國立交通大學

資訊科學與工程研究所

碩 士 論 文

用智慧型家庭環境中的無線感測網路雛形

A Wireless Sensor Network Prototype for Smart Home Applications

研 究 生：游騰葦

指導教授：易志偉 教授

中 華 民 國 九 十 七 年 三 月

用 智 慧 型 家 庭 環 境 中 的 無 線 感 測 網 路 雛 形
A Wireless Sensor Network Prototype for Smart Home Applications

研 究 生：游騰葦　　Student：Teng-Wei Yu

指導教授：易志偉　　Advisor：Chih-Wei Yi

國 立 交 通 大 學
資 訊 科 學 與 工 程 研 究 所
碩 士 論 文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

March 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年三月

國立交通大學

研究所碩士班

論文口試委員會審定書

本校 資訊科學與工程 研究所 游騰葦 君

所提論文:

用智慧型家庭環境中的無線感測網路雛型

A Wireless Sensor Network Prototype for Smart Home Applications

合於碩士資格水準、業經本委員會評審認可。

口試委員： 呂政修 易志偉

彭文志

指導教授： 易志偉

所　　長： 曾文貴

中華民國九十七年三月四日

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the thesis prepared by Teng-Wei Yu entitled A Wireless Sensor Network Prototype for Smart Home Application

and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Master of Science.

Committee Members:

Jeng-Shiou Leu

Wen-chih Peng

Thesis Advisor:

Director:

Date: 03/04/2008

國立交通大學

博碩士論文全文電子檔著作權授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學資訊科學與工程研究所 ＿＿＿＿＿＿組， 96 學年度第 二 學期取得碩士學位之論文。

論文題目：用智慧型家庭環境中的無線感測網路雛形
指導教授：易志偉

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：

本校及台灣聯合大學系統區域網路	■ 中華民國 100 年 3 月 11 日公開
校外網際網路	■ 中華民國 102 年 3 月 11 日公開

■ **全文電子檔送交國家圖書館**

授 權 人：游騰葦

親筆簽名： 游騰葦

中華民國 97 **年** 3 **月** 11 **日**

國　立　交　通　大　學

博碩士紙本論文著作權授權書

（提供授權人裝訂於全文電子檔授權書之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學資訊科學與工程研究所 ____________組， 96 學年度第 二 學期取得碩士學位之論文。

論文題目：用智慧型家庭環境中的無線感測網路雛形
指導教授：易志偉

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：____________________，請將論文延至____年____月____日再公開。

授 權 人：游騰葦

親筆簽名： 游騰葦

中華民國 97 **年** 3 **月** 11 **日**

用智慧型家庭環境中的無線感測網路雛形

學生：游騰葦　　　指導教授：易志偉

資訊科學與工程研究所

國立交通大學

摘要

ZigBee 是一個低資料傳輸率，低能源消耗，以及低價的無線網路協定，其被用來針對自動化與遠端控制應用。在本論文中的主要動機在越來越興盛的智慧型家庭應用的需求，我們基於使用 Zigbee 技術連接的無線感測網路而發展出一個智慧型家庭環境的雛形。以智慧型家庭環境可以提供居住者一個安全，舒適以及便利的生活方式。本智慧型家庭的平台包含了一群提供服務的感測節點，一個 sink 伺服器，以及一個具有資料庫功能的家用伺服器。這群在家庭環境中的感測節點執行感測環境狀態及控制電器的功能，其達成了居家安全，居家照護以及家用自動化的服務。上傳資料流以及下傳資料流透過一個變更過的 MultiHopLQI 路由協定在 sink 伺服器與無線感測網路間互相傳遞資料。 Sink 伺服器是一個與 PC 連結的專屬的感測節點，可視為無線感測網路與家用伺服器間的閘道器。家用伺服器則含括了一個封包管理系統，一個網頁形式的監視器以及一個提供控制的使用者介面。所感測到的資料儲存於資料庫中並且可以透過網際網路被檢索以及監看。更進一步，為了提供一個輕鬆植入的智慧型家庭平台，我們發展出了

一個簡單的網頁形式智慧型家庭服務程式的開發工具。

關鍵字：無線感測網路、智慧型家庭、居家安全、家用自動化

A Wireless Sensor Network Prototype for Smart Home Applications

Student: Teng-Wei Yu **Advisor: Dr. Chih-Wei Yi**

Institute of Computer Secince and Enginerring

National Chiao Tung University

Abstract

ZigBee technology is a low data rate, low power consumption, and low cost wireless networking protocol designed for automation and remote control applications. In this work, motivated by the blooming demand on smart home application, we develop a smart home environment based on a wireless sensor network connected by ZigBee technology. The smart home environment is able to offer householder a security, comfort and convenient living. The smart home platform includes a group of serving sensor nodes, a sink server, and a home server with database. The serving sensor nodes perform sensing environment and controlling home appliance in the home environment. These sensor nodes accomplish home security, home care and home automation service. The upstream and downstream messages between the sink server and WSNs are delivered via a modified MultiHopLQI routing protocol. The sink server that is a dedicated sensor node attached on a PC's interface is the gateway between WSNs and Home server. The home server contains a packet management subsystem, a web-based monitor and user interface. The sensing data stored in a database can be retrieved and monitored via Internet. Furthermore, in order to provide a easily-deployed smart home platform, a simple web-based development tool is developed for householder user.

Keywords

wireless sensor network, smart home, home security, home automation

Acknowledgements

Many people helped me to finish this thesis and project. I deeply appreciate my advisor, Dr.Chih-Wei Yi, for his intensive advice and instruction. I would like to thank all the classmates in Network Optimization Laboratory. Finally, I want to thank my parents for their endless love and support. This thesis is dedicated to my grandfather.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Wireless Personal Area Networks

WPANs (wireless personal area networks) are wireless networks for interconnecting devices around a personal area. The technology of WPANs permits communication within a very short range – (about 10 -100 meters). WPANs can be served to interconnect the ordinary computing and communication devices or served to more specialized purposes such as communication between spacemen. The technologies which support WPAN are Bluetooth, ZigBee, UWB (Ultra-Wide-Band), IrDA, HomeRF, etc.. Each of them serves as some specialized application and performs tasks in the best situation.

ZigBee: ZigBee is a low data rate, low power consumption, and low cost wireless networking protocol applied for automation and remote control applications [1].The protocol stack of ZigBee is shown in Figure 1.1.

ZigBee adopts physical and medium access control layer defined in the standard IEEE 802.15.4 and further defines network, security, and application framework.. ZigBee Al-

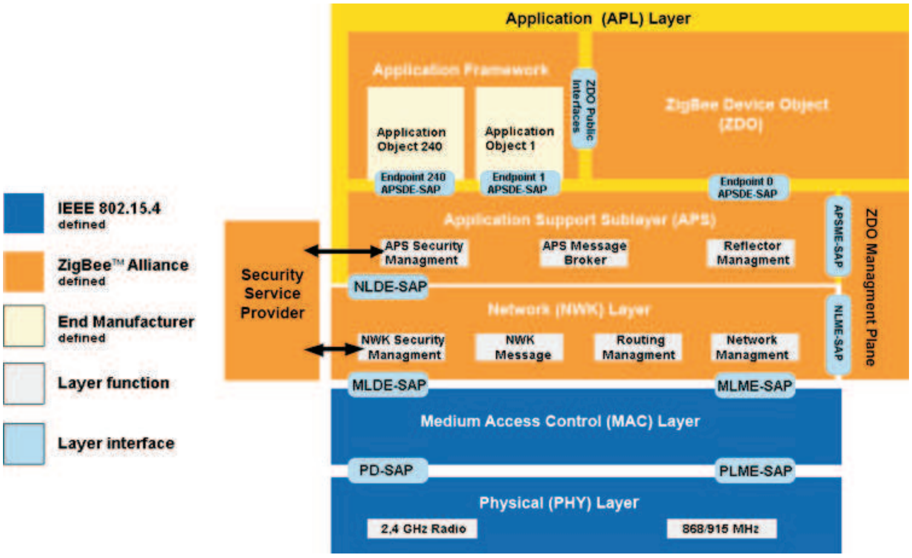

Figure 1.1: The Protocol Stack of Zigbee

liance and IEEE group decide to join forces and ZigBee is the commercial name for this technology.

Smart Home:A typical application of ZigBee technology is the smart home platform. Smart homes utilize home-based networks to connect household appliances and internet to outside world. The smart home environment can response and modify itself continuously according to diverse occupants and changeable needs[13]. ZigBee-embedded smart home environment makes home appliances communicated with each other wireless.

1.2 Motivation

Based on the recent research by ON World, which investigated 100 home installers, vendors, and suppliers, the result indicates that the wireless smart home market is boosting. Hundreds of products currently established service providers are starting to offer Wireless Sensor Network (WSN) based home monitoring services. ON World researcher pointed out that while proprietary WSNs systems have been used by professional installers in luxury homes for over a decade, wireless protocols, such as Z-Wave and ZigBee, will make smart home solutions affordable for the average household.

Therefore, the investigation brang the motivation of our smart home project. A set of ZigBee-ready hardware modules, such as micaz or telosb, is utilized to implement experimental smart home platforms over WSNs. To sense the home environment by using the sensor embedded in the ZigBee-ready hardware module is the major objective of this thesis. Besides, we proposed a web-based development tool that allows householders to build smart home environment by themselves. Furthermore, the prototype of integrated smart home hardware and software sensor node module is developed to serve main home appliances. The sensor node module contains application unit and networking unit. The application unit is responsible for sensing environment, controlling appliances, and monitoring home security. A simple scenario of assumption is shown as follows:

A specialized handheld device is carried by a householder. After installing a designate home server, householder is able to setup the relative program for smart home module onto the handheld device. The handheld device can be used as a electronic key, tracking target, and smart home control panel in the house. In addition, a group of sensor nodes with smart home module are installed for sensing environment, controlling home appliances, and detecting accident. The home server can build the sensor node program and configure the position of all sensor nodes during building period. During running time, the home server will control home appliances and monitor the house environment. The householder can use either the handheld device or the Web-browser to control other home appliances. Furthermore, when some urgent events, such as fire accident and thief intrusion, are taking place, the home server can monitor the process of this accident and notify the householder whether he is at home.

1.3 Thesis Organization

The framework of this thesis is organized as follows. In chapter 2, we present the concept of wireless sensor networks and smart home. In addition, the previous works are discussed in the chapter. In chapter 3, the development environments of this project, such as hardware and software environment description, are proposed. In chapter 4, designs of this project are proposed, including the platform architecture design and smart home services design. The implement result of our test-bed and the conclusions are proposed in chapter 5. In chapter 6, conclusions and future works will be given.

Chapter 2

Related Works

An overview of related works is presented in this chapter. This chapter is separated into two sections. In section 2.1, we will give a general wireless sensor network design principle and architecture, and introduce each layer's function of wireless sensor networks. In section 2.2, we will give a concept of smart home application and introduce some related projects.

2.1 Wireless Sensor Networks

Increasingly, wireless sensor networks (WSNs) are viewed as the key to innovate in many aspects of science and our life. Applications of WSNs ranging from environmental monitoring and business asset management to smart houses and health care automation and are unlimited. WSNs are composed of three fundamental functions, wireless transmission, data processing, and environment sensing. In the following, we'll introduce these three functions.

- **Wireless Transmission**

Generally, the wireless sensor network systems are designed according to IEEE 802.15.4 and ZigBee PHY (Physical Layer) standards. The working frequency bands of wireless sensor network systems are central at 915MHz, 868MHz, and 2.4GHz, as shown in Table 2.1. For 2.4 GHz band, which has 16 channels with a maximum over-the-air data rate at 250 Kbps, is adopted worldwide. Comparatively, 915 MHz band (10 channels and 40 Kbps burst rate) and 868 MHz (1 channel and 10 Kbps burst rate) are utilized in the Americas and most of the Pacific Rim and Europe, respectively [2]. In this study, 2.4GHz is selected as the designed frequency for the advantages of higher date rate, lower latency, and lower duty cycle, comparing to lower frequencies.

For applying WSNs that are composed of large number of low cost nodes to spread out and it is necessary to transmit data from sender to receiver in the multi-hops. Therefore, a mechanism is required to control the path of transmitting data, which is so called routing protocol.

- **Environment Sensing**

With the development of MEMS in the past few years, the technology of sensor device has been improved and many functions can be integrated into a micro node, which can sense some specific environment information, such as temperature, humidity, and acceleration. Besides, most the default sensor nodes equipped with sensing module can collect data from external IO interface in the Tmote Sky, such as ADC (Analog to Digital Converter) and GPIO(General Purpose Input and Output).

PHY	Freq.	Sperading parameters		Data parameters		
		(kchips/s)	Mod.	(kb/s)	(ksym./s)	Symbols(sym.)
868/915	868-868.6	300	BPSK	20	20	Binary
	902-928	600	BPSK	40	40	Binary
868/915	868-868.6	400	ASK	250	12.5	20-bit PSSS
	902-928	1600	ASK	250	50	5-bit PSSS
868/915	868-868.6	400	O-QPSK	100	25	16-ary Orth.
	902-928	1000	O-QPSK	250	62.5	16-ary Orth.
2450	2400-2483.5	2000	O-QPSK	250	62.5	16-ary Orth.

Table 2.1: Frequency band and data rate of 802.15.4

As shown in Figure 2.1, external sensors can interface with mote and pass sensing data via those interfaces played by 10-pin + 6-pin IDC. Besides, sensor developers can attach their sensors to the specific I/O port and then obtain data to convert from analog signals to digital ones. In this research, ADC and GPIO interfaces are applied to acquire information from sensing device.

- **Data Processing**

WSNs systems must operate under uncertainty physical environments, through unreliable communication, and with severe resource constraints such as battery power. These challenges require new and holistic approaches that bring together a variety of disciplines, such as statistical modeling, signal processing, networking, distributed algorithms, probabilistic reasoning, and databases. To develop proper processing approaches either at

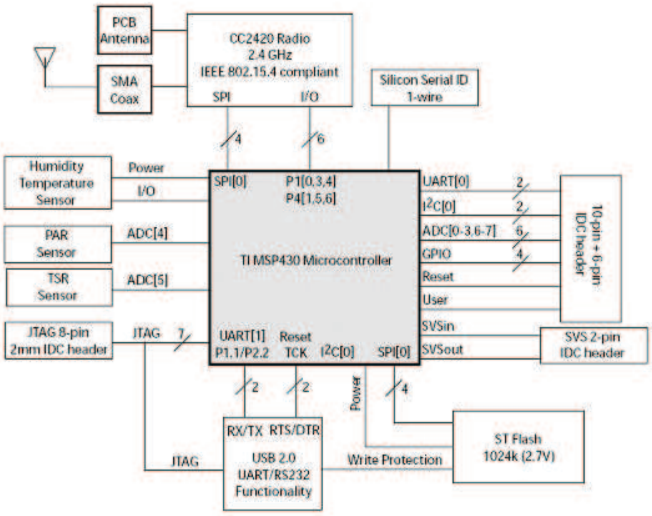

Figure 2.1: Functional block of Taroko hardware module

sensor nodes or at sink server is an essential issue on WSNs fields.

The flexibility of deployment of WSNs can ensure a wide range of applications for sensor networks. For example, in military, the rapid deployment, self-organization, and fault tolerance characteristic of sensor networks make them a very promising sensing technique for military command, control, communications, surveillance, reconnaissance, and targeting systems. Moreover, WSNs benefit in health care, inventory management, product quality monitoring, and disaster areas monitoring [3]. In the following subsections, sensor nodes architecture and sink server functions are illustrated.

2.1.1 Sensor Node Architecture

A typical WSN architecture is illustrated in Figure 2.2. Each scattered sensor node has capabilities to collect sensing and route data back to the sink through multi-hop infrastructureless path. The sink can communicate with the task manager node, such as PDA, via Internet and satellite [4].

There are various demands placed on WSNs software. Efficiency in processing, storing data and utilizing energy are essential. Besides, it is necessary for the software to allow multiple applications, e.g. communication, computation tasks, and memory management, to simultaneously run on the system. The most important of sensor node is the operating system(OS). OSs for WSNs nodes are typically less complex than general-purpose operating systems because of the resource constraints of the hardware of sensor nodes. Nevertheless, sensor nodes system are restricted to some specific embedded operating systems, such as uC/OS for sensor networks, for the reason of operating behavior such as event-driven. However, this kind of operating systems is generally designed for real-time applications, which is no necessary for the application of WSNs. Therefore, event driven OSs which can call an event handler to handle the task when an external event occurs, such as an incoming data packet and a sensor reading, is suitable for sensor nodes.

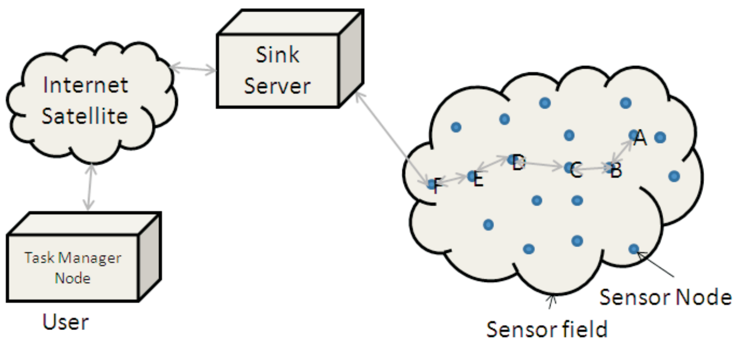

Figure 2.2: Sensor nodes scattered in a sensor field

2.1.2 Sink Server

Most applications of the WSNs are data centric. They need to gather sensing data and are managed by a server (or Host Computer), such as PC. The server can analyze received data, response accordingly and control the node in the sensor networks to perform some tasks. Thus, a device between the sensor networks and server is called sink or base station. Sink can receive data from radio and place packet to the buffer before transmitting it to Host Computer through interfaces, such as UART interface. A program run on host computer can obtain this packet from its related interface and parse it to usable information. For user friendly, visualization support of sensing data is essential as well. Besides the server, sensor readings received at server need to keep for further processing. The database service is provided by SQL-based software which is installed on the host computer. The logged data is accessed by the visualization tools and interface the data to the user.

2.2 Smart Home Concept

In September 2003, Housing Learning &Improvement Network Lab, which has carried out the project named DTI Smart Homes Project, published a smart home definition offered by Intertek. Intertek considered that a smart home is "a dwelling incorporating a communications network that connects the key electrical appliances and services, and allows them to be remotely controlled, monitored or accessed". Remotely here means both are within the house and from outside the house. Doubtless, a home, which is smart, must contain three major elements: networking, intelligent control and home automation. An internal network is the basis of a smart home, and it can be wired or wireless. Intelligent control means a centralized control center to manage the systems. Home automation means most of home appliances could be control remotely and intelligently.

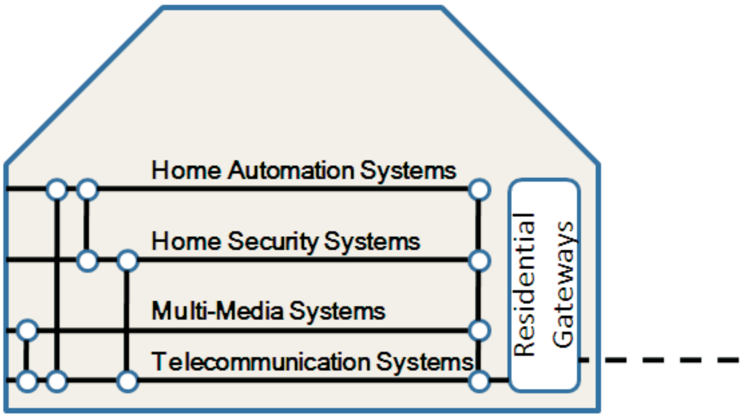

Figure 2.3: A schemtic model of a technical structure of a Smart Home

The Figure 2.3 which is proposed by Ilse Bierhoff [6] indicates that a Smart Home environment basically contains four components and a residential gateway. This smart home environment includes Home Automation Systems, Security Systems, Multimedia Systems, and Communication Systems. The residential gateway is a centralized control center of the smart home system and also connects to outside network to enhance its capability. All functions of the smart home system can communicate with the residential gateway. Mostly speaking, the gateway supports Internet access. Host can use PDA or other handheld system to monitor house. In the following two sections, we'll introduce related projects of smart home applications.

2.2.1 Wireless Sensor Network for Health Monitoring

A network architecture for smart healthcare based on WSNs is proposed by Virone et al.[5]. It targets assisted-living resident and others who may benefit from continuous, remote health monitoring.

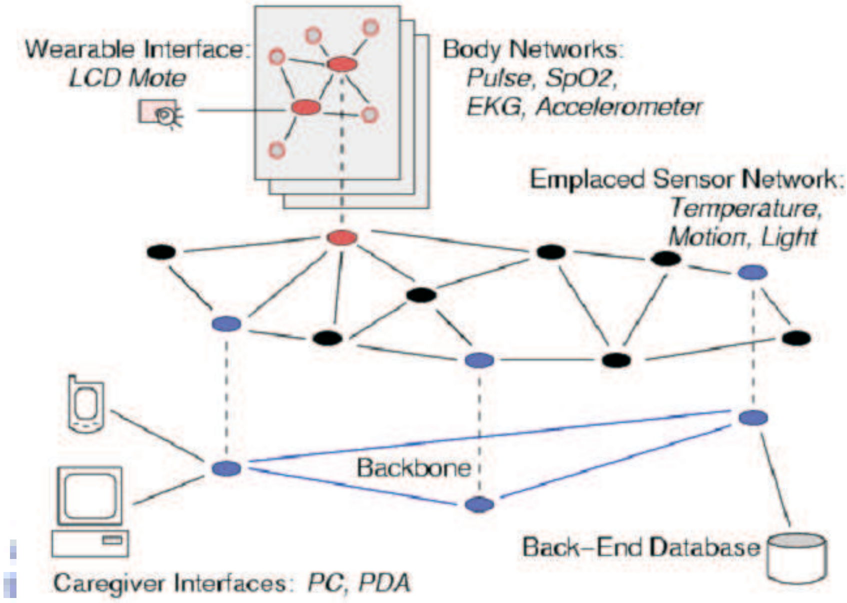

Figure 2.4: Multi-tierd system archtecture of Health Monitoring

The system architecture of that project is illustrated in Figure 2.4. There are three tiers in this system: Body Networks, Emplaced Sensor Networks, and Backbone Networks. The body network and its subsystem are tiny portable devices within a variety of sensors that perform biophysical monitoring, patient identification, location detection, etc.. The energy consumption of these sensor nodes is optimized. The emplaced sensor network includes sensor devices deployed in the environment to support sensing and monitoring. It also provides a spatial context for data association and analysis. Nodes in emplaced sensor network do not perform extensive calculations or store data. Emplaced sensor network interfaces to multiple body networks, seamlessly managing handoff of reported data and maintaining patient presence information. A backbone network connects traditional systems to the emplaced sensor network, such as PDAs, PCs and databases. These nodes process significant storage and computation capability for query processing and location services. The backbone may communicate wirelessly or overlay onto an existing wired infrastructure. Yet, the count of backbone network is minimized to reduce cost. The back-end database connected by a backbone node is dedicated for long-term archiving and data mining. Human interface interfaces with the network using PDAs, wearable devices. These are used for data management, querying, object location, memory aids, and configuration. Caregivers can use these to specify medical sensing and to view important data.

2.2.2 UbiQ Smart Home Solution

The UbiQ is a commercial product of Smart Home Solution [17]. The system architecture of UbiQ is shown in Figure 2.5. This system is divided into several components. A Home Terminal is responsible for centrally controlling all appliances in the system and other components, communicating with the home terminal through wired medium. Room controller is responsible for controlling the room access and checking the identification of hosts. Energy controller controls the curtain and other electric devices. A set of home security devices, such as reed switch, gas sensor, etc., are directly controlled by the home terminal. There is a scenario controller which performs controlling lighting relay or temperature sensor according to scenario configuration. Host can remotly monitor the house

Figure 2.5: UbiQ Smart Home solution

condition by a handheld device.

2.2.3 Gator Tech Smart House

The project's[19] goal is to create assistive environments. To create the Gator Tech Smart House, the author proposed generic reference architecture applicable to any pervasive computing space. As Figure 2.6 shows, the middleware contains separate physical, sensor platform, service, knowledge, context management, and application layers. Physical layer consists of the various devices and appliances the occupants use. In sensor platform layer, each sensor platform defines the boundary of a pervasive space within the Smart House, "capturing" those object attached to it. Service layer contains the Open Services Gateway Initiative (OSGi) framework, which maintains leases of activated services. Context management layer lets application developers create and register contexts of interest. The context engine is responsible for detecting, and possibly recovering from,

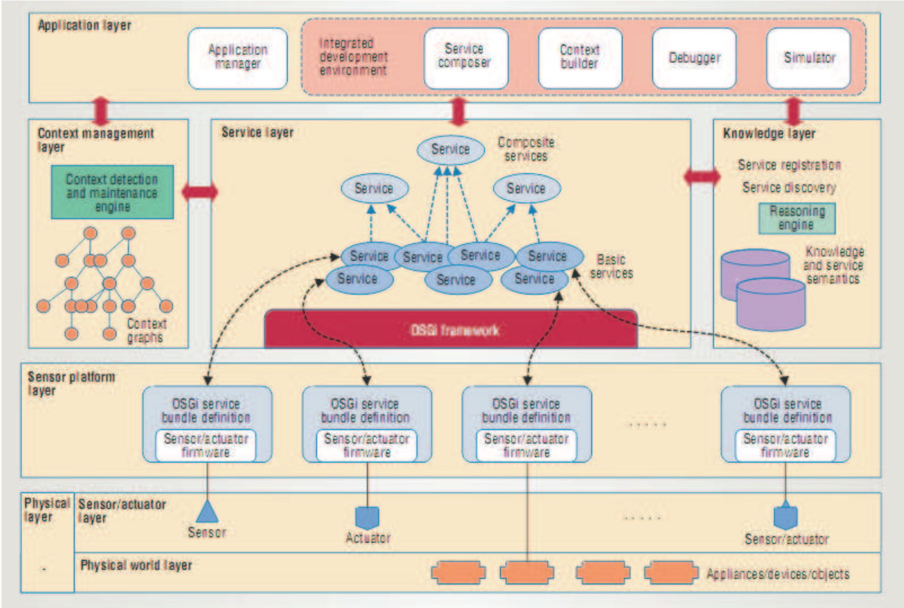

Figure 2.6: Smart-space middleware.

such states. Application layer consists of an application manager to activate and deactivate services and a graphical-based integrated development environment with various tools to help create smart spaces.

Moreover, the author proposed smart plugs, which provide an intelligent way to sense electrical devices installed in an intelligent space. Each power outlet in the Gator Tech Smart House is equipped with a low-cost RFID reader connected to the main computer. Electrical devices with power cords, such as lamps and clocks, each have an RFID tag attached to the plug's end with information about the device. When a user plugs the device into an outlet, the reader reads the tag and forwards the data to the main computer.

Chapter 3

Development Environment

In this chapter, we will give a description of the development environment of our project. In section 3.1, the wireless sensor nodes (called mote) and its external module are referred. An introduction of the software development tools and the concept of the specific language used in our project are proposed in section 3.2. Besides, the packet of the basic TinyOS environment is illustrated in Section 3.3. Finally, an abstract link layer protocol - sensornet protocol - is described in Section 3.4.

3.1 Hardware Description

In this section, a description of the primary platform of our smart home hardware is given. Taroko as shown in Figure 3.1 is selected as the mote hardware of this smart home platform. Taroko is a telosb [21] based sensor network hardware module, and it is a new generation mote hardware comparing to mica and mica2 [16]. Taroko features an 802.15.4 compatible 250kbps CC2420 radio which is controlled by an 8MHz 16 bit RISC MSP430

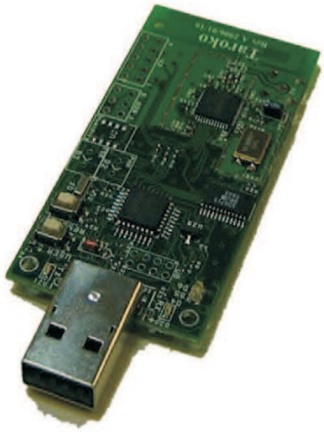

Figure 3.1: A picture of Taroko sensor node

microcontroller. The MSP430 microprocessor provides ADC ports, GPIO ports, USART and other peripherals. The key features of telosb based platform, such as Taroko, are described as following and the functional block diagram of the Taroko module is shown in Figure 2.1 :

- In Microprocessor, the MSP430F1611 8MHz microcontroller manufactured by TI [8] is utilized as the CPU of Taroko. It features 48KB ROM and 10KB RAM. A 48KB ROM means that the maximum size of the code that fits Taroko is only 48KB and the storage is smaller than ATmega128L. However, comparing to ATmega128L microcontroller of AVR which is utilized in micaz mote, Taroko is performed well in the energy consumption as shown in Table 3.1 [20] [14] [15]. This 16-bit RISC processor features extremely low active and sleep current consumption that permits Telosb based mote to run for years on a single pair of AA batteries. It supplys voltage ranging from 1.8 to 3.6 V, and then battery-operation is satisfied. Apart from low power consumption, there are 8 external ADC ports that can be utilized to receive external signals and a variety of peripheral, such as SPI (Serial port Interface), UART (Universal Asynchronous Receiver/Transmitter), I2C (Inter IC), etc..

Table 3.1: Mote Hardware Comparison

	CPU	min.Vol.	Active	Sleep	RX	TX	Radio
MICAZ	Atmel	2.7V	8mA	20uA	23.3mA	21.0mA	CC2420
Telosb	MSP430	1.8V	2mA	27uA	21.0mA	19.5mA	CC2420

- Taroko provides a CC2420 radio, which is manufactured by Chipcon chip [9], compatible of IEEE 802.15.4 spec.. It provides PHY and part of MAC layer function of mote's wireless hardware module. Furthermore, CC2420 radio provides different power level, 0 to 31, that user can modify radio power in a variety of environment. The CC2420 radio provides two metrics of RSSI (Receive Signal Strength Indicator) and LQI (Link Quality Indicator) to evaluate the signal strength or link quality from sender to receiver. The CC2420 chip is controlled by MSP430 microcontroller through SPI, and it shares flash with the external expansion connector. Therefore, bus arbitration is necessary for these peripheral. Moreover, this radio module also provides for clear channel assessment (CCA), which is used to implement CSMA-CA MAC protocol. Taroko provides internal antenna which has indoor range of around 50m and an external range of 125m at full battery power separately.

- Taroko appears as a COM port in Windows device manager. Multiple motes can be connected to the USB ports of a single computer at the same time. Each Taroko mote will receive a different COM port identifier through FTDI driver. Taroko communicates with the host PC through USART1 on the MSP430 microprocessor. Software developers can program the mote through the USB interface, and do not need a separated programming board. This feature is an advantage of developing comparing with micaz, which need an additional programming board, such as MIB510 or Stargate, to program the motes.

- Taroko hardware module supports 1MB STM25P80 40MHz serial code Flash. This can be conveniently utilized to store data and code permanently on the Taroko until that the flash is formatted. The flash shares SPI communication lines with the CC2420 radio and the external SPI pins. Therefore, Flash reading and writing have to carefully interleave with radio communication. In TinyOS embedded operation system for sensor hardware, there is a Bus Arbitration module to control all peripherals on SPI to prevent collision in data processing.

- The humidity and temperature sensor is manufactured by Sensirion AG. The SHT11 module is directly mounted on the Taroko hardware module in the U3 component position. The SHT11 sensor is calibrated and generates a digital output. The calibration coefficients are stored in the sensor's onboard EEPROM. This sensor is produced by a CMOS process and is coupled with a 14-bit ADC converter. The low power relative humidity sensor is small in size and is utilized for a variety of environmental monitoring applications.

- Taroko hardware module has connections between two photodiodes, S1087 and S1087-01, manufactured by Hamamatsu. The difference between these two chips is that S1087 is responsible for sensing photosynthetically active radiation and S1087-01 is responsible for sensing the entire visible spectrum which includes infrared.

Besides the primary mote hardware described above, in order to support the smart home applications of this project, some ready-made home security related products, such as reed switch and smoke sensor, are utilized. These ready-made external hardware modules are combined with the Taroko hardware module. Apart from the description above, a simple circuit is implemented, including relay attached to Taroko for controlling electric device by remote server or handheld device. Since there is a home gateway to be a centralized controlling and monitoring system, a PC is utilized to install the relative software, such as TinyOS development tools, database , Java Runtime Environment, etc.. The centralized home server is responsible for gathering information from sensor network and controlling sensor nodes behaviors, such as starting sensing, executing command.

3.2 Software Description

a. Cygwin

The home server (a PC connected with a mote, which is called base station) runs a UNIX like Cygwin [10] system that is required by TinyOS environment. Cygwin is a collection of free software tools originally developed by Cygnus Solutions to allow various versions of Microsoft Windows to act somewhat like a UNIX/Linux system. Cygwin creates a UNIX like environment with many of its standard utilities, including development tools, such as GCC and GDB.

b. TinyOS

TinyOS [11] proposed by UC. Berkeley is an open-source, event-driven, and real-time operating system for embedded network sensors. TinyOS is available for free and is ported to some embedded platforms, such as mica series motes(mica, mica2, and micaz) and telos series mote(telosb, tmote sky, and Taroko), etc.. In the beginning of developing smart home application, a TinyOS development platform is needed to install on PC. The TinyOS software package contains some useful library modules, such as network modules, hardware driver modules, and other optional application related libraries. In order to communicate with PC during executing period, TinyOS supports tools which allow the base station mote to communicate with the home server through a Java application with Java communication API.

According to the view of kernel, TinyOS provides a two-level scheduling hierarchy consisting of tasks and events. An event is a generalization of an interrupt handler. A "call" that is made from a low-level component to a higher-level component in response to some events. Events perform only lightweight works because they always run to completion and cannot be preempted. Tasks are utilized for performing normal processing operations, such as background data processing, and can be preempted by events.

The following Figure 3.2 is the TinyOS design architecture:

Main Scheduler part: It is responsible for arranging all TinyOS tasks. Main Scheduler maintains a FIFO queue. When a task is post to be executed, scheduler pushes it to FIFO queue and wait for further executing. The Main components start this scheduler before processes and threads. Main component is the entry point of TinyOS application

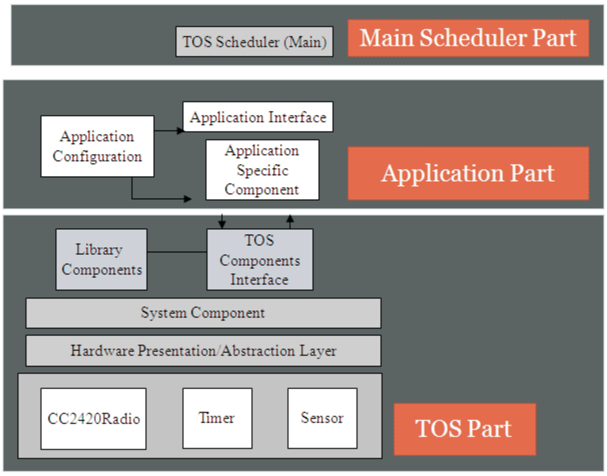

Figure 3.2: TinyOS design architecture

program. All other components are wired to Main component directly or indirectly. Main component initially initialize all components and then call the start function in StdControl interface of these components.

Application part: It is the primary section of developing TinyOS application. Developer can follow the nesC language syntax rule to accomplish TinyOS module or application program. In addition to the description above, there are a variety of preinstalled libraries that is included in user application section, such as 802.15.4 MAC module, variety of network module, low-layer hardware module, etc..

TOS part: Since TinyOS is a operating system for sensor hardware, a series of basic operating system component is required, such as scheduling, process management, etc.. For simplification, TinyOS does not recommend user to modify or create operating system by themselves. Hence TinyOS supports the basic operating system components which are automatically assembled while making the TinyOS image file. These components are not shown as the real hardware setting attribute. The abstract model (hardware abstract layer) is provided for user to use. According to a variety of platform, TOS is responsible to call the lower layer (hardware present layer) component through interface between TOS and hardware present layer.

The key limitations of developing TinyOS are illustrated below :

- **There is no concurrency in TinyOS:** Only one task can be excuted at the same time. This task can not be interrupted by other tasks. The only exception of this rule is hardware interrupts which may interrupt the current task at any time.

- **TinyOS does not support dynamic memory allocation:** There is no heap in TinyOS so each variable is allocated on the stack . Hence, data structures have to be initialized to its maximum size at compiling time and therefore probably reserves too much of the limited memory of small sensor nodes.

Apart from the TinyOS concept described above, because we must use Taroko Sensor module such as photodiodes, humidity and temperature sensor, we install Tmote sky library (Moteiv Co. distribution) to TinyOS directory in cygwin. Tmote sky library includes hardware module library function and other hardware relative library module, such as sensornet protocol, etc.. In order to insure the latest version of TinyOS,we run cvs to download the latest version of files.

c. NesC

TinyOS is written in a programming language called NesC. NesC has syntax similar to C but supports the concurrency model of TinyOS. NesC applications are built out of components which have well defined bidirectional interfaces. It also defines a concurrency model, based on tasks and hardware event handlers, to detect data races at compiling time. Each NesC application consists of several components which are wired together to form an executable program. A component provides some interfaces for other components and may use several interfaces of them. These interfaces are points of access to these components. An interface declares commands that the interface provider has to implement and events that the interface user has to implement. Therefore, if developers want to use a command provided by an interface, they have to implement all the events specified by that interface first. Modules are used to implement interfaces and configurations are used to wire other components together.

d. MySQL

We ported an open source database, MySQL, to the server in our implementation. It is a multithreading, multi-user SQL database management system. MySQL is popular for web applications and is often combined with PHP. PHP and MySQL are essential components for running popular content management systems. In our system, all the informations of the home environment are stored in the database and shown on web page written by PHP.

3.3 TinyOS Packet Format

The message buffer in TinyOS 1.x is TOS_Msg. A buffer contains an AM (active message) packet as well as packet metadata, such as timestamps, acknowledgement bits, and signal strength if the packet was received. TOS_Msg is a fixed size structure whose size is defined by the maximum AM payload length (the default is 28 bytes in Moteiv co.

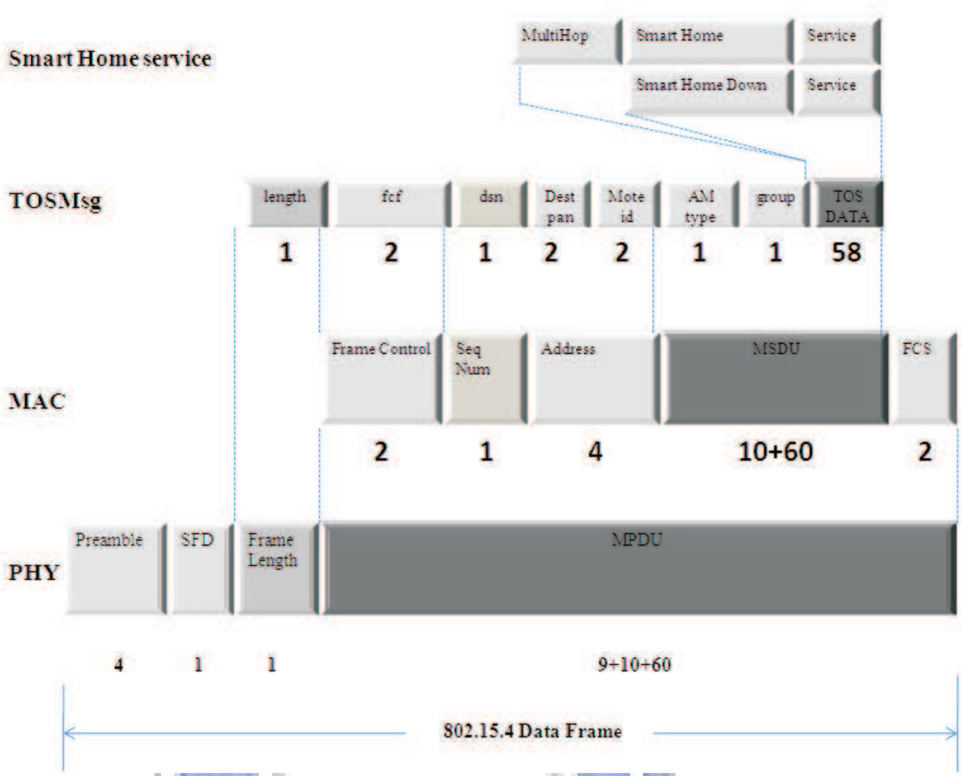

Figure 3.3: Smart home platform packet format

distribution). Fixed sized buffers allow TinyOS 1.x to have zero-copy semantics: When a component receives a buffer, rather than copys out the contents it can return a pointer to a new buffer for the underlying layer to use for the next received packet. The packet format of the smart home platform is shown in Figure 3.3. TOSMsg in TinyOS1.x is not well defined all the newly packet format of 802.15.4 MAC layer. The Length and FCS value in message header and message footer of MAC layer are generated by hardware platform. The TinyOS radio stack assigns the fcf, dsn, destpan, AM type and group field to beginning of data payload. All the active messages are packaged within the data field of the TOSMsg packet.

3.4 Sensornet Protocol

In this section, a description of the lower layer implementation of this wireless sensor network platform is given below. In Moteiv provided module, the sensornet protocol(SP) plays as a rule of coordinating various link layer protocols and various network layer protocols and we utilize SP to handle lower layer behavior.

The objective of utilizing SP is to simplify development and use of reliable communications in low power wireless network. The key challenge for SP is providing adequate insulation between the hardware below and the various communication abstractions above while still providing adequate efficiency. It should allow network level protocols to optimize for the underlying link in terms of the characteristics expressed at SP, rather than knowing which particular link and physical layer resides beneath. SP allows the coexistence of multiple transport and routing protocols. Moreover, SP must allow network

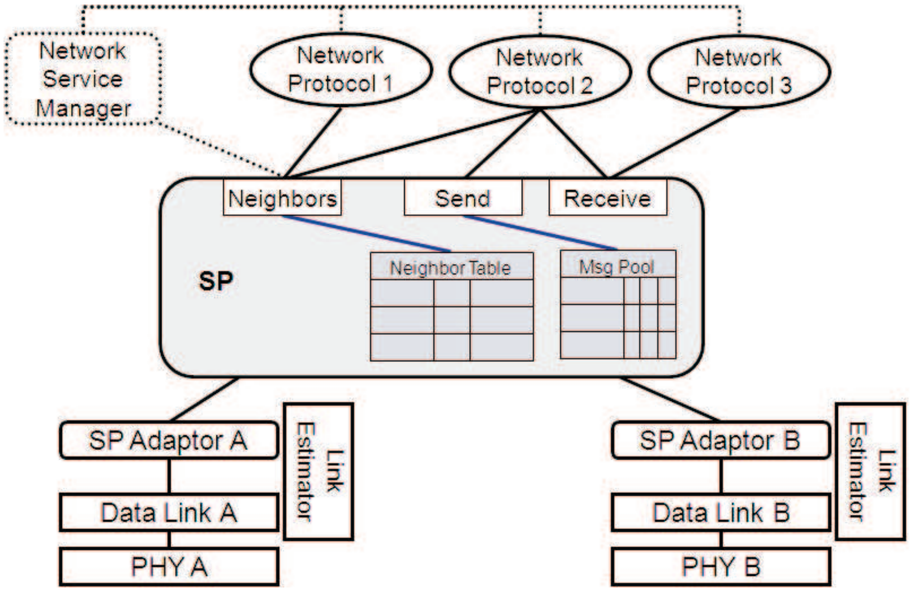

Figure 3.4: The sensornet protocol architecture

protocols to choose neighbors wisely, and be taken into account information available at the link layer [12].

Figure 3.4 is the SP architecture [12] which maintains a shared neighbor table and message pool. The shared neighbor pool allows SP to decide when to listen, receive, transmit, and sleep. The message pool allows network protocols to request message transmissions. The message pool contains references to messages which can be accessed out of order. The messages are not stored in the SP layer; They are either stored in the network or the link layers.

Figure 3.5 shows the operation scenario of sending a message by using SP [12]. First of all, the network protocol layer desires to transmit a message, which is submitted to SP. The pointer of transmitted message is added to message pool in SP. The SP decides the transmitting time which depends on the batching and link protocol inspection of the neighbor table. After transmitting, SP requests the next packet of the SP message to

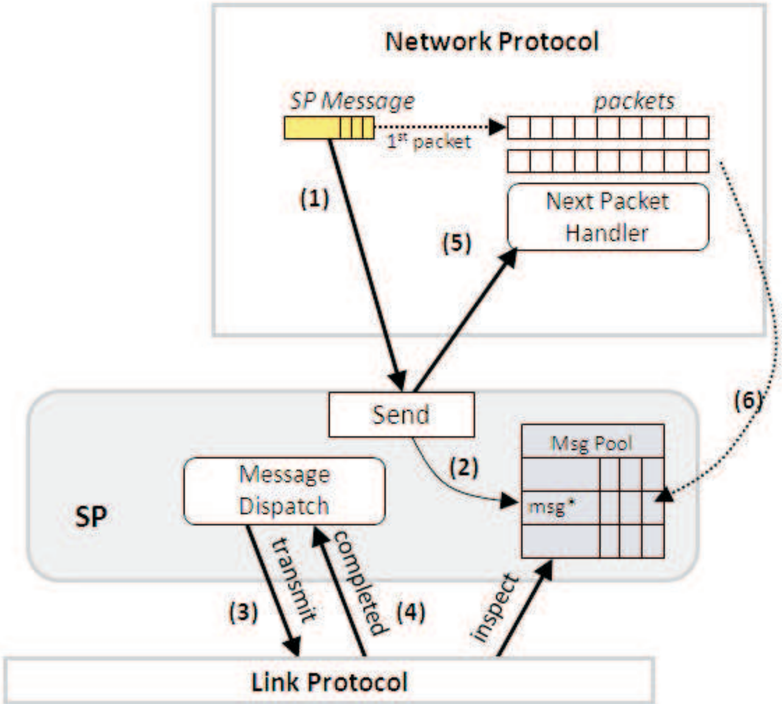

Figure 3.5: Operation of sending a message using sensornet protocol

transmit and the SP message pointer is updated to point to the next packet buffer.

The SP protocol can simplify the development of sensor program and use of reliable communications in low power wireless network. Figure 3.5 is the proposed stack of SP. It shows that the Sensornet Protocol separates the network and Link layer through a set of common interfaces. A common routing and transport interface is proposed by SP, allowing applications to easily use multiple routing protocols in a single application.

Chapter 4

Smart Home Platform Design

In the beginning of designing the smart home platform, we use the top-down approach to analyze smart home platform and bottom-up approach to design every service component. For the sake of constructing a smart home platform by wireless sensor network and its collaborated device, this smart home platform design is classified into mote-based smart home service design, sensor network design, and sink server design. Mote-based smart home service design described in section 4.1 is concerned about whole smart home service on the mote. Sensor network design described in section 4.2 is primary concerned about how to construct the sensor networks. Sink server design described in section 4.3 is concerned about the home server supported for smart home platform. We analyze detail feature in each component and design all functions in each component. Finally, an electric digital device emulator described in section 4.4 is proposed to make up the lack of smart home platform.

4.1 Smart Home Service Design

In the design of mote-based smart home service, the smart home requirements are break three kinds of service which are home security service, home care service, and home automation service respectively.

a. Home security service:

Home security service which includes two subsystem, entrance guard system and anti-thief system is a significant service in smart home platform. These two home security subsystems are described as below.

Entrance guard subsystem supports a safety gate control. Householder utilizes a reaction mote (Human Mote) to either get in or step out the house. The stranger who doesn't carry the Human Mote can't open the gate unless the host opens the door inside for him. The entrance guard system utilizes a mote attached to the gate (Gate Mote) to receive the tag information of Human Mote. If a stranger attempt to utilize a fake mote to get in the house, the Gate Mote checks the error tag information and sends a message back to the Home Server to notify householder and to trigger the alarm noise at the same time. Figure 4.1 shows the architecture of Entrance guard system.

Anti-thief subsystem supports a solution to react the intrusion of thief. To employ infrared rays sensor or reed switch to detect the intrusion is a common approach. Each anti-thief mote (Security Mote) integrated infrared rays or reed-switch circuit is attached on the window lock. When detecting the intrusion, the Security Mote reports intrusion event back to home server and turn on the alarm noise at the same time. It also integrates auto-dial system to notify police and householder and is able to combine with video

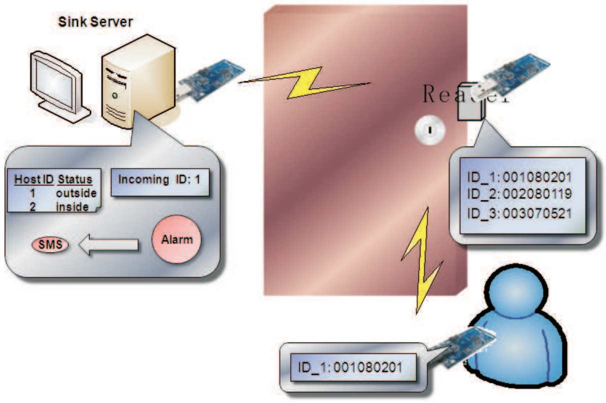

Figure 4.1: The architecture of Entrance Guard subsystem

monitor system. When a thief intrudes the house and triggers the sensor through open the window, the system automatically turns on the video monitor system to catch the thief face. Host is able to utilize web-based remote monitor system through the PDA or other handheld devices to monitor house condition. The following figure is the Anti-thief subsystem design architecture. Figure 4.2 shows the architecture of anti-thief system.

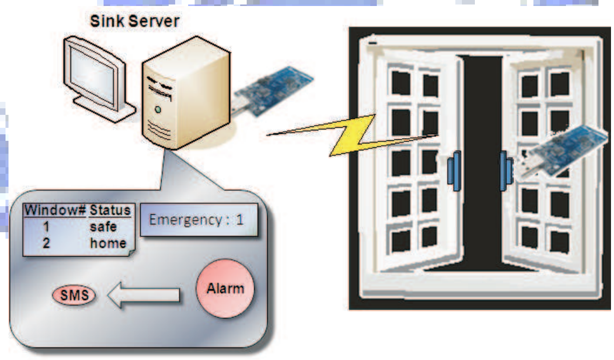

Figure 4.2: The architecture of Anti-thief subsystem

b. Home care service:

In recent years, there are many retired people in the world and the elder care commercial product becomes more and more popular. Smart home platform is one of these supplied solutions for elder and defective. If there is a home care service in the smart home platform, the condition of elder or house is able to be monitor. Therefore, the accident can be prevented and the elder and defective can be take care well.

Home care service in the smart home platform can be divided into indoor room lighting subsystem, environment sensing subsystem and fire detecting subsystem. These three home care subsystems are described as below.

Indoor room lighting subsystem is a kind of home care service application for elder and defective. It is an application of location awareness service. As the paper [7] mentioned that the mainly types of localization are physical location, symbolic location, absolute location, and relative location. The concept of symbolic location is employed as the principle to design indoor room lighting subsystem in the smart home platform. It means the location we expressed in natural language way, such as in the kitchen, in living room, in bath room, etc.. The metric to estimate the position in this smart home platform is RSSI (Received Signal Strength Indicator) which is included in the 802.15.4 PHY standard. The experiment result of the relationship between RSSI and distance is given in Chapter 5. It shows that there is a positive correlative property between RSSI and distance of pair mote. Therefore, a static mote (Room Mote) is attached in each room, and the Human mote receive the signals transmitted by each room in a little period. Human mote select the maximum mote RSSI which is also upper the threshold as the designated Room. This symbolic location approach is combined with automatic indoor light switch. If a host gets in a room and the irradiation is dark enough, the room light is automatically opened. The server can monitor the symbolic position of the Human mote. Figure 4.3 is the design architecture of the indoor room lighting subsystem.

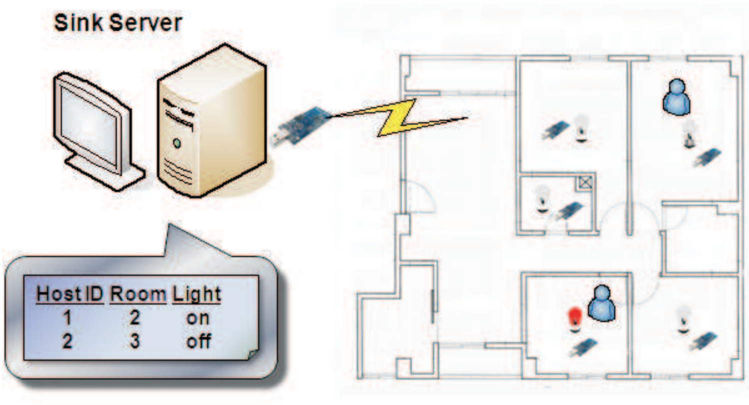

Figure 4.3: The architecture of indoor room lighting subsystem

Another home care service application we developed is environment sensing subsystem. There are a lot of service that can be accomplished through sensing the indoor environment, such as temperature, humidity, and luminosity, etc.. If a householder sleeps in a room and home server monitors the variation of temperature via subsystem, the related smart home appliance can be trigger. Figure 4.4 is our environment sensing service design architecture. The Room Mote is responsible to collect the sensing data and forward back to the home server, and then the home server perform the reaction according to the receiving data, such as turning on the HAVC.

The other home care service in the smart home platform is smoke sensing subsystem. The design objective is to prevent fire and prevent injuring if a fire accident occurs. Figure 4.5 is the design architecture of the smoke sensing service in the smart home platform. There is a smoke sensor integrated with the Room Mote. If the fire accident occurs, the generated smoke triggers the smoke sensor to notify Room Mote to transmit an alarm message to home server. The home server performs urgent tasks, such as notify

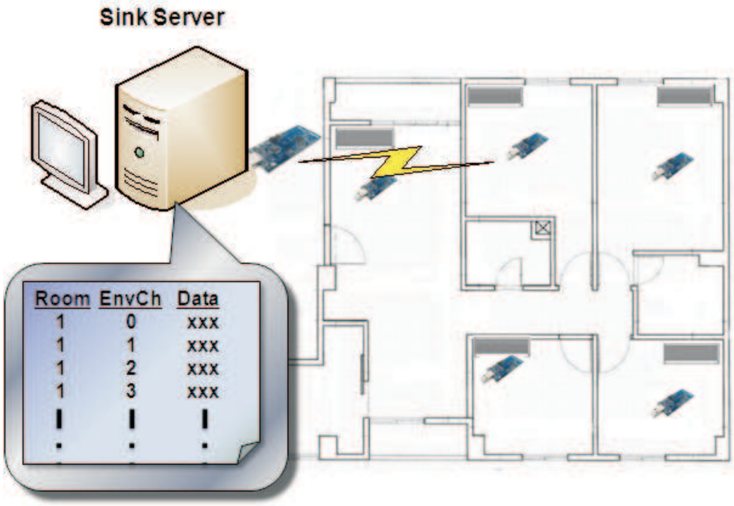

Figure 4.4: The architecture of environment sensing subsystem

householder via sending the SMS message.

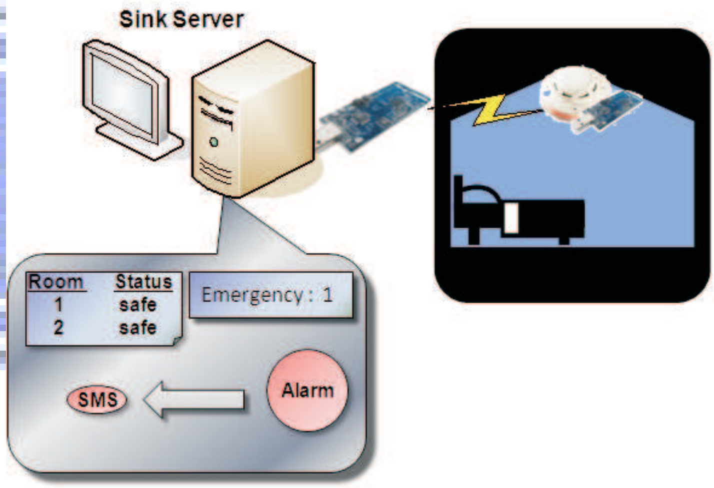

Figure 4.5: The architecture of the smoke sensing subsystem

c. Home automation service:

A home automation service includes the security, cooling, heating, and lighting automatically for comfort, energy savings, and convenience. It integrates home appliances into a whole system via data processing technologies. First of all, the home automation system needs a network to communicate with all controlled home appliances which are able to accept control commands and / or return execution results or internal states. Ex: Network Refrigerator, Media Server, etc.. There are some standardized protocols between home appliances and the control system, such as X.10, LonTalk, UPnP, etc.. Finally, context-aware service makes the computer aware of the home environment and provides a comfortable living environment for householder. The auto-dial system is proposed to response the alarm of home security. It is the represent of the context-awareness service.

However, there is a problem of controlling all home appliances in existence. Each appliance have to be control by itself controller or device. The home appliances can be classified by device signal:

1. Infrared rays (IR) appliances: It is common home appliances in being such as DVD, TV, etc.. They are controlled by infrared ray remote controller. The home appliances can learn all the control code of the remote control and then correspond to the relative key.

2. Relay: Commonly, the light or other large current appliances are controlled by relays.

3. RS-232, 422, 485: Such as projector. All the actions and behavior of the appliances is controlled by control code which is transmitted through RS-232 or RS-422 or RS-485.

4. Network: Such as IP Cam.

Briefly, for the sake of developing a whole home automation system, an integrated appliance controller has to be implemented. The controller is able to control either IR-based appliance or relay controlled appliance. However the complicated controller is not proposed in this smart home platform. For the sake of fulfilling the concept of home automation service design, a control circuit and an emulator is referred. The mote (Automation Mote) attached with the circuit is able to control non-digital home appliance, and the emulator is employed to demonstrate the completion of home automation service. The emulator includes several kinds of virtual appliance, such as TV, air conditioner, etc.. The design architecture of home automation service is shown in Figure 4.6.

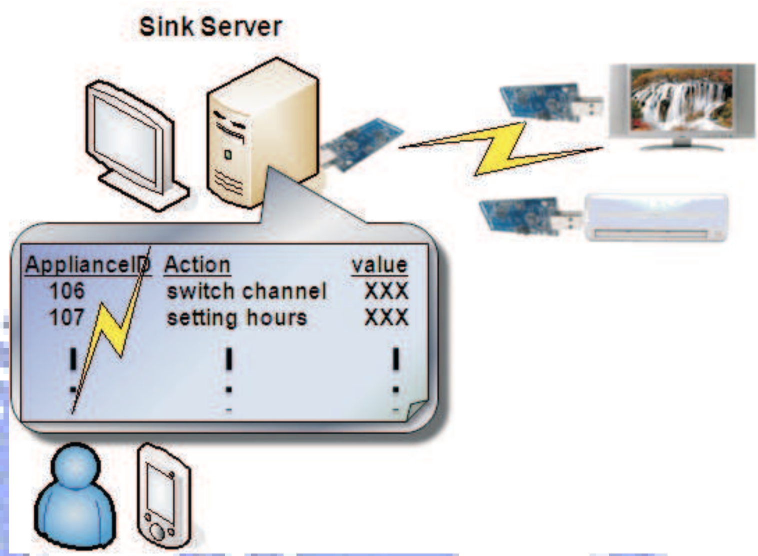

Figure 4.6: The architecture of the home automation subsystem

4.2 Sensor Network Design

The smart home motes on WSNs have several specific behaviors which depend on the objective. For example, if the mote is responsible for control room light, such as Room Mote, it should be static and non-mobile. On the other word, the mote's energy is able to be supplied by the AC of the room. Therefore, the design of sensor mote cannot consider the energy consumption issue. However, if the mote is responsible for householder, such as Human Mote, it is view as be mobile and the energy is constraint by 2 AA batteries. Therefore, the design of this kind of mote cannot ignore the energy consumption issue. Hence, the motes of the smart home platform can be broke into two types, mobile and static. The Human Mote belongs to the mobile mote, and the other motes proposed in this thesis are static motes, such as Gate Mote, Room Mote, Security Mote and Automation Mote. The lower layer function of mote is different from mote behavior. As we can see in section 4.1, most smart home services are data-centric based services, which have to forward data packet back to sink, the home server. Therefore, for the sake of improving the throughput of the smart home platform, a collection-based routing algorithm is utilized as the network layer function.

The MultiHopLQI module in Moteiv distribution is selected as the major routing algorithm in the smart home platform. Since most smart home services are many-to-one based services, the only sink server to collect the data sensed by the mote in the environment. For the sake of optimize many-to-one based service, the MultiHopLQI routing module is selected. Besides of the reason mentioned above, MultiHopLQI routing module collocate perfectly with sensornet protocol which is selected as the lower layer function of the smart home platform.

However, there is a problem using MultiHopLQI module. If the message has to be send to the sensor network, it is failure for original MultiHopLQI forwarding module. Therefore, the module is modified to support one-to-any packet delivery.

The routing algorithm in the smart home platform is not a dynamic on-demand routing algorithm. It is a tree-based distributed proactive routing algorithm. Therefore, since the Human Motes are not in the same area for a long period, this kind of mote not need to route packet for other motes. If the mote route packet for other mote, the failure occurs constantly. Hence, routing for other motes is not meaningful. However, the static motes, such as Gate Mote, Security Mote, Room Mote and Automation Mote, are deposited in a fixed position. Therefore, routing for other motes is the other task besides the main service task. Figure 4.7 shows a large scale view of the routing in the smart home network platform. The beacon message is sent to set up a routing tree from the sink server (As a root of the routing tree). Each node keeps the information of parent and s set of neighbors. The parent is selected by the link quality measured from LQI. The sink server keeps a table of forwarding path. The updated routing path from root to node in the network is listed in the table. As receiving a down-stream packet, the server gets the routing path and forwards to the designated mote hops by hops.

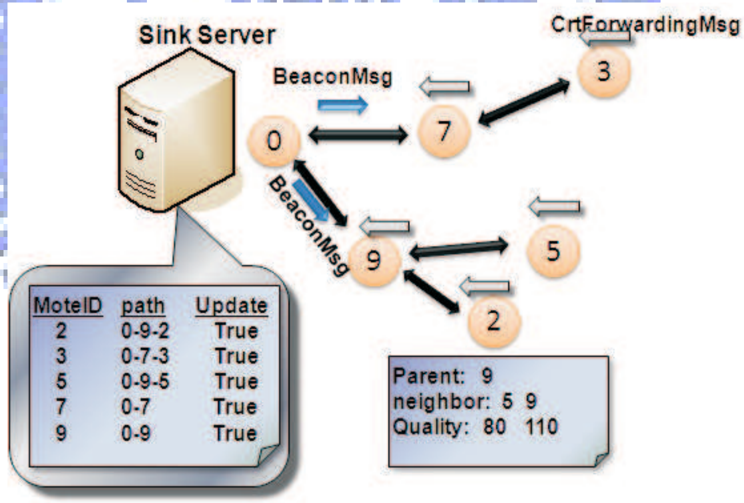

Figure 4.7: Large scale view of routing

4.3 Sink Server Design

The sink server takes the responsibility for data processing, data storing, and responding event. It is the core section of smart home platform. Many researches focus on design a systematic home server. Figure 4.8 is the framework of the home server in the smart home platform.

The packet level of the system is the component that directly sense context. Our categorization of the context is mainly come from Schilit et al [18]. Physical context includes the humidity data, the temperature data, the reed switch and smoke sensor reporting data. User context includes the user location, user ID, and another user data. Appliance context includes media device data and other device generated data.

Data from packet level of server stores into the related data table in the database respectively, since it must always be possible to utilize data on the basis of diverse criteria. Database management layer manages context information and offers upper layer the access to it.

The upper layer is Application layer. The application layer is divided into three sections, home security section, home care section and home automation section. Each section has the decision sublayer to determine and access the related context in database management layer. Task layer is used to response event according to the result of the decision layer. Therefore, each section exists several tasks to provide the whole platform behavior.

Besides of context-aware system architecture, the server also needs a development section during constructing the smart home platform. Hence, a development tool is proposed and built on the home server. The householder is able to generate the service program mote and create related server section support simultaneously.

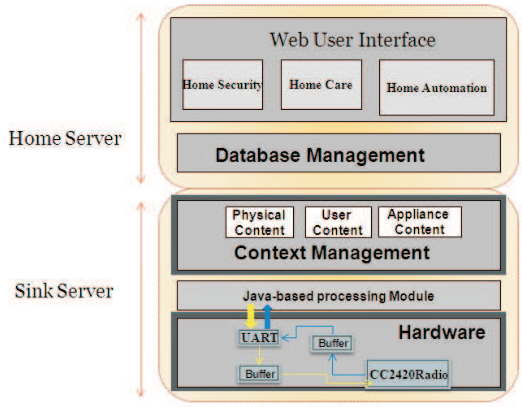

Figure 4.8: The framework of the home server in the smart home platform

4.4 Emulator Design

Since the appliances, such as TV, Air Conditioner, etc., have their own encode rule even though they all belong to infrared rays appliances, the encode information about controlling the appliances is not annexed from the manufacture company. For the sake of demonstrating the home automation service, an emulator is proposed to show the ability of controlling home appliances via the WSNs. Besides the home server, a visualized user interface is employed to emulate the behavior of home appliance, such as turning of or off, controlling other actions.

The emulator architecture is shown in Figure 4.9. On WSNs side, it performs radio transmitting and receiving. It looks like the radio module on the home appliances. Packet processing module is responsible for data encapsulation and packet parsing. Event handler module is responsible for triggering the emulated appliances to perform some specific action and managing emulator to add or delete controlled virtual appliances. Visualization module performs visualized user interface.

There is a problem that how to control the whole virtual appliances on a mote, which is utilized to receive and transmit all virtual appliances data. Logically the appliances being controlled have to perform at the same time. However, there is only one mote to responsible for data processing. For the sake of distinguishing each virtual appliance performs the related action. For simplicity, the virtual mote ID is proposed to verify each virtual appliance. According to the virtual mote ID, the data is able to be distinguished in processing module at the emulator and the home server (sink).

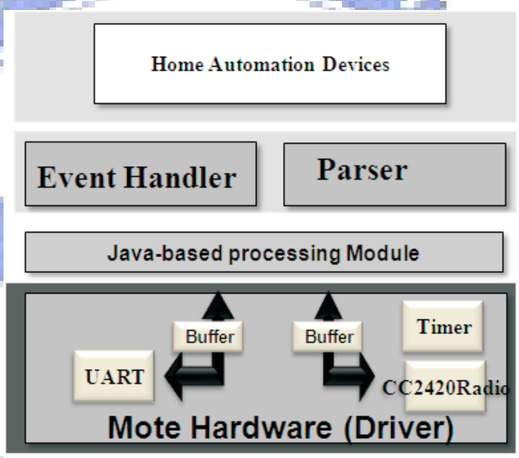

Figure 4.9: The emulator architecture

Chapter 5

System Implementation

The smart home platform implementation is described in this chapter. The modified sensor mote networking module in the smart home platform is proposed in section 5.1. It contains a tree-based routing module and a proactive forwarding module. In section 5.2, the whole mote application program of the smart home platform is implemented. It contains home security services, home care services and home automation services. Besides, a digital electric device emulator is implemented for home automation service. The emulator emulates the behavior of electric digital and its communication module. Finally, the smart home server and database design which includes the smart home development UI is proposed in section 5.3.

5.1 Sensor Mote Networking Module

The modified MultiHopLQI and Forwarding Module are proposed upon the sensornet protocol. As shown in Figure 5.1, there are four components in this tier. The default routing module, MultiHopLQI in Moteiv distribution, is modified for fulfilling the drawback discussed in Section 4.2. The MultiHopLQI and MultiHopDataM are responsible for data collection application and also provide the essential information for forwarding module to construct the forwarding tree which is utilized to forward down-stream packet into the network. The additional ForwardingTreeM and ForwardingM play as a rule of forwarding network function between base station and the other nodes.

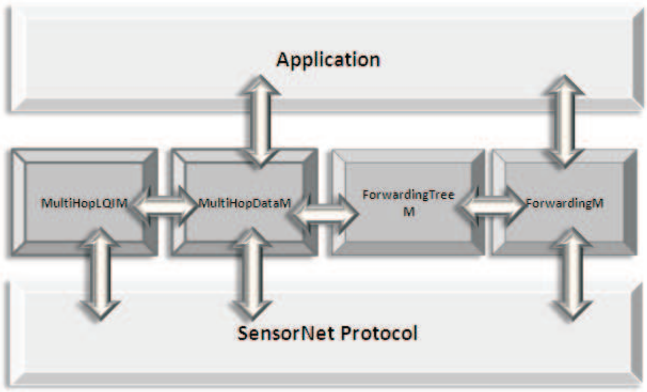

Figure 5.1: Sensor network implemanted architecture

The MultiHopLQI module is a distance-vector routing protocol. It generates one routing path per node from node to sink (base station). The routing path is a shortest path tree with minimum number of transmissions. The base station sends periodic beacons that are broadcasted by each node after receiving packet and incrementing a hop count in the packet. The mote selects the received beacons with lowest hop count from the ones it received, and adds the transmitter address as one of potential parents. Neighboring nodes exchange periodic beacons for the sake of evaluating link quality (utilize the LQI metric reading from the CC2420). MultiHopLQI has a continuously updated tree. The low-quality links and incorrectly attracting children is able to be avoided upon exploiting the MultiHopLQI protocol.

However, the problem of using MultiHopLQI protocol in the smart home platform is that there are some kinds of data need to be forward to the node in networks. The default MultiHopLQI protocol in Moteiv distribution does not support this kind of service. For simplicity, a modified the MultiHopLQI protocol is proposed to support the demands in the platform. The design idea is that each data packet is transmitted follow the routing path created by MultiHopLQI, and the path normally don't fail soon. Therefore, keeping the reverse path to transmit data from sink to node is a simple solution. In addition, the route recovery mechanism is to recover the failure path when the ACK is not received by sink.

The modified routing module is illustrated in Figure 5.2. At the base station mote, the ForwardingTreeM component accepts the Create-ForwardingTree message. This kind of message keeps the past mote address in an array. The base station maintains all mote routing path. When base station need to send a message, ForwardingM module is called to obtain the newly forwarding path from ForwardingTreeM module and copy the path into the sending message's header. In next step, ForwardingM checks the next hop mote address from the message and sends to next hop mote. The relay node also checks the next hop mote address at ForwardingM module and performs sending event. This kind of approach decreases the flooding problem in wireless sensor network if using broadcast mechanism, since every node drops the message which not sends to it.

However, if the path is not built correctly, the base station is not able to receive the ACK message from the exactly mote. Then the base station utilizes broadcast approach and TTL (Time-to-Live) mechanism to send this message until getting the newly forwarding path. By experimental result, there are few failing path over a long time, therefore the broadcast approach is not performed frequently.

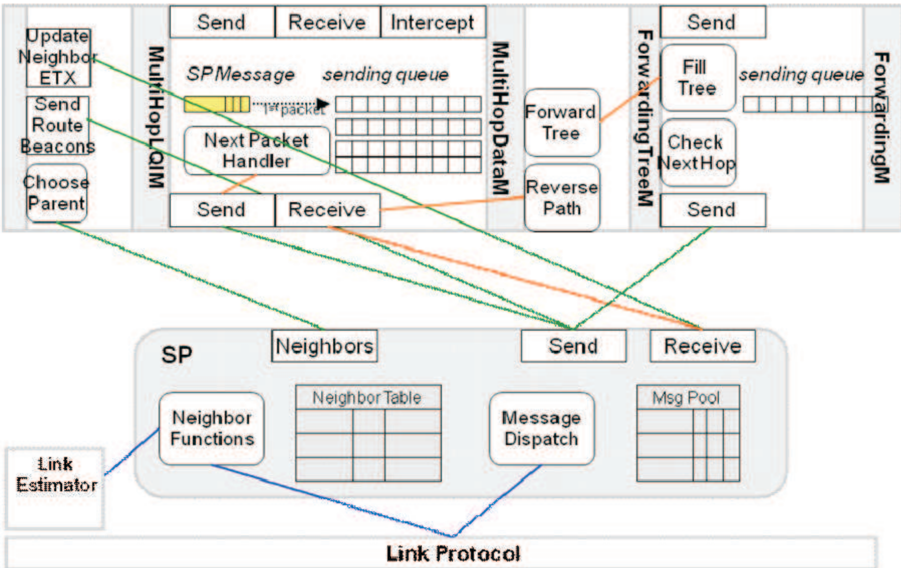

Figure 5.2: The modified sensor network module

5.2 Smart Home Service Implementation

As the description in Chapter 4, the smart home platform is proposed a set of the mote-based smart home services. In this section, the implement approach of the three kinds of smart home service is illustrated. Furthermore, for the sake of showing the workable of the home automation with the digital home appliance through WSNs, an emulator is implemented to emulate the behavior of the mote which is attached to the home appliance.

5.2.1 Home Security Service Implementation

The home security service contains a set of Security Motes, which can be attached on the window to protect thief intrusion. Besides Security Motes, the Gate Mote is implemented as a gate guard controller.

a. Anti-thief Application:

This program is performed via the GPIO interface on the expansion connector of Taroko. A simple conducted circuit is manufactured which is integrated with a reed switch. As the datasheet of Taroko described, there are four GIO port on the processor. The GIO2 (P2.3) is selected as our interface of this program. The module configuration of the anti-thief application is shown in Figure5.3. The operating flow of this service is as following. If the householder is at home, he can select which Security Mote has to be disabled. Besides, before he goes outside, he can send a signal to every Security Mote to wake up and perform security service. If a thief intrudes the house, the Security Mote is opened. It continuously sends alarm event message by short interval. However, the thief tries to close the window and home server not detects event. For the sake of preventing this condition, a simple state transition version of Anti-thief Application is proposed. If the host is at home and not sleeps, he can configure the Security Mote at Home mode and the window is able to be opened. If he goes outside, the Security Mote is able to be configured at Safe mode. When a thief opens the window, Security Mote starts to send alarm event message, and even the window is closed. Until receiving a SAFE_ACK message by server, the Security Mote is return to Safe Mode.

b. Entrance Guard System Application:

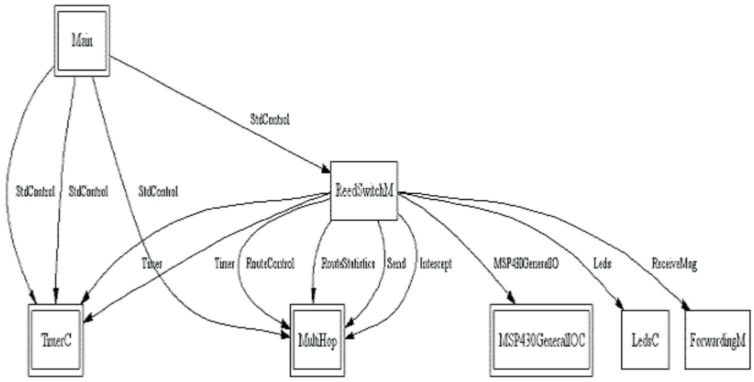

Figure 5.3: Anti-thief implement module

The entrance guard service supports a safety environment of the smart home. This service is performed by two kinds of mote, the Gate Mote and the Man Mote. The Gate Mote can be integrated with electric-lock of the gate. The Gate Mote maintains a list of householder information which is stored in the Man Mote. The Man Mote can update the data as receiving EntranceUptateMsg message. The Gate Mote sends EntranceHelloMsg message periodically. If the Man Mote's status is outside and receive the EntranceHelloMsg message, the Man Mote responses the EntranceACKMsg message. The Gate Mote transmits the EntranceConfirmMsg back to the Man Mote and Home Server for monitor the home environment. Besides, if the Man Mote is in the house, it does not response EntranceACKMsg.

The working flow of entrance guard service is illustrated in Figure 5.4:

5.2.2 Home Care Service Implementation

The home care service of the smart home platform is desired to provide a comfortable and safety house. In the testbed, a fire detecting application is implemented to detect

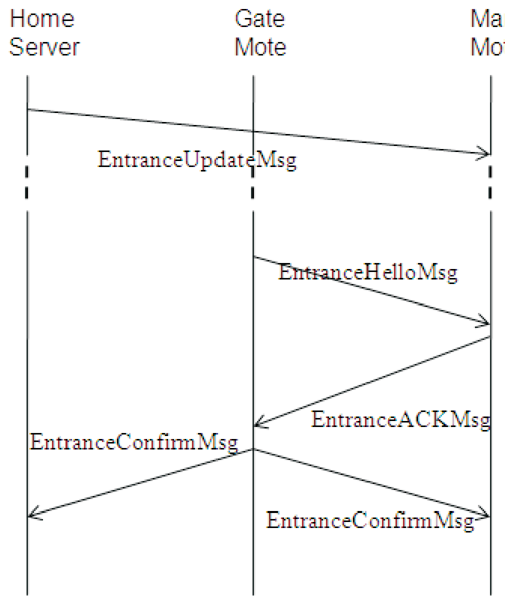

Figure 5.4: Entrance Guard subsystem working flow

unusual smoke from fire or hot plate. Besides, an environment sensing service is proposed to sense house temperature, humidity and photosynthetically. Furthermore, an indoor room lighting application is implemented for handicap and elderly people. The packet format of the home care service is illustrated as in Figure 5.5.

Service	Field	Description
Locate Report	man_id	The mote id carry by host
	room_id	The mote id in the room
	lqi	Link quality
Smoke Sensing	status	Smoke event
Environment Sensing	sourceMoteID	The mote id which sensing the environment
	channel	Specify which kinds of data
	lastSampleNumber	
	data[OSCOPE_BUFFER]	Raw sensing data

Figure 5.5: Home care service packet format

a. Fire Detecting Application:

Smoke sensing is a simple instance of integrated external sensor. For the sake of providing smoke sensing service which is not support by Taroko embedded sensor, a smoke sensor product is integrated with the Room Mote. This sensor send electric signal for detecting unusual smoke. The signal line is connected to the interrupt IO port on Taroko. The MSP430Interrupt module provided by TinyOS is able to fulfill this service.

b. Environment Sensing Application:

Environment sensing is an instance of utilizing the embedded sensor of Taroko. This service programmed on Room Mote periodically senses indoor information, such as humidity, temperature and photosynthetically. The information is sent back to server for performing context-aware service. The Room Mote also accepts command from home server to get run-time indoor information. A set of sensor module library provided by Moteiv co. is utilized to implement this application. The components graph of environment sensing application is shown as in Figure 5.6. There are four channels which are allocated to store four types of sensor: Humidity, Temperature, TSR and PAR. The TSR and the PAR sensor are provided to sense total solar radiation and photosynthetically active radiation respectively. The transform functions for these four sensors are shown as in Figure 5.7. Since the FPU is not provided by Taroko, the real meaning value of each channel cannot been obtained on Taroko. The sensing data is forwarded back to the home server and transformed into real meaning data

c.Room Light Control Application:

This service is proposed for handicaps and elderly people. The Human Mote is carried by elder or handicap. The Room Mote is settled at the position with a fixed distance to each room. The room mote periodically sends a hello message to Human Mote. If the

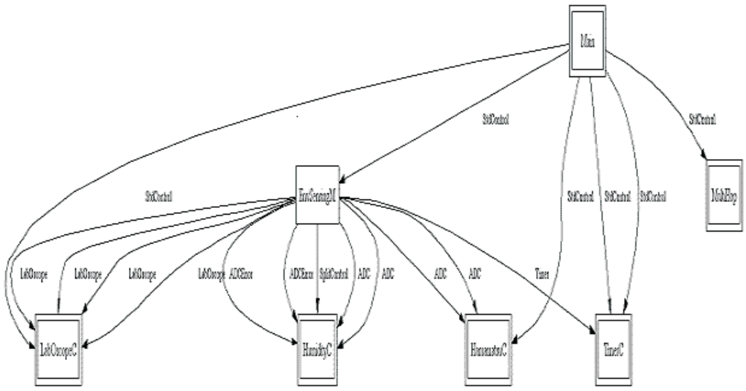

Figure 5.6: Environment Sensing implemented module

Channel	Name	Description
0	Humidity	raw_data : 12 bits value $tmp\_humidity = -4 + 0.0405 * raw\_data + (-2.8e * 10^{-6}) * ((raw\_data^2)$ $True\_humidity\ (relative\ humidity) = (Tc - 25) * (0.01 + 0.00008 * rawdata) + tmp\_humidity$ $(+) Tc\ is\ the\ temperature(^{o}c)\ measured\ by\ channel$
1	Temperature	rawdata : 12 bits value $temperature(^{o}C) = -39.60 + 0.01 * rawdata$
2	TSR (Total Solar Radiation)	rawdata: 12bits value $TSR\ (Lux) = rawdata / 4096 * 1.5 * 10000$
3	PAR (Photosynth-etically Active Radiation)	rawdata: 12bits value $PAR(Lux) = rawdata / 4096 * 1.5 * 1000$

Figure 5.7: Transform function for the raw data of embedded sensor

Human Mote receives message, it can make a decision that select a Room Mote with good RSSI value in a period time. If the value is higher than the threshold, the Human Mote is determined as a mote coming into the room. If the Room Mote receive a confirm message sent by Human Mote, the Room Mote and its neighbor Room Mote decide the designated room and perform lighting task upon the insufficient of sunlight. The designated Room Mote sends a report message to home server. The message indicates symbolic position of the Human Mote.

RSSI is register values and have no units. The radio chip CC2420 provides RSSI (Received Signal Strength Indicator) that may be read anytime. RSSI is part of the TOS_Msg packet but never transmitted. It can be accessed by Msg->strength (In TinyOS terms). The correlation of RSSI with the distance between the motes is very important for our work. With the help of the following experiment we would like to see how the RSSI vary with distance. The experiment result is illustrated as in Figure 5.8. The radio power can be changed by CC2420 radio chip. The level 3 is the minimal radio power and the level 31 is the full radio power. However, the distance cannot be clear verified via RSSI value if the radio power higher than level 5 in the indoor environment. Hence, the radio power of level 4 and the distance of 4 meters is recognized as the proper configuration for sensing correlated RSSI value.

5.2.3 Home Automation Implementation

Home automation is a fundamental application of smart home service. Every home electric appliance, such as Media Player, Air conditioner, and coffee set, are able to be

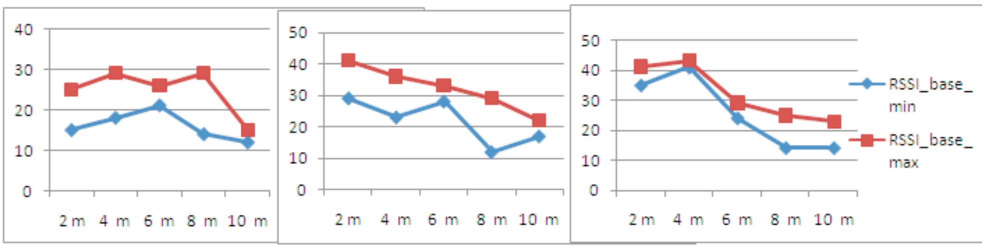

Figure 5.8: RSSI experiment result

controlled by the Automation Mote. The smart home platform performs the workable of home automation service by mote. The circuit to control complex digital home appliances is not implemented in this project. However, a simple circuit to control a simple switch appliance is manufactured to demonstrate the home automation service. Furthermore, a digital appliance emulator is implemented to emulate the behavior of Automation Mote controlling digital home appliances.

Remote Control Application:

This service can control home appliances, with on/off switch, by Human Mote or server. The following Figure 5.9 is the circuit of remote control service. The circuit attached with Automation Mote supports a simple relay to control switch. A command message can be sent by server or some specific mote such as Human Mote. If the Automation Mote receives command message, it triggers a signal via MSP430GeneralIO interface. Afterwards the relay will be triggered to control switch.

5.2.4 Emulator Implementation

This service mainly provides a demonstration that Automation Mote can control digital home appliance on WSNs. The emulator is proposed to emulate the appliance's

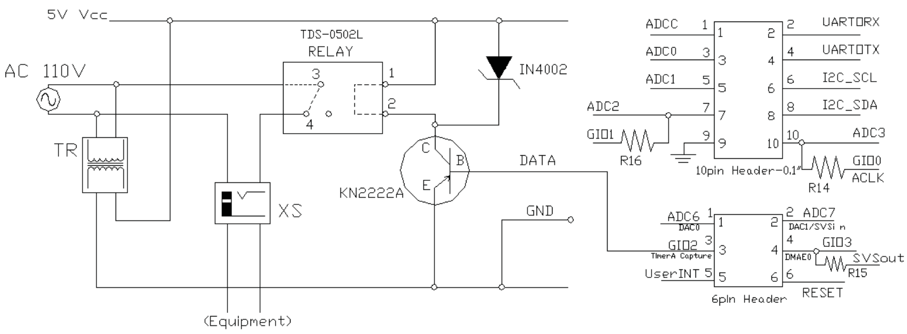

Figure 5.9: The circuit of remote control service

behavior when receiving all kinds of commands. A mote is attached with the PC and communicates with it via UART interface. The mote can be view as a digital home appliance coordinator. The architecture of the emulator is based on Figure 4.9. The mote receives and sends emulator message between UART and radio. A Java-based packet processing module is responsible for packet parsing and generating. An event handler triggers GUI and packet processing module generates relative response. The GUI can employ several virtual home appliances, such as TV, Air conditioner. The GUI reacts on the event message sent by Home Server or Human Mote. However, in virtual implementation, the mote ID which is used to recognize different device has only one ID in each mote. Therefore, a virtual appliance ID is proposed in message payload. The server or PDA can send command to each virtual appliance by the virtual appliance ID.

5.3 Sink Server Implementation

In our testbed, the sink server is another significant section besides WSNs. Actually the sink server is the core of the Smart Home platform. There are many popular researches on building context-awareness smart home environment. However, this smart home does not mainly focus on building context-aware environment. The mainly contribution of our server is to build a development tools for smart home environment. The development tool on the designate Home Server can help to indicate the mote position and build mote program. After building period, the server is performed as a sink via the attached base station mote. It receives all messages and insert usable information into the database on the server. All events and command can be shown on the Web-based UI. Host can see the information of the house through Internet. The implement result of our server including development tools is illustrated at the following paragraph.

Server Implementation

Figure 5.10 is our server with MySQL database. This server monitors the house environment and trigger related action, such as sending SMS and opening the air conditioner.

Figure 5.10: The implemented web-based home server

Development Tools Implementation

Figure 5.11 and 5.12 is the development period of the server. The designated smart home service is built into the mote. The mote icon can be set on the home plan figure after build mote program.

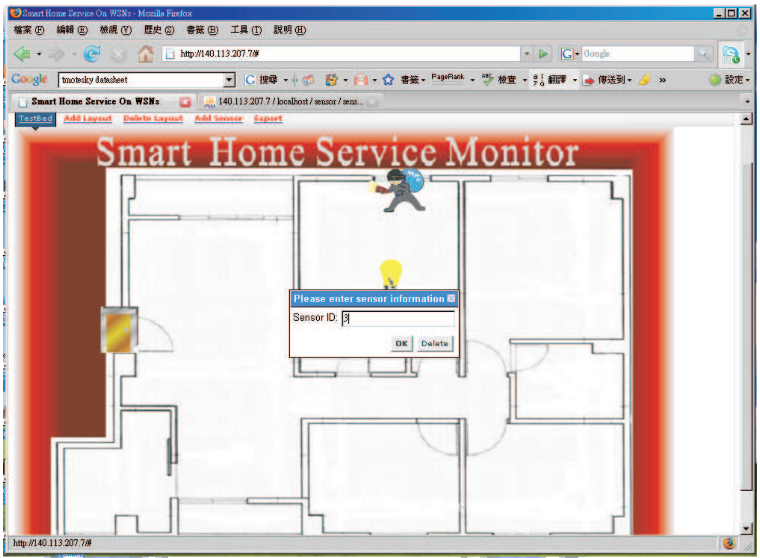

Figure 5.11: The development tools of smart home platform

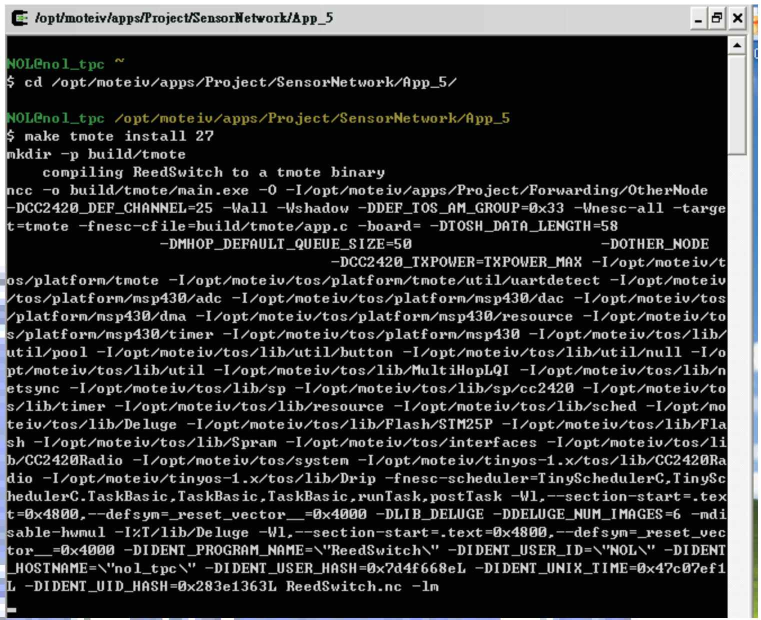

Figure 5.12: The development tools of smart home platform

Chapter 6

Conclusions and Future Works

Conclusions:

The smart home platform provides most applications including building a smart home environment. In this thesis, a testbed of smart home platform on WSNs is implemented. The smart home platform is divided as mote-based smart home application design, sensor network design and sink server design. Mote-based smart home application program is build in Taroko sensor hardware module. The smart home services are classified as home security, home care and home automation service. A modify the sensor network module is proposed for the smart home environment. The network module performs bidirectional routing behavior. The downstream routing path is created by sink mote via CrtForwardMsg message sending by sensor motes in the home environment. Home server and sink server are implemented to monitor the event of the home sensor and control the home appliance. The sink server is responsible for low-level packet parsing and sending and the home server performs friendly user interface and illustrates the status of the home environment. The home server also provides a development tool for smart home service program. The householder can easily construct a smart home environment by using the development tool.

Future Works:

The future works of this project will focus on network protocol, specific smart home service, and home server design. The network routing protocol should perform two ways packet transmitting in smart home platform. Moreover, for the sake of designing a robust smart home platform, it is better that some specific service motes can communicate with each other directly. There are other useful smart home service that can be developed in the platform, such as exactly position tracking, human behavior monitoring, etc.. The positioning technique in our smart home platform is symbolic positioning technique. The drawback is that it only indicates the room number in the smart home. Hence, if a positioning technique can be applied the smart home platform, some specific smart home services can be developed by all means. Home server is the central processing component of the smart home platform. Improving the feature of home server is the most benefited way for designing a faultless smart home platform. However, mote-based smart home services also need to operate with home server in coordination. For example, a universal plug and play interface can let any home appliance to be controlled if they are attached to the mote. Besides, a development tool on home server can provide householder a simply way to construct the smart home environment.

Bibliography

[1] ZigBee Alliance, Frequently Asked Questions, http://www.ZigBee.org/en/about/faq.asp, 2007.

[2] LAN-MAN Standards Committee of the IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE*, 2006.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor network: a survey", *Computer Networks*, 38(4), 393–422, March 2002.

[4] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*, 102-114, August 2002.

[5] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J.A. Stankovic, "An Advanced Wireless Sensor Network for Health Monitoring," In Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2), Apr 2006.

[6] Ilse Bierhoff, Ad van Berlo, Julio Abascal, Bob Allen, Anton Civit, Klaus Fellbaum, Erkki Kemppainen, Noemi Bitterman, Diamantino Freitas, Kristian Kristiansson, "Smart home environment," *Towards an inclusive future : Impact and wider potential of information and communication technologies*, 110-155, 2007.

[7] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, And Cybernetics - Part C: Applications and Reviews*, Vol.37, No.6,November 2007.

[8] TI Technology, MSP430, Available from: http://www.ti.com/msp430.

[9] Chipcon Inc, CC2420 DataSheet, Available from: http://www.chipcon.com/files/CC2420-Data-Sheet-1-3.pdf, 2005.

[10] Cygwin Solutions, Available from: http://www.cygwin.com,2007

[11] TinyOS Community Forum,Available from: http://www.tinyos.net,2007

[12] Joseph Polastre,Jonathan Hui,Philip Levis,Jerry Zhao,David Culler,Scott Shenker,and Ion Stoica, "A Unifying Link Abstraction for Wireless Sensor Networks," SenSys , November 2005.

[13] Alladi Venkatesh, "Smart Home Concepts: Current Trends," Center for Research on Information Technology and Organizations. I.T. in the Home, July 1 2003.

[14] Crossbow Technology Inc., "MICAz wireless measurement system," Available from: http://www.xbow.com/Products/productsdetails.aspx?sid=101.

[15] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. Int. Conf. Information Processing in Sensor Networks: Special Track on Platform Tools and Design Methods for Network Embedded Sensors* (IPSN/SPOTS 2005), 364-369, 2005.

[16] Crossbow Technology Inc., "MICA2 wireless measurement system," Available from: http://www.xbow.com/Products/productdetails.aspx?sid=174.

[17] Advantech Inc., "UbiQ Smart Home Solution," Available from: http://www.advantech.tw/eService-Applied-Computing/UbiQ/Default.aspx.

[18] B. Schilit, N. Adams, R. Want, "Context-Aware Computing Applications," IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, December 1994.

[19] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, E. Jansen, "The Gator Tech Smart House: A Programmable Pervasive Space," *Computer*, Vol. 38, No. 3, 50-60, Mar. 2005.

[20] Crossbow Technology, Micaz datasheet, Available from: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Kit_Datasheet.pdf, 2006.

[21] Crossbow .Technology,.Telosb datasheet, Available from: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf, 2005.