redundantes e técnicas de codificação para a reconstrução dos dados mesmo em caso de perda de pacotes. Também é abordado o uso de controlo de potência de transmissão, de modo a diminuir a interferência.

No que diz respeito à receção múltipla de dados e a mobilidade dos dispositivos, foi implementado um protocolo *multicast* do tipo *publish/subscribe* chamado ADMR (*Adaptive Demand-Driven Multicast Routing*), por ser um protocolo simples e muito estudado em simulações. A implementação deste algoritmo em TinyOS foi uma tarefa complicada (foi necessária uma otimização das tabelas de rotas e vizinhos para minimizar o espaço ocupado na memória dos nós, etc.). A arquitetura do software existente nos nós pode ser vista na Figura 2.2 (o módulo TinyADMR é o responsável pelo protocolo *multicast*).

Por último, para segurança das comunicações, a utilização de criptografia de chave pública, apesar de ser aquela que mais se adequa às necessidades das comunicações deste tipo de sistema, é muito limitada nas plataformas utilizadas (por exemplo, os 4 Kbytes de memória da plataforma

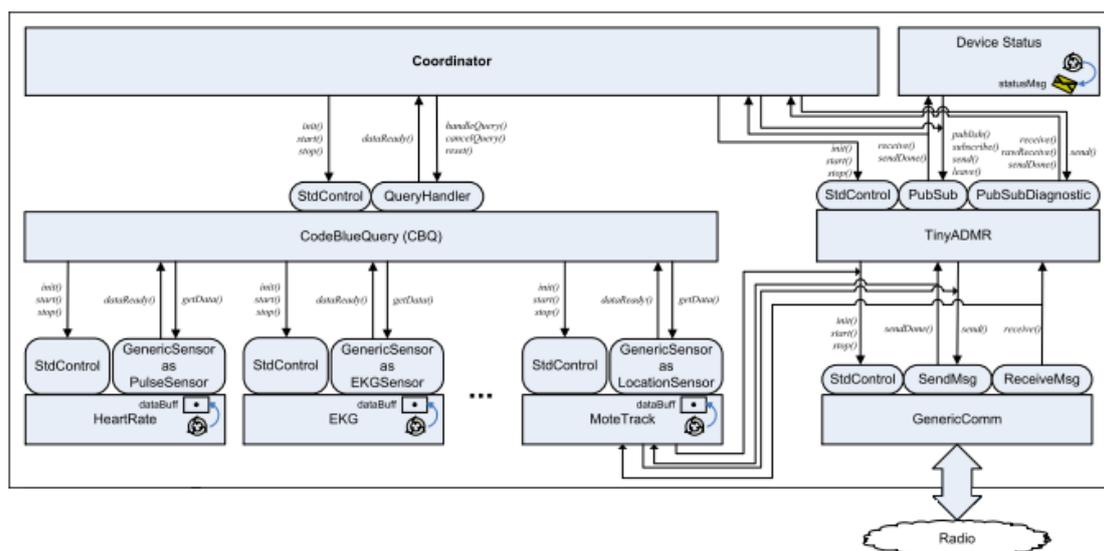


Figura 2.2: Arquitectura do software dos nós (Fonte: [1]).

Mica2 permitem apenas armazenar algumas chaves RSA (Rivest Shamir Adleman) de 1024 bits). Os autores estudaram então a criptografia de curva elíptica (ECC - *Elliptic Curve Cryptography*), que é mais eficiente do ponto de vista computacional, e mais leve a nível de espaço ocupado pelas chaves (uma chave ECC de 163 bits equivale a uma chave RSA de 768 bits). A implementação da ECC permite gerar chaves em cerca de 35 segundos. Um dos desenvolvimentos futuros passa pelo uso dos dispositivos móveis (PDAs (*Personal Data Assistant*), etc.) para tarefas relativas à segurança computacionalmente mais pesadas.

Este projeto também implementou um sistema de localização especialmente desenhado para uso em cenários de emergência (ver descrição mais à frente).

Para testes globais ao sistema, foi implementada uma aplicação, a correr num PDA iPaq com Windows CE, que permite monitorizar vários pacientes, sendo possível configurar vários parâmetros, desde a frequência de atualização dos dados, até a configuração de alarmes (visuais e sonoros) em situações críticas. Os dados recolhidos pelo PDA podem ser passados para um programa que trata do armazenamento dos dados relativos aos cuidados de saúde dos pacientes, como é o caso do iRevive da empresa 10Blade. Um *screenshot* da aplicação pode ser visto na Figura 2.3.

De salientar que grande parte das aplicações desenvolvidas neste projeto (programas em nesC para os sensores, aplicações para PDA, entre outras), assim como esquemas de hardware, encontram-se disponíveis no site do projeto.

2.1.2 Projeto MobiHealth

O projeto MobiHealth [17], financiado pela União Europeia, oferece uma plataforma de monitorização de pacientes em redes UMTS (*Universal Mobile Telecommunications System*) e GPRS, através do uso de vários tipos de sensores (pressão sanguínea, eletrocardiograma, etc.). Estes estão interligados através de uma BAN (*Body Area Network*). O módulo principal da BAN é a MBU (*Mobile Base Unit*), que recolhe os dados provenientes dos sensores e os envia via UMTS ou



Figura 2.3: Programa de monitorização de pacientes (Fonte: [2]).

GPRS (*General Packet Radio Service*) para um servidor, onde os dados são processados e podem ser utilizados por pessoal médico.

As comunicações entre entidades da BAN são referidas como sendo comunicações intra-BAN e baseiam-se em redes de curto alcance, como o Bluetooth e o ZigBee. Às comunicações entre módulos da BAN e módulos externos (servidor central, por exemplo) é atribuído o nome de comunicações extra-BAN (essas comunicações têm sempre num dos extremos a MBU) e utilizam comunicações de longo alcance (UMTS e GPRS).

A MBU têm as seguintes funções:

- Servir de *gateway* entre a BAN e as redes GPRS/UMTS;
- Coordenar os vários blocos da BAN;
- Permitir o uso da aplicação para a visualização dos dados dos sensores (ver Figura 2.4b).

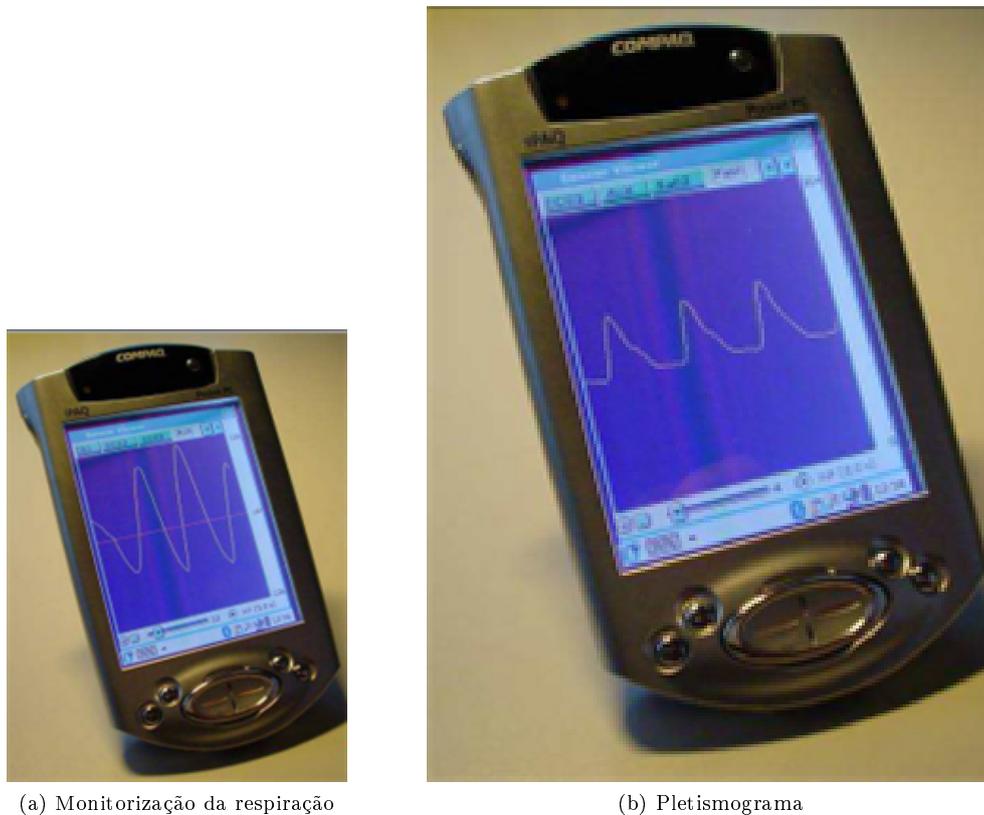


Figura 2.4: Exemplos de aplicação instaladas na MBU (Fonte: [3]).

Para a MBU, foram utilizados dois modelos de telemóvel / PDA:

iPAQ H3870 Utilizado como MBU para comunicações extra-BAN via GPRS (ver Figura 2.5a).

Nokia 6650 Utilizado como MBU para comunicações extra-BAN via UMTS (ver Figura 2.5b).



Figura 2.5: Dispositivos móveis usados como MBU (Fonte: [3]).

O consórcio Mobihealth é composto por 14 membros de 5 diferentes países da União Europeia, entre os quais:

- Empresas ligadas às tecnologias (Ericson, Compaq, Phillips);

- Empresas ligadas às telecomunicações (Telefónica Móviles España, Telia Mobile);
- Empresas ligadas à saúde (Corporació Sanitària Clínic, Twente Medical Systems IntŠl BV);
- Universidades (University of Twente, Luleå University of Technology, Universitat Pompeu Fabra).

2.1.3 Projeto Aid-N

Em [4], é apresentado um sistema de suporte ao pessoal médico em situações de emergência, como é o caso de atentados, catástrofes naturais, etc. O sistema é composto por quatro blocos principais:

- MiTags;
- Rede *mesh* de repetidores de suporte;
- Dispositivo de acesso às MiTags;
- Servidor central.

A arquitetura está representada na Figura 2.6.

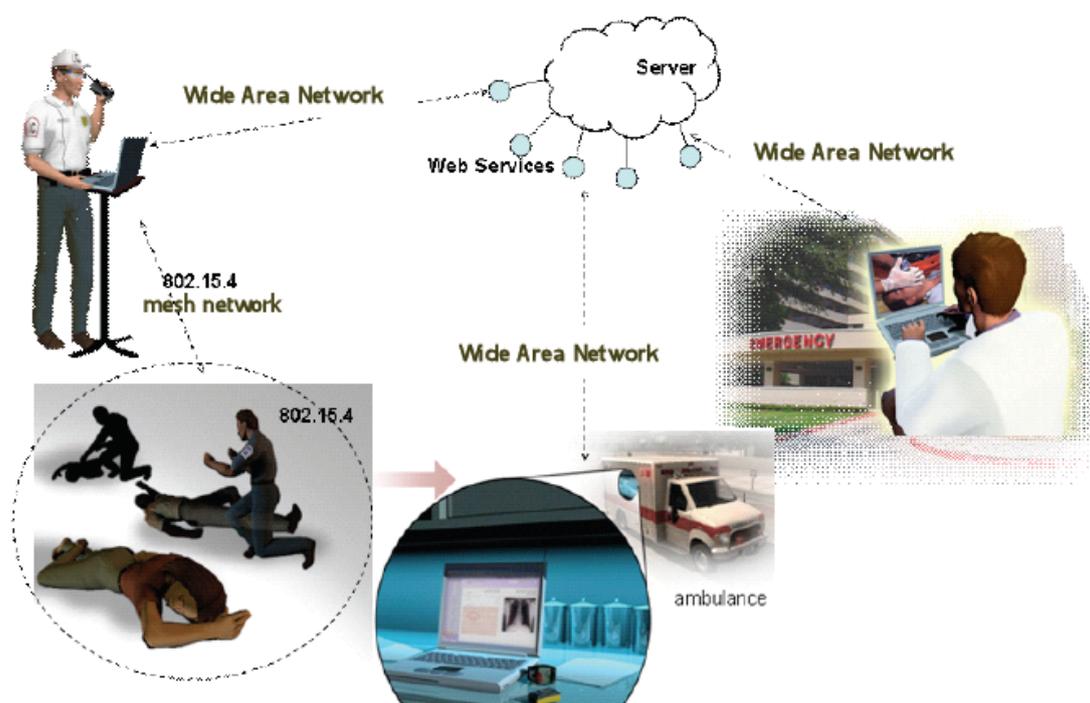


Figura 2.6: Arquitectura do sistema MiTags (Fonte: [4]).

As MiTags são plataformas de sensores sem fios modulares e extensíveis. Estes módulos suportam dois tipos de comunicação: curto alcance, de modo a ser possível criar uma *Wireless Body Area Network* (WBAN) e longo alcance, de modo a se poder ligar a rede *mesh* de suporte. A nível de sensores, a plataforma suporta vários tipos: pressão sanguínea, temperatura, oximetria, entre outros. Cada MiTag é munida de uma antena *patch* e suporte a GPS (*Global Positioning*

System). Estas MiTags são distribuídas às pessoas de modo a ser possível o servidor central fazer a triagem e orientar as equipas médicas consoante as necessidades no terreno.

A rede de suporte é composta por vários repetidores. Depois de uma tragédia ocorrer, é necessário colocar vários repetidores no local, de modo a estes repetidores criarem uma rede *mesh* que cobre o espaço necessário. A colocação dos repetidores é feita manualmente, com a ajuda de um PDA com um programa que avisa em que zonas é mais sensato colocar os repetidores (zonas com pouca cobertura, etc.).

Os dispositivos de acesso às Mitags podem ser PDAs ou portáteis e são utilizados pelas equipas médicas para perceber o estado dos pacientes. É assim possível existir troca de dados entre equipas médicas e entre estas e o servidor central.

O servidor central é o responsável pela análise dos dados das MiTags e pelo envio de atualizações para os PDAs. Outra das funções do servidor é atualizar portais web com informação relevante (que pode ser acedida, após autenticação, por parte de equipas médicas e outros intervenientes).

Para testes, foi simulado um cenário de emergência, e criaram-se 2 equipas distintas, cada uma com 7 pessoas:

- Um capitão;
- Um responsável pela triagem;
- Um médico;
- Um responsável pelo transporte;
- Três assistentes médicos.

Uma das equipas utilizou métodos tradicionais de resposta a situações de emergência. A outra utilizou o sistema de MiTags (tendo sido necessária uma explicação de 10 minutos sobre o funcionamento do sistema). As conclusões foram as seguintes:

- A equipa das MiTags foi capaz, em média, de aceder aos dados vitais das vítimas cerca de 2 vezes mais frequentemente;
- O facto de existir um servidor central, com informações disponíveis nos portais web, permitiu existir uma melhor troca de informação entre as equipas de campo e os hospitais e outros estabelecimentos de cuidados médicos;
- Os sensores das MiTags forneceram informação mais detalhada sobre o estado das vítimas, permitindo diagnósticos e tratamentos mais corretos;
- As MiTags diminuíram as comunicações via rádio entre intervenientes, muito por causa da aplicação nos dispositivos de acesso às MiTags oferecer toda a informação necessária para as equipas médicas.

2.1.4 Empresa eHit

A empresa eHit [5] é uma empresa finlandesa, fundada em 1995, que oferece serviços e tecnologias relacionados com assistência médica à distância.

Uma das tecnologias desenvolvida pela empresa é o Health Gateway, uma plataforma composta por um módulo móvel (telemóvel, *smartphone* ou PDA) e um servidor (ver Figura 2.7). O módulo móvel tem como objetivo ler os dados provenientes de sensores ligados a um paciente e enviá-los para o servidor. O servidor tem como objetivo receber os dados dos módulos móveis e encaminhá-los para o serviço onde estes dados irão ser processados. A componente de segurança

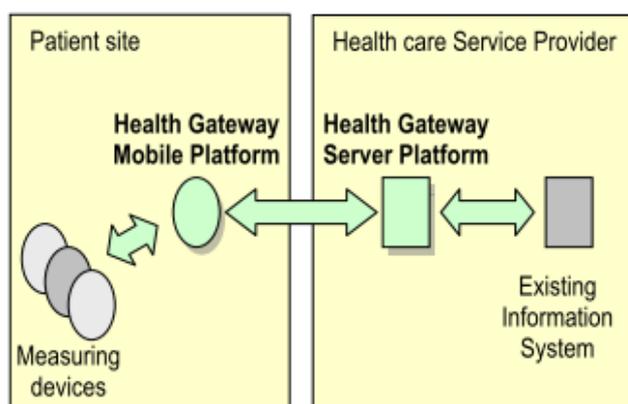


Figura 2.7: Arquitectura da plataforma eHit Health Gateway (Fonte: [5]).

nas comunicações é muito importante para a plataforma. Se os mecanismos de segurança forem fracos, a privacidade das pessoas pode estar posta em causa (para além de se poder violar leis relativamente a dados médicos). Se forem muito elaborados, torna o sistema difícil de implementar. Os autores decidiram então que as leis locais e as políticas da empresa é que deveriam definir estes mecanismos.

A ligação entre os sensores e o módulo móvel é feita via Bluetooth (exceto casos especiais em que não seja possível). Para implementar ligações seguras via este tipo de comunicação, é necessário configurar os dois dispositivos (bloco sensor e módulo móvel), tendo sido escolhido o modo de segurança 3 do Bluetooth, um modo que inicia a segurança antes de um canal de comunicação ser estabelecido, sendo utilizado em aplicações críticas, onde a segurança é um requisito fundamental. A fase mais crítica numa conexão Bluetooth é aquela onde é criada uma chave de ligação, pois este passo utiliza canais não seguros. O Bluetooth SIG (*Special Interest Group*) recomenda que seja utilizada uma área privada na altura da ligação, uma área onde se saiba que não existem dispositivos estranhos nas proximidades. Outro ponto abordado é a utilização de um PIN (*Personal Identification Number*), que é trocado entre os dispositivos na altura da conexão. Se for utilizado um PIN do tipo '1234', por exemplo, é muito fácil penetrar no sistema, sendo recomendado utilizar PIN mais complexos (a especificação do Bluetooth permite PIN de até 16 bytes).

Na ligação entre o módulo móvel e o servidor, podem ser utilizados GSM, GPRS, CDMA (*Code Division Multiple Access*) ou 3G, havendo vários mecanismos de ligação disponíveis (HTTPS (*HyperText Transfer Protocol Secure*), PKIs (*Public Key Infrastructure*), etc.), sendo possível configurar o sistema de modo a utilizar o mais apropriado.

A empresa eHit fornece mais dois produtos interessantes:

Patient Care Device Gateway Um módulo que permite que o Health Gateway descrito anteriormente seja utilizável também para a monitorização de pacientes dentro do hospital, não

sendo obrigatório o uso de um módulo móvel (PDA, etc.). É assim possível ligar dispositivos de monitorização ao Health Gateway sem ser através de ligações wireless, utilizando por exemplo uma LAN (*Local Area Network*) instalada no hospital. É porém necessário, nalguns casos, colocar um adaptador entre o dispositivo e a LAN, pois alguns dispositivos não possuem suporte direto a ligação via LAN, ou estão ligados a uma rede que utiliza um protocolo proprietário (é mais fácil criar mais um adaptador cada vez que é utilizado um novo protocolo proprietário do que alterar o próprio Care Device Gateway).

eLink Um módulo que permite que sensores que não possuem interface de ligação Bluetooth possam estar ligados a um módulo móvel, que está configurado para apenas tratar dos dados que são recebidos via Bluetooth. O eLink é então um adaptador que possui 3 tipos de interfaces físicas: infravermelhos, série e Bluetooth. A ligação Bluetooth é utilizada para ligar ao módulo móvel. As outras duas são utilizados para ligar a sensores. Uma ilustração destas ligações é apresentada na Figura 2.8.

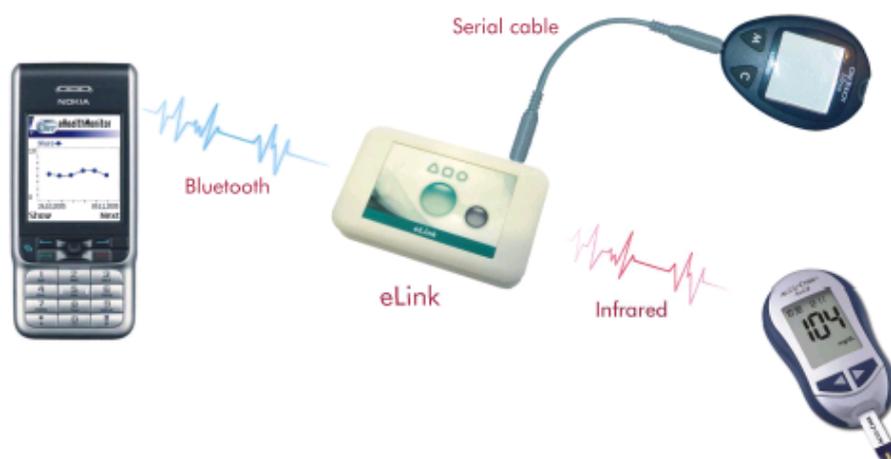


Figura 2.8: Ligações entre dispositivos via eLink (Fonte: [5]).

2.2 Área da monitorização ambiental

2.2.1 Projeto NoiseTube

O objetivo do projeto NoiseTube [18] é permitir que cidadãos possam contribuir na medição da poluição sonora. Em condições normais, as medições são feitas por pessoas especializadas, em sítios específicos, levando aos seguintes problemas:

- O facto de as medições serem efetuadas em sítios específicos, às vezes muito distanciados, leva a que sejam elaborados ‘mapas de poluição’ às vezes errados (um local situado entre dois outros locais muito poluído a nível sonoro não implica que este local também o seja);
- O facto de ser necessário pessoas e equipamentos especializados não permite que todas as cidades possam fazer estudos relativos a este tipo de poluição;

- As medições efetuadas pelos especialistas deveriam incidir sobre as zonas onde as pessoas estão mais sujeitas ao ruído, o que muitas vezes não acontece (devido ao uso de estratégias do tipo traçar uma grelha sobre o espaço que se quer estudar e medir em pontos fixos, sendo assim excluída a componente social deste tipo de estudos);
- As medições efetuadas pelos especialistas apenas dizem respeito a locais exteriores, não existindo forma de se medir a poluição sonora em ambientes interiores (empresas, casas, etc.).

O projeto NoiseTube permite que sejam os próprios cidadãos a fazer medições de poluição sonora, sendo depois esta informação enviada para um servidor central onde é processada. Para isso, os cidadãos apenas precisam de ter um telemóvel com GPS. Neste telemóvel é instalada uma aplicação (ver Figura 2.9), disponível no site do projeto, que permite, através do microfone do telemóvel, calcular o nível de poluição sonora (através de um algoritmo de processamento de sinal implementado na aplicação).

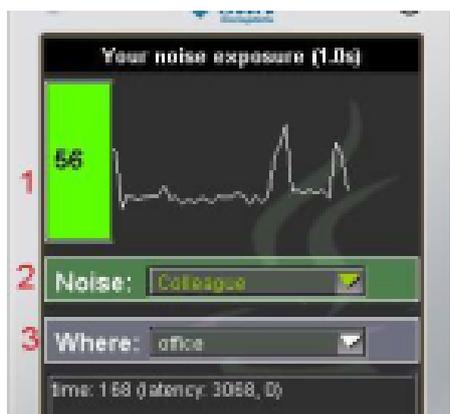


Figura 2.9: *Screenshot* da aplicação do telemóvel(Fonte: [6]).

A interface da aplicação tem vários objetivos:

- Permitir ao cidadão visualizar a medição efetuada (linha 1 na Figura 2.9);
- Permitir ao cidadão inserir informações extras relativas à medição (linhas 2 e 3 na Figura 2.9).

Depois de efetuada a medição, esta é enviada para o servidor central, onde é processada e utilizada para atualizar um mapa de poluição, acessível via Google Earth (ver Figura 2.10). É ainda disponibilizada uma API (*Application Programming Interface*) web para publicar e aceder aos dados da plataforma.

2.2.2 Projeto N-Smarts

O projeto N-Smarts [19] tem como objetivo implementar um sistema de monitorização da poluição do ar. Tal como o projeto NoiseTube, a monitorização é feita pelos cidadãos. Porém, a arquitetura deste projeto é diferente, uma vez que não existe qualquer sensor integrado num dispositivo móvel (PDA, *smartphone*, etc.) capaz de medir a poluição do ar (papel desempenhado pelo microfone



Figura 2.10: Mapas de poluição(Fonte: [6]).

no projeto NoiseTube). Portanto, foram desenvolvidos vários módulos de hardware para resolver este problema:

- Um módulo para uso em carros ou uso fixo (ver Figura 2.11a);
- Um módulo para uso pessoal (ver Figura 2.11b).



(a) Automotive Platform



(b) Personal Platform

Figura 2.11: Módulos desenvolvidos no âmbito do projeto N-Smarts (Fonte: [7]).

Estes módulos possuem interface GPS para *upload* da localização das leituras, assim como sensores para a medição dos valores de monóxido de carbono, dióxido de azoto, ozono, entre outros. Foi ainda desenvolvida uma plataforma para uso em conjunto com um telemóvel ou outro dispositivo móvel (ver Figura 2.12). Para esta integração, utiliza-se a bateria do telemóvel como ponto de ligação, estando acoplado a ela uma PCB (*Printed Circuit Board*) onde podem estar ligados os sensores referidos anteriormente, assim como um sensor de temperatura, um acelerómetro e um módulo de comunicação Bluetooth para comunicar com o telemóvel. A ligação ao telemóvel via Bluetooth foi escolhida por ser mais genérica a nível de software.

Para testes ao sistema, foi montada uma sala de testes onde é possível determinar as concentrações dos gases e a percentagem de humidade da sala (ver Figura 2.13).

Tal como no projeto NoiseTube, os resultados estão disponíveis via Google Earth (ver Figura 2.14).

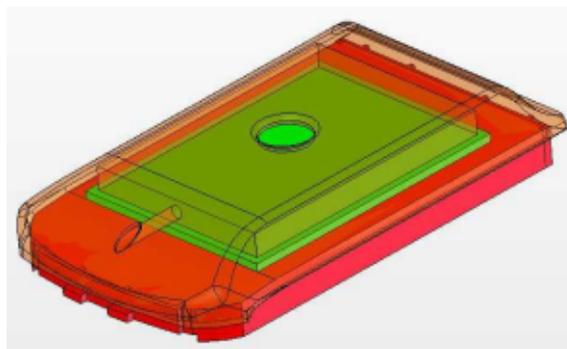


Figura 2.12: Módulo integrado com um telemóvel já existente (LG VX9800)(Fonte: [7]).

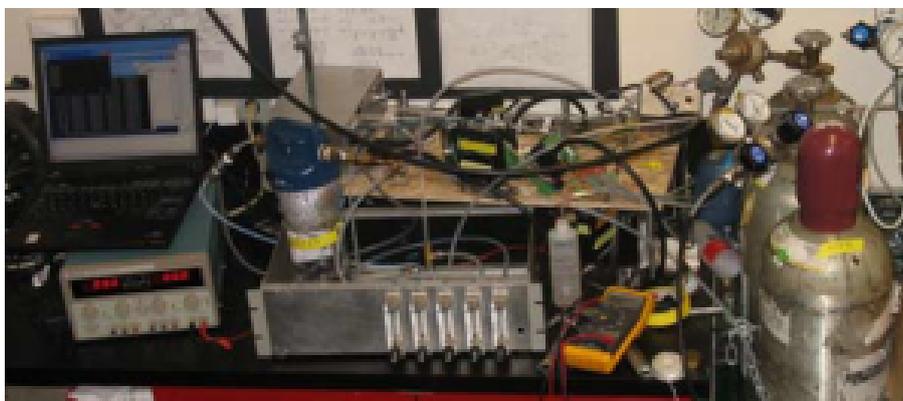


Figura 2.13: Centro de controlo da sala de teste(Fonte: [7]).

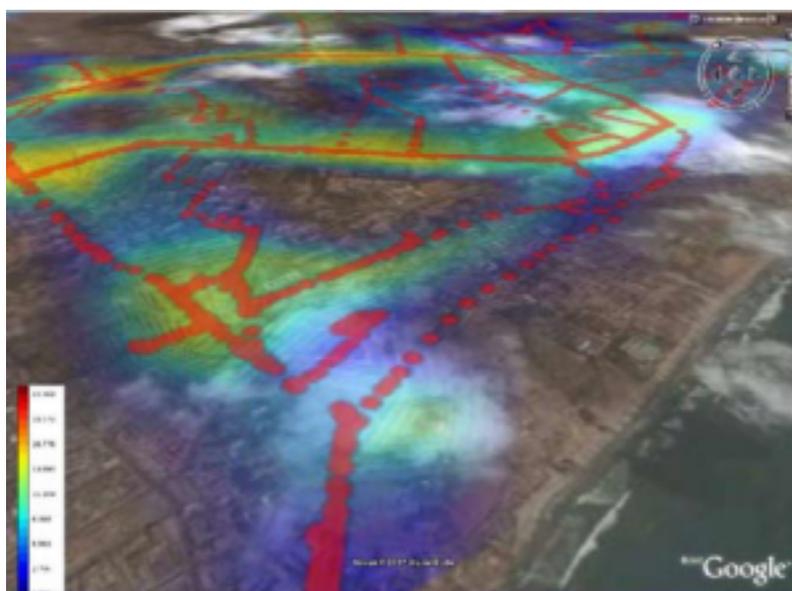


Figura 2.14: Visualização dos resultados via Google Earth(Fonte: [7]).

2.2.3 Projeto SIREN

Em [8], é apresentado um sistema de suporte para bombeiros em situações de incêndios. O sistema baseia-se numa rede de sensores dispersa pelo local do incêndio e pelo uso de PDAs como recetores dos dados recolhidos pelos sensores. A arquitetura do sistema baseia-se em um modelo de programação chamado Information Spaces (cada PDA é um Information Space). Cada Information Space representa um módulo de armazenamento de dados acessível por rede através das seguintes funções:

- Enviar informação;
- Receber informação;
- Inscrever-se num Information Space;
- Anular a inscrição de um Information Space.

Estas funções permitem a um PDA X enviar, por exemplo, as suas leituras de sensores para um PDA Y , ou mesmo estar sempre a receber a informação do PDA Z . Para ter um melhor ponto de vista sobre as necessidades dos bombeiros, os autores fizeram várias entrevistas a vários bombeiros, tendo obtido as seguintes conclusões:

- Em casos de incêndio, é muito difícil para os bombeiros terem uma visão global da situação, muito por causa da natureza dinâmica do fogo (uma informação válida pode já não o ser passados 5 minutos);
- As comunicações entre bombeiros num incêndio são muito complicadas. Por um lado, existe o ruído sonoro na área do incêndio (sirenes no exterior, vidros a partir, etc.). Por outro, o facto de as comunicações entre bombeiros ser feita em *broadcast* origina uma elevada congestão na transmissão/recepção de dados;
- Existem zonas onde as comunicações são impossíveis, por não existir cobertura (caves, entre outros);
- Qualquer dispositivo a ser utilizado pelos bombeiros tem de ser leve e resistente a temperaturas elevadas.

Após estas entrevistas, foram tomadas as seguintes decisões para a implementação do sistema Siren:

- Utilizar uma arquitectura *peer-to-peer* para aumentar a fiabilidade do sistema (cada bombeiro carrega um dispositivo que se pode ligar a outros, formando um grupo, existindo assim uma maior fidelidade da informação trocada).
- Utilizar comunicações *store-and-forward*, de modo a ser possível o próprio PDA comunicar com outros sem ser necessário a acção do bombeiro (envio de ordens de evacuação, aviso de perigo, etc.);
- Utilizar mecanismos de redundância, que pode ser de comunicações (por ser *multihop*, a arquitetura do Siren pode implicar a receção da mesma mensagem várias vezes, maximizando

as probabilidades da mensagem ser entregue), de armazenamento (uma informação pode estar ao mesmo tempo em vários PDAs, o que implica uma maior probabilidade de outros dispositivos a receberem), e de versões (armazenar não só a informação atual do bombeiro, mas também a informação anterior, permite ter uma visão mais abrangente da situação).

O Siren, depois de implementado (ver interface do PDA na Figura 2.15), foi testado com a ajuda de bombeiros, tendo obtidos resultados muito satisfatórios. Em [20], é apresentado um

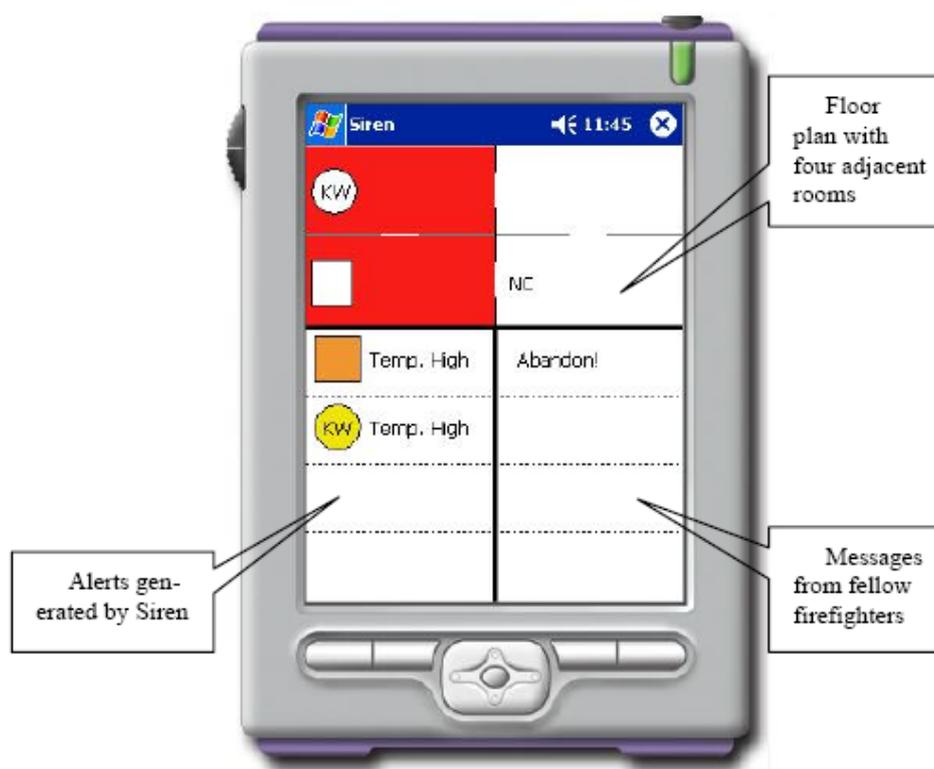


Figura 2.15: GUI (Graphical User Interface) disponibilizada pelo Siren (Fonte: [8]).

sistema para ajudar a evacuação de edifícios em caso de incêndio e outras catástrofes. Tal como o sistema SIREN, este baseia-se na leitura de sensores para se conhecer a melhor forma de evacuação.

2.3 Outras áreas de aplicação

2.3.1 Multimedia Guidebook

Em [9], é apresentado um sistema de visita guiada a museus e outros locais, baseado num servidor central, numa rede de postos de informação (IPS - *Information Point Station*) e em PDAs munidos de ponteiros (ver Figura 2.16) para a interação com a rede de IPS. Os IPS podem ter associados a eles módulos de comunicação com o ponteiro do PDA (via infravermelho, para ser possível o utilizador apenas receber uma dada informação se este apontar para o IPS), ou podem apenas ter módulos de comunicação direta com o PDA (via Bluetooth, de modo a que uma dada informação

seja sempre recebida pelo utilizador, não sendo necessário este apontar para o IPS). Também podem ser instalados módulos de ligação infravermelhos (para comunicação com o ponteiro) sem ligação a um IPS.



Figura 2.16: Dispositivo móvel com ponteiro associado (Fonte: [9]).

Existem 3 formas de interação entre o PDA e a rede de IPS:

Ligação direta entre o PDA e o IPS Ao se aproximar de um ISP, o PDA recebe, via Bluetooth, informação relativa aos pontos de interesse na periferia do ISP, sendo possível receber um menu de modo a poder escolher que informação deseja receber (ver Figura 2.17). Neste cenário, o PDA deve suportar Bluetooth e não precisa de ter o ponteiro.

Ligação entre o PDA e o IPS via ponteiro Nesta forma, é necessário um ponteiro e um IPS com um módulo de comunicação por infravermelhos. O utilizador, ao apontar para o sensor laser associado ao IPS, está a enviar-lhe o seu endereço Bluetooth. Essa informação é então transferida para o IPS, que a usa para enviar os dados associados àquele lugar via Bluetooth. Neste cenário, o PDA deve suportar Bluetooth e deve ter o ponteiro.

Ligação indireta entre o PDA e um IPS, via sensor isolado . Neste forma, o utilizador aponta para um bloco sensor ao qual não está associado nenhum IPS. O bloco sensor fornece então, para o PDA, o identificador do local ou objeto a ele associado, via infravermelhos. O PDA, com essa informação, liga-se ao IPS mais próximo via Bluetooth e submete a referência, recebendo a informação desejada. Neste cenário, o PDA tem de suportar Bluetooth e tem de ter o ponteiro. De salientar que este é o único cenário que utiliza uma comunicação por infravermelhos bidirecional.

A Figura 2.18 representa graficamente estes tipos de ligação. Em [21], é apresentado um sistema parecido, que permite aos visitantes do museu partilhar notas relativamente aos objetos expostos, de modo a ser possível melhorar a forma como o museu apresenta os objetos, de acordo com os gostos dos visitantes (se uma peça for vista por muita gente e estiver num local isolado, convém passar esta peça para um local mais visível).

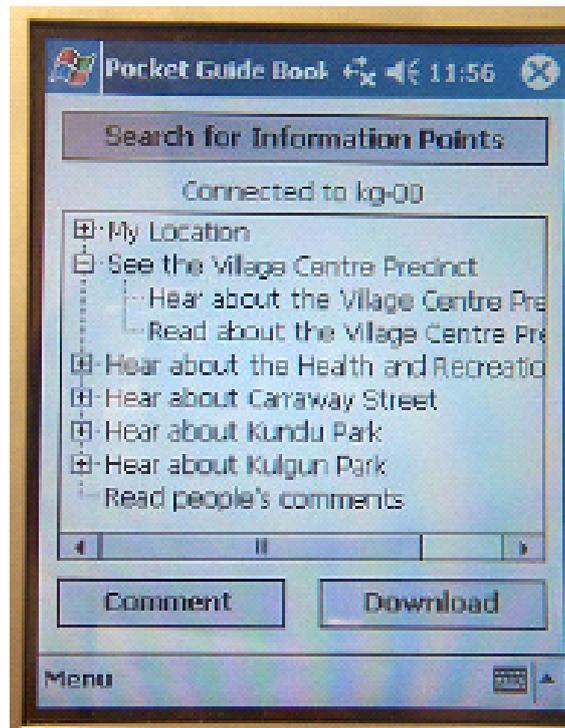


Figura 2.17: Menu de interação com o utilizador (Fonte: [9]).

2.3.2 Projeto Bikenet

O projeto BikeNet [22] é um projeto que permite a ciclistas compartilhar informação entre eles, desde informação sobre desempenhos desportivos (distâncias percorridas, velocidades, etc.), até informações sobre determinados espaços geográficos (poluição, etc.). Os principais componentes da arquitetura do projeto são uma rede de sensores, um servidor de dados e um módulo de ligação entre a rede de sensores e o servidor (designado de SAP - *Sensor Access Point*). De um modo geral, as principais funcionalidades do projeto são [10]:

- É possível armazenar os dados e só depois enviá-los para o servidor. Caso se utilize um PDA/smartphone em conjunto com o sistema, é possível enviar os dados para o servidor em tempo real;
- É possível monitorizar vários aspetos do desempenho do ciclista (velocidade atual, velocidade média, distância percorrida, calorias queimadas), assim como aspetos relativos à saúde (frequência cardíaca, GSR (*Galvanic Skin Response*), indicadora de excitação emocional / nível de stress) e ao próprio terreno (inclinação);
- Os ciclistas podem depois da atividade desportiva fazer uma análise a todos os dados recolhidos;
- Os ciclistas podem configurar a forma como todos os sensores atuam, assim como a forma como os dados são apresentados ao ciclista durante a atividade desportiva (via visor do dispositivo móvel, fixado no guiador);

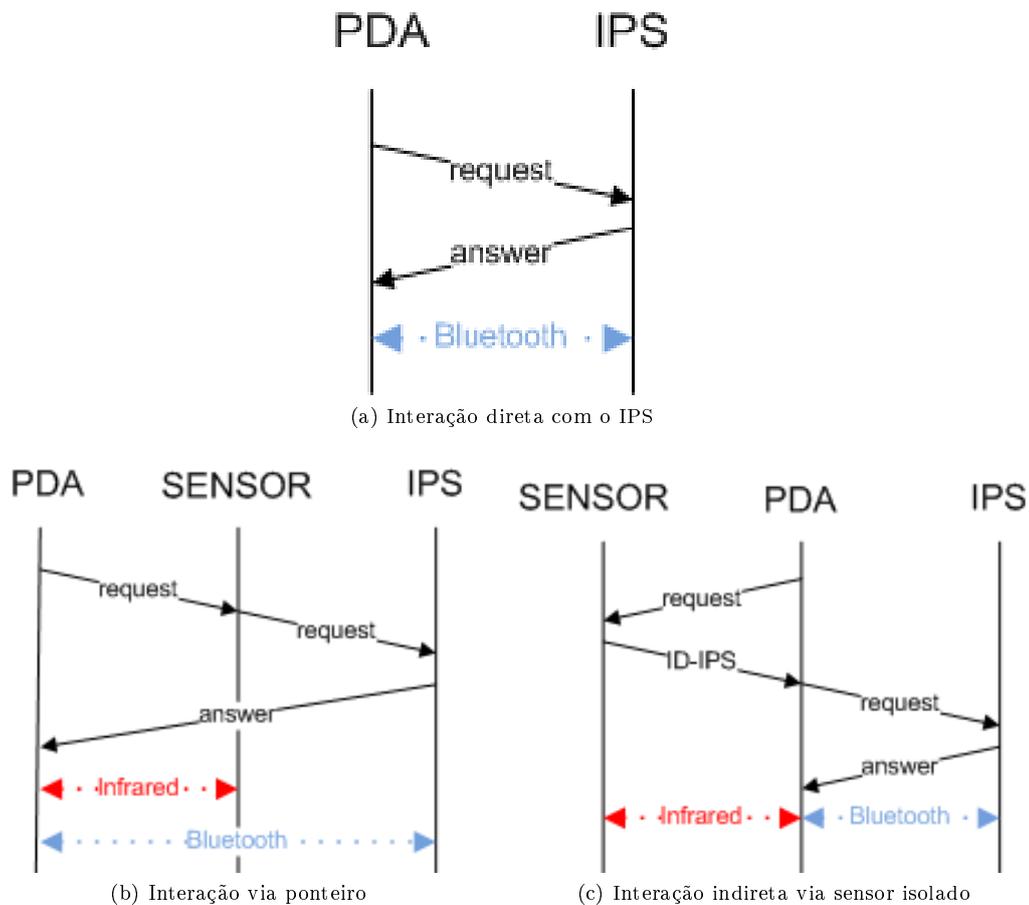


Figura 2.18: Tipos de interação possíveis.

- Existe um portal Web onde é possível elaborar qualquer tipo de pesquisa, assim como partilhar informação (*screenshot* na Figura 2.19).



Figura 2.19: Portal Web disponibilizado aos utilizadores do projeto BikeNet (Fonte: [10]).

Relativamente ao software, existem 3 grandes módulos: sensor, SAP e servidor (ver Figura 2.20). Dos blocos apresentados na figura, os mais importantes são:

BAN Management and Control Responsável pelo controlo da rede de sensores, incluindo a sincronização dos nós, a gestão da recolha de dados, a gestão de eventos e o controlo da apresentação de dados através do PDA.

TinyOS Networking Stack Responsável pela ligação entre os sensores e o SAP.

Data Exchange Services Responsável pela gestão dos dados recolhidos pelos sensores, incluindo o armazenamento das leituras em situações *offline* e o *upload* de dados para o servidor.

Back End Communications Networking Stack Responsável pela ligação entre o SAP e o servidor.

Sensor Data Repository and Visualization Responsáveis pelo armazenamento dos dados e pela apresentação dos dados ao utilizador via portal Web.

Query Submission Portal Responsável pela interface de acesso aos dados por parte do utilizador.

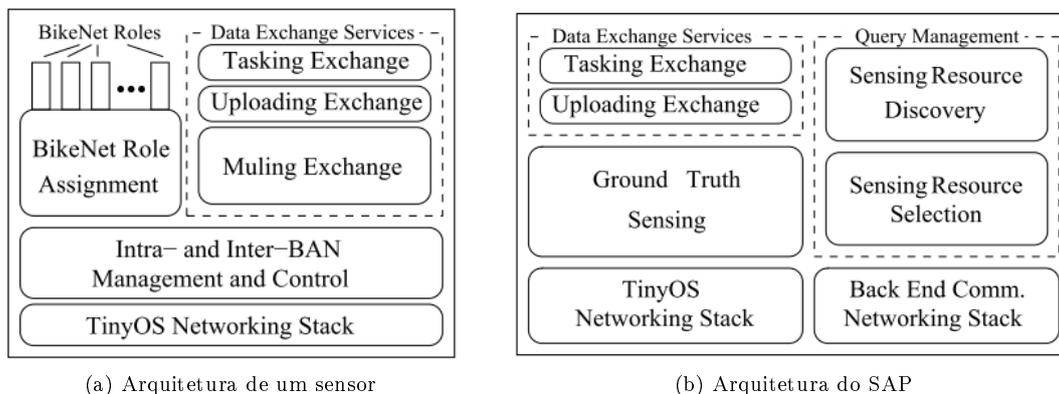


Figura 2.20: Arquitetura de software dos vários componentes do projeto BikeNet.

2.3.3 Projeto Motetrack

O projeto MoteTrack [23] foi desenvolvido pelos investigadores do projeto CodeBlue e tem como objetivo implementar um sistema de localização baseado em sensores sem fios, em vez dos sistemas já implementados baseados em WLAN (*Wireless Local Area Network*) (RADAR), ultrassons (Cricket) ou mesmo infravermelhos (Active Badge). O principal objetivo do projeto MoteTrack é melhorar algumas fragilidades dos sistemas já implementados:

Processamento Centralizado A maioria dos sistemas utiliza um servidor central para o cálculo da localização (em caso de falha do servidor, todo o sistema é afetado).

Alta dependência da infraestrutura Muitos dos sistemas de localização baseados em WLAN são muito afetados pela perda de um ou mais pontos de acesso Wi-Fi (em caso de avaria de um ou mais pontos de acesso, a localização calculada pode não ser a correta).

O sistema implementado utiliza uma arquitetura semelhante aos sistema de localização via Wi-Fi: são colocados *motest* (que têm um papel parecido ao dos pontos de acesso Wi-Fi na área onde é necessário existir o sistema (para cenário de teste, foi utilizado um dos edifícios da universidade de Harvard, representado na Figura 2.21). Estes nós são configurados de modo a enviar via *broadcast* mensagens em intervalos de tempo regulares, em vários canais de frequência e com diferentes potências de emissão. Com esta configuração, os nós conseguem elaborar uma base de dados com a informação relativa ao posicionamento dos outros nós (a base de dados está distribuída pelos vários nós, pois cada um apenas conhece os nós cujas mensagens consegue receber). Quando um dispositivo móvel precisa de saber a sua localização, fica à escuta das mensagens *broadcast* durante um determinado intervalo de tempo, criando uma assinatura, que associa os identificadores dos nós das mensagens recebidas, assim como a potência de sinal (RSSI - *Received Signal Strength Indicator*). Esta assinatura é então utilizada para se descobrir a localização do dispositivo, via algoritmos descritos em [11].



Figura 2.21: Cenário de teste do projeto MoteTrack (Fonte: [11]).

CAPÍTULO 3

DESENVOLVIMENTO

Everything should be made as
simple as possible, but not
simpler.

Albert Einstein

Conteúdo

3.1	Projeto QWisNet	26
3.2	Gateway inter-redes	27
3.2.1	Objetivo	27
3.2.2	Arquitetura	27
3.2.3	Conceção	28
3.3	Deteção de quedas	34
3.3.1	Objetivo	34
3.3.2	Trabalho relacionado	34
3.3.3	Arquitetura	35
3.3.4	Conceção	35
3.4	Hardware de interface	39
3.4.1	Objectivo	39
3.4.2	Análise do problema e montagem electrónica	39

Este capítulo começa com uma breve descrição do projeto QWisNet, no âmbito do qual foi desenvolvido este trabalho de dissertação. O resto do capítulo é dedicado à descrição dos três módulos desenvolvidos nesta dissertação.

3.1 Projeto QWisNet

O projeto QWisNet (*Quality of Service in Wireless Sensor and Actuator Networks*) é um projeto de investigação financiado pela FCT (Fundação para a Ciência e Tecnologia) que decorreu de Outubro de 2007 a Março de 2011. As tarefas deste projeto estão identificadas abaixo:

1. Estudo e especificação dos detalhes do projeto.
2. Desenvolvimento de modelos de simulação detalhados de protocolos de redes de sensores sem fios, através dos quais as soluções propostas foram analisadas e otimizadas.
3. Desenvolvimento de um protótipo do sistema de aquisição de dados sensoriais escolhido para possibilitar a avaliação do seu comportamento em condições reais.
4. Desenvolvimento de uma interface entre dispositivos móveis e redes de sensores sem fios.
5. Execução de testes experimentais para validação e otimização das soluções propostas.

No âmbito da tarefa 1, identificou-se um domínio de aplicação das redes de sensores sem fios no qual a provisão de qualidade de serviço é relevante: a monitorização de sinais sensoriais no corpo humano. O desenvolvimento do sistema de deteção de quedas efetuado no âmbito desta dissertação enquadra-se neste domínio de aplicação.

No âmbito da tarefa 2, foi efetuado um estudo dos protocolos de comunicação para redes de sensores sem fios, seguido da implementação de modelos de simulação [24] para o protocolo LEACH (*Low Energy Adaptive Clustering Hierarchy*), para o mecanismo de encaminhamento MTE (*Minimum Transmission Energy*) e para a estratégia de comunicação direta. A seguir, foram desenvolvidos modelos de simulação do protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) da norma IEEE 802.15.4 nas topologias em estrela e em árvore, bem como modelos de simulação do protocolo LPRT (*Low Power Real Time*) [25] e dos seus diversos aprimoramentos concebidos no âmbito do projeto, que deram origem ao protocolo eLPRT (*Enhanced Low Power Real Time*) [26]. Ainda no âmbito desta tarefa, foram implementados os modelos de erros no canal utilizados nos cenários de simulação e um modelo para análise do fenómeno da estação oculta.

No âmbito da tarefa 3, a nível de software, foram implementados dois protótipos de um sistema de captura de movimentos sem fios, baseado em acelerómetros e magnetómetros de 3 eixos, primeiro no circuito integrado CC2430 e depois no CC2530. Estes protótipos têm em consideração não só as funcionalidades do protocolo de comunicações, mas também a execução de tarefas como a aquisição de dados, calibração dos sensores e interface com o PC [27]. A nível de hardware, foram também implementados dois protótipos do módulo sensorial do sistema de captura de movimentos sem fios, ambos com antena integrada. O primeiro protótipo integra um CC2430 e sensores para captura dos movimentos. Já o segundo protótipo utiliza um CC2530, e possui conectores para ligação a uma placa com sensores. Foi ainda desenvolvida uma aplicação computacional para PC que efetua a interface com o utilizador, permitindo a visualização dos movimentos em tempo-real em 3D e o armazenamento dos dados [28].

A tarefa 4 é aquela em que se enquadra a maior parte do trabalho desenvolvido nesta dissertação.

No âmbito da tarefa 5, foram desenvolvidas ferramentas de software para medição do consumo dos módulos sensoriais monitorização da intensidade de sinal no canal e medição da taxa de perda de pacotes do protocolo de comunicações. Estas ferramentas foram utilizadas para otimização dos mecanismos propostos e a validação de diversas funcionalidades do protocolo implementado.

3.2 Gateway inter-redes

3.2.1 Objetivo

Os projetos descritos no capítulo 2 foram feitos partindo do zero, ou seja, tiveram de ser implementados módulos de hardware e de software especialmente para os projetos. Os programas criados no âmbito destes projetos têm como principal desvantagem o facto de existir uma elevada dependência da componente de hardware. Por exemplo, se numa primeira fase se utilizar um módulo com apenas um sensor e existir uma aplicação que apenas precisa da informação deste sensor, se a componente de hardware sofrer uma atualização e cada módulo passar a possuir 2 sensores (ou continuar a possuir um único, mas com um formato de trama diferente, por exemplo), a probabilidade de a aplicação precisar de uma atualização é muito elevada.

O módulo de *software* desenvolvido tem como objetivo servir de intermediário entre a componente de hardware e a lógica aplicacional, ou seja, conseguir adaptar as necessidades das aplicações aos mais variados tipos de sensores. Se por exemplo estiverem 2 tipos de sensores distintos numa área e uma aplicação quiser apenas os valores lidos por um tipo, basta ter uma regra no *gateway* a indicar isso para essa aplicação passar a receber apenas estes dados.

Em várias situações (sobretudo em casos de monitorização remota de pacientes), é necessário estender o alcance da rede de sensores para zonas inalcançáveis através dos protocolos usados em redes de sensores sem fios. Em situações de monitorização remota, pode ser necessário armazenar os dados recolhidos pelos sensores num servidor central, ou avisar determinada pessoa em casos críticos. O *gateway*, para além de servir de intermediário entre sensores e aplicações locais (a correr no dispositivo), serve também de intermediário entre sensores e destinatários remotos, utilizando os vários tipos de comunicações oferecidos pelo dispositivo. Por fim, existem situações em que as comunicações por parte do PDA não são possíveis, por ausência de cobertura (GSM, etc.), inexistência de rede de suporte (Wi-Fi), ou mesmo devido a situações em que os aspetos económicos tornam indesejado o uso de determinado tipo de comunicações (por exemplo, em situações de envio de dados para um servidor remoto é melhor esperar que o dispositivo móvel tenha acesso a uma rede Wi-Fi para enviar essa informação). Ao configurar o *gateway*, podem ser criadas regras que restringem o uso de determinadas redes em determinadas situações. Nesse caso, as leituras dos sensores são armazenadas localmente até que o dispositivo tenha acesso a uma infraestrutura de comunicação mais adaptada às necessidades do utilizador.

3.2.2 Arquitetura

De um ponto de vista global, a arquitetura do *gateway* está representada na figura 3.1.

A aplicação base é o *gateway*, que tem acesso a dois componentes:

- O gestor de interfaces;

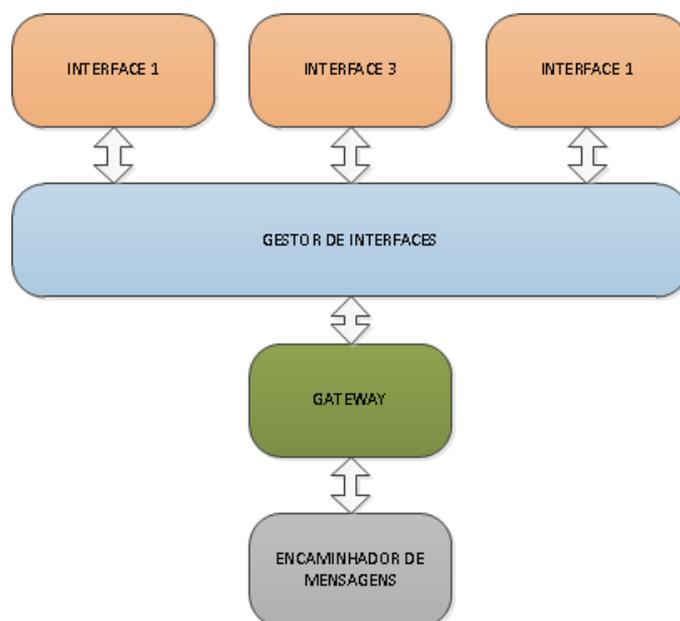


Figura 3.1: Estrutura lógica do *gateway*.

- O encaminhador de mensagens.

O gestor de interface tem como funções:

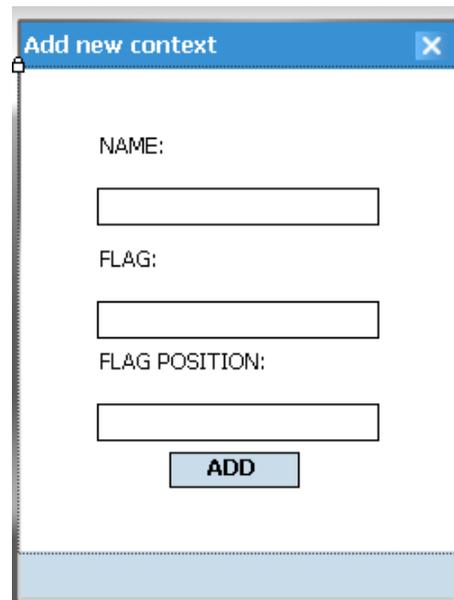
- Adicionar / remover interfaces de dados a utilizar;
- Iniciar / parar os processos de escuta e envio nessas interfaces;
- Avisar o módulo *gateway* sempre que uma das interfaces recebe alguma mensagem.

O encaminhador de mensagens tem como principal função receber as mensagens provenientes das interfaces de comunicação e verificar para que destinatário(s) encaminhar a mensagem. Este módulo tem acesso a uma base de dados para a consulta, entre outros, das regras de encaminhamento. As interfaces de comunicação têm a simples função de enviar / receber dados.

3.2.3 Conceção

Neste projeto, foi utilizado um *smartphone* HTC com sistema operativo Windows Mobile, tendo sido seleccionada o C# como linguagem de programação. A maior parte das redes de sensores sem fios possui uma estrutura de mensagem definida. Para ser possível determinar que tipo de leitura foi recebida pelo *gateway* (para depois encaminhar as leituras de um dado tipo para um dado destinatário), o utilizador tem de inserir a informação relativa ao sensor (formato da mensagem e identificador). Para simplificar, decidiu-se que uma mensagem seria identificada por uma *flag X* colocada na posição *Y* da mensagem. Assim, a inserção desta informação é feita através do formulário da figura 3.2.

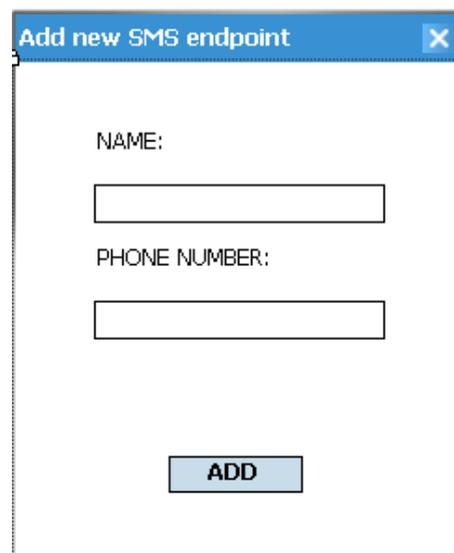
A função principal de um *gateway* é encaminhar mensagens para destinos. No *gateway* desenvolvido, estes destinatários podem ser de vários tipos (SMS (*Short Message Service*), Wi-Fi, Bluetooth, e-mail e HTTP (*HyperText Transfer Protocol*)). Cada um destes destinatários possui



The image shows a software window titled "Add new context" with a close button (X) in the top right corner. The window contains three text input fields. The first field is labeled "NAME:", the second "FLAG:", and the third "FLAG POSITION:". Below the input fields is a blue button labeled "ADD".

Figura 3.2: Formulário de inserção de formato de mensagem.

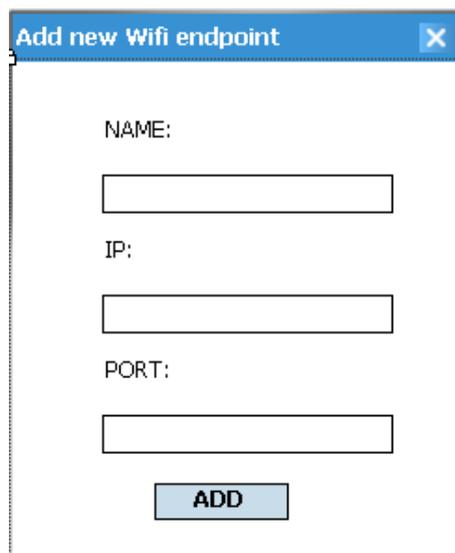
parâmetros diferentes (por exemplo, para enviar dados por SMS tem de se saber o número de telemóvel do destinatário, mas para enviar dados via Wi-Fi tem de se saber o endereço IP (Internet Protocol) e a porta em que o destinatário está à escuta). Para cada um dos destinatários, foi desenhado um formulário onde são inseridos os parâmetros de acesso ao destinatário (alguns destes formulários estão apresentados nas figuras 3.3, 3.4 e 3.5).



The image shows a software window titled "Add new SMS endpoint" with a close button (X) in the top right corner. The window contains two text input fields. The first field is labeled "NAME:" and the second "PHONE NUMBER:". Below the input fields is a blue button labeled "ADD".

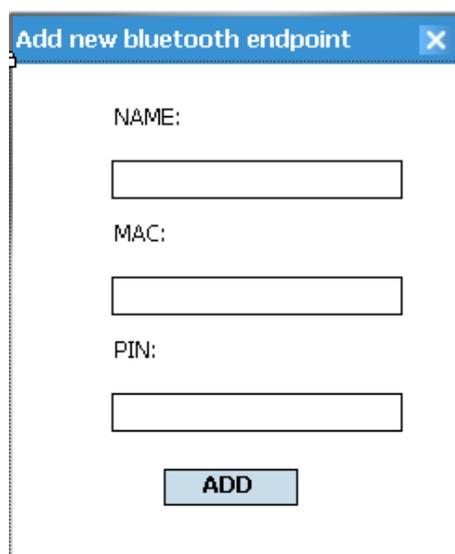
Figura 3.3: Formulário de inserção de um destinatário SMS.

Para um destinatário receber mensagens, é preciso que haja uma regra que o especifique. Estas regras podem ter parâmetros variáveis, consoante o tipo de destinatário escolhido. Por exemplo, no caso de o destinatário ser o e-mail, pode-se querer definir o título do e-mail que é enviado, assim



Formulário de inserção de um destinatário Wi-Fi. O formulário tem um cabeçalho azul com o texto "Add new Wifi endpoint" e um ícone de fechar (X). O formulário contém três campos de texto rotulados "NAME:", "IP:" e "PORT:". Abaixo dos campos, há um botão azul com o texto "ADD".

Figura 3.4: Formulário de inserção de um destinatário Wi-Fi.



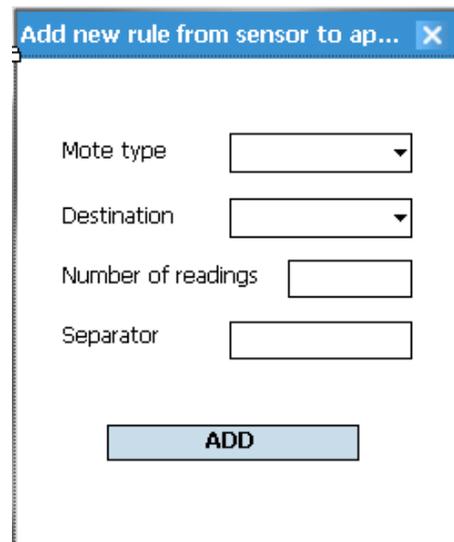
Formulário de inserção de destinatário Bluetooth. O formulário tem um cabeçalho azul com o texto "Add new bluetooth endpoint" e um ícone de fechar (X). O formulário contém três campos de texto rotulados "NAME:", "MAC:" e "PIN:". Abaixo dos campos, há um botão azul com o texto "ADD".

Figura 3.5: Formulário de inserção de destinatário Bluetooth.

como o número de leituras a enviar por e-mail. Assim, o formulário para inserir uma regra tem a estrutura apresentada na figura 3.6. Nesse formulário, pode-se selecionar o tipo de sensor/contexto a ser processado, o nome do destinatário (o nome é definido na altura da inserção dos dados do destinatário, como pode ser visto nas figuras 3.3, 3.4 e 3.5.

Para aceder a todas as funcionalidades apresentadas anteriormente, foi utilizado um simples menu, por ser a forma mais direta de apresentar os conteúdos de uma forma organizada e por ser muito utilizado nas aplicações móveis. O menu apresentado ao utilizador tem a seguinte estrutura:

- Sensores
 - Adicionar novo;



The image shows a dialog box with a blue title bar containing the text "Add new rule from sensor to ap..." and a close button (X). The dialog box has a white background and a thin border. It contains four input fields: "Mote type" and "Destination" are dropdown menus; "Number of readings" and "Separator" are text input boxes. Below these fields is a blue button with the text "ADD" in white capital letters.

Figura 3.6: Formulário de inserção de uma regra de encaminhamento.

- Ver existentes.
- Destinatários
 - Bluetooth
 - * Adicionar novo;
 - * Ver existentes.
 - E-mail
 - * Adicionar novo;
 - * Ver existentes.
 - SMS
 - * Adicionar novo;
 - * Ver existentes.
 - Wi-Fi
 - * Adicionar novo;
 - * Ver existentes.
 - HTTP
 - * Adicionar novo;
 - * Ver existentes.
- Regras
 - Para Bluetooth
 - * Adicionar nova;
 - * Ver existentes.

- Para E-Mail
 - * Adicionar nova;
 - * Ver existentes.
- Para SMS
 - * Adicionar nova;
 - * Ver existentes.
- Para Wi-Fi
 - * Adicionar nova;
 - * Ver existentes.
- Para HTTP
 - * Adicionar nova;
 - * Ver existentes.
- Interfaces
 - Wi-Fi;
 - SMS;
 - Bluetooth.

- Configuração

No menu apresentado, existe uma funcionalidade que ainda não foi analisada: a configuração. A configuração do *gateway* é muito simples e permite configurar:

- As interfaces ativas;
- Os parâmetros de ligação (por exemplo, a configuração do *proxy* para ligações à rede);
- Os parâmetros relativos ao envio de dados (nomeadamente o número de tentativas de envio, e o intervalo de tempo entre estas tentativas).

Estando a interface apresentada, a estrutura de base de dados de suporte à aplicação é o próximo componente a ser analisado.

Toda a informação recolhida através dos formulários precisa de ser armazenada. Para este efeito, optou-se por utilizar uma base de dados interna ao dispositivo (SQL Server Compact Edition). Este tipo de base de dados, em que ao nível físico existe um ficheiro, mas ao nível estrutural tem-se uma capacidade de organização de informação semelhante aos sistemas de base de dados relacionais, permite consultar os dados de uma forma muito simples, através de ADO.NET (ActiveX Data Objects). Para além disso, torna a cópia de dados do dispositivo móvel para um computador fixo, por exemplo, muito simples.

A base de dados tem as tabelas apresentadas na figure 3.7. Todas as tabelas estão relacionadas com os formulários anteriores, excepto uma: a tabela mensagens, que é utilizada pelos módulos de comunicação (que são analisados a seguir) para armazenar informação temporária.

Depois de analisadas a interface e a base de dados, falta apenas descrever os módulos de comunicação.

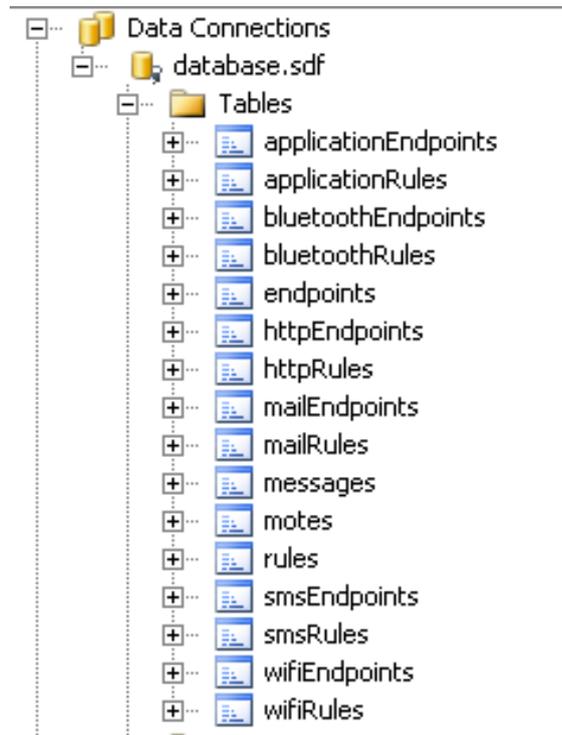


Figura 3.7: Estrutura da base de dados.

Todos os módulos de comunicação desenvolvidos implementam a interface `IComm`, que define as operações que estes devem implementar. As operações declaradas nesta interface são as seguintes:

```

void start();
void stop();
void send(string message, string[] destination);
event receiveMessage OnReceiveMessage;
  
```

As duas primeiras operações permitem ligar / desligar a receção de dados na interface. A terceira operação permite enviar dados para uma mensagem para um determinado destinatário. A última operação define que estas interfaces podem lançar o evento `receiveMessage`. Este evento é lançado sempre que uma interface recebe uma mensagem.

No caso do *gateway*, existe uma interface que está sempre à escuta (a interface USB), sendo a única a lançar o evento `onReceiveMessage`. As outras interfaces, apesar de terem as rotinas de receção de dados implementadas, não as utilizam no contexto do *gateway*, limitando-se apenas a enviar mensagens. Todos os módulos foram implementados através de funções disponibilizadas pelo .NET, à exceção do módulo de Bluetooth, que foi implementado através do uso da biblioteca `32feet.NET`, da empresa `In The Hand` [29].

3.3 Detecção de quedas

3.3.1 Objetivo

A deteção de quedas (e outros tipos de monitorização de postura) é uma das muitas possibilidades de utilização das redes de sensores sem fios. Em ambientes cobertos pela infraestrutura da rede de sensores (em casas, hospitais, etc.), esta deteção funciona bem pelas seguintes razões:

- Pode ser utilizado um computador como estação base, sendo necessário haver uma comunicação entre os sensores e a estação base;
- Caso a aplicação que monitoriza os sensores detetar uma queda, esta aplicação pode ativar um alarme audível pelas pessoas que estão próximas, de modo a ir ajudar a pessoa monitorizada;
- Caso a pessoa monitorizada esteja isolada a estação base pode enviar avisos a uma outra aplicação, através de uma rede (por cabo ou *wireless*), de modo a que mesmo as pessoas que não tenham ouvido o alarme possam ajudar a pessoa monitorizada.

Em ambiente não cobertos, tal não acontece. Por exemplo, caso se queira monitorizar uma pessoa que tenha total liberdade de movimentos, sendo possível ela sair de carro, passear, etc., a monitorização torna-se mais complexa, uma vez que os sensores não conseguem comunicar com a estação base. Em casos como este, um dispositivo capaz de aumentar o alcance dos sensores torna-se muito importante. Foi a pensar em casos como este que foi desenvolvida uma aplicação móvel para a deteção de quedas.

3.3.2 Trabalho relacionado

Existem vários trabalhos relacionados com a deteção de quedas. Estes podem ser classificados quanto à forma como a queda é detetada:

Queda detetada através do uso de acelerómetros Esta forma de deteção é a mais comum, sendo utilizada em várias implementações ([30], [31], etc.). Aos acelerómetros muitas vezes é adicionado um giroscópio, como acontece no artigo [32].

Queda detetada através do uso de câmeras Existem alguns projetos (por exemplo, [33] e [34]), que analisam o vídeo captado por uma câmara para detetar se houve ou não uma queda. Alguns destes projetos utilizam vídeos na terceira pessoa, em que se tem uma câmara a filmar uma sala, por exemplo, enquanto outros utilizam câmeras na primeira pessoa, em que essa câmara está acoplada à pessoa que se quer monitorizar.

Queda detetada através do uso de um sensor de pressão atmosférica Estes sistemas baseiam-se na variação da pressão atmosférica para detetar as quedas.

Neste trabalho, optou-se por utilizar a deteção de quedas através do uso de acelerómetros, por vários motivos:

- A deteção de quedas através do uso de sensores de pressão atmosférica, nos poucos artigos estudados, não oferece tantas certezas quanto a deteção de quedas através da acelerometria;

- A deteção de quedas através de câmeras (primeira ou terceira pessoa), para além de ser de implementação mais cara, também precisa de muito processamento de imagem, não se enquadrando na temática desta dissertação.

3.3.3 Arquitetura

Numa fase inicial, a ideia era ter como base uma arquitetura simples, que tirasse partido das capacidades de processamento e comunicação do dispositivo móvel. A arquitetura inicial é apresentada na figura 3.8.

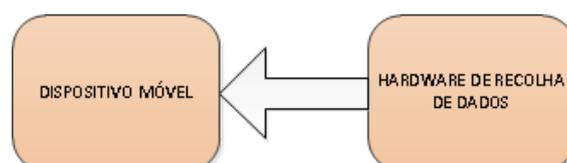


Figura 3.8: Arquitetura inicial do sistema de deteção de quedas.

Nesta arquitetura, um módulo de hardware (para a receção dos sinais dos sensores) é ligado ao telemóvel e a aplicação recebe os dados deste módulo, verificando se uma queda ocorreu ou não. Caso tenha ocorrido, é emitido um alarme sonoro durante um determinado período de tempo, sendo possível desligar este alarme pelo utilizador (por exemplo, em casos de falsos positivos). Caso o alarme não seja desligado, é enviada uma mensagem para um número de telefone configurável na aplicação. Porém, foi encontrado um entrave após a implementação do algoritmo de deteção de quedas: após criadas as funções em MATLAB, foi gerado um *assembly* para ser utilizado pelo dispositivo móvel. Porém, para se utilizar este *assembly*, é necessário que o dispositivo onde este é utilizado tenha instalado o MATLAB Compiler Runtime (MCR), e tal não é possível na plataforma utilizada. Este problema técnico levou a uma alteração da arquitetura do sistema, passando a ser a da figura 3.9.



Figura 3.9: Arquitetura final do sistema de deteção de quedas.

Nesta arquitetura, a aplicação recebe os dados da rede de sensores e envia-os para um computador onde estes dados são analisados. Caso seja detetada uma queda, é enviada uma mensagem para o dispositivo móvel, e a partir deste ponto o funcionamento passa a ser igual ao da arquitetura inicial (alarme, etc.).

3.3.4 Conceção

Numa primeira fase, foram recolhidas leituras de sensores em vários tipos de movimentos, incluindo vários tipos de queda. Para isso, foi utilizada uma placa composta por 3 sensores (um para cada eixo) e uma aplicação que recebe os dados e os grava para um ficheiro de texto. Estas leituras podem ser classificadas em dois grupos:

ADL (Activities of Daily Living) Leituras relativas a movimentações que uma pessoa faz no dia a dia, como, por exemplo, andar, levantar-se, sentar-se, entre outras.

Quedas Leituras relativas a situações de queda.

Os gráficos a seguir apresentam os valores dos sensores para algumas situações (valores brutos, sem qualquer tipo de alteração):

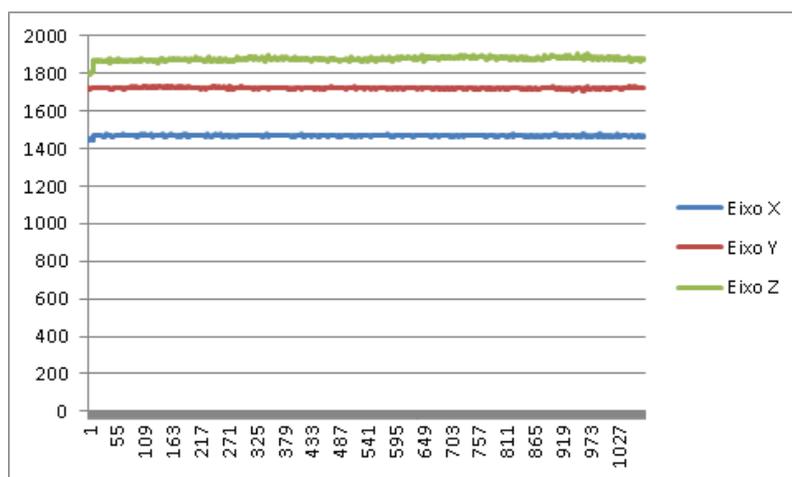


Figura 3.10: Parado de pé.

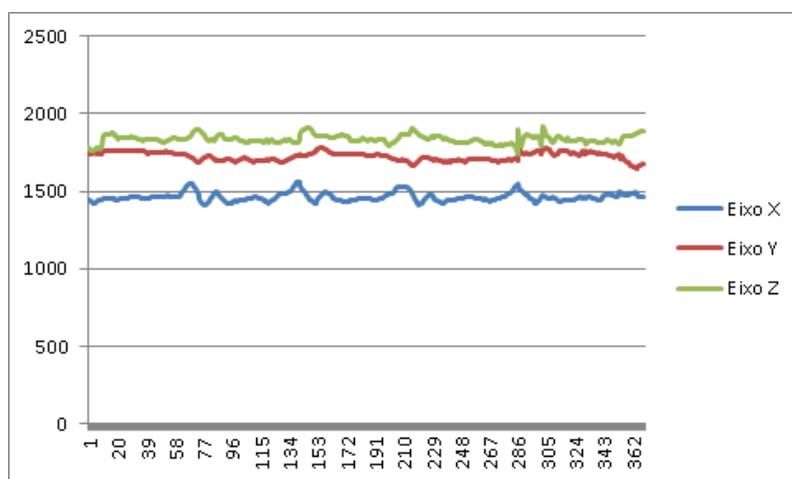


Figura 3.11: Andar.

Após a recolha das amostras, foi implementado o algoritmo. O algoritmo é descrito no artigo [35], estando este artigo mais focado numa implementação através de *hardware*. Este baseia-se na seguinte sequência de passos:

- 1 - **Filtro mediana** Permite remover o ruído apresentado pelos sinais recebidos;
- 2 - **Filtro FIR (*Finite Impulse Response*) passa-alto sobre uma janela de duração fixa** Permite remover a componente gravitacional do sinal, passando estes a estar com um valor médio de zero;

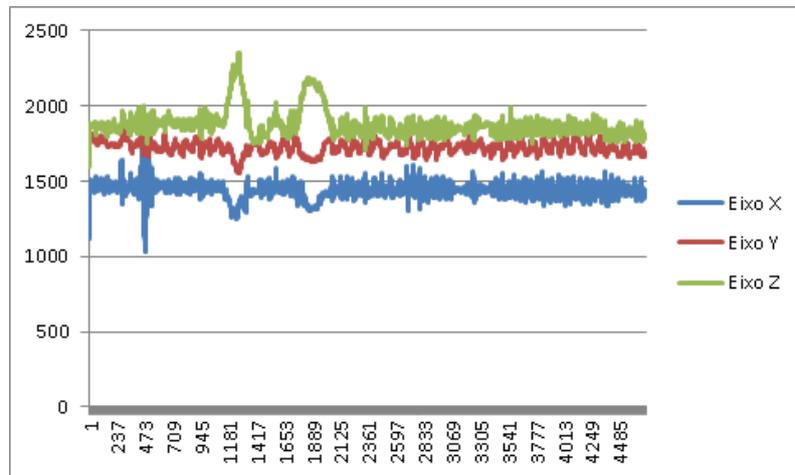


Figura 3.12: Vários movimentos seguidos (entre os quais andar, sentar, levantar, abaixar, etc.).

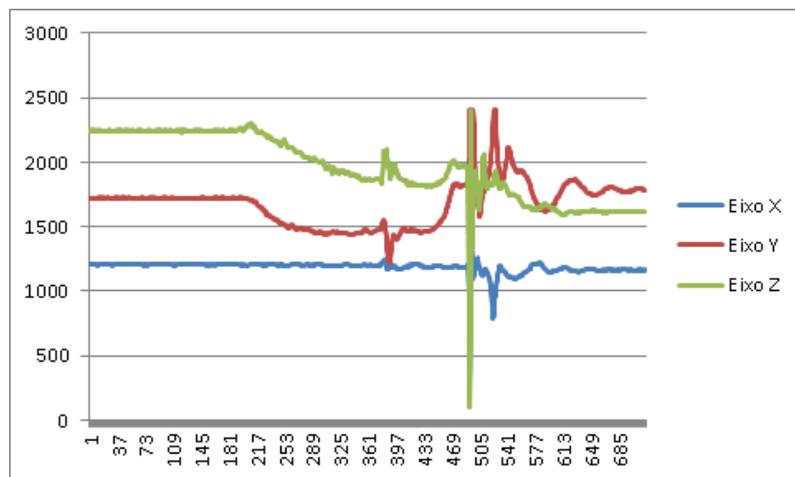


Figura 3.13: Queda normal.

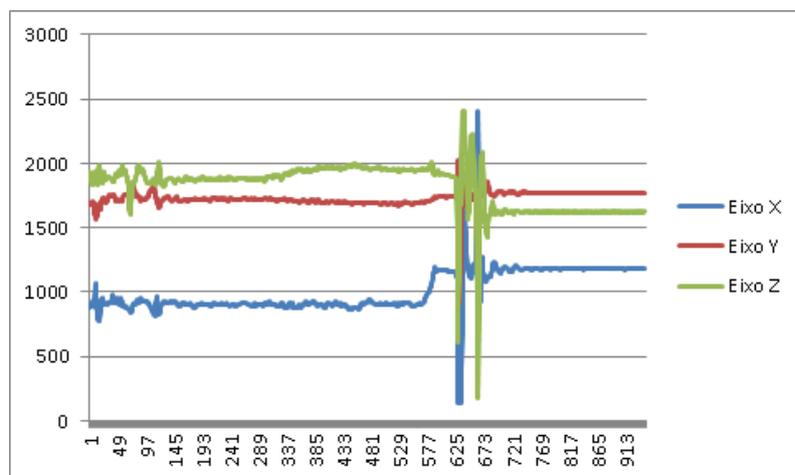


Figura 3.14: Queda de cama.

3 - Cálculo do gasto energético relativo à janela Permite avaliar numericamente o movimento de um ponto de vista energético;

4 - Verificação da existência de queda Permite, ao comparar os valores calculados no ponto anterior com um valor de *threshold*, classificar movimentos como sendo queda ou não queda;

Neste conjunto de passos, existem vários parâmetros que influenciam o resultado do algoritmo:

- Ordem do filtro mediana;
- Duração da janela;
- Valor do *threshold*.

No artigo [35], o filtro mediana tem ordem 13 e a duração da janela é 0.8 segundos.

Numa primeira fase, foi desenvolvida uma aplicação Matlab que permitisse visualizar os dados recolhidos anteriormente e analisar a forma como cada parâmetro afeta o resultado do algoritmo. A aplicação desenvolvida permite definir os parâmetros do algoritmo e verificar o estado do algoritmo em todos os seus passos. Ao iniciar a aplicação, é apresentada a interface da figura 3.15.

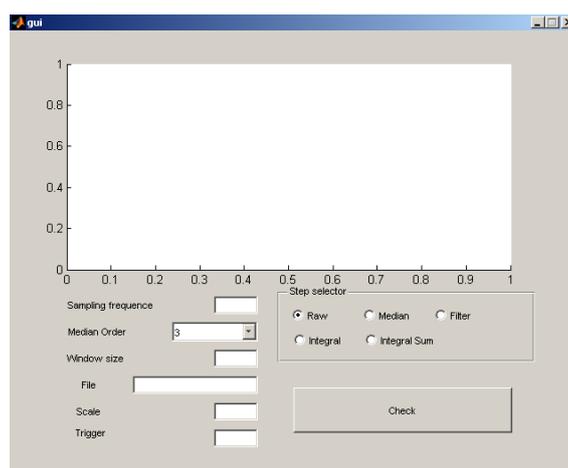


Figura 3.15: Interface gráfica da aplicação implementada via Matlab.

Na parte superior são apresentados os dados graficamente.

Na parte inferior esquerda, pode-se configurar o algoritmo através dos parâmetros citados anteriormente. Nota-se que existem três parâmetros extras: a frequência de amostragem, a escala e o *trigger*. A escala e a frequência de amostragem são parâmetros que foram adicionados porque, apesar de as medições terem sido feitas pelo mesmo conjunto de sensores, também foram utilizadas medições proveniente de uma outra placa. As duas placas tinham frequências de amostragem diferentes e possuíam fatores de escala diferentes, aplicados aos valores lidos. Ao adicionar estes dois parâmetros à interface, foi possível processar a informação dos dois conjuntos de leituras (leituras feitas pela primeira placa de sensores e pela segunda placa de sensores) com a mesma aplicação. O *trigger*, que aparece na interface, limita-se a desenhar uma linha horizontal na interface, com o valor definido.

Na parte inferior direita, pode-se seleccionar qual o passo do algoritmo a apresentar graficamente.

Os resultados obtidos com o uso desta aplicação para o processamento dos sinais do sistema de detecção de quedas desenvolvido são apresentados no capítulo seguinte.

3.4 Hardware de interface

3.4.1 Objectivo

Actualmente, não existem no mercado dispositivos móveis (PDAs, smartphones, etc.) que suportem o *standard* 802.15.4 ([36]), que é um dos *standards* mais utilizados pelas redes de sensores sem fios (apesar das muitas promessas desses virem a ser implementados). Para conseguir comunicar com uma rede sem fios através de um dispositivo móvel, é então necessário utilizar um módulo intermédio, que permite fazer a ligação entre uma interface física do PDA e a rede de sensores sem fios que comunica via 802.15.4. Existem no mercado várias implementações de módulos deste tipo. Por exemplo, a empresa Libelium (<http://www.libelium.com>) disponibiliza o XBee Programmable USB Stick (ver Figura 3.16), um módulo programável, alimentado por pilhas, que se liga via USB.



Figura 3.16: XBee Programmable USB Stick (Fonte: [12]).

Existem outras implementações (UZBee ZigBee USB dongle da empresa RF Solutions, ETRX2USB da empresa Telegesis, entre outros). Estes módulos têm como principais inconvenientes o preço, o tamanho e o facto de estes utilizarem um conector do tipo USB-A, que pode ser utilizado directamente em *desktops* e *laptops*, mas que para ser utilizado em PDAs e *smartphones* precisa de um adaptador (para passar para uma interface mini-USB ou micro-USB, que são as mais difundidas nos dispositivos móveis).

Por estes motivos, foi decidido implementar um módulo original.

3.4.2 Análise do problema e montagem electrónica

A componente de electrónica desta dissertação começou pela análise de um *development kit* CC2430DK (Texas Instruments), utilizado no projeto QWiSNet. Este módulo (CC2430EM, apresentado na Figura 3.17) é composto por um circuito integrado integrado CC2430 (que inclui um microcontrolador baseado no 8051, um *transceiver* compatível com a norma IEEE 802.15.4 e uma UART), um conector para uma antena externa e por componentes secundários (cristais, resistências, etc.).



Figura 3.17: Módulo CC2430EM.

Os portos do CC2430 estão disponíveis através dos conectores dos CC2430EM. Estes portos disponibilizam vários pinos: 4 correspondem a uma UART, que é utilizada como interface com o módulo e os restantes pinos dos conectores são:

- 2 pinos de alimentação;
- 1 pino de reset;
- 10 pinos de ground;
- 15 pinos de I/O programáveis.

O CC2430EM tem 2 conectores, cada um com 2x10 pinos, totalizando um total de 40 pinos. Fazendo as contas, chega-se à conclusão que 8 pinos não são utilizados.

Depois de analisado o CC2430, a primeira tarefa foi seleccionar que tipo de interface utilizar para a ligação do módulo. As possíveis interfaces para esta implementação são:

- USB (*Universal Serial Bus*)
- SD (*Secure Digital*)
- IRDA (*InfraRed Data Association*)
- Bluetooth

A ligação via infravermelhos, por obrigar a um alinhamento entre o módulo 802.15.4 e o dispositivo móvel foi logo descartada.

A interface Bluetooth, por obrigar a utilização de dois módulos distintos (o que torna a utilização do dispositivo móvel menos 'cômoda') também foi descartada.

Sobrando as interfaces SD e USB, optou-se inicialmente por utilizar a interface SD.

O primeiro passo foi encontrar um integrado que implementasse o standard SD.

Após alguma pesquisa em lojas online de eletrónica, chegou-se à conclusão que a aquisição de um módulo que servisse de intermediário entre a interface UART e a interface SD não seria possível (ou porque apenas era possível encomendar a partir de um número elevado de unidades, ou porque algumas empresas simplesmente não responderam aos contactos feitos). Optou-se então

por implementar o módulo de comunicação através da interface USB, para a qual é muito mais fácil encontrar componentes de ligação. Um destes componentes é o integrado FT232R, da empresa FTDI, que, para além de servir de interface entre uma UART e uma porta USB, possibilita configurar os níveis de tensão da interface UART.

Tendo o CC2430 e o FT232R, o passo seguinte foi interligar os dois.

No *datasheet* do FT232R, é apresentado um esquema onde o integrado é alimentado pela porta USB (um cabo USB tem 4 fios: VCC, Data+, Data- e GND).

O FT232R tem um pino de output, chamado 3V3OUT, que oferece uma tensão de 3.3V. Ligando o pino VCCIO (utilizado para controlar o nível da UART) ao pino 3V3OUT, obtêm-se os níveis de tensão nos pinos de I/O apresentados na Figura 3.1 (o uso dos 3.3 V como referência para os níveis dos pinos de I/O deve-se ao facto que o CC2430 tem como máximo de alimentação o valor 3.6 V, que leva a um valor máximo de tensão nos pinos de I/O igual a estes 3.6 V. estando estes 2 valores mais próximos de 3.3 V do que dos 5 V de alimentação do FT232R).

Tabela 3.1: Características DC do integrado FT232R (Fonte: [13]).

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.2	2.7	3.2	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Para que a comunicação entre os módulos seja funcional, é necessário que:

$$VMin_{receptor} \leq VMin_{emissor} \leq VMax_{emissor} \leq VMax_{receptor} \quad (3.1)$$

Esta equação tem de ser verdadeira nos dois níveis lógicos (0 e 1). Utilizando os valores de tensão de entrada do CC2430 (apresentados no *datasheet*) e os valores de I/O do FT232R (Figura 3.1), obtêm-se o esquema apresentado na Figura 3.18. Esta figura permite calcular a tensão com a qual se tem de alimentar o CC2430. Para que não haja problemas na ligação dos dois módulos, é necessário que:

$$3.2 \leq VDD \quad (3.2)$$

$$2.2 \geq VDD - 0.5 \quad (3.3)$$

$$VDD \leq 3.8 \quad (3.4)$$

$$VDD - 0.5 \geq 1.5 \quad (3.5)$$

De salientar que estas inequações apenas dizem respeito ao nível lógico 1, pois no que diz respeito ao nível lógico 0, o VDD não entra nas contas, e não há problemas de níveis de tensão:

$$0 \leq 0.3 \leq 0.5 \leq 0.9 \quad (3.6)$$

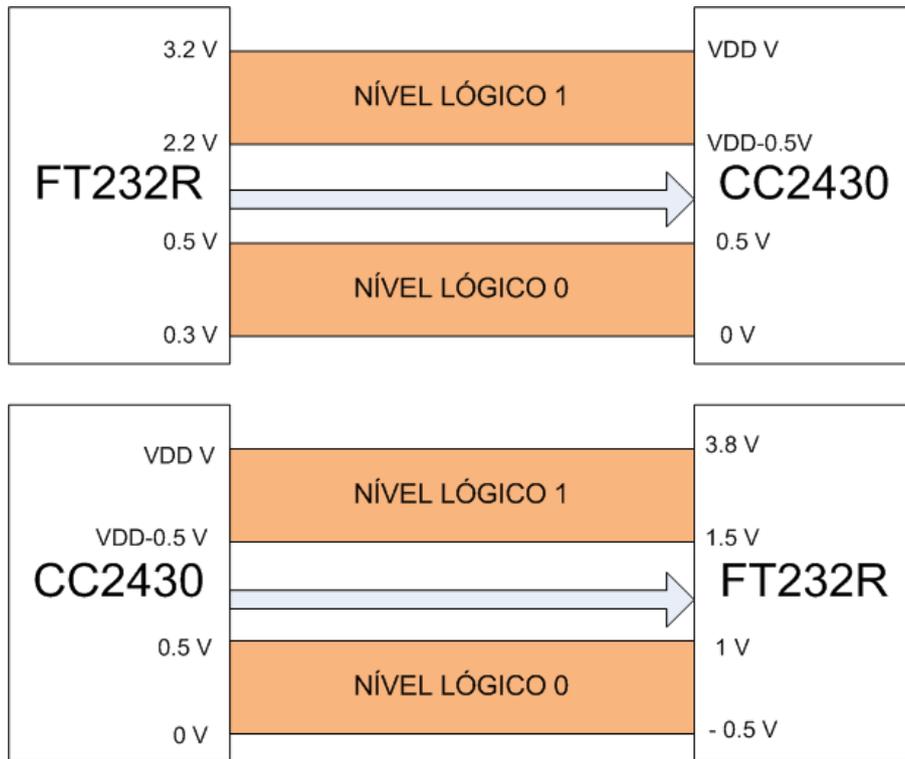


Figura 3.18: Ligação entre o FT232 e o CC2430.

$$-0.5 \leq 0 \leq 0.25 \leq 1 \quad (3.7)$$

Simplificando as equações anteriores, obtém-se:

$$3.2 \leq VDD \quad (3.8)$$

$$2.7 \geq VDD \quad (3.9)$$

$$VDD \leq 3.8 \quad (3.10)$$

$$VDD \geq 2.0 \quad (3.11)$$

As duas primeiras equações são incompatíveis (como se pode ver pela representação gráfica na Figura 3.19, onde a zona entre os 2.7 V e os 3.2 V está apenas coberta por 2 equações). Porém, estas contas foram feitas tendo por base valores mínimos e máximos dos níveis lógicos. Uma vez que dificilmente estes valores são atingidos, pode-se dizer que um valor de VDD igual a 3 V deve ser satisfatório.

Tendo como tensão de referência os 3.3V, pode-se criar um divisor de tensão (ver Figura 3.20) de modo a conseguir os 3 V. Este divisor de tensão é composto por 2 resistências, tendo uma um valor 10 vezes superior à outra.

Após estas contas, passou-se para a implementação do circuito. Para a elaboração deste, são necessários:

- 1 Conector USB

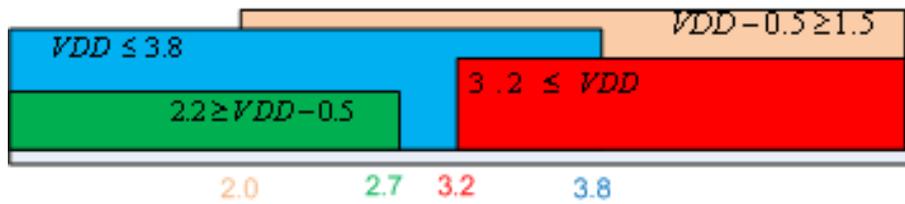


Figura 3.19: Representação gráfica das equações de adaptação entre os 2 módulos.

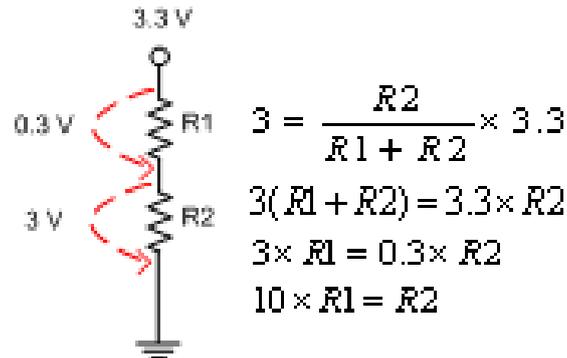


Figura 3.20: Divisor de tensão.

- 1 Condensador de 10 nF
- 1 Condensador de 100 nF
- 1 *Ferrite bead*
- 1 Integrado FT232R

O uso da *ferrite bead* não é obrigatório, tendo como principal função reduzir a interferência eletromagnética. Optou-se por dispensar este componente.

Quanto ao CC2430EM, apenas são necessário 2 conectores para ligar o componente à PCB. Os conectores são disponibilizados pela empresa Samtec ([37]), e são do tipo *Tiger Eye Header*.

O desenho do circuito foi feita através da ferramenta PADS Logic ([38]), que permite, após completada a inserção do circuito, passar a montagem para uma PCB (*Printed Circuit Board*), gerando os ficheiros necessários para a sua fabricação. Para uma melhor compreensão do circuito, optou-se por dividi-lo em partes (em anexo está uma análise das várias ligações efetuadas entre os componentes).

Após desenhado o circuito, e com a ajuda da ferramenta PADS Layout, obteve-se a PCB representada na Figura 3.21.

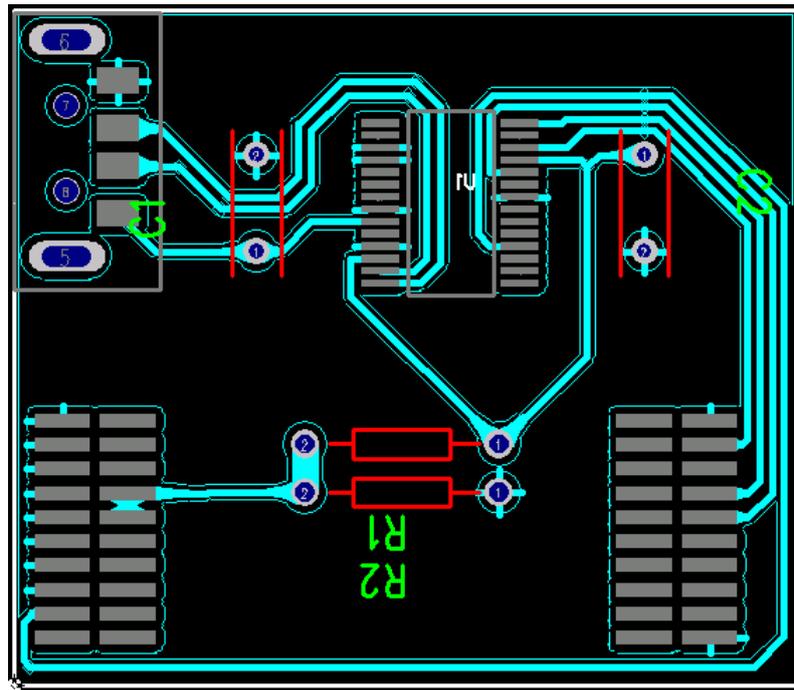


Figura 3.21: Representação da PCB pela ferramenta PADS Layout.

CAPÍTULO 4

RESULTADOS E DISCUSSÃO

I was bold in the pursuit of knowledge, never fearing to follow truth and reason to whatever results they led, and bearding every authority which stood in their way.

Thomas Jefferson

Conteúdo

4.1	<i>Gateway</i> inter-redes	45
4.2	Detecção de quedas	46
4.2.1	Cenário com ADLs	46
4.2.2	Situação de queda	49
4.2.3	Eficácia do sistema	52
4.3	Hardware de interface	54

4.1 *Gateway* inter-redes

O *gateway* inter-redes é um componente que, no âmbito desta dissertação, está um pouco limitado, pois as necessidades eram apenas receber dados do sensor e encaminhar para os destinatários. A nível de interface, apenas é possível definir regras que encaminham mensagens da interface USB para as outras interfaces. Os módulos de comunicação desenvolvidos têm as funções de envio e receção implementadas, mas apenas algumas estão ativas. A não conclusão do módulo de *hardware* obrigou a encontrar outras formas de testes ao sistema. Optou-se por simular o módulo de *hardware* através do envio-receção de mensagens via Wi-Fi.

No que diz respeito a testes, o *gateway* comportou-se como era esperado. O uso da SMS do forma de encaminhamento dos dados da rede de sensores para o exterior é muito limitada, devido à sua baixa capacidade de envio (o módulo desenvolvido permite enviar uma SMS por cada mensagem recebida da rede de sensores ou agrupar várias leituras numa SMS; nos dois casos, este módulo, em situações em que a taxa de receção de dados proveniente da rede de sensores é elevada, não conseguia responder de acordo com o desejado). O módulo Bluetooth tem o problema de este demorar algum tempo encaminhar os dados. A agregação de várias leituras permite de alguma forma contornar esta limitação. Os módulos com melhor desempenho foram os módulos associados à interface Wi-fi.

Por si só, este módulo de software é muito simples. Envia dados de uma interface para outra interface. Porém, se utilizado como base para outros sistemas, este módulo permite criar uma espécie de abstração entre uma WSN e um outro dispositivo. Por exemplo, um computador que precise de ler os valores de um sensor não precisa de saber de que forma se tem de ligar à WSN. Apenas precisa de ficar à escuta numa dada porta para receber os dados, desde que o encaminhamento esteja bem configurado no dispositivo móvel.

4.2 Detecção de quedas

O detetor de quedas foi de certa forma o módulo mais trabalhado. Os primeiros resultados apresentados a seguir são para um cenário com ADLs, sem quedas, enquanto os resultados seguintes incluem uma situação de queda.

4.2.1 Cenário com ADLs

Tendo como base o cenário de movimentações variadas sem quedas (andar, sentar-se, levantar-se, etc.), o algoritmo é analisado passo a passo nas figuras a seguir. A duração desta medição é 47 segundos, tendo sido obtidas 4710 leituras dos três sensores.

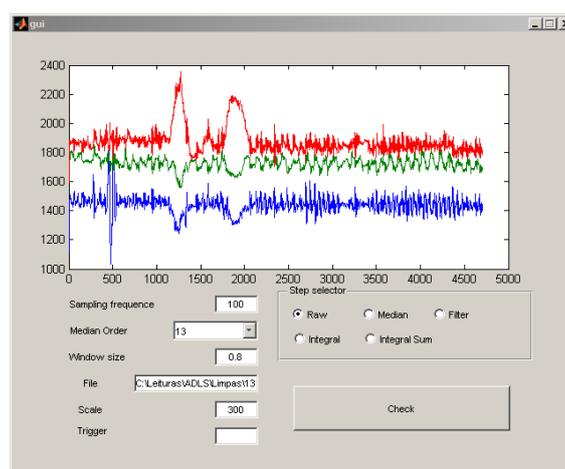


Figura 4.1: Visualização dos dados provenientes dos sensores (sem qualquer alteração).

Na figura 4.1, são apresentados os dados da medição sem qualquer tipo de alteração. Ao analisar o gráfico, pode-se ver que durante a medição, ocorreram 3 situações em que os os valores lidos

pelos sensores sofreram uma alteração mais brusca, situações correspondentes às ADL "levantar", "sentar" e "abaixar". No restante tempo, a movimentação foi mais uniforme, correspondente à ADL "andar". O passo seguinte do algoritmo é remover o ruído dos sinais. Neste passo, antes de aplicar o filtro mediana, os sinais são escalados com o fator explicado anteriormente (e definido na interface da aplicação, tendo, neste caso, o valor 300). Utilizando como ordem de mediana o valor definido no artigo(13), obtêm-se os sinais apresentados na figura 4.2.

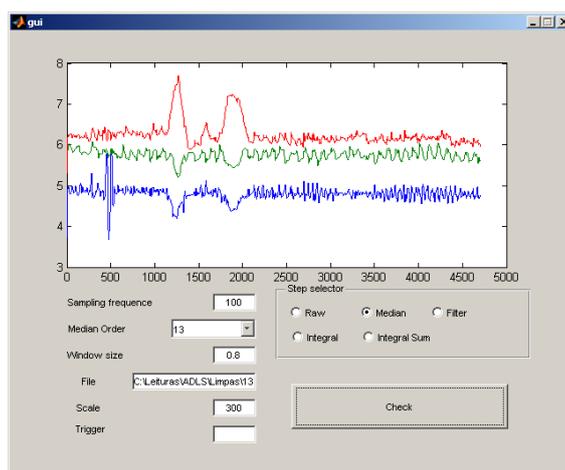


Figura 4.2: Aplicação do filtro mediana de ordem 13.

Este filtro remove o ruído dos sinais. Quanto menor a sua ordem, menos ruído é removido. Porém, caso o seu valor seja muito elevado, a saída do filtro pode ser muito diferente do sinal original, sendo assim um problema para o algoritmo. Os dois fenômenos podem ser visualizados nas figuras 4.3 e 4.4.

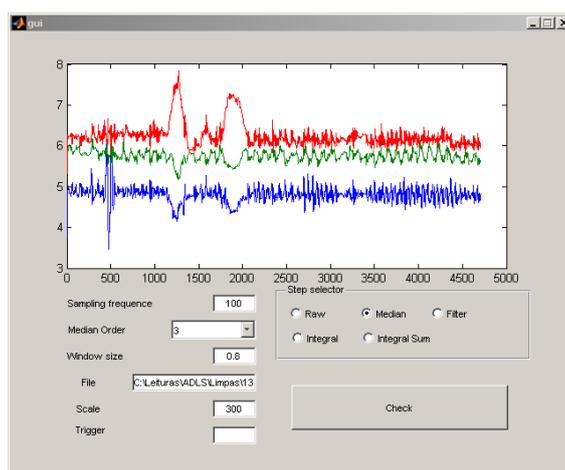


Figura 4.3: Aplicação do filtro mediana de ordem 3.

A seguir ao filtro mediana, é aplicado um filtro FIR passa-alto. Este filtro remove a componente gravitacional dos sinais, passando estes a ter um valor médio de 0. Tem como parâmetro a duração da janela sobre a qual o filtro é aplicado. Utilizando os valores apresentados no artigo, obtêm-se os sinais apresentados na figura 4.5.

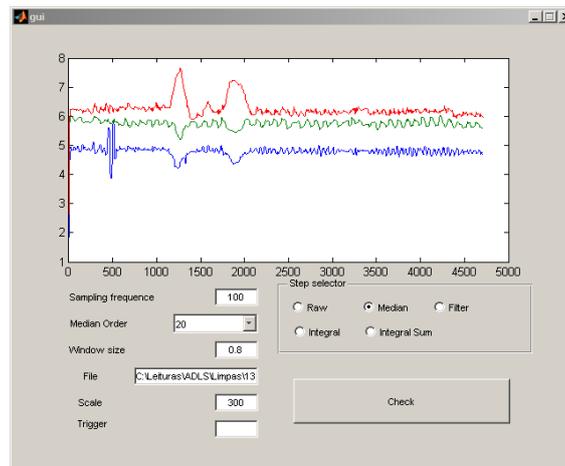


Figura 4.4: Aplicação do filtro mediana de ordem 20.

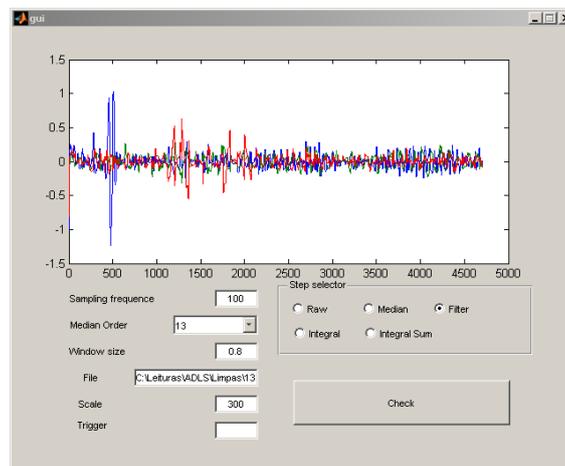


Figura 4.5: Aplicação do filtro passa-alto sobre uma janela de 0.8s.

Neste filtro, a janela permite de certa forma controlar a amplitude máxima do sinal: quanto mais pequena a janela, mais pequena é a amplitude do sinal resultante. Este efeito pode ser observado nas figuras 4.6 e 4.7.

O passo seguinte do algoritmo é calcular o gasto energético em cada janela. A forma como calcular este valor é apresentada no artigo [39], como sendo:

$$energyExpenditure = \alpha IA = \alpha \left(\int |x^2| + \int |y^2| + \int |z^2| \right) \quad (4.1)$$

onde α é uma constante de proporcionalidade e IA é um integral de área. Na interface da aplicação, são apresentadas 2 opções de apresentação para este filtro:

- Integral;
- Integral sum.

A primeira opção corresponde à apresentação dos integrais de X, Y e Z em separado. A segunda

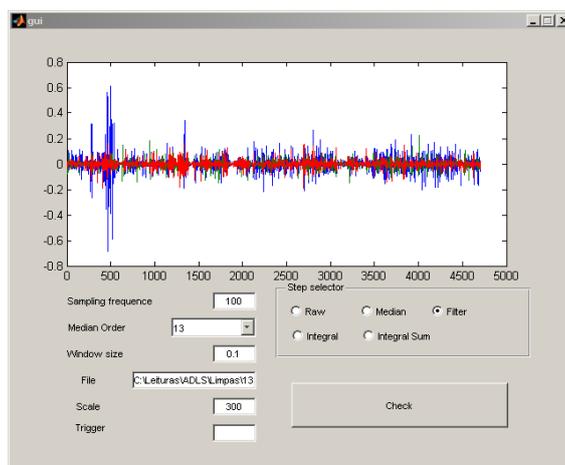


Figura 4.6: Aplicação do filtro passa-alto sobre uma janela de 0.1 s.

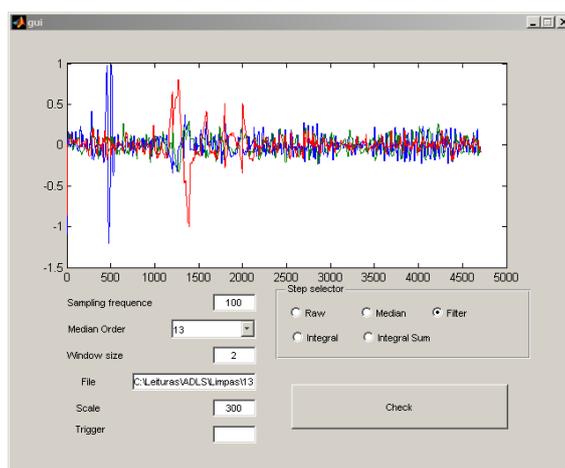


Figura 4.7: Aplicação do filtro passa-alto sobre uma janela de 2 s.

corresponde à soma destes 3 integrais. Utilizando os parâmetros definidos no artigo, obtêm-se os resultados apresentados nas figuras 4.8 e 4.9.

Tal como foi analisado no início da explicação do algoritmo, podem-se ver várias regiões onde o gasto energético é maior. Estas regiões correspondem às ADL "levantar", "sentar" e "abaixar". De uma forma muito resumida, o que interessa neste último gráfico são os máximos obtidos, uma vez que o *threshold* deve estar localizado acima dos valores máximos de gasto energético em situações normais, mas abaixo dos valores máximo de ADL em situações de queda.

4.2.2 Situação de queda

Analisando uma situação de queda com a mesma aplicação (neste caso, uma queda de uma cama), obtêm-se os resultados apresentados nas figuras 4.10, 4.11, 4.12, 4.13 e 4.14.

Como se pode ver no gráfico da figura 4.14, o gasto energético neste movimento é muito superior ao gasto energético em situações do quotidiano. No caso de queda apresentado, o valor do gasto energético máximo é o dobro do valor do gasto energético máximo apresentado no caso

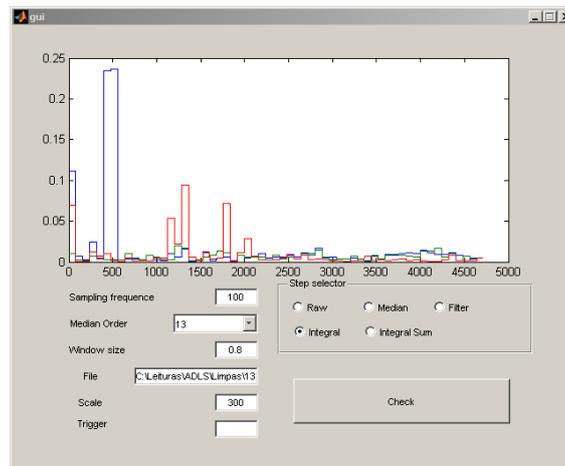


Figura 4.8: Apresentação gráfica dos 3 integrais.

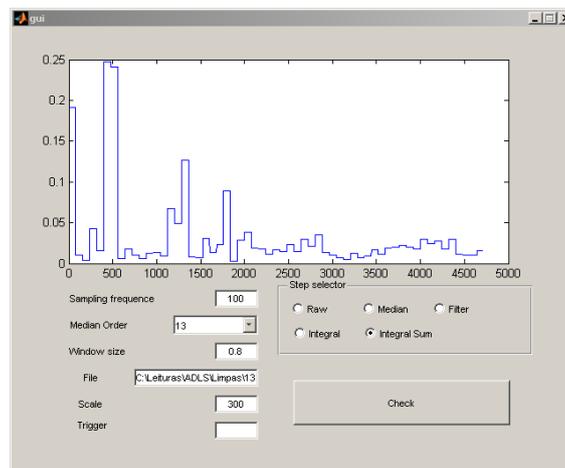


Figura 4.9: Apresentação gráfica da soma dos 3 integrais.



Figura 4.10: Visualização dos dados provenientes dos sensores (sem qualquer alteração).

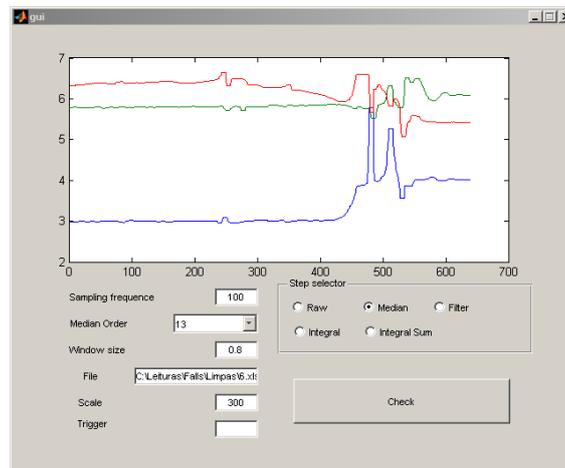


Figura 4.11: Aplicação do filtro mediana de ordem 13.

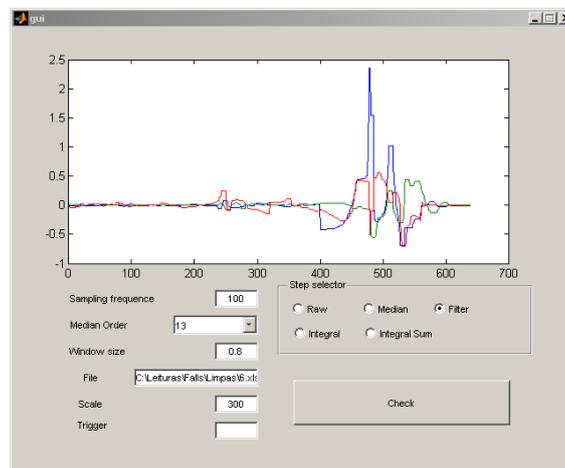


Figura 4.12: Aplicação do filtro passa-alto sobre uma janela de 0.8s.

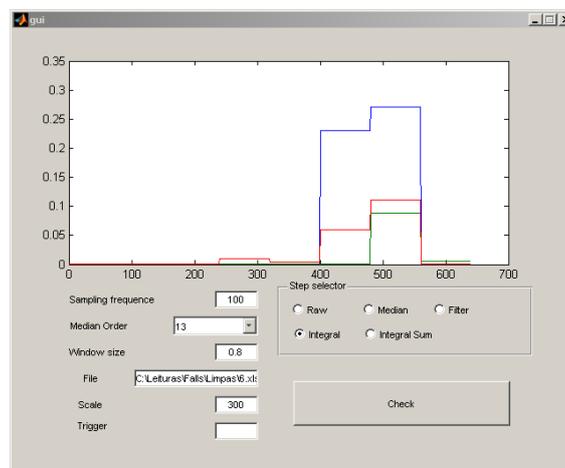


Figura 4.13: Apresentação gráfica dos 3 integrais.

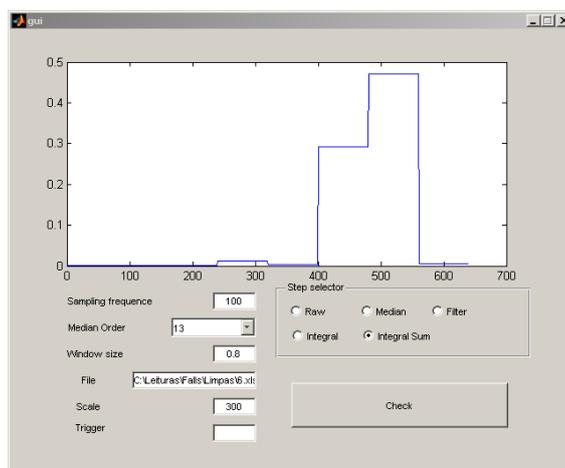


Figura 4.14: Apresentação gráfica da soma dos 3 integrais.

de movimentações variadas.

Porém, existem situações mais complexas, como é o caso de movimentações mais bruscas (como é o caso dos saltos). Utilizando o algoritmo sobre um conjunto de dados recolhidos pelos sensores durante vários saltos seguidos, o gráfico correspondente ao gasto energético é o apresentado na figura 4.15. Nota-se que os valores medidos ultrapassam muito os valores calculados na situação de queda apresentados. É por isso complicado distinguir entre quedas e movimentações bruscas.

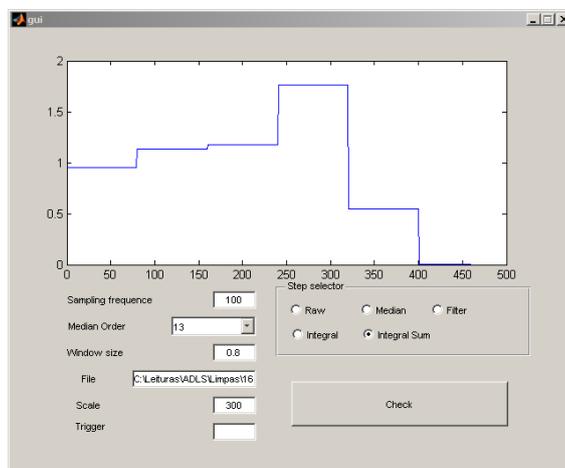


Figura 4.15: Apresentação gráfica da soma dos 3 integrais em situações de movimentações bruscas.

4.2.3 Eficácia do sistema

Em termos de resultados, foram feitos testes com 17 medições de ADLs (entre as quais algumas medições em que são feitas várias movimentações) e 7 medições de quedas (quedas de pé, de uma cama, etc.). Para todas as medições, o algoritmo implementado foi utilizado com uma vasta gama de parâmetros:

- Duração da janela : [0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2,3,4] (s);

- Ordem da mediana : [5,7,9,11,13,15].

Para cada combinação de duração da janela e ordem da mediana, foram calculados os 2 primeiros máximos da função resultante no último passo do algoritmo. Para cada ficheiro de medição, foram calculados 144 máximos (12 (número de valores para a qual a duração da janela foi testada) * 6 (número de valores para a qual a ordem da mediana foi testada) * 2 (número de máximos calculados por combinação)), sendo depois cruzados os dados das leituras relativas às ADLs e os dados relativos às quedas, para se determinar a combinação ideal de parâmetros.

No que diz respeito à duração da janela (parâmetro utilizado no filtro passa-alto e no cálculo do gasto energético), valores entre 0.8 e 1s foram os que deram melhor resultado. Valores abaixo disso levam o algoritmo a detetar falsas quedas mesmo em situações normais. Valores acima de 1s levam a que algumas quedas (nomeadamente as quedas das camas) não sejam detetadas. Apesar de existir um elevado gasto energético durante a queda, a seguir a esta o gasto energético é muito baixo. Caso a janela dure mais que a queda, o gasto energético médio é atenuado devido ao baixo gasto energético pós-queda.

A ordem da mediana afeta menos o resultado do algoritmo. Tal pode ser visto pelas tabelas 4.1 e 4.2.

Tabela 4.1: Máximos absolutos calculados (ordem mediana = 3).

	Ficheiro 1	Ficheiro 2	Ficheiro 3	Ficheiro 4	Ficheiro 5
0,2	0,959305	1,16506	0,485634	0,65763	1,142442
0,4	1,185821	1,350005	0,622636	0,843576	1,258291
0,6	1,467087	1,417147	0,803673	1,164613	1,70574
0,8	1,281042	2,046842	0,821649	1,27124	1,842028
1	2,508978	2,090375	1,133354	1,567429	1,797898
1,2	2,549121	1,515606	1,114303	1,327787	1,908975
1,4	2,718061	2,298269	1,627966	2,092174	2,118657
1,6	1,332404	2,182813	1,2156	1,354551	1,940567
1,8	2,945387	1,563033	2,057677	2,463631	2,031434
2	2,626517	2,526301	2,021567	2,386734	2,412937
3	2,664495	2,366736	2,945771	3,11503	2,791192
4	4,176609	2,784874	1,6135	5,199184	3,75075

Tabela 4.2: Máximos absolutos calculados (ordem mediana = 5).

	Ficheiro 1	Ficheiro 2	Ficheiro 3	Ficheiro 4	Ficheiro 5
0,2	0,517783	0,38622	0,505402	0,199306	0,682722
0,4	0,906809	0,762626	0,09591	0,343129	0,186866
0,6	0,479935	0,873562	0,16183	0,498721	0,193973
0,8	1,034361	0,692634	0,166068	0,511783	0,291243
1	0,680675	0,71095	0,955887	0,483522	1,246865
1,2	0,735537	0,952427	0,433024	0,544366	0,358935
1,4	0,842693	0,794312	0,818716	0,840089	0,624069
1,6	1,066175	0,78441	0,539076	0,559625	0,403697
1,8	1,06388	0,990272	1,078233	1,112649	0,642731
2	0,847075	1,152258	1,20218	1,120485	0,909097
3	0,901948	0,999547	1,95285	1,703081	1,334983
4	2,255713	1,418125	1,597176	2,023065	1,184471

Nestas tabelas estão apresentados os máximos da função no último passo do algoritmo para 5 medições, em função da duração da janela. Nota-se uma maior variação do máximo fixando

a ordem da mediana e variando a duração da janela (ou seja, analisando os dados apresentados numa coluna de uma tabela), do que fixando a duração da janela e variando a ordem da mediana (ou seja, comparando o valor na posição X,Y da primeira tabela com o valor na posição X,Y da segunda tabela). Por exemplo, para a situação [ordem de mediana = 15] no ficheiro 4, o máximo passa de 0.199306 (janela = 0.2s) para 2.023065 (janela = 4s). Já para a situação [duração de janela = 0.8s] no ficheiro 4, o máximo varia entre 1.27124 (ordem de mediana = 3) e 0,511783 (ordem de mediana = 15). Não foi encontrado nenhum valor ideal para este filtro, tendo existido, porém, um melhor acerto do algoritmo para uma ordem de média à volta de 13.

No que diz respeito ao *threshold*, e tendo por base a análise dos máximos efetuada, para uma janela de 0.8 segundos e uma mediana de ordem 13, o *threshold* que melhor distingue as ADLs das quedas, nas situações medidas, encontra-se entre o 0.3 e o 0.4. Com este valor, todas as quedas são detetadas. Porém, das 17 medições de ADL medidas, duas são classificadas como queda pelo algoritmo (nessas duas medições são dados uns saltos um pouco bruscos). Estes dados estão expressos na matriz de classificação (*confusion matrix*) apresentada na tabela 4.3. Com base nesses resultados, em casos de monitorização de pessoas idosas, por exemplo, parece seguro dizer que o algoritmo deve funcionar com poucas falhas, pois as movimentações são mais limitadas. Para pessoas mais jovens, o algoritmo implementado pode não ser suficiente.

Tabela 4.3: Matriz de classificação obtida nos testes de desempenho do sistema de deteção de quedas.

	ADL Detetadas	Quedas detetadas
ADL reais	15	2
Quedas reais	0	7

4.3 Hardware de interface

Apesar de este módulo ter sido desenhado (ver figura 4.16), até a data de término da dissertação não se conseguiu colocar o módulo desenvolvido a funcionar como seria esperado. Esta limitação foi de alguma forma superada através da simulação do módulo, mas não existem resultados reais sobre o seu funcionamento.

Apesar do módulo de hardware não ter sido completamente implementado no âmbito desta dissertação, um módulo com características semelhantes foi implementado por dois alunos no âmbito da UC de projeto II do Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores da Universidade do Minho e com auxílio do bolsheiro de investigação Pedro Macedo [40]. O desenvolvimento deste novo módulo, que se encontra a funcionar conforme previsto, foi inspirado na escolha do circuito integrado FT232 para a interface UART-USB. O novo módulo utiliza o circuito integrado CC2530, em vez do CC2430 utilizado nesta dissertação. O CC2530 representa a geração seguinte dos circuitos integrados da Texas Instruments baseados na norma IEEE 802.15.4, apresentando diversos aprimoramentos em relação ao CC2430. O CC2530 também foi utilizado para implementar a versão mais recente do protocolo eLPRT que foi desenvolvido no âmbito do projeto QWiSNet, em detrimento do CC2430. Outra vantagem do novo módulo é a integração de todos os componentes (incluindo o FT232 e o CC2530) numa única PCB, diminuindo o seu tamanho.



Figura 4.16: PCB implementada.

Outro motivo para não se dar continuidade ao desenvolvimento do módulo de hardware foi o surgimento no mercado de cartões microSD que integram o CC2530. Estes cartões são bastante compactos, sendo assim ideais para utilização em dispositivos móveis como PDAs e smartphones. O cartão SDZ-537 [41], fabricado pela Spectec, além de proporcionar a interface IEEE 802.15.4, disponibiliza 512 MB livres de armazenamento. Além desse cartão, já existe atualmente uma outra versão (SDZ-539) com capacidade de 2 GB.

CAPÍTULO 5

CONCLUSÃO

Esta dissertação exemplificou a forma como dispositivos móveis e redes de sensores sem fios podem ser complementares. Tendo como base as principais limitações das redes de sensores sem fios (alcançe limitado, pouca diversidade de interfaces de comunicação, etc.), os módulos desenvolvidos foram pensados de modo a que a dissertação tivesse componentes que abrangessem várias áreas disciplinares e que a sua temática central fosse abordada a vários níveis (hardware, comunicação e aplicação). O módulo de hardware, peça fundamental para a integração das tecnologias 'dispositivos móveis' e 'redes de sensores sem fios', apesar de ter sido desenhado, não ficou concluído. Esta não-conclusão do módulo limitou muito os desenvolvimentos dos dois outros módulos, sendo utilizada a simulação deste através de vários meios (uso da interface Wi-Fi, uso de ficheiros com leituras provenientes de um acelerómetro, entre outras). O primeiro módulo de software desenvolvido (*gateway*) teve como principal objetivo servir de base para outros programas. Por exemplo, este módulo permite criar um log com todas as leituras recolhidas, ou encaminhar estas leituras para um servidor HTTP para futuro processamento. Foram desenvolvidos módulos de comunicação para todas as interfaces disponibilizadas pelo dispositivo móvel. Porém, apenas algumas seriam utilizáveis em cenários reais (pelos motivos apresentados na secção anterior). O segundo módulo de software desenvolvido (detetor de quedas) teve como principal objetivo implementar uma aplicação que mostrasse todo o potencial da integração entre dispositivos móveis e redes de sensores sem fios. Utilizando como base um algoritmo disponibilizado num artigo, foram feitas várias experiências de modo a conseguir obter os melhores resultados para a aplicação.

Finalizada a escrita desta dissertação, é visível que, do ponto de vista técnico, existem alguns aspetos que poderiam ser melhorados, e que correspondem ao possível trabalho futuro. Relativamente ao módulo de hardware, a principal tarefa seria pôr o módulo a funcionar como seria esperado. Os testes ao módulo de *hardware* foram adiados e quando estes foram feitos, o módulo não estava funcional, e o tempo que sobrava até a data inicial da conclusão desta dissertação não permitiu corrigir este aspeto. O *gateway* é o módulo que menos precisa de melhoria. Todas as interfaces de comunicação implementadas são configuráveis e permitem a transmissão de dados de diversas formas (por exemplo, utilizando a agregação de leituras) e o seu módulo de encaminha-

mento é ao mesmo tempo simples e flexível. As únicas melhorias a ser feitas seriam relacionadas com a interface gráfica da aplicação e com a forma de execução do programa. A interface gráfica atual, apesar de estar adequada às funcionalidades e ao modelo de dados da aplicação, é muito simples e pouco atrativa. Quanto à forma de execução, seria útil esta aplicação correr em *background* de modo a permitir que outras aplicações possam executar ao mesmo tempo. Por último, no que concerne ao detetor de quedas, a implementação do algoritmo foi concluída com sucesso. O simulador implementado através da ferramenta Matlab permite visualizar através de gráficos a forma como o algoritmo funciona, e foram calculados os parâmetros desejados para o seu bom funcionamento. Estes cálculos basearam-se no cruzamento de valores discretos para os vários parâmetros de entrada do algoritmo (ordem do filtro de mediana e duração da janela) e na análise dos 2 máximos obtidos para cada situação. A utilização de outros valores para os parâmetros de entrada ou uma outra forma de análise dos resultados das várias iterações efetuadas sobre o algoritmo talvez levassem a outros valores que poderiam tornar o detetor mais ou menos fiável. A utilização de mais dados, quer para definir os parâmetros ideais (através do algoritmo), quer para verificar os resultados (através da classificação correta ou incorreta do movimento) levaria quase sempre a uma melhoria do sistema. Hoje em dia, e como foi referenciado anteriormente, existem vários sistemas de monitorização física no mercado. A comparação de desempenho entre estes sistemas e os módulos desenvolvidos seria uma possível extensão para a temática da dissertação.

BIBLIOGRAFIA

- [1] V. Shnayder, B.-R. Chen, K. Lorincz, Thaddeus, and M. Welsh, "Sensor networks for medical care," Division of Engineering and Applied Sciences, New York, NY, USA, Tech. Rep., 2005.
- [2] D. Malan, T. Fulford-jones, M. Welsh, and S. Moulton, "CodeBlue: An ad hoc sensor network infrastructure for emergency medical care," in *In International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.
- [3] K. Wac, R. Bults, D. Konstantas, A. H. van, V. Jones, I. Widya, and R. Herzog, "Mobile health care over 3g networks: the mobihealth pilot system and service," in *Proceedings of 2004 Global Mobile Congress*. Delson Group Pub., 2004, pp. 71–76.
- [4] T. Gao, C. Pesto, L. Selavo, Y. Chen, J. Ko, J. H. Lim, A. Terzis, A. Watt, J. Jeng, B. rong Chen, K. Lorincz, and M. Welsh, "Wireless medical sensor networks in emergency response: Implementation and pilot results," in *Technologies for Homeland Security, 2008 IEEE Conference on*, 2008, pp. 187–192.
- [5] P. W. da empresa eHit. <http://www.ehit.fi/>.
- [6] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "NoiseTube: Measuring and mapping noise pollution with mobile phones," in *Information Technologies in Environmental Engineering*, ser. Environmental Science, R. Allan, U. Förstner, and W. Salomons, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ch. 16, pp. 215–228.
- [7] R. Honicky, E. A. Brewer, E. Paulos, and R. White, "N-smarts: networked suite of mobile atmospheric real-time sensors," in *Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*, ser. NSDR '08. New York, NY, USA: ACM, 2008, pp. 25–30.
- [8] X. Jiang, N. Y. Chen, J. I. Hong, K. Wang, L. Takayama, and J. A. Landay, "Siren: Context-aware computing for firefighting," in *Pervasive Computing*, 2004, pp. 87–105.
- [9] M. Ros *et al.*, "Using wireless sensors as selection devices for a multimedia guidebook scenario," *Workshop on Real-World Wireless Sensor Networks (RealWSN 2005)*, 2005.

- [10] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The bikenet mobile sensing system for cyclist experience mapping," in *Proceedings of the 5th international conference on Embedded networked sensor systems*, ser. SenSys '07. New York, NY, USA: ACM, 2007, pp. 87–101.
- [11] K. Lorincz and M. Welsh, "Motetrack: a robust, decentralized approach to rf-based location tracking," *Personal Ubiquitous Comput.*, vol. 11, no. 6, pp. 489–503, Aug. 2007.
- [12] P. W. da empresa Libelium. <http://www.libelium.com>.
- [13] F. Chip. FT232R USB UART IC Datasheet. [Online]. Available: <http://www.ftdichip.com/Products/ICs/FT232R.htm>
- [14] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, no. 9, pp. 6869–6896, 2009.
- [15] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2007.
- [16] P. W. do projecto CodeBlue. <http://fiji.eecs.harvard.edu/codeblue/>.
- [17] P. W. do projecto MobiHealth. <http://www.mobihealth.org/>.
- [18] P. W. do projecto NoiseTube. <http://noisetube.net/>.
- [19] P. W. do project N-Smarts. <http://www.cs.berkeley.edu/honicky/nsmarts/>.
- [20] Y. Inoue, A. Sashima, T. Ikeda, and K. Kurumatani, "Indoor emergency evacuation service on autonomous navigation system using mobile phone," in *Universal Communication, 2008. ISUC '08. Second International Symposium on*, 2008, pp. 79–85.
- [21] E. Bruns, B. Brombach, and O. Bimber, "Mobile phone-enabled museum guidance with adaptive classification," *Computer Graphics and Applications, IEEE*, vol. 28, no. 4, pp. 98–102, 2008.
- [22] P. W. do projecto BikeNet. <http://bikenet.cs.dartmouth.edu/>.
- [23] P. W. do projecto MoteTrack. <http://www.eecs.harvard.edu/konrad/projects/motetrack/>.
- [24] P. Macedo, "Desenvolvimento de modelos de simulação de redes de sensores sem fios," *Dissertação de Mestrado, Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores, Universidade do Minho*, Novembro de 2010.
- [25] J. A. Afonso, L. A. Rocha, H. R. Silva, and J. H. Correia, "Mac protocol for low-power real-time wireless sensing and actuation," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems (ICECS 2006)*, December 2006.
- [26] J. A. Afonso, H. D. Silva, P. Macedo, and L. A. Rocha, "An enhanced reservation-based mac protocol for ieee 802.15.4 networks," *Sensors*, vol. 11, pp. 3852–3873, April 2011.

- [27] H. Silva, "Protocolos de acesso ao meio em redes de sensores sem fios aplicadas à captura de movimentos corporais," *Dissertação de Mestrado, Mestrado Integrado em Engenharia de Comunicações, Universidade do Minho*, Dezembro de 2010.
- [28] H. D. Silva *et al.*, "Design and implementation of a wireless sensor network applied to motion capture," *Proceedings of 1st Portuguese Conference on Wireless Sensor Networks (CNR.S 2011)*, March 2011.
- [29] P. da empresa In The Hand. <http://32feet.net/>.
- [30] Freescale, "Human fall detection using 3-axis accelerometer," *Reference Manual*, December 2005.
- [31] A. Bourke, J. O'S'Brien, and G. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait and Posture*, vol. 26, no. 2, pp. 194 – 199, 2007.
- [32] Q. Li, J. Stankovic, M. Hanson, A. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, 2009, pp. 138–143.
- [33] J. M. Caroline Rougier, "Demo : Fall detection using 3d head trajectory extracted from a single camera video sequence," *First International Workshop on Video Processing for Security*, 2006.
- [34] A. Doulamis *et al.*, "A real-time single-camera approach for automatic fall detection," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, 2010.
- [35] R. Reyna and K. T. with Dr. Daniel Copado, "Beyond accidental falls - human fall detection using accelerometry," *Beyond Bits*, <http://freescale.com/beyondbits>, 2009.
- [36] "Ieee std 802.15.4-2006 ̄ part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans)," September 2006.
- [37] P. W. da empresa Samtec. <http://www.samtec.com>.
- [38] P. W. da ferramenta PADS. <http://www.mentor.com>.
- [39] M. Mathie, N. Lovell, A. C. F. Coster, and B. Celler, "Determining activity using a triaxial accelerometer," in *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, vol. 3, 2002, pp. 2481–2482 vol.3.
- [40] J. M. e Ana Trigo, "Estação base com interface usb para sistema de monitorização cinética sem fios," *Relatório final da UC de Projeto II, Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores, Universidade do Minho*, Julho de 2012.
- [41] Spectec, "Microsd zigbee card - sdz-537," <http://download.spectec.com.tw/sdz537.htm>, 2009.

APÊNDICE A

MONTAGEM ELETRÓNICA

Este apêndice permite explicar a forma como se desenhou o integrado na ferramenta PADS.

O conector USB (Lumberg 2410 07) é composto por 4 pinos:

- 1 pino de ground
- 2 pinos de dados
- 1 pino de alimentação

Os dois pinos de dados ligam directamente ao FT232R, assim como o pino de alimentação, utilizado para alimentar o FT232R (Figura A.1). Nestes esquemas, todos elementos representados por uma seta rectangular correspondem a nomes de malha. Se dois pinos estiverem ligados a malhas com o mesmo nome, estes estão directamente ligados.

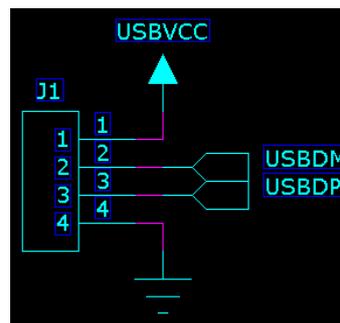


Figura A.1: Ligações relativas ao conector USB.

Tendo por base um circuito apresentado no *datasheet* do FT232R, onde é apresentada a forma como este é alimentado e como se ligam as linhas de dados, desenhou-se as ligações relativas a este integrado (Figura A.2).

Como se pode ver, os sinais USBVCC, USBDP e USBDM proveniente do conector USB (A.1) ligam directamente ao FT232R. Os pinos 3V3OUT e VCCIO estão ligados entre si (estão ligados

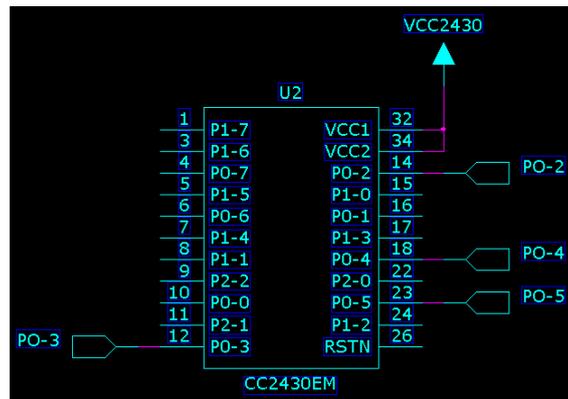


Figura A.5: Ligações relativas ao CC2430.

Na sua totalidade, o circuito desenhado é o da Figura A.6.

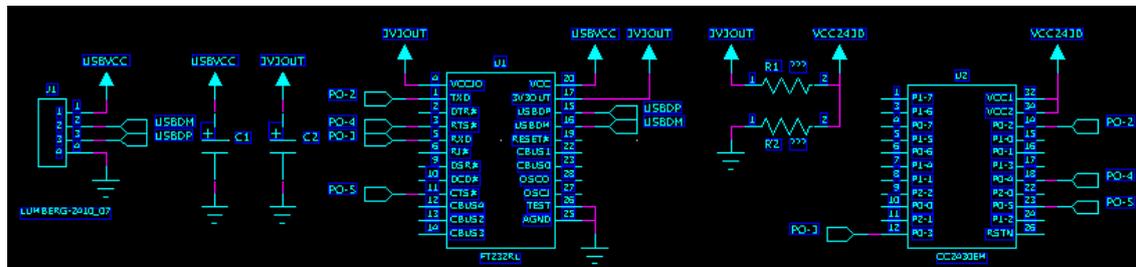


Figura A.6: Visão global da montagem eletrônica.