FRUGAL & SCALABLE FRAMEWORK FOR ROBUST & INTELLIGENT REMOTE MONITORING IN AN AGING DEMOGRAPHY

YUAN JIAN

(B.Eng. (Hons.), NUS

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Yuan Jian 15/12/2012

Yuan Jian

Acknowledgments

First and foremost, I would like to express my deepest appreciation to my supervisor, Prof. Lee Tong Heng, and my co-supervisor, Assoc. Prof. Tan Kok Kiong for their enlightenment, guidance and inspiration during my PhD candidature.

I would also like to thank my Thesis Advisory Committee members, Prof. Ben M. Chen and Dr. Koh Eng Kiat, for their reviews and invaluable suggestions.

I am thankful to my department, NUS Graduate School, for providing me with the PhD scholarship as well as financial support for numerous research activities.

I have had the pleasure of working with a number of talented graduate students at Mechatronics and Automation Lab. I owe a great deal of thanks to Ms. Er Poi Voon, who helped me explore digital accelerometers and discussed with me about embedded system designs. Special thanks must be made to Mr. Liu Lei for his advice on how to approach problems from theoretical perspectives. Great thanks to Mr. Tan Chee Siong for providing high-class laboratory environment for my research.

Thanks also go to NTUC Eldercare for allowing me to conduct a trial study at one of its subsidiaries – Jurong Central Daycare Centre.

Finally, I would like to thank my fiancee Jiannan for her endless love and support.

Table of Contents

Summary

Remote health monitoring not only enables the elderly to live independently but also offers peace of mind to their families. Currently, this need is met by PERS (Personal Emergency Response Systems) and their modern variations which detect falls, monitor heart rate, etc. However, PERS are expensive and have limited range due to their use of point-to-point wireless communications. Thus, they cannot scale up and be shared by multiple seniors.

This thesis proposes an economical and scalable health monitoring system, *e-Guardian*, for the elderly both at home and in communities. *e-Guardian* takes advantage of technologies such as wireless sensor networks (WSN), cellular networks, microelectromechanical systems (MEMS) and machine learning. The heart of the system is a Base Station (BS) which acts as a gateway between a local WSN and cellular networks. By placing a number of Range Extender (REs) around, the WSN can be extended to a large area. At seniors' sides are small and battery-powered Wearable Devices (WDs) which can automatically detect falls, infer activities of daily living (ADLs), possibly monitor body temperatures and heart rates, etc. *e-Guardian* is scalable so a single system can be shared by many, which not only reduces cost per capita, but also enables community

collaborations towards more cost-effective eldercare.

e-Guardian employs a low bandwidth approach to ensure scalability without network congestion. Data flowing in *e-Guardian* networks must be high-level information. This requires WDs to process raw sensor data locally instead of streaming to a remote unit. By taking advantage of advanced sensors which support low duty cycles or hardware interrupts, power-efficient algorithms can be designed to run inside resource-constrained WDs.

This thesis presents *e-Guardian*'s architecture and proposes various features. It then describes the realization of an *e-Guardian* prototype which implements all essential low-level functions for those features. Implemented features include simple location tracking, fall detection, SMS and web browser interfaces, reminders for medications, and a TCP/HTTP server. Finally, this thesis elaborates the designs of a fall detection algorithm and an ADL classifier for WDs. Both algorithms are interrupt-driven and consume minimum computational resources. ADL classification results can be fed back into the fall detection algorithm to improve accuracy, automatically placing the accelerometer in sleep mode, and adaptively changing duty cycles of sensors.

e-Guardian took a big step beyond laboratories by deploying *e-Guardian* at NTUC Eldercare center in Jurong Central (Singapore), for a trial study of one month. In this study, one BS, three REs were deployed and three seniors participated. The prototype system remained robust for the entire month. Statistics collected include the number of false positives of the fall detection algorithm and associated nearest locations.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Aging Demography

The world is rapidly aging, as a result of longer life expectancy and declining fertility rates [3]. In 2009, an estimated 737 million people were aged 60 years or over. This number is projected to increase to two billion in 2050 [4]. Most of this increase is occurring in developing countries, where the number of older people will rise from 400 million in 2000 to 1.7 billion by 2050 [3]. This demographic change is presenting tremendous challenges to both developed and developing countries, including strains on pension and social security systems, increasing demand for health care and long-term care [5].

From another perspective, population aging also represents an opportunity for societies. If older people can retain their health, their experience, skills and wisdom will continue to be a fortune for societies. However, most people at older ages need accessible and effective acute and long-term care. Developing seamless eldercare services thus has to be a priority for both developed and developing countries [3].

People confront increased health challenges as they age. Cardiovascular diseases remain the leading cause of death and disability, and cardiovascular risk increases steadily with age. Heart diseases and stroke incidences rise steeply after age 65. They can strike suddenly and could be fatal if assistance is not sought immediately [6].

Aging is normally accompanied by gradual reduction in muscle strength, which results in an increased risk of falling. There are another dozens of causes that contribute to falls, including poor lighting, stairs, poor balance as a result of stroke or medications, visual impairment, dementia, and cardiovascular diseases. Falls occur in 30-60% of older adults each year, and 10-20% of these result in injury, hospitalization and/or death. Without immediate help, seniors may suffer pain, emotional distress or even develop other complications including pneumonia, dehydration, hypothermia, etc [7]. Therefore, immediate help after fall or similar emergencies is of critical importance as it could lower the risk of complications and death, and greatly increase the likelihood of a return to independent living.

Aforementioned age-related problems are calling for a health monitoring system that can automatically detect emergencies, immediately raise alerts so family members and/or medical personnel could intervene in the earliest stage when interventions are the most effective.

1.2 Aging in Place

Aging in place is the ability to live in one's own home and community safely, independently and comfortably, regardless of age, income, or ability level. Statistics have shown that most seniors prefer to age in place as shown in Figure.1.1. In fact, in many Asian countries, it is part of the cultural beliefs that older adults should age in place, live with and be taken care of by their adult children. Helping the elderly live independently in their own home must be a priority for healthcare systems.

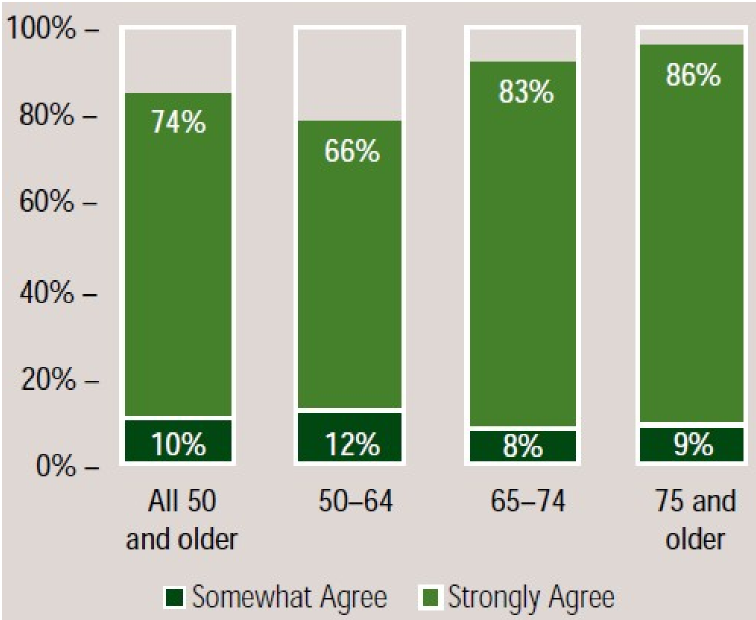

Figure 1.1: Most Seniors Prefer to Age in Their Homes [1]

In 2009, an estimated 14% of people aged 60 years or over live alone [4]. Even if the majority of them live with their family, many still spend considerable amount of time alone at home when others leave for work. A study showed that almost one-third of full-time working adults in the US served as informal caregivers, mostly to an elderly parent in 1997 [8]. Meanwhile, it is worth noting that, in 2009, nearly two thirds of people who were aged 60 or over lived in developing countries [4]. Even in developed countries like the United States, elderly poverty was as high as 16.1 percent in 2009 if cost of medical care is factored in [9].

There is an urgent need for delivering low-cost and effective health care to a rapidly growing aging population.

1.3 Evaluation of Existing Solutions

Currently, above needs are primarily addressed by commercial products marketed as Personal Emergency Response Systems (PERS), which are also known as medical alarms or medical alerts.

1.3.1 PERS

PERS has existed in the marketplace for more than 40 years. PERS typically consists of a pendant/watch with a microphone, a speaker, a radio transceiver and a panic button; a console connected to a home telephone system; and a remote emergency response center that monitors calls. Examples include Life Alert [10] (as shown in Figure 1.2) and Rescue Alert [11].

Figure 1.2: PERS: Life Alert Pendent

In case of emergencies, seniors may be unable to press buttons to summon help from emergency hotlines independently because they may be disoriented, immobilized or unconscious, thus missing the best time for treatment.

1.3.2 PERS 2.0

As technologies evolve, various sensors such as fall sensors, smoke detectors, motion detectors are gradually integrated into PERS. These emerging solutions for aging in place are coined as PERS 2.0.

1.3.2.1 Lifeline with AutoAlert

Lifeline with AutoAlert from Philips (as shown in Figure 1.3) offers everything that a legacy PERS has, but in addition to fall detection. It claims to have a range of 800 feet ($\approx$ 243 meters) and a battery life of 18 months. Its service costs US$53 a month, in addition to installation and monitoring fees [12].

Figure 1.3: Philips LifeLine with AutoAlert

1.3.2.2 Halo Monitoring

myHalo chest strap (as shown in Figure 1.4) from Halo Monitoring is a more comprehensive health monitoring solution which has a panic button, measures heart rate and skin temperature, and detects fall. It covers a range of 300 feet ($\approx$ 91 meters) [13]. It has only 48 hours of standby and charges a monthly fee of US$59 in addition to one-time equipment cost [14].

Figure 1.4: myHalo Monitoring

1.4 Limitations of PERS & PERS 2.0

The above systems are innovative and state-of-the-art PERS solutions currently in the market. However, to enjoy such services, users have to bear a financial burden monthly for services that their family members could almost equally provide. These systems typically rely on landline telephone systems or Ethernet cable connections, and also emergency response teams which are rare or inaccessible in remote or underdeveloped regions. As a result, PERS services seldom make their presence there. Even if there is any, monthly subscriptions are simply unbearable to most seniors in those areas.

Almost all PERS devices establish point-to-point connections with a base unit at home, which is analogous to how most cordless phones work. The advantage gained is high bandwidth allowing for seamless voice communications which are invaluable during emergency. However, a major drawback of point-to-point connection is limited range which is typically about a hundred meters, preferably with line-of-sight. The inability to scale limits PERS systems to single households only. In fact, the claimed ranges could be a problem for a single big house. In Lifeline's main web page [15], it clearly states at the bottom that "signal range may vary due to differing environmental factors". Also myHalo Monitoring states in its support forum, "myHalo Knowledgebase" [13], that "myHalo chest strap device has a range of at least 300 feet and will cover a typical 3,000 square foot house, as long as there is a direct line-of-sight between the chest strap and the gateway" and "note the myHalo gateway location will have a big impact on actual coverage, and will be tested thoroughly during installation".

In addition, most PERS vendors adopt a business strategy that one senior uses one PERS system. It is impossible for several seniors to share a single base unit. It is also impossible for care centers to integrate into their own solutions. Thus, such products could not penetrate markets in regions where they have not established local business representatives, and more importantly call centers and emergency response teams. For instance, at the time of this writing, Philips Lifeline with AutoAlert has only been available in the US and Canada while myHalo has only been available in the US.

In communications with various residential communities in Singapore, there is a strong demand for monitoring seniors' conditions at a community level, so that staff from neighborhood committees could intervene in the first minute after emergencies. Also, in discussion with a few eldercare centers in Singapore, they expressed their need for a systematic solution that can keep track of their senior clients' attendance, daily routine checks, accidental falls, unexpected leaves (especially for dementia patients). Each senior has more or less different health conditions and thus special care must be given to each senior. For example, some require routine rehabilitation exercises; some need to take medications after every meal; dementia patients always need close watch. Currently, these tasks are manually managed by caregivers. Such repetitive tasks keep caregivers constantly busy and are very likely to cause stress.

1.5 Motivation

The above reasons motivate the design of a home/community-based health monitoring and alert system called *e-Guardian*. The system is low cost, low power and highly scalable. *e-Guardian* aims to transform eldercare into a collaborative effort involving mainly people who know and care about the elderly, rather than a third-party vendor that provides service at a monthly service charge. By totally eliminating recurring monthly subscription fees, *e-Guardian* will be able to reach a much broader range of users, especially in the less developed regions when such need is generally overlooked. The scalability allows an *e-Guardian* system to be shared by hundreds of seniors living in the same residential area, thus reducing cost of use per senior to minimum.

1.6 Proposed System

e-Guardian takes advantage of recent advances in wireless sensor networks (WSN), hardware miniaturization, Microelectromechanical sensor (MEMS) technology, ubiquitous and mobile computing, and machine learning. The highlight of *e-Guardian* is a set of small, light weight and low-power wearable devices. It consists of miniaturized sensors and an MCU (microcontroller unit) with a RF (radio frequency) transceiver. It can detect accidental falls, monitor simple activities such as walking, casual movements and inactivity, and can be used to summon help. A base station relays data between wearable devices and caregivers. This system is simple and low-cost, yet still capable of alerting caregivers at those life-critical moments. *e-Guardian* features the utmost simplicity in design, deployment and operation. Minimum expertise is required to install and deploy *e-Guardian*. One could simply order a set of *e-Guardian* devices online and set them up within a few simple steps.

In remote or underdeveloped areas where it is difficult for medical services and ambulances to show up within tens of minutes, *e-Guardian* will be a reliable means for seniors to receive help as quickly as possible. In developed countries where PERS and similar services are common, it will be a low-cost alternative. Though designed primarily for the lone elderly, *e-Guardian* is by no means limited to the elderly only. After slight changes, it will work equally well for those who frequently need help or are more prone to falls, including babies, the handicapped and people with mobility impairments.

1.7 Contributions of Thesis

In this thesis, the following contributions have been made by the author:

1. A configurable, scalable and decentralized remote health monitoring system, *e-Guardian*, has been proposed, implemented and undergone a trial study. Traditional solutions such as PERS and PERS 2.0 are based on point-to-point wireless communications which make them impossible to scale up and be shared by multiple seniors. In contrast, *e-Guardian* can scale by using multi-hop WSNs. In order to scale freely without network congestions and high latency, *e-Guardian* takes a decentralized approach that sensor data is processed within wearable devices so as to save network bandwidth. Processing massive sensor data continuously drains battery quickly. This is avoided by designing sensor algorithms based on advanced digital sensors that support low duty cycling or interrupt/event-driven designs.

2. A power-efficient fall detection algorithm has been proposed for a digital accelerometer which supports interrupts and data buffering. This algorithm takes an interrupt/event-driven approach which is completely different from conventional approaches where an algorithm must examine and process every piece of data sampled at high frequencies. The interrupt-driven approach allows a host MCU to examine significantly less data and only process upon an accelerometer interrupt or a timer interrupt. Thus, it conserves both power and bandwidth.

3. An ADL (activities of daily living) algorithm is built upon the fall detection al-

gorithm. It takes advantage of resources which are already available in the fall detection algorithm thus additional power consumption is minimal. A few uses of ADL classification results are also proposed, including feeding back to the fall detection algorithm, saving more power by placing the accelerometer into sleep mode, and adaptively changing duty cycles of certain functions which are correlated with ADLs.

4. The proposed system has undergone a trial at an eldercare center. It proves that the prototype was robust for an entire month. It also collects statistics including the number of false positives of the fall detection algorithm within the month, and also nearest REs of fall alerts.

Chapter 2

Design Considerations

2.1 Design Criteria

Before the actual design of *e-Guardian*, a number of criteria were proposed and carefully reasoned for any system to be qualified for an effective remote health monitoring solution to the majority of the world's aging population.

1. **Simplicity**: Such system ought to be simple in design and easy to use. Wearable parts, if any, should be small, lightweight and non-invasive. Seniors must not be required to check, read or operate them in any complicated way other than pressing one or two buttons. In addition, setups should require minimal expertise so it could be easily adopted.

2. **24/7 & Realtimeness**: Monitoring is expected to be always on and alerts must be real-time. Wearable parts are preferred to be waterproof so seniors do not need to take them off when bathing during which falls are mostly likely to occur.

3. **Ubiquitousness**: Preferably it should also work when seniors go out occasionally and even track their geographical locations.

4. **Affordability**: It should be economical so that the majority of the elderly in the world could afford one, or local governments or charities could support them without heavy financial burdens.

5. **Power-efficiency**: People become increasingly forgetful as they age. Thus, it is desirable that charging/replacing batteries of wearable parts should be as infrequent as possible.

6. **Scalability**: A solution is significantly cheaper (per capita) if shared by more people. In this sense, scalability boosts affordability. In large care centers, scalability is an absolute requirement for management purposes. In extreme cases, the solution should effectively monitor either a single senior or thousands of seniors, however sparse or dense their living environments are.

7. **Adaptability**: To suit the elderly with different social and cultural backgrounds, the solution should be able to accommodate and adapt to differences in their living environments, whether in single households or densely populated care centers, whether in remote rural villages or metropolitan residential communities, whether in developed or developing countries, etc.

8. **Autonomy**: The solution should be autonomous in the sense that it can be operated completely independently of any company if needed. That is, service providers can be family members, relatives, caregivers, doctors, social workers, neighbors. This implicitly requires a solution to be highly configurable. In this

way, users could customize or integrate the solution to their existing solutions. Autonomy also boosts affordability by eliminating service charges.

2.2 Technology Background

2.2.1 MEMS Accelerometers

MEMS sensor is an emerging technology. Microscopic machines are fabricated on silicon wafers using techniques developed in Integrated Circuits (IC). Advances in MEMS technology offer unprecedented abilities to incorporate miniaturized and cheap sensors into mobile devices. Common MEMS sensors include inkjet printheads, accelerometers, gyroscopes, microphones, pressure sensors, etc. MEMS accelerometers, as used in *e-Guardian*, have extensive applications in handsets, game consoles, image stabilization in cameras, hard disks, automotive airbags, anti-theft systems and various industrial sectors [16]. It is due to massive applications in these areas that MEMS sensing technologies are rapidly advancing and cost of production continues to decrease.

MEMS has gone through four generations since its emergence [17]:

1. Sensor element mostly based on a silicon structure, sometimes combined with analog amplification.

2. Sensor element combined with analog amplification and ADC (analog to digital converter).

3. Fusion of sensor element with analog amplification, ADC and digital intelligence for linearization and temperature compensation.

4. Memory registers for calibration and temperature compensation are added.

It can be seen that MEMS sensors are shifting from simple analog types to digital types with more functionalities. Take tri-axial (a.k.a. 3-axis or 3D) accelerometers for example, there are currently three types of interfaces: analog, pulse-width modulated and digital.

An analog accelerometer has three output pins, each producing a voltage that is directly proportional to the sensed acceleration at the corresponding axis. A PWM (pulse-width modulation) accelerometer, which is less common, produce square waves with a fixed frequencies, with duty cycles varying with the sensed acceleration. Both types require dedicated modules, be it an ADC or a pulse-width modulator, to continuously convert output signals from accelerometers into digital representations (typically in two-byte forms) which are understood by microcontroller units (MCU).

A digital accelerometer features a serial interface, typically SPI (Serial Peripheral Interface) or I^2C (Inter-Integrated Circuit). Digital accelerometers typically have a FIFO (first-in/first-out) memory block for buffering data and various interrupt features such as data ready, free fall, inactivity, activity, wake-up, single tap, double tap, orientation, etc. They are also less susceptible to noise than their analog counterparts. Some have more advanced capabilities such as low/high-pass filters. Some MEMS vendors took even bigger steps in incorporating advanced technologies into a single MEMS chip. Notably, InvenSense released a family of MotionTracking^{TM} devices that combine several inertial sensors with an on-chip low-power Digital Motion Processor^{TM} (DMP) [18]. This DMP could carry out simple processing and calculations so as to offload the host MCU thus reducing overall power consumption.

2.2.2 Wireless Sensor Networks (WSN) & ZigBee

In year 2003, IEEE finalized 802.15.4 standard which is optimized for low bit rate, low duty cycle applications. 802.15.4 specifies PHY (physical layer) and MAC (medium access control) layers for low-rate Wireless Personal Area Networks (WPAN). WSN is further backed up by advents of real-time operating systems such as TinyOS [19] and Contiki [20]. Leading chip manufacturers such as Texas Instruments, FreeScale, Silicon Labs announced dozens of low-power RF chips and performances are getting better by the day.

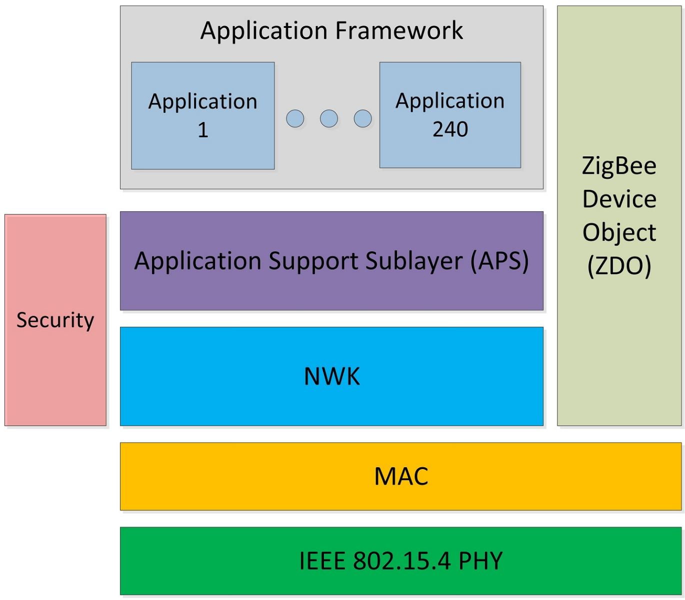

Figure 2.1: ZigBee's Architecture

ZigBee quickly adopted IEEE 802.15.4 and built NWK (network layer) as well as other higher layers on top of it, as shown in ZigBee's architecture in Figure 2.1. ZigBee is a low-cost, low-power, wireless mesh network standard. The ultimate advantage of ZigBee is its wireless mesh networking capability which sets itself apart from other wireless technologies in the same frequency band such as Bluetooth and Wi-Fi. Wireless mesh networks are multi-hop, self-organizing and self-healing. That is, if an existing route is broken, a route discovery will be initiated automatically and a new route will be established. Thus, ZigBee is robust against environment uncertainties.

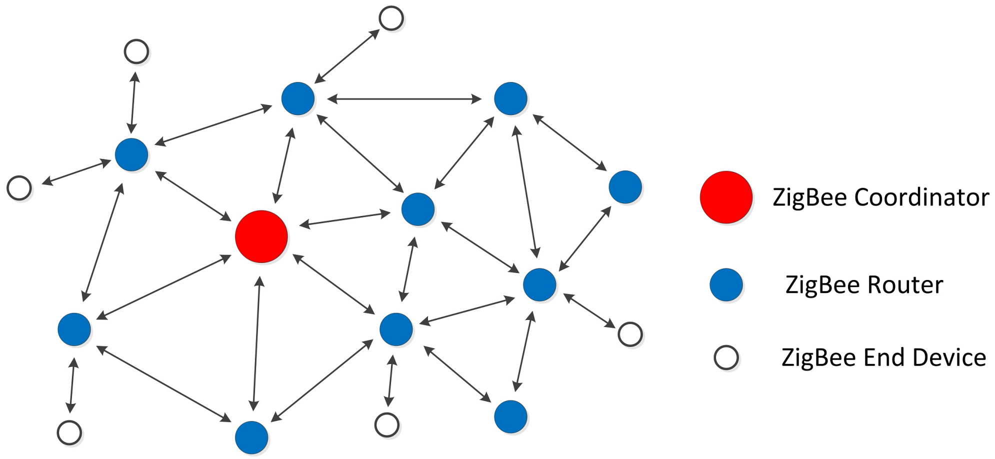

Figure 2.2: ZigBee Mesh Network Topology

Figure 2.2 shows a mesh network topology of ZigBee. ZigBee have three types of devices: ZigBee Coordinator (ZC), ZigBee Router (ZR), ZigBee End Device (ZED). ZC initiates network formation and acts as ZR once network is formed. ZR can associate with ZC or previously associated ZR. ZR participates in multi-hop routing of messages. ZED contains just enough functionality to communicate with its parent ZR. It does not relay data for other devices. This configuration allows ZED to be asleep a significant amount of time thereby giving long battery life.

2.2.3 Cellular Network

Cellular networks are pervasive due to wide adoptions of mobile phones. From 1990 to 2011, worldwide mobile phone subscriptions grew from 12.4 million to over 5.6 billion, penetrating the developing economies and reaching the bottom of the economic pyramid. Emerging economies in fact rely much more heavily on the cellular networks instead of cable or landline infrastructures to deliver voice, data, and Internet services, as shown in Figure 2.3. Laying wire across these countries' large and rugged geographies, particularly in remote areas, is expensive and time consuming.

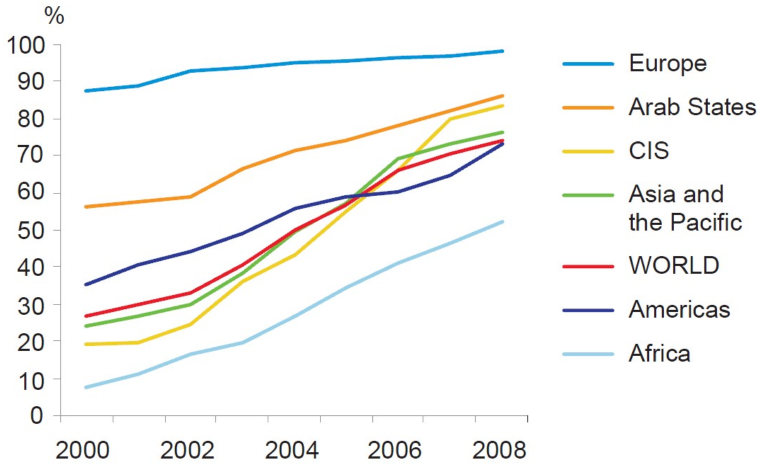

Figure 2.3: Rural Population Covered by Mobile Signals, 2000-2008, by Region [2]

Cellular network is by far the best medium to accommodate SMS (Short Message Services), voice and packet data all within a single ecosystem. GPRS (General Packet Radio Service) is one of old yet still popular technologies for delivering packet data via cellular networks. Though some latest technologies such as UMTS (Universal Mobile Telecommunications System) will eventually supersede GPRS, they are not deployed as widely as GPRS currently is. As most mobile operators still keep the backward compatibility with GPRS when they upgrade their networks, GPRS remain the best choice for Internet access via cellular networks if pervasiveness is important and high bandwidth is not required.

Some monitoring devices have been built right on top of a GSM/GPRS module. Urgentys [21] from Medical Intelligence is a good example. It can be worn as a watch or as a belt clip and was designed for lone workers at dangerous environments. It is GPS-enabled and has a standby hour of 10-100 hours which is similar to a typical mobile phone.

2.2.4 Web Servers: Ajax & WebSocket

The primary function of a web server is to deliver web contents in response to requests from clients using the Hypertext Transfer Protocol (HTTP). Traditional web contents are static, which are delivered to users in exactly the same forms as stored. Static contents are typically HTML (HyperText Markup Language) documents. Nowadays, it is dynamic web pages that make Internet interesting and interactive. Dynamic web pages are generated at the time of request from clients or change as a result of interaction with clients. Together with client-side technologies Ajax (Asynchronous JavaScript and XML), web applications can send data to and retrieve data from a web server without interfering with behaviors of the existing page, providing user experiences similar to that with a desktop GUI (graphical user interface) client. One of the most prominent web applications created with Ajax is GMail by Google.

Traditionally, web servers only response upon client requests but not the other way around. That means web servers could not push data to clients without clients explicitly requesting it. A non-standard workaround is to create a long-held HTTP request to achieve a similar effect. This problem is resolved in HTML5 specification which standardizes a web technology called WebSocket. WebSocket is created to provide for bi-directional, full-duplex persistent communications over a single TCP (Transmission Control Protocol) connection. WebSocket makes it possible for servers to send content to browsers without being solicited by clients, and allowing for messages to be passed back and forth while keeping the connection open.

WebSocket achieves communication over the regular TCP port number 80, which is rarely blocked by a firewall. Currently, WebSocket is supported in several browsers including Safari, Firefox and Google Chrome. WebSocket enables seamless two-way communications with a remote system from web browsers, without requiring users to download anything.

2.3 Related Research

2.3.1 Fall Detection & Activities of Daily Living

In recent years, MEMS inertial sensors have sparked an intense interest in studying fall detection and classifications of activities of daily living (ADLs).

[22] and [23] separately implemented fall detection algorithms right within Android phones, taking advantage of their built-in accelerometers and gyroscopes as well as high-level APIs (application programming interface). [24–31] used a single tri-axial

accelerometer attached to certain part of torso (waist, chest, back or pelvic). Some used more than one accelerometer. For example, [32] studied accelerometers on waist, wrist and head simultaneously. Some others [33, 34] used gyroscopes in addition to accelerometers. [33] used a tri-axial accelerometer on waist and a bi-axial gyroscope on thigh. [34] studied a tri-axial accelerometer and a tri-axial gyroscope on both chest and thigh.

[35–39] used accelerometers to study ADLs. In this area, Wockets [40] is a notable open source project that aims to build hardware and software that permits automatic, 24/7 physical activity and context detection on mobile phones. Wockets are miniature and low-cost devices that collect human motion using accelerometers, and send raw data via Bluetooth to users' mobile phones. It is designed in a way that several Wockets attached to different positions of a human body so as to increase robustness and accuracy and detect a wide range of activities. This design creates a star network and is in fact a WBAN (Wireless Body Area Network). A single BCU (body central unit), a mobile phone worn by human, analyzes raw data from all Wockets simultaneously in real-time. Using mobile phone has an advantage that wearers are not restricted to certain areas but can be anywhere with cellular network coverage. Bluetooth has high bandwidth for continuous data streaming but is very power consuming for both Wockets and mobile phones. As a result, Wockets are only designed to meet a goal of 24-hour performance on one charge.

2.3.2 Remote Physiological Monitoring

Sensor miniaturization allows a large number of sensors to be incorporated into a single device. Naima and Canny developed a wireless health monitoring platform named the Berkeley Tricorder [41]. Tricorder is a small portable device which supports multiple measurements including Electrocardiography (ECG), Electromyography (EMG), respiration, acceleration, blood oxygenation, galvanic skin response and skin temperature. Sensor data are either streamed to mobile phones/computers via Bluetooth for real-time analysis, or stored in a microSD card.

2.3.3 WSN in Medical & Health Areas

Extensive researches have been carried out to employ WSN in medical and health areas. One of the early works is CodeBlue [42] developed by Malan et al., which is a wireless infrastructure intended for emergency medical care at clinics, hospitals and large casualty sites. CodeBlue provides routing, naming, discovery, and security for wireless medical sensors, PDAs and PCs. Wearable nodes consist of location sensors for both indoor and outdoor uses, a pulse oximeter, a blood pressure sensor and an electronic triage tag.

Since the emergence of WSN, numerous home health monitoring systems have been proposed. [43] uses WSN to study ADLs. [35] and [44] use WSN and accelerometers to detect falls. [45] uses WSN, accelerometers and EMG sensors to study standing balance. [46] uses WSN in several projects including infant monitoring, heart-related monitoring. [47] uses WSN in ECG monitoring. [48] uses WSN to monitor temperature, luminosity, ECG and oxygen saturation (SaO2). [49] uses WSN with sensors that measure pressure, galvanic skin response, flex and temperature.

2.4 Design Choices

2.4.1 Choice of Wireless Technology & Network Topology

An ideal solution will be a single wearable device that monitors the elderly and communicates to caregivers directly without going through an intermediate system such as a base unit. A wireless module like GSM/GPRS or alternatives will be a good choice. An advantage of this approach is true ubiquitousness achieved. Seniors are no longer restricted to a certain residential area. However, such design will always suffer from limited standby hours, just as mobile phones do. As cellular technologies are much more complicated than technologies used in WPAN, GSM/GPRS modules are also much bulkier and more expensive than Bluetooth or ZigBee transceivers. Using GSM/GPRS modules directly in a wearable device will not be a good design choice for long-term monitoring based on current state-of-the-art battery technologies.

Most PERS devices employ a point-to-point wireless communication between a base unit and a wearable unit. Lifeline with AutoAlert operates at the 312 MHz frequency band while myHalo works at 2.4 GHz frequency band. All other factors (antenna gain, transmit power) being equal, as the frequency of electromagnetic waves increases, signal will experience more power loss leading to smaller range of coverage. This inverse relationship between transmission range and frequency is expressed in the free space path loss equation.

$$FSPL(dB) = 20log_{10}\left(\frac{4\pi df}{c}\right)$$

where f is the signal frequency, d is the distance from the transmitter, and c is the speed of light.

Frequencies also impose restrictions on antenna sizes. A rule of thumb in antenna design for optimal reception is that the height of an antenna should be at least 1/4 of the wavelength. Thus, higher frequency will result in smaller antennas. Antenna size is important in the design of a wearable device. 2.4 GHz technologies such as 802.15.4 have a wavelength of about 12.5cm, thus requiring an antenna of about 3.1cm. In this regard, 802.15.4 is well-suited for small wearable devices.

Due to the actual physics behind radio frequencies, if small antennas are desirable, point-to-point transmission will be limited to a certain length. Increasing transmit power and improving receiver sensitivity will help initially but beyond a certain point they will become impractical.

To cover larger areas, multi-hop wireless technologies have to be resorted to eventually. ZigBee is one such technology. ZigBee gains range but loses data rate. The maximum data rate of ZigBee is 250 kbps but by the time retries, acknowledgement, etc, the actual through-put is closer to 25 kbps [50]. Its low rate and multi-hop nature basically rule out any applications that intend to do real-time streaming of massive data such as voice or raw sensor data sampled at high frequencies.

2.4.2 Suitability of Sensors

Depending on actual applications, choices of sensors vastly differ. For short-term monitoring applications such as CodeBlue, wearable units could incorporate many heavy-duty sensors to form a multi-functional wearable device. In contrast, for long-term monitoring, care must be taken when selecting sensors and algorithms as they affect standby hours greatly. In addition, from an ergonomics point of view, certain sensors, such as ECG and SaO2 sensors which are typical in remote physiological monitoring devices, cannot be easily designed to be worn 24/7 in a comfortable and unobtrusive way due to their nature of operation.

2.4.3 Centralized vs. Decentralized Computing

Sensors such as MEMS sensors, usually small and low-power, are well-suited for wearable devices. However, they can quickly generate huge amount of data. In studies of falls and ADLs using accelerometers, a sampling rate ranging from 50 Hz to 250 Hz is typically required [27], in order not to miss critical high frequency components (e.g. a sudden fall) of acceleration data. At a sampling rate of 50 Hz, even if each axis is represented by a single byte, there will be 150 bytes of information per second for a tri-axial accelerometer. Analyzing each piece of these data is no easy task and will demand a considerable amount of computational power.

In studies of falls or ADLs evaluated above, except for a few which chose to store data on an SD card [30] for offline processing at a later time, all others took a centralized computing approach where raw sensor data is sampled and wirelessly streamed to a remote base unit via either Bluetooth [28, 29, 51, 52] or ZigBee [24, 31, 33] transceivers. Note that even though [24, 31, 33] used ZigBee transceivers, they still establish a point-to-point connection like Bluetooth.

Processing raw sensor data in a remote processing unit is acceptable and usually the most convenient for research studies. However, it is difficult to make practical use of such algorithms while achieving low power and real-time performance in a multi-hop wireless network. In a WSN remote monitoring application, a practical design is only left with one of the three approaches below:

1. Centralized: Stream raw accelerometer data wirelessly to a base unit which has more processing power or,

2. Decentralized: Implement an algorithm completely within a wearable unit which is typically resource and power constrained or,

3. Heterogeneous: Pre-process raw accelerometer data (e.g. feature selection, dimensionality reduction) so as to compress it into smaller size, and then transmit preprocessed data wirelessly to a base unit for further processing.

The centralized approach will eat up a lot of bandwidth and also consume tremendous power due to continuous wireless transmission. In a WSN where multiple wearable units exist and some might be several hops away from the base unit (a.k.a. a data sink or a data concentrator), the total amount of data aggregating at the base unit creates

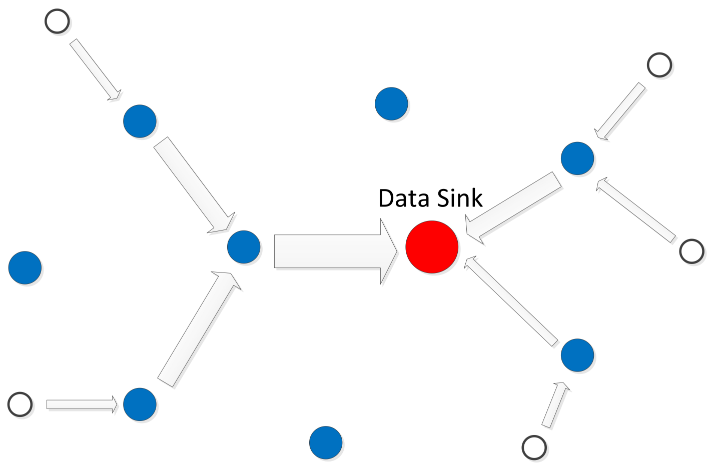

Figure 2.4: Centralized: All Sensors Stream Raw Data to a Sink

too much network traffic around it, as shown in Figure 2.4 where thickness of arrows indicates the amount of data. The decentralized approach will drain batteries quickly due to continuous processing of accelerometer data and classification algorithm. The heterogeneous approach seems promising. However, the challenge lies in the design of a pre-processing algorithm that is lightweight and capable of transforming raw data into feature sets that are much smaller in size. A possible solution might be streaming only segments of raw data that are of interest.

2.4.4 Analog vs. Digital Accelerometers

It is noticeable that in studies of falls and ADLs evaluated above, all used an analog accelerometer except for [24] and [31]. However, [24, 31] used it in a way no different than analog accelerometers. That is, no advanced features of digital accelerometers were exploited. Raw accelerometer data was simply read and blindly streamed to a remote base unit for processing.

Accelerometers are usually used in mobile devices, where there is a strong need to relieve the burden on host MCUs. Digital accelerometers are gaining increasing popularity and most MEMS vendors come out with digital accelerometers. Depending on MEMS vendors, each digital accelerometer offers slightly different sets of advanced features. Nevertheless, most digital accelerometers support FIFO and interrupt features like activity, inactivity, free fall. Accelerometers of this type include ADXL345 from Analog Devices, MMA8450Q from FreeScale, LIS3DH from ST Microelectronics. Creative use of these features could save a lot of computational resources. For instance, acceleration data can be queued in FIFO instead of being read at each sampling instant. When FIFO reaches a predetermined amount, it will generate an interrupt so the host MCU will retrieve all data at once. In this way, the host MCU could read less frequently. If the accelerometer is stationary, inactivity interrupt will be asserted and the host MCUs could go to sleep without examining any accelerometer data.

However, taking advantage of such advanced features comes at a price which is portability or generalizability. If a particular algorithm exploits a particular feature of an accelerometer that is not found elsewhere, it could not be ported to other accelerometers without tweaking algorithm. Besides, if an upgraded version of an accelerometer has certain features that could be exploited or a certain feature is discontinued, changes to be made on the original algorithm may be non-trivial. In contrast, analog accelerometers are free from such concerns.

Despite possible difficulties of portability of algorithms designed for digital accelerometers, a digital accelerometer will outperform its analog counterpart in a myriad of aspects. In this thesis, digital accelerometers will be used to implement a fall detection algorithm and an ADL algorithm, as described in Chapter 5.

2.4.5 Desktop vs. Web Applications for Remote Access

With HTML5, Ajax and all server technologies available, it is possible to develop a web client which replaces functionalities of a desktop client. For instance, Google Docs, a free web-based office suite developed by Google, is replacing most basic functions of Microsoft Office. Either approach has its own merits and demerits. Web applications do not require installation and updates, can be accessed from anywhere via Internet and are platform-independent. In comparison, desktop applications lack these advantages but can do more calculations, are usually more responsive and often more secure. In addition, a desktop application can be run as a daemon process in the background thus it can be always on.

2.4.6 Duty Cycling

One approach to reduce energy consumption is to duty cycle. Certain sensors do not require continuous operation as measured parameters are unlikely to change quickly in the near future, such as a body temperature sensor and heart rate sensor. These sensors are suitable for duty cycling, i.e. to turn them off periodically and turn on just enough time for one reading.

Chapter 3

System Architecture Design

3.1 Device Types

An *e-Guardian* system consists of three types of devices: Base Station (BS), Range Extender (RE) and Wearable Device (WD). Inter-device communications are supported by ZigBee network. In ZigBee's terminology, BS, RE and WD correspond to Coordinator, Router and End Device respectively. BS starts a ZigBee network and many REs extend this network to create a multi-hop wireless mesh network. In this way, WDs could move around in a much larger region while still connected with BS. Communications with the outside world via SMS, phone call and Internet are made possible with a GSM/GPRS module integrated into BS.

3.2 Overall Architecture

Design of the system architecture has taken discussions in Section 2.4 into account. Figure 3.1 shows a complete picture of *e-Guardian*'s architecture. Essentially, *e-Guardian* embodies a wireless infrastructure which provides a seamless and bi-directional wireless link between seniors wearing WDs and their caregivers. This infrastructure is an open-platform on top of which actual implementations will continuously evolve. For instance, the GSM/GPRS module in BS could be replaced by a more advanced UMTS module when UMTS networks are more widely deployed than GPRS networks. Various features for BS, RE and WD can be integrated depending on actual application scenarios. A lot of features are proposed below and some representative ones have been implemented.

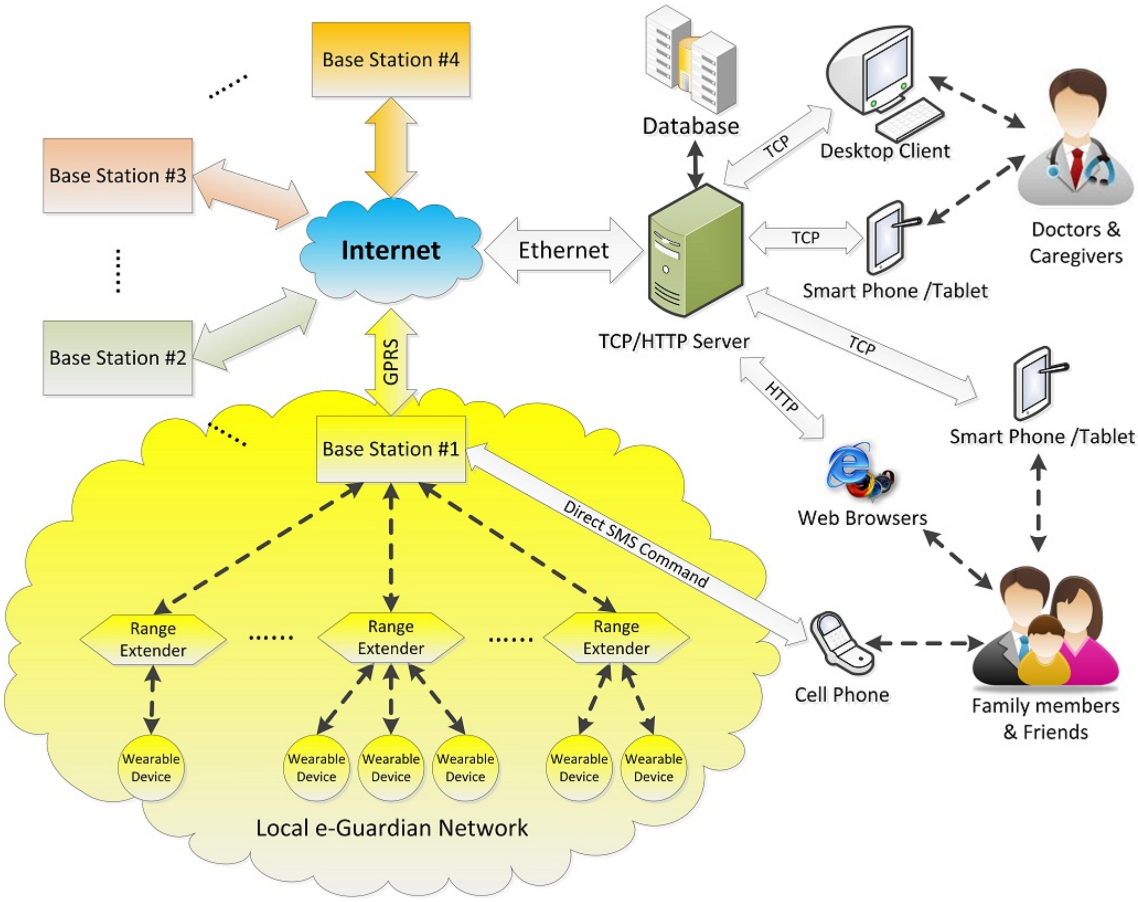

Figure 3.1: System Architecture of *e-Guardian*

This architecture provides two alternative channels, SMS and Internet, for commanding and accessing information from the local *e-Guardian* network. Either one can be used depending on application scenarios or users' preferences.

3.3 Application Scenarios

e-Guardian is designed to be scalable to suit different application scenarios, which generally fall into two categories: small scale and large scale. Small scale typically refers to a household deployment which monitors one or two seniors of a family. *e-Guardian* devices for a household deployment is collectively called a home kit. Large scale refers to deployment at care centers, rehabilitation centers, hospitals, residential areas, rural villages, etc.

In the following sections, for the convenience of reference, home setup is used to refer to small scale scenarios, while community setup is used to refer to large scale scenarios.

3.3.1 Home Setup

In the simplest case, the system can be deployed in a single household. One could simply order a home kit which consists of one BS, several REs and WDs for his old parents. He could configure and instruct the system purely via SMS. He could add other family members, caregivers, doctors, relatives, friends as he wishes so those people could also receive alerts and statistics. They could live with it without concerning about GPRS settings, TCP/HTTP servers, etc. However, a home kit can always connect to a generic TCP/HTTP server provided by *e-Guardian*'s developer. Figure 3.2 shows a typical home setup.

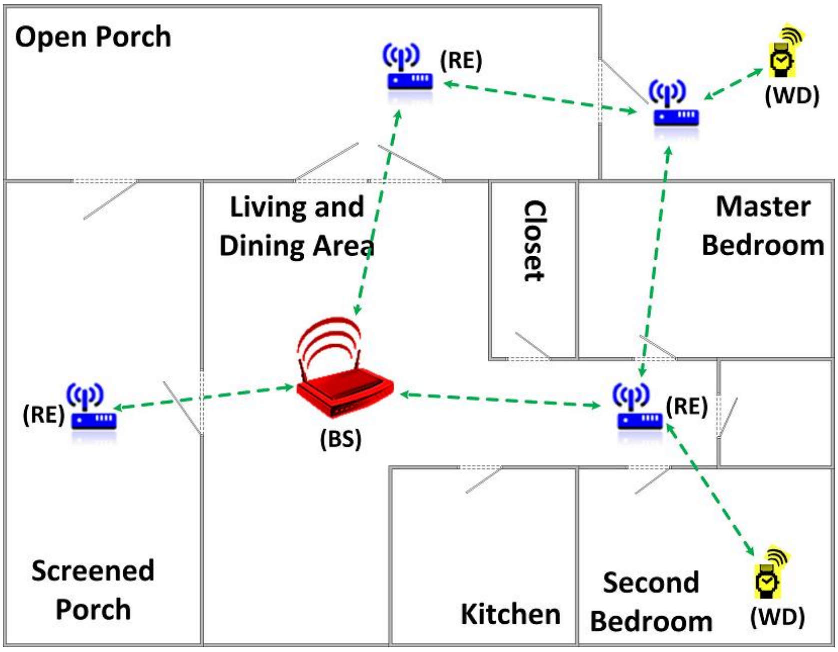

Figure 3.2: *e-Guardian* in a Home Setup

3.3.2 Community Setup

e-Guardian could be deployed in a care center, a rehabilitation center, a hospital, a residential area or an entire village, using REs to reach out to individual households or rooms. The BS will be kept at the center of the deployed area. As the GSM/GPRS module is the most expensive component of BS, sharing a BS (together with SMS and data subscription) can reduce cost per capita significantly.

In a community setup, *e-Guardian* system will be in a full-blown shape, as shown in Figure 3.2. The optional TCP/HTTP server is useful and almost always necessary for a community setup, where an administrator has to manage hundreds of WDs. Sending instructions from a desktop client or a web browser will be more convenient. Nevertheless, the primitive SMS interface will still be occasionally useful when an administrator temporarily loses access to Internet. The *e-Guardian*'s TCP interface will be publicly available so system integrators could easily incorporate *e-Guardian* into existing solutions.

3.4 Local *e-Guardian* Network

e-Guardian is scalable, which means the same *e-Guardian* devices could be deployed in either a small household or a large residential area. This is enabled by using a WSN. The realization of this WSN is not limited to ZigBee which the implementation of *e-Guardian* is based on. Other possible WSN choices include WirelessHART and custom implementations using WSN RTOS (real-time operating systems) such as TinyOS and Contiki.

In *e-Guardian*, BS and RE are mains-powered and their locations are usually fixed. They can be occasionally moved around or replaced by new units however they are not expected to be constantly moving. WDs are battery-powered and truly mobile devices. They could move freely in the same network. When moving, they will dynamically attach to different REs (including BS).

3.4.1 Base Station (BS)

3.4.1.1 Design of BS

BS is the root and heart of a local *e-Guardian* network. It is responsible for forming a local *e-Guardian* network. BS is also the trust center and repository for security keys for data encryption.

An isolated WSN is not particularly interesting unless its information can be remotely accessed. A gateway that translates information between the local network and another universally available network is required. A GSM/GPRS module is suitable for this purpose. BS, which has an integrated GSM/GPRS module, acts as a gateway between the local *e-Guardian* network and cellular networks.

However, BS does more than just relay information back and forth between the local *e-Guardian* network and a cellular network. There is also a built-in multiplexing mechanism for interpreting queries send via SMS or Internet. It verifies a sender's identity against a local database which tracks system administrative information and configurations, and then passes the command down to corresponding low level system services.

3.4.1.2 BS in a Home Setup

In a home setup, a number of particularly useful features are proposed for BS.

3.4.1.2.1 *Email Notifications*

As SMS is normally charged at much higher prices than GPRS data carrying the same amount of information. For a home setup, it is economical to add an email notification feature so that family members could optionally receive non-urgent information such as statistics via email instead of SMS. Most GSM/GPRS modules support email protocols so this could be implemented easily.

3.4.1.2.2 *Context-Awareness*

The relative position of a WD in the local *e-Guardian* network, together with time, local weather and news, represents abundant contextual information that can be used to offer intelligent and pertinent recommendations to seniors.

A WD can be programmed to broadcast a presence message periodically to all its neighboring REs (including BS) in the hearing range. In this way, when a senior wearing a WD walks near the BS centered at home, the BS would be aware of this proximity by constantly evaluating RSSI (received signal strength indicator) values of presence broadcasts within its hearing range. Assuming that the system is connected to a TCP/HTTP server offered by *e-Guardian*'s developer, local information such as weather forecast, important local news, etc, could be pushed from the server into the BS. In addition, most cellular networks provide local time information through a standard service called NITZ (Network Identity and Time Zone). Thus, a BS is capable of precise time-keeping. With knowledge of such contextual information, a BS could play a number of pre-recorded audio messages via an integrated speaker when seniors are nearby at appropriate time. Audio messages could, for instance, be "good weather for a morning walk", "avoid outdoor activities because of imminent storms or blizzards", "time for bed as it has been late and do not forget to take medications before sleeping", etc.

3.4.1.2.3 *Voice Communication*

There is already a GSM/GPRS module in a BS. With a few additional hardware components such as a microphone and a speaker, voice communications can be implemented. Family members can call in to speak to their loved ones. Even though there might be already a landline telephone at home, the voice call functionality in a BS could be customized thus could be different in a number of aspects.

- Whenever seniors need help, they just need to press a help button in a BS instead of having to remember and dial all digits of a phone number. The BS will retrieve all caregivers and dial them one by one until one of the caregivers answers the call.

- Whenever family members call their loved ones at BS, if seniors fail to answer the phone in 10 seconds, a voice message can be recorded. Meanwhile, BS could send a command to blink the WD's LED for a few seconds. It will keep on sending blink commands to the WD periodically (e.g. every five minutes). Until seniors see notification and walk near BS, BS will then play recorded voice messages out automatically.

3.4.1.2.4 *GPS Tracker Variation*

e-Guardian only monitors seniors inside the local *e-Guardian* network. Seniors need to go out for exercises and social life. When they do leave, a GPS tracker might be needed, especially for seniors with dementia.

A solution is to design a BS with GPS functionality. There are GSM/GPRS modules with GPS functionality. In fact, the GSM/GPRS module in the *e-Guardian* prototype described in Chapter 4 is of this type. This requires compact design for small GPS and GSM antennas. This BS will be powered by either the mains or Li-ion batteries. It can be designed such that when it is powered by the mains, GPS functionality will be turned off and immediately when it is powered by batteries, the GPS functionality will be turned on.

When a senior leaves home, a leave alert will be detected, as proposed later in Section 3.4.2.2.3. He will be prompted to unplug the BS and bring it along in his pocket.

3.4.2 Range Extender (RE)

3.4.2.1 Design of RE

The primary role of an RE is to route data packets. Due to movements or introduction of new obstacles, a WD may lose connection with its original RE. In such cases, it will issue an orphan notification and get associated with a new RE. This new RE will be the new parent of this WD and will be responsible for holding all messages sent from other devices to this WD until the WD wakes up and polls for messages. It also helps forward messages from this WD to other devices in the network.

By introducing new REs, a local *e-Guardian* network can scale up to cover a larger area. Whenever there is any dead spot, an RE is all that is needed.

3.4.2.2 RE in a Home Setup

A few advanced features which require no or very little changes on a regular RE are proposed below.

3.4.2.2.1 *Danger Zone Alert*

Certain places at home such as kitchens, bath rooms and toilets are statistically more dangerous than others. These areas can be classified as danger zones. Seniors staying in those zones longer than a certain period could be a sign for abnormality. A firmware modification for RE could make an RE a special one without any hardware changes.

One such RE placed in a danger zone will actively monitor RSSI values of presence broadcasts periodically emitted from WDs. The RE will assert a WD is within the danger zone if the average value of RSSI is beyond a threshold value. If a WD spends more than a certain time there, a danger zone alert will be automatically triggered and sent to the BS.

3.4.2.2.2 *Alarm-equipped RE*

As shown in Figure 3.2, in a home setup, certain REs will be placed outside a house. REs placed outside could be a special version of RE which has an alarming siren or an emergency light to notify neighbors who have been informed of this feature beforehand.

3.4.2.2.3 *Leave Alert*

Seniors with dementia will inadvertently leave their homes. Such cases could be prevented by adding a leave alert functionality. As every WD polls for incoming messages from its parent periodically, a parent which fails to receive polls from any of its child WDs within three to four polling periods will notify the BS of this loss. On the other hand, if this WD rejoins another RE in the network, the WD will send a device announcement message, announcing its new network address. The BS could then check if an incoming report for loss of WD is matched with a recent device announcement. If this fails, a leave alert will be generated. The system could utilize an alarm-equipped RE outside of the house to draw attention to neighbors.

In addition, in such cases, the WD worn by a leaving dementia patient could also play some mild but noticeable tones using its built-in buzzer to draw attentions to passers-by.

3.4.2.2.4 *Security Sensors*

e-Guardian's motivation is to make home a safe place for seniors. Many security sensors such as smoke/gas detectors, intrusion sensors could be integrated into REs. When any danger is detected, alerts will be sent to caregivers and meanwhile alarm-equipped REs outside will go off.

3.4.2.2.5 *Automatic Night-Light*

Poor lighting is one of the prominent reasons for falls among the elderly. To prevent such scenarios, night-lights should be installed in seniors' bedrooms, hallways, near stairs and in bathrooms.

An RE could incorporate a light bulb and a light sensor. Similar to danger zone alert, such RE constantly evaluates RSSI values of presence broadcasts from WDs to know if any WD is near. When a WD walks near, an RE should turn on its light bulb if its light sensor senses that ambient light is poor.

Night-light REs could be installed in all places seniors might visit at night, except for their bedrooms for obvious reasons that seniors need to sleep at night with lights off. A night-light RE for bedrooms has to be designed differently. It should only turn on at night when a senior leaves bed. There are many ways to achieve this, either automatically or manually. An automatic method is having an RE to constantly measure pressure under a mattress or a bed leg.

3.4.2.3 RE in a Community Setup

All proposed features in Section 3.4.2.2 are also suitable for community setups, except for alarm-equipped RE, which will be used slightly differently.

3.4.2.3.1 *Location Tracking*

In a community setup, location tracking will be an indispensable feature. In a care center of hundreds of seniors, the ability to locate an emergency as quickly as possible is invaluable. RSSI-based location tracking is a simple approach for WSN. There have been various researches on RSSI-based algorithms for indoor localization using WSN [53–55]. Theoretically, RSSI values can be expressed as a function of the distance between a sending node and a receiving node. Even though in reality RSSI is affected by a lot of factors including obstacles, multipath fading, antenna polarization and cross-body shielding, on average it is still able to provide a good estimation of distances. As REs are placed at fixed and known locations, by capturing RSSI values of presence broadcasts from a WD at these REs and then aggregating them to a BS, the BS can estimate approximate distances from this WD to all its neighboring REs. By using indoor localization techniques such as triangulation or fingerprinting, it is able to estimate a location of the WD in reference to REs.

Depending on actual needs, location tracking can be real-time which provides continuous updates of locations of all WDs. Alternatively, it could be computed only when needed to save network bandwidth.

3.4.2.3.2 *Alarm-equipped RE*

In a community setup, alarm-equipped REs can be used in conjunction with location tracking. In case of an emergency, if medical personnel fail to show up for any reason, nearby REs could alarm at a sound level proportional to average RSSI values of the anomalous WD so as to give some hint to those who are searching.

3.4.2.3.3 *Activity Statistics*

Location tracking can be useful for collecting activity statistics if privacy is not a concern. Time, frequency and duration of presence in certain rooms such as toilets, fitness rooms and TV rooms will provide useful information to assess a patient's recovery progress.

3.4.3 Wearable Device (WD)

3.4.3.1 Design of WD

e-Guardian is open to innovative designs for WD such as chest strap, wrist watch, belt, necklace, hat and shoe, depending on their nature of operation. However, two important restrictions are imposed on designs of a WD in addition to ergonomic aspects such as size, waterproofness and comfortability. Complying with these rules will make sure a WD is a good citizen of the *e-Guardian* ecosystem.

1. Long standby hour: A WD should meet a standby hour expectation of at least one month within a button cell battery.

2. No massive streaming: Due to the decentralized nature of *e-Guardian* discussed in Section 2.4.3, a WD should not stream massive data in multi-hops to BS.

At a bare minimum, a WD may only have a panic button. Non-invasive sensors such as thermometers, accelerometers, heart rate sensors can be integrated to WDs as long as their algorithms can meet the restrictions stated above and their designs meet ergonomic expectations.

3.4.3.2 Proposed Features

3.4.3.2.1 *Fall Detection and ADL Classification*

Activity information can be measured using non-invasive MEMS sensors such as accelerometers, gyroscopes and magnetometers. Activity information can be used to assess seniors' health, energy expenditure, physical activity level, sleeping quality, etc.

A fall detection algorithm and an ADL classification algorithm are proposed and will be elaborated in Chapter 5.

3.4.3.2.2 *Heart Rate Monitor*

For elderly patients with cardiovascular disease, a heart rate monitor will be invaluable. Heart rate monitors can be chest-worn or wrist-worn. The exact design is not discussed here. Heart rate monitoring is another application where low duty cycle applies. Heart rates need not be taken continuously but once several minutes will suffice.

3.5 TCP/HTTP Server

As shown in Figure 3.2, a TCP/HTTP server can be anywhere between none and a full-blown one. A home kit will mostly likely connect to the generic TCP/HTTP server provided by *e-Guardian*'s developer, while a community setup can either use the generic one or implement their own TCP/HTTP servers.

3.5.1 TCP Server

As the shell command parser described in Section 4.2.4 provides an alternative interface for Internet connections via a TCP socket. A TCP server is required to serve TCP connections from *e-Guardian*. A single TCP server could simultaneously serve several *e-Guardian* systems. Each system is identified by a unique IMEI (International Mobile Equipment Identity) number of the GSM/GPRS module in its BS.

3.5.2 HTTP Server

As discussed in Section 2.4.5, a web application is more natural for family members to access an *e-Guardian* system than a desktop client which requires lengthy installations. A smooth interaction with *e-Guardian* via a web browser means a user can send commands at any time and receive messages at any time. Responses from *e-Guardian* are asynchronous in nature. Commands such as "querystatus" described in Section 4.2.4 typically have several seconds of response delays. Spontaneous alerts must also be pushed to users' browsers if they happen to occur during browsing. WebSocket is currently the best solution for such asynchronous communications in web applications. Using WebSocket, a user could establish a TCP connection within a web browser to the TCP server described in Section 3.5.1.

3.5.3 Desktop Client

Although web applications are convenient, a desktop client running on a dedicated computer could be faster and provide more computationally extensive analysis. In addition, a desktop client can run in the background and notify users of alerts using pop-up windows. A desktop client is suitable for medical personnel for their extensive daily uses.

Chapter 4

Implementation & System Integration

4.1 Hardware & Software

4.1.1 GSM/GPRS Module

The selected GSM/GPRS module is GM862 [56] from Telit. GM862 has supports for SMS, voice call and GPRS. It has an embedded TCP/IP stack which supports TCP, UDP, SMTP (for Email) and FTP (for file transfers) protocols.

GM862 has peripherals of 13 GPIO (General Purpose Input/Output) pins, 1 ADC (Analog-to-Digital Converter), 1 UART (Universal Asynchronous Receiver/Transmitter). GM862 has a built-in Python (version 1.5.2) interpreter which could host 1.9 MB of Python programs and has 1.2 MB RAM space for Python run-time. A comprehensive set of APIs for Python is available for controlling peripherals, directly calling low-level functions, and also sending AT commands to the module.

However, GM862 lacks support for interrupt and concurrency. Thus, customer programs have to feature a super-loop architecture.

4.1.2 RF Transceiver

The chosen RF Transceiver is CC2430 [57] from Texas Instruments (TI). CC2430 is an SOC (system-on-chip) solution for IEEE 802.15.4 and ZigBee applications. It combines a CC2420 RF transceiver with an enhanced 8051 MCU which has up to 128 KB flash memory , 8 KB RAM, 8 ADC channels, 21 GPIO pins, 5 DMA (Direct Memory Access) channels, two USART (Universal Synchronous/Asynchronous Receiver/Transmitter) channels. Each USART can be configured as either a UART or an SPI (Serial Peripheral Interface) bus.

Being a transceiver for low power RF, CC2430 offers four power modes, described in Table 4.1. By turning off high frequency and crystal oscillators, and turning off power supply to its digital core, ultra-low power consumption can be achieved.

Table 4.1: Power Modes of CC2430

<table>
<tr><td>Power Mode</td><td>Oscillators</td><td>Voltage Regulator</td><td>Go to PM0 Upon</td><td>Current Consumption</td></tr>
<tr><td rowspan="3">$PM0$</td><td rowspan="3">All on</td><td rowspan="3">On</td><td rowspan="3">Nil</td><td>MCU Active ($7mA$)</td></tr>
<tr><td>Rx ($27mA$)</td></tr>
<tr><td>Tx @ $0dBm$ ($24.7mA$)</td></tr>
<tr><td>$PM1$</td><td>32.768 kHz on</td><td>On</td><td>reset,
external interrupt, or
sleep timer expiration</td><td>$190\mu A$</td></tr>
<tr><td>$PM2$</td><td>32.768 kHz on</td><td>Off</td><td>reset,
external interrupt, or
sleep timer expiration</td><td>$0.9\mu A$</td></tr>
<tr><td>$PM3$</td><td>All on</td><td>Off</td><td>reset,
or external interrupt</td><td>$0.6\mu A$</td></tr>
</table>

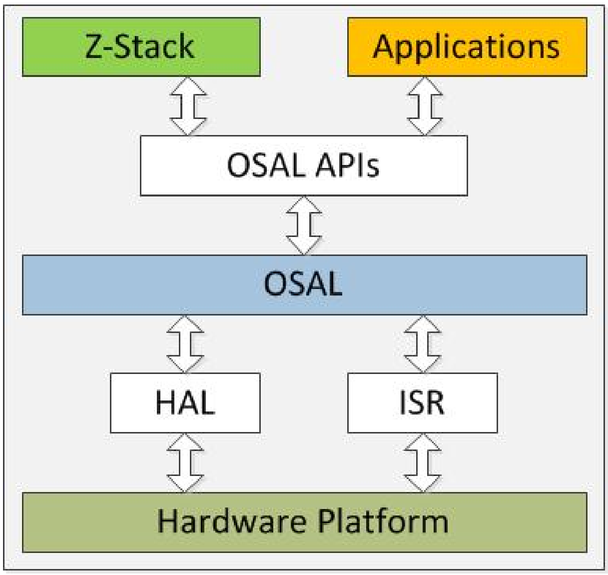

Figure 4.1: OSAL and HAL in TI Z-Stack

4.1.3 ZigBee Software Stack

The prototype uses a software stack, Z-Stack from TI, for ZigBee radio. As shown in Figure 4.1, it is built on a simple round-robin scheduler called OSAL (Operating System Abstraction Layer) which provides timers, memory allocation, inter-task communication, interrupt handling, and power management. At the low level, OSAL interfaces with ISR (Interrupt Service Routines) and HAL (Hardware Abstraction Layer). HAL hosts device drivers for peripherals so application programs could access peripherals via high level API calls.

4.1.4 RF Antenna & Microstrip Balun

To reduce the cost of components, the prototype devices use PCB (printed circuit board) antennas. There are other types of antennas available for 2.4GHz radios such as whip antennas and chip antennas. A PCB antenna does not use any components except requiring slightly more board area. An inverted-F antenna has been used in the design of all *e-Guardian* devices, as shown in Figure 4.2. The inverted-F antenna has a dimension of $3.58cm \times 0.68cm$.

The inverted-F antenna is a single-ended unbalanced antenna. CC2430 has a differential RF port which is balanced. Thus, a balun is needed to transform a balanced signal into an unbalanced signal. A balun will combine the two outputs of the differential RF pins in CC2430 into a single ended 50Ω RF signal in transmit mode, and split the singled ended 50Ω antenna signal into a differential RF signal in receive mode. The balun implemented is a microstrip balun, which does not incur any component cost other than PCB area. The microstrip balun is also shown in Figure 4.2.

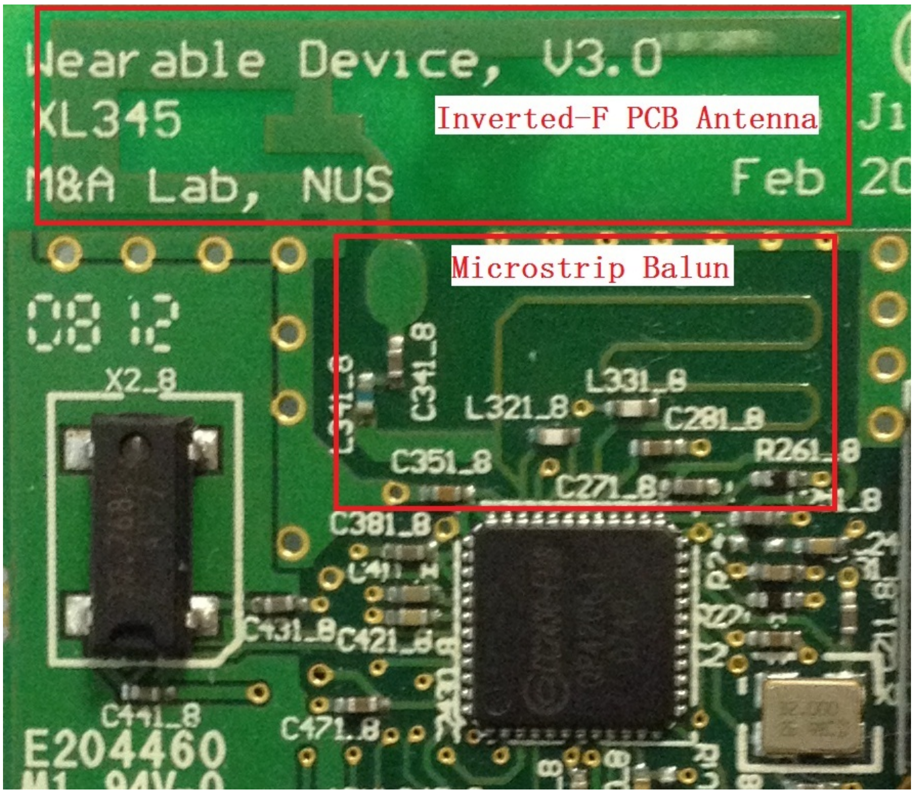

Figure 4.2: PCB Antenna and Microstrip Balun

4.2 Base Station

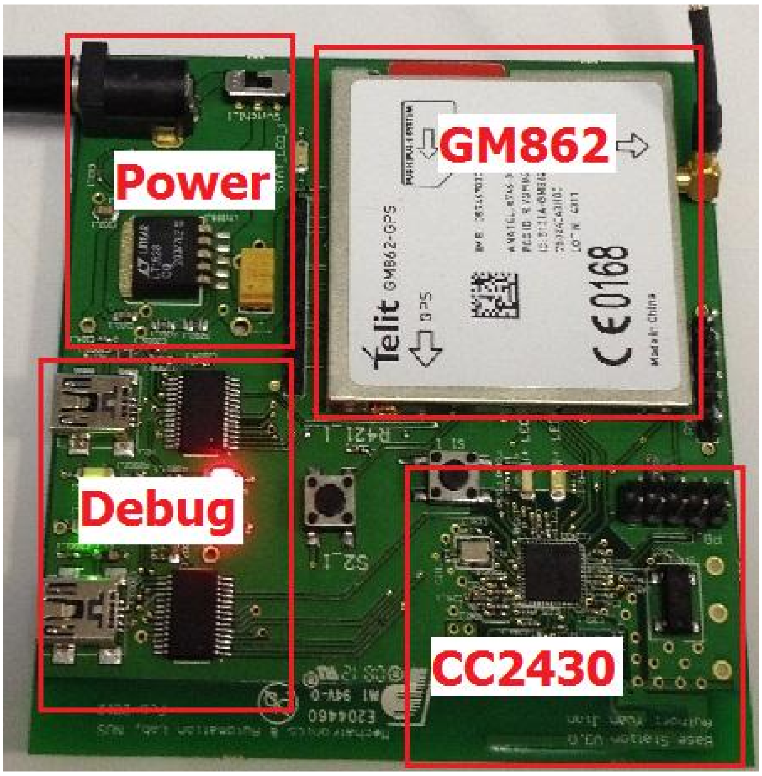

Figure 4.3: Prototype: Base Station

A prototype of BS has been implemented, as shown in Figure 4.3. The dimension of the board is $8.4cm \times 8.8cm$. The whole board area can be divided into four sections: GM862, CC2430, power supply and debug. The debug section is for debugging purposes and could be removed in the mass productions. The power and CC2430 parts could be moved to the bottom layer so the overall board area is minimized. A smaller BS allows itself to be portable and carried when seniors leave, as proposed in Section 3.4.1.2.4.

In this BS prototype, a CC2430 and a GM862 are connected via UART. The CC2430 is programmed to be a ZigBee Coordinator. BS is typically powered by the mains supply so power consumption is not a concern.

The architecture of the BS prototype is shown in Figure 4.4. This architecture is reflected in the Python program implemented for GM862. The implementation features an OOP (Object Oriented Programming) style so each module in the GM862 box in Figure 4.4 has a corresponding Python class.

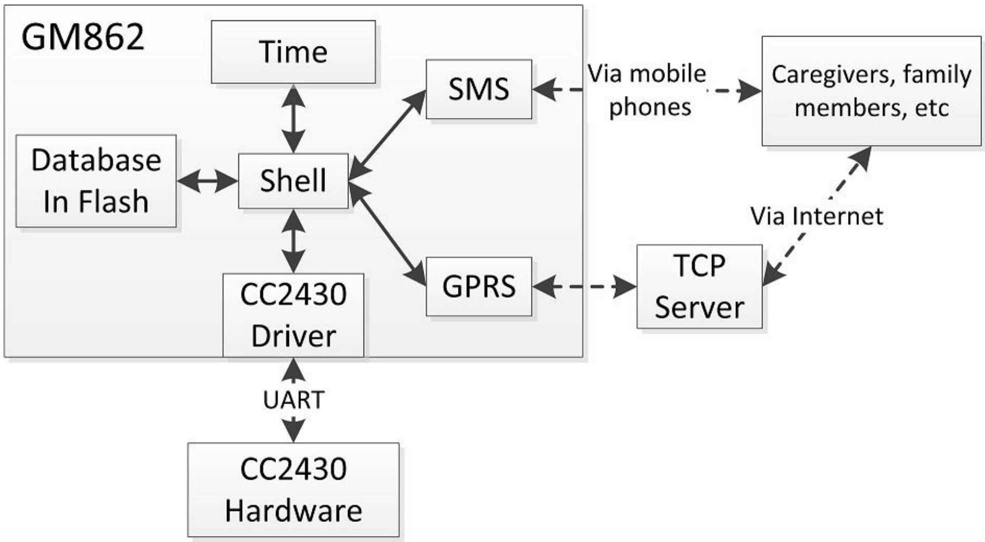

Figure 4.4: Architecture of Base Station

4.2.1 Communication with WDs

The implementation is based on ZigBee Stack Profile 0x01 which employs a tree addressing scheme called Cskip to assign network addresses to ZigBee Routers (e.g. REs) and End Devices (e.g. WDs). Each device in ZigBee has a 64-bit IEEE address which is unique in the world and also a 16-bit network address. As shown in Figure 4.5, Cskip address assignment is determined by three parameters: the depth of the tree (D_m), the maximum number of children routers (R_m) and the maximum number of all children (C_m). In *e-Guardian*, a setting of $D_m = 5$, $R_m = 6$ and $C_m = 20$ is used. This means a

RE can be a parent of 6 REs and 14 WDs at maximum.

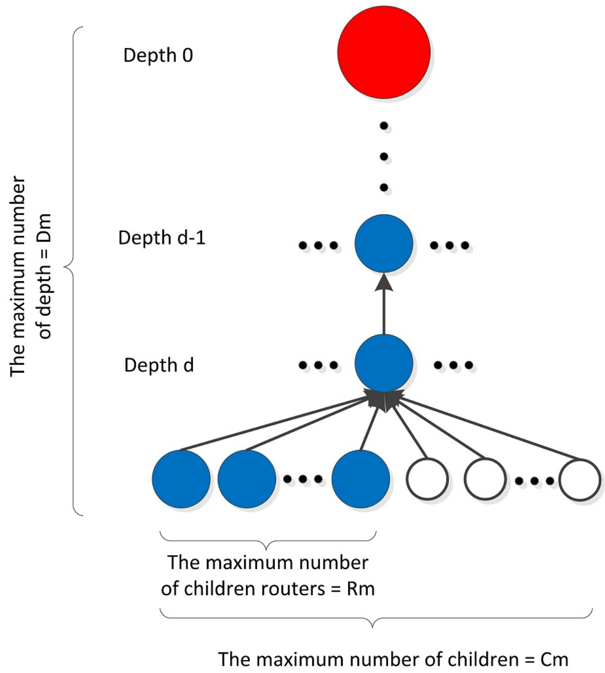

Figure 4.5: Cskip Tree Addressing Scheme

Routers at depth $d+1$ are separated by a Cskip value defined in Equation 4.1.

$$Cskip(d) = \frac{1 + C_m - R_m - C_m \cdot R_m^{L_m - d - 1}}{1 - R_m} \tag{4.1}$$

Address of n^{th} End Device at depth $d+1$ is defined in Equation 4.2.

$$A_n = A_{parent} + Cskip(d) \cdot R_m + n \tag{4.2}$$

According to Cskip addressing scheme, a ZigBee Coordinator (e.g. BS) always have a network address of 0x0000 and is the root of the address tree. A WD's address is determined by its parent RE, as in Equation 4.2. Its address changes whenever it switches to a new parent RE.

When a WD sends a message to the BS, it simply sends the message with a destination address of 0x0000. Its parent RE keeps a routing table, which records the next RE to forward this message to in order to send the message to the BS. The combination of a destination address and the next router is a single entry in the binding table. In an *e-Guardian* network, as long as all REs know how to route a message to the BS, no matter how WDs move in the network and dynamically attach to different REs, it is always easy for a WD to send a message to the BS. From the point of the BS, this is an inbound communication. For inbound communication, each RE just needs to keep one entry in its routing table, which is no problem for any RE.

However, the outbound communication, which means sending a message (e.g. a reminder alert) from the BS to a WD, is challenging. The BS has to remember addresses of all WDs in the network in order to be able to address them individually. A single entry in a routing table typically takes 6 bytes. In a network of 1000 WDs, this corresponds to a routing table of 6KB. CC2430 has a RAM space of 8KB only which is mostly taken by the ZigBee stack software.

A solution to this outbound communication problem is proposed and implemented. Instead of sending a message directly to a WD, the message is sent to its parent RE instead. The WD's address is embedded in the payload of the message so the RE knows which child WD it should forward to. The address of the parent RE of a given WD is a deterministic value, which can be actually computed using Equation 4.1 and Equation 4.2. However, to simplify the implementation in GM862, the parent address of a WD is announced by that WD. Every time a WD switches parent, it automatically sends out a message announcing its new address. Following this announcement message, a

WD is programmed to send another message which is a binding of the WD's address and its parent RE's address. This message will be sent to the BS and stored in the GM862 module. In this way, the BS knows the network address of a WD's parent RE. Assuming an *e-Guardian* network has an RE/WD ratio of 1/10, to address 1000 WDs, a BS only needs to keep a routing table for 100 REs, which takes only 600 bytes. This solution will significantly boost up the number of WDs a single *e-Guardian* system could accommodate.

4.2.2 CC2430 Driver

GM862 has a buffer of 4096 bytes for data sent from CC2430 via UART. As interrupts and threads are not supported in GM862, Python programs have to poll data from this buffer faster than CC2430 could send. As the number of WDs increase, the amount of data converging to BS also increases. Eventually it will hit the limit of 4096 bytes and a new hardware design for BS is needed. Nevertheless, in this prototype, this buffer size works well for several dozen WDs and is enough for illustrating the concept of *e-Guardian*.

When GM862 sends data to CC2430 via UART, it appends a byte indicating the length of data to the front of actual data. CC2430's UART interrupt will be generated and a DMA channel will be automatically armed to move the number of data indicated in the length byte. Taking advantage of DMA channels alleviates CC2430 from moving data using its processor.

4.2.3 TCP Client

BS can be optionally configured to connect to a TCP server. A TCP client functionality has been implemented. Once GPRS settings and TCP server's socket information are properly configured, it opens up a persistent TCP connection to a TCP server. It will check if the socket is alive and automatically retry if it is dead. A persistent connection allows commands from a TCP server to be sent to BS at any time and also allows BS to upload data to the TCP server.

4.2.4 Shell Command Parser

There is no user interface such as LCD, keyboard in BS's hardware. BS implements a user interface abstraction layer named shell (see Figure 4.4). Shell is a command parser and multiplexer which dissects commands and pass them down to corresponding system services. Users could interact with a shell via either SMS or Internet. To interact with a BS via Internet, users can either use a desktop client or a web browser, as described in Section 3.5. The desktop client or a web browser cannot connect to a BS directly as neither side typically has a static IP address so they cannot address each other. This issue is resolved by using a middleman which is a TCP server with a static IP address or a fixed DNS (Domain Name System) name.

Family members and caregivers could interact with a BS using a set of SMS/Internet commands which can be easily mastered within minutes. Most commonly used commands and their synopses are listed below.

1. ***format 12345***

   This command, sent by any phone number, resets a system to the factory settings and meanwhile assigns the sender's phone number as the new administrator. 12345 is the current system's password.

2. ***changepwd Hello123***

   This command, only if sent from the administrator, changes *e-Guardian*'s old password to a new password Hello123

3. ***initgprs "hicard" "" "" "www.malresearch.nus.edu.sg" "5309"***

   This command, only if sent from the administrator, configures the GPRS settings of the BS. "hicard" is the APN (Access Point Name) of the network operator which issued the SIM card used in the BS. Two blank fields are user name and password for cellular network authentication (which might be non-blank for certain mobile operators). "www.malresearch.nus.edu.sg" is the DNS name for the TCP/HTTP Server at Mechatronics and Automation Lab, National University of Singapore. "5309" is a particular port number the TCP server is listening to.

4. ***store 96368888 2726252423220001 2726252423220002***

   This command, only if sent by the administrator, assigns WDs (by referring to their 64-bit IEEE address which is unique to each ZigBee device in the world, which is expressed in hexadecimal numbers here) 2726252423220001 and 2726252423220002 to a caregiver whose phone number is 96368888.

5. ***alias 96368888=jack 2726252423220001=mary 2726252423220002=tom***

   This command (without space around equal signs), only if sent by the administrator, assigns each WD or caregiver to a convenient alias name so in future commands, phone numbers and WD's IEEE address could be optionally addressed by these alias names.

6. ***querystatus all***

   This command, if sent by the administrator, will send a querystatus command to all devices in the system. If sent by a caregiver, it only will send a querystatus command to all devices associated with this caregiver.

   querystatus jack

   This command, if sent by the administrator or by the caregiver "jack", will send querystatus commands to all devices associated with the caregiver "jack".

   querystatus tom mary

   This command, if sent by the administrator or by the caregiver who are associated with "tom" and "mary", will send querystatus commands to "tom" and "mary".

   Note: In fact, this command is intelligent enough to parse all its arguments and extract only those WDs which the sender has access rights to. The arguments can be phone numbers and WD's IEEE addresses, or aliases. As a single WD's could be associated with more than one caregiver, caregiver "A" could send a command *querystatus B* to all devices that are both associated with "A" and "B".

7. ***reminder 0930 all***

   reminder 0930 jack 987654321

   reminder 0930 tom mary

   This command works similarly as the command "querystatus", it sets a reminder alert which will be scheduled by the BS and sent to addressable WDs at 9:30am every day.

There are a dozen of other commands, mostly for an administrator, to view, configure, operate and even debug a BS. The rest of non-debugging commands include:

- ***delete***: to delete caregivers and WDs;
- ***retrieve***: to print a list of information about the system;
- ***allowjoin***: During initial installation, it could temporarily allow certain devices to join the network within a certain period. After that network join will be closed for security reasons;
- ***router***: to manage REs as well as their alias names
- ***balance***: to check the balance of the SIM card.

4.2.5 Database

A simple database has been implemented using Python's object serialization module, *marshal*, in non-volatile flash memory of GM862. It keeps records of administrative information, caregivers' phone numbers and WDs' IEEE addresses as well as their alias names, reminders for medication, GPRS settings, etc.

4.2.6 Scheduled Tasks & Alerts

A time module (see Figure 4.4) is also implemented which is accurately synchronized to cellular network operator's time. The time module supports various scheduled tasks for BS and WDs, one of which is a reminder functionality for medications or physical exercises. Caregivers can set reminder alerts for their WDs. BS will push alerts to corresponding WDs at predefined times by setting their LEDs to blink and buzzers to ring with certain patterns. In this way, WDs are freed from keeping precise timing and memorizing alerts by themselves. They only need to wake up periodically (every few seconds) to poll for messages from their parent REs. Current implementation does not support multiple reminders within one day and do not support one-time reminders. Such features will eventually be necessary and will be implemented.

4.2.7 Security & Access Control

Security and privacy are of great concern to seniors themselves and their family members. Two important security features have been implemented to keep out eavesdroppers and prevent unauthorized devices from joining the network.

First, ZigBee packets are encrypted using AES-128 (Advanced Encryption Standard), which is supported by CC2430 hardware so additional power consumption is minimal. The challenge is how to distribute a cryptographic key to REs and WDs securely. The key can be either programmed into all devices during manufacturing or sent to REs and WDs wirelessly during deployment. The former way is inflexible and insecure as keys must be handed to manufacturers. The proposed design adopts the latter approach and attempts to minimize its weakness that keys must be sent in the clear which is vulnerable to one-time eavesdropper attacks. In the proposed solution, BS is the trust center of a network. During a deployment, system administrator first reduces CC2430's radio transmit power in BS to a level that only REs and WDs within a few meters could pick up. Then, in a relatively secluded room where eavesdropping nearby is less likely to happen, REs and WDs are turned on to join the network started by the BS. The key is then transmitted and permanently stored in non-volatile flash memories of CC2430. The BS's transmit power is restored to normal thereafter.

Second, to prevent unauthorized WDs from joining a network, a method for controlled join has been implemented. When a RE/WD wants to join a network, it sends out its IEEE address to the BS. BS will determine whether to allow or reject based on a list of allowed IEEE addresses. However, CC2430 has limited memory thus cannot keep an arbitrarily long list of allowed devices, especially in large scale deployments where hundreds of WDs may exist. This list has to be stored elsewhere. The prototype stores the list in GM862 which has larger memories. A round-robin type of mechanism has been proposed to transfer a large allowed IEEE list to CC2430. During a deployment, administrators can send up to ten WDs' IEEE addresses each time to GM862 via shell commands. These addresses will be then sent via UART to CC2430 ZigBee coordinator which reserves ten address slots beforehand. The corresponding ten WDs will be turned on right after that. When all ten WDs successfully join the network, another group of IEEE addresses can be sent to overwrite previous addresses. The process goes on till all devices join the network. Network information will be kept permanently in CC2430s' flash memories so controlled join only needs to be performed once throughout a device's life time.

4.3 Range Extender

An RE prototype has a CC2430 chip configured as a ZigBee Router. There is neither any sensor nor any user interface but only a LED as a status indicator.

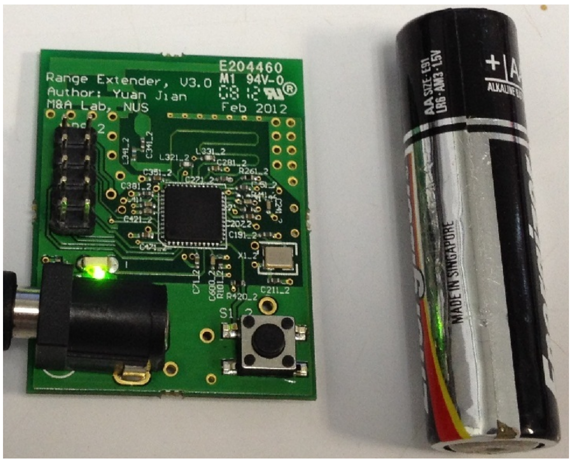

Figure 4.6: Prototype: Range Extender

4.3.1 Location Tracking & Leave Alert

A trimmed-down version of location tracking is implemented together with leave alert. When an RE captures a presence broadcast message from a WD, the RE records the RSSI value corresponding to that WD received at this RE's position. When it gathers a certain number of RSSI values or exceeds a timeout, it will forward these RSSI values to the BS. The BS will have RSSI values of all WDs captured by REs in the system.

Using RSSI and strategically located REs, it is possible to construct a 2D map showing a rough location of a WD in real-time. However, this will require substantial investment and technical expertise in constructing the map and tuning placements of REs to achieve good results beforehand. For example, location fingerprinting consists of two stages: offline and online. The offline phase is carried out during the deployment. It measures RSSI values at many known locations and constructs a map based on these measurements. During this phase, effects of antenna polarization and moving people or objects have to be taken account of. The online phase samples RSSI values, looks them up against the map and figures out the most possible locations. As each deployment will be somewhat different, the offline phase requires significant investments of expertise and equipment every time.

In fact, precise location tracking is unnecessary in *e-Guardian*. *e-Guardian* is designed to be a low-cost and easy-to-deploy system. Localization techniques will add significant cost to each deployment and also maintenance thereafter. In *e-Guardian*, location tracking is not meant for continuous and real time tracking, which might also be controversial in terms of privacy. The most important purpose of location tracking is to quickly narrow down the search space during an emergency. An easier and more generalizable approach to the problem, as implemented, is simply summarizing the top three REs which have recently reported RSSI values of the anomalous WD, sorted by average RSSI values in descending order. As long as all REs are meaningfully aliased (e.g. by their locations), medical personnel can quickly narrow down their search to a small region within which not much extra effort is needed to locate the WD. For example, after a fall, medical personals will receive a message such as: "Fall alert from patient Robert, nearest REs: level 2 toilet B, level 2 corridor outside TV room, level2 TV room". This amount of information is believed to be sufficient to give a timely clue to medical personnel about the rough location of the emergency site.

The implementation of leave alert takes advantage of resources available in location tracking and thus the implementation is a bit different from the proposed solution in 3.4.2.2.3. As the BS receives RSSI values of all WDs reported by various REs in the system, the BS simply looks up all recent presence broadcasts to check if any WDs have no RSSI values received within three polling periods.

4.3.2 Automatic Removal of Stale Children

WDs in *e-Guardian* are mobile nodes, as they move, they are dynamically attached to new parent REs. As mandated by the Cskip addressing scheme, each RE can only parent a maximum of 14 WDs at any given time. Each RE keeps an internal association table to keep track of its child REs and WDs. As a new WD requests to associate with an RE, an entry in the association table will be used. As WDs in *e-Guardian* are allowed to move around frequently, this association table runs out of its space limit easily. Thus, the association table of an RE has to frequently clear stale child WDs to make room for new WDs. An automatic stale children removal mechanism has been implemented, taking advantage of WDs' presence broadcasts. If any child WD fails to report within five periods of presence broadcast, RE will remove it from the association table, making room for new WDs.

4.4 Wearable Device

A prototype WD shown in Figure 4.7 is designed as a wrist watch device, which consists of a CC2430 chip, two LED indicators, a buzzer and an ultra low-power digital MEMS accelerometer, ADXL345 [58], from Analog Devices.

The WD is configured as a sleeping ZigBee End device, which means its radio will be turned off when idle and its parent RE will hold messages for it when it is sleeping. The design philosophy of the prototype WD is that seniors are only expected to press a panic button when they need help. The rest of functionalities are automatic.

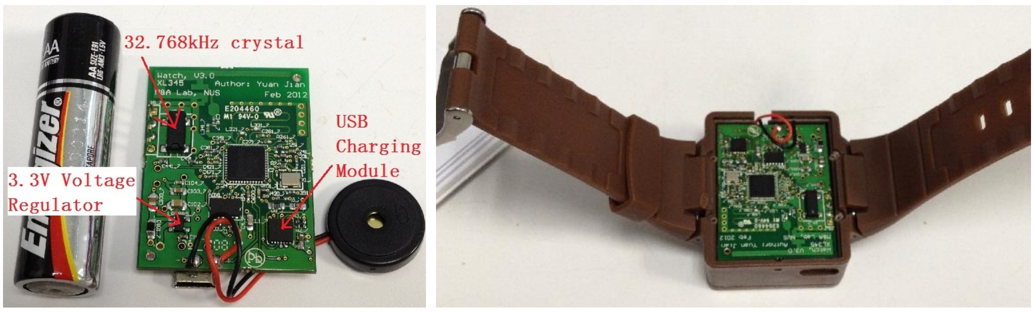

Figure 4.7: Prototype: Wearable Device

4.4.1 Dimension

The WD prototype has a dimension of $3cm \times 4cm$, which can be fit into a customized watch case. This dimension can be further reduced. The current WD prototype is powered by a 3.7V Li-ion battery. A USB charging module and a 3.7V-to-3.3V voltage regulator are needed, as shown in Figure 4.7. CC2430 has a wide supply voltage range from 2.0V to 3.6V. If a button cell battery with nominal voltage of 3.0V is used instead, these two modules as well as their external components can be eliminated. In addition, the 32.768kHz crystal, which is optional and is for precise timing functions in sleep mode, can be removed. It is not used currently. Furthermore, the PCB antenna and the microstrip balun could be replaced by a smaller chip antenna and a smaller chip balun respectively, at a higher cost. A dimension of $2cm \times 2cm$ could be easily achieved and the overall size of a watch can be much smaller than the current one.

4.4.2 Low Power Management

A WD takes advantage of CC2430's power modes described in Table 4.1. In a WD, CC2430 stays in $PM2$ most of the time. In this mode, CC2430 only consumes $0.9\mu A$ and can be wakened up by an external interrupt or when the sleep timer expires. In this WD prototype, external interrupts can be button presses or ADXL345's interrupts. The sleep timer expires when a WD needs to poll from its parent RE periodically.

4.4.3 Presence Broadcasts

WD sends out periodic presence broadcasts which are extensively used by functionalities discussed in Section 3.4.1.2.2, Section 3.4.2.2.1, Section 3.4.2.2.5 and Section 4.3.2. Presence broadcasts are messages broadcasted with a radius of one to all REs (including BS). Broadcasting with a radius of one makes sure that REs capturing a broadcast will not forward it further to other REs. In ZigBee, typically, a broadcast from a sleeping

WD will always be a unicast to its parent router (e.g. RE) in the MAC level. Only in its NWK level, it is shown as a broadcast. Such a broadcast will not be received by other neighbors as this message will be filtered out in the MAC level. To address this issue, a presence broadcast must be sent with ZigBee routing capability disabled. In this way, the ZigBee stack will bypass its ZigBee routing rule and issue a MAC level broadcast. This will make a message to all its neighboring REs (including BS) in the hearing range.

A presence broadcast, captured by a sensor network analyzer, is shown in Figure 4.8. It can be seen that in the MAC level (i.e. IEEE 802.15.4), the destination address is 0xFFFC which indicates a broadcast to all routing devices (i.e. REs and BS). The radius counter is also set to 1.

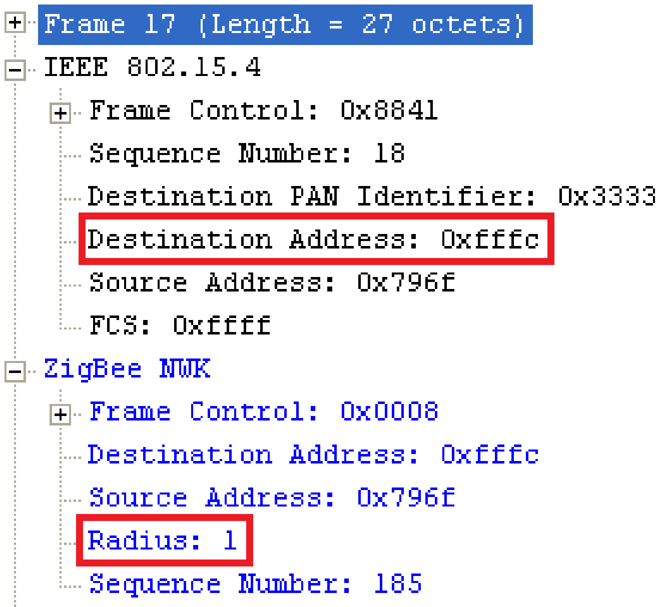

Figure 4.8: Anatomy of a Presence Broadcast

4.4.4 Fall Detection & ADL Classification

In this prototype, a fall detection algorithm and an ADL classification algorithm are designed and will be elaborated in Chapter 5. Both algorithms take advantage of advanced hardware features of ADXL345 to alleviate CC2430 from continuously sampling and processing accelerometer data, thus being able to achieve low power consumption.

4.5 Cost Estimation

In mass production, the primary cost is the cost of components. The PCB cost per piece will drop significantly as the number of boards increase.

At the time of this writing, the price for GM862 (specifically model number GM862-QUAD-PY) is US$79.96 each for orders more than 100 units [59]; the price for CC2430 is US$5.9 each for orders more than 10,000 units [60]; the price for ADXL345 is US$3.69 each for orders more than 5,000 units [61].

A realistic estimation of *e-Guardian* devices is approximately US$120 for a BS, US$10 for a RE, US$15 for a WD. As more and more seniors share the same BS, the cost per senior will be close to the price of a WD a senior wears. The advantage of being low cost will make *e-Guardian* affordable to the general public, especially in less-developed regions.

4.6 TCP/HTTP Server

A prototype of TCP/HTTP server has been implemented. The prototype has the same infrastructure that a full-blown TCP/HTTP server should have. This prototype serves as a start for building more features. A desktop client and a web application are both command-line-based. The TCP server, HTTP server and desktop client are all developed using Python 2.7.3.

4.6.1 TCP Server

The TCP server is developed using Twisted, an event-driven network programming framework for Python. The TCP server accepts connections from an *e-Guardian* system, a desktop client or the HTTP server on behalf of a client connection from a web browser.

When an *e-Guardian* system connects to the TCP server, it will send out its IMEI number and its system password. Upon a client connection, the TCP server will populate all online *e-Guardian* systems. The client could choose one of the *e-Guardian* systems to communicate with if a valid password is produced.

4.6.2 HTTP Server & Web Application

A WebSocket application and a HTML document with embedded JavaScript programs are developed. The application is served by a small standalone WebSocket server called pywebsocket, which can be also integrated into an Apache HTTP server. The HTTP server is hosted in the same server computer where the TCP server is hosted. Figure 4.9 shows a web connection from a Safari browser in an iPhone 4S smart phone.

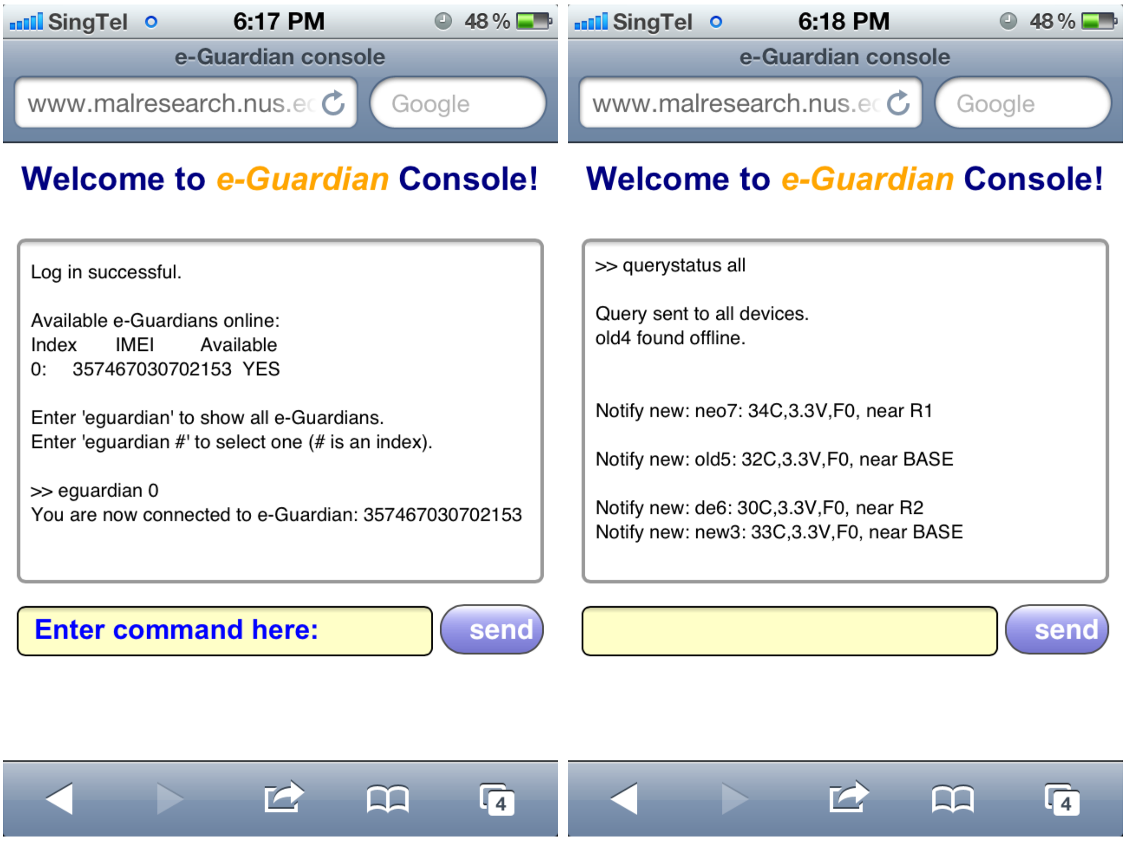

Figure 4.9: *e-Guardian* Web Application

The *e-Guardian* web console's URL is http://www.malresearch.nus.edu.sg:8000.

4.6.3 Desktop Client

The desktop client is implemented using wxPython, which is a cross-platform GUI API for Python. Figure 4.10 shows a connection to the TCP server directly via this desktop client in a Windows 7 PC.

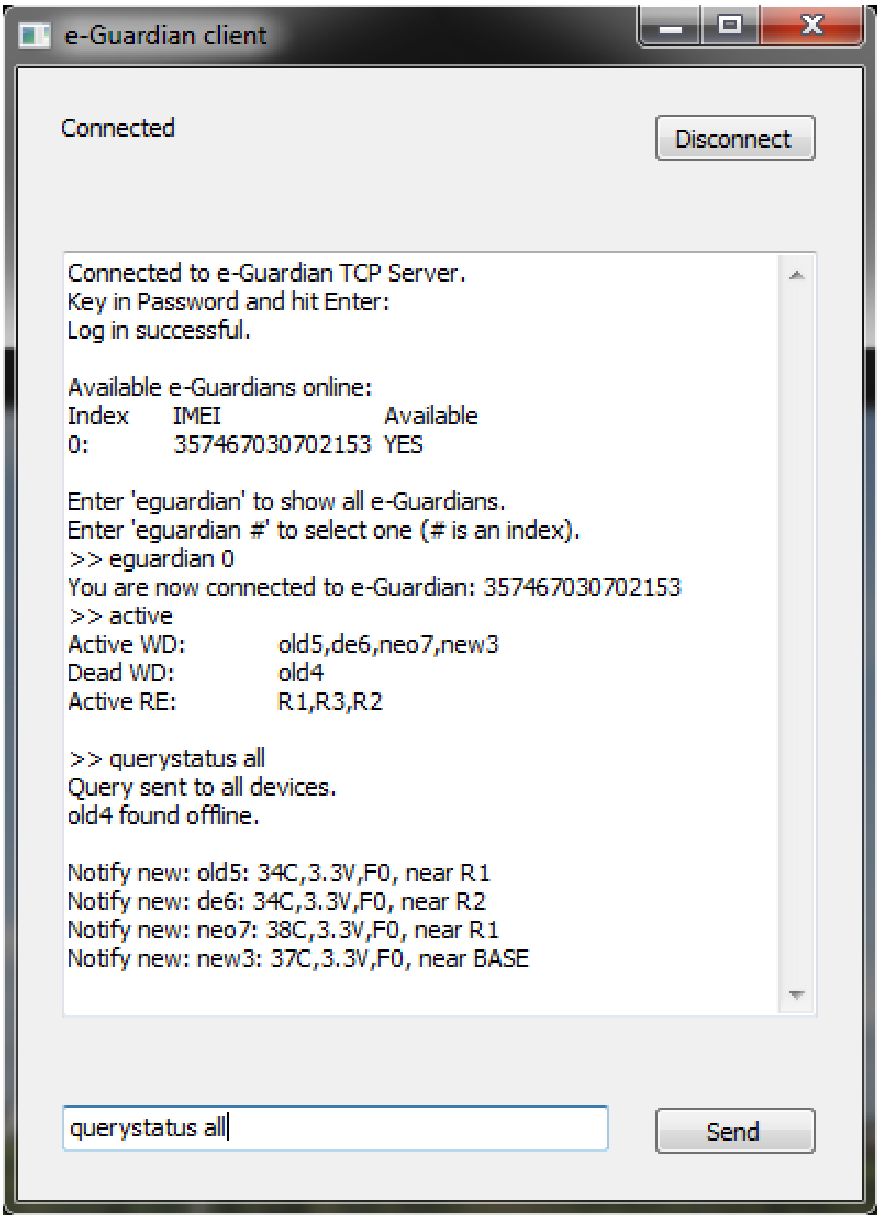

Figure 4.10: *e-Guardian* Desktop Client

Chapter 5

Design of Algorithms

A fall detection algorithm and an ADL classifier have been proposed. Both are interrupt-driven and hardware-dependent, which are completely different from conventional fall detection algorithms and ADL classifiers discussed in Section 2.3.1. The algorithms are based on the digital accelerometer ADXL345. The fall detection algorithm has been implemented in a wrist-worn prototype WD described in Section 4.4. The ADL classifier has not yet been implemented but porting the ADL classifier designed offline into the hardware is straightforward.

5.1 Hardware Setup

5.1.1 ADXL345

ADXL345 supports various interrupts, of which INACTIVITY, ACTIVITY, FREE_FALL are important for the proposed algorithms. All interrupts can be arbitrarily mapped to either one of the two interrupt pins: INT1 and INT2. These interrupt features allows the host MCU to go to sleep and serve accelerometer interrupts using an ISR. This has significant benefits over continuous polling of accelerometer data and can tremendously reduce overall current consumption.

For ACTIVITY and INACTIVITY interrupts, there are two modes, DC mode and AC mode respectively. In DC mode, the current acceleration magnitude is directly compared with THRESH_ACT and THRESH_INACT, threshold values for ACTIVITY and INACTIVITY respectively. In contrast, in AC mode, for ACTIVITY detection, a reference value is taken as the acceleration value at the start of activity detection. New samples of acceleration are compared to this reference value and ACTIVITY is triggered if the difference exceeds THRESH_ACT. For INACTIVITY detection, a reference value is used for comparison and is updated whenever the device exceeds THRESH_INACT. Subsequent samples are compared against this reference value. AC modes are very useful in detecting small movements, which are extensively used in the proposed algorithms.

To reduce the number of ACTIVITY and INACTIVITY interrupts, ADXL345 provides a link mode functionality, which serially links ACTIVITY and INACTIVITY interrupts. Link mode delays the start of ACTIVITY detection after an INACTIVITY is detected. Only after ACTIVITY is detected, INACTIVITY detection then begins, preventing the detection of ACTIVITY. Link mode makes it possible that when seniors are sleeping, no INACTIVITY interrupts will ever be generated.

In addition to interrupts, ADXL345 offers an FIFO memory block which could hold 32 sets of accelerometer data. With FIFO, the host MCU can examine accelerometer data of up to 32 samples in the past upon accelerometer interrupts. FIFO has four modes, of which TRIGGER mode is important for the two algorithms. A trigger event refers to interrupts attached to either INT1 or INT2, which is configurable. A trigger level for FIFO can be set to a number n, which can be anywhere between 0 and 31 (inclusive). Upon a trigger event, the FIFO will discard all accelerometer data in FIFO except the latest n samples. The FIFO will continue to fill up till it is full. This feature allows FIFO to retain a window of 32 samples around a trigger event and the host MCU can choose to run a relatively complex algorithm on these 32 samples.

Interrupts and FIFO allow an algorithm to skip an overwhelming majority of accelerometer data and only examine a tiny portion which is of interest. As both algorithms proposed are hardware-dependent, two schematics in Figure 5.1 and Figure 5.2 show connections of an SPI (Serial Peripheral Interface) bus and two interrupt pins between CC2430 and ADXL345 in the prototype WD. The host MCU CC2430 is configured as an SPI master while ADXL345 is an SPI slave.

5.1.2 Interrupt Service Routines

OSAL in Z-Stack has a high level software timer implemented on top of one of the on-chip hardware timers. This high level timer allows an application to schedule an event to be set so that its corresponding event handler will be executed at a later time. A scheduled timer event can be restarted or canceled. This feature is used to implement a flexible recurrent timer interrupt that can be enabled and disabled any time the program wishes.

Besides the timer interrupt, there are two hardware interrupt service routines implemented for INT1 and INT2 pins respectively.

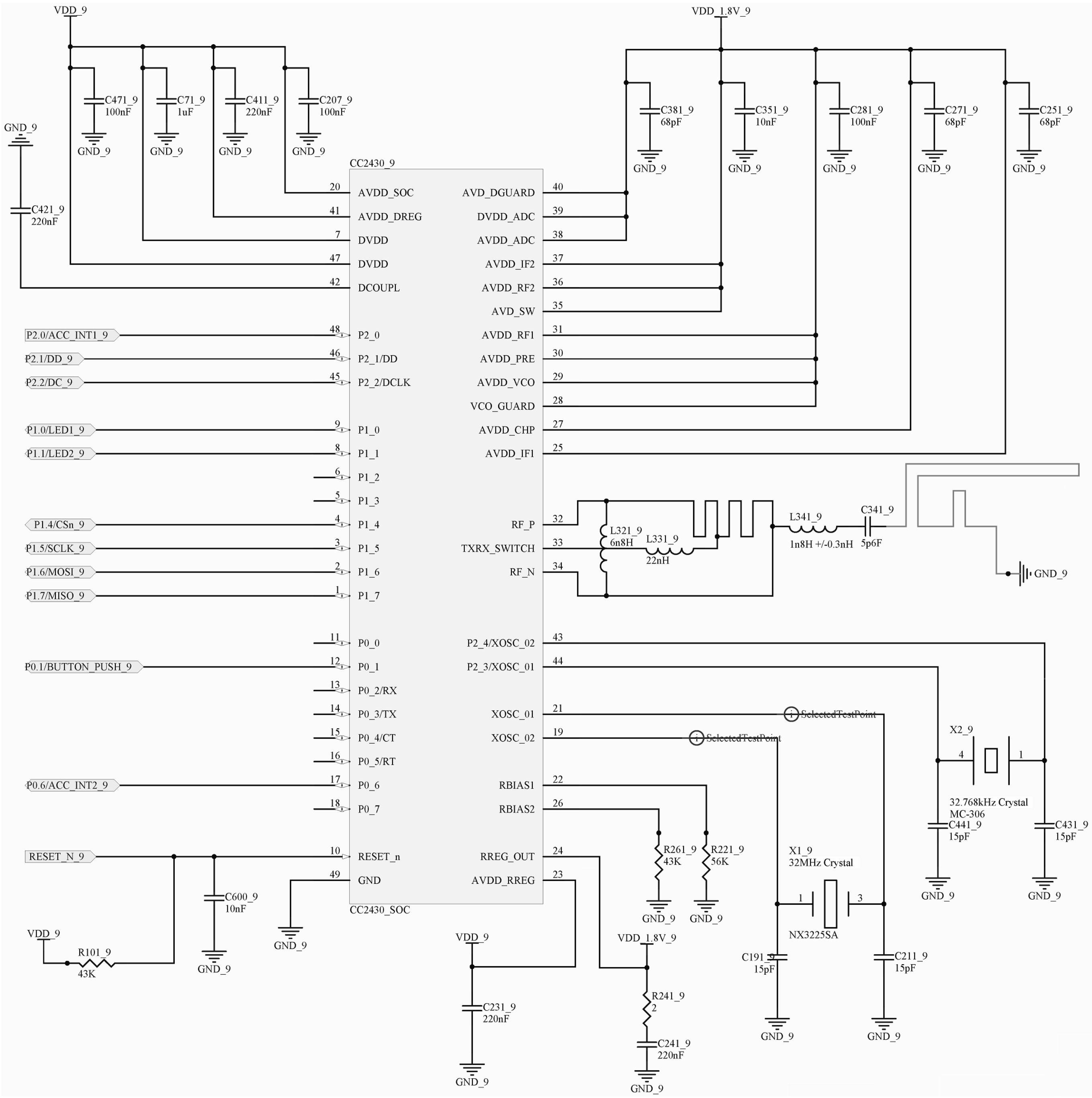

Figure 5.1: Schematic: CC2430 in Wearable Device

5.2 Sensor Placement

Currently, most fall detection algorithms and ADL classifiers are based on accelerometers attached to torso, as discussed in Section 2.3.1. Accelerometer data collected at

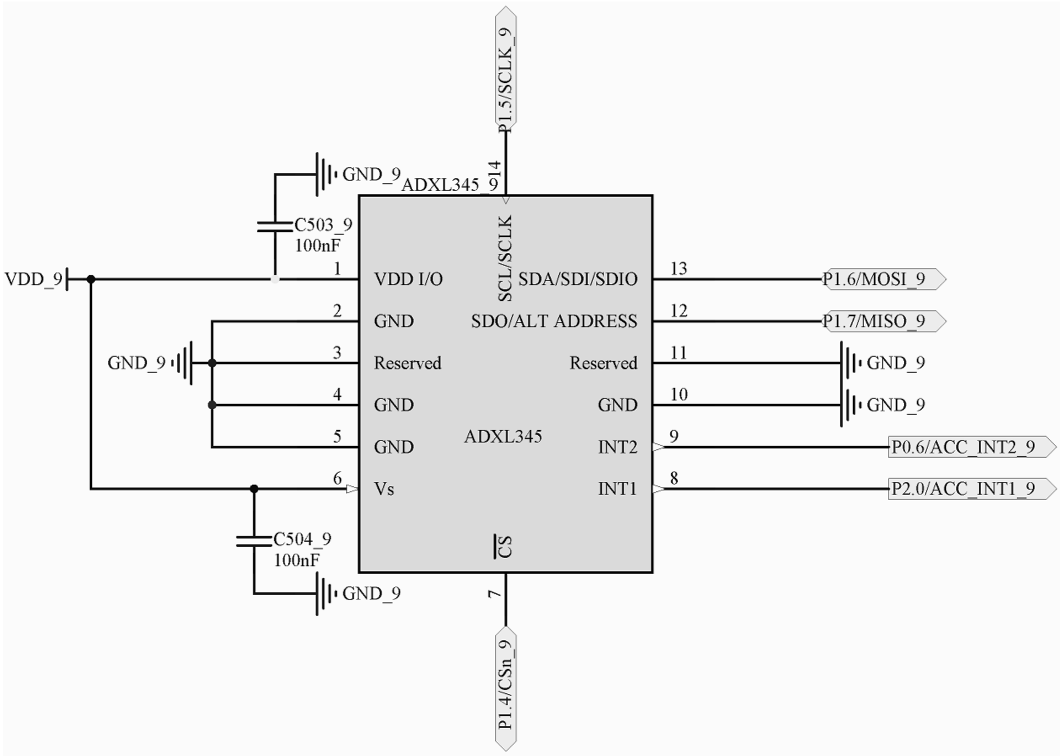

Figure 5.2: Schematic: ADXL345 in Wearable Device

wrists are typically used to complement accelerometers attached to torso. [32] evaluated accelerometers placed at waist, wrist and head and concluded that wrist was not optimal for fall detection. [62] showed that a wrist-worn accelerometer could not reliably distinguish falling and sitting down. Despite being rare and unreliable, [63, 64] both implemented a wrist-worn accelerometer-based fall detector.

Placing accelerometers at the center of gravity is the most reliable means, but also the least comfortable on a daily basis. A wrist-watch type would be the most acceptable as it can remove social stigma associated with wearing a medical or health care device. Another advantage of a wrist device is high acceptability for wearing at night and during bathing. The proposed algorithms are based on wrist-worn accelerometers despite the challenge to detect falls due to complicated movements of wrists in daily activities.

5.3 Fall Detection Algorithm

The proposed fall detection algorithm derives from an algorithm [65] proposed by Analog Devices. The original algorithm characterizes a fall by four phases as shown in Figure 5.3. WEIGHTLESS, IMPACT and MOTIONLESS are detected by using ADXL345's FREE_FALL, ACTIVITY and INACTIVITY interrupts respectively. In a valid fall, WEIGHTLESS, IMPACT and MOTIONLESS must occur in succession within predefined time intervals. A fall is finally qualified if vector difference between final orientation and initial orientation surpasses a threshold, which indicates a significant orientation change.

The original algorithm has three major limitations.

1. It relies heavily on the host MCU to check the accelerometer's interrupt status every 20ms using a recurrent timer interrupt, even if seniors may be sleeping through the whole night. Even though this is significantly better than a continuously polling approach, it is still similar in the way that both approaches waste a majority of their time doing useless checking.

2. It is not designed for wrist-worn devices. It assumes a fixed initial status of gravity orientation (x = 0, y = -g, z = 0). Immediately after a senior falls, the final orientation will be significantly different than the fixed initial orientation. This assumption only makes sense if an accelerometer is worn with one axis parallel to the gravity. For example, the accelerometer is embedded within a belt worn by a

standing human subject. The assumption is totally invalid if the accelerometer is wrist-worn.

3. The proposed FREE_FALL threshold will lead to continuous assertions when an accelerometer is positioned in certain orientations, even though the wearer is not falling. This will be explained thoroughly in Section 5.3.1.2 below.

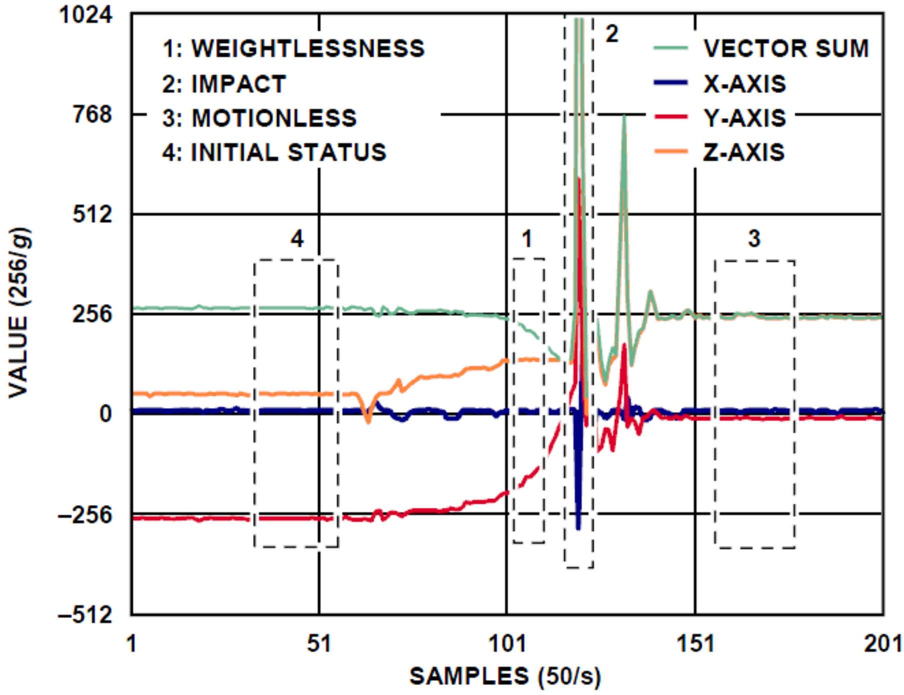

Figure 5.3: Four Phases During a Fall, Image Courtesy of Ning Jia

The proposed algorithm has made significant improvements in terms of power saving by removing its timing dependency on the host MCU as much as possible. As sleeping is an instinct of a ZigBee End Device, this algorithm keeps the host MCU, CC2430, in the sleep mode as much as possible by relying on ADXL345's interrupts. It uses the host MCU's timer sparingly, i.e. only when absolutely necessary.

5.3.1 Accelerometer Configurations

5.3.1.1 Output Data Rate

Conventional fall detection algorithms require sampling rates of no less than 50Hz, as discussed in Section 2.4.3. A high sampling rate is required to capture high frequency components which might contain a fall.

ADXL345 provides lower ODR (output data rate) by decimating a common sampling frequency. High frequency components such a sudden and short impact is likely not be present in sampled data. However, high frequencies up to 1600Hz can be accurately captured in interrupts such as ACTIVITY, FREE_FALL as these interrupts are based on undecimated accelerometer data. Thus, ADXL345 could be safely set to a lower ODR without worrying that high frequency signals such as a fall will be missed.

The proposed ODR is 25Hz. At this rate, FIFO will be able to hold $32/25 = 1.28$ seconds of data. As FIFO TRIGGER mode is used and upon a trigger, only the latest 16 samples are kept. These 16 samples, corresponding to 0.64 seconds of accelerometer data, will be enough for estimation of initial postures of wrists.

5.3.1.2 Threshold of Falling

During a fall, a wrist typically falls from a high level to a low level within a very short time. During this time, the accelerometer attached to the wrist will experience some gravity loss, which is reflected in the accelerometer reading by all axes converging towards zero. This pattern can be exactly captured by using a FREE_FALL interrupt.

[65] proposed 0.75g and 30ms for THRESH_FF and TIME_FF which are threshold

and time window of FREE_FALL interrupt respectively. There is a serious flaw with this setting. ADXL345 asserts FREE_FALL when all axes are smaller than THRESH_FF for a time longer than TIME_FF. This assertion is not based on sum of vector of all axes which is $sv = \sqrt{x^2 + y^2 + z^2}$. It is desirable to have a FREE_FALL interrupt feature based on sv. However, none of the digital accelerometers in the market right now provide this feature.

There is a slight possibility that, the accelerometer is positioned such that all axes have equal angles with the gravity vector, as shown in Figure 5.4.

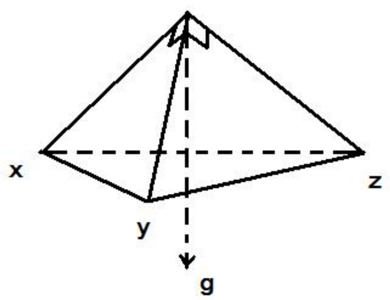

Figure 5.4: All Axes with Exactly Equal Angles with the Gravity

The projection of the gravity vector onto each axis can be calculated by using volume of the triangular pyramid. Assume the three sides on the top all have a length of 1, the volume of the triangular pyramid will be

$$V = (\frac{1}{2} \times 1 \times 1) \times \frac{1}{3} = \frac{1}{6} \tag{5.1}$$

Another way to calculate the same volume is

$$V = \frac{1}{3} \times h \times S$$

where h is the height of the triangular pyramid when positioned as in Figure 5.4 and S is the area of the bottom face. S can be easily calculated as it is easily obtainable that all sides of the equilateral triangle equal to $\sqrt{2}$. Together with area of triangle using trigonometry,

$$\begin{aligned} V &= \frac{1}{3} \times h \times S \\ &= \frac{1}{3} \times h \times \frac{1}{2} \cdot \sqrt{2} \cdot \sqrt{2} sin(\frac{\pi}{3}) \end{aligned} \tag{5.2}$$

Equating Equation 5.1 for Equation 5.2 gives $h = \frac{\sqrt{3}}{3}$. Thus, the projection angle is $\theta = sin^{-1}(\frac{\sqrt{3}}{3})$. Thus, the projection of the gravity into each axis will be $\frac{\sqrt{3}}{3}g \approx 0.57735g$. Thus, when the accelerometer is positioned as in Figure 5.4, FREE_FALL interrupts will be continuously asserted as all axes are smaller than THRESH_FF ($0.75g$). This will be inaccurately detected as a long free-fall in the original algorithm.

The new threshold proposed is $0.5625g$ which is the largest configurable value under $0.57735g$. The TIME_FF is correspondingly smaller, which is set to $20ms$ instead. This setting has been tested to be responsive enough when wrists lower down relatively quickly in experiments.

5.3.1.3 Threshold of Impact

During a fall, a wrist will typically hit the ground, or occasionally walls. The resultant impact can be detected by using ACTIVITY interrupt. However, during ADLs, there are occasionally actions which lead to relatively high impacts. Impacts caused by ADLs are normally smaller than impacts caused by falls but there is certain region of overlap between the two, as studied by [66]. A threshold must be set to differentiate fall impacts from ADL impacts. From extensive experiments, during a fall, this impact will easily exceed 4g while only occasionally ADL exceeds this value.

The ACTIVITY interrupt is asserted when any of the axes exceeds THRESH_ACT. Due to a problem similar to that in Section 5.3.1.2, during an impact, the vector sum of the impact could have a direction with the same angles with all axes. Thus, this impact will be projected equally to each axis, leading to smaller values detected at each axis. This scaling factor is still $\frac{\sqrt{3}}{3}$. Therefore, THRESH_ACT is set to be 2.25g, the largest configurable value under $\frac{\sqrt{3}}{3} \times 4g \approx 2.31g$.

5.3.1.4 Assumption of Orientation Change

Most fall detections using accelerometers around torso could easily assume that before and after a fall, there will be significant change in orientation [29, 31, 34, 67]. For wrists, it is less certain to assume an orientation change. However, extensive experiments performed by human test subjects show that during an actual fall, orientation or posture of wrist will almost certainly change. This is due to the fact that, most falls occur when seniors are upright (walking or standing). As a result, their lower arms usually point roughly downwards. After falls, their lower arms will be typically horizontal. Therefore, after a fall, at least a slight orientation change could be legitimately assumed.

The assumption of slight change in orientation will help eliminate the majority of ADLs. As discussed in Section 5.3.1.3, there is an overlap between impacts due to falls and impacts due to ADLs. During ADLs such as sitting down and placing arms on tables, their wrists will typically hit surfaces of tables, thus leading to impacts, which can be relatively high sometimes. However, most conscious human actions usually do not result in much change in orientation before and after impacts. This is certainly not always true. Nevertheless, the assumption of slight change in orientation is able to filter out quite a lot of ADLs while it is less likely to filter out a fall mistakenly.

The slight change in orientation is quantitatively defined as a relatively small vector difference of 0.5g between the gravity vector measured in the accelerometer before and after a fall.

5.3.2 Algorithm Description

The proposed algorithm is modeled as a finite-state machine. A flow chart shown in Figure 5.5 is used to illustrate the algorithm. It essentially consists of six states, from F0 to F5. The details of this algorithm are described as below.

(F0) F0 is the initial state as well as reset state. ADXL345 is initialized as follows:

- (a) Data rate: 25Hz.
- (b) ACTIVITY, INACTIVITY are enabled and mapped to INT1. ACTIVITY threshold is 2.25g. INACTIVITY threshold is 0.5g and its detection time window is 1 second.
- (c) Link mode is enabled. By enabling link mode in the state F0, INACTIVITY will not be asserted repeatedly if seniors are not moving (e.g. sleeping).

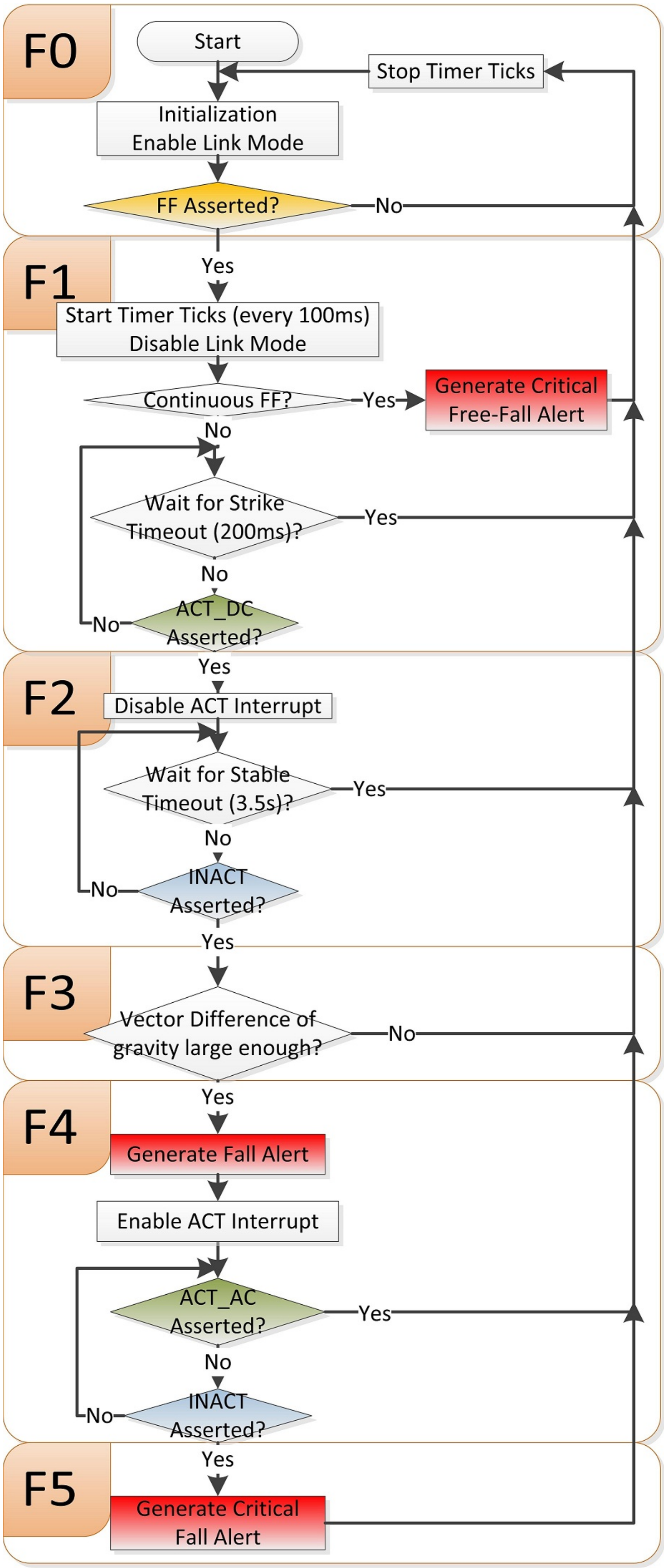

Figure 5.5: Flow Chart of the Proposed Fall Detection Algorithm

(d) FREE_FALL is enabled and mapped to INT2. FREE_FALL threshold is 0.5625g and its detection window is 20ms.

(e) FIFO is initialized to TRIGGER mode and is triggered by FREE_FALL. When triggered, FIFO will hold the latest 16 samples before the trigger, discard earlier ones, continue to collect until full.

(f) When returning from other states, CC2430 timer ticks (started in F1) will be stopped and the system is reinitialized.

(F1) When FREE_FALL is asserted, the algorithm enters F1.

(a) FIFO is triggered and will hold latest 16 samples.

(b) CC2430 starts a 100ms timer. This timer is recurrent, which means it will start another 100ms timer when the previous one expires.

(c) Link mode is disabled, so both ACTIVITY and INACTIVITY are ready to be detected.

(d) If FREE_FALL is asserted for more than 300ms (i.e. 3 timer ticks), a critical free-fall alert will be generated.

(F2) If ACTIVITY is asserted within 200ms after FREE_FALL was asserted, the algorithm enters F2, otherwise F0.

(a) ACTIVITY is disabled. During an impact, ACTIVITY might be asserted multiple times since link mode is disabled in state F1. One assertion is enough for judging an impact and subsequent assertions are inhibited by disabling ACTIVITY.

(b) CC2430 retrieves 16 data samples in FIFO, averaging the first 10 samples as the initial posture of wrist. At a data rate of 25Hz, 10 samples equal to 0.4 (10/25) seconds of data, 0.24 ((6-6)/25) seconds before FREE_FALL. This initial status will be useful in F3.

(F3) If INACTIVITY is asserted within 3.5s after ACTIVITY, the algorithm enters F3, otherwise F0. In F3, CC2430 retrieves the latest 10 samples and does an average on them as the final posture of wrist. The vector difference between the final and initial postures is calculated.

(F4) If the difference exceeds 0.5g which suggests a slight change in orientation, a fall alert is generated and the algorithm enters F4.

(a) ACTIVITY is enabled again.

(b) INACTIVITY detection time window is set to 10s to determine if there is a long fall (long motionless).

(c) If ACTIVITY is asserted in F4, the system goes to F0 and gets ready for another round of detection.

(F5) If INACTIVITY is asserted after 10s, this indicates long motionless and the system goes to F5. A critical fall alert is raised and the system goes to F0 immediately.

In this algorithm, when a senior wearing a wrist-worn WD is static (e.g. sleeping or resting), the algorithm will stay in state F0. Only a FREE_FALL interrupt will be able to bring the algorithm to state F1. FREE_FALL interrupt will be generated when the senior lowers down his/her wrist. In state F1, if four continuous FREE_FALL (corresponding to a 300-millisecond period) occur, a critical FREE_FALL will be immediately generated as normal falls will not take as long and this certainly corresponds to a fall from high level. In state F1, the algorithm will consider an ACTIVITY interrupt (i.e. an impact) which follows the first FREE_FALL interrupt within 200 milliseconds (Strike Timeout) to be a possible fall. This timeout limits the maximum time the algorithm can stay in state F1 each time.

Similarly in state F2, the fall detection algorithm starts to wait for an INACTIVITY interrupt. The INACTIVITY interrupt is configured for a time window of one second. Within the Stable Timeout (3.5 seconds), if the senior's wrist is inactive for at least one second, the fall detection algorithm will move to the next state to check further if it is a valid fall. The Stable Timeout also limits the maximum time the algorithm can stay in state F2 each time.

State F3 is a transient state which only determines if the orientation change is large enough to be qualified for a fall. If yes, the algorithm will go to state F4 and a fall alert will be triggered. The rest of the algorithm will still check if the senior is lying in coma (i.e. long lie).

It can be seen that all states except F0 are temporary states which could not last longer than certain timeouts. As a result, the algorithm will spend the majority of its time in state F0.

5.3.3 Power Improvement

As shown in Figure 5.5, the 100-millisecond timer ticks will be stopped immediately when the algorithm enters F0 and will only be turned on when state F1 is entered. In this way, the host MCU (CC2430) will be able to completely stay in sleep mode when the algorithm is in state F0. To simplify the estimation of improvement in power consumption compared to the original algorithm, WD is assumed to be static which means it will not trigger FREE_FALL and ACTIVITY interrupts. This will be true the majority of the time, especially when WDs are worn by seniors who are relatively inactive. The original algorithm requires CC2430 to trigger a timer event every 20 ms. CC2430 consumes 492 μA current in the active mode $PM0$ (with radio off) and 0.9 μA in the sleep mode $PM2$. It takes at least 54 μs to switch from $PM2$ to $PM0$ and vice versa. Context switch due to interrupt handling, calculations and serial communications with ADXL345 are all assumed to be quick enough and thus neglected for the convenience of estimation. The duty cycle of the 20 ms timer interrupt will be $54\mu s \times 2/20ms = 0.54\%$. The average current consumption will be $0.54\% \times 492\mu A + (1 - 0.54\%) \times 0.9\mu A \approx 3.55\mu A$. Compared to the new algorithm which never uses this 20 ms timer when static, the power consumption is nearly four times ($3.55/0.9 \approx 3.94$). In reality, interrupt handling, calculations and serial communications will take quite some time, therefore, the actual power saving is much more than four times.

5.3.4 Discussions

There are a variety of different approaches to automatic fall detection, which can be based on video-cameras, acoustic or inertial sensors, and mobile phone technologies [27]. The comparison here will focus on fall detection algorithms based on inertial sensors and more specifically, designs based on a single accelerometer.

In assessing performance of fall detection algorithms using accelerometers, besides accuracy, there are two other important aspects which must be taken account of. These aspects are:

1. Accelerometer position
2. Power consumption

As discussed in Section 6.3.2, there is a trade-off between accelerometer position and accuracy. Due to the nature of human movement, torso is more stable than wrist thus it is easier to differentiate a fall from accelerometer data collected at torso. However, attaching sensors to torso is much less comfortable than wrist and sensors must inevitably take them off when they bathe or change cloths. From the practical design point of view, a wrist-watch device is the most acceptable for seniors. *e-Guardian* aims to be a practical system beyond research labs. Seniors will simply refuse to wear a device if it causes discomfort or social stigma of wearing a medical or assistive device. The proposed fall detection algorithm chose wrist other than waist or chest because comfortability is one of most important factors in designing a wearable device. If a fall detection device is 100

There is an interesting observation that a big discrepancy exists between academia and commercial area in terms of their approaches to fall detection. In research area, almost all published algorithms use accelerometers attached to torso. Some used more than one sensor. It is no coincidence that these algorithms unanimously chose to attach accelerometers to torso because their goal is always better accuracy for fall detection. Power consumption and comfortability are secondary. In contrast, many commercial fall detectors are wrist-worn. It is also no surprise that wrist-worn types are prevalent in the market as comfortability is not less important than accuracy. However, no published details of those algorithms are available. It is difficult to assess their effectiveness without getting hold of one such device. Philips Lifeline with AutoAlert, even though it is not wrist-worn, is among the most famous and successful fall detectors in the market. However, there is neither published data about its accuracy nor its implementation details available currently.

The utmost advantage of the proposed algorithm over conventional ones is the power consumption. Conventional algorithms, even if some use digital accelerometers with interrupts and FIFO, did not take advantage of hardware features of a digital accelerometer but still use an approach to retrieve all accelerometer data and process each one of them. The host MCU will be kept busy all the time. The proposed algorithm allows the host MCU to be sleeping most of time and wake up occasionally to process accelerometer data upon accelerometer interrupts. However, the proposed algorithm only makes decisions based on a tiny portion of accelerometer data, it is unlikely to outperform the conventional algorithms in terms of accuracy. The detailed comparison of the proposed algorithm with conventional ones is in Section 6.3.2.

5.4 Classification Algorithm for Activities of Daily Living (ADL)

The proposed fall detection algorithm features a finite state machine. Depending on the wrist's movement, the algorithm flows dynamically among the six states. The time it spends in each state gives a lot of information about activities of a wearer.

Traditionally, studies of ADLs such as Wockets [40] aim to identify a number of activities including walking, climbing stairs, sitting, lying, sleeping, bathing, doing laundry, cooking, using multiple sensors at various parts of human body. Since only one sensor is used in the prototype WD, there is only a subset of activities which can be inferred from accelerometer data. The identifiable activities are "Walk", "Random"(random wrist movements), and "Quiet" (no movement at all). These activities are purely inferred from accelerometer information. However, it is possible to categorize these activities into more specific activities or even abnormalities by using accelerometer data along with additional contextual information such as time and location. For instance, "Quiet" at mid night most likely means sleeping. "Random" during night indicates poor sleeping quality. "Quiet" in bath rooms longer than certain time (e.g. 30 minutes) will be abnormal and an alert has to be raised.

5.4.1 Feature Extraction

The features used to study ADL are extracted from the fall detection algorithm. However, a minor change is necessary, which do not affect the correctness of the proposed fall detection algorithm but will increase the number of interrupts generated when the wrist is doing "Random" activities. This change is made to state F0. Originally, in state F0, ACTIVITY threshold is set to 2.25g while INACTIVITY threshold is 0.5g, as shown in Figure 5.5. Even though these two thresholds are initialized in state F0, they are not used until state F2 and state F4 respectively. To capture ADL, a change needed is to set both thresholds to 0.75g in state F0, an empirical threshold obtained by experiments. Both interrupts need to operate in AC mode and link mode is still enabled. By setting both thresholds to the same small value, when a wrist moves a small amount, ACTIVITY will be asserted and when the wrist is relatively static, INACTIVITY will be asserted in 1 second. This setting only lives within state F0. Whenever the fall algorithm transitions to state F1, both thresholds will be immediately configured to 2.25g and 0.5g for ACTIVITY and INACTIVITY respectively.

With above hardware settings, four types of events can be identified from the fall detection algorithm and they are named with the following conventions:

1. $E0$: generated whenever an INACTIVITY interrupt asserted in state F0

2. $E1$: generated whenever an ACTIVITY interrupt asserted in state F0

3. $E2$: generated whenever the fall detection algorithm transitions from state F0 to

state F1

4. $E3$: generated whenever the fall detection algorithm transitions from state F1 to state F2

There are underlining physical meanings associated with these four events. $E0$ is generated whenever a wrist changes from active movements to motionless. $E1$ is generated when the wrist slightly moves or changes orientation. This corresponds to casual movements human makes during ADLs. $E2$ corresponds to weightlessness at the wrist, which is generated when the wrist is lowered from a higher level. This happens when the subject falls, walks (swings arms) and sits down. During ADLs, accelerometer values will typically alternate around the gravity value. Capturing only the weightless part is good enough to estimate the activity level. $E3$ is typically generated during drastic movements such as a fall, suddenly sitting down and putting down arms onto desks.

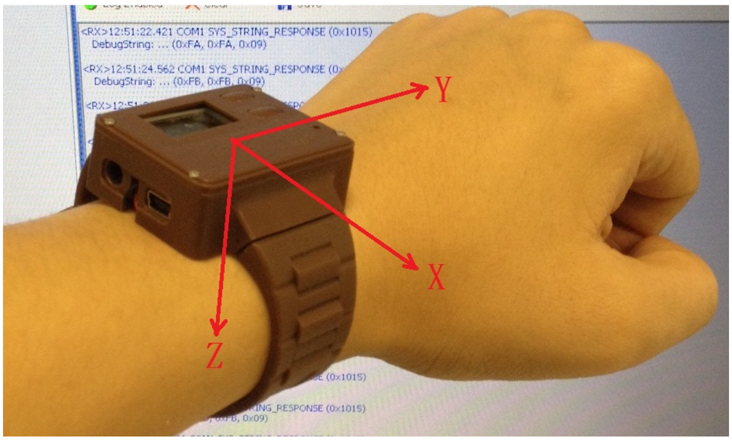

Figure 5.6: Alignment of Accelerometer Axes in a WD Worn by a Human Subject

Along with above events, directions of Y-axis along with these events are also taken as records. Figure 5.6 shows the alignment of axes of the accelerometer when worn by a human subject on his left wrist. Y-axis is always aligned with the subject's lower arm. Y-axis information is critically important for estimating ADL as Y-axis is typically pointing approximately downwards when the human subject is standing or walking. During sleeping, Y-axis is usually approximately horizontal. As the accelerometer's FIFO mode is used, the latest value is deep inside FIFO. Retrieving the latest sample requires retrieving all previous ones, which will mess up the fall detection algorithm. A relaxed strategy is to retrieve the first item in FIFO, corresponding to the oldest value in FIFO. During studies of ADL, the focus is on macro-movements which typically last several seconds or minutes. At an ODR of 25Hz, all values in FIFO are within $32/25 = 1.28$ seconds, which are decent representations of "recent" Y-axis values.

5.4.2 Preprocessing

Figure 5.7 shows raw data collected from a WD worn by a human subject who performed three types of actions including random, walk and static. Each action lasted a full minute exactly. Below is a list of intuitions based on the raw data.

- During "Random", only $E0$ and $E1$ are generated in alternating manners (as link mode is enabled).

- During "Walk", as arms swing, FREE_FALL interrupts are generated and the fall detection algorithm moves from state F0 to state F1. As there is no impact following FREE_FALL during walking, the fall detection algorithm returns to state F0 shortly. This pattern repeats and generates a lot of $E2$ as well as $E0$ and $E1$

along the way. Also observe that Y-axis values are always positive as Y-axis typically points downwards during walking.

- During “Quiet”, no events will be generated.

An ADL classifier is to be designed to differentiate these actions automatically.

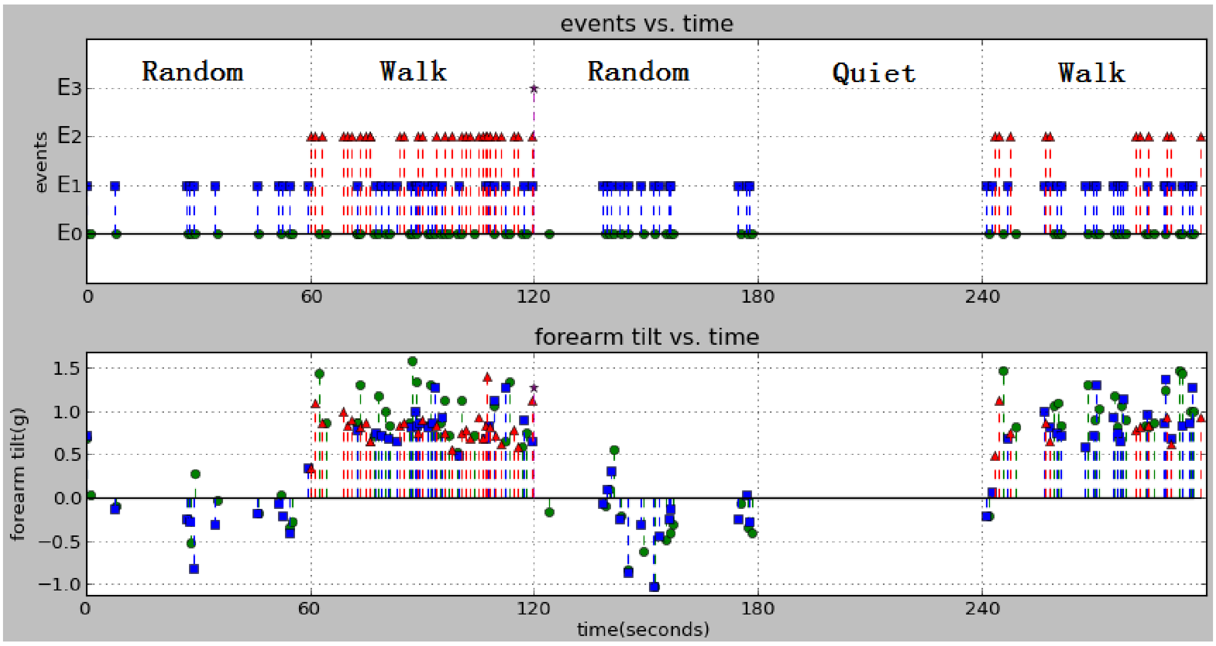

Figure 5.7: Four Events and Corresponding Y-Axis Values

It is to be noted that events are generated at any time. These events are asynchronous and non-uniform with respect to time, as opposed to typical uniform time series generated at a constant sampling frequency. Thus, the proposed ADL classifier is an asynchronous event-driven classification algorithm that only computes upon occurrence of an event. The ADL classifier is designed to be run completely inside a WD, similar to the fall detection algorithm. When a new event occurs, the classifier needs to determine the type of ADL based on the current event as well as events in the past few seconds. Currently, the “past” has been defined to refer to a time window of 10 seconds before the present time. To take the past into consideration, the classification algorithm can take either one of the two approaches proposed below:

1. Upon occurrence of an event, the host MCU retrieves all events in the past 10 seconds it stored earlier, together with their time stamps, and determine the current ADL based on these events.

2. The classification algorithm can construct an energy function for each event which records this event's activeness in the past 10 seconds. This energy function decays according to time. Thus, an event just occurred recently leaves more energy at the present time than an event which occurred a long time ago. The host MCU could tell if an event has just occurred recently by evaluating current energy level of this event.

The proposed ADL classifier takes the second approach. The first algorithm is difficult to implement in a small MCU which has limited RAM space. When WD is actively moving, the host MCU needs RAM space to keep record of a lot of events, which should be avoided if possible in an 8-bit MCU. In addition, this windowing approach will compute the same data multiple times as the 10-second time-window shifts by one each time.

The second approach is preferred due to easy implementation and less computational resources required. For each of the four events ($E0$, $E1$, $E2$ and $E3$), an energy function is proposed to keep track of the activeness of this event in the recent past.

The effect of an event to its energy function decays with respect to time. An exponential decay function is proposed in Equation 5.3.

$$\eta(t) = e^{-t/\tau}. \tag{5.3}$$

where t is the elapsed time and $\tau = 10$ seconds. 10 seconds for τ means that an event's energy decays to $e^{-1} \approx 0.37$ of its original value after 10 seconds.

Upon the occurrence of an event, energy of all events in the past are reduced by a factor η $(0 < \eta \leq 1)$ which is related to the elapsed time of the previous event. The energy of the occurred event will be incremented correspondingly. The exponential function for computing the decay factor need not be very accurate. It can be efficiently approximated by using a look-up table and interpolation in the host MCU.

The proposed ADL classifier is completely event-driven as it only computes upon occurrence of an event. This leads to a problem when a WD is "Quiet". Consider a human subject wearing this WD walks for a while, then suddenly stops and keeps still thereafter. An event $E0$ will be generated one second after he keeps still. As no event will be generated after this $E0$, the algorithm will not compute at all during this motionless period. At the time the algorithm computed $E0$, the ADL classifier would have determined the action during the past 10 seconds is "Walk". During this long "Quiet" period, there is no generated event that can trigger a computation. Thus, the ADL classifier will assume ADL during the whole "Quiet" period to be still "Walk" which is obviously wrong. To resolve this problem caused due to no events generated during "Quiet", an artificial event is created and named as E_{-1} to signify its special purpose. Upon occurrence of an E_0 event, after which might follow a long "Quiet" period, a dummy event is scheduled to be set in 5 seconds using a software timer in the host MCU. Any other event which occurs before this timer expires will cancel this timer. If no other event cancels, the algorithm, upon seeing this dummy event, will reset energy functions of all events to zero.

The proposed energy function is in fact a modified exponential smoothing algorithm for non-uniform time series. However, the energy function is not meant for transforming an input of non-uniform time series into a continuous output. Instead, the energy function is only calculated at discrete time instants, i.e. upon occurrence of an event.

The modified exponential smoothing algorithm is elaborated in Algorithm 1.

Algorithm 1: Exponential Smoothing for Non-uniform Time Series

---

Data: $D = \{d^0, d^1, ..., d^i, ..., d^n\}$ is a non-uniform time series, where $d^i = (t^i, e^i)$ is a time-and-event pair at time t^i and $e^i \in \{-1, 0, 1, 2, 3\}$.

Result: Energy value ${A_j}^i$ for each event E_j where $j \in \{0, 1, 2, 3\}$.

begin
- ${A_j}^0 \longleftarrow 1$ for $j = e^i$
- ${A_j}^0 \longleftarrow 0$ for all $j \in \{0, 1, 2, 3\}$ where $j \neq e^i$.
- $\alpha \longleftarrow 0.6$
- **for** $i \in \{1, 2, ..., n\}$ **do**
  - $\Delta t \longleftarrow t^i - t^{i-1}$
  - $\eta \longleftarrow e^{\left(-\frac{\Delta t}{10}\right)}$
  - **if** $\Delta t \geq 5$ **then**
    - ${A_j}^i \longleftarrow 0$ for all $j \in \{0, 1, 2, 3\}$
  - **else**
    - **for** $j \in \{0, 1, 2, 3\}$ **do**
      - ${A_j}^i \longleftarrow \eta \times {A_j}^{i-1}$
      - **if** $e^i \geq 0$ **then**
        - ${A_{e^i}}^i \longleftarrow (1 - \alpha) \times {A_{e^i}}^i + \alpha / \eta$

Applying the proposed smoothing algorithm to the non-uniform series in Figure 5.7 yields energy levels for all four events, as shown in Figure 5.8. Upon occurrence of any of the four events, energy functions for all events will be evaluated. Thus, at any time instant when an event occurs, there are four energy values corresponding to four events. Together with Y-axis value at this instant, there are totally 5 features which can be fed into a classifier.

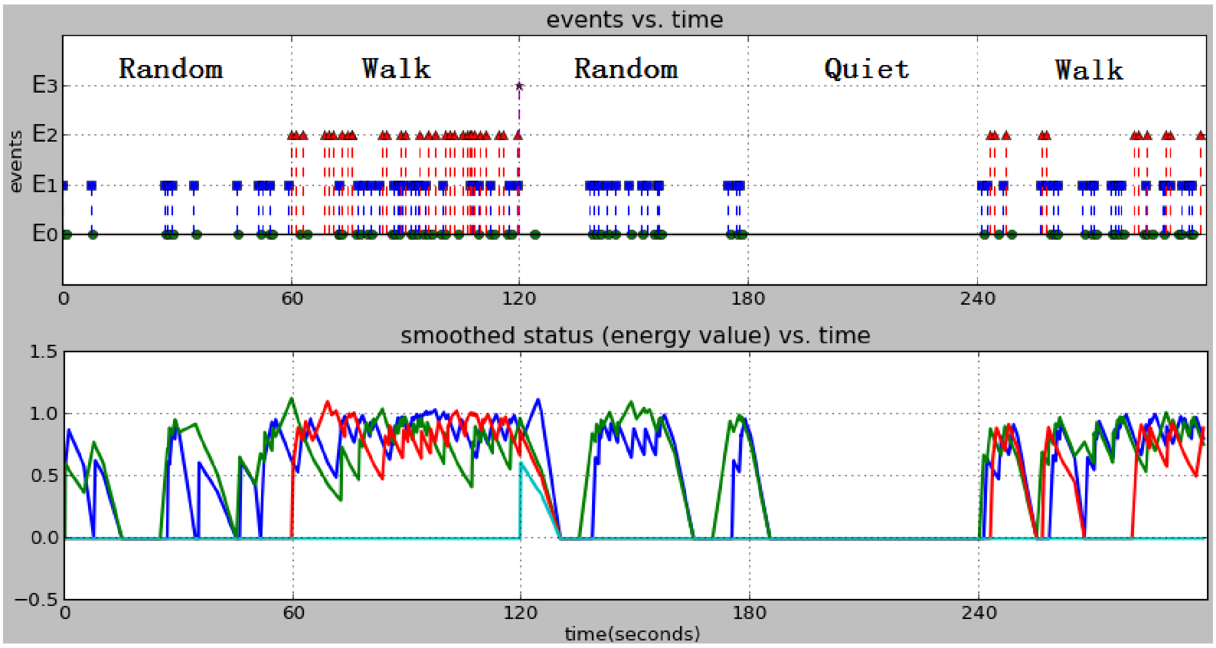

Figure 5.8: Energy Values for All Four Events

5.4.3 Decision Tree Training

ADL classifications using analog accelerometers typically use well-known classifiers such as LDA (linear discriminant analysis), naive Bayes, SVM (support vector machine), ANN (artificial neural network), decision tree classifier, etc, which have been compared by [37, 68, 69]. The proposed ADL classification employs decision tree learning as decision trees can be efficiently implemented in computers or MCUs as a number of if-else or switch statements. In contrast, ANN and SVM classifiers involve multiplications of many floating point numbers. In an 8-bit MCU where floating point arithmetic is not supported in hardware, a single multiplication typically takes tens of instructions cycles. The overall algorithm will be much more computationally expensive than simple if-else statements.

Decision tree learning has been extensively used in statistics, data mining and machine learning. It constructs a decision tree as a predictive model which maps observations to target values by the means of learning. ID3 (Iterative Dichotomiser 3), developed by Ross Quinlan, is a well-known learning algorithm for generating decision trees. Another popular method is CART (Classification And Regression Tree) which always produces a binary tree. Unlike CART, ID3 is not restricted to binary splits.

Decision tree learning is a type of supervised learning. Data is prepared with a number of attributes corresponding to the desired output values (a.k.a. target values).

One of the important concepts in ID3 is information entropy, as defined in Equation 5.4 [70].

$$E(X) = -\sum_{j=1}^{n} p_j log_2 (p_j) \tag{5.4}$$

where X is a variable whose n possible values have probabilities (a.k.a. proportions) p_1, p_2, ..., p_n.

Another important concept called information gain is defined in terms of information entropy, as in Equation 5.5.

$$G(S,A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E(S_v) \tag{5.5}$$

where:

- $Gain(S, A)$ is the information gain of a set of training data S after a split over the attribute A;
- $E(S)$ is the entropy of all target values in S; $E(S_v)$ is the entropy of target values of all training samples whose A attribute equals to v;
- $|S_v|$ is the number of samples whose A attribute equals to v; $|S|$ is the total number of training samples, thus $\frac{|S_v|}{|S|}$ is the frequency (proportion) of samples whose A attribute equals to v.

Entropy of 0 identifies a perfectly classified set. Entropy is used to determine which node to split next in the algorithm. Information gain measures the expected reduction in entropy by splitting over an attribute. ID3 selects the optimal split which means to select an attribute which yields the highest information gain.

C4.5 is an extension of ID3 and is also developed by Quinlan. It improves ID3 in a number of ways including handling both nominal and numerical attributes, handling missing attributes and pruning trees after creation to prevent over-fitting. In this project, C4.5 is used to train a decision tree. As C4.5 accepts numerical attributes, data obtained from the Algorithm 1 can be directly fed into the C4.5 training algorithm.

This project uses an implementation of C4.5 provided in Weka, a data mining suite written in Java, developed at the University of Waikato, New Zealand. Weka accepts training data in ARFF (Attribute Relationship File Format). A snippet of the training data is provided in Listing 5.1.

As there is no event data generated during a "Quiet" period, the training data will be biased towards "Walk" and "Random" as they have more training samples. To prevent this, artificial data is inserted into the training data every 1 second during "Quiet" periods. These artificial data has $E0$, $E1$, $E2$ and $E3$ all equal to zero, to indicate that none of the interrupts will be generated during a "Quiet" period. Y-Axis values are random values between $-g$ and $+g$, which is based on an assumption that tilt level of an arm can form any angle with the gravity vector during "Quiet".

Listing 5.1: Part of the Training Data in ARFF

```
@RELATION ADL

@ATTRIBUTE E0   NUMERIC
@ATTRIBUTE E1   NUMERIC
@ATTRIBUTE E2   NUMERIC
@ATTRIBUTE E3   NUMERIC
@ATTRIBUTE YAXIS    NUMERIC
@ATTRIBUTE TARGET   {Random, Walk, Quiet}

@DATA
0.594445866751,0.605606027623,0.0,0.0,0.7176,Random
0.874125264632,0.55309493264,0.0,0.0,0.0312,Random
...
0.868803201755,0.844929173529,0.78902974499,0.0,1.4352,Walk
0.800964506366,0.778954632099,0.941785751765,0.0,0.8736,Walk
...
0,0,0,0,0.197011500712,Random
0,0,0,0,-0.815356019351,Random
...
```

```
0,0,0,0,0.725084125307,Quiet
0,0,0,0,−0.218702173083,Quiet
...
0.650817753982,0.618025460739,0.0,0.0,−0.2184,Walk
0.591658526558,0.884732100327,0.0,0.0,0.0624,Walk
```

5.4.4 Classification Algorithm

Running the C4.5 algorithm in Weka on the training data leads to the following decision tree shown in Figure 5.9.

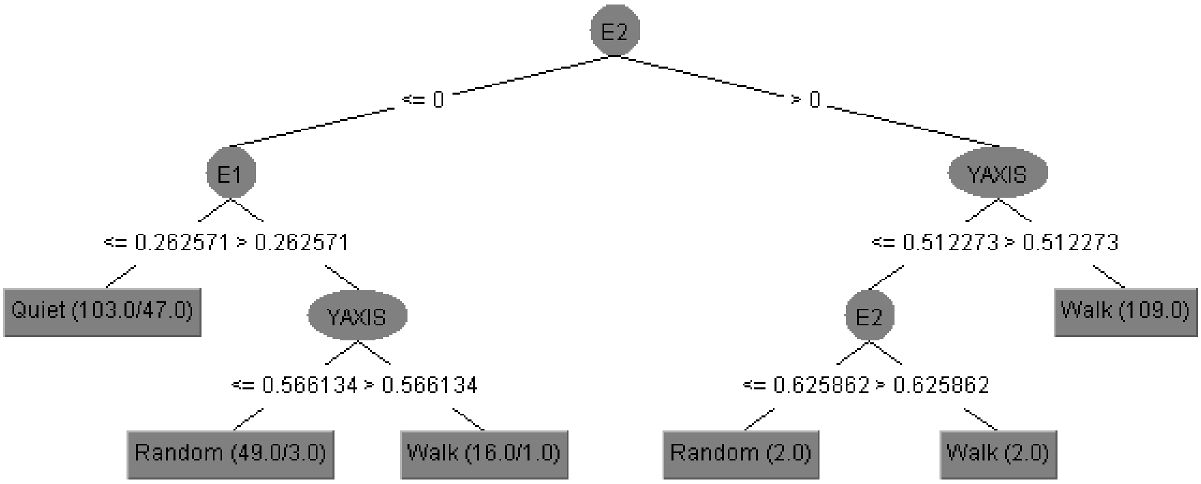

Figure 5.9: Decision Tree After Training

The resultant decision tree accords well with human intuitions. It selects energy values of $E2$ as the root split, which is the most obvious feature that differentiates "Walk" from other ADLs. Between "Quiet" and "Random", it determines that energy values of $E1$ are the best to differentiate them. It is also to be noticed that, two other features $E0$ and $E2$ do not appear in the decision tree. $E0$ has a strong correlation with $E1$ because link mode of ADXL345 makes them always appear in an alternating manner. $E2$, corresponding to impacts, which seldom appear in the training data, thus is not shown in the decision tree as a dominant factor. The resultant decision tree suggests that feature $E0$ and $E2$ could be removed from the feature set as they either has little effect or is redundant.

5.4.5 Classification Accuracy of Training Data

Figure 5.10 shows the classification result using the decision tree in Figure 5.9. The training samples are used as testing data. ADLs are color-coded. It can be seen that the decision tree performs very well. In fact, it is robust enough that some "Random" training samples have certain "Quiet" gaps in between. The algorithm is still capable of recognizing those gaps as "Quiet".

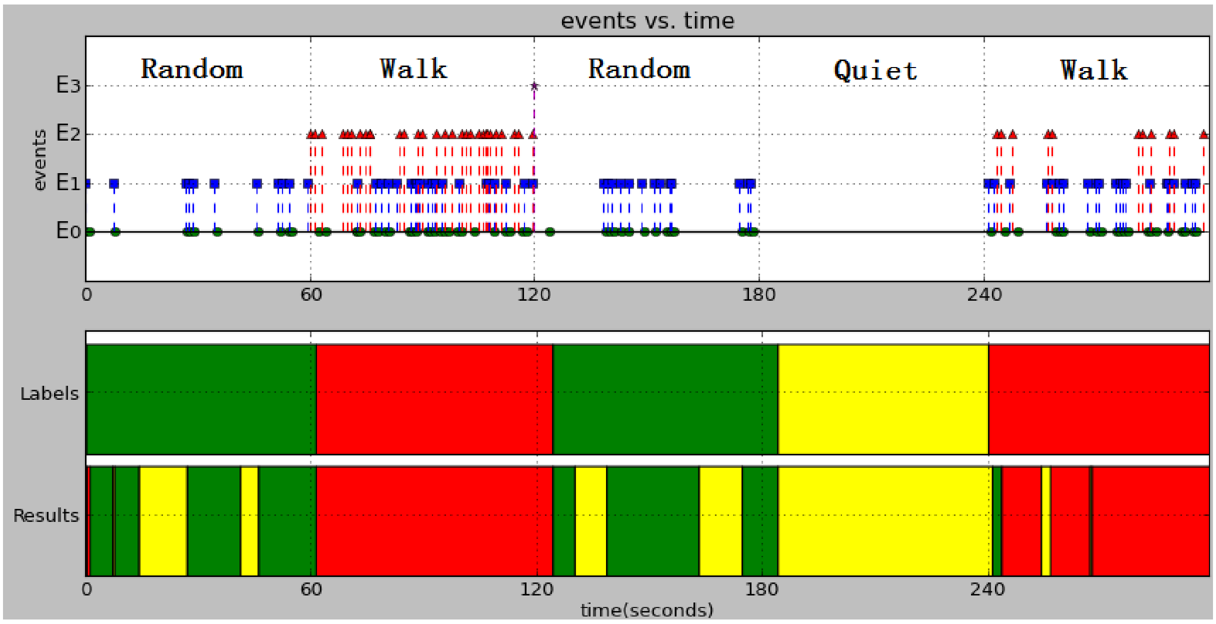

Figure 5.10: ADL Classification of Training Data

Note that the ADL classification algorithm has not been implemented in the prototype.

5.4.6 Uses of ADL Information

The proposed ADL classifier, despite being simple in terms of activities it could infer, offers high level information that could be fed back to the fall detection algorithm it is derived from, and also to low level applications to optimize power and bandwidth consumption. A few applications have been proposed below.

5.4.6.1 Improvement of the Fall Detection Algorithm

The proposed fall detection algorithm only considers a time window of slightly more than 10 seconds around a possible fall event. However, the next few minutes or even hours after a fall could still be related to that fall. Those ADLs could be used to estimate the severity of the fall. For example, after a fall, a senior lies in numbness (or coma) for a minute, then gradually recovers and is able to walk away. The system could relate this information to the fall and provide constant updates to caregivers. Or if a senior walks away right after a fall has been detected, the fall alert could even be safely disqualified and this fall goes to daily statistics only.

5.4.6.2 Accelerometer Power Minimization

ADXL345 has an auto-sleep feature, which can be enabled if link mode has already been enabled, which is the case in the fall detection algorithm. Auto-sleep allows ADXL345 to be automatically switched to sleep mode if INACTIVITY is enabled and triggered. During sleep mode, FIFO If ACTIVITY is enabled, ADXL345 will return to normal operation after detecting ACTIVITY.

However, the auto-sleep feature cannot be directly enabled as it will affect both algorithms. The fall detection algorithm requires FREE_FALL interrupt to be working. When ADXL345 is in the sleep mode, FREE_FALL is automatically inhibited and only ACTIVITY could wake up ADXL345. It can be seen from Figure 5.7 that, during walking, INACTIVITY and ACTIVITY are still asserted because INACTIVITY and ACTIVITY are both set to be sensitive about tiny movements. If auto-sleep is enabled and ADXL345 is placed into sleep mode every time INACTIVITY is detected, it will affect the detection of FREE_FALL thus changing the behavior and validity of the fall detection algorithm.

Auto-sleep could be enabled by using information from the ADL classifier. Upon detection of "Quiet", the host MCU will start a one-minute timer. During this one minute period, if no "Walk" and "Random" are detected, the host MCU could safely enable the auto-sleep feature of ADXL345. A realistic assumption here is that a senior wearing a WD could not fall suddenly after a long "Quiet" period without having any activity in between. Just enabling auto-sleep feature does not place ADXL345 into sleep mode, as ADXL345 only goes to sleep mode when INACTIVITY is triggered. To generate an INACTIVITY interrupt and make ADXL345 sleep, the host MCU could disable link mode and then immediately enable link mode again. According to the settings in the ADL classification algorithm, an INACTIVITY interrupt will be triggered in 1 second and ADXL345 will be in sleep mode after that. Immediately when the senior moves a little during the long "Quiet" period, an ACTIVITY will be generated. The host

MCU could then disable auto-sleep feature and ADXL345 returns to normal operation.

5.4.6.3 Not-worn Alert

To prevent that seniors might take off WDs for some reason and forget to put it back on, a not-worn detection could be inferred from ADL classification results. BS could simply check if a continuous “Quiet” period is more than 12 hours, which is unlikely to happen if a WD is worn, assuming the WD is not placed on something that moves.

5.4.6.4 Adaptive Duty Cycling

Simply knowing that a WD is “Quiet” offers tremendous potentials for power and bandwidth conservation by means of adaptive reducing duty cycles.

Presence broadcast, transmitted periodically by a WD to all its neighboring REs, have been used extensively in *e-Guardian* for RSSI-based location estimation. While a WD is “Quiet”, which suggests it is stationary and thus not moving. The need for presence broadcast could be completely eliminated. As soon as BS receives WD’s report of its latest ADL status being “Quiet”, BS would stop its check for leave alert and assume the WD stays in the same position throughout the whole “Quiet” period. Meanwhile, its parent RE will also be updated of this ADL status so as not to remove this WD as a stale child WD. Presence broadcast, as generated periodically by WDs, consumes the highest bandwidth among all proposed features. By not sending presence broadcasts during “Quiet” period, a WD saves not only bandwidth but also power consumed in periodic transmissions.

5.5 Additional Uses of Accelerometer

An accelerometer can have a lot of creative applications. ADXL345 has another two built-in interrupt features called SINGLE_TAP and DOUBLE_TAP. As SINGLE_TAP could be inadvertently generated in daily activities, DOUBLE_TAP is considered here. DOUBLE_TAP will be generated if two taps are within a predefined minimum and maximum time window. Each tap has a configurable threshold and duration. In a prototype WD, DOUBLE_TAP could be triggered by patting the WD using the right hand, assuming the WD is worn on the left wrist.

As the only push button in WD is used for distress alerts, DOUBLE_TAP could be used as an alternative to a push button. Having more than one push button in a WD should be avoided as it will be confusing to seniors. A good example of its use is to manually turn on night-light REs in bedrooms proposed in Section 3.4.2.2.5. DOUBLE_TAP is even more advantageous than a push button in this case as it can be easier to perform than a push button in the dark.

5.6 Future Accelerometer Upgrades

As MEMS technologies continuously advance, MEMS sensors with even lower power consumption and/or more features will be released. In Analog Device's MEMS accelerometer family, ADXL346 [71] and ADXL362 [72] chips are more advanced than ADXL345. ADXL346 is almost identical to ADXL345 except for a new interrupt which senses orientation change. ADXL362 was pre-released in June 2012 and is not yet available in the market at the time of this writing.

ADXL362 has significantly better performance than ADXL345. First, its active current at a 100Hz ODR is $1.8\mu A$ and its sleep current is $270nA$. In comparison, ADXL345's active current at 25Hz ODR is $40\mu A$ while its sleep current is slightly more than $23\mu A$. Thus, at any time, ADXL362 consumes less than 1/20 of ADXL345's current. This represents a significant opportunity for designing WDs with even longer battery life.

Another notable improvement in ADXL362 is that it has a deep FIFO which could hold 170 samples, compared to 32 in ADXL345. Assuming ODR is 100Hz and TRIGGER mode is enabled with a trigger level of 150 samples and trigger upon any impact greater than $2g$, it is able to capture 1.5 seconds before an impact (which is a potential fall), the MCU can collect as much data as it wishes after this impact by continuously reading from FIFO. This is a potential for practical implementations of conventional fall detection algorithms, which are based on streaming raw accelerometer data to a remote processing unit. Using ADXL362's deep FIFO, a wearable unit only streams data centered around an impact, thus being able to save bandwidth and transmit power when the wearable unit is static. However, this may still be problematic in a few rare cases in a community setup. For example, a group of seniors gather together and perform certain exercises together, this may lead to many WDs streaming concurrently and the network will be temporarily congested. Nevertheless, this approach will work in a home setup where most likely only one senior is using the bandwidth.

Chapter 6

Test Bedding

6.1 Device Level Testing

6.1.1 Power Consumption of BS

The power consumption of a prototype BS is measured using a digital power supply which provides real-time feedback of current consumption. During the initialization phase when the BS registers to a GSM/GPRS network, the BS consumes up to $1500mA$ on a 3.7V power source. During normal operation with TCP connection on, the current consumption fluctuates between 88mA and 105mA. The prototype BS in fact integrates a specific model, GM862-GPS, in the GM862 GSM/GPRS family. It supports GPS functionality. As discussed in Section 3.4.1.2.4, a BS will occasionally be unplugged and work as a portable GPS tracker. GPS can be highly duty-cycled thus extra power consumption is minimal. At power consumption of somewhere between 88mA and 105mA, a Li-ion battery of 1500mAh could easily power up a BS (after unplugged) for about 15 hours, which is sufficient for a senior who goes out for a whole day.

6.1.2 Analysis of WD's Battery Life

To illustrate that a prototype WD consumes little power, the current consumption of a WD is measured by connecting a 7.5Ω resistor in series with a prototype WD. The polling rate is 5 seconds per poll. The rate of presence broadcasts is 5 seconds per broadcast. Figure 6.1 shows a measurement of power consumption of a WD without movement. It can be clearly seen that two current peaks are spaced by 5 seconds. Each current peak corresponds to CC2430 waking up from sleep mode for polling and broadcasting. The magnitude of the current peak is approximately $4.4 \times 50mV/7.5\Omega = 29.33mA$. The sleep current appears to be noisy and is too small to be visible. Instead, it was measured by a multimeter to be $69\mu A$. Within the $69\mu A$ sleeping current, $40\mu A$ is attributed to the current consumption of ADXL345 at an ODR of 25Hz. The remaining $29\mu A$ is consumed by CC2430 in PM2 mode and its peripherals.

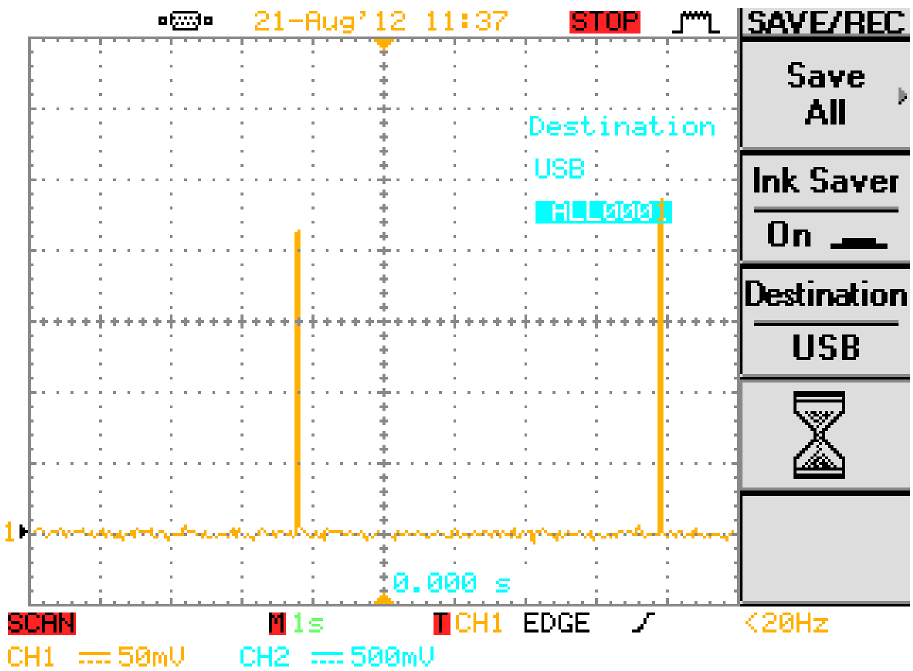

Figure 6.1: Power Consumption Measured by an Oscilloscope

Zooming in to a current peak reveals more details, as shown in Figure 6.2. There are basically three stages involved in each poll and broadcast. First, CC2430 wakes up and starts its 32MHz crystal, which is always needed by the transceiver during transmission and reception. The current consumption goes up. Second, it turns on its radio in receive mode to sense if it is clean for transmission. It then switches to transmit mode and transmit a poll message and a presence broadcast. The current consumption reaches the maximum. Third, after finishing transmission, it waits for and processes data from its parent RE. The current consumption lowers down. Finally, it goes back to sleep mode.

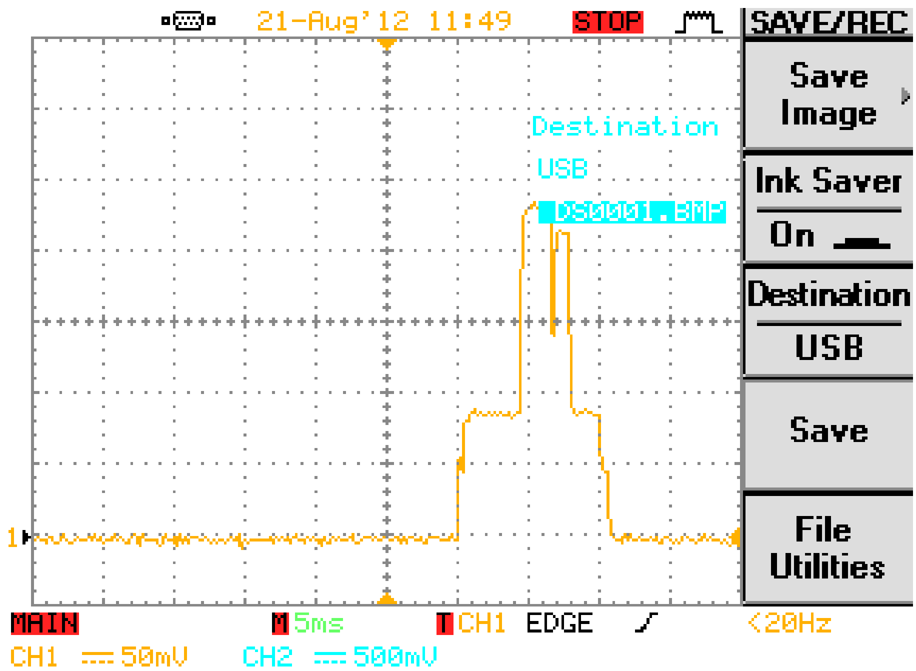

Figure 6.2: Current Peak Measured by an Oscilloscope

Equation 6.1 estimates the power consumption during a single current peak by evaluating the area of the current peak. The bottom of the current peak spans a time frame of approximately 2.1 horizontal divisions. Then the 2.1 division is subdivided into 0.9, 0.5, 0.2 and 0.5. The total area is calculated by summing the four approximately rectangular blocks.

$$
\begin{aligned}
P_{peak} &\approx \frac{(1.7 \times 0.9 + 4.6 \times 0.5 + 4.3 \times 0.2 + 1.7 \times 0.5) \times 50mV \times 5ms}{7.5\Omega} \qquad (6.1) \\
&\approx 184.7mA \cdot ms \\
&= 0.1847mA \cdot s
\end{aligned}
$$

The power consumption during sleep in a 5-second time frame is $69\mu A \times 5s = 0.345mA \cdot s$. Therefore, the total current consumption within 5 seconds is $0.1847 + 0.345 = 0.5297mA \cdot s$. Then, it follows that the current consumption of a day is $0.5297 \times (12 \cdot 60 \cdot 24) = 9153.2mA \cdot s = 2.5426mAh$. Assuming a button cell battery with a nominal capacity of 200 mAh, it could power a WD continuously for $200/2.5426 \approx 78.7$ days.

6.1.3 Indoor Wireless Range

As *e-Guardian* devices are used in indoor environments, it is difficult to measure range in terms of concrete values. During various deployments, *e-Guardian* devices could communicate well within 50 meters with light-of-sight in indoor environments. When there is no line-of-sight with obvious obstacles in between, it will be around 20 to 30 meters.

REs will be typically deployed with good or approximate line-of-sight with each other. With sufficient density of REs, an *e-Guardian* network has no problem to cover every corner of a house or a care center.

6.1.4 Indoor RSSI Evaluation

RSSI values provide meaningful information about the proximity between a WD and an RE (or the BS). A test was carried out in an indoor environment. A human subject wearing a WD moves in steps of one meter away from an RE. Figure 6.3 shows ten RSSI values taken for each d where $d = 1, 2, ..., 10m$. As RSSI values reported by CC2430 are discrete values, RSSI values at the same distance might overlap. This test shows a strong correlation between RSSI values and distances when distances are small. As distances go beyond 8 meters, RSSI values decrease at much smaller rates than those below 8 meters. Most rooms in a household or a care center will be within 8 meters in diagonal. High RSSI values will most likely indicate a WD is close to an RE. Even higher RSSI values will indicate a WD is in the same room with an RE. This implication is important for functionalities such as night-light and automatic voice notifications in BS.

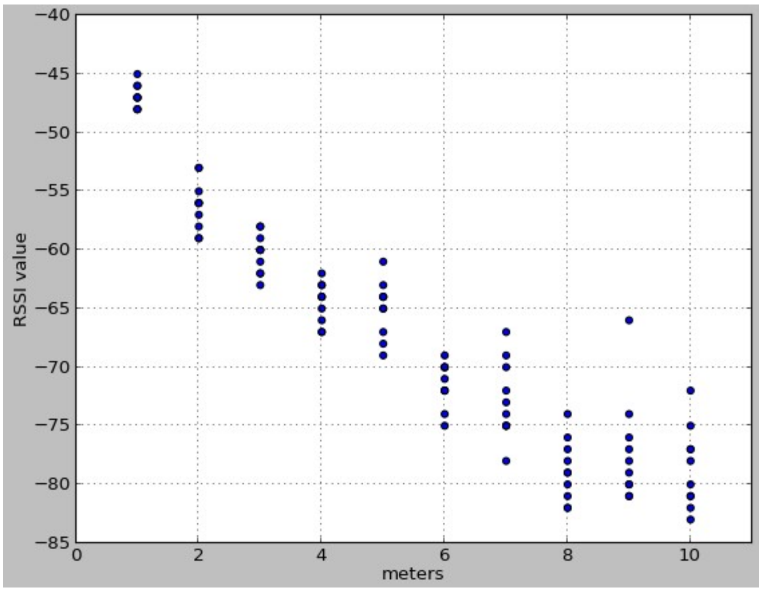

Figure 6.3: RSSI values Measured by a Reference RE

Theoretically, the relationship between RSSI values and distance is described in Equation 6.2 [73].

$$RSSI = -10nlog_{10}d + A \tag{6.2}$$

where n is the signal propagation constant or exponent which is generally more than one depending on the environment, d is the distance from sender, A is the received signal strength at a distance of one meter.

It can be seen that Figure 6.3 reflects this logarithmic relationship. From a practical implementation point of view, to assert if a WD is close (e.g. within two to four meters), a threshold of -65 for RSSI values averaged over three RSSI samples will suffice.

6.2 Community Test Bedding at NTUC Eldercare

A trial study has been carried out for two main objectives: robustness and fall detection accuracies. A set of *e-Guardian* prototype devices have been deployed at Jurong Central Daycare Centre. This care center is a subsidiary of NTUC Eldercare and is a typical eldercare center in Singapore. It takes care of seniors from 9 a.m. to 6 p.m., Monday to Friday, when seniors' children are busy at work.

6.2.1 Motivation

As one of the fastest aging populations in Asia, Singapore faces the challenge of developing public policies to accommodate this changing age structure. NTUC Eldercare was founded in 1997 as the government's effort to provide quality and affordable eldercare services for seniors in the community. NTUC has two types of services: center-based day care program called "Silver Circle" and home-based service called "Care@home". Jurong Central Daycare Centre is one of its eight Silver Circle centers in Singapore. The daycare program is meant to engage seniors in meaningful and interactive activities during the day while their children are at work.

At NTUC Eldercare, each daycare center accommodates about 30 or more seniors and about 4 to 5 caregivers are recruited to look after these seniors. Among seniors, some have dementia; some are in recovery phases after stroke; some have cardiovascular diseases; some are frail in walking, etc. Caregivers are constantly busy with all kinds of tasks including: taking attendance, serving food, assisting frail seniors to walk to toilets and back, helping seniors after stroke with their recovery exercises, helping senior patients take medications, and watching out for dementia patients as they may leave unexpectedly. Currently, such tasks are managed manually. Occasionally, scheduled tasks such as medications could be forgotten; some seniors' needs might not be heard as caregivers were busy taking care of other seniors elsewhere; and dementia patients may leave unnoticed.

From discussions with NTUC Eldercare staff, currently there is no electronic system which could help them systematically keep track of seniors' attendance, daily routine checks (such as medications, recovery exercises), accidental falls, unexpected leaves. *e-Guardian* is well-suited for such need. Also, when seniors need help, they could press a help button which they otherwise have to voice out loud.

The initiative of this trial study is to test bed *e-Guardian*'s robustness in a real eldercare center. *e-Guardian* will be beta-tested in the daycare center and then gradually integrated into their "Care@home" services.

The *e-Guardian* prototype deployed at the care center has support for distress alert, leave alert, medication reminder, and fall detection.

6.2.2 Deployment

The eldercare center features a long three-room flat occupying an area of about $50m \times 10m$, as shown in Figure 6.4. One BS and three REs were deployed. Three REs are labeled as "rest", "act" and "dine", corresponding to their locations in the rest room, the activity area and the dining room respectively. The "rest" RE and the "act" RE could communicate with the BS directly, while the "dine" RE communicates with the BS indirectly via the "act" RE as the BS is too far away from it and there is no direct line-of-sight.

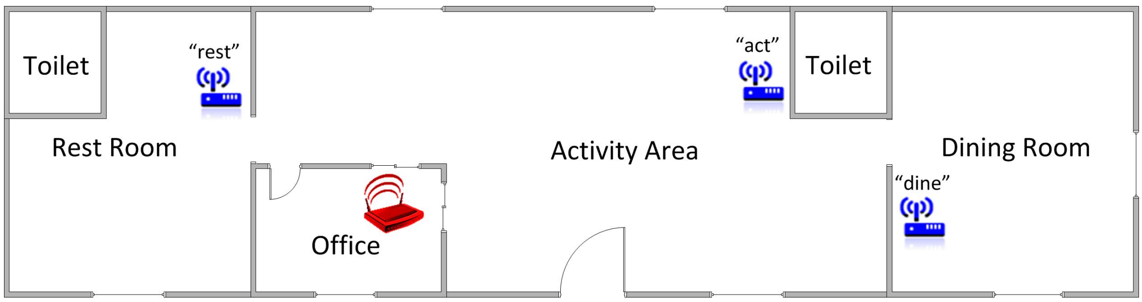

Figure 6.4: Deployment of one BS and three REs in Jurong Central Daycare Centre

Figure 6.5 shows the topology of the network deployed at the eldercare center, captured by a sensor network analyzer in the activity room. It can be seen that the "dine" RE connects with the "act" RE. Also, three WDs exist in the network. One of them is attached to the "dine" RE and the other two are attached to "act" RE.

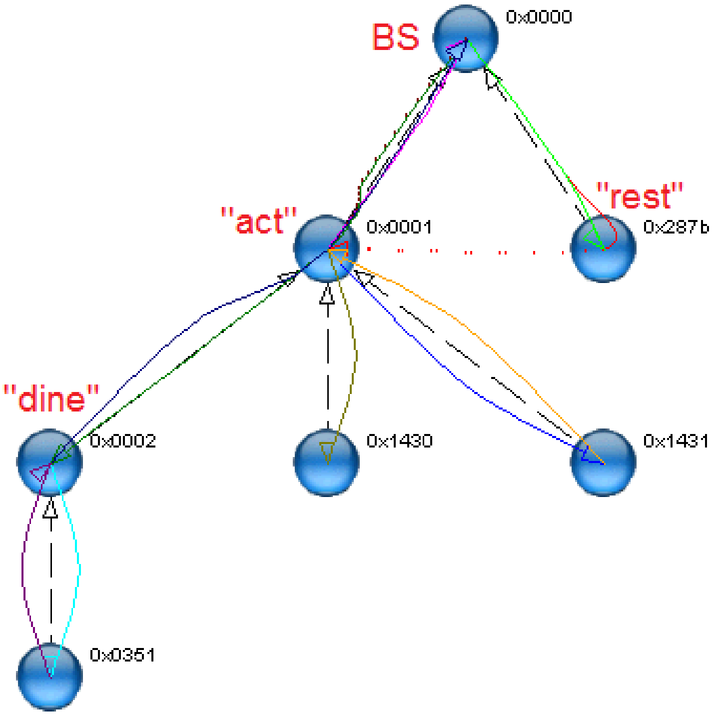

Figure 6.5: Topology of the Deployed Network

The trial started from 9th July to 9th August, 2012. There are altogether 24 week days during the whole trial period. Three seniors were willing to participate in this study. The study also obtained consents from their family members. These three senior volunteers wear WDs when they arrive at 9 a.m. and take them off before they leave at 6 p.m..

The *e-Guardian* system in the trial records basically three types of events: (1) a distress alert, (2)a leave alert, (3)a fall.

6.2.3 Standby Hours

All three WDs are powered by Li-ion batteries with a nominal capacity of $150mAh$. There were all fully charged before the trial study. All lasted a full month without

(a) BS: “BASE” (b) RE: “rest”

(c) RE: “act” (d) RE: “dine”

Figure 6.6: Installations of BS and REs for the Trial Study

needing recharge.

6.2.4 Distress & Leave Alert

Throughout the trial study, except for a few distress alerts generated during installation for testing purposes, there has been no real distress alert and leave alerts generated.

6.2.5 Fall Detection Result

During the period of this trial study, there was no real fall. Capturing real-world falls are challenging. In year 2011, only two fall cases were identified in this care center. However, this study still provides important information about the fall detection algorithm - false positives.

A total number of 83 false positives were registered. This means that, on average, each WD has a false positive rate of $83/(24 \times 3) \approx 1.153$ times per day. For comparison, the best false positive rate in all fall detection algorithms for measurement at torso is 0.6 false positive per day, as evaluated by Bagala et al. [27]. If it is further assumed that a senior is awake for 18 hours a day. The false positive rates of the simulated study will be $1.153 \times 2 = 2.306$ times per day. Considering that wrist is much more difficult to study than torso, this rate of false positives is encouraging.

The false positive rates have shown to be related to the time of a day. Figure 6.7 shows a histogram plot of false positives. Most false positives occurred in two time intervals $[10, 12)$ a.m. and $[1, 4)$ p.m.. These two time intervals correspond to their leisure time when they have fewer activities. During breakfast(9:30 a.m.), lunch(12 noon) and afternoon tea(4 p.m.), false positives of fall detection are less likely to occur because they are more active.

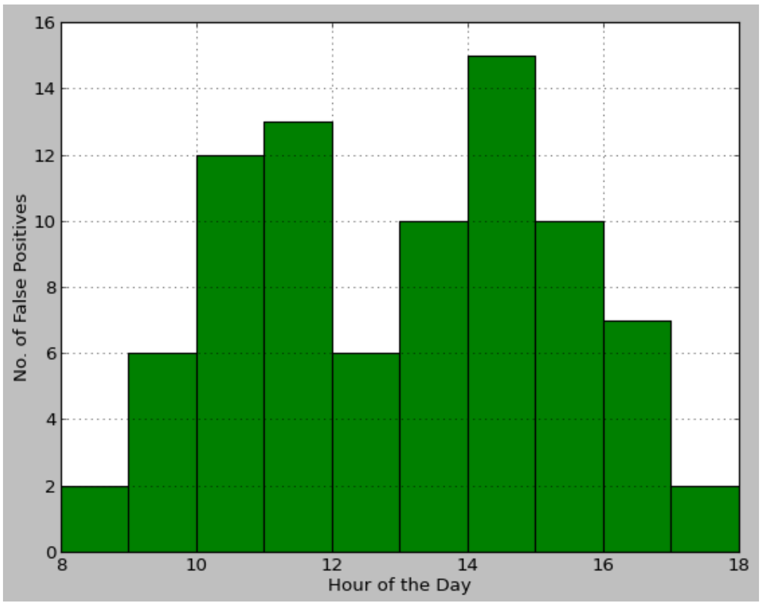

Figure 6.7: Total False Positives in the Trial

The nearest REs during falls have also been identified, as shown in Figure 6.8. It can be seen that none of false positive falls occurred around in the rest room. The rest room is for them to take a short nap but very few of them actually use it. In fact, most seniors spend the majority of their time in the activity area where both the BS and the "act" RE are visible to the seniors' WDs.

Figure 6.8 showed a strong correlation between time and location. The majority of them occurred around the BS and the "act" RE. The rest occurred in the dining room, especially during the lunch time.

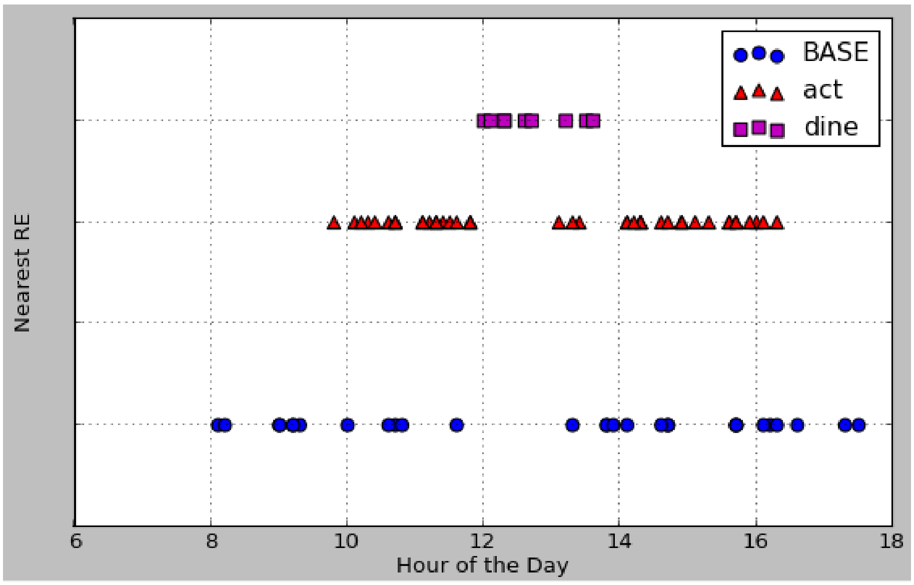

Figure 6.8: Nearest REs of a Detected Fall

6.3 Evaluation of Fall Detection Accuracies

6.3.1 Simulated Falls

As it is difficult to obtain real-world falls, tests of the proposed fall detection algorithm are simulated by four young subjects. Each of them wears a WD on his/her left wrist and performs 4 types of activities: (1) walking, (2) walking upstairs, (3) walking downstairs, (4) walking from distance away, sitting down in a chair beside a desk and doing things such as writing, picking up and putting down random objects. Each person performs each activity five times for two minutes each time. Then activities (1), (2), (3) are performed again for five times, but each with a fall in the middle. Test results are shown in Table 6.1.

Sensitivity (percentage of falls correctly detected as falls) and specificity (percentage of ADLs correctly identified as non-falls) can be calculated from Table 6.1 as $(19 + 19 +$

Table 6.1: Simulated Fall Results

Activity type	With a fall	Falls detected
(1)	yes	19/20
(1)	no	0/20
(2)	yes	19/20
(2)	no	0/20
(3)	yes	17/20
(3)	no	0/20
(4)	no	3/20

$17)/60 = 91.7\%$ and $1 - (0 + 0 + 0 + 3)/80 = 96.25\%$ respectively.

It is observed that during continuous walking, falls will never be falsely triggered as INACTIVITY interrupt will not have a chance to assert. While sitting down, wrists occasionally hit the desk with high impacts, a false positive fall will be accidentally triggered.

6.3.2 Comparisons with Conventional Algorithms

A direct comparison between the proposed algorithm and conventional algorithms is difficult due to different sensor placement. As discussed in Section and Section 5.3.4, placement at torso will lead to better results than wrist. However, due to the lack of fall detection algorithms with accelerometers on wrist in the literature, the proposed algorithm will use conventional algorithms as benchmarks for comparison.

Bagala et al. have done a thorough assessment of the major accelerometer-based fall detection algorithms in the literature. All algorithms reviewed are conventional fall detection algorithms based on accelerometers placed at torso. They were able to collect a total of 29 sets of real-world fall data. They stored accelerometer data on SD cards and re-implemented all algorithms in MATLAB. The algorithms evaluated mostly considered a few characterstics during a fall such as (1)start of a fall by monitoring the root sum vector (SV) of three axes lower than a threshold, (2)impact of a fall by monitoring SV greater than a threshold, (3)change in orientation, (4)posture before and/or after a fall, (5)timeout intervals between these events, and even (6)velocity of the fall by integrating the area of SV from the smallest value (i.e. free fall) to the highest point (i.e. impact). Although different algorithms might take different combinations of these characterstics, the strategies of all accelerometer-based fall detection algorithms are somewhat similar as they all try to find an initial free fall period, shortly followed by an impact, then some algorithms will check if there is orientation change or suspicious posture (as postures are useful if accelerometers are attached to torso).

Among all the 13 algorithms (some are the same algorithm with different parameters) studied, the best result is achieved by the third algorithm of Bourke et al. [65]. This algorithm, denoted as Bourke3, claimed by the original authors to have 100% sensitivity and 100% specificity and was tested by Bagala et al. using real-world fall data as 83% sensitivity and 97% specificity actually. As the author of the thesis lacks access to the real-world fall data, it is only possible to compare the simulated results indirectly with Bagala's real-world results. It can be seen that the simulated results have slightly better sensitivity and slightly worse specificity than the best results of the real-world tests. However, as Bagala et al. studied that sensitivity and specificity of simulated falls would tend to be better than real-world falls as parameters of algorithms were calibrated for test subjects and simulated falls cannot consider all types of falls, the actual sensitivity and specificity of the proposed algorithm would usually be less. Thus, there is still a need for carrying out significantly longer trial studies in future, possibly for a whole year, so as to obtain real-world fall analysis.

6.3.3 Practical Work-Around

As it is difficult to avoid false positives, a mechanism is needed to allow a senior to manually disable false fall alerts. A grant period of 30 seconds could be employed. Once a fall is detected, the buzzer will go off. If the senior pats the WD (as discussed in Section 5.5) within the grant period, the fall alert will be canceled. The distress alert button should not be used for this purpose for two reasons: (1) it is confusing to seniors, and (2) whenever possible, seniors are encouraged to press the distress alert button during an emergency, even if an alert has been triggered automatically.

6.4 Evaluation of ADL Classification Accuracies

Section 5.4.5 shows the classification result using training data. To test the ADL classifier thoroughly, additional six groups of ADL data have been collected from three male subjects and three female subjects, respectively. Each subject performed 4 minutes of "walk", 4 minutes of "quiet" and 5 minutes of "random". The testing data has been labeled. The classifier used is based on parameters obtained in Figure 5.9, the same ones used in the training data.

It can be seen that, the ADL classifier has been able to accurately classify most "walk"

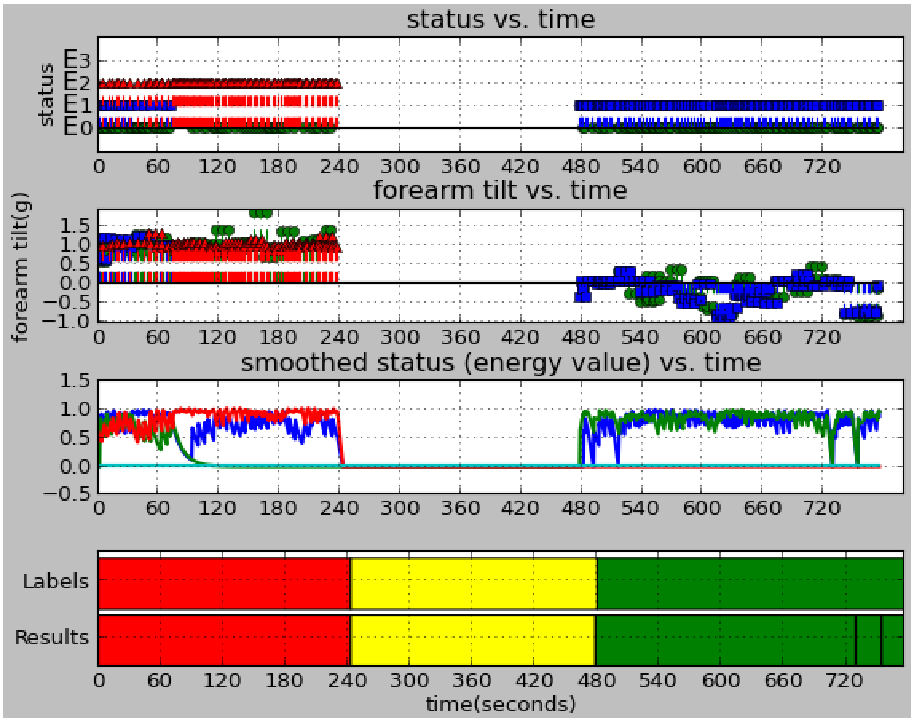

Figure 6.9: ADL Classification of Male Subject One

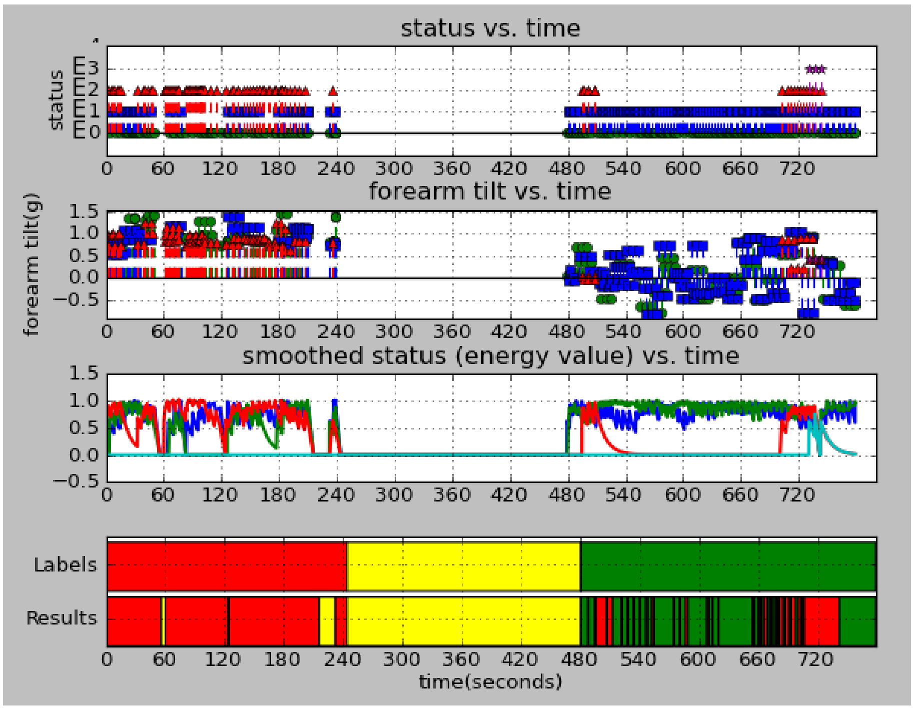

Figure 6.10: ADL Classification of Male Subject Two

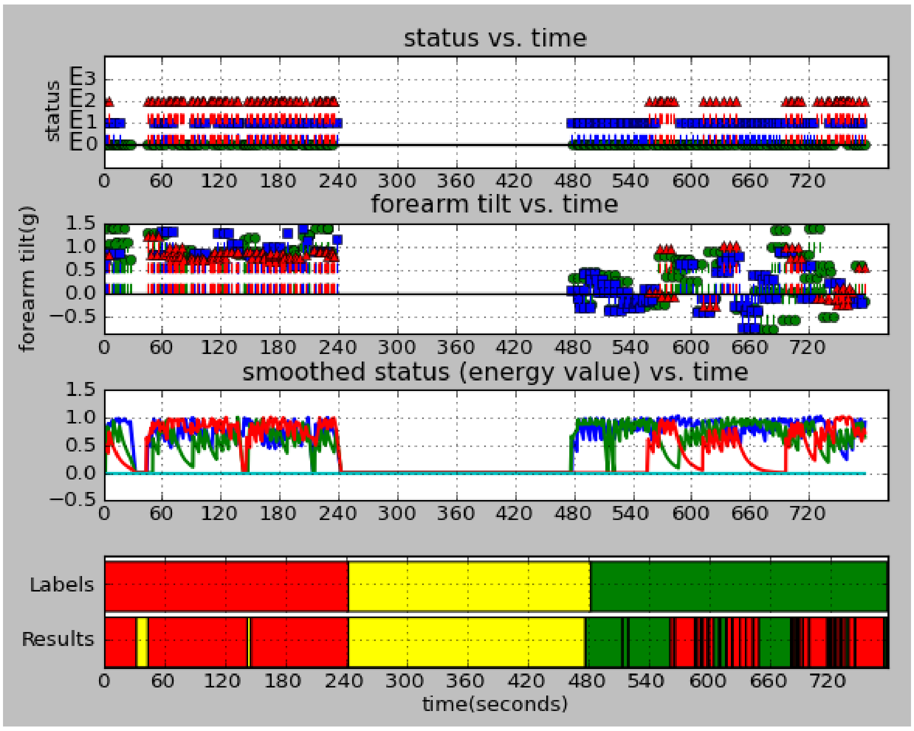

Figure 6.11: ADL Classification of Male Subject Three

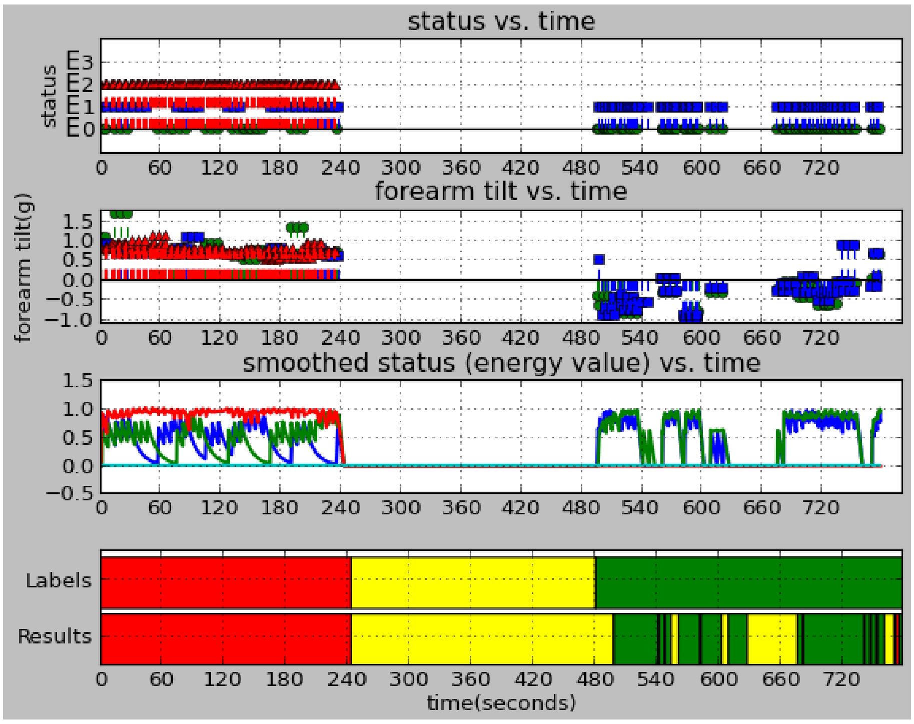

Figure 6.12: ADL Classification of Female Subject One

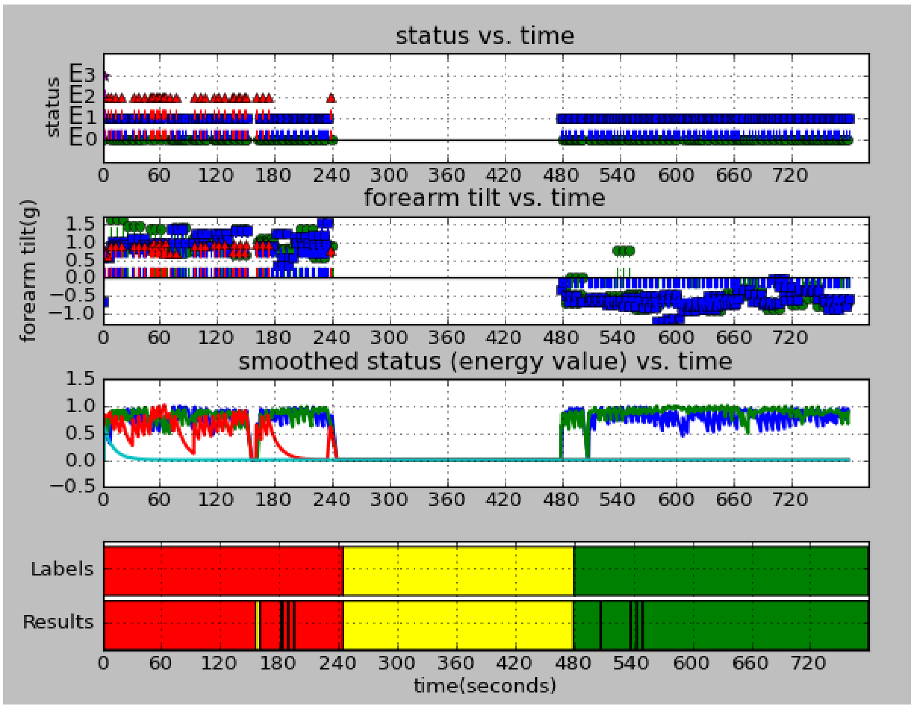

Figure 6.13: ADL Classification of Female Subject Two

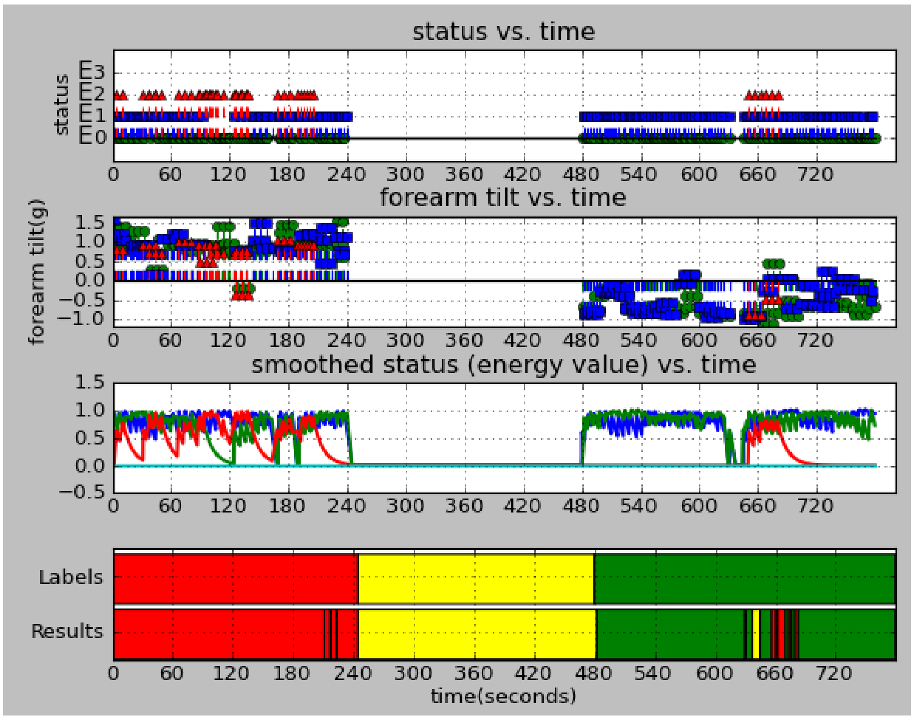

Figure 6.14: ADL Classification of Female Subject Three

and "quiet" activities of the six test subjects because these activities have deterministic behaviors. However, as "Random" is used to collectively refer to all other irregular activities, its accuracy is very much dependent on how fierce an activity is. Figure 6.10 and Figure 6.11 show that the two male subjects' wrists registered a lot of $E2$ events (i.e. FREE_FALL interrupts) during their "random" activities. Those have been mistakenly classified as "walk" by the classifier. Even if so, there is still a significant difference between these misclassified "walk" activities and the real ones. Real "walk" tends to be more consecutive than false "walk" activities due to irregular movements. Thus, higher level analysis can choose to eliminate those short segments of "walk" activities.

Note that, in Figure 6.12, the female subject one pauses a few times in between during the "random" phase. As a result, the test labels for those small time segments are actually incorrect. Nevertheless, the ADL classifier is still able to classify them correctly as "quiet" activities.

Chapter 7

Conclusions

7.1 Summary of Contributions

In this thesis, a simple, low-cost, low-power and scalable health monitoring system, *e-Guardian*, for the elderly has been proposed, implemented and undergone a trial study at NTUC Eldercare. It provides a fundamental infrastructure for scalable remote health monitoring so more functionalities can be incorporated in future. Various functionalities have been proposed which can be easily built on top of the implemented features. For instance, even though features like danger zone alert, automatic night-light are not implemented, they could be easily realized by using presence broadcasts which have already been implemented in the prototype.

The fall detection algorithm and the ADL classifier are based on a digital MEMS accelerometer which supports interrupts and data buffering. These interrupt/event-driven algorithms are hardware-dependent but are significantly more power efficient than continuously polling and processing with an analog accelerometer. As MEMS technologies continue to advance, more sensors could be integrated into MEMS chips, and more

MEMS sensors will be digital and have advanced functionalities such as interrupts, filters and built-in DSP (Digital Signal Processors). As ubiquitous computing becomes more common, more interrupt/event-driven sensor algorithms will emerge to keep up with people's growing desire for smaller size and longer battery life of portable electronics.

Sensor algorithms are processed locally within WDs so no massive data is streamed to the local *e-Guardian* network. Data in the network is mostly high level information, such as alerts, presence broadcasts, commands and ADL results. High level information is usually small in size and also intermittent. Thus, a single *e-Guardian* network could accommodate hundreds of WDs. The total amount of data generated by these WDs will not cause network congestions when aggregated at BS. Therefore, *e-Guardian* is a highly scalable health monitoring system.

e-Guardian uses inexpensive wireless modules and MEMS sensors. A single *e-Guardian* system, together with SMS and GPRS data subscription, can be shared by hundreds of seniors in the same community or care center. This makes the cost of using an *e-Guardian* WD approaching the non-recurring manufacturing cost of a WD. Therefore, *e-Guardian* can be easily adopted by the general public, especially in the less-developed regions.

7.2 Future Work

7.2.1 Dedicated Processor for BS

Currently, the main processing unit in BS is GM862, as described in Section 4.2. GM862 allows programming in Python. However, it does not support interrupts or multi-threads. When CC2430 sends data to GM862, data is stored in GM862's 4096-byte serial buffer. GM862 only fetches data from this buffer when it finishes processing the rest. This will not create problems when the number of WDs is only a few tens.

However, in a large care center where hundreds of WDs may exist in one system , not only will GM862 be unable to process data for hundreds of WDs quickly but there will be more time gap between consecutive reads of the serial buffer. The serial buffer will eventually overflow and data from CC2430 will be lost.

A more powerful processor such as a DSP or an ARM (Advanced RISC Machine) processor could be inserted between GM862 and CC2430. In this new design, GM862 becomes a typical GSM/GPRS module which is only instructed by the powerful processor to send/receive SMS, GPRS data, etc.

7.2.2 Upgrade of Accelerometer

ADXL345 will be replaced by the more power-efficient accelerometer – ADXL362. As discussed in Section 5.6, ADXL362 has a deeper FIFO. However, it lacks an important interrupt feature that ADXL345 has, which is FREE_FALL. In ADXL362, FREE_FALL interrupt can only be realized by using INACTIVITY interrupt. In ADXL345, IN-

ACTIVITY has the smallest time window of 1 second, which could not be used for FREE_FALL interrupt which normally lasts less than a second. In ADXL362, INACTIVITY interrupt could be set to a much smaller time window, which overlaps the functionality of FREE_FALL. Thus, in ADXL362's design, it is removed.

The lack of FREE_FALL interrupt does not affect the proposed fall detection algorithm as in state $F0$, only FREE_FALL interrupt is needed to bring it to state $F1$, which could be implemented by using INACTIVITY interrupt in ADXL362. However, the ADL classifier will be affected, as it needs both FREE_FALL and INACTIVITY enabled in state $F0$. Thus, changes on the ADL classifier will be needed in order to upgrade to ADXL362 and take advantage of its ultra low power feature.

7.2.3 Improvement of the Fall Detection Algorithm

As falls are the major barriers for aging in place, the fall detection algorithm will remain the focus of further improvements in *e-Guardian*.

The proposed fall detection algorithm is very power efficient. However, as real-world falls do not occur frequently on testing subjects, its accuracy could only be evaluated using simulated falls. Also, as the algorithm is interrupt-driven instead of streaming data to a remote processing unit, accelerometer data around a fall event is not kept for further analysis. In future studies, for research purposes, accelerometer data around a fall event will be kept, possibly using ADXL362's deep FIFO, and only that portion of data will be streamed into the network for further analysis. Doing it this way will allow a WD to be worn on human subjects for several months while still be able to collect raw accelerometer data around falls. This will provide more accelerometer information to study the reasons for false positives, true negatives, etc.

7.2.4 Body Temperature Monitoring in a Wrist-worn WD

Some preliminary research has been carried out by the author to monitor body temperature on wrists. Taking skin temperature on wrists is comfortable but inaccurate in reflecting true body temperatures because wrist temperature is more easily affected by the environment temperature than oral or armpit temperatures.

However, the real intention of monitoring body temperature is not to obtain core body temperatures but to know if seniors' temperature level is normal. A heuristic method is to take skin temperature by integrating a medical grade infra-red thermometer in the wrist band of a wrist-worn WD. The infra-red thermometer is mounted inwards to measure temperature of wrist. The preliminary exploration used an infra-red thermometer – MLX90615 from Melexis which has a medical accuracy of 0.1C. It has a diameter of 4.7 mm and a thickness of 3.2 mm, which is suitable for integrating into a wrist band. By incorporating another environment temperature sensor on the surface of the watch, it is able to get a pair of readings: wrist skin temperature and environment temperature. Both can be sampled with a low duty cycle, e.g. once every minute. Another two features are considered to compensate temperature variations due to (1) ADL in the past few minutes, and (2) circadian rhythms (i.e. diurnal temperature variation). Machine learning algorithms can be explored to see if a feature space could be defined adaptively for normal wrist temperature for each individual senior.

Bibliography

[1] Andrew Kochera, Audrey Straight, and Thomas Guterbock. Beyong 50.05 a report to the national livable communities: Creating environments for successful aging. *AARP The power to make it better*, page 48, May 2005.

[2] World Telecommunication/ICT Development Report 2012, Monitoring The WSIS Targets A Mid-term Review, 2010.

[3] Ageing. `www.wpro.who.int/mediacentre/factsheets/fs_201203_ageing/en/index.html`.

[4] Population Ageing and Development 2009, 2009. Department of Economic and Social Affairs, United Nations.

[5] Our ageing world. `www.who.int/ageing/en/index.html`.

[6] R J. Hodes. Overcoming barriers: human mobility and development. *National Institute on Aging*, page 3, Apr 2005.

[7] Lifeline with autoalert option. `www.lifelinesys.com/content/lifeline-products/auto-alert`.

[8] Terrance J. Dishongh and Michael McGrath. *Wireless Sensor Networks for Healthcare Applications*. Artech House, November 2009.

[9] K. S. Short. Who is poor? a new look with the supplemental poverty measure. *Session: A New Supplemental Poverty Measure for the U.S.*, page 22, 2011.

[10] Medical Protection. `www.lifealert.com/medical.aspx`.

[11] Rescue Alert Medical Alert System. `www.rescuealert.com`.

[12] Product review: Philips lifeline with auto alert. `www.theseniorlist.com/2011/10/product-review-philips-lifeline-with-auto-alert`.

[13] What range can the myhalo device cover? `kb.myhalomonitor.com/index.php?View=entry&EntryID=19`.

[14] Halo Monitoring Independence Redefined. `www.myhalomonitor.com/order`.

[15] Philips lifeline. `www.lifelinesys.com/content/home`.

[16] Kenneth Holmberg et al. *E-maintainence*. Springer, 2010.

[17] Mems sensor generations. `en.wikipedia.org/wiki/MEMS_sensor_generations`.

[18] MPU-6000/6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking^{TM} Devices. `www.invensense.com/mems/gyro/mpu6050.html`.

[19] Tinyos. `www.tinyos.net`.

[20] Contiki. `www.sics.se/contiki`.

[21] Urgentys. `www.medicalintelligence.ca`.

[22] F. Sposaro and G. Tyson. ifall: An android application for fall monitoring and response. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 6119 –6122, sept. 2009.

[23] Gokhan Remzi Yavuz, Mustafa Eray Kocak, Gokberk Ergun, E Alemdar, Hulya Yalcin, Ozlem Durmaz Incel, Lale Akarun, and Cem Ersoy. A smartphone based fall detector with online location support.

[24] Marco Benocci, Carlo Tacconi, Elisabetta Farella, Luca Benini, Lorenzo Chiari, and Laura Vanzago. Accelerometer-based fall detection using optimized zigbee data streaming. *Microelectronics Journal*, 41(11):703 – 710, 2010. IEEE International Workshop on Advances in Sensors and Interfaces 2009.

[25] Thuy-Trang Nguyen, Myeong-Chan Cho, and Tae-Soo Lee. Automatic fall detection using wearable biomedical signal measurement terminal. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 5203 –5206, sept. 2009.

[26] Pepijn Ven, Alan Bourke, John Nelson, and Hugh OBrien. Design and integration of fall and mobility monitors in health monitoring platforms. In Aim Lay-Ekuakille and Subhas Chandra Mukhopadhyay, editors, *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment*, volume 75 of *Lecture Notes in*

Electrical Engineering, pages 1–29. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-15687-8_1.

[27] Fabio Bagala, Clemens Becker, Angelo Cappello, Lorenzo Chiari, Kamiar Aminian, Jeffrey M. Hausdorff, Wiebren Zijlstra, and Jochen Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLoS ONE*, 7(5):e37062, 05 2012.

[28] A.K. Bourke et al. Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities. *Journal of biomechanics*, 10 2010.

[29] M.R. Narayanan, S.R. Lord, M.M. Budge, B.G. Celler, and N.H. Lovell. Falls management: Detection and prevention, using a waist-mounted triaxial accelerometer. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4037 –4040, aug. 2007.

[30] Min-Seok Lee, Jong-Gwan Lim, Ki-Ru Park, and Dong-Soo Kwon. Unsupervised clustering for abnormality detection based on the tri-axial accelerometer. In *ICCAS-SICE, 2009*, pages 134 –137, aug. 2009.

[31] A. Lombardi, M. Ferri, G. Rescio, M. Grassi, and P. Malcovati. Wearable wireless accelerometer with embedded fall-detection logic for multi-sensor ambient assisted living applications. In *Sensors, 2009 IEEE*, pages 1967 –1970, oct. 2009.

[32] Maarit Kangas, Antti Konttila, Per Lindgren, Ilkka Winblad, and Timo Jamsa. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & Posture*, 28:285–291, 08 2008.

[33] E. Murugasu M.N. Nyan, Francis E.H. Tay. A wearable system for pre-impact fall detection. *Journal of biomechanics*, 41(16):3475–3481, 12 2008.

[34] Qiang Li, J.A. Stankovic, M.A. Hanson, A.T. Barth, J. Lach, and Gang 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*, pages 138 –143, june 2009.

[35] Wan-Young Chung, S. Bhardwaj, A. Purwar, Dae-Seok Lee, and R. Myllylae. A fusion health monitoring using ecg and accelerometer sensors for elderly persons at home. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 3818 –3821, aug. 2007.

[36] D. Curone, A. Tognetti, E.L. Secco, G. Anania, N. Carbonaro, D. De Rossi, and G. Magenes. Heart rate and accelerometer data fusion for activity assessment of rescuers during emergency interventions. *Information Technology in Biomedicine, IEEE Transactions on*, 14(3):702 –710, may 2010.

[37] I.C. Gyllensten and A.G. Bonomi. Identifying types of physical activity with a single accelerometer: Evaluating laboratory-trained algorithms in daily life. *Biomedical Engineering, IEEE Transactions on*, 58(9):2656 –2663, sept. 2011.

[38] Yu-Jin Hong, Ig-Jae Kim, Sang Chul Ahn, and Hyoung-Gon Kim. Mobile health monitoring system based on activity recognition using accelerometer. *Simulation Modelling Practice and Theory*, 18(4):446 – 455, 2010. Modeling and Simulation Techniques for Future Generation Communication Networks.

[39] Sung-Ihk Yang and Sung-Bae Cho. Recognizing human activities from accelerometer and physiological sensors. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 100 –105, aug. 2008.

[40] Open source accelerometers for phones. `http://web.mit.edu/wockets/index.html`.

[41] Reza Naima and John F. Canny. The Berkeley Tricorder: wireless health monitoring. *Wireless Health 2010*, pages 212–213, October 2010.

[42] D. Malan, T. F. Jones, M. Welsh, and S. Moulton. CodeBlue: an ad hoc sensor network infrastructure for emergency medical care. *International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.

[43] C.H.S. Gibson, T.L.M. van Kasteren, and B.J.A. Krose. Monitoring homes with wireless sensor networks. In *In Proceedings of the International Med-e-Tel Conference*, 2008.

[44] Youngbum Lee, Jinkwon Kim, Muntak Son, and Myoungho Lee. Implementation of accelerometer sensor module and fall detection monitoring system based on wireless

sensor network. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 2315 –2318, aug. 2007.

[45] R. Ramachandran et al. Body sensor networks to evaluate standing balance: Interpreting muscular activities based on inertial sensors. In *HealthNet 08 Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, Breckenridge, CO, USA, June 2008.

[46] C.R. Baker, K. Armijo, S. Belka, M. Benhabib, V. Bhargava, N. Burkhart, A. Der Minassians, G. Dervisoglu, L. Gutnik, M.B. Haick, C. Ho, M. Koplow, J. Mangold, S. Robinson, M. Rosa, M. Schwartz, C. Sims, H. Stoffregen, A. Waterbury, E.S. Leland, T. Pering, and P.K. Wright. Wireless sensor networks for home health care. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, volume 2, pages 832 –837, may 2007.

[47] R.S. Dilmaghani, H. Bobarshad, M. Ghavami, S. Choobkar, and C. Wolfe. Wireless sensor networks for monitoring physiological signals of multiple patients. *Biomedical Circuits and Systems, IEEE Transactions on*, 5(4):347 –356, aug. 2011.

[48] G. Virone, A. Wood, L. Selavo, et al. An advanced wireless sensor network for health monitoring. In *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare D2H2*, pages 2–5, 2006.

[49] R. Jafari, A. Encarnacao, A. Zahoory, F. Dabiri, H. Noshadi, and M. Sarrafzadeh. Wireless sensor networks for health monitoring. In *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, pages 479 – 481, july 2005.

[50] D. Gislason. *ZigBee Wireless Networking*. IT Pro. Elsevier, 2008.

[51] Matt Van Wieringen and J. Mikael Eklund. Real-time signal processing of accelerometer data for wearable medical patient monitoring devices. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 2397 –2400, aug. 2008.

[52] Mars Lan, Ani Nahapetian, Alireza Vahdatpour, Lawrence Au, William Kaiser, and Majid Sarrafzadeh. SmartFall: An Automatic Fall Detection System Based on Subsequence Matching for the SmartCane. In *Fourth International Conference on Body Area Networks (BodyNets)*, 2009.

[53] Giovanni Zanca, Francesco Zorzi, Andrea Zanella, and Michele Zorzi. Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks. In *Proceedings of the workshop on Real-world wireless sensor networks*, REALWSN '08, pages 1–5, New York, NY, USA, 2008. ACM.

[54] Masashi Sugano. Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. In *Wireless and Optical Communications*, pages 1–6. IASTED/ACTA Press, 2006.

[55] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej. Using rssi value for distance estimation in wireless sensor networks based on zigbee. In *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 303 –306, june 2008.

[56] GSM GPRS Product Range. Technical report, Telit Wireless Solutions Making Machines Talk. www.telit.com/en/products/gsm-gprs.php.

[57] CC2430 System-on-Chip Solution for 2.4 Ghz IEEE 802.15.4 / Zigbee. Technical report, Texas Instruments. www.ti.com/product/cc2430.

[58] ADXL345: 3-axis, ±2g/±4g/±8g/±16g digital accelerometer. Technical report, Analog Devices. www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl345/products/product.html.

[59] GM862 cellular quad band module. www.sparkfun.com/products/757.

[60] CC2430F128RTC. www.digikey.com/product-search/en?mpart=CC2430F128RTC&vendor=296.

[61] ADXL345BCCZ-RL. www.digikey.com/product-detail/en/ADXL345BCCZ-RL/ADXL345BCCZ-RLTR-ND/2237748.

[62] S.D. Bersch, C.M.J. Chislett, D. Azzi, R. Khusainov, and J.S. Briggs. Activity detection using frequency analysis and off-the-shelf devices: Fall detection from

accelerometer data. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pages 362 –365, may 2011.

[63] Chankyu Park, Jaehong Kim, and Ho-Jin Choi. A watch-type human activity detector for the aged care. In *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, pages 648 –652, feb. 2012.

[64] T. Degen, H. Jaeckel, M. Rufer, and S. Wyss. Speedy:a fall detector in a wrist watch. In *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 184 – 187, oct. 2003.

[65] J. Ning. Fall detection application by using 3-axis accelerometer ADXL345. *Application Note AN-1023 from Analog Devices*, 2009.

[66] M. Kangas, A. Konttila, I. Winblad, and T. Jamsa. Determination of simple thresholds for accelerometry-based parameters for fall detection. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1367 –1370, aug. 2007.

[67] P. Jantaraprim, P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa. Improving the accuracy of a fall detection algorithm using free fall characteristics. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pages 501 –504, may 2010.

[68] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recogn.*, 43(10):3605–3620, October 2010.

[69] Chankyu Park, Jaehong Kim, and Ho-Jin Choi. A watch-type human activity detector for the aged care. In *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, pages 648 –652, feb. 2012.

[70] D.T. Larose. *Discovering Knowledge in Data: An Introduction to Data Mining.* Wiley, 2004.

[71] ADXL346: 3-axis, ±2g/±4g/±8g/±16g ultralow power digital accelerometer. Technical report, Analog Devices. `www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl346/products/ product.html`.

[72] ADXL362: Micropower, 3-axis, ±2g/±4g/±8g digital output mems accelerometer. Technical report, Analog Devices. `www.analog.com/en/mems-sensors/mems-accelerometers/adxl362/products/product.html`.

[73] K. Aamodt. Cc2431 location engine. Technical report, Texas Instruments.

Publications & Showcases

Journal Papers:

1. J. Yuan, K. K. Tan, T. H. Lee, and C. H. Koh. Development of an Expensive Health Monitoring System for the Lone Elderly. In *Control and Intelligent Systems*, 2012, accepted.

2. J. Yuan, K. K. Tan, T. H. Lee, and C. H. Koh. Power-efficient Interrupt-driven Algorithms for Fall Detection and Classification of Activities of Daily Living. In *IEEE Sensors Journal*, 2012, submitted.

Conference Papers:

1. J. Yuan, Y. Zhang, K. K. Tan, and T. H. Lee. Text extraction from images captured via mobile and digital devices. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2009. AIM 2009*, vol., no., pp.566-571, 14-17 July 2009.

2. J. Yuan, K. K. Tan, and T. H. Lee. Robust Approach towards Text Extraction from Natural Scene Images Captured via Mobile Devices. In *Proceedings of the IASTED International Conference Modeling, Simulation, and Identification*, 12 -

14 October 2009, China

3. J. Yuan, K. K. Tan, and T. H. Lee. Development of an e-Guardian for the Single Elderly or the Chronically-Ill Patients. In *2010 International Conference on Communications and Mobile Computing (CMC)*, vol.3, no., pp.378-382, 12-14 April 2010.

4. J. Yuan, K. K. Tan, K. Z. Tang, and T. H. Lee. Remote Monitoring of Wildlife and Environment via GPS/GPRS. In *Proceedings of the 12th IASTED International Conference Control and Applications*, pp. 614-620, Canada, 15-17 Jul 2010.

5. J. Yuan, K. K. Tan, and T. H. Lee. Development of an inexpensive ubiquitous health monitoring system for the elderly. In *Proceedings of the 2011 Canadian Congress of Applied Mechanics*, pp. 291-294, 5 - 9 June 2011, Canada.

6. J. Yuan, K. K. Tan, C. H. Koh, and T. H. Lee. Remote Monitoring of Wildlife and Environment via GPS/GPRS. In *Proceedings of the 12th IASTED International Conference Control and Applications*, pp. 614-620, Canada, 15-17 Jul 2010.

7. J. Yuan, K. K. Tan, C. H. Koh, and T. H. Lee. A Scalable Remote Health Monitoring System for Seniors in the Community. In *The 7th International Conference for Internet Technology and Secured Transactions*, London, UK, 10-12 Dec 2012, accepted.

8. J. Yuan, P. V. Er, S. Yu, K. K. Tan, C. H. Koh, A. S.Narayanan, and T. H. Lee. Development of an Inexpensive Health Monitoring System for the Lone Elderly. In

The IASTED International Conference on Engineering and Applied Science, Sri Lanka, 27-29 Dec 2012, accepted.

Showcases

1. A trial study was conducted to show robustness and fall detection accuracy of the proposed *e-Guardian* at Jurong Central Daycare Centre (Singapore) from 9 July to 9 Aug, 2012.

2. An *e-Guardian* prototype is being showcased in the iExperience organized by IDA (the Infocomm Development Authority of Singapore) from June to December 2012.