



**TEKNISKA HÖGSKOLAN**

HÖGSKOLAN I JÖNKÖPING

**A Smart Gateway Design for WSN Health  
Care System**

Yaoming Chen

THESIS WORK 2009

SUBJECT: Embedded System (Wireless Sensor Network)



# TEKNISKA HÖGSKOLAN

## HÖGSKOLAN I JÖNKÖPING

### **A Smart Gateway Design for WSN Health Care System**

Yaoming Chen

This thesis work is performed at Jönköping Institute of Technology within the subject area [Embedded System](#). The work is part of the university's three-year engineering degree. The work can also be a part of the master's degree. The authors are responsible for the given opinions, conclusions and results.

Supervisor: [Youzhi Xu](#)

Credit points: [30 points \(D-level\)](#)

Date:

Archive number:

---

Postal Address:

Box 1026  
551 11 Jönköping

Visiting Address:

Gjuterigatan 5

Telephone:

036-10 10 00

## **Abstract**

Using Wireless Sensor Networks (WSNs) in health care system has yielded a tremendous effort in recent years. However, in most of these researches tasks like sensor data processing, health states decision making and emergency messages sending are done by a remote server. Numbers of patient with large scale of sensor data consume a lot of communication resource, bring a burden to the remote server and delay the decision time and notification time. In this paper, we present a prototype of a smart gateway that we have implemented. This gateway is an interconnection and services management platform especially for WSN health care systems at home environments, by building a bridge between WSN and public communication networks, compatible with an on-board data decision system (DDS) and a lightweight database, which enable to make the patient's health states decision in the gateway in order to get faster response time to the emergencies. We have also designed the communication protocols between WSN, gateway and remote servers. Additionally Ethernet, Wi-Fi and GSM/GPRS communication module are integrated into the smart gateway in order to report and notify information to care-givers. We have conducted experiments on the proposed smart gateway by performing it together with a wireless home e-health care sensor network. The results show that it is reliable and has low latency and low power consumption.

## **Acknowledgement**

First of all, I would like to thank my supervisor Professor Youzhi Xu, who concerns for me so much during my entire master study and provided me with many opportunities to participate in international seminars. He is erudite and lenient. His great ability in grasping the research direction of wireless sensor network deeply inspired and educated me.

I would like to thank master program coordinator Alf Johansson for being always helpful and give me a chance to start working on my thesis in this year.

I would like to thank guest researcher Dr. Wei Shen and Dr. Hongwei Huo who give me so much good ideas in this thesis study and lead me the way to finish it. I would like to thank Dr. DongYang, who gave me a lot of help in the research works.

I would like to thank all my teachers and classmates who help me during my finial thesis study.

.

**Key Words**

Smart gateway,  
WSN,  
Healthcare system,  
GSM/GPRS,  
Socket,  
Communication protocol

# Table of Contents

<b>I</b>	<b>Introduction .....</b>	<b>1</b>
1.1	BACKGROUND.....	1
1.2	PURPOSE AND AIMS .....	1
1.3	DELIMITS .....	3
1.4	OUTLINE .....	3
<b>2</b>	<b>Theoretical background .....</b>	<b>5</b>
2.1	WIRELESS SENSOR NETWORK.....	5
2.2	IEEE 802.15.4 .....	6
2.3	TINYOS .....	6
2.4	GSM/GPRS .....	7
2.5	SOCKET INTERFACES .....	8
2.6	WSN FOR HEALTH CARE SYSTEM .....	8
2.6.1	Why Using WSN for Health Care System .....	8
2.6.2	WSN Health Care System Introduction.....	10
2.7	GATEWAY .....	11
2.8	RELATIVE WORKS.....	11
<b>3</b>	<b>Design and Implementation .....</b>	<b>14</b>
3.1	RESEARCH AND DEVELOPMENT METHOD .....	14
3.1.1	Research Method .....	14
3.1.2	ARM Linux System Development Processes .....	15
3.2	SYSTEM OVERVIEW .....	15
3.3	HARDWARE STRUCTURE.....	17
3.3.1	Base Station.....	20
3.3.2	LAN/WLAN Router .....	21
3.3.3	GSM/GPRS Module.....	21
3.3.4	Center Control Board.....	22
3.4	WORK MODEL .....	27
3.4.1	Simple Model.....	29
3.4.2	Intelligent Model.....	32
3.5	SOFTWARE STRUCTURE .....	38
3.5.1	Software Structure Overview.....	38
3.5.2	Data Decision System.....	39
3.5.3	Service Manage Platform .....	41
3.5.4	Database Manager .....	42
3.5.5	I/O Interface .....	42
<b>4</b>	<b>Experiment.....</b>	<b>46</b>
4.1	COEXISTENCE OF THE WSN AND WLAN.....	46
4.1.1	Experiment setup .....	46
4.1.2	Implementation .....	47
4.2	GATEWAY DATA SAMPLING EXPERIMENTS .....	48
4.2.1	Experiment setup .....	48
4.2.2	Implementation .....	48
4.3	SYSTEM DELAY MEASURE.....	49
4.3.1	Experiment setup .....	50
4.3.2	Implementation .....	51
<b>5</b>	<b>Conclusions and discussions.....</b>	<b>52</b>
<b>6</b>	<b>References .....</b>	<b>53</b>

<b>7</b>	<b>Appendix .....</b>	<b>55</b>
7.1	NFS .....	55
7.1.1	<i>NFS Advantages .....</i>	<i>55</i>
7.1.2	<i>Install NFS Server in Ubuntu.....</i>	<i>55</i>
7.1.3	<i>NFS Server Configuration.....</i>	<i>56</i>
7.1.4	<i>Configuration on develop board.....</i>	<i>56</i>
7.1.5	<i>Testing Your Configuration .....</i>	<i>57</i>
7.2	USB INTERFACE INITIAL FUNCTION .....	57
7.3	GT64 CONNECTION .....	<b>ERROR! BOOKMARK NOT DEFINED.</b>

## **List of Figures**

Figure 2-1 Typical Multi-hop Wireless Sensor Network Architecture.....	4
Figure 2-2 Linux system protocol stack with socket.....	8
Figure 2-3 Structure of Healthcare system interconnections.....	10
Figure 3-1 System development research method processes.....	13
Figure 3-2 Embedded system development processes.....	15
Figure 3-3 WSN health care system architecture.....	16
Figure 3-4 Smart gateway abstract architecture.....	17
Figure 3-5. Hardware structure of Smart gateway.....	18
Figure 3-6. (a) MIB520 USB interface board, and (b) interface board attaching with sink node.....	20
Figure 3-7. Photo of D-link DIR-301.....	21
Figure 3-8. Photo of GT64 with antenna.....	22
Figure 3-9. Photo of center control board.....	23
Figure 3-10 Embedded system software layout.....	24
Figure 3-11 Simple Model Architecture.....	28
Figure 3-12 Intelligent Model Architecture.....	29
Figure 3-13 gateway work flow in simple model.....	30
Figure 3-14. Work flow of gateway in intelligent model.....	32
Figure 3-15. An example of emergency SMS.....	37
Figure 3-16 Software architecture of smart gateway.....	38
Figure 3-17. System initialization process.....	39
Figure 3-18. Overview of decision process of HSM.....	40
Figure 3-19 process of data decision system.....	41
Figure 3-20. Processes of socket connection .....	44
Figure 4-1.Coexistence experiment setup.....	47
Figure 4-2.Temperature and humidity data in bathroom.....	49
Figure 4-3.Photo of GPRS Module, WSN Module, Flash DB, Center Control Unit and WLAN AP.....	50

## **List of Tables**

Table 3-1 Memory Partition Table.....	25
Table 3-2 Frame structure of message_t using in TinyOS 2.0.....	31
Table 3-3 Payload segment design for TinyOS 2.x.....	30
Table 3-4. Command packet format.....	32
Table 3-5 Data, Alarm and Question packet structure.....	35
Table 3-6 Health report packet structure.....	36
Table 3-7 Mote state report packet structure.....	36
Table 3-8 HCC command packet structure.....	36
Table 3-9. Emergency SMS structure.....	37
Table 4-1.Coexistence experiment result.....	48
Table 4-2.Time record of co-research relative to the bathroom.....	49
Table 4-3.Statistic of notification delay.....	51

# **I Introduction**

Because of the numerous advantages of Wireless Sensor Networks (WSNs), include wide coverage, low cost, low power, self-configuration and real-time data access, WSNs have been used in various areas such as military, health care, agriculture, environmental monitoring , industry, natural disaster prevention, wildlife tracking system, intelligent transportation, building monitoring, space exploration and other fields. It is considered to be one of the top ten technologies which will change the world in the future.

In this paper, we examine on of these research hot spots, which is using WSNs in health care system. Using WSNs in health care system which integrates wireless communications, health care and sensor network, have attracted a lot of research efforts in recent years.

We are going to discuss how a smart gateway will be used in a health care system, as well as how to design this smart gateway following the processes of embedded system development.

## **1.1 Background**

### **Home health care**

Home health care is the largest part of the Swedish elderly care. According to the statistics on October 1, 2008 around 140000 older persons received old age care and health care in their homes. In many municipalities, home care is the largest “enterprise”. This type of care is of utmost importance for old people to be able to live a relatively independent life. Most elderly also prefer to get help at home instead of at an institution. One of the main reasons why people cannot stay at their home is fear that they may have sudden illness or fall down and not able to get up. Therefore, they have to move to an old age care institution that they think would be safer. Many older people, especially in the higher ages live alone and their children and other relative might not live close to them. Today, relatively crude alarm systems are available but there is a great need of more sophisticated system that can monitor, support and alarm when needed.<sup>[3]</sup>

## **1.2 Purpose and aims**

The goal of this master thesis is to develop and evaluate a gateway for WSN home healthcare system in order to establish communication between WSN and Health Care Center (HCC), and meet the requirement of providing care-givers (home assistant /nurse /HCC /relative) with health state of the elderly.

In a health care system, time required to detect old people’s health state and time to send message to care givers are very important.

In the services view, requirements from smart gateway are defined as below:

1. For the elderly, any emergency situation happen to him or her should be detected and handled immediately.
2. For WSN, sensor nodes are initialized into an efficiency working mode and less interfere channel. Data from WSN is store and analysis in a proper way (by gateway or central server).
3. For HCC, it can get the elder's health state data on time. It should also have ability to maintenance the health state of sensor nodes in WSN.
4. For care-givers, they can get notice immediate when the elderly is in danger.

In order to achieve these requirements, these functional requirements are necessary for smart gateway:

**Requirements functions of smart gateway:**

Smart gateway is divided into two models: simple model and intelligent model.

In simple model, smart gateway has to:

1. Build up connection between WSN and smart gateway, between smart gateway and HCC.
2. Transform communication protocol between WSN and PCN (Public communication Network) in transport layer, also between WSN and HCC in application layer.
3. Receive data from WSN, and then transmit it to central server in HCC.
4. Implement commands coming from HCC.

In smart model, the smart gateway has to do these tasks:

1. Providing interconnection between WSN and smart gateway, and between smart gateway and remote server via public communication network (such as Ethernet, Wi-Fi, and GSM/GPRS).
2. Providing communication protocol transformation between WSN and various public communication networks.
3. Receiving sensor data from WSN, updating and storing them into on-board database
4. Providing a DDS to detect the real-time health state (normal, questionable, dangerous or oncoming dangerous) of the elderly base on the current received data and the historical data stored in the database.
5. When a dangerous or an oncoming dangerous state of the elderly health is detected, the gateway sends notifications to the remote server. Meanwhile, the gateway also sends an emergence SMS to all care-givers.

6. Providing a wireless access interface for care-givers to check the health states of the elderly on-site or for the system maintainers to check the system operating state with laptop or PDA.
7. Implementing response messages for the request requests from remote servers, such as health data report, sensor network and gateway configuration commands, etc.
8. Reporting various statistics of the elder's health state (like average body temperature, blood pressure) to remote server periodically.
9. Sending short and regular reports in a higher frequency to remote server with the main purpose to show both the elderly and the system in the normal state without any problems.

### **1.3 Delimits**

This smart gateway research is part of the WSN healthcare system project from the research group of Jönköping University. This research is still in the prototype stage, all data are obtained only through laboratory experiments, did not use the actual data.

The smart gateway is designed on the assumption that only one elder live in that house.

Size, price and level of integration of the modules are less considered.

### **1.4 Outline**

The rest of this article is structured as follows.

Followed by Introduction, we will in section 2 focuses on the theoretical background related to of the smart gateway. Chapter 2 also includes a universal health care system for wireless sensor networks design presentation, and related works of embedded gateway design.

Chapter 3 describes the detailed design of the gateway. This chapter contains the hardware and software design of the gateway, as well as two modes of the working pattern design.

Chapter 4 describes the process and results of the experiment. We designed various experiment to test the features of the smart gateway. These features include wireless sensor networks and wireless network compatibility issues, system data acquisition and storage experiments, as well as the gateway to send data packets and emergency messaging latency test.

In Chapter 5, we have summarized this smart gateway design, and pointed out the direction for future research work.

## 2 Theoretical background

### 2.1 Wireless Sensor Network

A Wireless Sensor Network (WSN) consists of spatially distributed individual sensor nodes which work together to monitor physical and environment parameters, and transmit monitor result to base station.

One of the common use structures of WSN is showed in figure 2-1. Sensor nodes are deployed into a monitoring region, and constitute a wireless ad-hoc network automatically. Sensor nodes support multi-hop algorithm, and together, they forward data packets to the base station. Some of the sensor nodes in network may be fixed at a certain position in order to monitor some unmoved parameters at its appropriate place. Some of the sensor nodes in the network may be install in the mobile object.

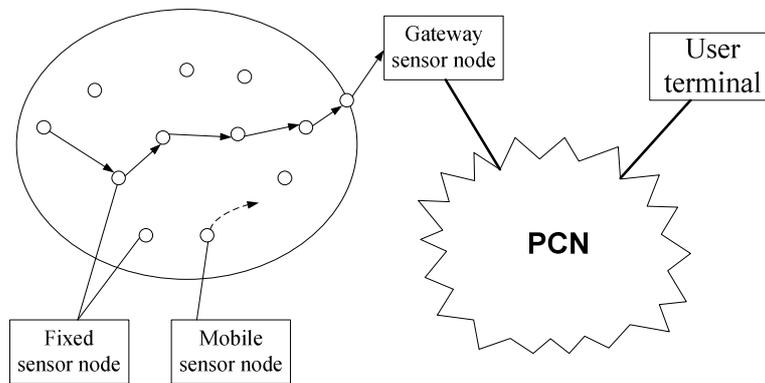


Figure 2-1. Typical Multi-hop Wireless Sensor Network Architecture

Sensor node typically consists of one or more sensors, radio transceiver, a small microcontroller, and an energy source, usually a battery. Its size can be as small as a grain and its price can down to a penny.

Unique characteristics of a WSN include:

- Limited power they can harvest or store
- Ability to withstand harsh environmental conditions
- Ability to cope with node failures
- Mobility of nodes
- Dynamic network topology
- Communication failures
- Heterogeneity of nodes
- Large scale of deployment
- Unattended operation
- Node capacity is scalable, only limited by bandwidth of gateway node.

Because of these superior characteristics, WSN now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control. <sup>[2]</sup>

Here in this thesis, we use WSN, with self-development standard, in health care system to monitor old people's health states.

## **2.2 IEEE 802.15.4**

Our WSN specification is base on IEEE 802.15.4-2006 standard which specifies the physical layer (PHY) and media access control (MAC) protocol for low-rate wireless personal area networks (LR-WPAN). It is maintained by the IEEE 802.15 work group. The targeted application for IEEE 802.15.4 is focus on low-cost, low-speed areas like wireless sensor network, home network, industrial control, remote monitor, building automation, and so on. Beside, these applications usually has low bitrates (up to some few hundreds of kbps), flexible, ad-hoc self-organize, not too stringent delay guarantees, and sometime low power consumption requirement.

The physical layer, based on direct sequence spread spectrum (DSSS), offers bitrates of 20 kbps (a single channel in the frequency range 868-868.6 MHz), 40 kbps (ten channels in the range between 905 and 928 MHz) and 250 kbps (16 channels in the 2.4 GHz ISM band between 2.4 and 2.485 GHz with 5-MHz spacing between the center frequencies). Even there are 27 channels available, but the MAC protocol use only one of these channels at a time. This standard is not a multichannel protocol. <sup>[1]</sup>

ZigBee, WirelessHART, and MiWi specification use the services offered by IEEE 802.15.4 and attempts to offer a complete networking solution by developing the upper layers which are not covered by the standard. Our designed specification also follows this standard (using 2.4 GHZ ISM band in physical layer and super frame structure with CSMA/CA protocol in MAC layer) and adds network construction (mesh networks), security, application services, and more.

## **2.3 TinyOS**

TinyOS is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools –all of which can be used as-is or be further refined for a custom application. TinyOS's event-driven execution model and fine-grained power management yet allows the scheduling flexibility made

necessary by the unpredictable nature of wireless communication and physical world interfaces. <sup>[5]</sup>

Sensor node used in our WSN health care system prototype all have installed TinyOS version 2.0. It uses message\_t structure which is tinyos-2.x standard message buffer. Sensor nodes use this message\_t structure both to form network route and send/received messages. We are going to discuss this message\_t structure in the later chapter.

## **2.4 GSM/GPRS**

Global System for Mobile Communications (GSM) is the most popular standard for mobile telephone systems in the world. According to the statistics of GSM World Association in 2009<sup>[16]</sup>, GSM standard is used in more than 80% of the global mobile phones and more than 4 billion users. GSM supports short message service (SMS) which is used in our gateway design. Release '97 of the GSM standard also added General Packet Radio Service (GPRS) capability.

General Packet Radio Service (GPRS) is a packet oriented mobile data service support by GSM standard. It is a method of enhancing 2G/3G phones to enable them to send and receive data more rapidly. GPRS standard support both TCP/IP protocol and point-to-point protocol (PPP). With a GPRS connection, the phone is "always on" and can transfer data at any moment, and at higher speeds: typically 32-48 kbps. An additional benefit is that data can be transferred at the same time as making a voice call. GPRS is now available on most new phones. <sup>[14]</sup> GPRS is widely used in modern industry. GPRS uses data terminal to access into internet, and provide customers with stable, high-speed, always-on, low-cost data transmission channels. Therefore widely used in a variety of remote data transmission and monitoring system.

In the control areas, the traditional wireless data terminals usually use the system architecture with Micro Control Unit (MCU) and GSM/GPRS modules. By the hardware, computing capability constraints of this kind of terminal, the overall function is weak, especially in the network protocol development and support, both with considerable difficulty. In recent years, with ARM as the representative of the embedded 32-bit microprocessor technology has made rapid development, many high-performance ARM microcontroller chip has almost the same in both power and the hardware cost. So in many industrial applications, the use of ARM chips to replace the traditional 8/16 bit microcontroller to work with GSM/GPRS module is already a very economical, ideal choice.

## 2.5 Socket Interfaces

Socket is a communications middleware abstraction layer between the application layer and the TCP/IP protocol suite, which is a set of interfaces. In the design mode, Socket hides the complicated TCP/IP protocol suite behind the Socket interfaces. What users have to do is use this group of simple interfaces and let Socket to organize data in order to comply with the specified protocol. Figure 2-2 shows the Linux system protocol when using socket interfaces.

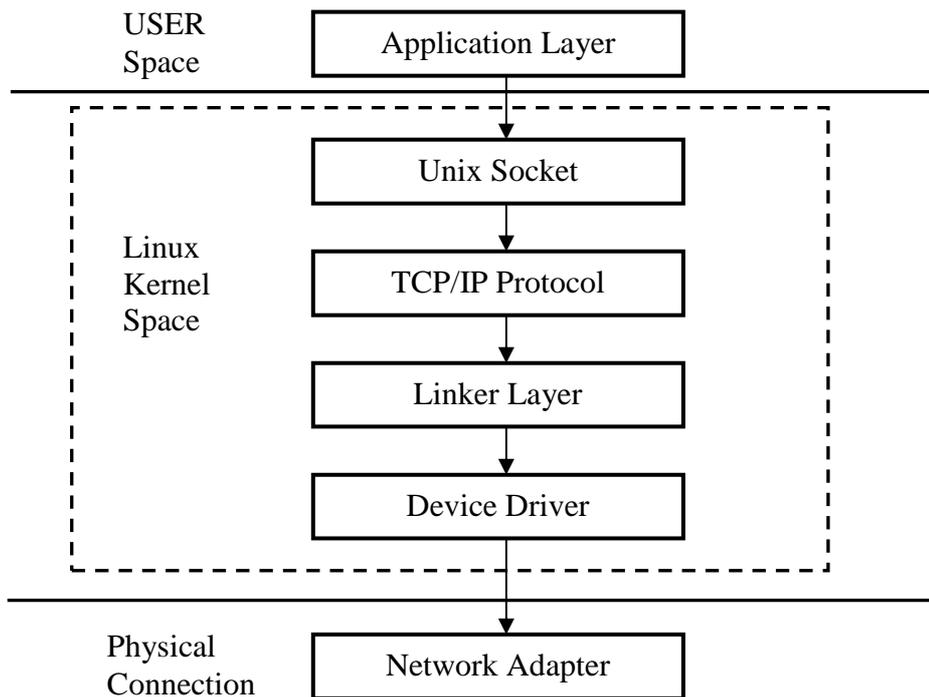


Figure 2-2 Linux system protocol stack with socket

Smart gateway has to transmit data through Ethernet and WLAN which are TCP/IP networks. In this design, we use socket interfaces to implement send and receive data through these TCP/IP networks.

## 2.6 WSN for Health Care System

### 2.6.1 Why Using WSN for Health Care System

#### 1) WSN vs. video cameras

Comparing to video cameras, multiple sensors provide plenty of different type of information with quantified parameters about the environment and health state in which alarm and emergent signal is much easier to be detected, and

alarm signal can be generate automatically and immediately, without the help of people. Video camera needs powerful processor, high data rate and large data memory comparing to sensor node, therefore the price and power consumption is much higher.

One advantage of using video cameras is that it is able to provide real-time picture, but some people must keep surveying at the video and further more, this may cause a privacy problem.

## 2) WSN vs. wire network

The uppermost advantages of WSN compare to wire network are mobility and flexibility. In WSN, sensor nodes are no longer bound with the fetters of cable. Eliminating the need for a large number of wiring, WSN would be much easier to set up. We don't have to re-wire everything even when the deployments of sensor nodes need to be altered in the future. Sensor nodes can communication anywhere within the coverage of wireless network; health information of the patient can be transmitted indoor or outdoor; sensors are even able to attach on the body of the patient and move with them. Wireless sensor network is also highly scalable. The coverage of WSN is easy to expand with a new sensor node. You just need to plug it in, and then it would joint the network without any extra setting.

## 3) WSN vs. Bluetooth

Same as WSN, Bluetooth communication is also design for short rang, low power wireless intercommunication between two wireless devices. The main drawbacks of Bluetooth compare to WSN are:

- First, Bluetooth does not support ad-hoc. Devices in Bluetooth network communicate in a master-slave mode. Master device has to spend more power to choose hopping sequences and the active slaves before starting the communication.
- Second, active slaves' number limited to not more than seven. This will limit the performance of sensor nodes.
- Third, the active slave must always be switch on, waiting to be polled by the master.
- Last, devices need tight synchronization when trying fast frequency hopping operations.

## 4) WSN vs. WLAN

WLAN (wireless local area networks) using IEEE 802.11 is designed for high speed communication of multimedia devices. This technology is used in present-day network and will course more power consumption. It is suitable for the equipments which have higher processing speed and more memory. Using TCP/IP protocol in sensor network is also high overhead. In sensor network

application, we need a low data rate, low power consumption technology like IEEE 802.15.4 to fit the resources limited sensor nodes.

## 2.6.2 WSN Health Care System Introduction

Research group in Jönköping University has been research on the WSN health care system field for several years and has made remarkable achievement and great contribution on it. This group have established a relatively comprehensive WSN Health Care System theory and have developed a prototype to do experiment on it. Our gateway designs as part of Health care system base on the existing WSN Health Care System prototype in Jönköping University.<sup>[4]</sup> See figure 2-3. This prototype consists of three parts.

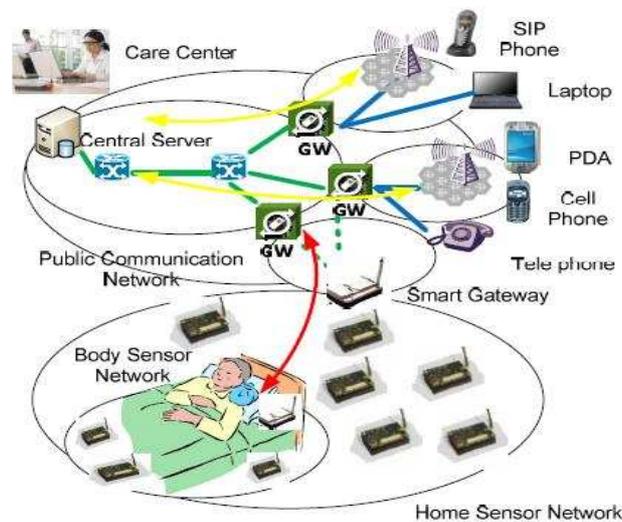


Figure 2-3 Structure of Healthcare system interconnections

The first part is monitoring sensor networks. It consisted of body sensor networks (BSN) and home sensor network (HSN). BSN are sensor nodes attached on old people's body and provide the on body sensing information of him or her. HSN are group of location-fixed sensor nodes with multiple sensors providing temperature, relative humidity, light sensors, microphone, and so on. HSN is distributed in living room, bedroom, kitchen, bathroom and corridor hiding in the sofa, bed or chairs.

The second part is the access devices, which mean gateway. This gateway is usually a base station of the home sensor networks, which provide an interconnection between the monitoring sensor networks and central server system via PCN. It has to handle the connection problem between multi-network communications.

The third part is central server. It can be divided into four sub-parts: a conceptual database, a decision mechanism, a smart application gateway and a

service management platform. The conceptual database is used to store the profiles information of the elders, the normal data collected by the sensor systems, the detection result, and report or alarm logs. Decision mechanism use Hidden Markov Model to detect the elder's coming action and health state. Once the dangerous situation is detected, the decision mechanism will drive the smart gateway to issue an alarm to the relatives to report the emergency situation. The service management platform is an interface of the whole system. The smart application gateway can establish communication with the caregivers to report the situation of the elders. <sup>[4]</sup>

This WSN health care system prototype has been test in Jönköping University lab and has obtained the expected results.

## **2.7 Gateway**

A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, or signal translators as necessary to provide system interoperability.

In the health care system, central server in HCC, which is “far away”, can not send requests or receives any data directly from WSN. That is because central server is not within the coverage of WSN. It can only access services of a WSN through Public Communication Network (PCN). But sensors nodes in WSN do not have the necessary protocol component at its disposal to communicate with PCN and thus requiring the assistance of a gateway.

Gateway acts as a proxy for the set of sensor nodes in WSN. It represents all WSN to answer the requests from HCC. In this way, on one side, sensor node does not have to know the existence of HCC and can dispense with all mechanisms connect to PCN. On the other side, central server in HCC does not have to handle multi-hop routes in the sensor network any more to find the exactly node.

With the rapid development of embedded system hardware, gateway becomes more and more powerful due to better and better resources e.g. faster control unit and bigger memory. These make it possible that some tasks which originally belong to the central server can be transferred to the gateway. In this way, we can move the processing intelligence closer to the sensing device.

## **2.8 Relative Works**

As what we have discuss above, Gateway design is very important for the whole healthcare system. WSN gateway design has attracted a lot of research effort in recent years and a number of articles have been published. There are many noteworthy aspects when designing WSN gateway, e.g. energy

consumption, memory space, reacting speed, data latency, protocol compatibility, and so on.

In book [1], base concepts of WSN gateway are bring forward, including existence signification, basic roles, network address translation, multi gateway selection, grouping sensor nodes, locating specific sensor node and how to integrate WSN with general middleware architecture, etc. These are part of essential questions we must consider about before starting to design our own gateway.

Paper [4] has commendably summarized the WSN health care system. In this paper, a prototype of WSN health care system is build up. Construction of the whole system and function of each part has been described in detail. From this paper, we can clearly see the interconnection between WSN, gateway and public communication networks. This system structure is mainly use for reference in our whole system design. Further, for as much as this, more detail about the gateway design and a novel gateway working model will be discussed in our thesis.

In some case, like disaster management, combat field reconnaissance and secure installations [6] [7], gateway placement is important due to sensors number is large and they are miniaturized working with small battery. In paper [8], genetic algorithm for hop count optimization and genetic algorithm for distance optimization method are used in order to select the best sport for placing gateway for each group of sensor nodes so that sensor nodes' data can be delivered to gateway with the least latency. For our thesis, gateway is designed for an old people living in his home, which usually is a small size compare to disaster locale or a combat field. Latency produce from distance between of sensor nodes would not be much correspondingly. Moreover, location of the gateway is usually not an arbitrary option in an old people's home. But we can try to adopt the hop count optimization proposal when doing simulation in laboratory.

In some case, gateway, on one hand, has to communicate with different WSNs with different protocol for data acquisition, routing, and various applications; on the other hand, has to communication with multi TCP/IP protocol such as IPV4 and IPV6, to compatible the need of modern network. In order to achieve this requirement, authors in [9] have designed a gateway with a configurable engine for protocol translation. This gateway is based on the application-level gateway <sup>[10]</sup> concept, in which the gateway contain all the protocols for both networks and protocol translation happens in the application layer. Services and protocols for both WSN and Internet side are declared as extern and export modules, so that these services and protocol are able to be selected and configured and updated on line remotely. Further more, gateway designed in this way can connect heterogeneous networks and provide protocol and service translations to different combination of protocols on both sides of the gateway. Building network protocols as models and configured and updated on line are

really very good features for a gateway design that it can improve the gateway compatibility. We can consider it as one of the future work.

In some situation, WSN in health care system give a large quantity of real-time vital signs to calculator computer. It is not a good idea to do the calculation by only one computer. In paper [12], authors suggest to use Grid computing technology to analysis the vital signs collected from WSN and deriving results. So authors designed and implement a SensorGrid gateway to connect the WSN and Grid network. Unfortunately, authors did not clearly described protocol conversion process. The preconditions of using Grid computing are existence of Grid computers and high speed and stable network connection. If these preconditions are not satisfied, it is different to test if your gateway design is good or not. Further more; grid computing requires great financial and technique support. If the vital signs can be analysis in the gateway, system can be more economical and stable. In our gateway design, we suppose we have already finished the data decision system and use it to finish this job in a single gateway.

For health care system, recognizing old people's health state is a very important work, and at the same time, a very difficult work. In paper [11], author suggests to use Hidden Markov Model to recognizing the actives of the old people. Beside the data from sensor nodes, circumstances information like older people's current location and identity, activity and time are use to form the context categorization. Sensor nodes data are aggregated and form this categorization then given to Hidden Markov Model. Old people's states and activities are defined in the system and the output of deduced conclusion is one of these state and activities. Hidden Markov Model is one of the most important algorithms of the data decision system in our smart gateway design. Use it the abstract health state of the elderly, the influence of measurement errors and transmission errors can be reduced. It is one of the research projects in our research group and it is now in prototype stage. In our gateway design, we consider combining Hidden Markov Model into the gateway design as one of the most important future work.

An online sensor data access service in the gateway can provide the users with great convenience. Authors in [17] have proposed a lightweight approach for interacting with networked devices. Base on the success of Web 2.0 mashups, they tried find a way to maximize reuse the existing shared devices using standard Web technologies. This design is a loosely coupled infrastructure for Web of Things and gateway is able to access sensor nodes through a RESTful interface [18]. Implementing a web-based interaction and management in a gateway can also be one of the future works to provide remote data access to the WSN.

### 3 Design and Implementation

#### 3.1 Research and Development Method

##### 3.1.1 Research Method

This thesis design follows the system development research method <sup>[13]</sup>. The systems development approach denotes a way to perform research through exploration and integration of available technologies to produce an artifact, system or system prototype. System development focuses on the theory testing, more than theory building aspects of research, allowing a smooth progression from development to evaluation. Base on these, system development research method is properly for researching and testing our gateway prototype. System development research process is an iterative nature. Combined with our actual situation, the smart gateway research process is illustrated in Figure 3-1.

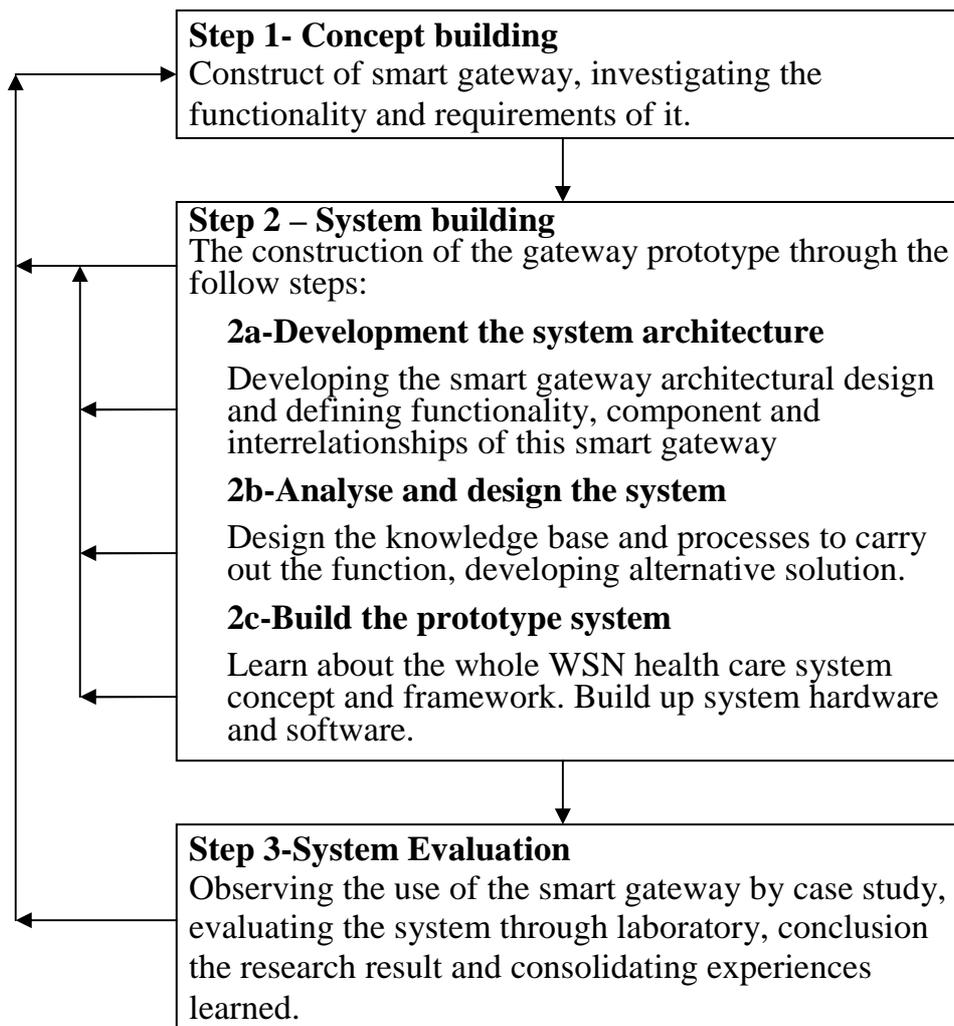


Figure 3-1. System development research method processes

In this section, we are going to follow this research method processes and discussed about the smart gateway design in detail.

### 3.1.2 ARM Linux System Development Processes

General embedded systems development methodology is shown in figure 3-2. When the programmers begin to develop an embedded system based applications, first, develops in a host computer with the development tools which relative to the embedded devices you chose, and then use the software simulator or evaluation board for debugging, and finally generate the image file on the host computer, and programmed into the stand-alone embedded products.

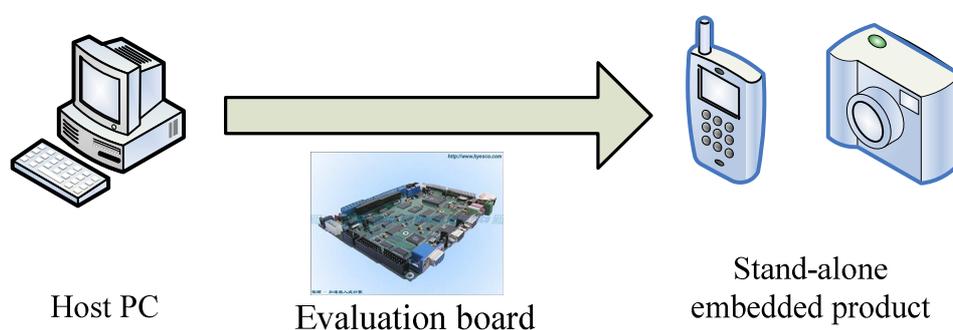


Figure 3-2 Embedded system development processes

Generally, there are three steps when developing the ARM embedded system

- (1) Objectives hardware system development. Like Microprocessor, memory, and peripherals selection etc.
- (2) Basic system development. This includes system boot up program development, kernel porting, file system development, hardware drivers development, etc.
- (3) Application development.

In this design, we follow this development processes to build up the smart gateway system.

## 3.2 System Overview

Base on the system development research method, it is important to investigate the functionality and requirements of the smart gateway at the beginning. On one hand, the requirements analysis takes the system specification and project planning as a basic starting point of activities analysis, and performs inspection and adjustment from the software point of view; on the other hand, the

requirements specification is the main basis of software design, implementation, testing and maintenance.

The WSN health care system architecture is illustrated in figure 3-3. The whole system consists of four parts. The first part is the monitoring object; means elder's home where sensor nodes are probed to get multiple data of old people likes behave, activity, health state, and living environment information. Health care system has many old people, so this monitoring object is a plural. The second part is the monitor - Healthcare center. Main job of healthcare center is to monitor health state of all old people in this system and make sure the normal operation of the entire system. Center server is located there to save necessary information of the elderly and provides variety of monitoring methods to indicate the current situation of the elder. Third part is care-givers including doctors or nurses in the hospital and the old people's relatives. They are responsible for dealing with the report message (normal or alarm message, from internet or SMS) that sent to them. They can also check the elder's current state through webpage which provide by Healthcare care. The fourth part is Public Communication Network (PCN) including Internet, GSM/GPRS, Ethernet, and WI-FI. PCN connects all the other three parts together.

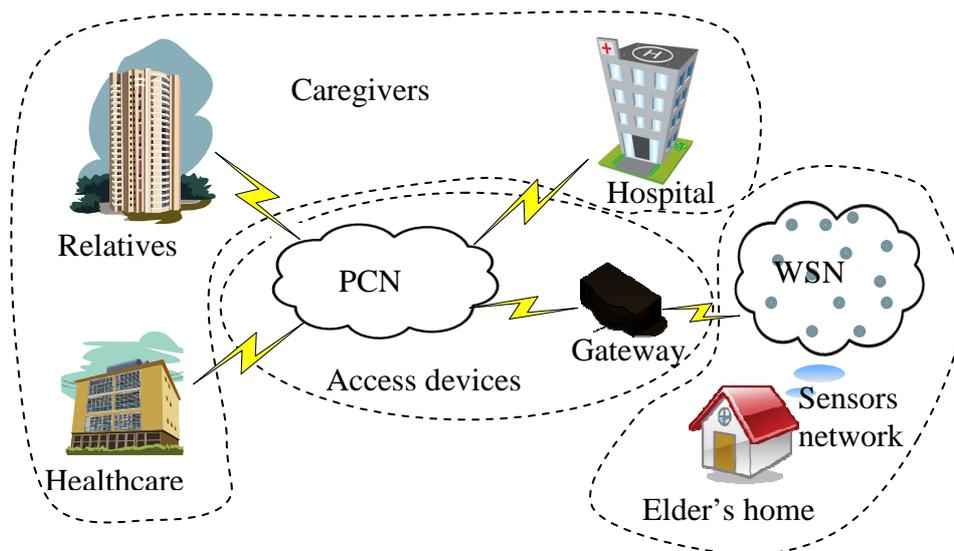


Figure 3-3. WSN health care system architecture

From this figure, we can see that gateway belong to part one (monitoring object), and it acts as bridge between WSN and PCN, offering multi communication ways to make sure required message can be transmit to desire destination.

Detail requirements of smart gateway design are as follow:

- 1) Providing a mechanism to connect the WSN base station (sink) and received sensor collection data from it or send command to WSN.
- 2) Providing DB to save sensor data.

3) According to the current input sensor data and data stored in DB, determine the current health status of the elderly.

4) Sending notify to HCC periodically through internet if the old people's state is normal. Send urgent notify to HCC through Internet and SMS to caregivers through GSM network.

5) Providing WLAN connection for short distance wireless devices like PDA and laptop so that they can access to the DB and receive notify.

6) Receiving and execute commands from HCC.

7) Providing management software to control the working flow of all the requirements that mention above.

8) Providing fast enough processor, big enough memory and adequate external ports for these system requirements.

### 3.3 Hardware Structure

In the previous section, we have mentioned the requirements of smart gateway. First thing to do is, base on these requirements, consider the whole smart gateway as a black box so that we can see the external interface of it clearly.

For hardware design, these requirements can be grouped into three different categories. This can be reflected in figure 3-4

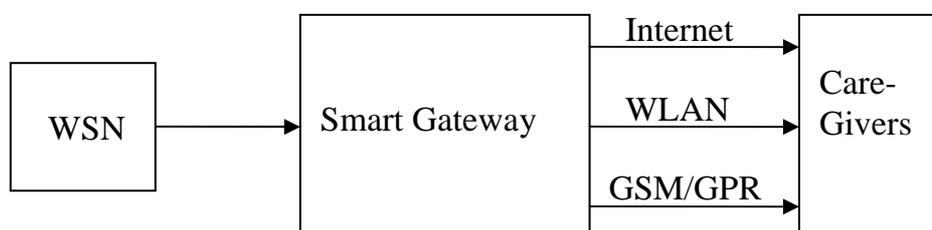


Figure 3-4. Smart gateway abstract architecture

Combining this figure with the above-mentioned requirements, the design of the gateway can be divided into three parts. WSN connection mechanism, interconnection section and center control unit. See figure 3-5

The first part is WSN connection mechanism. Smart gateway has to receive data from sensor nodes and gives commands to them. To achieve this goal, smart gateway has to provide mechanism to join into the WSN. Through this part, smart gateway provides connection to WSN in both hardware and software ways. At hardware way, smart gateway has to handle interface compatible and signal translation and rate conversion. For software, it has to translate the protocol using in WSN and extract the valuable data which can be used by the program using in smart gateway.

Our smart gateway design is developed from the WSN healthcare system prototype from the research group of Jönköping University, so the WSN module should be compatible with the existing WSN used in the prototype. To do this, there are many WSN modules we can choose to connect to the sink node. Sensor nodes in this prototype are MICAz MPR2400 manufactured by Crossbow Company. Crossbow Company provided three basic interface boards with different types of connecting ports to connect with MPR2400 nodes. They are MIB510 Serial Interface Board, MIB520 USB Interface Board and MIB600 Ethernet Interface Board. It is fast and easy to match the existing WSN in the prototype if we choose one of these three interface boards as the WSN module. Comparing MIB520 and MIB510, the former one uses a USB interface while the latter one uses a serial port. Relatively, USB has these better characteristics like: higher data transfer rate, plug-and-play, smaller size, and larger number of ports. MIB600 provides Ethernet connectivity to a LAN connected to a host device. But if we use MIB600, the host device, which means the smart gateway in our situation, must include the DHCP function. This will take more memory space, making the system structure more complicated and consuming more power. After comprehensively considering all these, we design to use MIB520 as the WSN Module.

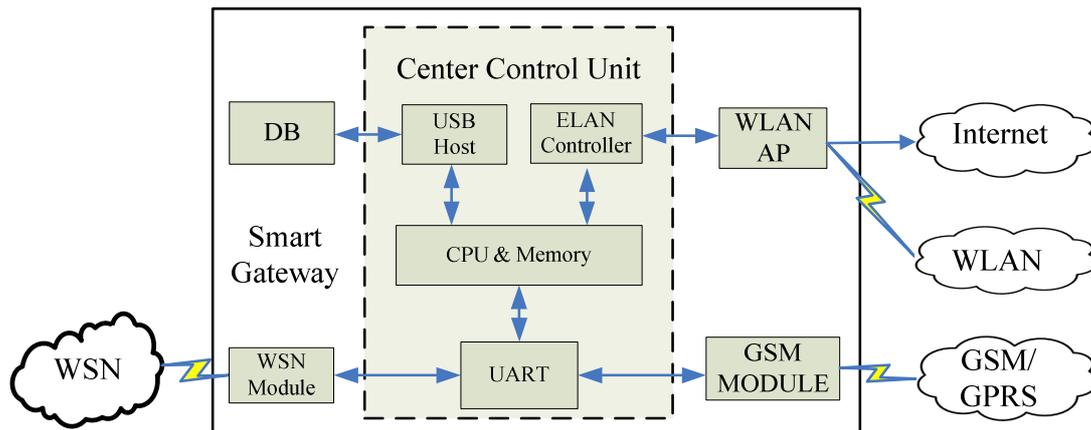


Figure 3-5. Hardware structure of Smart gateway

The second part is interconnection mechanism. In this part, smart gateway provides physical connection to Internet, WLAN and GSM/GPRS network and finishes protocol and signal translation and rate conversion.

For Internet connection, an ELAN controller is needed. It can be provided either by an external unit or integrated in the center control unit. On the basis of this controller, coupled with the WLAN Access Points, then the smart gateway is able to connect to a WLAN. This WLAN AP allows our smart gateway to connect to a wireless network using Wi-Fi to the destination host within the wireless signal coverage of this AP.

There are two steps to connect to internet and WLAN. First step, smart gateway uses the IP address provided by the server in the internet. Second, smart

gateway provides DHCP to build up WLAN with an extra wireless AP. These two steps can be implemented by center control unit or a powerful wireless router. When this task is completed by center control unit, system will have a higher integration and the wireless AP can be simpler and smaller size. If a powerful wireless router is used, center control unit will have simpler OS structure and can focus on data processing more. This can also reduce the design duration time for prototype building up. The disadvantage is an extra router will have bigger size and consumes more power. In order to speed up the calculation time and shorten the design duration time, we chose D-Link Company's DIR 301 wireless router as our wireless AP in our smart gateway prototype design.

A GSM/GPRS module is also needed to connect to GSM/GPRS network and send SMS and GPRS data to caregivers. In our design, we chose GT64 GSM/GPRS terminal to achieve these connections.

WSN module, WLAN AP, and GSM module together, we call them external communication modules (ECMs).

The third part is center control unit. This is the most important component of smart gateway, and we need a cost-effective, low power, high performance control board to finish this job. Requirements to the center control unit are as follow:

- Enough storage space

Considering the operation system kernel, input/output buffer for communication, develop program, also DB to store all messages come from WSN, center control unit require a large capacity memory. Generally, embedded system board does not have a large storage space, so center control unit has to support high speed external memory.

- Fast enough processing speed

In this health care system, gateway needs to process messages send over from 10 to 20 sensor nodes. All these messages have to be computed in the DDS system to determine the current state of the old people. Concurrently, the two-way communication with HCC must be processed in time. So the smart gateway must have fast enough computation speed for these tasks. Memory allocation and program optimize also have to be considered to speed up the system. We will discuss this in later section.

- Adequate external interfaces

As show in figure 3-5, center control unit has to control both WSN Module and all interconnection modules. Furthermore, take into account of the need of USB memory. These require the center control unit to provide enough

necessary hardware interfaces (including RS232, USB, Ethernet) to all these modules at the same time.

In this smart gateway design, we use a S3C2410 develop board as the center control unit.

### 3.3.1 Base Station

In our smart gateway design, MIB520 USB Interface Board is used as the WSN connecting module. MIB520, which show in Figure 3-6(a), manufactures by Crossbow Company. The MIB520 provides USB connectivity to the MICA family of Motes, which use in WSN of the Jönköping University prototype, for communication and in-system programming. The USB bus connection provides 56.7K baud rate, sensor node on board programming interface and power surprise to the devices.



Figure 3-6. (a) MIB520 USB interface board, and (b) interface board attaching with sink node

MIB520 USB Interface Board uses FTDI FT2232C chip which allowed the gateway to use USB port as virtual COM port. Host device requires a FT2232 chip driver to communicate with MIB520. With this driver, the host driver can read and write the USB bus easily like a serial port. The MIB520 offers two separate of these COM ports to finish two main functions.

First port is dedicated to in-system Mote programming. Through the MICA-series connector, MIB520 has an on-board in-system processor (ISP)—an Atmega16L located at U14—to program the sensor node. Code is downloaded to the ISP through the USB port. Next the ISP programs the code into the sensor node. This required the host device, which it connected to having Mote Works/TinyOS installed. It is too much load for smart gateway, so this work is usually given to the host PC.

Second port is used for data communication with host device over USB. MIB520 connects to WSN by attaching the sensor node to its MICA-series

connector. Any sensor node can function as a base station when mated to the MIB520CB USB interface board. See Figure 3-6 (b). After a sensor node is attached to MIB520 and become a base station, it will detect all messages which have destination address of host device and send it over MICA-series connector to MIB520, then go through USB bus to host device.

### 3.3.2 LAN/WLAN Router

In order to reduce the load of center control unit and build up the prototype fast, a D-link DIR-301 wireless G router (See figure 3-7) is chose to connect smart gateway to Internet and WLAN.



Figure 3-7. Photo of D-link DIR-301

The D-Link Wireless G DIR-301 Router which is capable of transferring data with a maximum wireless signal rate of up to in the 2.4GHz frequency. Indoors wireless operation ranges up to 328 ft. (100 meters).

The DIR-301 fully compatible with the IEEE 802.11b and 802.11g standard, so it can connect most wireless connection devices like PDA or intelligent cell phone which compatible with these two standard.

The DIR-301 provides up to 54Mbps wireless connection with other 802.11g wireless clients. The performance of this 802.11g wireless router gives you the freedom of wireless networking at speeds 5 times faster than 802.11b. This capability allows caregivers and maintainers, who come to the old people's home to check the operation of the healthcare system, to participate fast connect to the smart gateway. This router also offers four Ethernet ports to support multiple computers.

### 3.3.3 GSM/GPRS Module

The GT64 which used in smart gateway design is a Wavecom based Quad band GSM / GPRS programmable Terminal. The GT64 Terminal is an intelligent

GSM/GPRS control terminal that encapsulates everything necessary for the wireless Mobile to Mobile (M2M) communication including SMS and GPRS class 10<sup>[14]</sup> that we use in our smart gateway design. Figure 3-8 show GT64 terminal with a RS232 Serial cable and a GSM 1800/1900 MHz antenna.



Figure 3-8. Photo of GT64 with antenna

GT64 terminal can be used as a standalone and powerful GPRS modem with its intrinsic TCP/IP stack. The GT64 has an integrated standard SIM card holder and connectors, like Serial and Mini USB connection, which means less need for additional hardware. The numerous inputs and outputs, which are common used can be reconfigure by command setting, provide for additional functions and features to make the M2M solution innovative and cost efficient. In our design, RS232 Serial Interface is used to communication with the center control unit.

GT64 is a programmable telemetry device. In connection with the M2mpower Software-Editor, it allows us to program our own application special for health care system and store it on the GT64 Terminal and thus minimizes the need for extra components. It is compatible with the standard AT commands which is widely used in modem control. This makes the terminal easy to control and reduce the time to build up the prototype.

### **3.3.4 Center Control Board**

In our smart gateway design, we use a S3C2410 development board to implement the center control unit. See figure 3-9.

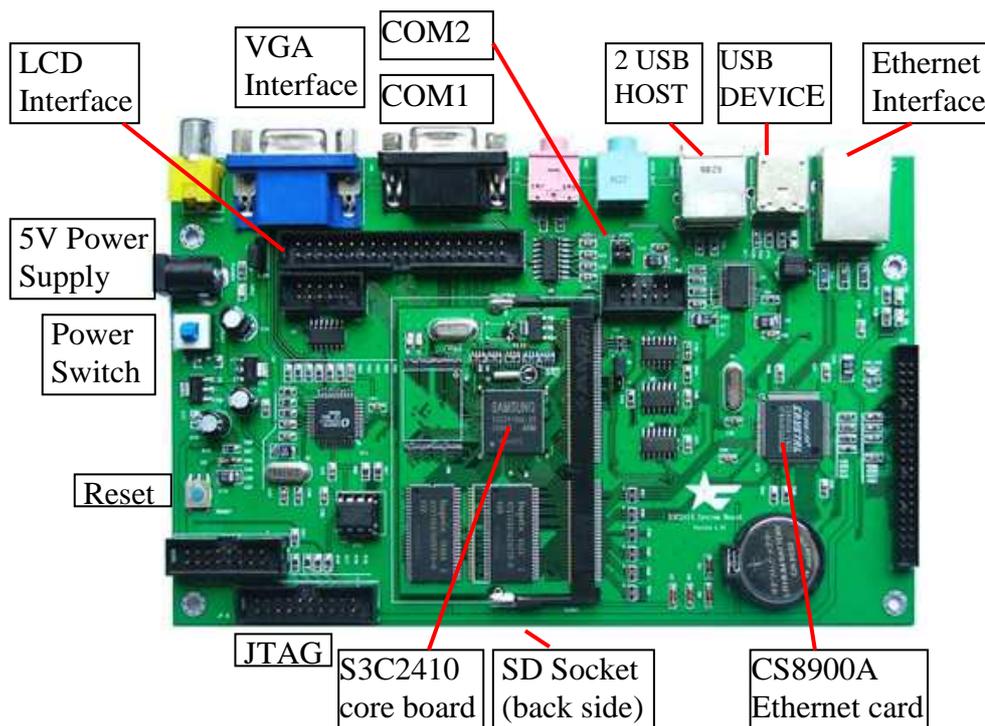


Figure 3-9. Photo of center control board

This S3C2410 development board uses SAMSUNG S3C2410 core board. S3C2410 microprocessor is designed to provide hand-held devices and general applications with cost-effective, low-power, and high-performance microcontroller solution in small die size.

S3C2410 microprocessor has a wealth of features and high price-performance ratio. It is developed using an ARM920T core, 0.18um CMOS standard cells and a memory compiler. The ARM920T implements MMU, AMBA BUS, 5 steps pipeline and Harvard cache architecture with separate 16KB instruction and 16KB data caches, each with an 8-word line length. Supports the ARM debug architecture, provides 200MHz (maximum 266MHz) operating frequency in standard mode, and 64-way set-associative cache with I-Cache (16KB) and D-Cache (16KB).

S3C2410 core board we use in this design provides 64MByte SDRAM and 64MByte NAND Flash onboard, 32bits bus width and 100MHz front side bus. This guarantee all application programs can be store onboard and have fast executed speed.

This development board has abundance of peripherals. Ethernet controller (CS8900A) providing a RJ45 10BASE-T Ethernet interface to help the smart gateway to connect to local computer or LAN; Two RS232 Serial UART ports for GSM/GPRS terminal connection and computer console; Two USB host for MIB520 module and external memory to DB; JTAG debugging / programming

interface; SD card interface offer one more option for external memory. Interfaces all located in the side of the circuit board, facilitate the users directly installed in the chassis.

### 3.3.4.1 ARM Linux System Development Processes

An embedded system development is separated into several layers. Different layer have different task. Figure 3-10 show the typical development sequence of an ARM Linux system.

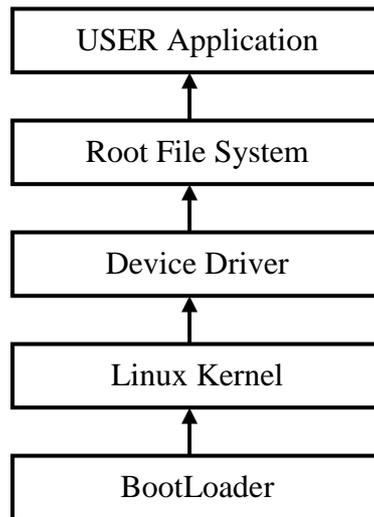


Figure 3-10. Embedded system software layout

Bootloader lay at the bottom of an embedded system software layout, mainly responsible for arm board boot up initialization. Follow by the embedded system kernel and kernel is device driver. Finally, root file system and user application.

Before we start our smart gateway application developing, all those lower layers must be established, design to fit for our health care system design and port into the NAND flash onboard.

### 3.3.4.2 BootLoader

Bootloader is the first paragraph of the code that runs after the system power-up. It is equivalent to the BIOS of PC. To put it simply, bootloader first load the Linux kernel from memory to RAM. Then it initializes all the necessary hardware devices on board. During this step, some messages which are required by the system kernel are created and passed to kernel through relevant mechanisms. This will bring the system hardware and software environment to a proper state. Last thing it does is system test and give the control of the development board to Linux operation system.

In our S3C2410 development board, we use the open-source bootloader program VIVI, which is developed by MIZI Company in South Korean. Base functions of this VIVI are as follows:

- Download image file, like OS kernel and root file system, into memory through Serial port or network.
- Setup the system boot up parameters.
- Initial hardware and boot up the operation system.
- Memory partition and bad block detect.
- Boot up delay setting.

When doing memory partition, one thing should pay attention to is the application space must be big enough. In some of the embedded system, root file system and user application are put into two different partitions. This is will waste some space of the memory and is not suitable for our smart gateway design since our program is big. Also, size of other partitions should be set to not too much bigger than its original size. After measuring the size of VIVI, system boot up parameters and Linux kernel, partition table for our smart gateway design is set to be as show in Table 3-1.

Table 3-1 Memory Partition Table

Name	Offset	Size	Flag
VIVI	0x00000000	0x00020000 (128KByte)	0
Param	0x00020000	0x00010000 (64KByte)	0
Kernel	0x00040000	0x001C0000 (1.768MByte)	0
Root	0x00200000	0x03CF8000 (60.992MByte)	16

### **3.3.4.3 Linux Kernel Tailoring and Kernel Porting**

Linux kernels support many different hardware platforms or also call hardware architectures, like ARM, MIPS, I386, and Alpha etc. All supported platforms are stored in the file names “arch”. Each of this architecture contains number of sub-system. If the hardware architecture is not support in the Linux kernel, you have to find or build a patch match your hardware architecture and the using Linux kernel. A lot of code modify is needed also. Therefore, the selection of a suitable kernel becomes the first task.

Samsung S3C2410 has become a standard Linux support platform in ARM architecture. Linux kernel can run very well in the S3C2410 target board without any patch. After Linux 2.6.11, Linux kernel starts to include its own NAND flash partition information and support Yet Another Flash File System (YAFFS) which is the only file system, under any operating system, that has been designed specifically for use with NAND flash. YAFFS not only speeds up the loading speed of the file system, but also increase the file access speed and lift the maximum file size limited. Both of NAND flash and YAFFS will be used in this design. In our smart gateway design, we decided to use Linux 2.6.14 kernel.

Normally, the Linux kernels we can download from internet are for X86 architecture. They can not be used directly to an ARM system. Porting a Linux operation system from X86 architecture to development board usually includes three steps. First, establish the cross-compiling tool chain in PC because C compiler can not be use in ARM architecture. Second step is kernel tailoring, recompile, and porting; and also the necessary hardware driver porting. Third step is user application porting and realize the missing libraries.

The kernel is the core of the system software. Kernel porting is a complex task and also a very important process for embedded system development. Even though S3C2410 core has been supported by Linux 2.6.14 kernel, but we still have to do a number of modification base on the actual hardware configuration of our specify development board. These following steps have to be done:

1. Add NAND Flash support into kernel code and configure it. A partition table has to be setup for kernel compiled and will be later use in development board starting up. This partition table must be same as the one we set in VIVI.
2. Driver for Ethernet controller CS8900A and YAFFS are not included in this kernel. We have to download it, put it into the corresponding folder and modify all relative head files to include them into the optional table manually.
3. Linux 2.6.14 kernel code can not handle two USB ports using at the same time well due to the incorrect setting of MISCCR register. So we must modify the code of it. This code is enclosed in appendix.
4. Kernel options configuration. Linux kernel source code contains a lot of optional modules (more than 100 optional modules). How to choose between these modules in order build a small size and full-featured kernel which meet the requirements of our smart gateway become a key issue. Choose these modules carefully base on the need of the smart gateway with the principle of saving power consumption and space. Pay attention to Network File System (NFS), FTDI support. They are not

included in default setting. The reason of why we need NFS is discussed in Appendix.

After the kernel is configured, it can be compiled, and then use HyperTerminal or JTAG downloaded it to the development board.

#### **3.3.4.4 Root File System**

Root filesystem is an important component for Linux system booting up, and also necessary for normal operation of the operation system. Kernel code image files are stored in root filesystem. When system booting up, kernel will load the root file system to RAM from NAND flash, and then mount the necessary system devices into it.

The original Root Filesystem installed in development board is a Compressed RAM File System (CRAMFS). This file system follows Filesystem Hierarchy Standard (FHS) and was made by BusyBox 1.0. Standard folders like *bin*, *dev*, *etc*, *lib*, *usr*; and common commands like *ls*, *vi*, *cat*, *mount*, *tar* etc are included in this filesystem.

One shortcomings of this filesystem is that the C language library using in it is version 2.2.2. This doesn't meet the requirements of the SQLite and DDS in our gateway design in which they require the glibc having at least version 2.3.2. In order to save the time in building a new root filesystem, instead, I extracted all file from this original CRAMFS system to PC; updated all the link files in folder *lib* to the version 2.3.2. Instead to compress this new filesystem to CRAMFS again, I decided to use YAFFS this time. As already mentioned above, YAFFS can speeds up the loading speed of the file system, increase the file access speed and lift the maximum file size limited. Actually, we will wait until all user applications development are accomplished, and then compress it into the new filesystem at one time. During development, we use NFS to test our program.

### **3.4 Work Model**

Smart gateway acts as a communication bridge between WSN and Public Communication Network. It stands in the centre of data transmission of Health care system. Even through many researchers are committing themselves to apply WSN in medical application, but there is still neither industry standard nor protocol for using WSN to Healthcare system until now. Here, base on the prototype of WSN health care system research by Jönköping University, we have work out a set of corresponding communication protocol for between gateway and WSN, and between gateway and public communication network.

In order to make better use of the original system, while developing a more efficient and more reasonable system, our gateway design is divided into

simple model and intelligent model. From the perspective of the overall system, these two models are based on two complete different concepts.

In simple model, gateway acts only as a protocol translator and data transmitter. Architecture of simple mode is show in figure 3-11. It just connects to the WSN on one side and HCC on the other, and transmits data packets between these two sides without storing them. Elderly health status monitoring, data storing, mote status monitoring, and send alarm and question message to hospital and relative works are pass to the central server.

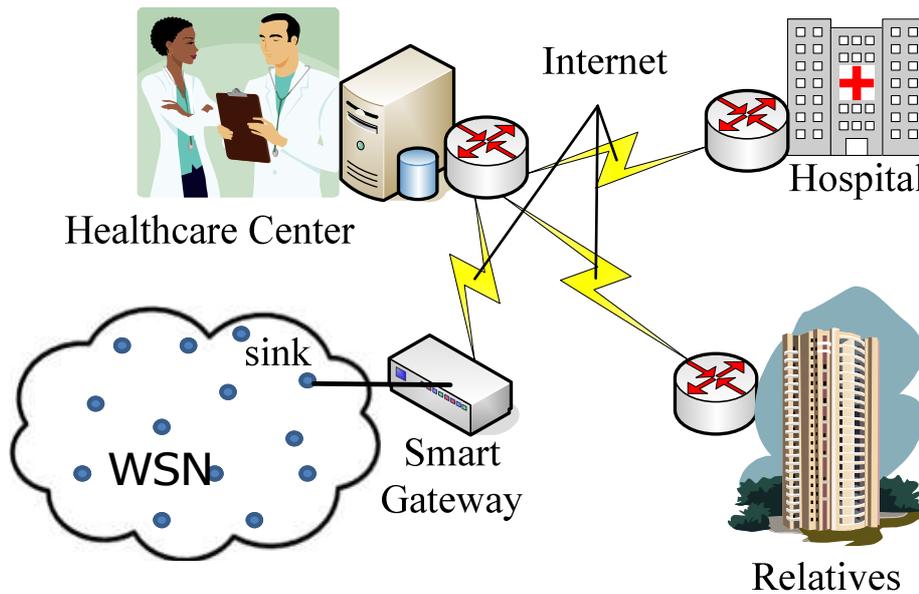


Figure 3-11 Simple Model Architecture

Intelligent model is on the contrary to it. Smart gateway becomes the central part of the whole system. It takes responsible for most of the data processing works, like data analysis and storage, elderly status, mote status and WSN status monitor works are distributed to it. In normal situation in intelligent model, gateway sent health status report to HCC regularly. Alarm and question messages will also be sending to hospital and relatives by smart gateway when those situations are detected. Further more, gateway in intelligent model also provide WLAN access capabilities to facilitate the use of system maintainers and elder's relatives to check the operation status of the smart gateway by laptop or PDA when they are closed to the smart gateway. Architecture of WSN Healthcare System in intelligent model is show in figure 3-12.

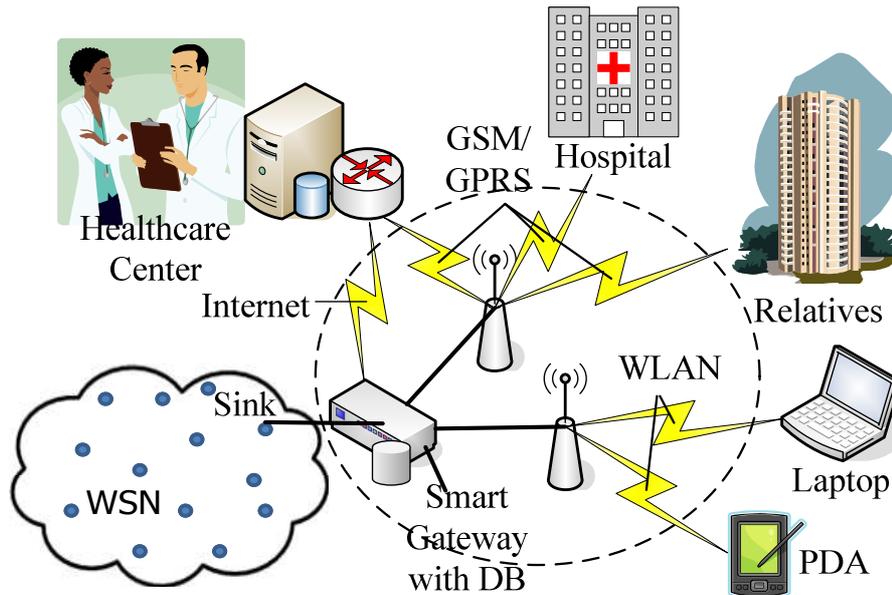


Figure 3-12 Intelligent Model Architecture

The idea of simple mode is that to have all data stored in the central server, so that HCC can get the latest state information of the elder quicker and also easier to do the statistical analysis to the health state of elder who live disperse. Simple mode is also designed to compatible the original health care system. However, due to the correspondence between HCC and elder is now one-to-many, that is, HCC monitor multi number of elders at the same time. Therefore, under simple mode, central server processing speed and storage space requirements will increase rapidly with the increase in the number of elder. Meanwhile, importing large amounts of data to central server from disperse gateways will be a burden on transmission network. Long-distance data transmit will also increase the error rate. In the intelligent mode, long-distance network data transfer will be reduced significantly, and central server's workload can be reduced. Since the data analysis is bring forward to the smart gateway, alarm and question problem can be detected earlier and the relevant people can be notice sooner.

### 3.4.1 Simple Model

#### 3.4.1.1 Simple model work flow

Generally, there are three entities in this health care system: HCC, gateway and WSN. The intercommunications between these three entities are shown in figure 3-13. In this figure, the internal functions of entities are hidden, and the communications between the entities are highlighted.

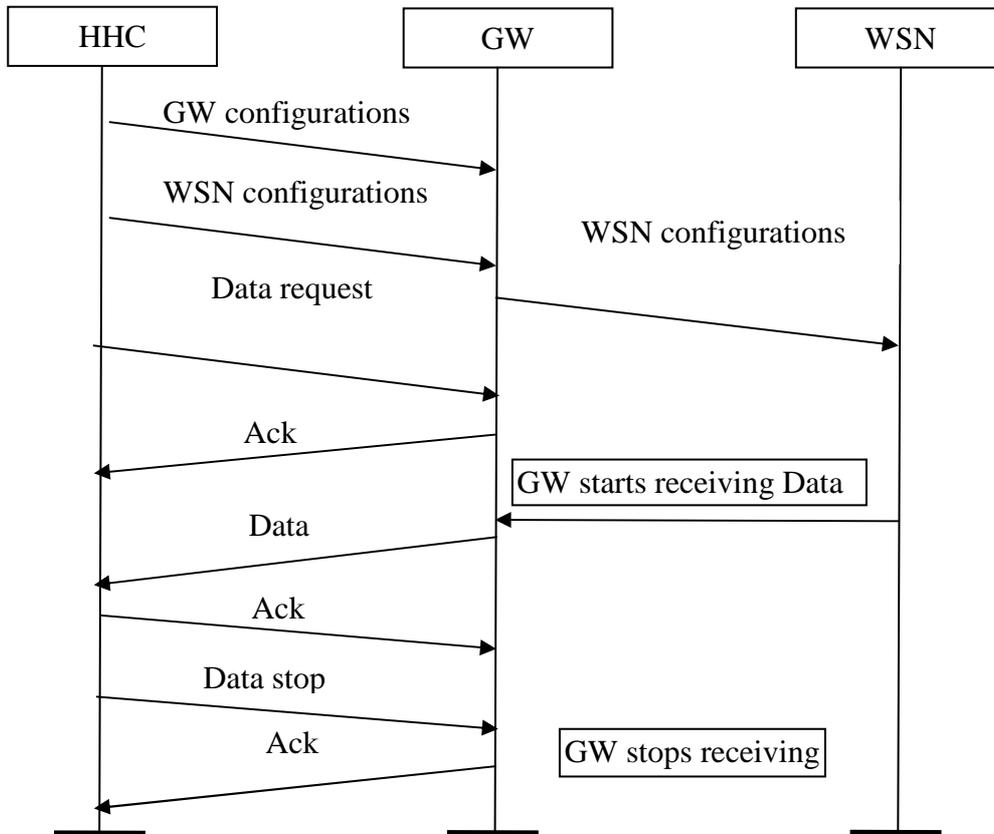


Figure 3-13 gateway work flow in simple model

When gateway is configured to simple model, it is only a data transmit tool and protocol translator. From then on, it mainly completes two tasks.

On one hand, gateway offers physical connection interfaces for WSN sink, which we have discussed about at session 3.3, and receives data packets from the WSN. Then gateway starts to do the interpretation to the received data packet and extract useful information in it, and stored them into send waiting buffer. Meanwhile, the gateway keeps checking whether the send waiting buffer is empty or not. If not, it read the data from the send waiting buffer, packaged it into a TCP / IP packet, and sent it to the central server of the HCC through internet.

On the other hand, gateway also offers interface for waiting commands come from HCC to WSN. Those commands are used to control the sensor nodes' working state in order to reduce interference in WSN or save power of sensor nodes. These commands packets will not be issued in a high frequency, so the gateway does not need a big receive and send buffers for them. To handle the commands packets, gateway only has to explain the TCP / IP packets from HCC, and then sent directly to the sink in serial format.

**3.4.1.2 Packet format between WSN and gateway**

Sensor nodes used in this research have TinyOS-2.0 installed. In TinyOS-2.x, the standard message buffer is called message\_t. Frame structure of message\_t is shown in table 3-2.

Table 3-2 Frame structure of message\_t of TinyOS-2.0

Content	Destination address	Link source address	Message length	Group ID	Active message handler type	Payload
Size(bytes)	2	2	1	1	1	User design

Frame head (from destination address to active message type) is used for packet routing in WSN. They are useless to the HCC. Only the “Payload” segment is useful for smart gateway or HCC.

We consider the “Payload” segment design is very important to the smart gateway design because, first, it realizes the functions of WSN healthcare system, and second, the data package has the largest transmits number in the WSN. Its designed significantly affects the overall system functionality and efficiency. We designed the Payload format is as shown in table 3-3.

Table 3-3 Payload segment design for TinyOS-2.0

Content	Mote ID	Counting	Data Type	Data value	Parent ID
Size (bit)	16	16	8	16	16

We try to keep this payload segment design as efficient as possible. “Mote ID” segment indicates the packet sender’s id. “Counting” segment is the packet sequence number or time stamp. “Data Type” identified the type of data packets. We have considered nine different kinds of data packet for both home sensor network (HSN) and body sensor network (BSN) in our health care system prototype including heart-beat rate, temperature, microphone, Received Signal Strength Indication (RSSI) etc. “Value” segment represents the data type value got by the sensor. “Parent ID” segment indicates the mote ID of the sensor node of the last hop in its transmission path.

Experiments in chapter 4 proved this payload design meets the requirements of data communication between nodes in WSN.

**3.4.1.3 Command packet format from HCC to gateway**

There are four types of command from HCC to gateway.

One command is to configure the gateway’s work model which means to change the gateway between simple model and intelligent model.

The other three types of commands are used to configure WSN. Gateway receives these types of commands, transfers them to serial format, and sends them to the sink node. Three commands are defined as follow:

- (1) Change the running channel between the motes of WSN.
- (2) Change working/resting time percentage of a specified node.
- (3) Change the packet sending rate of a specified node.

Among them, command 1 will issued only when the central server detected that the current channel interference used in WSN has overrun the permissible limits. HCC issues commands 2, 3 according to the activities frequency of the elder. For example, when the elder is sleeping in his bedroom, we can reduce the working/resting time percentage and packet sending rate of the pressure and sound sensors which located in the sofa in living room, and so on. The purpose is to reduce the node's power consumption. The command packet has the structure like table 3-4.

Table 3-4. Command packet format

Content	Mote ID	Command Type	Value
Size (bit)	16	8	16

One thing has to be noted that is different types of sensor node send packet to smart gateway in different frequency. It is easy to understand that like the pressure sensor in the sofa do not need as high packet frequency as the blood pressure one. And the packet frequency has to be changed due to some special situation that we mention above. By this packet structure, smart gateway can changes the packet sending rate of a specific sensor node. The node number can put into "Mote ID" segment, "Command Type" tells this is a command to change the packet sending frequency of the node and the "Value" is the required frequency.

### **3.4.2 Intelligent Model**

As what we had discussed above, smart gateway become central part of the health care system in intelligent model. We consider the smart gateway as the proxy of all sensor nodes in this elderly home. On one hand, data packets do not send to HCC directly anymore. Smart gateway provides Data Decision System (DDS) to sort and analysis all data send from WSN and detect the elder's current health state. The results of DDS will be stored into the database in smart gateway. On the other hand, smart gateway provides functions to handle the requests come from HCC, sent reply back by the statistic results in the database.

#### **3.4.2.1 Intelligent model work model**

Generally, there are three entities in this health care system: HCC, smart gateway and WSN. The intercommunications between these three entities are

shown in figure 3-14. In this figure, the internal functions of entities are hidden, and the communications between the entities are highlighted.

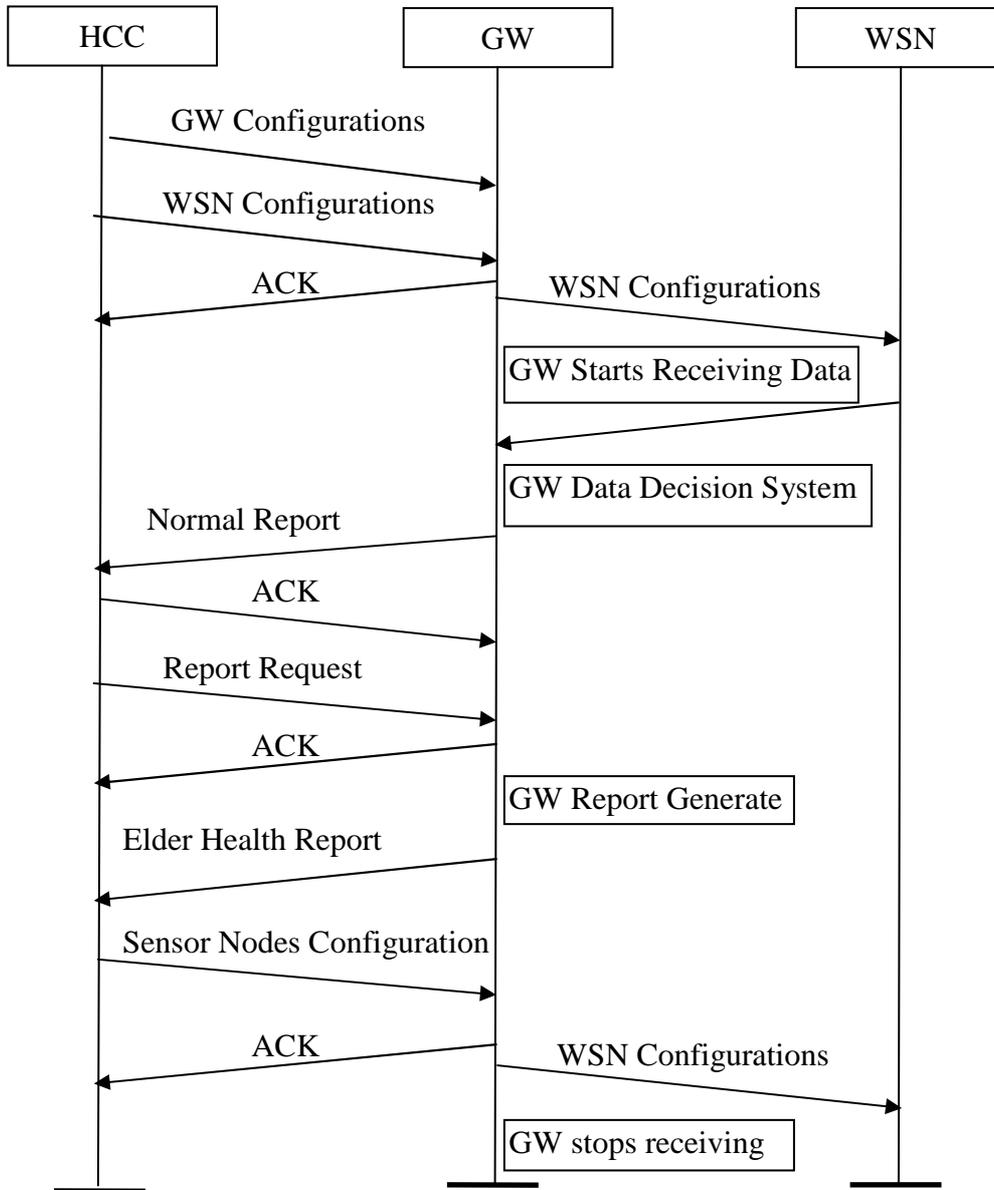


Figure 3-14. Work flow of gateway in intelligent model

From this figure, we can see that the HCC and smart gateway are communication in hand shake protocol. At the beginning, Gateway and WSN boot up with their default configuration. Then HCC sends both gateway configuration commands and WSN configuration commands to initial the gateway and WSN. After the WSN is configured, it starts to collect sensed data and transmit to gateway. And DDS in the gateway starts to monitor the elder's current health state. If no questionable and dangerous states are detected, gateway will periodically sends normal state packets to the remote server. The remote server gives ACK to each of this packet. When necessary, the remote server will send commands to gateway asking for the sensor data during a

specified time period. To response this, gateway generates the report packets by accessing to the database, and sends them to the remote server. The remote server may also send commands to control changing some configurations of the sensor nodes, as we have discussed in section 3.4.1.3. The gateway treats this kind of commands in the same manner as the initialization requests and forwards them to the WSN directly.

In intelligent model, we have designed special packet structure for the communication between WSN and gateway, and between HCC and gateway. These packet structures design is significant because a rational design of these packet structure can reduce data flow rate, so that the burden on the channel will be reduced and furthermore to improve communication efficiency. Main task of smart gateway in application layer is packet conversion between WSN and HCC.

General speaking, base on the transmits direction and role of packets, the gateway has to handle five different kind of packets including packets from WSN to gateway, packets from gateway to HCC, packets from HCC to gateway, packets form gateway to WSN and packets from gateway to Cell phone.

#### **3.4.2.2 Packet from WSN to gateway**

In intelligent model, WSN sends packet to smart gateway in the same packet structure as it did in simple model which including packet head and payload. See table 3-2.

In intelligent model, smart gateway has to analysis the packets come from WSN, so the packet structure has to be clearly understand. As we have discussed above, sink node of the WSN is attached on the MIB520 USB interface board to form the smart gateway WSN module. When the “Destination address” segment of the packet is 0, which means this packet should be sent to sink node, this packet will be passed to MIB520 interface board and then sent to the center control unit through USB cable in the serial communication format.

In the serial communication format, packet is wrapped on both ends by reserved byte “0x7E”. The “0x7E” reserved byte is a frame synchronization byte which used to detect the start and the end of the packet. If the data byte in the “Payload” segment is the same as this reserved byte, this data byte will be extended into two bytes. An escape byte “0x7D” will be used preceded and the payload data will be execute logical OR operation with byte “0x20”. For example, when data in the payload is “0x7E”, it will appear in “0x7D 0x5D” in the data packet.

### **3.4.2.3 Packet from gateway to HCC**

#### 1) Data, Alarm and Question packet

Smart gateway will sent five types of packet to HCC. They are health state data packet, alarm packet, question packet, report packet and node state packet.

We decided to use the same packet structure for data packet, alarm and question packet. Packet structure is shown in table 3-5. Every segment may have different meaning in data, alarm and question packet.

For data packet, “Packet Type” segment indicates this is a data packet. “Type” includes humidity, RSSI, temperature, microphone and heart-beat rate etc. We have designed nine different kinds of data packet which will sent from smart gateway to HCC, corresponding to those nine data packet types from WSN to smart gateway. “Value” is the average date value of this type within its own period time.

For alarm packet, “Type” segment indicates alarm types. Form example, we can use “0x01” to represent a fire alarm, and use “0x02” to represent heartbeat rate is out of limit, and so on. “Value” can be the position that this alarm case happened or the alarm sensor value which overrun the limit.

Question packet is formed when smart gateway found something which can affect the health of elderly, but not for sure, happen to him. Question will be send to all care-givers to let them make the judgement. For question packet, “Type” represents the number of question types. “Value” is can be the relative value of this case, some time can be considered as the severity of the case. For example, we can use type “0x01” and value “60” to represent “The elderly has take shower for more than 60 minute”. Or use type “0x02” to represent “The elderly did not have breakfast (no activities in the kitchen from 4:00-12:30)”. At this case, the “Value” segment can be ignored.

Table 3-5 Data, Alarm and Question packet structure

Content	Packet Type	Type	Value
Size (bit)	8	8	16

If the elder’s state is normal, smart gateway sends routine health state information to HCC periodically through a certain number of data packets. These packets tell the average value of elder’s health state parameter or living condition (such as heart beat rate, blood pressure, temperature and room humidity). According to actual demand, data packets from smart gateway has different packet sending period. For example, in normal situation, smart gateway only has to send room temperature data packet to HCC every one hour, but heartbeat rate should be up to every 10 minutes. Data packet is the most common and frequent packet between gateway and HCC. Refined and efficient packet structure is required to improve the performance of the gateway.

## 2) Health report packet

Health report packet is used when smart gateway is making health status report of the elder to HCC. Smart gateway sends health report to HCC at a fixed time every day or whenever HCC sends a command to ask for it. To report the health status of the elder, smart gateway has to send a group of this report packet. Report packet structure is shown in table 3-6.

“Packet Type” indicates this is a report packet. “Type” represents the content of the packet. “Value Type” indicates what type of value is given by this packet, including minimum, maximum, average, how many times or how long time. For example, we can use type “0x01”, value type “0x01”, value “80” to represents “minimum heartbeat rate 80 times/min”. Or use type “0x02”, value type “0x04”, value “480” to represent “bedtime 480 minutes”.

Table 3-6 Health report packet structure

Content	Packet Type	Type	Value Type	Value
Size (bit)	8	8	8	16

## 3) Mote state report packet

Mote state report packet is used in two cases. First, when smart gateway reports the sensor nodes’ state of WSN to HCC. We call it normal mote report. Normal mote report, like the health report, is send at a fix time every day or whenever HCC sends a command to ask for it. A group of mote state packets are required each time to finish normal state report. Second, when smart gateway found one of the sensor nodes is not in good state. We call it alarm mote report. Alarm mote state report can be multi number of packets at one time. Each mote state packet tell the state and battery value of one sensor node. Mote state packet structure is shown in table 3-7.

Table 3-7 Mote state report packet structure

Content	Packet Type	Mote ID	State	Battery Value
Size (bit)	8	16	8	16

### **3.4.2.4 Packet from HCC to gateway**

Packet from HCC to smart gateway is used to transport commands. These commands are divided into two categories: Configuration commands, report request commands. Configuration commands are used to configure the gateway and the WSN nodes. We have discussed them in section 3.4.1.3.

Report requests command is divided into two types. First, request for health report. Second, request for mote state report. The packet structure is shown in table 3-8.

Normally, HCC would not send these requests to gateway and just wait for the data packet and health report packet. These commands are decided for some the query function.

Table 3-8 HCC command packet structure

Content	Request	Value
Size (bit)	8	16

#### **3.4.2.5 Packet from gateway to WSN**

Packets from smart gateway to WSN remain the same as it is in simple model which are 1) change the running channel between the motes of WSN; 2) change working/resting time percentage of a specified node; 3) change the data sending rate of a specified node. And packet structure is the same too. See table 3-4.

#### **3.4.2.6 SMS from gateway to cell phone**

SMS from gateway to cell phone is decided for alarm report and question report mechanism. As what we have discussed above, when question case is found, smart gateway sends SMS to each care-giver in HCC and the elder's relative. When alarm case is found, smart gateway will also send SMS to hospital directly. SMS structure is show in table 3-9.

In SMS PDU coding 7-bits pattern, one message can contain maximum 160 bytes. According to the SMS structure, we can see that we are actually sending at least 2 messages in one time.

Table 3-9. Emergency SMS structure

Content	Name	Tel num	Time	Address	Problem head	Value
Size (Char)	40	30	40	50	100	int

An example emergency short message may look like figure 3-15.

Alarm Message from WSN Healthcare System  
 Name: Karan Albert  
 Telephone: 080 4872930  
 Message time: 2009-03-12, 12:23:53  
 Address: karhoksgatan 30, 221-43 Stockholm  
 Problem: Body temperature over 40 degree.

Figure 3-15. An example of emergency SMS

### 3.5 Software Structure

The software of smart gateway realized the health care functions and provides smart gateway management. In this section, we are going to describe the detail software architecture of the smart gateway using in smart model. This architecture was designed with three main goals: feasible, reliable and high efficiency. Feasible and reliable make sure that the gateway provides stable connection to WSN and health care center. Efficiency so that the packet delay and network burden can be keeps as low as possible.

#### 3.5.1 Software Structure Overview

The software implementation of this smart gateway is written in C language except a part of the VIVI bootloader is written in Assembler. The overall software structure of the smart gateway is shown in figure 3-16. It can be divided into four parts including I/O interface, data decision system (DDS), service manage platform (SMP), and database manager. Each of these parts is responsible for a set of well defined tasks.

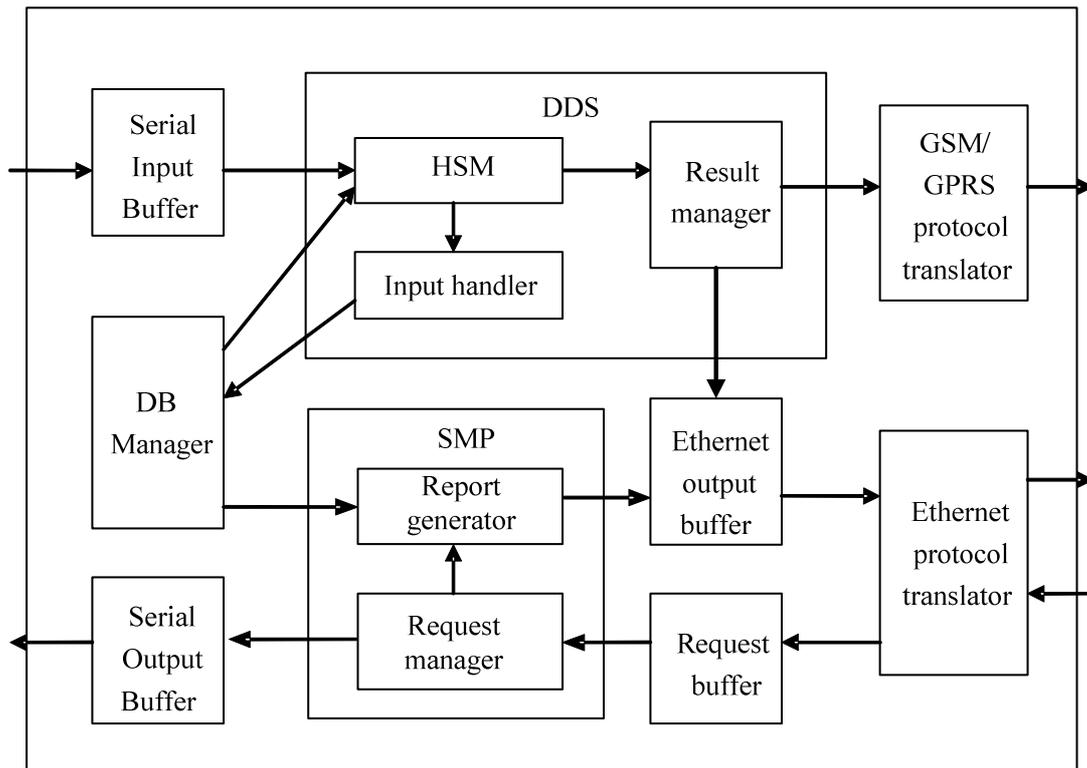


Figure 3-16 Software architecture of smart gateway

When starting system initialization, the first step is parameters initialization. This step mainly completes the DB parameters setting, buffer size defining, and processes controlling. Then the main program turns into modules start-up phase. Modules show in figure 3-17 are start-up sequentially. Each Module is started in an individual system process. Their roles are to assign the functional

modules port address and establish the connection. Among these Modules, serial communication module means both WSN base station and GSM terminal communication. WSN base station and GT64 terminal follow the POSIX standard. They can be initiated and connected with the standard serial port read/write function providing by Linux OS. Meanwhile, Linux socket communication is used by the Ethernet protocol translator. After the start-up phase is complete, the main program goes to waiting state. All the functions of smart gateway are completed by these started modules and the intercommunications between them.

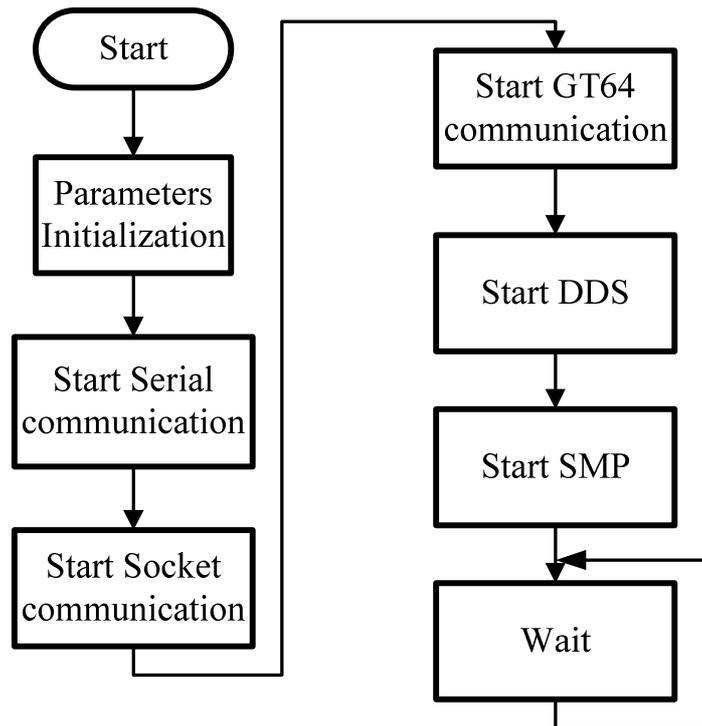


Figure 3-17. System initialization process

### 3.5.2 Data Decision System

Elder's health state detection is the most important function in a health care system. Whether we can achieve this function well is directly related to the overall performance of the gateway. In this paper, we aim at design an intelligent data decision system which determines the elder's current health state by comprehensive consideration of the current received sensor data and elder's last health state. The data decision system (DDS) consists of three parts: Health State Monitor (HSM) and input handler and result manager.

HSM play a role of elder's health state decision. In this design, we assumed that the elder's activity set is a finite set. With the help of medical researchers, all these activities can be obtained. We also assumed that we have deployed enough sensor nodes to react to all these activities.

All sensor data are collected and send to HSM. The received packet format has been introduced in section 3.4.1. First step the HSM do is rearrange the in coming data. “Mote ID” and “Data type” segments are considered to divide the coming data into different Categories. Second step, HSM reads the Specific numerical of “Value” segments in the same category and abstract it back to the value that the program can deal with. After these new data are rearrange and abstract, elder’s current position and activity (such as sleeping in bed or cooking in kitchen) is determined. From the database, HSM can fetch the elder’s health state and activity over a period of time. We call them Pass State. Elder’s current health states is decided by HSM through an integrated consideration of current position, activity and pass state. Figure 3-18 show the decision process of HSM.

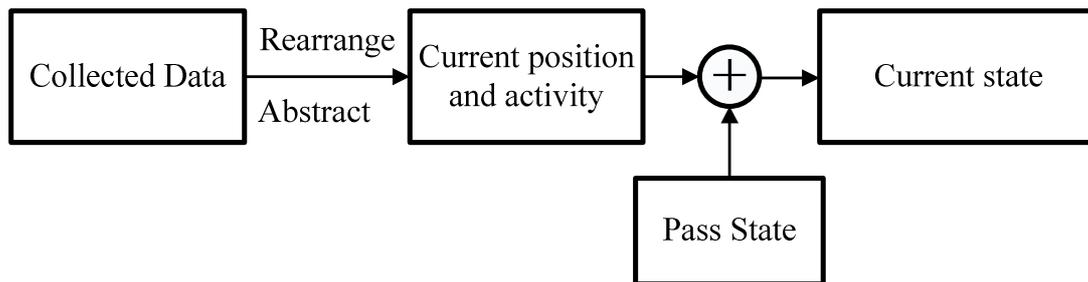


Figure 3-18. Overview of decision process of HSM

In this paper, we defined three different elder’s health state levels as the output of DDS, which are normal, question and alarm. Question level means not exact dangerous is detected, but something is suspicious. An example of question level situation is like the elder has take shower for more than two hours, or he didn’t have breakfast. Alarm level means an exact dangerous is detected such as heartbeat rate is lower than 50, body temperature is higher than 40, etc. Normal level means all collected sensor data are within the defined safe range and no question level is detected. Both question level and dangerous level will lead the result manager sends an emergence message to remote server and sends a SMS to care-givers.

HSM handles different level of results in different methods. If a normal level situation is detected, HSM will send the rearranged and abstracted data to input handler. If it is an alarm or question level result, HSM will call the result manager function with the determined result as parameters pass to it.

Input handler plays a role as a data compressor for the database. Size of a database in an embedded system is limited, so we are not going to save all the normal data into the database. Some sensor data will be compressed before it is stored into database. Input handler will take the average value of these kinds of data in a proper time period, and only the average value will be passing to database manager and store. Environment sensors data like room temperature and humidity can take average value in 10 minutes or even half an hour. Body

sensors data like blood pressure and pulse is more critical than environment sensors data. They should be taken average value in not more than one minute when no question or alarm situation is detected.

Result manager is in charge of all detected questions and alarm level situation. As soon as HSM decided an alarm or question level state, result manager will construct an emergence SMS and an emergence Ethernet packet base on the exact situation. SMS will be sent to GSM/GPRS protocol translator. Emergence Ethernet packet is push into Ethernet output buffer with high priority. After these, all question and alarm level data will be stored directly into database.

Work flow of the data decision system is show in figure 3-19.

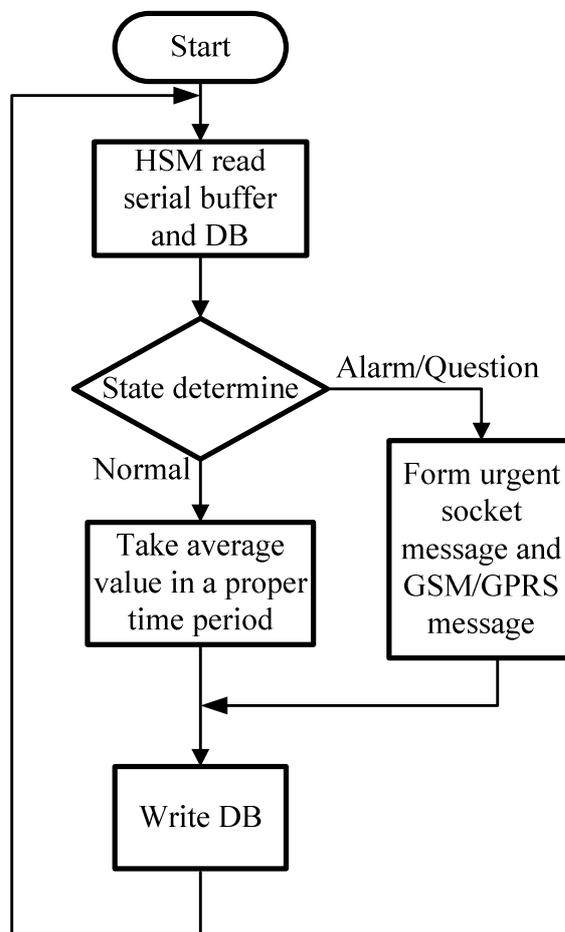


Figure 3-19 process of data decision system

### 3.5.3 Service Manage Platform

Service Manage Platform (SMP) is designed with the purpose of executing services and generating report for remote center. It includes request manager and report generator.

Request manager in SMP keep checking if there are any new requests in the request buffer. Two kinds of requests are stored in request buffer. One is report request for elderly health report and WSN work state report. This request is to ask the gateway to send all sensor data or network work state between the start time and end time that it specified. Another one is sensor network configuration request. Request manager has to distinguish these two kinds of request. Request manager will send all health report request to report generator. If it is a sensor network configure request, request manager will push it into the serial output buffer directly.

Main task of report generator is to generate all kinds of report. When no dangerous or questionable level state is detected, SMP will control the report generator module to send a short report to remote sever periodically (e.g. every 1 minute), just contains a signal telling the elder is in healthy state. The motivation for sending this report is, on one hand, to notice the healthy state; on the other hand, to let the remote server knows the gateway is working properly. An overall statistic report will be generated once a day. In the statistic report, maximum, minimum and average value of all kinds of sensing items will be given. It will also report how many question and alarm situations have happened in that day. Once the report generator receives a report request, it will take the start time and end time form it and read the required data from the database.

#### **3.5.4 Database Manager**

In this gateway design, we chose SQLite 3.0 as our database. SQLite is an open source embedded relational database management system with very small size (225KB) and full functional library is provided.

Size of the database in an embedded system is limited. All dangerous data will be stored directly into database while some kinds of data like room temperature will be compressed before it is stored. Input handler will take the average value of these kinds of data in a proper time period. Only the average value will be stored.

Database manager is in charge of database connecting when system boot up, and provides R/W interface for it. Sensor data are stored into SQLite database after pretreatment by input handler. Current time is added when data were written into database.

#### **3.5.5 I/O Interface**

Center control unit has to connect each ECMs (External Communication Modules) with different protocol. In this system, ECMs mean these three parts: WSN Module, WLAN module, and GSM Module. I/O interface is responsible

for initiating the connection to all ECMs and provides software read/write interface to transmit data with them. I/O interface includes Serial Input/Output Buffer, GSM/GPRS Protocol Translator, Ethernet Protocol Translator, Ethernet Output Buffer and Request Buffer.

#### 1) Serial Input Buffer

Serial Input Buffer in charge of configuring the serial port which connects WSN module and the center control unit, and provides data buffer for DDS. As soon as HSM decided a dangerous or question level state, result manager will construct the emergency SMS and Ethernet packet base on the exact situation. The emergency SMS and Ethernet packet follow the structure we have discussed in section 3.4.2. SMS is sent by the GSM/GPRS protocol translator and the emergence Ethernet packet is push into Ethernet output buffer with high priority.

#### 2) GSM/GPRS Protocol Translator

GSM/GPRS Protocol Translator is actually a GSM/GPRS terminal controller. The translator sets all the interface parameters to initial the connection between center control unit and GT64 GSM/GPRS terminal. The GT64 GSM/GPRS terminal we use in this design is controlled by extensive AT command set which is the standard language for controlling modems. GSM/GPRS protocol translator uses C language to realize the encapsulation of AT command set and achieved GSM/GPRS network communication functions. Same as other GSM terminal, GT64 terminal sends SMS using 7-bits coding PDU pattern. Special translate function is needed to convert the 7-bits codes to ASCII code before we read the SMS. The translate function is also need in IP address and password configuring when connecting to Internet with GPRS.

#### 3) Ethernet Output Buffer

Ethernet Output Buffer provides buffers for Ethernet Protocol Translator. Linked list structure is used in this buffer and provides the priority for the Ethernet packets storing in it. The Ethernet packet stored close to the list head has higher priority to be sent. We have two levels of Ethernet packet in this gateway which are, first, normal for all types of report packets; and second, high for emergency packet. So we are going to have two different buffer linked list insert functions here. One is to insert a new packet to the Linked list head and another is to insert it to the end of the list.

#### 4) Ethernet Protocol Translator

Ethernet Protocol Translator (EPT), on one hand keeps taking data packet from the output buffer and sent it through WLAN AP; on the other hand, push all received packets into request buffer. It communicates with the WLAN AP by opening a socket to sends and receives TCP/IP messages. Beside the remote server, other care-givers may also connect to the gateway, through Internet or

WLAN, when it is running in intelligent model. It means there may have multi clients connect to the gateway at the same time.

Ethernet protocol translator implements this multi clients' socket connection in the way shown in figure 3-20 (a). When the gateway boots up, EPT initialize the socket connection following the way we have discussed in section 3.3.2. After the initialization, EPT listens on the socket for connections. Each connected client will be assigned a socket file description. Client's IP address is bind to this socket file description. With this file description as the identity of the client, the EPT creates two new threads to handle the data packets transmitting. One thread is in charge of writing data packet to client. Its work flow is shown in figure 3-20 (b). As soon as this thread is created, it keeps fetching the data packet which located at the head of linked list of Ethernet Output Buffer and writes this data packet to client through the socket file description. Another thread is in charge of receiving request from client. Its work flow is shown in figure 3-20 (c). The `read ( )` system call will block this thread until a data packet is received from corresponding client. Once a data packet is received, this thread will put it into the Request buffer. After these two threads are created, the EPT will go back to listen for new client connection.

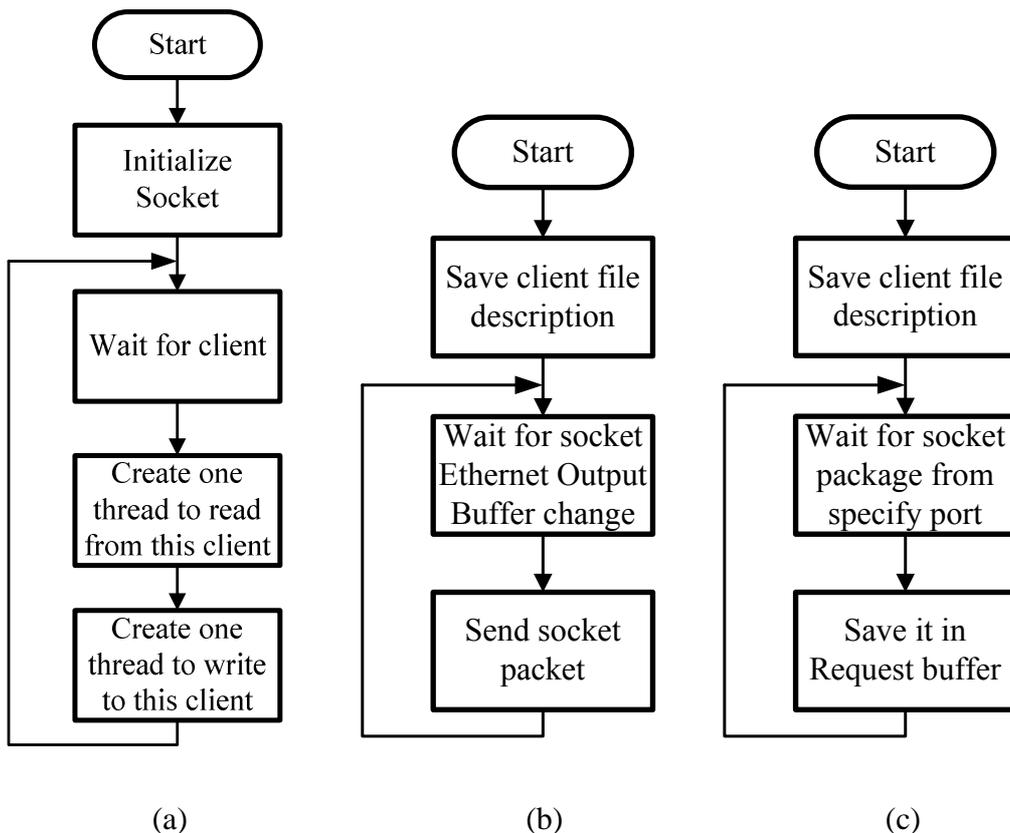


Figure 3-20. Processes of socket connection

## 6) Serial Output Buffer

Serial Output Buffer provides buffer for WSN configuration requests. It also provides serial communication protocol transform for WSN Module

## **4 Experiment**

### **4.1 Coexistence of the WSN and WLAN**

WSN in our system run with IEEE 802.15.4 and the WLAN AP use IEEE802.11. There maybe interference between these two kinds of devices because both of them are running in 2.4GHz band. WLAN AP used in this gateway design follows 802.11g standard. The band wide of it is between 2.4000–2.4835 GHz, and is divided into 13 channels. Each channel width 22 MHz but spaced between two adjacent channels is only 5 MHz apart. IEEE 802.15.4 shares the same bandwidth and divided it into 16 channels (from channel 11 to 26) with 3 MHz bandwidth in each of it.

In this experiment, only Received Signal Strength Indication (RSSI) is considered. It is the power present of a received radio signal. RSSI is an important parameter because it is used in many aspects such as choosing the most reliable link, estimating the link quality, implementing the frequency hopping and location detecting.

#### **4.1.1 Experiment setup**

To reduce the interference of the RF devices, this experiment is implemented in our WSN lab which is under ground. The experiment setup is shown in figure 4-1. In a gateway design, WSN base station should be placed very close to the WLAN AP. So, in this experiment we fixed their distance to 10 centimeter. A laptop acted as the remote server was located 1 meter away from the WLAN AP. One extra mote is used to transmit packets to the base station. The link connection between the mote and the base station is perpendicular to the link connection between WLAN AP and laptop.  $d_r$  is the distance between WLAN AP and laptop, and  $d_s$  is the distance between WSN mote and base station. All devices are located 1 meter high from ground.

TinyOS-2.0 is installed in the mote and the WSN base station. Their TX power (sensor node antenna transmit power) is set to 31 which represent 0dbm. WSN is set to run in channel 26. Data packet which has been discussed in section 3.4.1 is used to be transmitted from mote to base station. Together with packet stream head and CRC, total length of one packet is 20 bytes. Packet frequency is set to 4 packets per second.

A special program is developed for gateway in this experiment. This program has three functions. First, it received all the data packets which come from WSN base station and save all RSSI value of the packet into a text file in the gateway. Second, it establishes the wireless connection with no data packet transmit between them. Third, in order to simulate a high load wireless

network, it sent a pre-designed packet to laptop circulating. The packet size is 100 bytes, and packet rate is 1000 packets/s. The wireless card in laptop and the WLAN AP are set to communicate in channel 6.

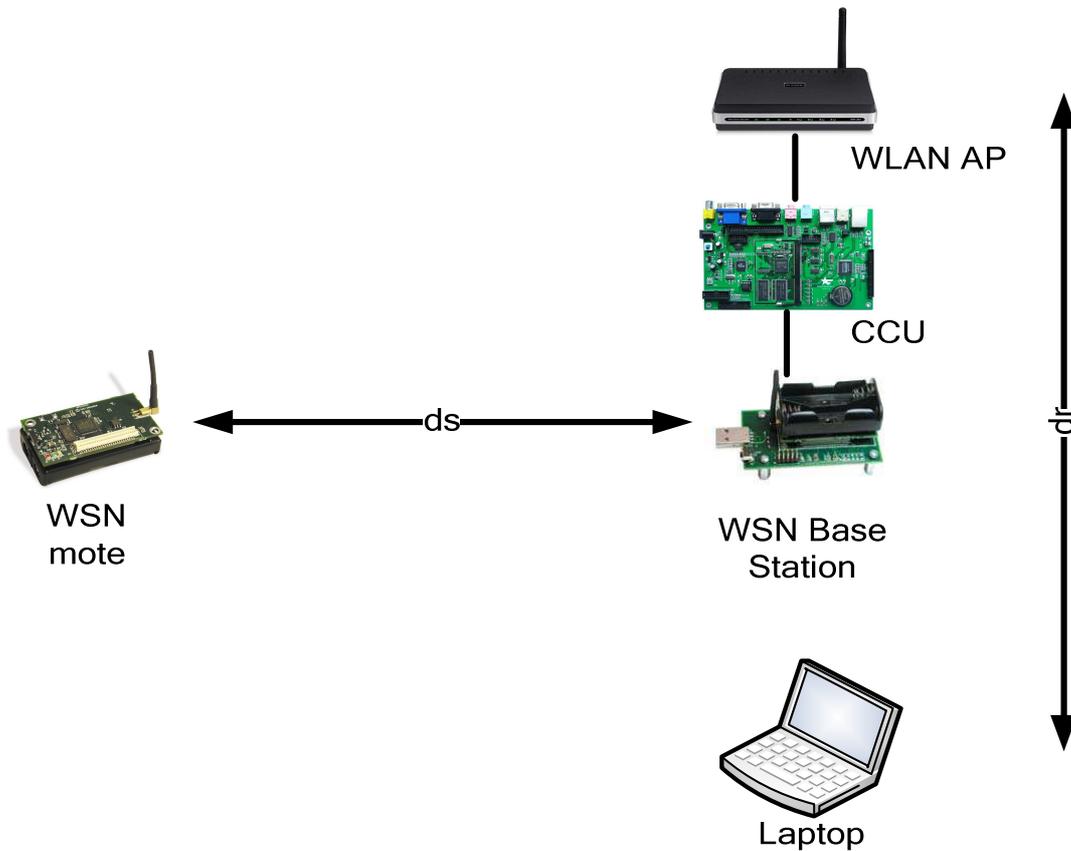


Figure 4-1. Coexistence experiment setup

#### 4.1.2 Implementation

Three different scenarios of experiments are implemented to get the RSSI value.

- (1) Without WLAN. WLAN AP and laptop are switched off in this experiment.
- (2) With WLAN but no packet transmitting. Gateway just establishes the wireless connection to laptop without running the packet sending program.
- (3) With WLAN and packets transmitting.

First, we implemented the experiments with  $ds$  equal to 3 meter and finished all three scenarios. And then we changed the  $ds$  to 6 meter and repeat these scenarios. All experiments are run 5 minutes. Final result is shown in table 4-1.

Pay attention that the obtained values from the experiment are only the first part of the following equation:

$$P = \text{RSSI value} + \text{offset (dBm)}, \quad \text{where offset} = -45 \text{ dBm.}$$

Table 4-1. Coexistence experiment result

<b>Distance(m)</b>	<b>Without WLAN</b>	<b>WLAN with no packet transmit</b>	<b>WLAN with packet transmit</b>
3	-75	-78	-81
6	-80	-89	-90

### 4.1.3 Conclusion

From table 4-1, we can conclude that the sharing frequency band of 802.11g will affect the network performance of 802.15.4 even just have the WLAN connected and the interference increase when high load data packet is transmitting in the WLAN. The gap between WLAN with/without packet transmit is smaller than the gap between with/without WLAN. From the table, we can also see when distance of the WSN mote increased, the gap between with and without WLAN increased significantly.

## 4.2 Gateway Data Sampling Experiments

This experiment is motivated to test the sensor nodes in their capability of environment sampling and also the capability of the gateway storing data into its on-board database. We test these two capabilities in the bathroom of one of the co-researches.

### 4.2.1 Experiment setup

A sensor node which attached with a temperature sensor and a humidity sensor was placed close to the sink in bathroom of the co-research. One more sensor node was used to help hopping to the gateway which located in the living room. Sensor data were stored in the gateway.

### 4.2.2 Implementation

In order to compare the experiment result, time of the co-research using the bathroom is recorded. Total recording time is 400 minutes. The time record is shown in table 4-2

Table 4-2. Time record of co-research relative to the bathroom

Time	Action
11:45	Start recoding
12:26-12:40	Shower
13:26-13:28	Wash hands

The sensing temperature and humidity data is shown in figure 4-2.

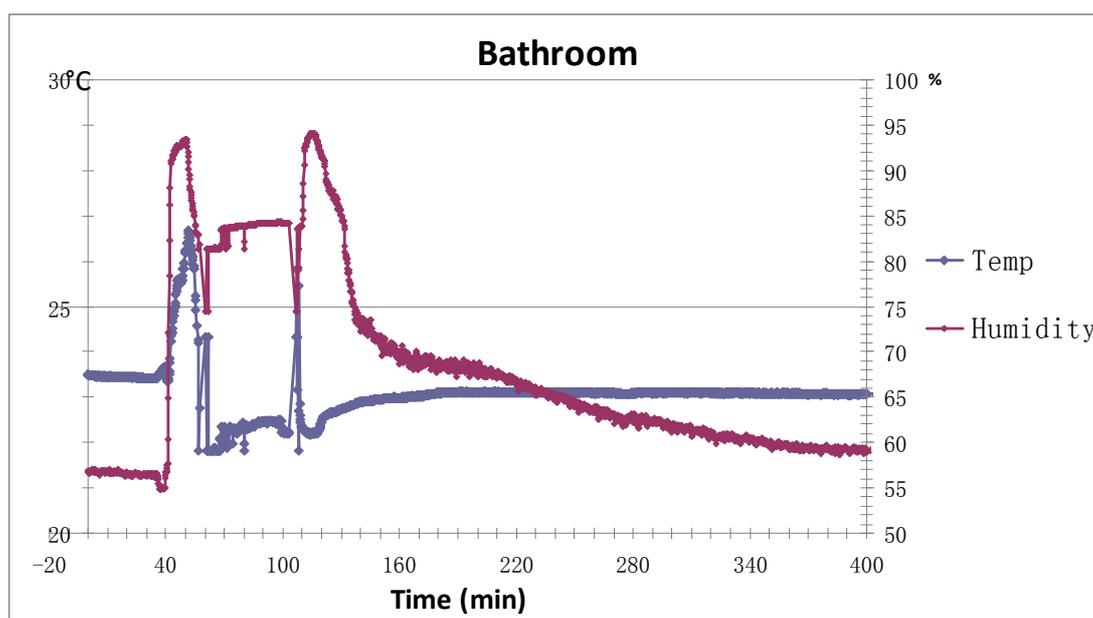


Figure 4-2. Temperature and humidity data in bathroom

### 4.2.3 Conclusion

From figure 4-2, we can see that at around 40 minute after the record starting, the humidity and the temperature start to go up rapidly. By checking the time record, we know the co-research is taking shower. After around 20 minutes, the sensing values start to go down, indicating that the co-research has finish shower. Similar situation happen again at around 100 minutes when co-research was washing his hand. Obviously, the activity that the co-researcher's used the bathroom can be detected by the temperature and humidity data collected by the sensors.

## 4.3 System Delay Measure

In this experiment, we are going to test the performance of the gateway to see the delay time to send data packet to remote server using our designed

communication protocol in both cable connection and wireless connection. Delay time of emergency SMS has also been tested.

### 4.3.1 Experiment setup

This experiment was performed in Jönköping University. The first step of this experiment is to establish the experiment environment. We deployed 20 Crossbow Micaz sensor nodes in different rooms. Some sensor nodes were attached with one or several sensors. A co-researcher was wearing a heart beat sensor. Normal heart beat rate is defined between 60 and 100 in these tests. The intelligent gateway is showed in figure 4-3. The normal report cycle time is set to one minute. GT64 is set to require the SMS server to send a confirm acknowledgment when the SMS is received by the target. A time counting program is plugged into the gateway to count the delay time of Ethernet data package and SMS transmission. The time counting result will be saved in a file on-board. Remote server is connected with gateway via the local area network. Simulator software was built up for remote server in order to show the experiment results, and send ACKs/requests to the gateway. A 16GB flash memory was plugged in as external DB.

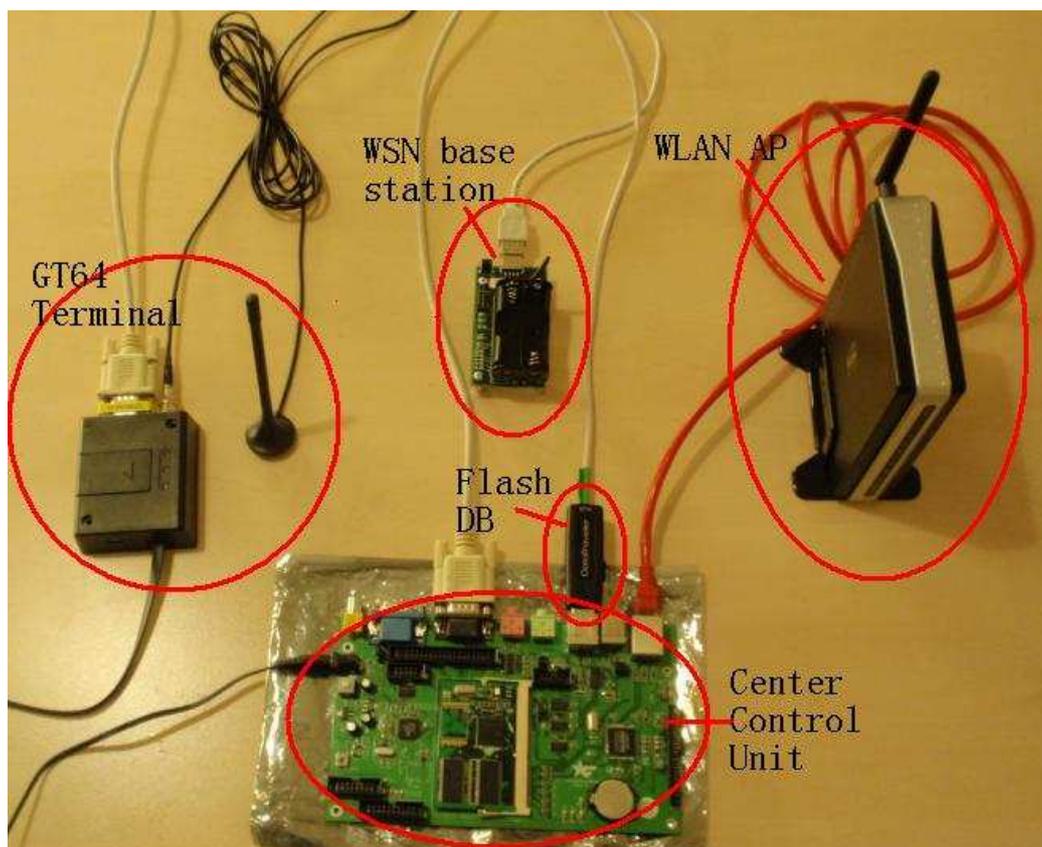


Figure 4-3. Photo of GPRS Module, WSN Module, Flash DB, Center Control Unit and WLAN AP.

### 4.3.2 Implementation

After the prototype was established, we executed a series of tests to evaluate the performance of the gateway. First we test in the normal situation without emergence situation happen. Sensor nodes were turned on and the co-researcher stay quiet. After one minute, simulator in remote server received the normal report packet from the gateway as we designed. Second, we restart the system to test with emergence situation. When sensor nodes were on, the co-researcher started to do sports. As soon as his heart beat rate run over the normal range, simulator received an emergence data packet and a SMS is received by cell phone. Then read the time counting file and check Ethernet packet delay and SMS delay. We repeated the same experiment with the remote server connected through wireless area network. And the gateway also performed as how it is designed.

Here we sum up the delay time. When remote server is connected through LAN, it took average approximately 0.13 seconds between gateway sends Ethernet package to remote server and received ACK from it. And it took approximately 0.17 seconds when remote server was connected through WLAN. Delay time between GT64 sent a SMS and got a reply from GSM server is 10.84 seconds. Detail results of experiments are show in table 4-3.

Table 4-3.Statistic of notification delay

	<b>Minimum delay (second)</b>	<b>Maximum delay (second)</b>	<b>Average Delay (second)</b>
<b>Package(LAN)</b>	0.11	0.14	0.13
<b>Package(WLAN)</b>	0.11	0.20	0.17
<b>SMS</b>	9.89	11.23	10.84

## **5 Conclusions and discussions**

In this paper, we propose a smart gateway design prototype for health care system using WSN. In this smart gateway design, tasks like sensor data storage, elder's current health state detection and real-time report are done in the low power embedded system in the intelligent model. Hardware and software design of the gateway are presented and transmit protocol is designed for this gateway-central system architecture. A series of experiments results show this prototype system is feasible and reliable.

In the future, we aim at optimizing the interconnection by employing GPRS communication between gateway and remote server to extend the available coverage of the health care system and use the Hidden Markov Model (HMM) to upgrade the DDS. Then, we may consider integrating internet-base webpage and voice call function in the gateway. Security issues will also be considered in the future work.

## 6 References

- 1 [1] Holger Karl, Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, Ltd, 2006. pp. 78-81 and pp.139-142.
- 2 [2] *Wireless sensor network*: Available at: <http://en.wikipedia.org/wiki/WSN>. [Accessed on Oct 20, 2009].
- 3 [3] Youzhi Xu, *Draft of a project Description- Wireless Sensor Networks for Healthcare at home*. Unpublished. Tekniska Högskolan i Jönköping. 2007, April.
- 4 [4] Hongwei Huo, Youzhi Xu. *An Elderly Health Care System Using Wireless Sensor Networks at Home*. Third International Conference on Sensor Technologies and Applications. sensorcomm, pp.158-163, 2009.
- 5 [5] TinyOS Mission Segment: Available at: <http://www.tinyos.net/special/mission>[Accessed on 5 November].
- 6 [6] I.F Akyildiz, et al., “*Wireless Sensor Networks: A Survey*”, Computer networks, Vol. 38, pp.393-422, 2002.
- 7 [7] C-Y. Chong and S. Kumar, “*Sensor Networks: Evolution, Opportunities, and Challenges*”, Proceedings of IEEE, 91(8), pp. 1247-1256, 2003.
- 8 [8] Waleed Youssef, Mohamed Younis. “*Intelligent Gateways Placement for Reduced Data Latency in Wireless Sensor Networks*”. Communications, 2007. ICC '07. IEEE International Conference, pp. 3805-3810, June 2007.
- 9 [9] Lili Wu, Janne Riihijärvi, Petri Mähönen. “*A Modular Wireless Sensor Network Gateway Design*”. Communications and Networking in China, 2007. CHINACOM '07. Second International Conference pp. 882 – 886, Aug. 2007
- 10 [10] Marco Zuniga Z. and B. krishnamachari. “*Integration Future Large-scale Wireless Sensor Networks with the Internet*”. USC, Computer Science Technical Report 03-792, 2003.
- 11 [11] Stephanie Schleinkofer. “*HMM Based Activity Recognition of Elderly at Home using Wireless Sensor Networks*”. Final thesis of engineer school of Jönköping University, 2008.
- 12 [12] Se-Jin Oh, Chae-Woo Lee. “*u-Healthcare SensorGrid Gateway for connecting Wireless Sensor Network and Grid Network*”. Advanced Communication Technology, 2008. 10th International Conference, Volume: 1, pp. 827-831, Feb. 2008

13[13] Williamson, K. *Research methods for students and professionals, 2nd ed.*, Centre for Information Studies, Wagga wagga, NSW 2002, pp. 151-153.

14[14] *GPRS (General Packet Radio Service), HSCSD & EDGE*. Available at: <http://www.mobile-phones-uk.org.uk/gprs.htm>

15[15] 李亚锋, 欧文盛, 《ARM嵌入式Linux系统开发从入门到精通》. 清华大学出版社. 2007. pp 36-106.

16[16] *Market Data Summary*. Available at: [http://www.gsmworld.com/newsroom/market-data/market\\_data\\_summary.htm](http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm)

17[17] Vlad Trifa, Samuel Wieland. “*Design and Implementation of a Gateway for Web-based Interaction and Management of Embedded Devices*”.

18[18] Dunkels A., Vasseur J. “Architectural Styles and the Design of Network-based Software Architecture”. PHD thesis, University of California, Irvine, California, 2000.

## 7 Appendix

### 7.1 NFS

During root file system development process, we strongly recommend the use of NFS file system.

#### 7.1.1 NFS Advantages

- All files are saved in the PC which can help us maintain and develop them. In this way, we don't have to compress and download the whole file system to the board again whenever we modified anything of it. This will save us a lot of research and development time and extend the life time of the develop board.
- Local workstations use less disk space because commonly used data can be stored on a single machine and still remain accessible to others over the network.
- There is no need for users to have separate home directories on every network machine. Home directories could be set up on the NFS server and made available throughout the network.
- Storage devices such as floppy disks, CDROM drives, and Zip drives can be used by other machines on the network. This may reduce the number of removable media drives throughout the network.

#### 7.1.2 Install NFS Server in Ubuntu

First, use this command in control to install nfs and portmap software.

```
sudo apt-get install nfs-kernel-server nfs-common portmap
```

When configuring portmap do NOT bind loopback. If you do you can either edit `/etc/default/portmap` using the following

```
sudo vi /etc/default/portmap
```

or use the following command

```
sudo dpkg-reconfigure portmap
```

Restart Portmap using the following command

```
sudo /etc/init.d/portmap restart
```

### 7.1.3 NFS Server Configuration

1) NFS exports from a server are controlled by the file `/etc/exports`. Each line begins with the absolute path of a directory to be exported, followed by a space-separated list of allowed clients.

You need to edit the exports file using the following command

```
sudo vi /etc/exports
```

Here are some quick examples of what you could add to your `/etc/exports`

In our design, we give Full Read Write Permissions to the develop board to us `"/home/ben/board"` file. Develop board uses address 192.168.1.10. We wrote like in this file:

```
/home/ben/board 192.168.1.10(rw,no_root_squash,async)
```

Save this file and exit.

A client can be specified either by name or IP address. Wildcards (\*) are allowed in names, but should usually be avoided for security reasons.

A client specification may be followed by a set of options, in parenthesis. It is important not to leave any space between the last client specification character and the opening parenthesis, since spaces are interpreted as client separators.

2) Add client IP address into `/etc/hosts` using command

```
sudo vi /etc/hosts
```

Add this command into it:

```
192.168.1.10 clientA clientA
```

Restart NFS server using the following command

```
sudo /etc/init.d/nfs-kernel-server restart
```

If you make changes to `/etc/exports` on a running NFS server, you can make these changes effective by issuing the command

```
sudo exportfs -a
```

### 7.1.4 Configuration on develop board

Change the vivi setting to use NFS while starting up the board.

- 1) Go to vivi setting state by keep space key pressed while you pressing the power button or restart.
- 2) Use this command line to change the vivi start up parameter (with no ENTER in it):

```
param set linux_cmd_line: " noinitrd root=/dev/nfs nfsroot=
192.168.1.3:/home/ben/board
ip=192.168.1.10:192.168.1.3:192.168.1.1:255.255.255.0 init=/linuxrc
console=ttySAC0,115200"
```

```
param save
```

### 7.1.5 Testing Your Configuration

Connect develop board and host PC with Ethernet cable. Restart the development board, if you see this information, it means you have success in using NFS function.

## 7.2 USB interface initial function

This code is the driver to initial the USB interface for Linux 2.6.14. Put them in arch/arm/mach-s3c2410/mach-smdk2410.c file in the kernel before compiling it.

```
int smdk2410_usb_init(void) /* USB */
{
    unsigned long upllvalue = (0x78<<12)|(0x02<<4)|(0x03);
    printk("USB Control, (c) 2006 s3c2410\n");
    s3c_device_usb.dev.platform_data = &usb_s3c2410_info;
    while(upllvalue!=__raw_readl(S3C2410_UPLLCON))
    {
        __raw_writel(upllvalue,S3C2410_UPLLCON);
        mdelay(1);
    }
    ////////////////add miscr////////////////////
    unsigned long miscr;
    miscr = __raw_readl(S3C2410_MISCCR);
    miscr |= S3C2410_MISCCR_USBHOST;
    miscr &= ~(S3C2410_MISCCR_USBSUSPND0 |
S3C2410_MISCCR_USBSUSPND1);
    __raw_writel(miscr,S3C2410_MISCCR);
    ///////////////////////////////////////////////////////////////////
    return 0;
}
```