行政院國家科學委員會專題研究計畫　成果報告

總計畫:EHOME ： 電子家庭雛型之設計與實作

計畫類別： 整合型計畫
計畫編號： NSC91-2218-E-002-009-
執行期間： 91 年 08 月 01 日至 92 年 07 月 31 日
執行單位： 國立臺灣大學資訊工程學系暨研究所

計畫主持人： 陳俊良

報告類型： 完整報告

處理方式： 本計畫可公開查詢

中　華　民　國 93 年 2 月 20 日

行政院國家科學委員會補助專題研究計畫成果報告

eHome：電子家庭雛型之設計與實作總計畫 (3/3)

計畫類別：整合型計畫
計畫編號：NSC 91－2218－E－002－009－
執行期間：91 年 08 月 01 日至 92 年 07 月 31 日

計畫主持人：陳俊良
共同主持人：許清琦、鄭士康、王勝德
計畫參與人員：

成果報告類型(依經費核定清單規定繳交)：完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立台灣大學資訊工程學系

中　華　民　國 92 年 12 月 31 日

中文摘要

隨著半導體相關技術顯著的成長與網際網路的盛行，愈來愈多的家庭網路系統標準與家庭網路介面標準被發展出來。這些標準依照開放性可分為國際標準，公開標準與私有/私人標準三大類。除此之外，近年來可攜式裝置，如手機與 PDA，有逐漸深入每個人家庭生活的的趨勢。其已成為未來家庭網路應用中重要的終端應用平台。

本篇論文著重於如何設計出一個新的家庭網路系統 – eHome system，其能整合所有的家庭網路標準並且提供與可攜式裝置相互連結的機制來實現遠端存取的能力。也就是說說，我們希望能夠建構出一個泛用型家庭網路系統，且希望能夠透過可攜式裝置，在家或在外都能控制家中的任一家電。

目前已經成功地實做出 eHome 家庭網路系統雛形，當我們在外的時候，可以使用內建 WAP 瀏覽器的 PDA 或是手機，經由 GSM/GPRS 無線網路連上 Internet，再進一步利用 HTTP 協定，連上家中伺服器來控制家中的任一家電。在家時，則可以直接透過 PDA 上的友善人機介面，經由藍芽無線傳輸來遙控家中電器，讓使用者無論在任何地方皆能掌握家中一切狀況。

Abstract

The semiconductor technology has extremely advanced and Internet is now in widespread usage. As a result, more and more international, open and proprietary home network standards (home systems and network interfaces) are evolved recently. Moreover, the portable device then becomes the popular and important appliance.

This report aims to propose a new home system, which integrates with complete home network standards, and the portable devices for providing capability of remote access. Moreover, we have implemented a tentative prototyping with low cost. Now we can use the mobile phone or PDA built in WAP browser to control the home appliances remotely through Internet and HTTP protocol outside home. Moreover, we can control the home appliances directly via friendly control user interface and BlueTooth inside home without any GSM/GPRS service charge

Table of Contents

Table of Figures

List of Tables

Chapter 1.

Introduction

1.1 Trend

The rapid growth of Internet and semiconductor technology has accelerated the coming of the Post-PC (Personal Computer) century. The Post-PC century is an emerging term because there are more and more consumer electronics becoming more intelligent due to falling costs and increased capability of today's microprocessors. The consumer electronics no more comprises a hardware-only domain, and they are both hardware and software solutions to deliver digital entertainment.

One emerging type of the consumer electronics, IA (also named Internet Appliance, Information Appliance, or Intelligence Appliance), usually has Internet capability and shares the information with each other. Moreover, more and more consumer electronics are linked by network and can be further connect to Internet in the future. The applications of IA are various especially in the field of home networking. Home networking is a popular topic will expand huge market in the future. Some marketing surveys of home networking are shown in the following lists:

- According to HAA (Home Automation Association, http://www.homeautomation.org) market survey. Demand for home automation will grow 20% every year.
- IDC (http://www.idcresearch.com/) and In-Stat (www.in-stat.com) the sales volume of digital home products would exceed PC by 2003. The sales volume of digital home products will be 10 times of PC by 2010, and the

global market will become a $580 billion industry when 0.9 billion families use digital equipment.

- Yankee (Yankee Group, http://www.yankeegroup.com) predicted. There will be 12.4 million of US homes use home networking equipment in 2002.
- Above market surveys are extracted from. [1]

Furthermore, there are some reasons to accelerate growing of home networking [2][3][4]. The reasons are listed as follows:

- Rise in Workspace changed to Small Office and Home Office.
- Multi-PCs (Personal Computers) in Homes.
- More and More consumer electronics will link together by network such as multiple PCs, printers, and other IAs.
- Sharing a broadband Internet access.
- Sharing content include multimedia and data among IAs.
- People want to access the Internet in different room.

1.2 eHome System

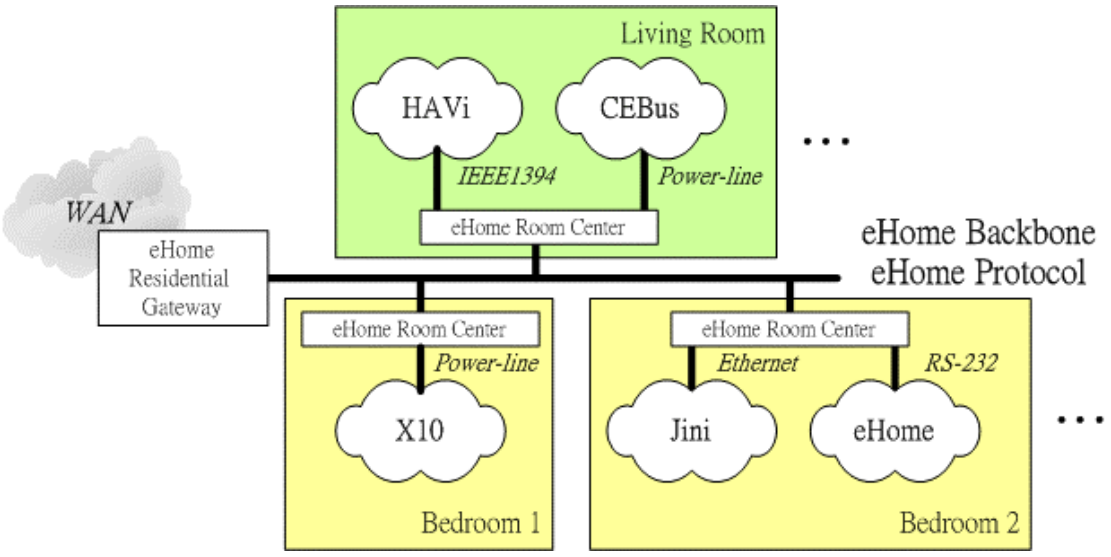


Figure 1.1: eHome System Architecture

According to the trend of home networking, we known that there are different multiple home networking systems and interface coexisting in future home. Therefore, it is urgent demand for designing a home system that can integrate with whole home networking systems and interfaces. Besides, it is very important that to integrates with

the portable devices to provide the capability of remote access in the local home network.

We propose a general home system named eHome. It is low-cost, easy to use and construct, highly flexible, and downward compatible. Furthermore, it integrates with the portable devices. Figure 1.1 illustrates the architecture of eHome system. We can see that eHome system is a gateway-based architecture to intend to integrate existing home network system standards such as HAVi, and proprietary home system but popular such as X10. There is only one eHome Room Center that plays the role of central control in each room. Every eHome Room Center has three main components: eHome Gateway, eURC Proxy, which integrates with portable to provide capability of remote access and Service Proxy which manages the services of the local home network. Besides, eHome Room Center will connect its local home systems into eHome backbone finally. Accordingly, the home systems will be connected all together and can cooperate with each other through eHome Room Center. The protocol called eHome protocol running over eHome backbone can be any protocol, such as HTTP, SOAP, etc., depended on the demand. Therefore, choosing what protocol as eHome protocol will cause some issues and the capabilities of eHome system. For instance, if you chose only HTTP as eHome protocol, then you will need to use the URL-based naming to identify each home appliance and eHome system will not be capable of the multimedia streaming.

EHome Gateway is specifically classified into three categories: Protocol Gateway, Control-Message Gateway, and Medium Gateway. The Protocol Gateway is responsible for the interoperability among eHome system and the home network system standards, which have the characteristics such as service discovery, proprietary network protocol. The Control-Message Gateway is dedicated for message conversion between two message based home network systems such as X10 and eHome. Medium Gateway, however, only take charge of conversing the message/data between the different network interfaces such Ethernet to RS-232 without any modification.

We also implement a prototype to verify the integrity. This prototype has integrated many technologies such as Java, BlueTooth, TCP/IP, GSM/GPRS, RS-232, ASK-RF, Embedded system etc. Moreover, we define a new control protocol as our eHome protocol in our implementing prototype. This protocol is easy to implement and is low data transferring overhead. We can use this control protocol to control all

appliances such as turning on lamp, turning on TV, etc. in eHome system. Moreover, you can access the home service such as Lamp ON/OFF by WAP Browser built in portable device through GSM/GPRS wireless network outside the house. And you can also use the friendly user interface of the portable device to access the home service through BlueTooth directly without any GSM/GPRS service charge.

1.3 Report outline

We have introduced the trend of home networking nowadays and propose new home network system architecture - eHome. In the remaining chapters, we will give more details about home network technologies, eHome system and our implemented eHome prototype.

There are many technologies in the home network area. Generally, we can divide the technologies into two major parts: the home network interface standards and the home network system standards, or called middleware. Besides, they home network interface standards can be further classified into wired and wireless technologies which will be discussed in Chapter 2.

The home network system standard is a system architecture, which may comprise both hardware and software or only software framework, and in which the devices connected by the network can cooperate and discover services among devices. Some of the home network standards such as HAVi, Jini, OSGi, UPnP, etc. We will introduce in Chapter 3.

Chapter 4, then, will give you an overview of eHome system. We will point out the consideration in designing eHome system and introduce every major component of eHome system. These components are intended to solve some issues of integration of different home networking systems. Besides, in Chapter 5, the implementation of eHome system will be discussed. We will introduce each component of implementation prototype of eHome system and point out our consideration in designing.

Finally, in Chapter 6, we will address some future works and problems which should be improved and modified in our proposed eHome system and implemented prototype.

Chapter 2.

Home Network

Interface Standards

2.1 Overview

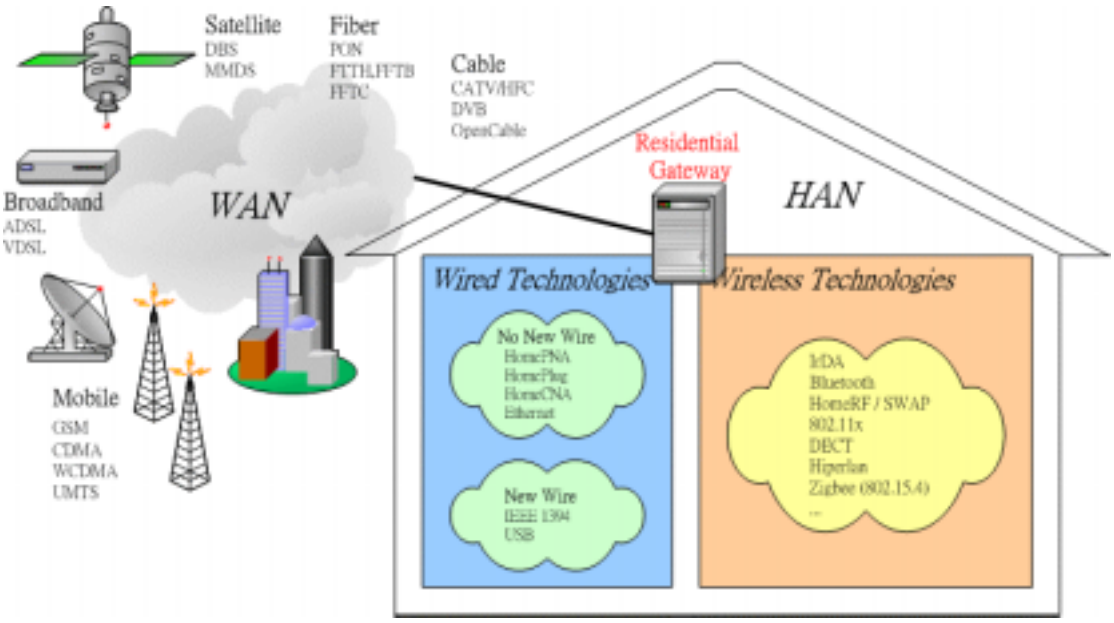


Figure 2.1: Network Access Technologies

Figure 2.1 gives us an overview of network access technologies. These technologies can be separated into two major categories: WAN (Wide Access Network) and HAN (Home Network Access). Many standards belong to HAN; for instance, Ethernet [7] and HomePNA [12], and they can be further categorized according to wired or wireless technologies. We will introduce some important standards and point out their

main features. Besides, we will analyze their disadvantages and advantages.

2.2 Wired Technologies

2.2.1 Ethernet

Ethernet technology is based on IEEE (Institute of Electrical and Electronic Engineers) 802.3 standard [7], which has many revisions because of its interoperability between many manufactures. And the original developer are DEC, Inter and Xerox. The wire structure can be twisted pair, coaxial cable, fiber etc. They are bi-directional, typically are strongly reliable, and thus they are popularly used as the standard choice for PCs and printers. Currently its speed is up to 10 Mbps. The new version called Fast Ethernet support data transfer rates of 100 Mbps. And the next new version, Gigabit Ethernet, will support data transfer rates of 1 Gbps.

Ethernet standard defines two layers of the OSI Reference Model. One is a physical layer, and the other is a link layer. The PHY (Physical layer) is dedicated for how to transmit unstructured raw data over a physical medium, and it also describes the electrical, mechanical, and functional interface to the network. There are many media specifications used for supporting different data rates, media, and topology configurations. Following Table 2.1 will show some common media specifications.

Standard	IEEE	Speed	Medium	Topology	Max. Cable Length	
					Half Duplex	Full Duplex
10Base5	802.3	10 Mb/s	Single 50-ohm RG-58 coaxial cable (thin Ethernet	Bus	500 M	N/A
10Base-T	802.3i	10 Mb/s	Two pairs of 100-ohm Category 3 or better UTP cable	Star	100M	100M
10Base-FL	802.3i	10 Mb/s	Two Optical Fibers	Start	2000M	>= 2000M
100Base-TX	802.3u	100 Mb/s	Two pairs of 100-ohm Category 5 UTP cable	Star	100M	100M
1000Base-TX	802.3ab	1Gb/s	Four pair of 100-ohm Category 5 or better cable	Star	100M	100M

Table 2.1: Ethernet Physical Layer Media Specifications

The data link layer is focus on how to get data packets in the network. It can be divided to two sub-layers LLC (Logical Link Control) and MAC (Medium Access Control). The LLC is the upper sub-layer in data link layer to support error checking. And the lower sub-layer MAC is focus on how to get data packet in the network. The

CSMA/CD technique used in Ethernet and Fast Ethernet MAC layer is very important protocol for the later MAC layer development

The CSMA/CD is the abbreviation of Carrier Sense Multiple Access/Collision Detection. There is an interesting metaphor, “The CSMA/CD is a protocol functions like a dinner party in a dark room. Everyone around the table must listen for a period of silence before speaking (Carrier Sense). Once a space occurs everyone has an equal chance to say something (Multiple Access). If two persons start talking at the same instant they detect that fact, and quit speaking (Collision Detection.)”. [9] We map it to Ethernet operation. First when one interface is transmitting data, there will be a signal on the channel called “carrier”. One interface needs to sense signal on the channel before sending the data out. This process is called “Carrier Sense”. Secondly, the priority of all Ethernet interfaces is the same. And we call it “Multiple Access”. It is possible that two interfaces sense no carrier to send data out simultaneously because every interface has the same priority and the signals take the finite time to travel one end of Ethernet to the other. We call this situation “Collision”. There is one way to solve this problem. When MAC senses the collision signal, it will stop current transmission and resend after back-off time.

2.2.2 HomePNA

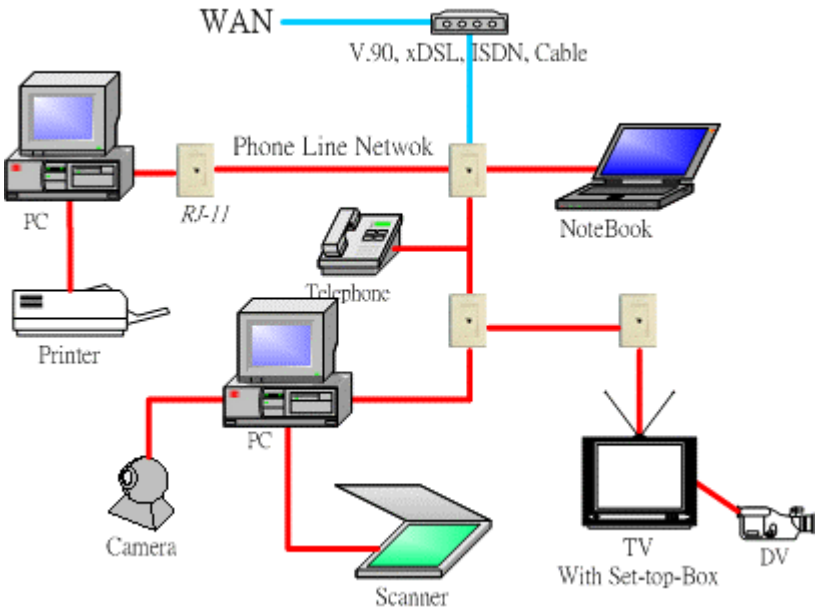


Figure 2.2: HomePNA Networking Topology

2.2.2.1 History

HomePNA (Home Phone-line Networking Alliance) [12], an organization founded in June 1998, is dedicated to the development of standards and specifications for interoperable, home-networked devices that use existing twisted pair phone wire. It was founded in June 1998. The major members of HomePNA currently are .2Wire, 3COM, Agere System, AMD, AT&T Wireless, Broadcom, Compaq, Conexant, CopperGate Communications, HP, Motorola, IBM, Inter etc.

2.2.2.2 Overview

HomePNA standard comprise physical layer and data link layer mapped to OSI reference model. It also defines the MAC (Media Access Control) layer which uses the CSMA/CD as same as the 802.3 Ethernet Standard. And refer to Figure 2.3 the data of upper layer is compatible with frame type in 802.3 Ethernet data frame type. So the application based on Ethernet can be reused in HomePNA too.

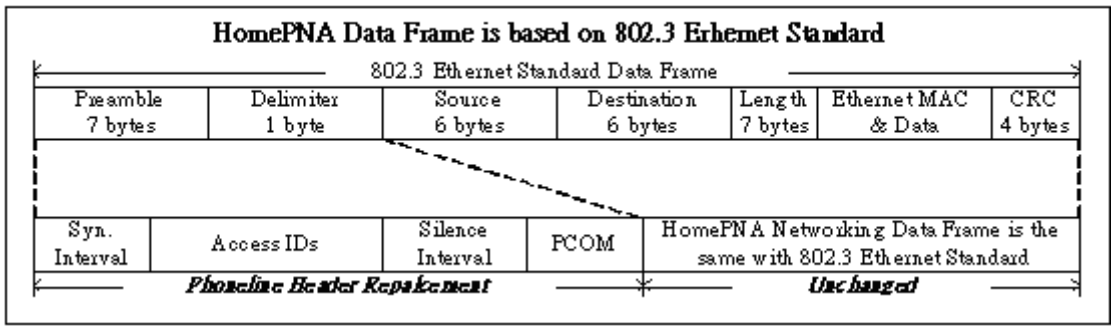


Figure 2.3: HomePNA Data Frame Format [8]

The bandwidth of HomePNA v1.1 is up to 1Mbps in the symmetric mode within distance up to 500ft. HomePNA v2.0 extends the distance to 1000ft. and the speed to 10Mbps. Moreover, the speed is further extended to 100Mbps in HomePNA v3.0 Standard. And HomePNA v3.0 will support the voice-over-HomePNA extend from HomePNA v2.0 by enabling eight simultaneous high-quality voice stream within home. Moreover HomePNA v3.0 can coexist with other services such as ADSL, ISDN and POTS, fully downward compatible with HomePNA v2.0, and support the QoS (Quality of Service) for transmitting data to end-user reliably.

Because HomePNA takes advantages of the existed phone-line used by some technologies such as ADSL, it needs the frequency division multiplex technology to

carry different analog signal on the same phone-line. The analog telephony services use the low part of spectrum: below 35kHZ. ADSL uses spectrum up to 1.1 MHz. However, HomePNA selected 4 to 10 MHz as its spectrum. HomePNA v1.1 utilizes the PPM and HomePNA v2.0 uses QAM (Quadrature Amplitude Modulation) modulation technologies separately to get more throughputs and to achieve greater robustness. To obtain more details, please refer to [8][10][11][12].

2.2.3 HomePlug

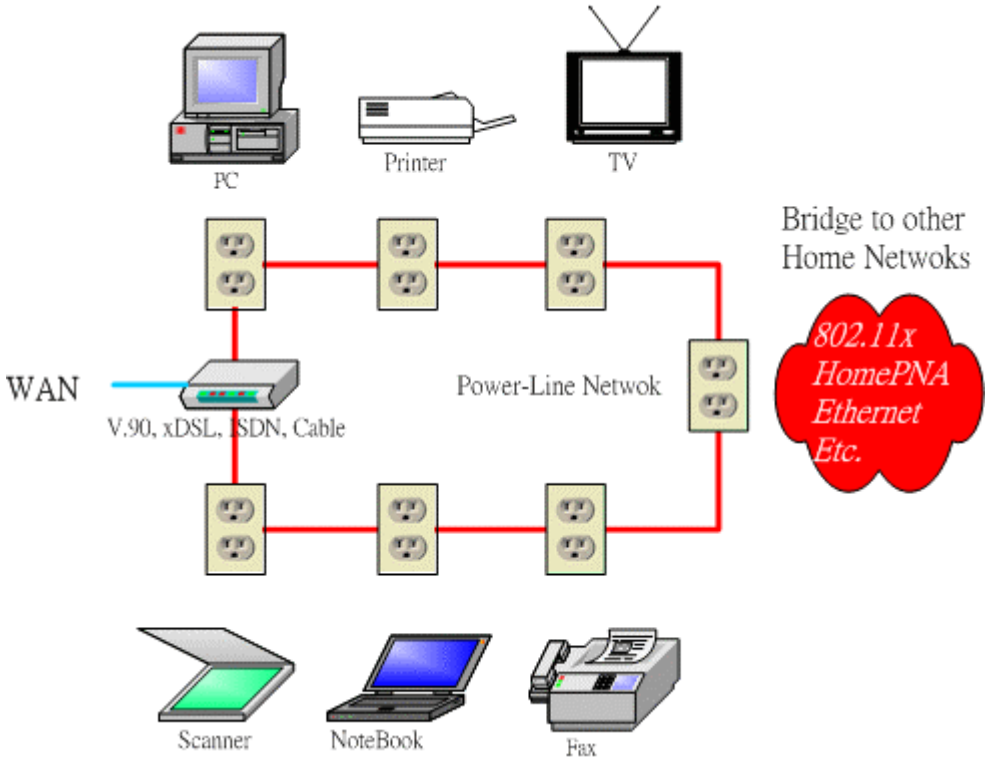


Figure 2.4: HomePlug Networking Topology

2.2.3.1 History

There are many technologies intended to carry the digital data on the existed power line such CEBus, Lonworks, X10 etc. Figure 2.4 is the illustration of power-line network structure of HomePNA. HomePlug [13] Power-line Alliance, established in 2000 is the organization emphasizing on standardization of in-home power-line network. There are 13 founding members include 3COM, AMD, Cisco, Systems, Compaq, Conexant, Enikia, Intel, Intellon, Motorola, Panasonic, Radio, Shack, SONICblue, and Texas Instruments. HomePlug has chosen the Intellon power-line technology as its baseline in the first-generation specification.

2.2.3.2 Overview

HomePlug has defined a robust PHY layer and an efficient MAC layer in order to make a reliable communication on the power-line medium. The PHY is focus on the modulation, coding, and basic packer formats. And the MAC is used to control the sharing of the medium among multiple clients.

The OFDM (Orthogonal Frequency Division Multiplexing) widely used in the DSL and terrestrial wireless distribution of television signals is used as the basic transmission technique in the PHY. HomePlug use ODDM in the robust mode rather than in the continuous mode. HomePlug uses concatenated Viterbi and Reed Solomon forward error correction with interleaving for payload data, and TPC (Turbo Product Coding) for sensitive frame control data fields.

HomPlug also utilizes the variant of well-known CSMA/CD technique protocol as the MAC layer protocol. Several features have been added to support priority classes, provide fairness, and allow the control of latency. The MAC also added the 56-bit DES (Data Encryption Standard) to provide the security of data transmission. It is probably announced by Intellon Corp that the data rate of HomePlug currently is up to 14 Mbps and 100 Mbps in [15]. HomePlug has released HomePlue 1.0 specification in June 2001. And the next generation HomePlug Specification will support distributions of data and multi-stream entertainment including High Definition television (HDTV) and Standard Definition television (SDTV) throughout the home. More details about HomePlug, please see the references [13][14][15][16].

2.2.4 HomeCNA

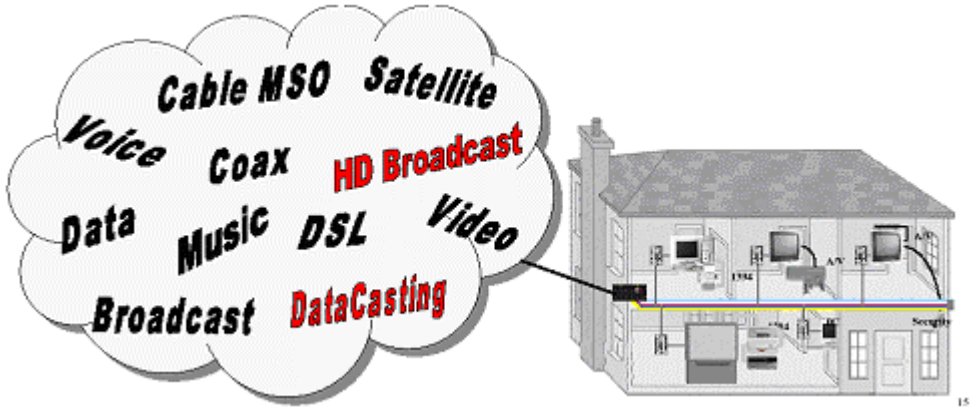


Figure 2.5: HomeCNA Networking Topology [18]

2.2.4.1 History

HomeCNA (Home Cable Network Alliance) [17] is established in April 2001. “HomeCNA Vision: Leverage the preponderance of coaxial wire in the home as the networking medium for entertainment, voice and data distribution in the connected home” described in HomeCNA Alliance presentation [18]. Figure 2.5 is the brief of HomeCNA Networking topology.

2.2.4.2 Overview

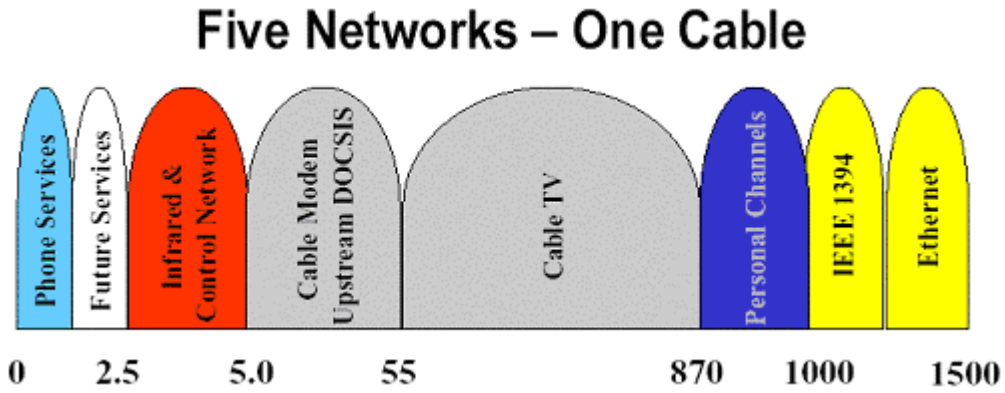


Figure 2.6: HomeCNA : Five Networks – one cable [18]

HomeCNA is an alliance focus on developing a coaxial coax solution that will distribute entertainment, telephony, voice, data, and controls simultaneously on the coaxial cable. The coaxial is common at home recently. And HomeCNA compatible products will coexist with existing coaxial services. Besides, The interoperability with other organizations that develop the home networking system standards such as OSGi is considered too. Figure 2.6 is the frequency band allocation of HomeCNA. Different network services are allocated into the different frequency bands. So those services can be coexist in the same coaxial cable.

The bandwidth of HomeCNA is up to 100 Mbps and can be extendable up to 400 Mbps. Besides it also support full QoS for video, audio and telephony through IEEE 1394 networking. See [17][18] to get more detail data bout HomeCNA.

2.2.5 IEEE 1394

2.2.5.1 History

The predecessor of IEEE 1394 (1394-1995 IEEE Standard for a High Performance Serial Bus) [19] is the “FireWire” which was conceived in 1986 by Apple Inc. The first version specification of this link was completed in 1987. Then it was adapted as IEEE 1394 standard in 1995 by IEEE (The Institute of Electrical and Electronics Engineers). IEEE 1394a is completed in 1998 and IEEE 1394b is introduced in 1999. Furthermore, IEEE 1394b is downward compatible with IEEE 1394 and IEEE 1394a.

2.2.5.2 Overview

IEE 1394 is the digital link specification focus on multimedia interface and has many features, for example, the low-cost, high-bandwidth real-time data communication and so on. It is very common standard interface in multimedia device such as DV, Digital Camera, DVD Player, VCRs, PCs, etc. IEEE 1394 support two data transferring mode – Asynchronous and Isochronous. The Asynchronous data transferring mode emphasizes more on guaranteed data delivery but less on guaranteed timing. The Isochronous is opposite, it emphasizes more on guaranteed timing and less on guaranteed data delivery. Therefore the Isochronous is quite suitable for multimedia data. And the Isochronous has the features such as no error correction, no retransmission. It always transfers data by broadcasting in a one-to-one or one-to-many fashion.

The bandwidth of IEEE 1394-1995 is up to 50 Mps within distance up to 4.5 meter. IEEE 1394a supports the speed of 100Mbps, 200Mbps and 400Mbps. IEEE 1394b extends the distance to 100 meter and data rate up to 800 Mbps further. The next version of the standard will support up to 3.2 Gbps data rate. Moreover, IEEE1394 standard also support play and plug so user do not need to reset network when they add or delete IEEE 1394 devices and up to 63 device attached via a single bus connection.

2.2.6 USB

2.2.6.1 History

The first version, USB (Universal Serial Bus) v1.0 specification [20], was introduced in January 1996, and the second version, USB v1.1 [21], in September 1998. In April 2000, the newest version, USB v2.0 specification [22], was introduced. The major members of USB organization are IBM, Intel, Philips, Microsoft, NEC, Northern Telecom and HP etc. Because Microsoft add USB support in their Windows products started since 1998, USB become the most common interface for high-speed data transmission currently in the PCs. Besides, there are more and more peripheral devices use USB as their communication interface to PCs.

2.2.6.2 Overview

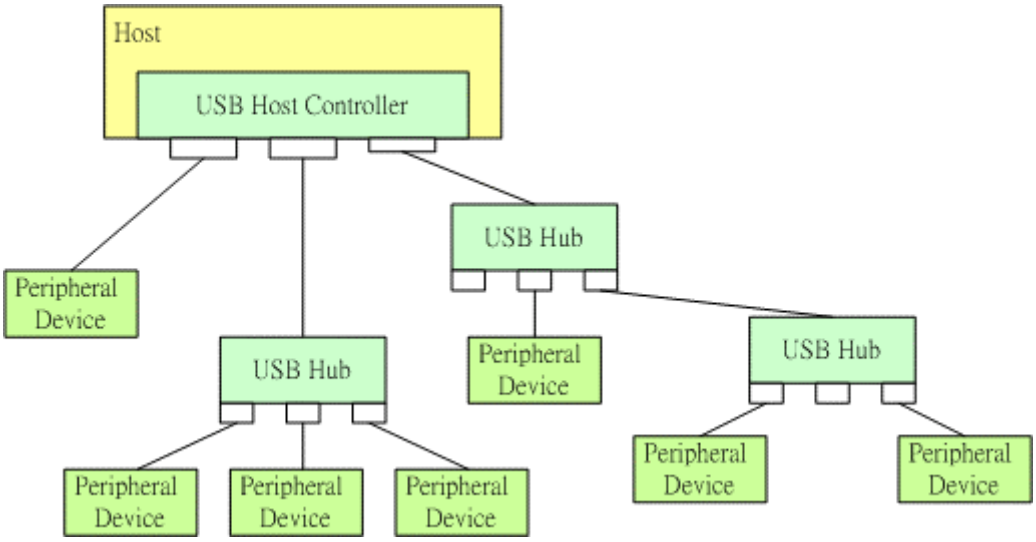


Figure 2.7: USB Network Topology

Figure 2.7 is the network topology of USB system that is host controlled. There are up to 127 devices, which can be connected to a single USB bus by using USB host controller, for example, the PC and USB hubs. A root hub (the host controller of Figure 2-7) can directly connect up to 7 USB hubs, and each hub on the root hub can be, in turn, connected to seven hubs, etc. to a maximum of seven tiers and 127 ports. This kind of network topology is generally called tiered star topology. A “Compound Hub” in USB standard means a USB peripheral device has a USB hub build into it, so it is comprised a function and one or more hubs. Besides, USB supports

plug-and-play and hot plug; therefore, if you plug the device directly, it will be configured automatically. Besides, the maximum length of USB cable is up to 5 meter.

There are two speed modes in USB v1.1 – 1.5Mbps (Asynchronous mode) and 12Mbps (Isochronous mode). USB v2.0 provides higher performance than USB v1.1. The speed of USB v2.0 is promoted to 480Mbps current, and it is also downward compatible with USB v1.1 speed modes. So when the peripheral device does not support USB v2.0 speed mode, USB v2.0 host will use the lower speed mode of 1.5Mbps or 12Mbps determined by peripheral device to work with it. Besides, USB v2.0 also supports a special transmission mode – OTG (On-The-Go). OTG is a specification allowed USB peripheral device can communicate each other without host. The more detail please refers to On-The-Go v1.0 specification [25]. And some related collections of USB technology is listed in [24].

2.2.7 RS-232

2.2.7.1 History

RS-232 was introduced in 1960, and is currently very common and widely used communication interface especially in PC and Industry field. It is standardized as EIA232 by EIA (Electronic Industries Association) [27]. The third version is RS-232C which is the standard of choice of PC in 1969. The fourth revision called RS-232D adopted in 1987. The newest version is named EIA232E introduced in 1991.

2.2.7.2 Overview

RS-232 is simple and low-cost network interface. It is a single-ended data transmission system. It uses a single wire for data transmission. So it is much useful to support two-way communication using two wires: one is for receiving data, another is for transmitting data. RS-232 utilizes the negative logic to identify the signal status (or logic status). In this case, it uses the negative voltage level (-3V to -12V) to represent ‘1’, positive voltage (+3V to +12V) to represent ‘0’.

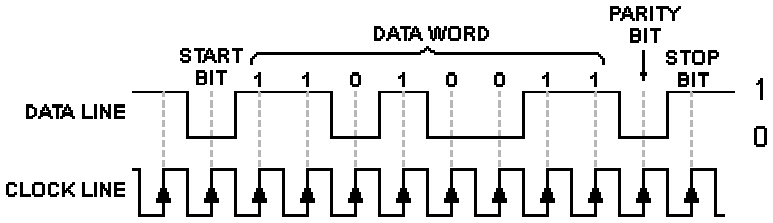


Figure 2.8: RS-232 data frame format

Moreover, RS-232 uses the asynchronous mode as its data transmission mode. So it needs a start bit for representing the start of data packet and a stop bit to represent the end of data packet. Besides, the length of data packet of RS-232 can be 5,6,7 or 8 bits. Furthermore, it also can add a parity bit, which can be odd or even parity bit form for error correction. The speed of RS-232 can be 1200bps, 2400bps, 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps or higher. Figure 2.8 depicts the brief waveform of RS-232 signal.

2.3 Wireless Technologies

2.3.1 Wireless LAN - 802.11

2.3.1.1 History

IEEE 802.11[28] is a very popular WLAN (Wireless Local Area Network) standard in the world and the responsibility of IEEE 802.11 Working Group is to develop the international standards for WLAN. And the Wi-Fi Alliance is the wireless standard group formed in 1999 to certify interoperability of WLAN products base on IEEE 802.11 specification. And now, there are many standards developed by IEEE 802.11 Working Group such as 802.11b, 802.11a, 802.11b etc. The 802.11b standard is the first finalized specification in September, 1999 by IEEE 802.11 Task Group b and it is current very popular and already has many compliant products in the world.

2.3.1.2 Overview

IEEE 802.11b chooses DSSS (Direct Sequence Spread Spectrum) as its modulation scheme in the physical layer. Its operation frequency band is 2.4GHz, the unlicensed ISM (Industrial Scientific and Medical) band, and the maximum data rate is up to 11

Mbps.

IEEE 802.11a is another standard version of WLAN. It chooses the OFDM as its modulation scheme and the maximum data rate is promoted to 54 Mbps. But it chooses the 5GHz band to be its operation frequency band. The 5GHz band has less interference than 2.4GHz band but not the unlicensed band in each country. By description above, we can know IEEE 802.11b and 802.11a are incompatible due to they adopt the different modulation scheme and operation band.

IEEE 802.11g is the standard focus on downward compatible with extension to the existing IEEE 802.11b and improves the data rate exceed 20Mbps. Due the MAC (Medium Access Controller) of IEEE 802 share a common LLC (Logic Link Controller), so all the services, such as the file sharing, http, e-mail etc. in the wired Ethernet, are made available to wireless Ethernet too.

IEEE 802.11 adopts the CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) as its MAC protocol. CSMA/CA is similar to CSMA/CD in the wired Ethernet MAC. It like CSMA/CD utilizes a “listen before talk “ mechanism to control access to share the same medium. But there no way to make sure that there is no collision between wanted client/receiver and server/transmitter due to it is possible existed a “hidden” node which wants to transfer data to receiver and is within range of the receiver but out of range of the transmitter. So we cannot monitor the idle or busy state of the medium while transmitting. Therefore IEEE 802.11 adopts the CSMA/CA can solve the above problem efficiently to replace CSMA/CD. The more detail information please refers [28][30][31].

2.3.2 BlueTooth

2.3.2.1 History

BlueTooth SIG (Special Interest Group) [32] was founded by Ericsson, IBM, Nokia, Intel, and Toshiba in February 1998. The mission of BlueTooth SIG is to develop an open standard for short-range wireless technology called BlueTooth. L. M. Ericsson of Sweden invented BlueTooth technology first in 1994. The first version of specification, BlueTooth V1.0b, is released in December 1999 and the second reversion, BlueTooth V1.1, in February 2001. Moreover, the second version is downward compatible to v1.0b.

2.3.2.2 Overview

BlueTooth is an open standard version of WPAN (Wireless Personal Area Network). Identical to IEEE 802.11b, BlueTooth v1.x adopts 2.4 GHz as its operation frequency band but it utilizes FHSS (Frequency Hopping Spread Spectrum) as its modulation scheme. However, compared with IEEE 802.11, BlueTooth has different market partition The hopping frequency of BlueTooth is 1600 hops per second, the frequency spectrum is normally divided into 79 channels with 1 MHz channel spacing per channel. The maximum of transmitting distance is up to 10 meter at 0 dBm transmitter RF power and up to 100 meters at 20 dBm.

BlueTooth supports two data transfer modes, one is ACL (Asynchronous Connection-Less) for data packet, and the other is SCO (Synchronous Connection-Oriented) for voice packet. The maximum of data rate of ACL is up to 723.2 kb/s asymmetric (and still up to 57.6 kb/s in the return direction) and up to 433.9 kb/s symmetric.

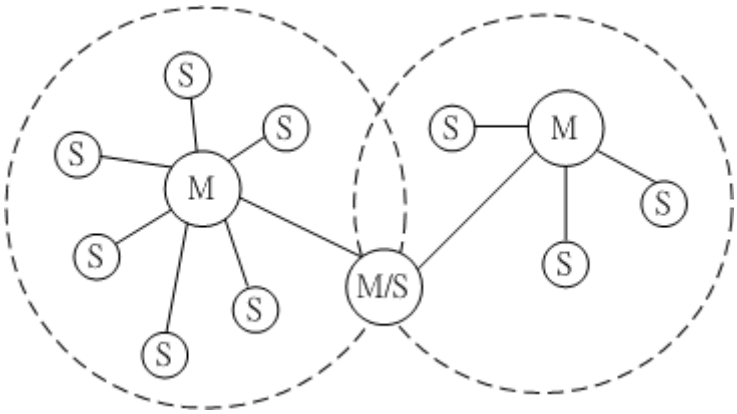


Figure 2.9: BlueTooth Scatter net Network Topology

The network topology of BlueTooth is a master-slave based Ad-Hoc network. It supports up to 7 slaves with a single master in a piconet organized into groups of two to eight devices, consisting of a single master and one or more slave devices. A device in a piconet may additionally belong to more than one piconet, either as a slave in both or as a master in one piconet and a slave in another. These devices are bridges to connect the different piconets and form the expanded the network topology called scatter net that is shown in Figure 2.9.

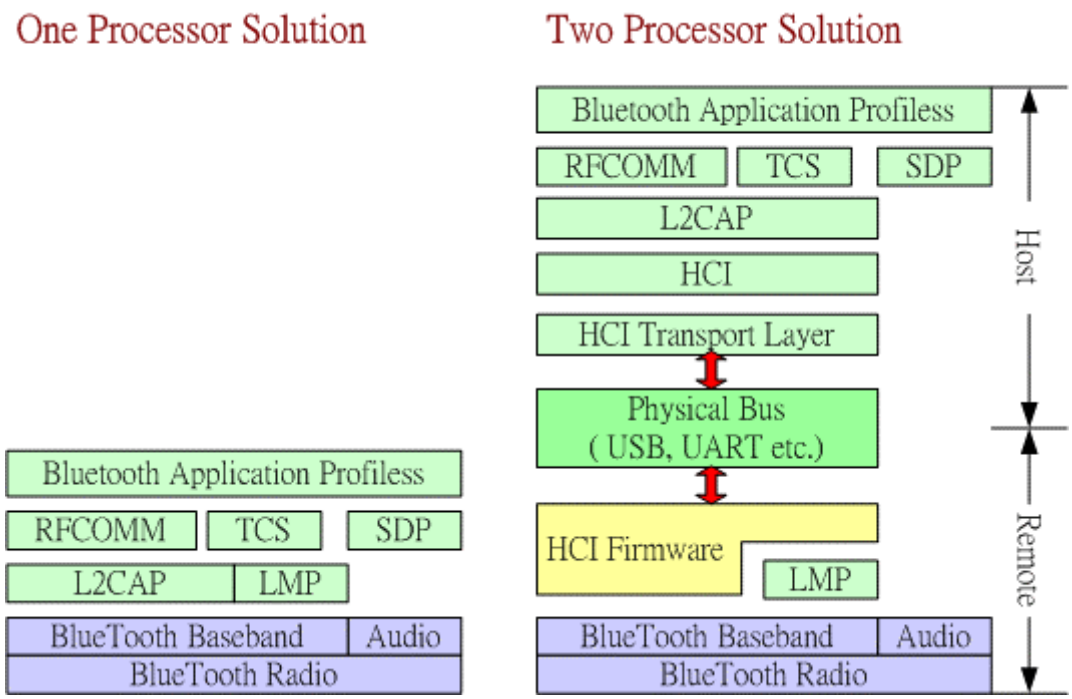


Figure 2.10: BlueTooth Protocol Stack

Figure 2.10 is BlueTooth protocol stack, the left of this figure is the One Processor Solution architecture and the right hand side is the Two Processor Solution architecture that means it has another processor as a host (usually called host controller) to control BlueTooth module at the remote side. The Two Processor solution uses the HCI (Host Controller Interface) interface to separate the protocol stack of the One Processor solution into host and remote sides. The HCI provides the uniform interface for different physical bus such as USB, UART, PC Card, etc. In this case, you need to choose the suitable HCI transporter layer standard and implement the HCI transport layer depending on different physical bus. Accordingly, host controller can communicate with BlueTooth module through HCI protocol through certain physical bus.

Each layer of the protocol stack of BlueTooth is very important and its responsibility for different functions. For example, the (Logical Link Control and Adaptation Protocol) L2CAP provides protocol multiplexing which allows different upper protocols, some of them are belong BlueTooth such as RFCOMM which emulate serial port which allows the legacy software can operate on a BlueTooth device, TCS (Telephony Control Protocol Specification) which provides for voice and data call control and SDP (Service Discovery Protocol) which provides a discovery mechanism to discover which BlueTooth specific services is available provided by

nearby BlueTooth devices, and the others are implemented by 3rd party company such as TCP/IP. The L2CAP layer also provides the fragmentation and reassembly of packets between layers, and negotiation and monitoring QoS (Quality of Service) between devices.

BlueTooth SIG also defines many profiles to ensure the interoperability between products from different companies. In the viewpoint of end user, the profiles ensure the common experience. In the viewpoint of application developer, the profiles specify the combined procedures from many basic standards, and reduce set parameter in basic standards. Each profile will lists the required protocol layers including BlueTooth specified such as base-band, LMP, L2CAP, SDP, RFCOMM, etc. and 3rd party specified such as TCP/IP, WAP, etc. Please refer [32][33][34][35] to get more details about BlueTooth technology.

2.3.3 HomeRF

2.3.3.1 History

HomeRF Working Group [36] was formed in 1997. The major members are Compaq, Ericsson, HP, IBM, Microsoft, Motorola, Philips, Proxim, and Symbionics. The main goal of this group is to enable the wireless data and voice networking within the home. HomeRF v2.0 is released in 2001.

2.3.3.2 Overview

HomeRF compared with the other wireless technologies, such as BlueTooth and IEEE 802.11, is the wireless technology focus on home networking at the consumer price points. HomeRF combined three technologies - IEEE 802.11, OpenAir, and DECT to provide a complete solution for wireless home networking. It integrates the data, voice, and media streaming in home networking. The voice channel is simultaneously up to 8 channels. It also incorporates the DECT (Digital Enhanced Cordless Telephony) standard, and provides a range of up to 150 feet currently. The speed of data rate is 10-20Mbps+ in HomeRF v2.x.

The same as BlueTooth, HomeRF also adopts the FHSS (Frequency. Hopping Spread Spectrum) technology as modulation scheme in 2.4GHz band due to FHSS is more interference immune than DSSS. And it also plans to use Adaptive FH

(Frequency Hopping) to avoid static interference to ensure the peak performance.

The MAC layer of HomeRF utilizes the CSMA/CA for data transferring, and the TDMA (Time Division Multiple Access) for delivery of voice traffic. It will be full compliance with IEEE 802.11a at 54Mbps for 5GHz band in the future version. The future version will be downward compatible with all previous released versions. It also supports the QoS and security. The issue of coexistence of existing wireless standards is also considered in HomeRF.

Chapter 3.

Home System Standards

3.1 Introduction

We have already introduced many home networking interface standards in Chapter 2. Then we want to give you an overview of the trend of the home system standards in this chapter.

A home system, which can be called “Framework” or “Middleware”, is system architecture to manage the services of the home appliances. Refer to the explanation of the Xilinx White paper [37] - “The home networking middleware is a layer of software that lies on top of an information appliance operating system. It provides “hook” of APIs to which home networking appliances can be attached”. On the other hand, each home appliance can access services and share resources with each other through home system. Besides, some of home system standards provide the capability of the multimedia service too. In conclusion, the home systems connect with all home appliances and manage all services of those appliances, so end users can access all services and share resources in the home network.

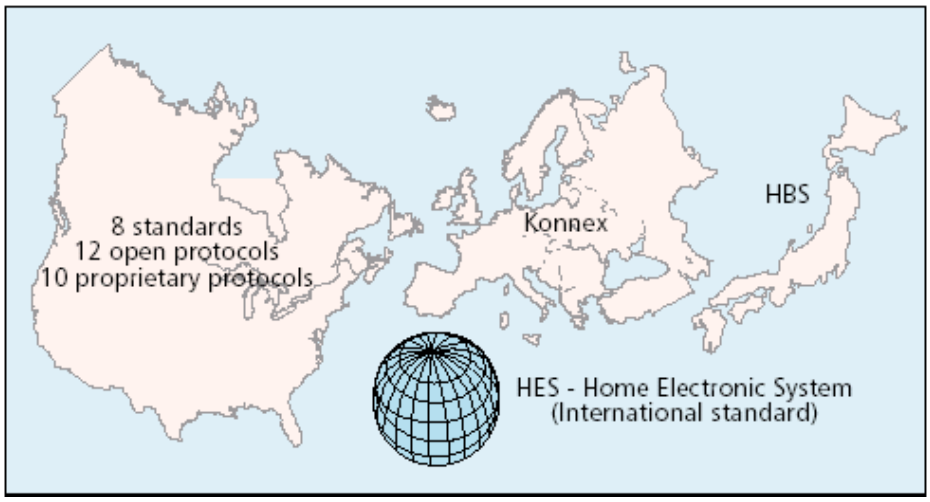


Figure 3.1: Home systems choice in 2002 [38]

In recent year, there are many home system standards, such as OSGi, UPnP, Jini and so on, are proposed and developed the related products around the world. As shown in Figure 3.1, we can see the choices of home systems in 2002. [38] Because there are so many choices of home systems especially in North America, it will confuse us to choose the suitable one. The Europeans have observed the phenomenon and start to combine three competing home systems in Europe into one called Konnex. Besides, HES (Home Electronic System) is an international home system standard, but it is not accomplished yet. The primary objective of HES is to provide a specify hardware and software that enable a manufacture to make only one version of product for connecting with other home networking standards. The more details are introduced in Section 3.3. Moreover, OSGi standard is similar with HES. It proposes a service framework running on the residential gateway to integrate all services provides by home appliances of multiple different home systems. The more details are described in Section 3.2. Furthermore, we introduce some representative home system in the remaining sections.

3.2 OSGi

3.2.1 History

OSGi (Open Service Gateway initiative) was founded in March 1999. There are over 60 members including Ericsson, Cisco, Nokia, Siemens, Sun Microsystems, and IBM. The fist version, OSGi SPR1, was released in May 2000, and the second version,

OSGi SPR2, in October 20001.

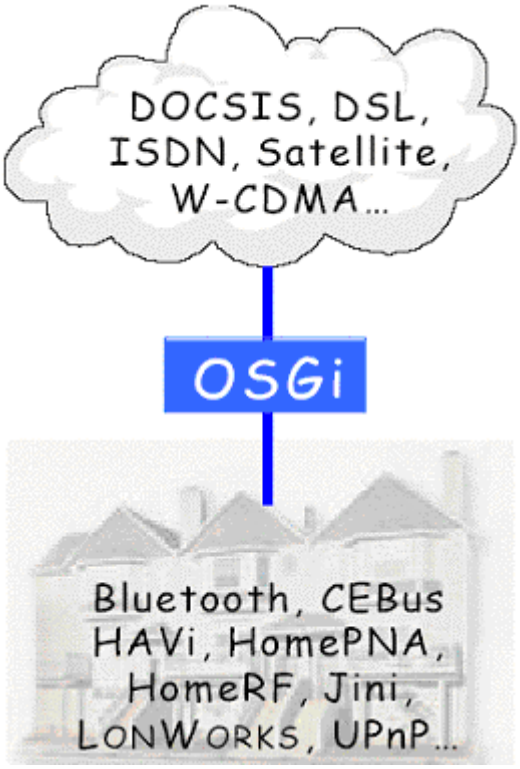


Figure 3.2: OSGi Gateway [44]

3.2.2 Overview

OSGi is a standard to integrate whole home appliances into a residential gateway by services. In Figure 3.2, we can see that OSGi attempts to integrate not only home networking interfaces but also home systems. It provides a open architecture for service providers, service aggregator, service gateway provider, WAN provider, Internet service provider, appliance manufacture, equipment manufacture, and home network system developer [38][42][44]. In other words, it will create a new business model and bring huge market.

3.2.3 Feature

The main features of OSGi are described following [37][42][43]:

- **Platform Independent:** OSGi only defines a set of API (Application program interfaces). The API can be implemented by any hardware platforms and operating systems. Besides, it adopts Java technology as its main choice for implementation. Java is an excellent solution with the

feature of platform independent. In this case, anyone is able to develop OSGi product individually and the interoperability can be ensure too.

- **Security:** OSGi specification provides many levels of security features, ranging from digital signing of downloading modules to fine-grained object access control.
- **Multiple services:** OSGi supports the multiple services from different service provider on one service gateway. It provides the flexibility to allow a gateway operator to offer wide services to customers.
- **Multiple local network and data access technologies:** The emerging technologies- the local network, such as Jini and HAVi, and the data access technologies, such as BlueTooth, IEEE 1394 and HomeRF, are all supported by OSGi specification. All of them can be cooperated and added in OSGi platform.
- **Coexistence with other standards:** OSGi tries to define a specification to integrate whole emerging and former standards together. In other words, OSGi is not an independent standard but is a standards which to make interoperability and coexistence with other standards.

3.2.4 Architecture

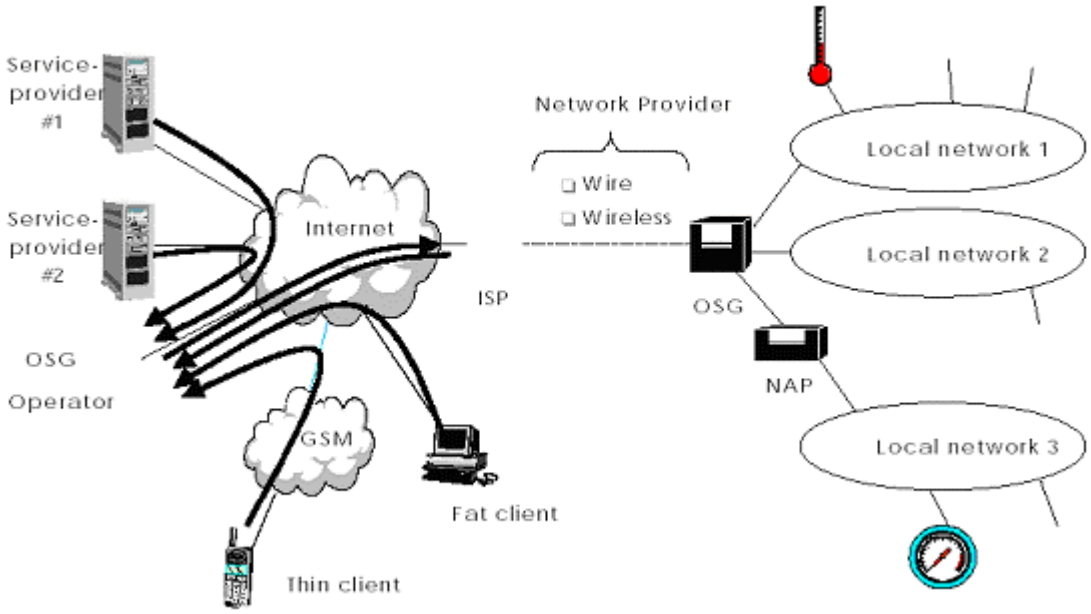


Figure 3.3: OSGi End-to-End Architecture [42]

As shown in Figure 3.3, OSGi end-to-end architecture, we can see the overview of the relationship between major entities. The major entities are explained individually following:

- **Service gateway:** The service gateway is the central component of OSGi and is the OSG (Open Service Gateway) in Figure 3.3. It is the gateway, which acts as the bridge between wide area network and internal network, and is attached to WAN to connect the external service provides to internal clients. The OSG is secure, zero-administration device to connect to devices on the local network. Its purpose is to provide communication between local devices and devices and services on the WAN. Besides, it acts as the service manger to services that operate on it.
- **Service provider:** The service provider contains a set of services and provides them to OSG. If OSG operator trusts a service provider, the services can be downloaded securely by OSG and installed on it from this service gateway.
- **OSG operator (Gateway operator):** The OSG gateway operator is responsible for managing and maintaining OSG and its services. Some functions of its are listed following:
  - Download, start/stop, update and remove a service
  - Gateway management
  - Link security between WAN and OSG
  - Check the validity and the rights of any dependencies that one service may have on other services
  - Manage services
  - Etc.
- **Wide area network and carrier/ISP:** WAN (Wide Area Network) provides the necessary communication service provided by ISP among OSG, service gateway, OSG gateway operation, service aggregator, and remote terminals. Therefore, OSG can connect to WAN through the carrier/ISP.
- **Local devices and networks:** The local devices are attached to OSG by connecting with hardware interfaces, for instance, parallel or serial connections directly or various home networking interfaces such as BlueTooth, IEEE 1394 or Ethernet indirectly.

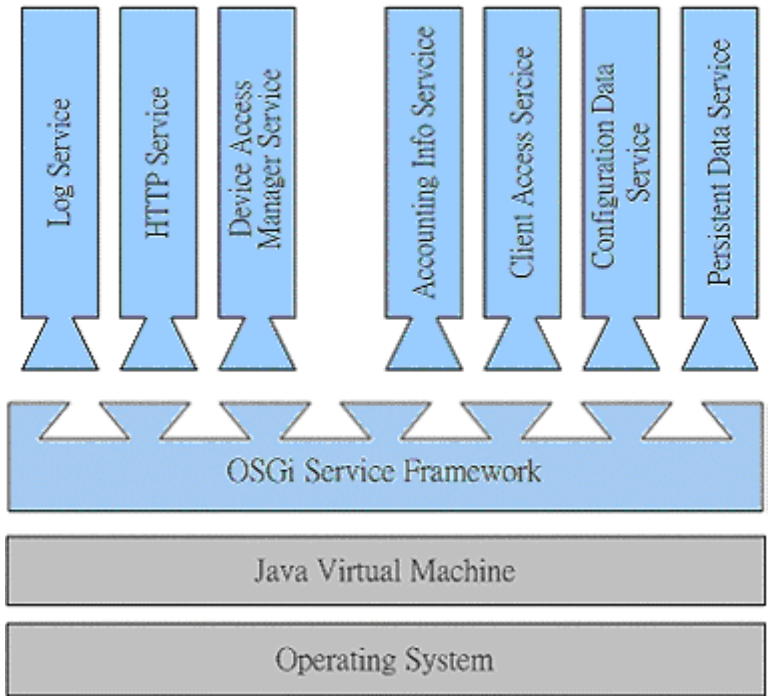


Figure 3.4: OSGi Framework Architecture

Figure 3.4 shows the framework architecture of OSGi, which is the service framework running on top of Java virtual machine. OSGi provides the execution environment for services that may be packaged as a bundle, which is a Java jar archive. The bundle consists of Java classes and some necessary data needed by these services. In Figure 3.4, the items above OSGi service framework are some important services defined in OSGi specification. Each of them will be briefly introduced briefly following [42][43][44]:

- **Persistent data service:** The persistent data service provides a set of APIs to help a service to store and retrieve information from other services. Therefore, you do not have to implement a code to efficiently store and retrieve information to a file system and recover form error. You just need to implement a common persistence data service to provide other services to store and retrieve information. That is, a service can utilize API to search the persistent information store by a high level query language recover from failure after partial updates, and synchronize the data with a server database. Besides, implementing persistent data service can provide a mechanism to backup the persistent information on the server as an extra measure of reliability.

- **Configuration data service:** Many services need more additional configuration information to perform their task. The configuration data service can provide the common service API for the services to store and access the configuration data and a means by which a remote administrator can change configuration information.
- **Client access service:** The client access service provides an interaction mechanism for end users to access the services on OSG. For instance, you can maintain the services from remote server or access the services from client terminals, such as mobile phone and PDA.
- **Accounting info service:** In OSGi Release 2.0, it has defined the accounting info service to provide a charging mechanism depending on service usage. For example, the power energy company can use the accounting info service API to charge for the bill of power usage.
- **HTTP service:** The HTTP service provides an API for the HTTP based web server, which runs as a part of OSG. Hence, you can use the API to configure the server, publish static content, and publish dynamic content generated from Java serverlet.
- **Device access manager service:** The OSG can on-demand, dynamically download related services for devices and the home networking hardware connected to OSG by the device access manager service. Besides, the service provider can access and communication with devices without being burdened with low-level communication details. The device access manager service has two major components: one is device bundle, which is dedicated for communicating with special type of device, and the other is network bundle which is dedicated for communicating with devices attached to the network. The network bundle may contain the needed protocol stacks, drivers, and the other resources. The device access manager service has following features:
  - ***Network Independent*** – If a new network needs to be supported by the device access service, you only need to add the new network bundle.
  - ***Device Independent*** - If a new device needs to be supported by the device access service, you just need to add the new device bundle.

- ***Automatic Discovery*** – The device access service will support the feature of automatically detecting the new device or network, when they are attached to the OSG. Besides, the new bundle, which belongs to the new device or network, will be downloaded and installed with the minimum user interactions.

- **Log service:** The log service provides that the Java-based OSGi services can write the entries to a log and read the entities from log in consistent way. Moreover, the log service is platform independent. The log service has the following general features and capabilities:

  - Records the current system time, the severity level, the calling bundle id, the text message and an optional Java Throw-able object that describes the exception condition in the log entry.
  - Records the identity of the Java Program running on the framework, which created the log entry.
  - Listens to framework events that Release 2.0 has defined the concise definition of the output format and create log entries representing these events.
  - Reads past log entries via an enumeration
  - Notifies the listeners of log entries as the entries are created

3.2.5 Related reaches

Actually, there are many OSGi related open source projects and commercial solutions in the world. We generally list some of them in Table 3.1. [44] [48] [49] [50] [51]

Open Source	
Project	Description
OSCAR	Oscar project implements OSGi specification. It is a platform for experimenting with service-oriented computing approaches. [45]
Mister Bundles	The goal of the "Mister Bundles" project is to implement Mister House-like services coded with Perl under the form of OSGi bundles (java code), compliant to OSGi v2.0 framework specifications. [46]

Oxygen	Oxygen is an OSGi gateway implementation in Java. It is designed to run on small footprint and embedded devices that run a Sun J2ME. [47]
Commercial Solution	
Company and Solution	Description
Sun – JES [48]	JES (Java Embedded Server) is provided by Sun. It is the key tool to develop OSGi-based products and a small footprint application server that can be embedded into a network device. JES is fully coded by Java and the newest version is JES 2.0 now. Please refer [48] to get the more details.
IBM – VisualAge Micro Edition [49]	IBM company has released the version 1.2 of VisualAge Micro Edition, which supports interactive OSGi bundle management and real-time Java components. Please refer [49] to get more details.
Ericsson – OSGi development tools	Ericsson has published the E-box, the first OSGi based product, in late 1999. And now, Ericsson offers two development tools: one is Residential e-Service-Application Development Package 2.0 and the other is Automotive e-Service – Software Development Kit to help developers to develop OSGi applications running on Ericsson e-Server system. Please refer [50] to get more details.
Gatespace – SGADK	Gatespace was founded in 1999 with Ericsson and Swedish research and consulting company Carlstedt Research & Technology (CR&T). Gatespace provides the tools: GDSP (Gatespace's Distributed Service Platform) and the subset of GDSK - SGADK (Gatespace's Service Gateway Application Development Kit) based on the Ericsson e-Service system. Therefore, developers can use them to develop OSGi based application. Please refer [52] to get more details.

Table 3.1: OSGi Related Projects

Furthermore, there are several related researches about OSGi. Someone has developed the solution to integrate current home system standards, Jini and UPnP, into OSGi system. [52] Currently, they have completed Jini-to-OSGi and UPnP-to-OSGi

transformation technologies. Prosyst Software AG has announced their implementation of OSGi. [53] It has integrated with many existing home networking standards. Moreover, there are two theses focused on the special topics of OSGi. One [54] implemented OSGi platform, X Service Gateway. It has integrated X10 home system based on Linux platform. And the other [55] adopted Sun JES 2.0 as its OSGi base to implement their OSGi platform.

3.3 HES

3.3.1 History

HES (Home Electronic System) [55] is an international home system standard proposed by the Working Group 1of ISO/IEC. The Working Group is officially called ISO/IEC JTC 1/SC 25/WG 1. WG1 has experts coming from different countries including United States, Germany, Europe, Japan, and Korea. HES does not accomplish the standard specification yet. It just announces the related technical reports [55].

3.3.2 Overview

HES specifies a platform with hardware and software that enable a manufacture to offer one version of product that can connect with various home networking systems. Following components are defined in HES draft [57]:

- **Universal Interfaces:** An interface module is incorporated into an appliance for communication with various home-networking systems.
- **Command Language:** A language is used for appliance-to-appliance communication independent of carrying message by what kind of home networking systems.
- **HomeGate:** A residential gateway standard provides a bridge between local network and WAN. In other words, device connected with local networks can access Internet resources via HomeGate. Moreover, authorized vendors or users can access the local services through it remotely.

3.3.3 Components

3.3.3.1 Universal Interface

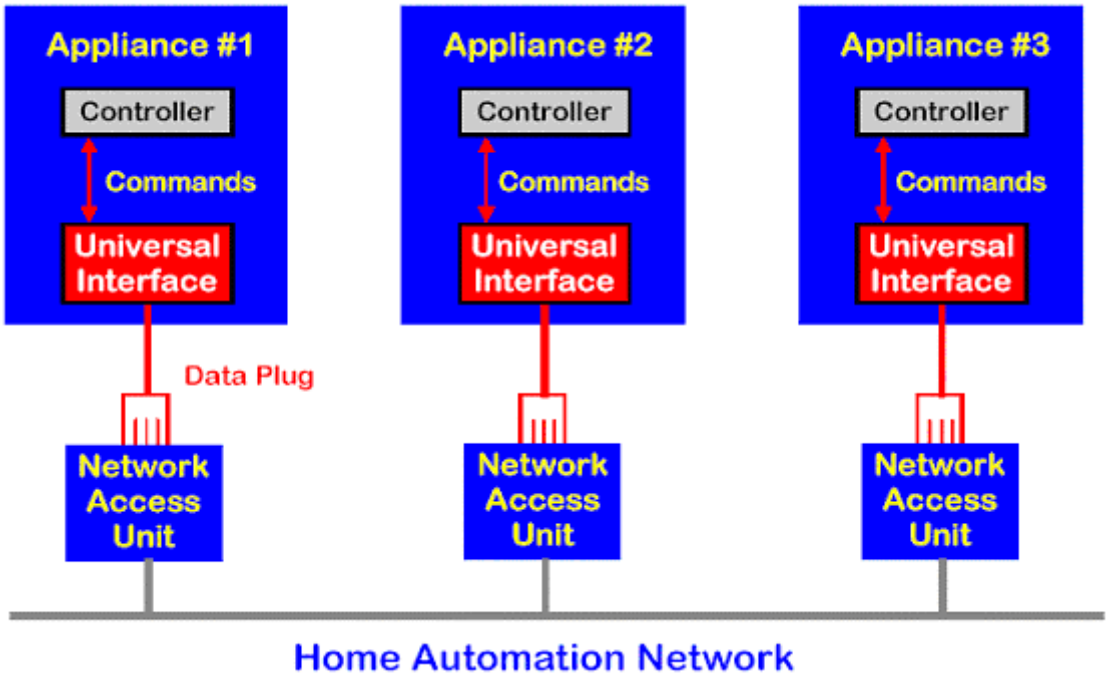


Figure 3.5: Appliances connect to Home Electronic System [57]

HES proposes a new home system, which can connect with any home automation network. Therefore, HES proposes a UI (Universal Interface) component that is incorporated into appliance and that has a standard data plug. Besides, every appliance that connects to the home automation network has a NAU (Network Access Unit), which is responsible for converting the data signal and appliance message to a particular home automation system protocol. Figure 3.5 depicts the architecture of UI. Also, HES specifies the communication protocol between UI and NAU [57].

3.3.3.2 HomeGate

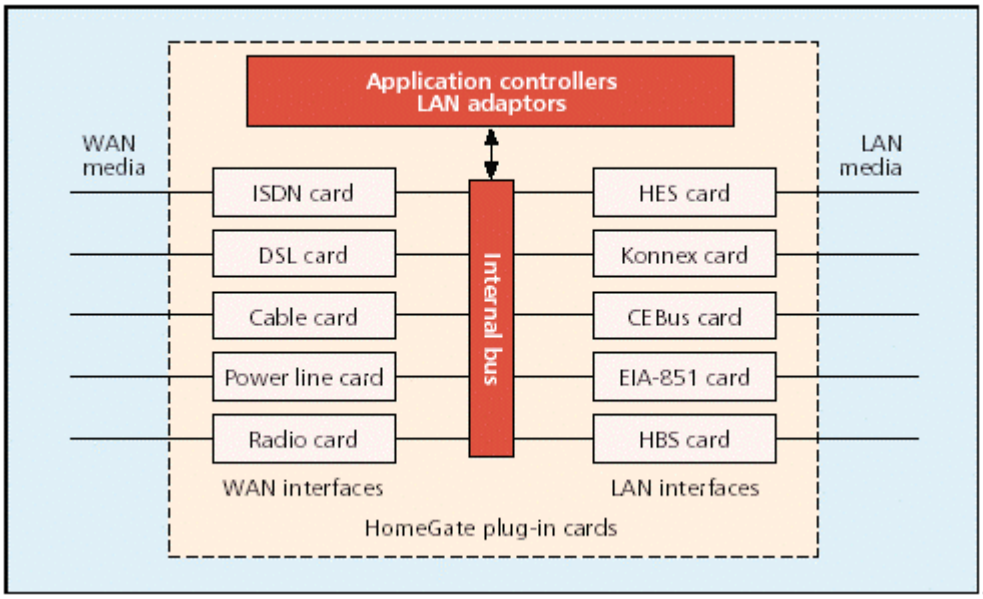


Figure 3.6: An example of HomeGate implementation [57]

HES also proposes a residential gateway solution – HomeGate. HomeGate plays the role of bridge between the local network and the external network (WAN). Moreover, HomeGate needs the firewall features to support the security and limits the types messages out and in the house. However, HES does not constrain how to implement HomeGate actually, but its suggested implementation is a box with plug-in cards as shown in Figure 3.6. In this case, HomeGate contains many home networking cards depending on different kinds of network interfaces used by WAN and LAN. Moreover, there are many home-networking protocol now, so HomeGate needs to add more LAN adaptors to translate the protocols among various local home networks. [57]

3.4 HAVi

3.4.1 History

HAVi (Home Audio Video Interoperability) organization [58] was founded in November 1999. The goal of HAVi association is to develop a home network standard based on IEEE 1394 network for interoperability and the multimedia data transmission among digital audio and video devices. In addition, Panasonic, Hitachi, Sony, Sharp, Toyota and Philips are members of HAVi association. The fist version,

HAVi v1.0, was released in January 2001.

3.4.2 Overview

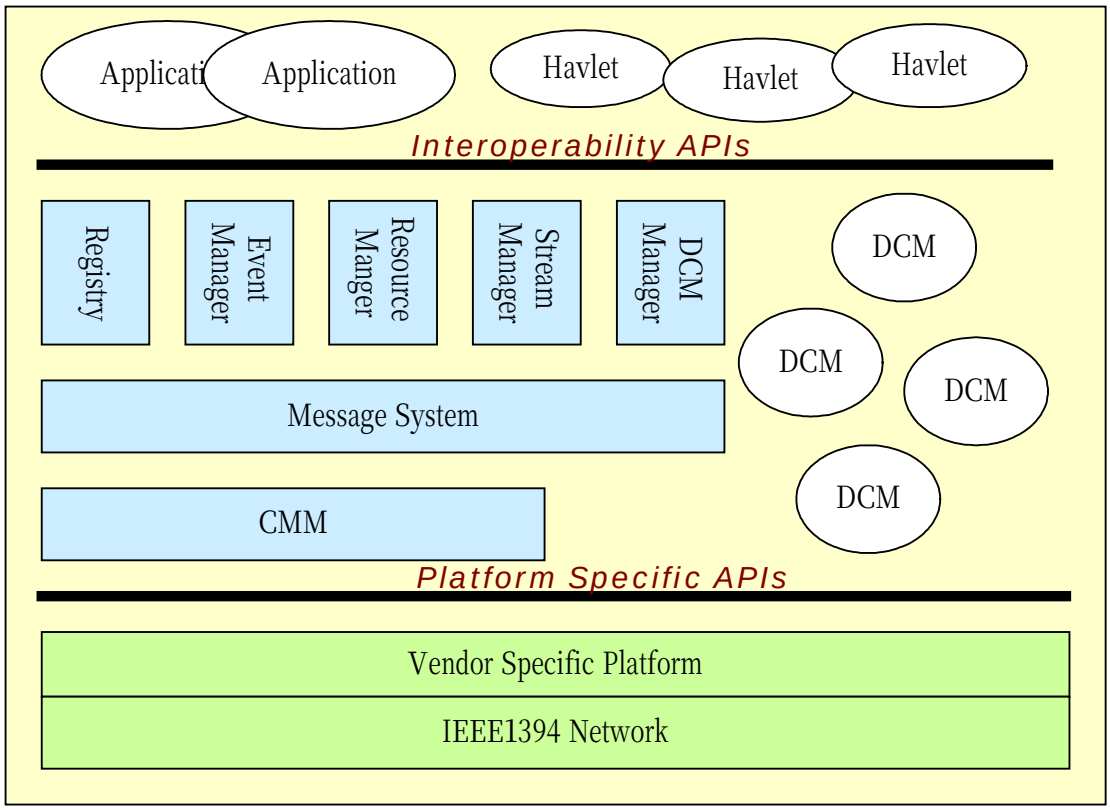


Figure 3.7: HAVi Architecture

HAVi is a home system for seamless interoperability among digital audio and video consumer electronics (CE) devices. Besides, HAVi organization defines a set of API to help developer to develop HAVi-based applications based on HAVi framework. Figure 3.7 depicts HAVi architecture; we can see HAVi is the software framework which comprises many system elements based on IEEE 1394 network. We describe each element of HAVi system briefly as follows [37] [58] [59] [60]:

- **Message System:** Message System provides communication facilities between HAVi software elements. It is independent of the network and transport layers.
- **CMM:** CMM (Communication Media Manger) is platform dependent. It provides the asynchronous and isochroous communication over IEEE 1394 network.
- **Registry:** It is responsible for managing a directory of software element available within HAVi network. Therefore, a software element can locate

other software elements on the network and detect its capabilities and properties.

- **Event Manager:** It is the event delivery service. Hence, Event Manager handles whole events of the software elements of HAVi system. For instance, delivering a local event to a specific device or a global event to all devices on HAVi network.
- **Resource Manager:** It facilitates sharing and allocation of resources and scheduling of actions.
- **Stream Manager:** It is responsible for managing the multimedia streaming service among A/V devices over IEEE 1394 network.
- **DCM:** It provides an interface of controlling general functions of a single device. It also exposes HAVi defined API for HAVi devices. Moreover, DCM is a software element that can be dynamically downloaded, removed and installed in HAVi network.
- **FCM:** FCM is contained within DCM. It represents each controlling function of a device. Besides, it is vendor specific.
- **DCM Manager:** DCM Manager is responsible for installing and removing DCM.
- **Havlet:** Havlet is a HAVi application written by Java. It offers a user interface for end users to control HAVi-base devices.
- **Application:** The application dependents on HAVi system and it can communicate with other software elements by mechanisms above.

3.5 Jini

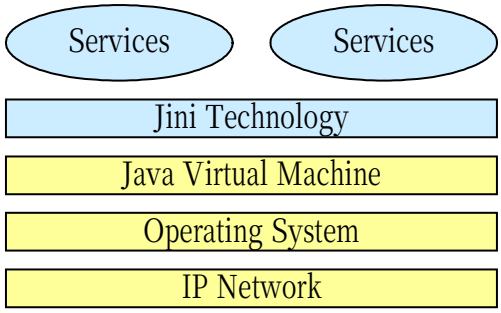


Figure 3.8: Jini Architecture

3.5.1 History

Jini [61] is a home system technology promoted by Sun Microsystems. Jini attempts to solve the interoperability of home appliances. It is announced during 1999.

3.5.2 Overview

Jini is similar to OSGi and HAVi; it is a middleware running on each device, which embeds the Java runtime environment as shown in Figure 3.8. Following lists the main features of Jini [37]:

- **Platform independent:** In Figure 3.8, we can see that Jini is a java program running on top of Java virtual machine, so it can get the platform independent feature of Java.
- **Small footprint:** To implement Jini solution requires very small code size even can be embedded into any type of home appliances. Sun Microsystems has implemented Jini into the coffee maker and the lamp.
- **Plug and Play:** Jini provides the mechanisms, lookup service, discovery and join-in protocols, to support the feature of Plug-and-Play. Therefore, the end users can access any service of home appliance with few interactions.
- **Legacy device support:** Jini utilize the proxy architecture to integrate the legacy devices into Jini network. Accordingly, Jini-based devices can access the services of the legacy devices through Jini proxy server.

3.6 UPnP

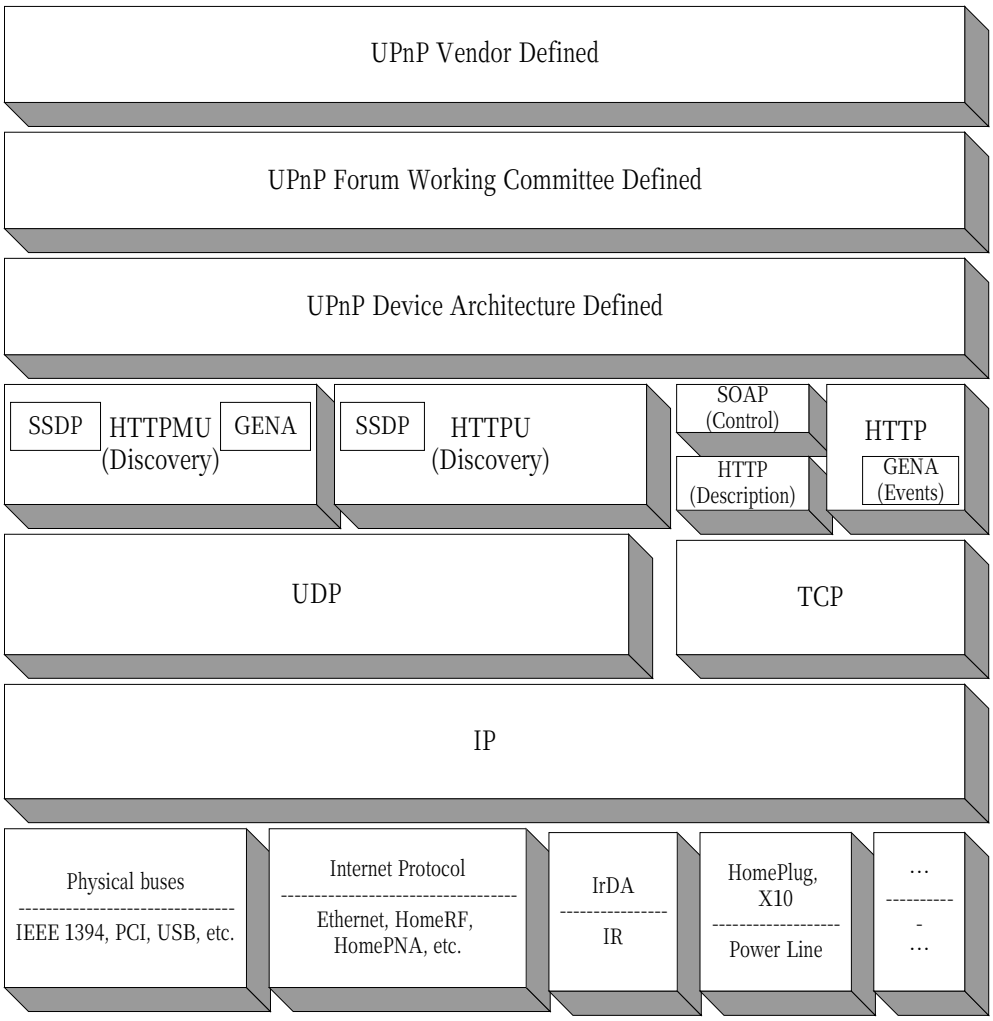


Figure 3.9: UPnP High Level Architecture [37] [63]

3.6.1 History

UPnP Forum [62] was formed by a group of companies in October 1999. Microsoft is the key player of UPnP Forum. The goals of UPnP Forum are to enable the simplify the implementation of network emergence of easily connected devices and to ks in the home and corporate environments.

3.6.2 Overview

Figure 3.9 illustrates the high level architecture of UPnP. We can see that UPnP comprises many industry standards including TCP/IP, DNS, HTTP, HTML, UDP,

LDAP, XML, XSL and ARP. Each protocol is responsible for special function. For instance, XML is used to represent the standard device and service description, HTTP is responsible for communication protocol among devices and so on. UPnP has following features [37]:

- **Plug and Play:** UPnP adopts the combination of industry protocols, such as XML, HTTP, SSDP, SCPs, etc., to cooperate the feature of Play and Play. Therefore, uses can access the services of UPnP-based devices with the interactions as less as possible.
- **Platform Independent:** UPnP is independent of any operating system and physical device. It is running on top of IP layer. Therefore, UPnP can be implemented into any device supporting IP network.
- **Small footprint:** UPnP can be implemented into the device with few RAM and ROM size. Typically, these devices consist of low-priced microcontroller, 200-1000 Kbytes of RAM and flash ROM [37].
- **Integration with legacy and non-IP devices:** The devices, without capability of sufficient hardware resources, can be also integrated with UPnP by UPnP-based bridges. So legacy devices and non-IP devices can share UPnP resources with each other via UPnP-based bridges.
- **Peer-to-peer network architecture:** UPnP is a peer-to-peer network architecture; it means that it does not need the central control device to handle whole interactions among UPnP network. In other words, each device can directly communicate with others.

3.7 X10

3.7.1 History

X10 Ltd. [64] is a USA company who designs and markets home automation devices based on X10 protocol. Pico Electronics in Scotland developed X10 protocol. X10 patent expired in December 1997. Therefore, X10 protocol is now an open standard. Moreover, the industry leaders including RCA (Thomson), Radio Shack, IBM, and others, market X10 compatible products presently.

3.7.2 X10 Protocol

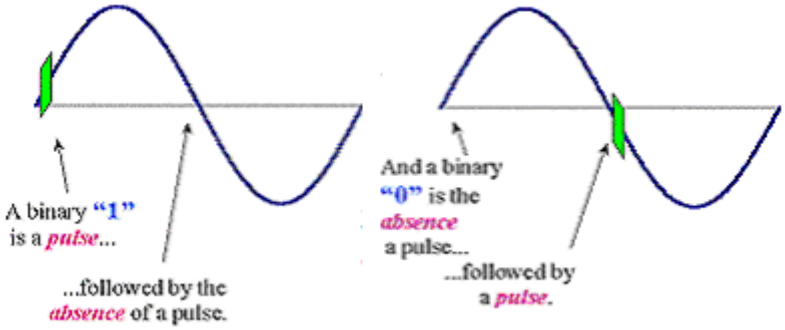


Figure 3.10: Waveform of binary data using X10 PLC [65]

X10 protocol first was developed for PLC (Powerline Carrier Technology). It utilizes the existing power lines of home as its network medium. Currently, it also supports the different medium: wireless RF and infrared. We introduce X10 protocol for PLC in this section. X10 transmits the binary data over the powerline by a 120KHz signal pulse for 1ms at the zero crossing point of the AC sine wave as shown in Figure 3.10. For reason of reducing error rate, it adopts two zero crossing points to transmit one binary data, either "1" or "0". The binary data, "1", is represented by a 120 KHz pulse at the first zero crossing point and no pulse at the second crossing point. The binary data, "0", is represented by no pulse at the first zero crossing point and a 120 KHz pulse at the second zero crossing point [65].

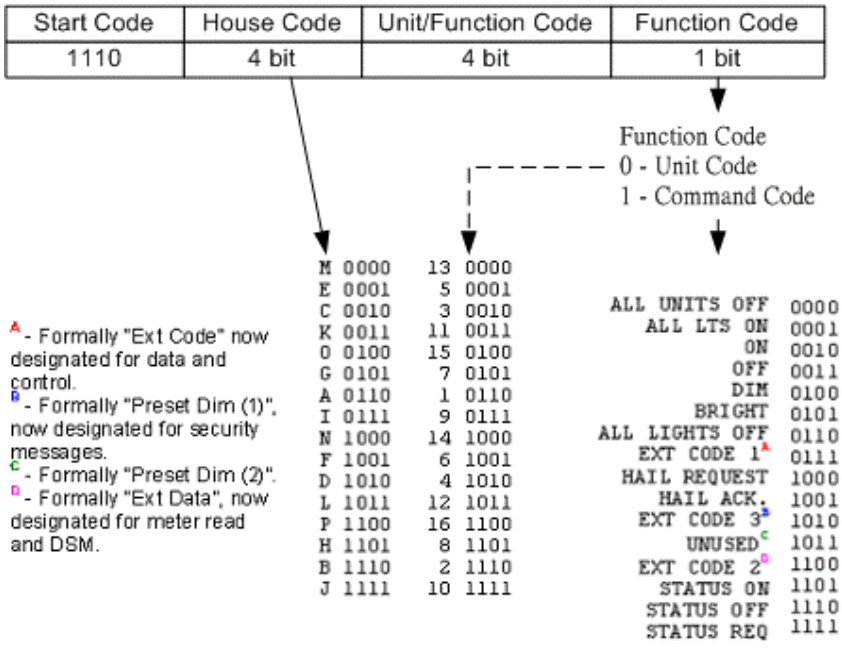


Figure 3.11: Basic X10 Message Format [65]

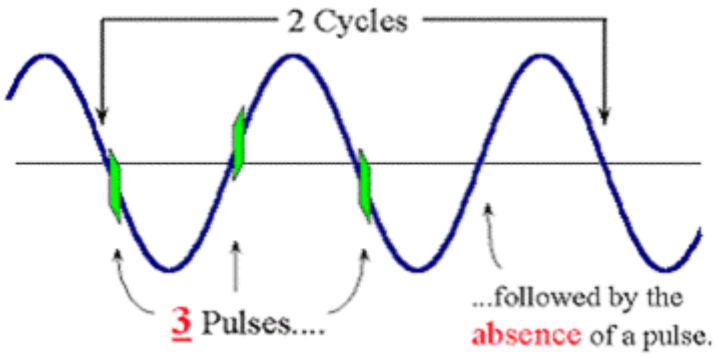


Figure 3.12: The Waveform of X10 "Start Code" [65]

A basic X10 message consists of 13 bits. The format is shown in Figure 3.11. It comprises 4-bits start bit, a 4-bits house code, 4-bits unit/function code, and 1 bit function bit. The start code is consecutive three pulses followed by no pulse to differentiate itself from regular data bits as shown in Figure 3.12. Moreover, the function bit indicates whether the previous 4 bits should be interpreted as a unit code or a command code. You can refer [65] to get the more details about X10 protocol.

3.8 Others

There are many home system standards including proprietary standards, United States Standards, open standards offered by companies and consortia and international standards. For instance, CEBus [87] and LONWORKS [88] are popular United States Standards. ECHONET [86] is an open standard promoted by Japanese companies. You can refer [38] [39] to get more details of other standards.

Chapter 4.

eHome System

4.1 Introduction

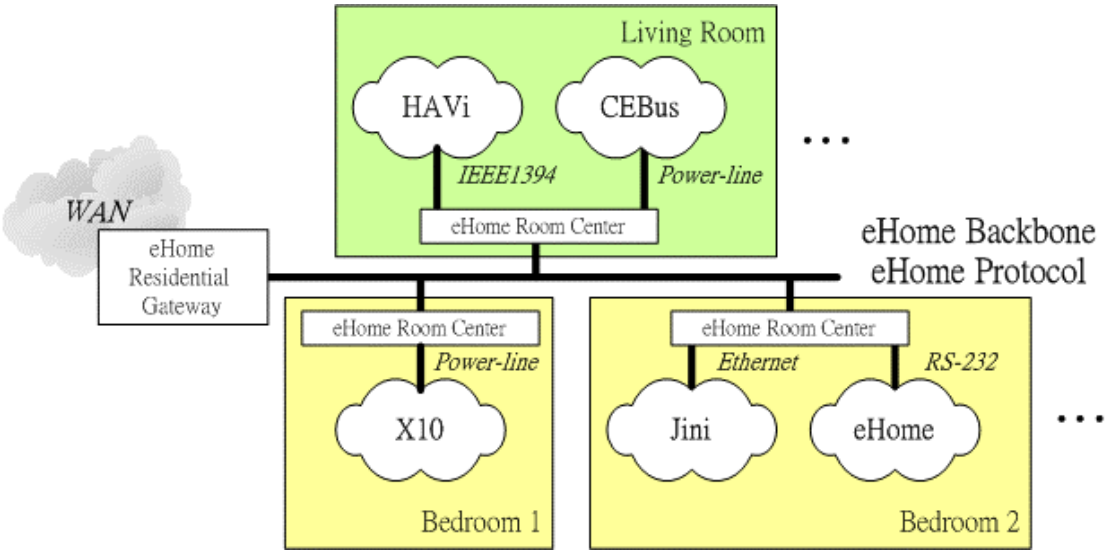


Figure 4.1: eHome Home System Architecture

In Chapter 3 and Chapter 2, we have already introduced many home systems and interface in home networking area. According to the trend, we can see that it is impossible to construct a home network system by the only one existing home system standard and interface. Moreover, we believe that there will be more and more home networking systems and interfaces will coexist in future home. Thereby, we try to

propose a new home system – eHome. The architecture is shown in Figure 4.1. The features of eHome system are low-cost, easy to construct, and high scalable and flexible. It is also integrated with the portable devices, which provide capability of remote access and the existing home systems and network interfaces. Moreover, whole existing home systems and interfaces will be cooperate through an eHome system.

The remaining sections of this Chapter are structured following. Section 4.2 will introduce some related projects in recent year. Their consideration in designing, proposed approaches and characteristics of proposed architecture will be also discussed. In Section 4.3, we will discuss and address the whole considerations in designing our eHome system. In Section 4.4, we will present our new home system – eHome. Section 4.5 describes the system components of eHome system. Finally, Section 4.6 defines eHome protocol and its characteristics.

4.2 Related Projects

4.2.1 VSG Based Home System

The VSG (Virtual Service Gateway) based home system is a framework architecture proposed by Eiji Tokunaga, Hiro Ishikaeam, and others [64]. This framework is dedicated for integrating heterogeneous home systems. It provides a desirable environment in which service based on every home system can connect any other services transparently. Moreover, a new home system can be connected to this framework without too much effort. Here home systems are also called middleware.

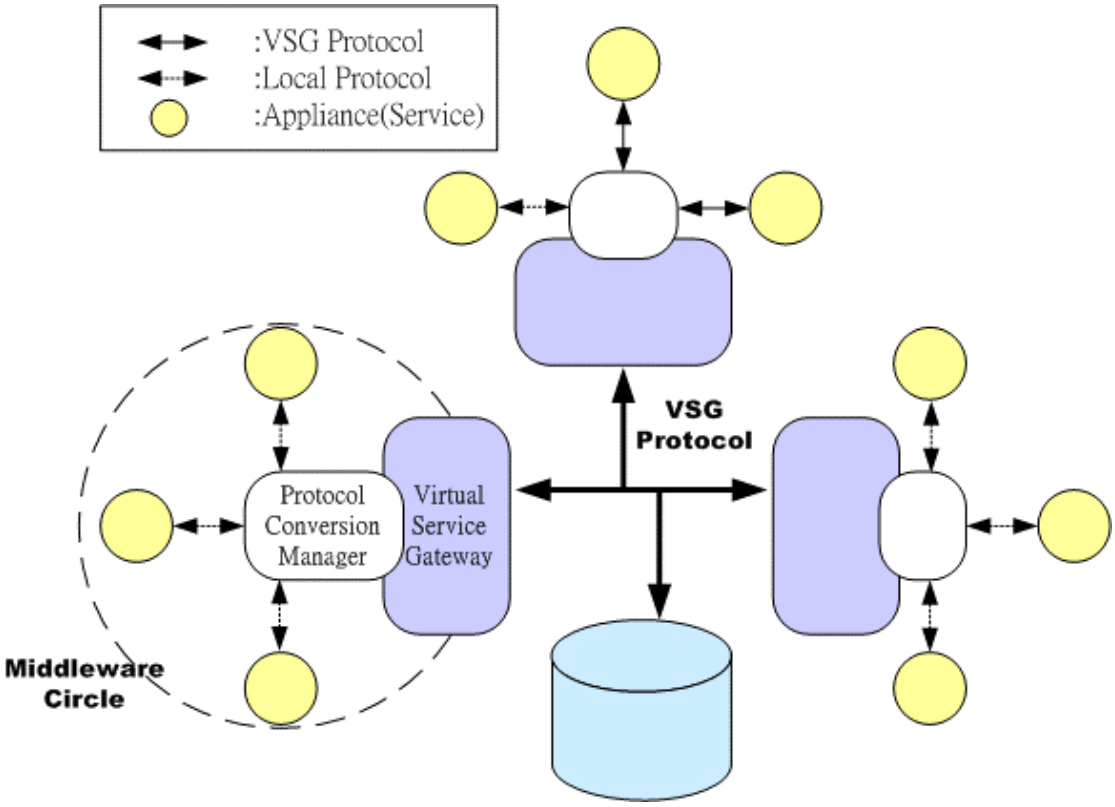


Figure 4.2: Overview of VSG based Framework

Figure 4.2 is the illustration of this framework. In Figure 4.2, each middleware has three main components: VSG (Virtual Service Gateway), Protocol Conversion Manager, and VSG protocol. Virtual Service Gateway connects a middleware to another middleware by certain protocol that decides the information of services such as interfaces, locations and data. Protocol Conversion Manager used to convert the local middleware protocol into protocol of Virtual Service Gateway and also VSG into local middleware. The VSG protocol is chosen depend on demand of how to integrate services. The service could be a multimedia streaming, a voice communication, or an action command. Lastly, Virtual Service Repository, a database, which has a lot of information of different middleware such as service locations and service contexts. The current status of implementation of this framework will be introduced in the following paragraph.

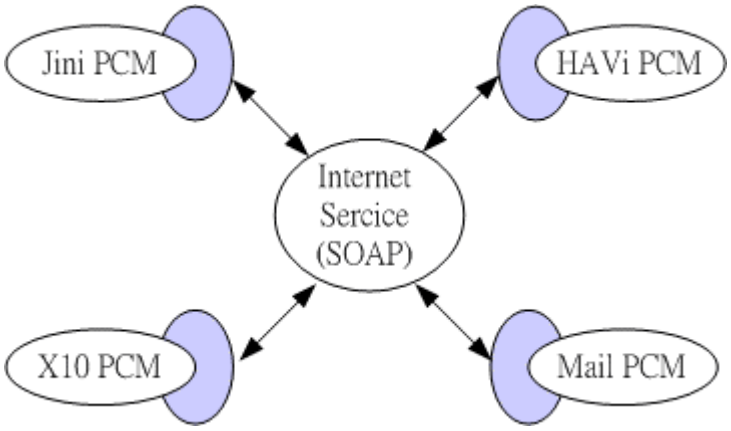


Figure 4.3: Prototype of VSG based Framework Implementation

They chose SOAP that is XML/HTTP-based protocol as the VSG protocol currently and utilize the Apache SOAP, a product of IBM developer works for current VSG prototyping. Figure 4.3 is the illustration of current prototype of their implementation. This prototype has integrated existing home systems: X10, Jini and HAVi. Please refer to [64] to get more details.

4.2.2 Virtual Overlay Network

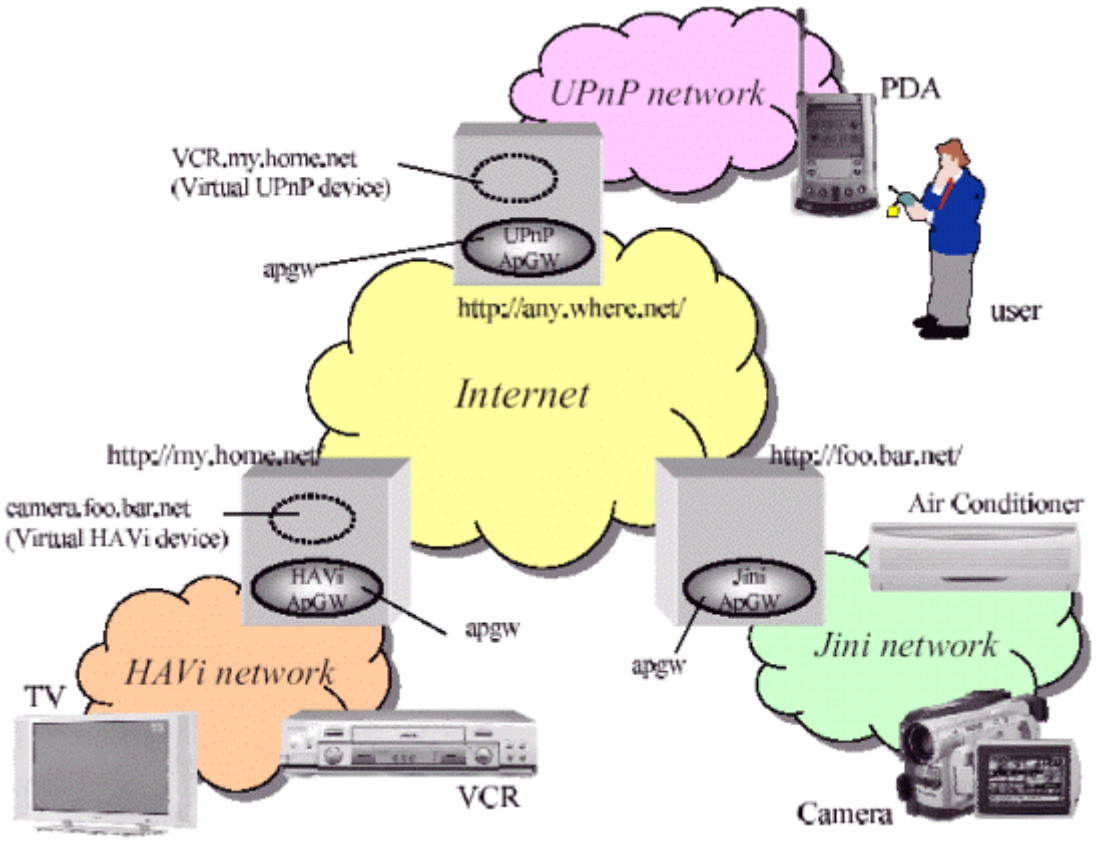


Figure 4.4: The Architecture of Virtual Overlay Network [66]

Daiki Ueno, Eiji Tokunaga, and others propose Virtual Overlay Network. [67] [68] It is similar with VSG-based home system. It also utilizes an application-level gateway to integrate existing home systems into their system. Application-level gateway of this system is responsible for integrating with all existing home systems and Internet. As shown in Figure 4.4 is the illustration of its architecture.

They [67] [68] also utilize the naming lookup and registration mechanism to access services provided by home appliances. They adopt URL as their identification of service/appliance location. Each appliance/service will have a unique URL. Typically, web browser accesses remote hosts by submitting a URI query request through either the HTTP GET mechanism or HTTP POST mechanism. HTTP returns the response of URI request. So your can send a URI request to further activate an operation of home appliance and the you can see the response of this operation through web browser. Following is the example of URI.

http://my.home.net/TV/!power=ON

The URL I just mentioned above can be use to turn on TV in HAVi home network. The application-level gateway will receive the URL request and then convert it into the request of HAVi home system to get the service (power function) of HAVi TV.

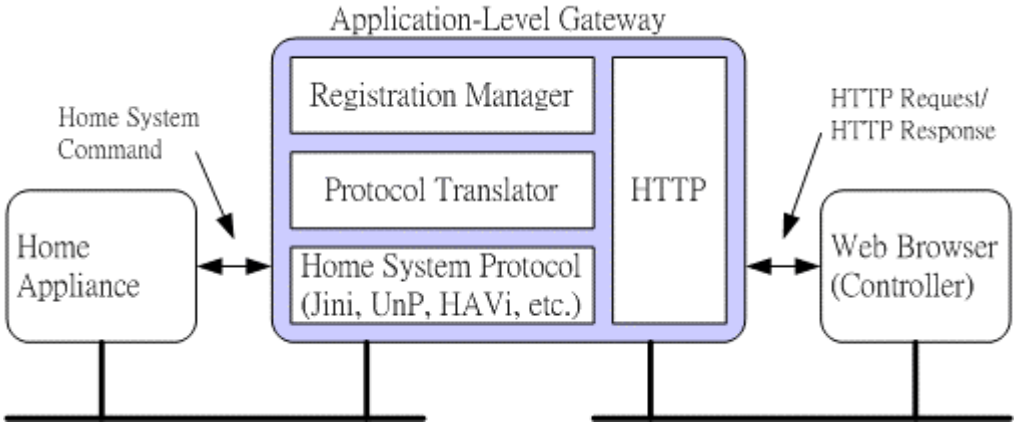


Figure 4.5: Prototype of VSG based Framework Implementation [67][68]

Figure 4.5 illustrates the architecture of Application-level gateway. There are four main components in application-level gateway. The first one implements home system protocol (middleware) such as Jini or HAVi. The second protocol implements the HTTP protocol. This component behaves as Web server. The third component implements a protocol translator that converts between the HTTP protocol and home

system protocol. The last component is registration manager. The registration manager is responsible for registering lookup services and managing the pseudo devices which are used to convert control commands that are delivered on home systems to HTTP protocol. It can be accessed from a web browser by the HTTP protocol or from appliances that implement the protocol by home system protocol.

For example, the device on Jini (Figure 4.4 Camera) wants to access the TV on HAVi network. It needs to register a unique URL and HAVi TV service as a pseudo device in registration manager of Jini application-level gateway first. When Jini Camera wants to access HAVi TV service, it will send the request to gateway (Figure 4.4 Jini ApGW), and then look up HAVi TV service in the registration manager of Jini ApGW. Moreover, the registration manager will return the pseudo device to Jini Camera. The control command will be delivered by Jini home protocol to pseudo device. Also the pseudo device will convert Jini-based request to HTTP protocol, then forwarding the request to HAVi ApGW according the host name part of "http://my.home.net/TV". For instance, if the completed URL is http://my.home.net/TV/!power=ON, HAVi ApGW will received the request deliver by HTTP protocol on Internet. It will find out "TV" by lookup service. Finally, the request will be further converted to HAVi command by HAVi home protocol to control HAVi TV.

4.3 Design Considerations

In following paragraphs we will address some design considerations before we propose new system – eHome system. Furthermore, we will offer some possible solutions to solve the integrating problems.

4.3.1 Multiple Home Systems and Interfaces

According to our observation, multiple different home systems will coexistence in the future home. Besides, there will be more and more network interfaces, such as BluTooth, Ethernet, IEEE 1394, HomeRF and Wireless LAN in the future home. Furthermore, digital appliances, such as PCs, PDAs, digital AV devices, and other various IAs will coexist and be connected by multiple different network interfaces and home systems in future home design. Finally, all of them will be able to share the

service with each other in the future.

However, two main problems may be raised. The first one is the interoperability of the services between different home systems. For example, it is difficult that HAVi-based service want to access the services provided by UPnP. The second one is interoperability between different home networking interfaces standards. For example, on basis of their standards, there is no method to transfer a data packet from IEEE 1394 to HomePNA.

In order to solve these problems addressed above, we propose a gateway-based architecture. Figure 4.1 is the illustration of gateway-based architecture. You can see that eHome system uses eHome Residential Gateway and eHome Room Centers to integrate existing home systems and network interfaces. Each of them is responsible for interoperability of services that are used by different home systems and network interfaces with each other, and some special functions. More details will be discussed in the remaining paragraphs.

4.3.2 Web services

Web services are widely used in Internet, especially HTTP. Furthermore, more and more appliances including PCs, the portable devices and the public devices (such as KIOSK), are built in the web browser. Therefore, we believe, HTTP is the best choice for our eHome system to provide the capability of remotely accessing services of the home appliances wherever the users are either inside or outside home.

The emerging technology, WAP, is the famous web service solution for potable device. It is more appropriate than HTTP for devices with limited resources. In other words, end user can access the resources of Internet though WAP with portable devices. Accordingly, we believe that HTTP protocol and WAP are the best approach to provide capability of remotely accessing services provide by home appliances outside or inside home.

4.3.3 Portable devices

More and more people carry the portable device with them or put it nearby, even at home. This shows that the demand and applications of the portable devices are significantly growing in recent years. For this reason, we believe it is very important

to integrate the portable devices into the future home system.

Moreover, there are no existed home systems integrating with the portable devices to support the capability of the remote access. Furthermore, we think that the portable devices will be very famous and important appliances in home networking application in the future. It could be a convenient device to remotely access all services of whole home systems. Besides, it can provide access services remotely inside home, but also outside home when the home appliances are either indirectly or directly connected to Internet. Hence, we propose to integrate the portable devices into eHome system. It plays a core and important role in eHome system.

4.3.4 eHome System Protocol

The home applicant services may be multimedia streaming, voice service, or the control commands only. Therefore, like VSG-based framework, we propose that what kind of the protocols should be chosen as an eHome protocol is depend on demand. Besides, choosing what protocol as an eHome protocol will influence many design considerations. For example, choosing HTTP as our protocol, we need to construct TCP/IP network as our infrastructure and use the URL as our naming approach.

4.3.5 High Flexibility, Scalability and Low Cost

It is very difficult to construct the whole home system by the existing home system standards because no home system meets all the requirements. However, according to the trend of home networking, there will be many home system standards coexistence in the future home. In other words, to construct the home system using the only one standard is unrealistic and inflexible. However, it is an urgent demand to design a home system, which can integrate with all home systems and interfaces.

For this reasons, we propose a general home system which is more flexible than current other home systems. We hope the system with the feature of scalability. That means that eHome system we proposed could be implemented with lowest cost but still keep the original features. Of course, it could also be implemented with higher cost and more functions. For example, you can construct an eHome system with lowest cost and the capability of the controlling home appliances, but this implemented eHome system cannot support the multimedia services. Besides, if you want to support the multimedia services, you should spend more cost to meet this

requirement. Because of the required features described above, we do not specify the implementing approaches and what kind of technologies should be used in this report. Consequently, eHome system become more flexible and scalable, but it also leaves many issues and unsolved problems for the future research.

We utilize the recent technologies to implement a tentative prototyping of eHome system with lowest cost. According to such prototyping, we can evaluate the practicability and observe the possible improvement in the future. The details about implementation are introduced in Chapter 5.

4.4 System Architecture

Figure 4.1 is the overall architecture of eHome system. An eHome system is the gateway-based home system architecture. We can see that each room has only one eHome Room Center (eRC) connected with different multiple local home systems. Besides, whole eHome Room Centers are connected by eHome backbone. An eHome backbone could be any kind of home network interfaces depending on demand. For example, if you need the multimedia services, such as the video streaming or the voice services, can be delivered on eHome backbone. You may choose IEEE 1394 or USB as eHome backbone.

The main responsibility of eRC is to connect local home systems into eHome system. There are three main missions of eRC. The first one is to respectively connect local home network interfaces into eHome backbone. The second is to convert protocols of local home system into eHome protocol. The third is to provide the interoperability of all home systems. In addition, it also provides the capability of remote access for the portable devices. Thereby, the portable devices can access all services provided by every home systems through eRC.

An eHome protocol is the certain protocol running in an eHome backbone. It can provide interoperability of whole existing home systems which are connected with an eHome backbone. An eHome protocol can be any kind of suitable protocol depending on demand for the purpose of the integration. It could also be adopt certain protocol of existing home system such as UPnP and HES as an eHome protocol. For example, you can choose HTTP protocol as the protocol of eHome backbone to provide URL naming-base for all services in eHome system. Or you can also choose a protocol that

can fit the demand for the multimedia data streaming, voice service and asynchronous data transmission. Besides, you can also choose UPnP-based protocol as an eHome protocol, and then each home system that is connected with eHome backbone will interoperate through UPnP-based protocol with each other. Finally, you can use Jini-based devices to access HAVi-based services through an UPnP-based eHome protocol and eHome Room Centers as shown in Figure 4.1.

Actually, we just propose a general architecture for the future home system, but we do not specify what kind of technologies should be adopted in eHome system. Also, we propose system components of eHome system which are dedicated to provide certain functions such as interoperability of different home systems, web-service, remote access capability, protocol translation and so on. We only address and describe the behavior of system component, expected functions, characterizations, interaction between system components, etc., but do not specifying the detail of implementation approaches. However, we will provide some suggestions of implementing approach. For example, we suggest which kind of home system standards or protocols are suitable and can be adopted in current eHome systems.

Consequently, the technologies, which are suitable for our proposed architecture or system components, can be adopted and implemented into eHome system. More details about eHome system components and suggestions of implementation will be discussed later.

4.5 System Components

The eHome system has two main system components: eHome Residential Gateway (RG) and eHome Room Center (eRC). Moreover, eRC consists of eHome Universal Remote Controller (eURC), Service Proxy, and eHome Gateways. Furthermore, eHome Gateway can either be embedded into eRC or be stand-alone devices connected with eRC.

4.5.1 eHome Residential Gateway

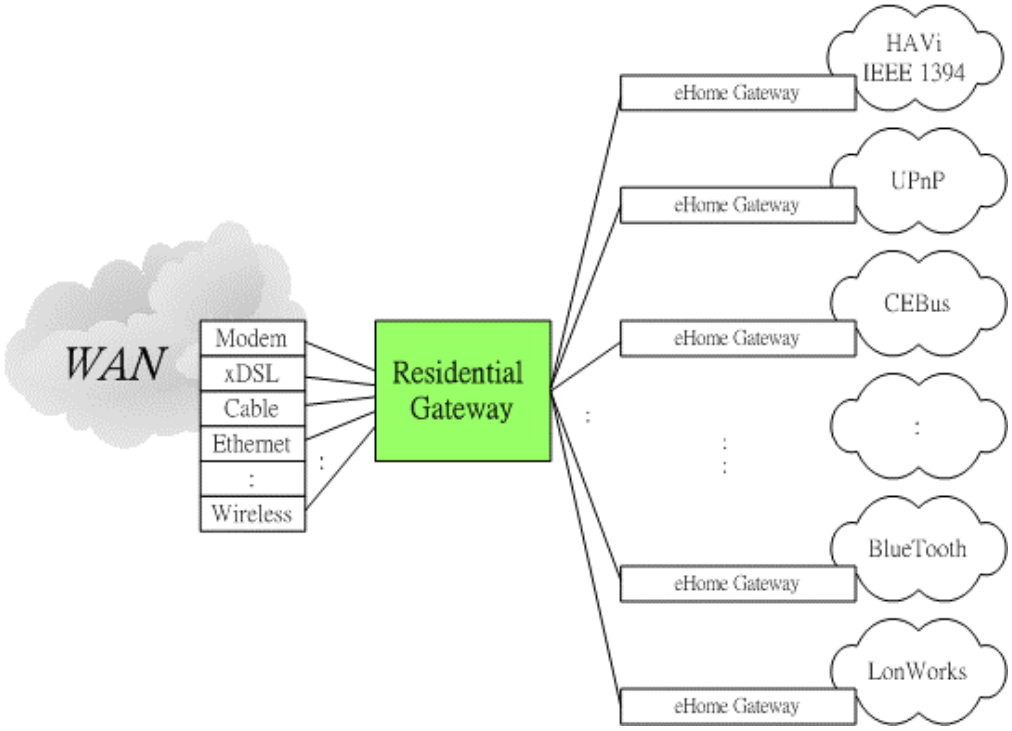


Figure 4.6: Architecture of eHome Residential Gateway

Due to different companies and industries have different explanations. Therefore, it is difficult to define or classify the term of “residential gateway”. In this paper, we define eHome residential gateway (shown in Figure 4.1) as a device, which can provide the capability of Internet access between WAN and the local home networks. Figure 4.6 illustrates the architecture of our proposed eHome residential gateway. The residential gateway will utilize the broadband access technologies such as xDSL, Cable modem, etc. to provide the capability of access resources from WAN. Therefore, the local home networks can access resources of WAN via it. Moreover, people can

remotely access services provided by the local home systems outside home. Besides, it may provide some extra services such as security of remote access, remote monitor service, and so on.

The most difference between eHome and original residential gateway is that eHome residential gateway is connected by multiple home systems and interfaces through eHome gateway as shown in Figure 4.6. So eHome RG provides a uniform interfaces, an eHome backbone and an eHome protocol, for the local home systems. In our consideration, we propose eHome residential gateway has following features and functions.

- Remote access
- Internet sharing
- Web-services: WAP and HTTP
- Firewall
- Multiple network interfaces
- Broadband access, for example xDSL or Cable Modem
- Other special capabilities, such as VOD (video on demand), intelligent agent, etc.

Some home system standards, such as OSGi and HomeGate in HES, are dedicated for the residential gateway. Therefore we can implement eHome residential gateway base on these standards in our eHome system. Besides, we also need to implement specific functions or software depending on eHome system requirements to fit our eHome system. For example, we need to implement and add eHome bundles to OSGi service gateway to integrate with an eHome system. Therefore, all home systems in eHome and OSGi can cooperate with each other. But we will not discuss the implementing details here. In our consideration, the limitation and implementing approaches to eHome system depend on the adopted technologies and the required functions in eHome system. We will introduce our simplest implementation of eHome gateway in Chapter 5. The related researches can be found in [69].

4.5.2 eHome Gateway

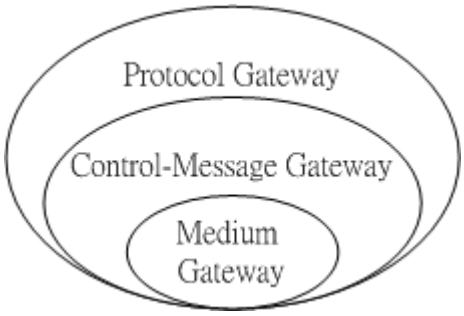


Figure 4.7: eHome Gateway Category

According to the eHome Gateways are classified functionality and characteristics into three categories: Protocol Gateway, Control-Message Gateway, and Medium Gateway. Figure 4.7 illustrates the relationship of eHome gateways. As shown in Figure 4.7, we can see that the Medium gateway is the subset of Control-Message Gateway and the Control-Message is the subset of Protocol Gateway. Therefore, the Protocol Gateway is the most powerful and complex eHome gateway of eHome system in our design.

4.5.2.1 Protocol Gateway

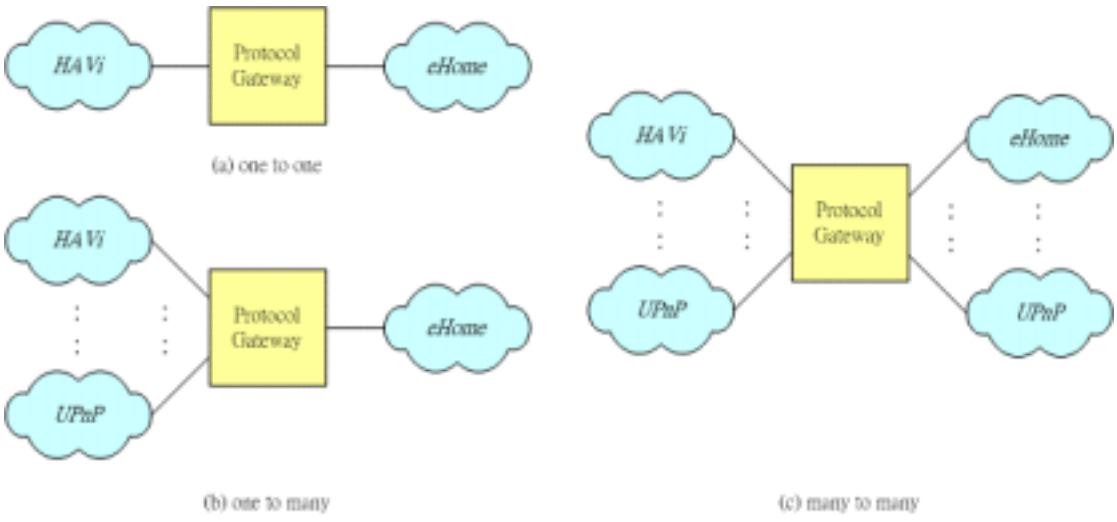


Figure 4.8: Architecture of Protocol Gateway

The term, “protocol”, refers to protocol sets, framework, or middleware of home system in this section. In the following paragraphs, we will use the term “protocol” uniformly to represent middleware, framework and protocol. The home system protocols are more complex than common protocols, such as HTTP, SOAP and so on. The protocols we defined have the following features:

- **Service registry/discovery or lookup service mechanism:** The services of home appliances can be, control operations such as turn on TV, monitor status such as get lamp on/off status, Internet sharing, and so on. Once home appliances connect to a home system, they will use home system protocol to register its services automatically to a home system. Then each home appliance can utilize service discovery mechanism to find out the services provided by others which home appliances, and then home appliances can access the services.
- **Message system:** The home system has the message system dedicated for processing the messages passing among home appliances of the home system. The message can be service registry/discovery, lookup service, Play-and-Play event etc. More complex or complete home systems generally have this system component, for example, HAVi home system.
- **Plug and Play mechanism:** The user could use the services of the home appliance immediately when they plug a home appliance into the home system. It can straightly forward to know such home system needs to cooperate with service registry/discovery and service access mechanisms to complete full functions.
- **Comprise with multiple protocols:** The home system protocol is comprises many different protocols. For example, UPnP standard needs TCP/IP, DNS (Dynamic Name System), HTTP (Hyper-Text Transfer Protocol), UDP (User Datagram Protocol), LDAP (Lightweight Directory Access Protocol), XML (eXtensible Markup Language), XSL (eXtensible Stylesheet Language), ARP (Address Resolution Protocol), etc. to implement UPnP home system.
- **High complexity**

The home system protocol of OSGi, Jini, HAVi, HES and UPnP have these features described above. Therefore the Protocol Gateway is responsible for converting one home system protocol into another. For example, it converts HAVi protocol into Jini protocol. As shown in Figure 4.8, the protocol gateway has three types: “one to one” in Figure 4.8 (a), “one to many” in Figure 4.8 (b), and “many to many” in Figure 4.8 (b). “One to one” type is a gateway which can convert only one protocol into eHome protocol. “One to many” type is a gateway which can convert multiple protocols into

eHome protocol. But “many to many” type is a gateway which can convert each protocol into every type of protocols. For example, it can convert Jini to HAVi, Jini to eHome, HAVi to Jini and so on.

The “many to many” type is a flexible architecture in eHome system. You can utilize this type of gateway to connect multiple home system protocols into eHome system. Besides, once the present eHome protocol needs to be changed into a new home system protocol. Then we just need to add the new protocol into this type of gateway to support the new protocol. On the other hand, “many to many” type is the most flexible gateway and can be adopted and modified quickly to fit the new eHome system.

4.5.2.2 Control-Message Gateway

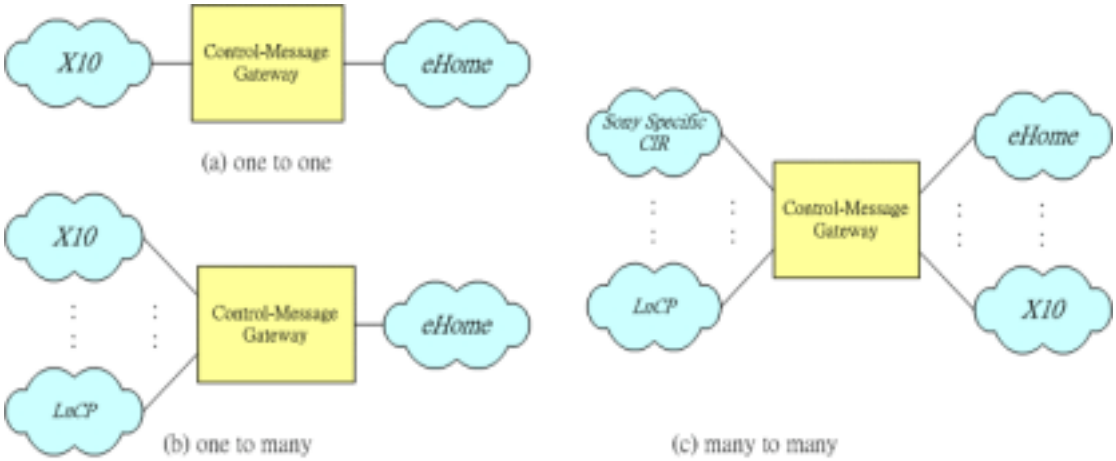


Figure 4.9: Architecture of Control-Message Gateway

Control-Message Gateway is responsible for converting the control-message based protocol into another. The control-message based home system protocol defined in this paper has some of following features.

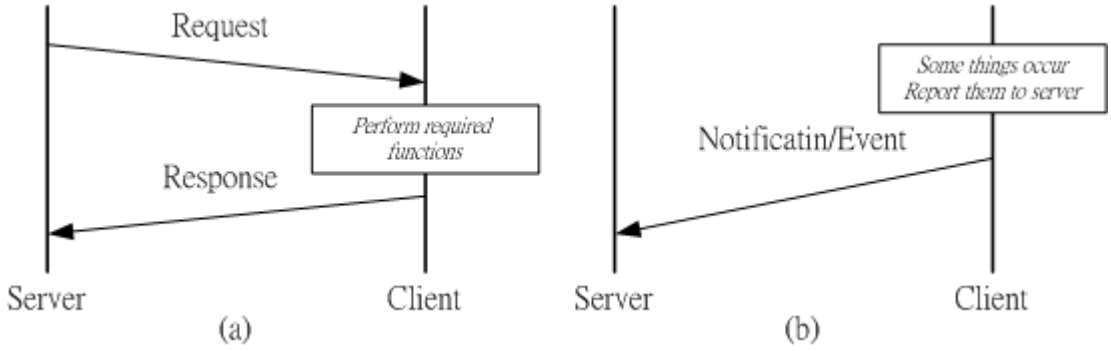


Figure 4.10: Request/Response and Notification/Event Diagram

- **Request/Response based mechanism:** Figure 4.10 (a) is the diagram of request/response base mechanism. The server will send the request message to the client to access the services of the client. The client will perform the required operation and then return the result by response message. The HTTP is an example of protocol which uses the mechanism.
- **Simple notification/event mechanism:** The notification/event is used when something occurs in home appliance. The home appliances (client) will report the message to server (control center) by notification/event mechanism. Figure 4.10 is the diagram of the simplest notification/event mechanism and it is usually adopted in control-message base home system protocol.
- **Static naming/addressing identification:** The control-message base home system protocol usually use the static and unique identify formed by n-bits sequence number to represent the home appliances and home system components. For example, LnCP, the control protocol, proposed by Koon-Seok Lee, uses hex code “0x0”1 to represent “air conditioner” product code in the physical address field of LnCP packet structure.
- **Represent command by unique ID or ASCII text string:** The command is the request to access appliance services such as power on/off, turn lamp on/off and so on. The control-message base protocol usually use unique ID formed by n-bits sequence number or ASCII text string. For example, the command “Turn TV power on” is represented by unique ID-0x14, or ASCII text string-“CMD, TV, Power, ON”.
- **Short length of the message:** The length of control-message base protocol is always short and it is usually smaller than 40 bytes. So it could be process and transmit quickly among home appliances. For example, the maximum, length of X10 protocol for PLC is 14 bits [65].
- **Low footprint:** The control-message based home system protocol can be implemented in micro-controller base devices. The micro-controller base devices are usually with few bytes of RAM and ROM. For example, X10 protocol is such protocol and is implemented usually By PIC (Microchip company’s. micro-controller). Therefore, all home systems protocols whichever can be implement by micro-controller base devices fill the bill.
- **Others:** The simple negotiation mechanisms such as retransmission,

administration, authentication etc. can be supported by control-message based protocol. But this kind of mechanisms is more complex.

4.5.2.3 Medium Gateway

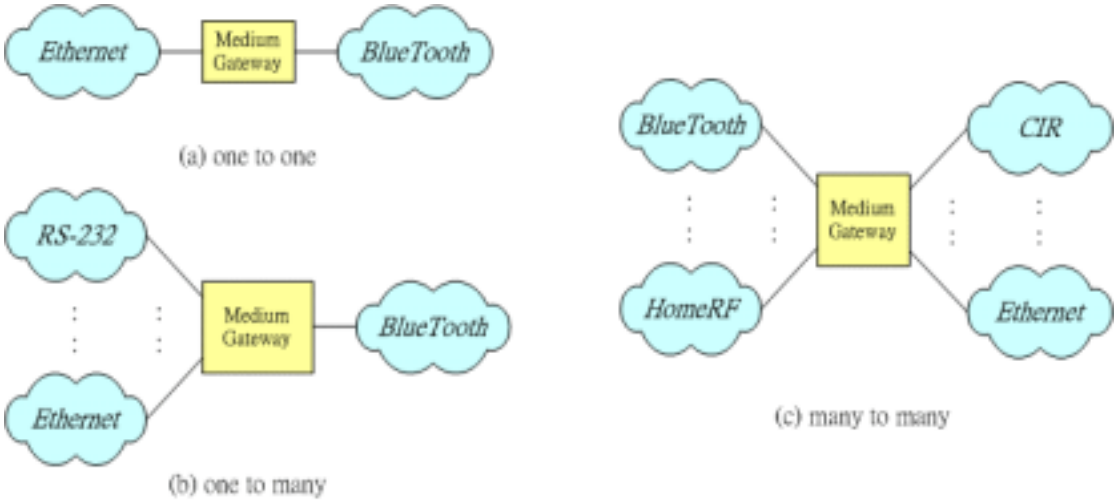


Figure 4.11: Architecture of Meditm Gateway

Medium Gateway is the smallest scale of eHome Gateways. It is responsible for forwarding just only data/payload excluding specific headers of network interface from one home network interface into another one without any modification. For example, it forwards data from Ethernet to WLAN. On the other hand, the Medium Gateway does not have any protocol translation, but only forwards data. Usually, it is either implemented by the hardware only or both the hardware and little software (compare with other type of eHome gateways), even by a SoC chip.

4.5.3 eHome Room Center

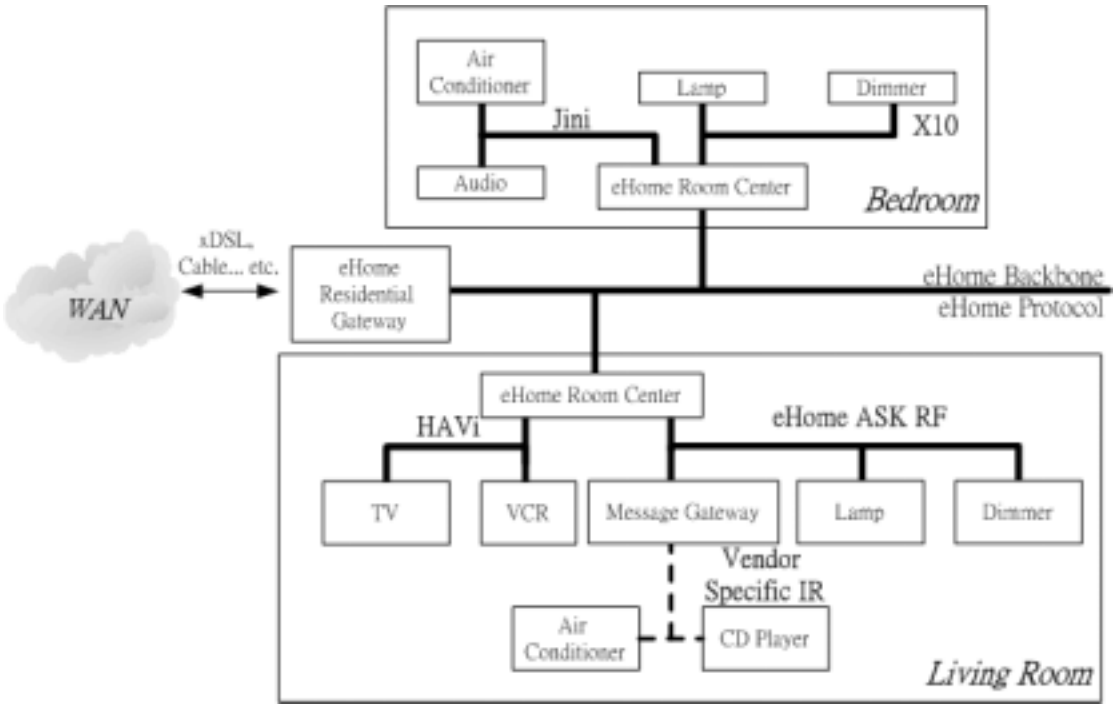


Figure 4.12: eHome System Architecture with eHome Room Center

In our consideration, we think the home system is the distributed system. When the whole home system is disturbed, every home appliance has to be connected together and they also need the capability of the disturbed processing. Moreover, many legacy devices are old home appliances without capability of distributed processing. Besides, the overhead of service discovery/access is heavy and it will increase the load of home system when the whole home system is distributed. Moreover, to construct this kind of system is very expensive and out of economic concerning. We think it is not suitable that the whole home system is distributed. We propose to combine with concept of central control and distributed control into our eHome system. For example, OSGi standard is the role that plays a central node to integrate all services in different home systems. But it is not easy and is out of economic concerning to connect all home systems into only one node. Besides, there are more and more home systems existing in the home. Then there will cause some problems such as heavy overhead of the residential gateway. Moreover, if the residential gateway implemented in OSGi Standard is broken, then devices of one home system cannot share resources with devices of other home systems. Therefore, the whole system is crashed.

Since a house is usually separated into many rooms, we suggest that every room

can be considered as a basic unit or a sub system of eHome system. In our consideration, we think that there is only one eHome Room Center (eRC) in each room. Besides, eRC is responsible for connecting all home appliances in a room to eHome backbone. It plays the role of a central control center to mange all devices in each room. But the home systems connected with eRC are still distributed. We mean that the home systems connected with eRC are still distributed systems. However, when one device of the home systems wants to access a device of another home system, it will need eHome Room Gateway as a medium to forwarding the request from one home system to another. At this time, eRC acts as the central control to home system.

eHome system will connect all eHome Room Centers by eHome backbone. Each eRC is a central control center for each local room and a sub system of eHome system. However, each component, such as eRC or eHome RG, further connects together via eHome backbone. And they can communicate with each other without any mediator. In this case, there is no central server when they connect with eHome backbone. Moreover, every component could be act as either a server or a client.

An example of network topology of eHome system is given in Figure 4.12. HAVi-based device-TV can send a request to control eHome-based device, for example a Dimmer, through local eRC. And it can also send a request to control Jini-based device, such as Air conditioner, which is in the bedroom through collaboration of local eRC in living room and remote eRC in bedroom.

In Figure 4.13, we can see that the eURC Proxy, Service Proxy and eHome Gateways are the main components of eHome Room Center. In our consideration, eRC is responsible for four things: connecting all home appliances in local room, integrating different home systems, managing all services of home appliances of different home system and providing capability of remote access through potable devices. The details of these components, eURC and Service Proxy, are examined in the following sections.

4.5.3.1 Service Proxy

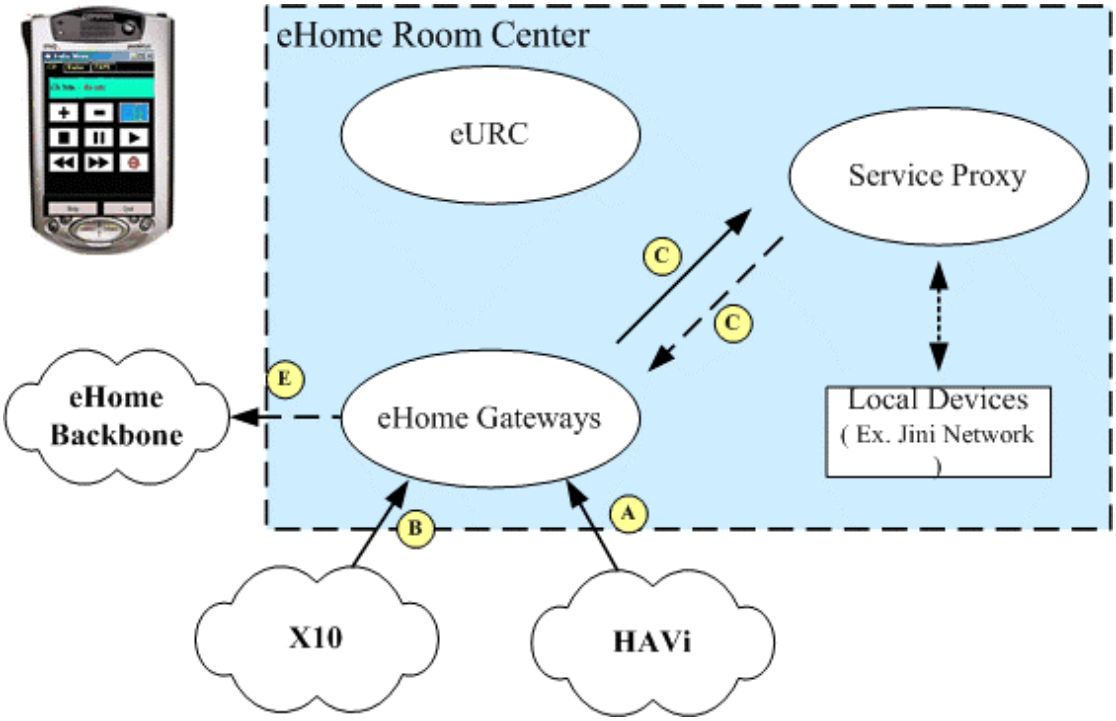


Figure 4.13: Service Proxy – Service Registry

The Service Proxy is responsible for managing all services of eHome system and there is only one Service Proxy in each eHome Room Center. It will keep the latest service list of eHome system. It also provides service registry and discovery/access mechanisms. Therefore, you can discover and access the services via Service Proxy. The registry and discovery/access mechanisms are described in detail as follows.

- **Service Registry:** When the appliances are plugged into the home system, they will send the suitable request by certain protocol to register their services into Service Proxy via eHome gateways. Then the Service Proxy will update its local service database. Therefore, all services of whole appliances connected to eHome Room Center can be discovered and accessed with each other. As shown in Figure 4.13, when the devices are plugged into their home systems, such as X10 or HAVi, they will send the request of service registry to eHome Gateways by path A or B. Then eHome Gateways will forward this request to Service Proxy to register the services and update the service database by path C. Therefore, the devices, which belong to certain home system, can be discovered and accessed with each other. Moreover, after the local Service Proxy updates the local service

database, it will further send the service registry request to remote Service Proxy to inform all remote eHome Room Centers including an eHome Residential Gateway by path D→E.

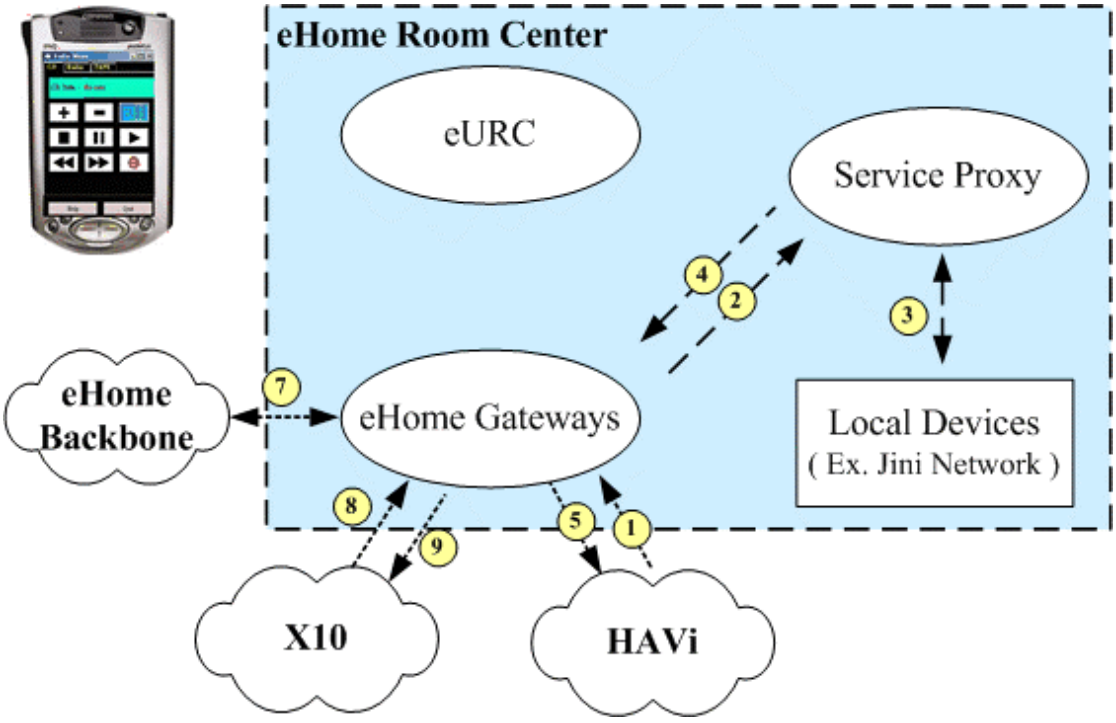


Figure 4.14: Service Proxy – Service Discovery/Access

- **Service Discovery and Access:** The steps of discovering/accessing services are described in Figure 4.14. If HAVi device wants to discover all services provided by eHome system, it will send the request by path 1→2→4→5. Then it will get a latest list of all services. If HAVi-based device tries to access the certain services provided by other home systems, it will send the service discovery request to eHome Gateways by path 1. The eHome Gateway will forward it to Service Proxy to discover requested services by path 2. If the request is provided by the local devices (for example Jini in Figure 4.14). The Service Proxy will send the request to the local devices and access the services directly by path 3. And then it will return the response to HAVi-based device via eHome Gateways by 4→5. Otherwise, if the requited service does not belong to the local devices, it will forward the request to requite eHome Gateways forwarding this request to suitable home system. For instance, if an X10-basd device wants to access the services provided by HAVi-based devices, it will send the request to X10

system. X10 system will forward the request to Service Proxy via eHome Gateway by path 8→2. Service Proxy will check if the service belongs local services. If the service does not belong to local services, it sends the request back to eHome Gateways by path 4. Then, eHome Gateways forward it to suitable home system, for example, a HAVi system by path 5. The result will be returned by path 1→9). Otherwise, if the services belong to the home systems of another rooms, it will send the request to eHome backbone via eHome Gateways by path 4→7. The other eHome Room Center or eHome RG will take care of the request according to the service discovery/access mechanism described above. Finally, the result of the accessing services will be returned by path 7.

4.5.3.2 eURC Proxy

The component-eURC Proxy is dedicated for providing the capability of remote access to the portable devices. You need to implement this component if a system wants to provide the capability of remote access. The implementing approaches are not restricted between the portable device and eURC Proxy. You can implement it by simplest request/response based protocol like HTTP protocol or eHome protocol, even a hybrid system comprising with some appropriate protocol and Java applet. It means that you can implement a hybrid system to meet our proposed mechanism. For example, we can implement a control program with user interface, such as a java applet, for each home appliance in an eHome system. Therefore, the user can download the java applet of the home appliances from eURC through a portable device, then he can control the home appliance through this downloaded java applet.

According description above, we can know there are various and possible implementation approaches. So we only propose the mechanism/functionality of remote access between them here. The steps of mechanism are described following in detail.

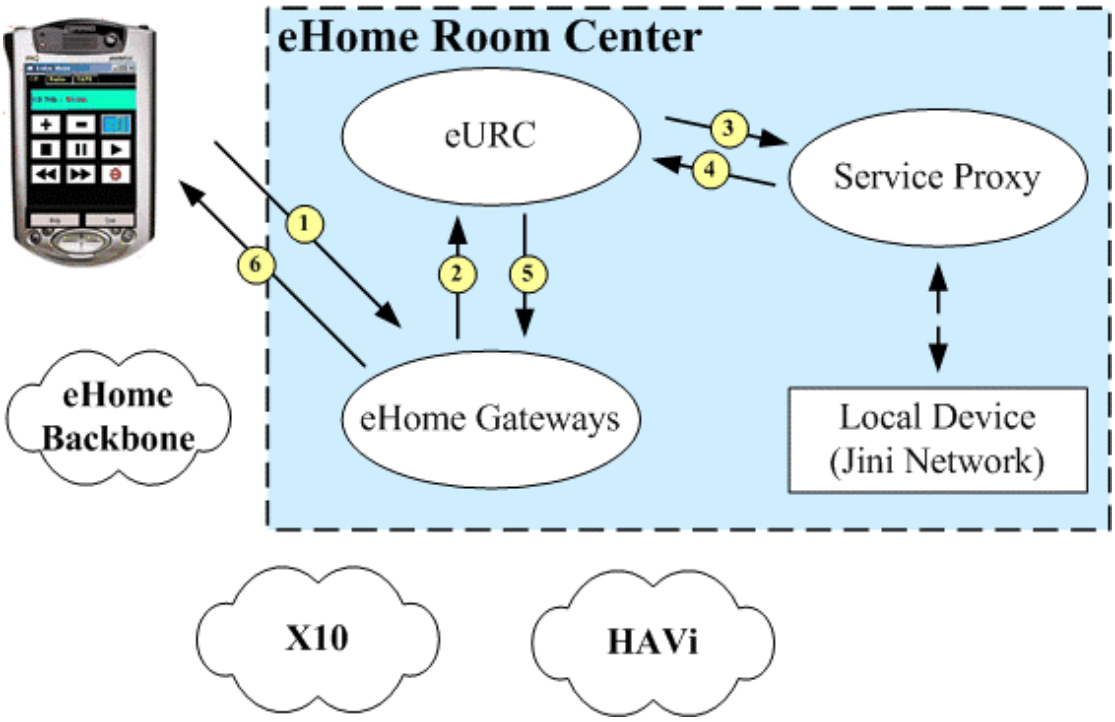


Figure 4.15: eURC Proxy – Service Discovery

- **Service Discovery:** As shown in Figure 4.15, when the portable device need to access services, it will send the request to Service Proxy to discover what kind of services are available through eHome Gateways through eURC by path 1→2→3. Then the Service Proxy will return the result back to eURC by path 4. Therefore, the eURC Proxy will get the newest list of services. Moreover, it will further convert the latest list according to the certain formats, for instance a HTML file or a new user interface profile displaying on the portable device. Then it will send the converted data back to the potable devices through eHome gateway by path 5→6. It is how the portable device gets the latest service list. Hence, the end user can access the services of the home appliances of eHome system via service list. Moreover, if the converted data format is a user interface profile, the user interface will be displayed in the portable device. Therefore, you can access services via friendly user interface. Moreover, the converting mechanism between eURC and the portable device can be implemented by another way. Fox example, it is possible that the converted data is actually a java applet program. It means that you can run the java applet downloaded from eURC to control the home appliances directly on the portable device.

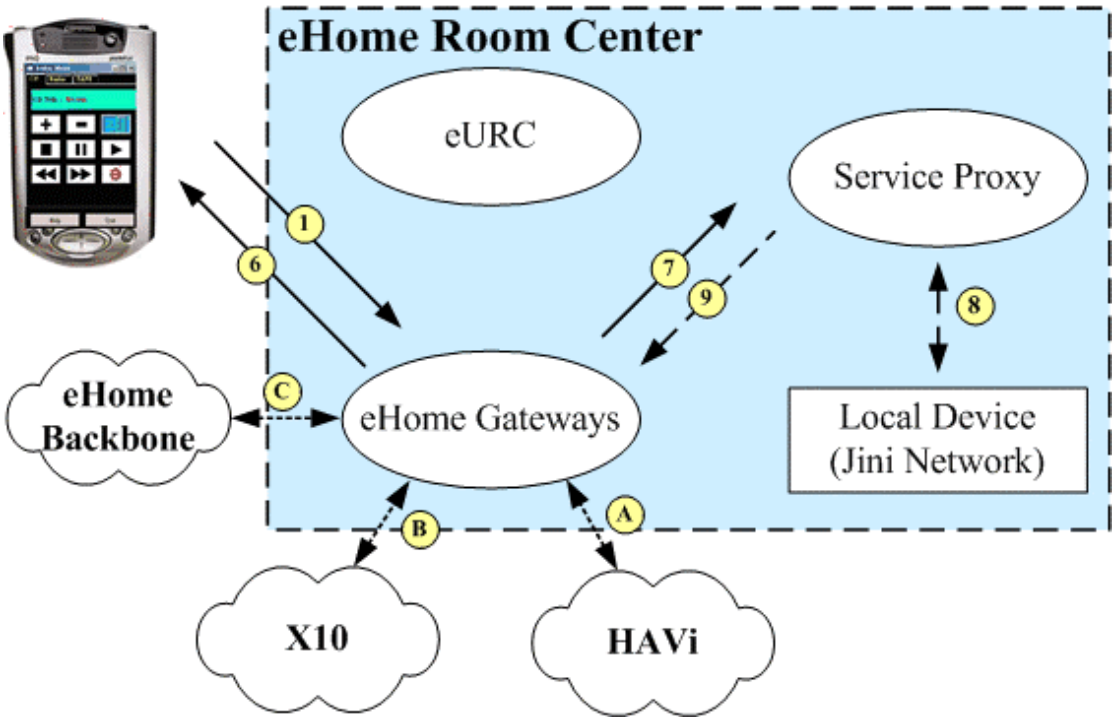


Figure 4.16: eURC Proxy – Service Access

- **Service Access:** The service access mechanism shown in Figure 6 is similar to the interactions between the devices of the home systems and Service Proxy. If the portable devices want to access the services, it will send the request to Service Proxy. If the requested services belong to the local devices, the path is 1→7→8→9→6. Otherwise, it will send the request to appropriate local home systems or other eHome Room Centers following the respective paths 1→7→9→A/B→6 and 1→7→8→9→C→6.

Note: The interactions between Service Proxy and eURC, Service Proxy and eHome Gateways, eURC and eHome Gateways, are communicate by eHome protocol or one common protocol.

4.6 eHome Protocol

An eHome protocol is the protocol used in eHome backbone.That is, eHome Room Centers and residential gateway adopt an eHome protocol to communicate and cooperate with each other. In our proposal, an eHome protocol should be chosen depending on demand. However, the choice of eHome protocol will lead designing some consideration and issues. For example, choosing HTTP as our protocol, we may need to construct TCP/IP network as our network infrastructure. Moreover, we also need the approach with URL-based naming to identify each home appliance of the whole home systems.

Besides, an eHome protocol will influence the capabilities of an eHome system. For instance, when you choose a protocol without capability to support the mechanisms such as service registry, the services of the whole home systems cannot be dynamic registered. Therefore, you need to add services to every home system including eHome system manually when the new home appliances are added into the existed home systems. Hence every device of home systems can access the new services now. However, it is very hard to maintain this kind of system and the system is difficult to use for end users. For this reason, to choose a suitable protocol is very important for an eHome system.

Chapter 5.

Implementation

5.1 Introduction

We intend to construct a prototype of eHome system using current emerging technologies including Embedded System, TCP/IP, GPRS, WAP, Java and BlueTooth. Furthermore, we have designed a control-message based eHome protocol – BHCP and integrated X10 home system into our prototyping. Therefore, we can evaluate the correctness and feasibility of our proposed home system, eHome system.

In our implemented prototype, it contains a eHome Server which plays the role of the eHome Residential Gateway, a X10 home system, several self-designed home appliances with different network interfaces, and eHome Room Centers which are responsible for playing a role of central control center of each room. We will introduce each system component of eHome system in Section 5.2 and the details of implementation of X10 home system in Section 5.3. Finally, Section 5.4 will mention how to integrate all of them into our eHome system prototyping.

5.2 System Components

Our prototype consists of many system components including one eHome Server as eHome Residential Gateway (eHome RG), two eHome Room Centers (eRCs), several self-designed home appliances. The eHome Server provides the capability of remote access between public network and local home networks. The eRCs are responsible for integrating local home networks with eHome backbone and providing service discovery and access mechanisms. Moreover, we classify self-designed home appliance into two types IA according to their network interfaces: one is BlueTooth IA, and the other is RS-23/ASK-RF IA. Furthermore, we have integrated the existing famous home system, X10 home system, into our prototyping. We introduce them respectively as follows.

5.2.1 eHome Server

The eHome Server is a home server playing the part of the eHome RG of eHome system. It provides the capability of public network access. Therefore, the home appliances, which connect to eHome system, can share the resources of WAN through eHome Server. Besides, eHome Server is a HTTP server. End users can access the services provided by the home appliances that connects to eHome system through eHome Server when they are outside home.

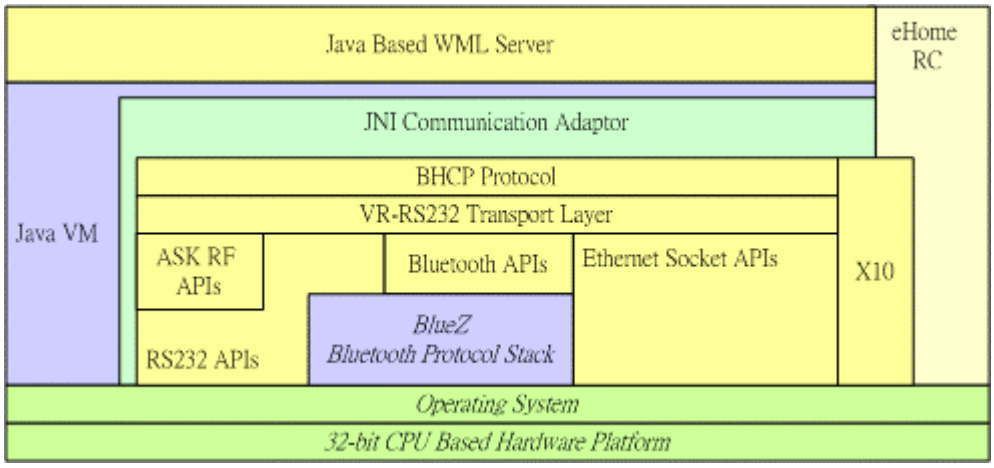


Figure 5.1: eHome Server Architecture

5.2.1.1 Software Components

Figure 5.1 is the architecture of eHome Server. It comprises many software components including device drivers, protocol stack, Java Virtual Machine, Java Based WML Server, and implemented eHome system components. Each software component is introduced respectively following paragraphs:

- **32-bit CPU Based Hardware Platform:** In consideration of the performance, we choose the 32-bit CPU based hardware as our platform. The hardware platform must contain the network interface required by the residential gateway. For example, if the residential gateway uses the ADSL to connect to the WAN through PSTN, the hardware must contain the Ethernet interface. Besides, if the residential gateway adopts IEEE 1394 as its networking interface for local network (eHome Backbone), then IEEE 1394 interface is also consisted in the hardware platform.
- **Operating System:** We consider that the residential gateway mush support Operating System (OS). The OS is a kernel of the system components of the eHome Server. It is responsible for providing a runtime environment for other software components. Moreover, we propose that the OS should contain at least capabilities of running Java Virtual Machine and device drivers of needed networking device.
- **Java VM:** We choose Java to implement our HTTP server with capability of WML. Therefore, we need a Java runtime – Java Virtual Machine, which can be JVM or CVM depending on the requirement of the residential gateway and HTTP server. Beside, we add Java Virtual Machine to our eHome Server considering the feature of the platform independent of Java. It will be very useful for the future implementation, improvement and integration.
- **BlueZ Protocol Stack:** BlueZ [73] is an Open Source project of BlueTooth protocol stack distributed under GNU General Public License (GPL). Qualcomm Incorporated developed it. The first version was released in 2001. Currently, the official BlueTooth protocol stack for Linux with kernel 2.4.x.
- **Communication APIs:** The communications APIs are software components

above the network drivers. We package the network driver as the higher level interfaces considering in the portability, performance issues and ease programming. For example, if the original network driver was not implemented completely enough, and it just provides the read/write and open/close I/O functions. Therefore, we need to code a set API based on the original low-level functions to implement more functions, such as flow control and blocking/non-blocking I/O mechanisms. And we further package those implemented functions in the communication APIs. Therefore, the upper software components can utilize this APIs to transmit and receive the data through the network interface. Besides, some network drivers comprise network drivers and some necessary functions for specific processing. In this case, we can implement it as the communication APIs to meet the requirements. For example, the ASK RF API comprises the RS-232 network driver and some specific functions to reducing the RF interference. So we implement the communication APIs to meet the requirements of ASK RF API as shown in Figure 5.1.

- **VR-RS232 Transport Layer:** This software component is implemented considering the portability. We try to implement a virtual communication transport layer which can be ported to any kind of the network driver or communication APIs. Once your platform replaces the old network interface with a new one, you just need to modify the code of VR-RS232 layer to support the new network driver. The upper software components do not need to be modified any more. Besides, we can also implement the VR-RS232 to support the multiple network interfaces. The VR-RS232 of eHome Server is an implemented example in Figure 5.1

- **BHCP Protocol:** This software component is responsible for parsing and processing the BHCP protocol. Each income/outcome BHCP packet is a handled by this component. The BHCP protocol is the lightweight control protocol for controlling home appliances. More details of BHCP protocol will be discussed later.

- **JNI Communication Adaptor:** JNI Communication Adaptor is implemented as a DLL (Dynamic Linking Library) following the standard of Sun JNI interface. Therefore, the Java programs can utilize the resources

of DLL, which is implemented by native code (non-Java code). In our implementation, Java based WML Server can transmit a BHCP control message to control the home appliances or get the status of the home appliances via the JNI Communication Adaptor.

- **Java Based WML Server:** The WML server is implemented in Java. It is a HTTP server with capability of processing WML content. Professor S.K. Jeng implemented the part of HTTP request processing of the server coded in Java. The current WML server supports the GET and POST method. However, we are responsible for implementing the part of the CGI function of the HTTP server. The CGI function consists of some Java classes (JNI API Package coded in Java) and a JNI Communication Adaptor, a dynamic library (DLL in Win32 version, so library in Linux version) coded in C. Figure 5.2 shows the flow char of HTTP request processing in our WML server. Until now, we are able to control the home appliance via the http request sent from the client, such as mobile phone with WML Browser, whenever user is either inside or outside home. For instance, when end users want to control home appliance outside home. They can connect to eHome Server and sends a HTTP request via WML Browser of mobile phone via GSM/GPRS wireless network. Then the HTTP request will be sent to eHome server and processed by server. The example of request is "http://140.112.28.198:80/?spaceUnit=MediaRoom&appliance=Linux_TV &status=53#Linux_TV". The server will call the related JNI function provided by JNI Communication Adaptor library to process and parse the CGI script. Then server sends the control request to corresponding appliance, TV in this example, through certain communication interface, such as RS-232 or Ethernet. The TV will activate according to the control request and return the processed result back to eHome server. Finally, the server combines the WML contents with the returned result into a completed WML page, and then sends this WML page back to the mobile phone by HTTP response protocol via Internet via GPRS/GSM wireless network.

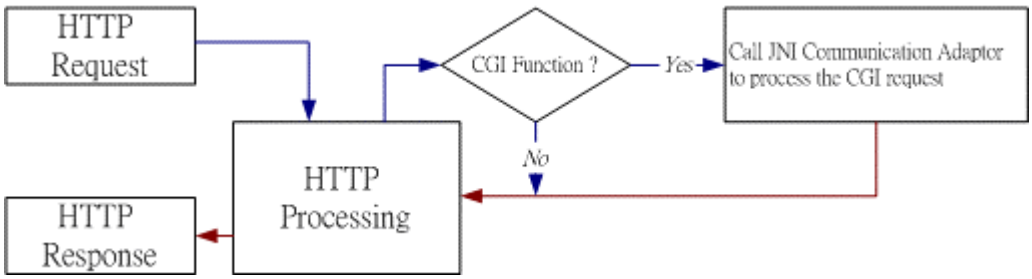


Figure 5.2: Flow chart of HTTP request processing of WML Server

- eHome RC: It is a software component, which is responsible for all functions of eHome Room Center mentioned in Chapter 4. The more details are described in Section 5.2.3.
- X10: It is also a software component. It is responsible for integrating with X10 system. Thereby, we can also access the services provided by X10 system through it. The more detail in mention in Section 5.3.

5.2.2 eHome IA

We have designed two types of eHome IAs ourselves. One is RS-232 IA. It also supports the capability of wireless communication by adding ASK RF module. On the other hand, it could transfer data by wire RS-232 interface or wireless ASK-RF module. ASK RF module is a very cheap RF module and is usually used for remote-controlled devices, such as remote-controlled aircraft, and wireless home appliances.

The other eHome IA is BlueTooth IA. It is a home appliance with BlueTooth wireless interface. We further classify BlueTooth IA into two categories according to their capabilities: one is Simple IA, and the other is Advanced IA. Simple IA is a low-priced IA and is managed by Advanced IA. Besides, the Advanced IA has full capabilities of BlueTooth, but Simple IA does not. In other words, the relationship between Simple IA and Advanced IA is client-server architecture. The simple IA can have full BlueTooth capability through Advanced IA. We will introduce more details later.

5.2.2.1 RS-232 IA

Figure 5.3 shows the hardware function block of RS-232 IA. RS-232 IA is a self-designed home appliance with the hardware and software design. Currently, we

have designed four kinds of appliances: simple lamp, IR_URC, dimmer and SAC (Simple Air Conditioner). Our designed RS-232 IA is 8052-based platform with the ASK RF module for providing wireless communication. The simplest IA, simple lamp, can turn on/off the lamp (AC) and get the status of the lamp (on/off.). Its hardware function block diagram is shown in Figure 5.3 (a).

IR_URC combines with commercial IR URC (Universal Remote Controller) as shown in Figure 5.3 (b). We can convert the BHCP control message received from ASK RF module to IR control signal, which is sent by the commercial IR URC, through IR_URC. Therefore, we can utilize IR_URC to control IR remote-controlled appliances, which are controlled by the commercial IR URC, such as the A/V device and TV.

The dimmer, a home appliance which can sense the current illumination and adjusts the illumination of the light, consists of the Triac Power Control, photosensor, 8052, the ASK RF module and 7 segment display circuit as shown in Figure 5.3 (c). Therefore, you can adjust the illumination of the light and get the current illumination via this dimmer.

The simple air conditioner appliance can sense the temperature and turn on/off cooler/heater according setting temperature. It simulates the behavior of the air conditioner. The hardware function block of SAC is shown is Figure 5.3 (d). We adopt the NTC resister as our temperature sensor and the relay control circuit for AC power on/off switch. Therefore, we can set the expecting temperature to SAC, and SAC will turn the heater/cooler on/off based on setting temperature ranged +/- 3℃. For example, if the sensed temperature is 23℃ but the expecting temperature is 20℃, then the SAC will turn the heater on and the cooler off due to expected temperature is greater than the sensed temperature 3℃. On the other hand, if the sensed temperature is 17℃, the SAC will turn the cooler on and the heater off.

☆ *Hardware Function Block*

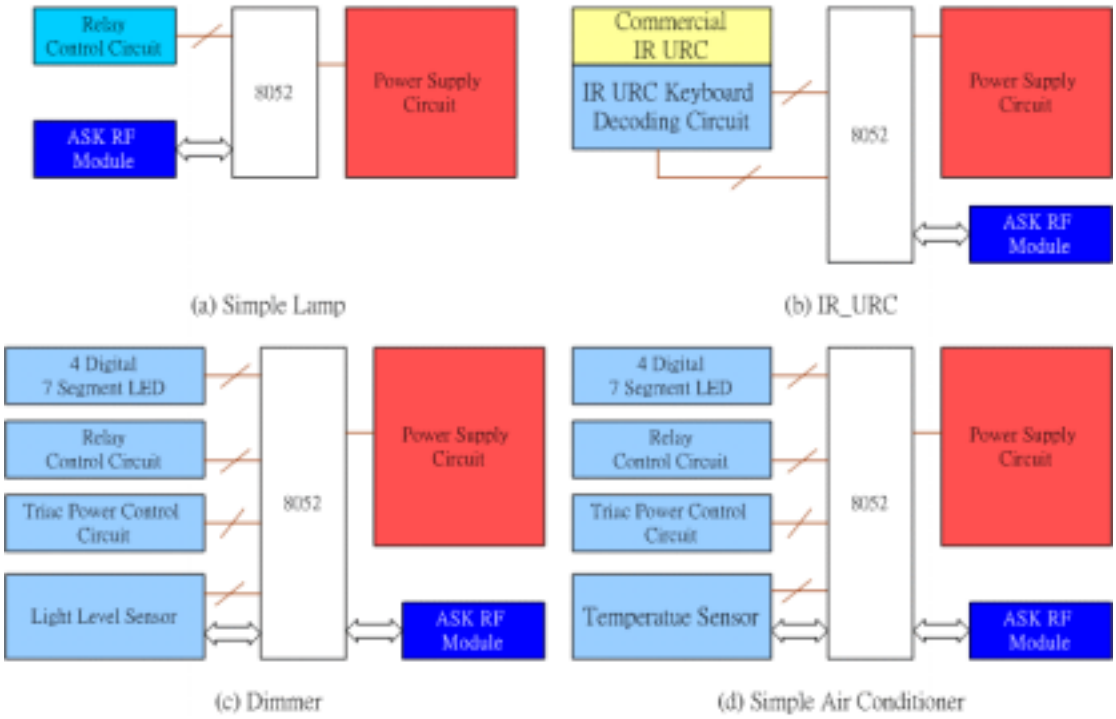


Figure 5.3: RS-232 IA Hardware Function Block

☆ *System Architecture*

Figure 5.4 illustrates the system architecture of RS-232 IA. RS-232 IA comprises many software components. Each of the software components is coded in C and compiled with KEIL C compiler. BHCP protocol, ASK RF APIs and RS-232 driver are the same as the related software components of eHome Server. Besides the native drivers are firmware codes for driving the I/O device, such as LCD, 7-segment display and Triac power control. The entire code size is totally about 4k bytes (binary image size) and is embedded into the Flash ROM of 8052 as firmware.

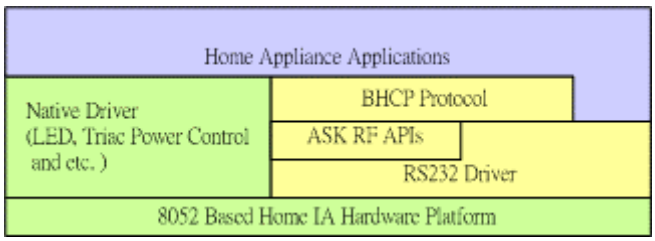


Figure 5.4: RS-232 IA Software Architecture

5.2.2.2 BlueTooth IA

Another type of eHome IA is BlueTooth IA. It is a home appliance with the capability of BlueTooth communication. In this case, BlueTooth IAs can communicate with each other via BlueTooth. We have implemented two kinds of BlueTooth IAs according to cost and functionality. One is Simple IA that is also an 8052-based appliance; another is Advanced IA, which is ARM-based appliance.

BlueTooth Simple IA is an 8052-based appliance. We have implemented the subset of BlueTooth protocol stack. In this case, the implemented protocol stack needs more hardware resources than RS-232 IA. Therefore, we expand the ROM and RAM size of 8052. Currently, both of ROM and RAM size of designed BlueTooth Simple IA are expanded to 64 bytes. Besides, we separate IA's hardware into two individual boards: one is 8052-based motherboard and the other is I/O expansion board considering in flexibility and convenience. Figure 5.5 depicts their hardware function block diagrams. In this case, I/O expansion board is hardware for controlling different appliances. Therefore, if you want to design the new appliance, you only need to redesign the I/O expanding board.

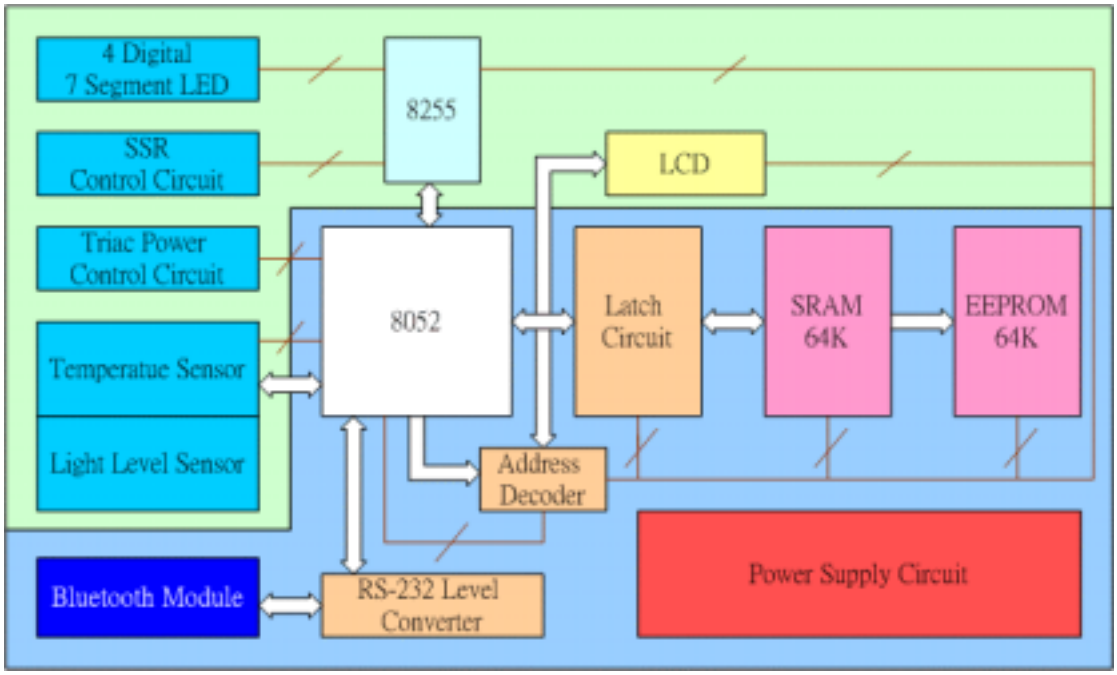
Because 8052 microcontroller provides UART interface and BlueTooth module has RS-232 interface. Therefore, we adopt RS-232 as the communication interface between 8052 and BlueTooth module. Moreover, the voltage level between UART and RS-232 is different. So we need to further convert the logic level signal of UART of 8052 to RS-232 voltage level through RS-232 converter circuit (for example, ICL232 or HIN232 IC). Consequently, 8052 can connect with BlueTooth module via UART via this converting circuit.

Furthermore, we combine three different hardware of distinct appliance into one I/O expanding board as shown in Figure 5.5 (a). We call it "3 in 1 IA". It contains three appliances consisting of Simple Lamp, Simple Air Conditioner and Dimmer. The function of these IAs are the same with RS-232 IAs excluding communication interface (replacing ASK-RF module with BlueTooth).

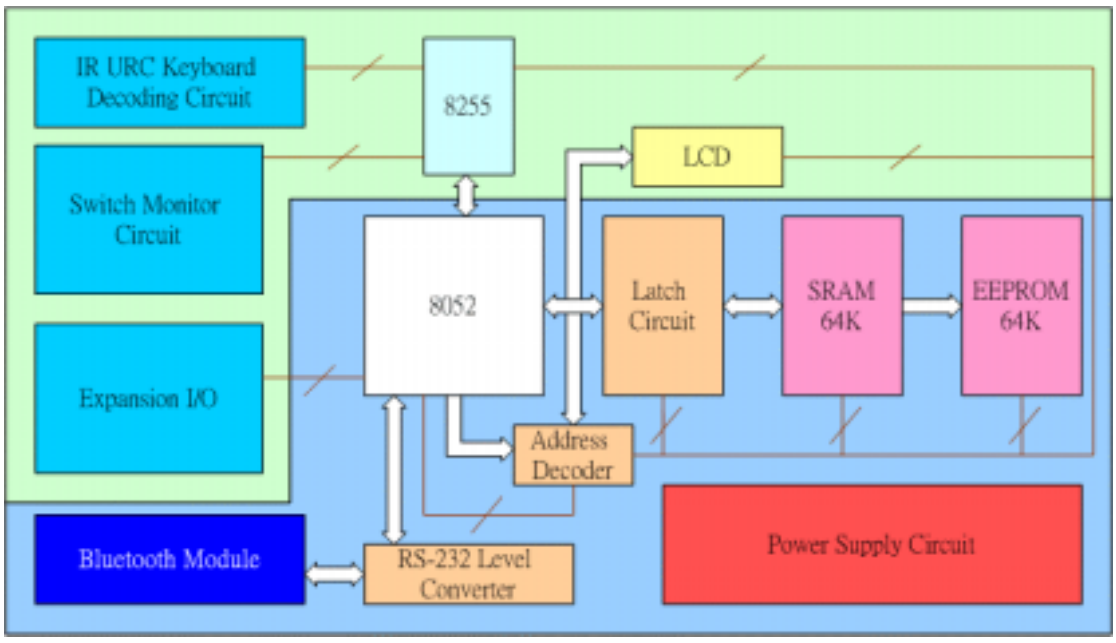
Figure 5.5 (b) is the hardware function of BlueTooth IA - BlueIR. It can covert BHCP control message from BlueTooth interface to IR signal sent by the commercial IR universal remote controller. Besides, it also contains LCD display, so you can display the messages on the LCD panel. We also reserve the expand I/O connector.

Hence, the hardware can be further expanded to implement more functions.

☆ *Simple IA - Hardware Function Block*



(a) BlueTooth Simple IA - 3 in 1 IA



(b) BlueTooth Simple IA – BlueIR IA

Figure 5.5: BlueTooth Simple IA HW Function Block

☆ *Simple IA - System Architecture*

Figure 5.6 depicts the system architecture of BlueTooth Simple IA. We can see that the obvious differences between RS-232 IA and BlueTooth Simple IA are BlueTooth protocol stack and hardware platform. To support BlueTooth communication, we need to implement BlueTooth protocol stack. We have implemented BlueTooth protocol stack, BlueTooth HCI and HCI UART Transport layer to support ACL/SCO data bi-directional communication. BlueTooth Simple IA has only parts of the functions of BlueTooth (ACL data communication), but BlueTooth Advance IA, which is ARM-based platform, has full functions of BlueTooth.

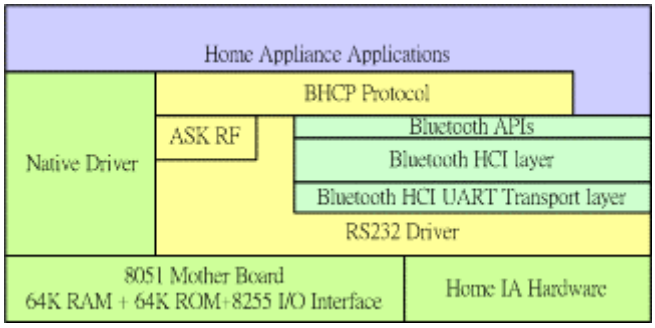


Figure 5.6: BlueTooth IA Software Architecture

☆ *Advanced IA - System Architecture*

As mentioned above, BlueTooth Advanced IA is an ARM-based platform. We have implemented it with iPAQ, a StrongARM-based PDA. Figure 5.7 illustrates the general system architecture of BlueTooth Advanced IA. The software components are almost the same as eHome Server excluding Home IA drivers and QT/Embedded. The function of BHCP Manager is the same as JNI Communication Adaptor in eHome Server. The only difference between them is BHCP Manager is fully coded in C but JNI Communication Adaptor has the functions following the JNI interface standard and one Java class declaring JIN APIs additionally.

Because of using iPAQ Linux as our development platform we adopt a Linux distribution – Familiar Linux for iPAQ platform.[77] It is also an open source project for developing a completed embedded Linux for iPAQ H3xxx series. Besides, we have developed a friendly user control interface based on QT/Embedded [75]. The QT/Embedded is an embedded Linux application development framework based on QT toolkit. It is a GUI development toolkit solution for the cross-platform and coded

in C++. QT and QT/Embedded are both developed by Trolltech Inc.[74] We adopt the open source version 2.3.2 as our current GUI development framework. We have implemented two application prototypes successfully. One is IR Wizard IA, and another is BlueURC. They are mentioned individually following.

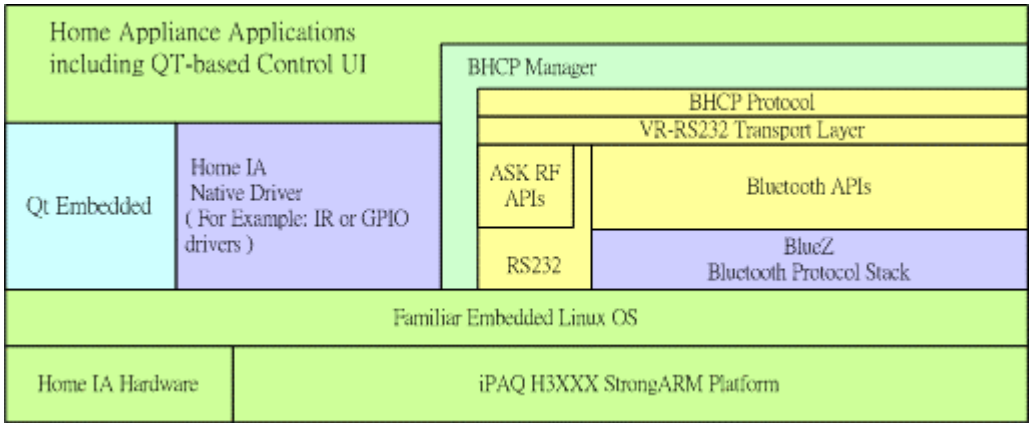


Figure 5.7: BlueTooth Advanced IA General System Architecture

☆ *Advanced IA – Management Architecture*

Advanced IA has more powerful computing power. Therefore, we propose Advanced IA can play the role of manager for Simple IAs. It is responsible for managing Simple IAs by client-server architecture. Because of the limitation of BlueTooth Pico-net, each Advance IA can manage up to seven BlueTooth-based Simple IAs. Nevertheless, Advanced IA can connect and manage the legacy appliances with different network interfaces as many as possible. For example, Advanced IA can connect legacy IA with RS-232 or ASK-RF interface. The limited amount of Simple IAs is depended on the specification of Advanced IA. On other words, the interface between Advanced IA and Simple IA can be either BlueTooth or other network interfaces. Furthermore, Simple IAs can support fully BlueTooth capability through Advanced IA management. Such approach can reduce the construction cost and it is more feasible. The similar approaches can be found in many home system standards, such as HAVi and Jini.

☆ *IR Wizard IA*

A sample implementation of BlueTooth Advanced IA, IR Wizard IA, is a IA which you can use to learn the IR signal of the legacy home appliances with IR remote-control and send the learned IR control signal to control the legacy appliances. Besides, we also need to implement a friendly user interface as the control panel for

those IR remote-controlled appliances. Then end user can control the appliances instinctively without learning. In order to support the functionality described above, we adopt the LIRC [76] and QT Embedded open source projects for our development.

LIRC, a software component of IR-Wizard IA, is an open source project for developing a software total solution including various IR drivers and applications for IR devices. It also supports an embedded version for iPAQ PDA. It utilizes the iPAQ FIR (Fast Infrared) module to emulate the CIR (Consumer Infrared) signal to control the CIR based appliances. Therefore, you can decode and encode IR signal for different IR remote-controlled appliances through LIRC. Furthermore, LIRC supports some IR drivers of the TV cards. We believe that LIRC is the best solution for controlling IR remote-controller appliances. On the other hand, we can use LIRC to learn IR signal of various IR appliances and store the decoding information. And next, we can send the IR control signal to control those appliances through LIRC. Figure 5.8 is the system architecture of IR Wizard IA.

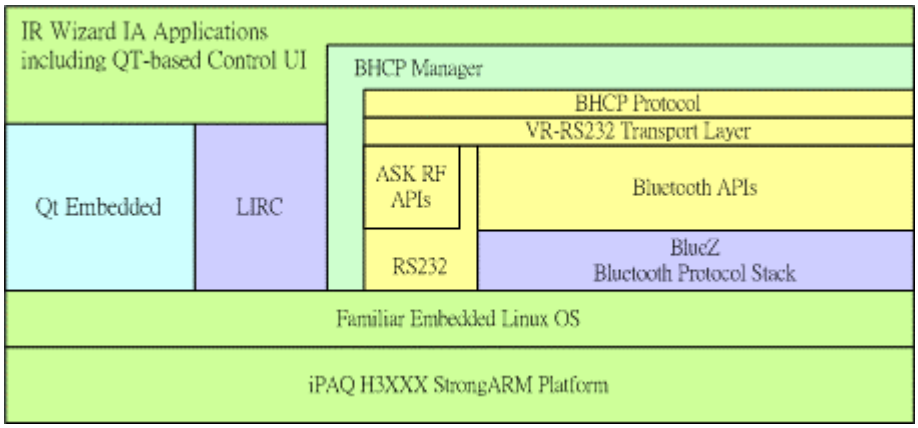


Figure 5.8: IR Wizard IA System Architecture

☆ *BlueURC IA*

BlueURC (BlueTooth Universal Remote Control) IA has the capability of remote access services provided by home appliances. Therefore, we need to add eURC software component into BlueURC IA. eURC is responsible for the interactions between BlueURC and eHome Room Center, and displaying services information by certain format on BlueURC. Currently, we have just implemented the simplest mechanism of the service discovery, static service scrip containing the information of the services of the home appliances. Consequently, you can get the related information of the appliance service from the static service scrip. Next, you can

package them to a complete BHCP control packet and send the packet to the target appliance for accessing the service via BlueTooth. The overall system architecture is shown in Figure 5.9.

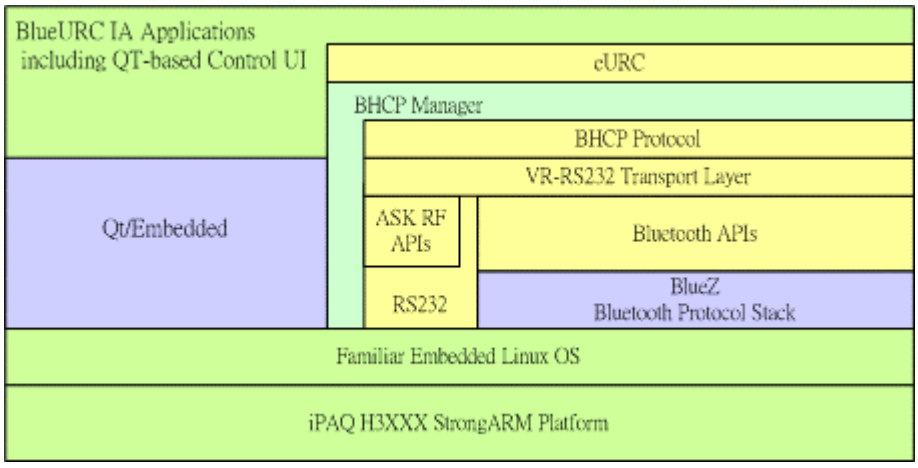


Figure 5.9: BlueURC IA System Architecture

5.2.3 eHome Room Center

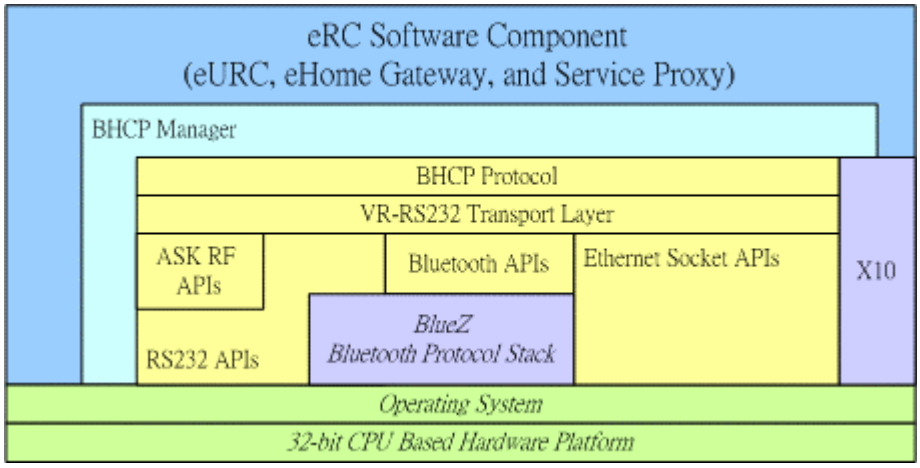


Figure 5.10: eRC Prototyping System Architecture

In our consideration, eRC needs 32-bit CPU based platform and Linux OS the same as eHome RG, but it does not need to support the capability of WAN access, and include any server, which provides remote services, for example, HTTP or FTP services. On the other hand, eRC is the smaller scale than eHome RG. It just needs to support all functions of eRC. Figure 5.10 depicts the overall architecture of eRC, we can see that current eRC prototype supports multiple network interfaces including RS-232, ASK-RF, BlueTooth, Ethernet, and X10 RF module. Furthermore, eRC contains two main system components, eURC and Service Proxy. Therefore, our portable device,

BlueURC, can access all services of home appliances of eHome system via eRC using BlueTooth communication.

5.3 X10 Home Network

X10 is a popular product and made by X10 Ltd. It adopts X10 protocol, a Control-Message based protocol like LnCP, as its control protocol to control every X10-based product. Therefore, if we want to integrate it into our eHome prototyping, we need to implement a Control-Message gateway for X10 home network. In addition, X10 home network and eHome backbone are integrated via eHome Gateway. In this case, eHome Gateway, a sub component of eHome Room Center, is responsible for protocol conversion between the eHome protocol and X10 protocol.

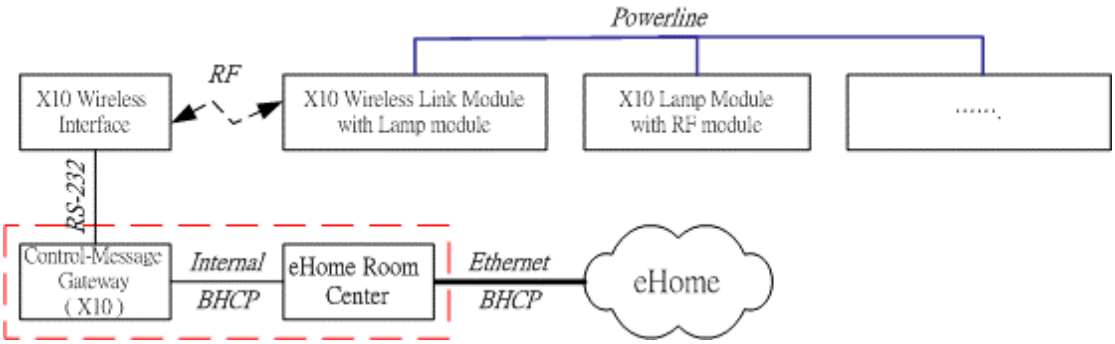


Figure 5.11 X10 Subsystem Integration

We adopt an open source, BottleRocket [78], as X10 control software. BottleRocket is an interface to X10 FireCracker home automation kit.[79] We can utilize BottleRocket to send X10 protocol to control X10 devices through RS-232 port. X10 FireCracker consists a X10 wireless interface for PC, a wireless link module, and Lamp modules. The wireless interface is used to connect with RS-232 port of PC. Also, the wireless link module is connected to power line and it will receive X10 control protocol from RF signal and then process the control request and further forward it to X10 Lamp module via the power line. Therefore, you can send X10 protocol to control X10 devices through these modules from RS-232 port of PC. On the other hand, we are able to control X10 Lamp wirelessly. So we can also control X10 devices through power line and RF communication. Figure 5.11 illustrates the current implementation of X10 subsystem.

5.4 Integration

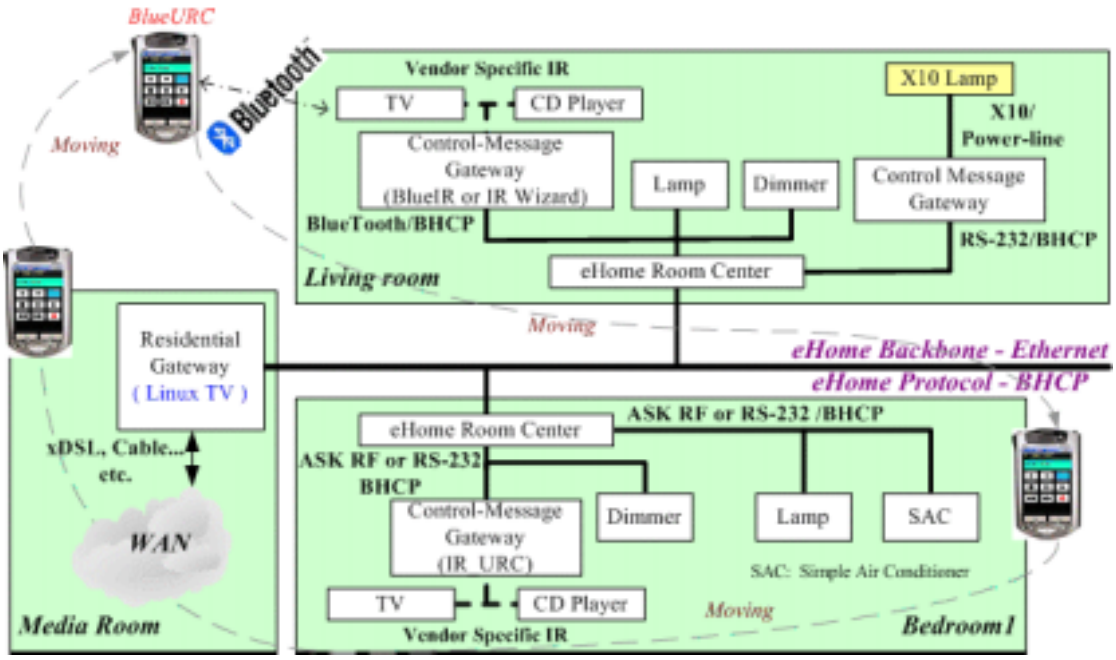


Figure 5.12 Overall Architecture of eHome system prototyping

5.4.1 eHome System Prototyping

We have integrated all subsystems and system components mentioned above to implement the tentative prototyping of eHome system. Figure 5.12 depicts the overall architecture of this prototyping. We can see that the prototyping consists of one eHome RG, two eRCs, and several appliances including self-designed BlueTooth based and ASK-RF/RS-232 based appliances, and legacy appliances. Furthermore, we also integrated an existing home system, X10, and a self-design portable device, BlueURC, into our implemented prototyping. All of them are indirectly or directly connected with eHome backbone, an Ethernet network, and cooperate with a common eHome protocol – BHCP, which is self-designed protocol by us, and it is a type of the control message protocols, such as LnCP [70].

Now, we are able to access all services provided by the appliances of multiple home systems connected by multiple home network interfaces. Moreover, we can control all of appliances via the portable devices anywhere. For instance, we can control Linux TV of “Media Room” either in “Bedroom1” or “Living room” in Figure 5.12 through the current implemented prototyping of the portable device, BlueURC.

But we do not implement the roaming scheme into BlueURC and eRCs. In other words, when BlueURC is out of the coverage of BlueTooth RF signal, we cannot control the appliances anymore. If we want to support this feature, every eRC should have BlueTooth network interface and supports the roaming scheme for solving handoff problem.

Consequently, all home appliances including the portable devices, eHome RG and eRCs are integrated into eHome system successfully through BHCP and Ethernet backbone. Now we can use the mobile phone or PDA built in WAP browser to control the home appliances remotely through Internet and HTTP protocol outside home. Moreover, we can control the home appliances directly via friendly control user interface and BlueTooth inside home without any GSM/GPRS service charge. For instance, we can control X10 Lamp of Bedroom2 from Bedroom1 through the PDA.

5.4.2 Ultimate Mobile Phone Prototype

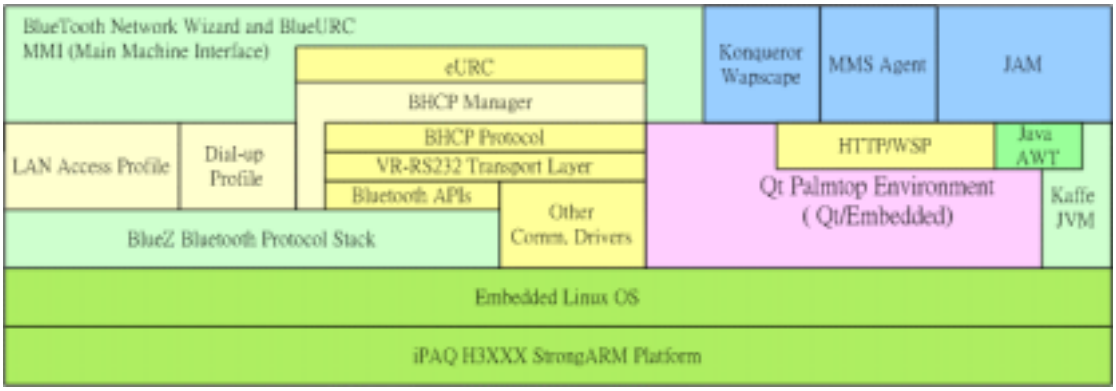


Figure 5.13: Ultimate Mobile Phone Architecture

Ultimate mobile phone, the prototyping of future smart portable device, is an integrated solution of our proposed portable device. It has the functions of BlueURC, JAM (Java Application Manager), WAP and MMS browser.

We cooperated with the members of NTU Lab 305 to develop this prototyping. Figure 5.13 illustrates the architecture. Therefore, we can access the WAP homepage, downloading and running java program and MMS multimedia message through ultimate mobile phone through GSM/GPRS wireless network. Beside, we can utilize the friendly user interface, which is provided by BlueURC, to control the appliances inside home. Moreover we can connect the GSM/GPRS wireless network via BlueTooth Dial-up service to download a certain WAP homepage with WAP format

from WML server of eHome RG to control these appliances through WAP browser outside home. The details of JAM, WAP and MMS browser are mentioned in [80][81][82]. Consequently, people can control all of appliances either inside or outside home by the ultimate mobile phone.

Chapter 6.

Conclusion

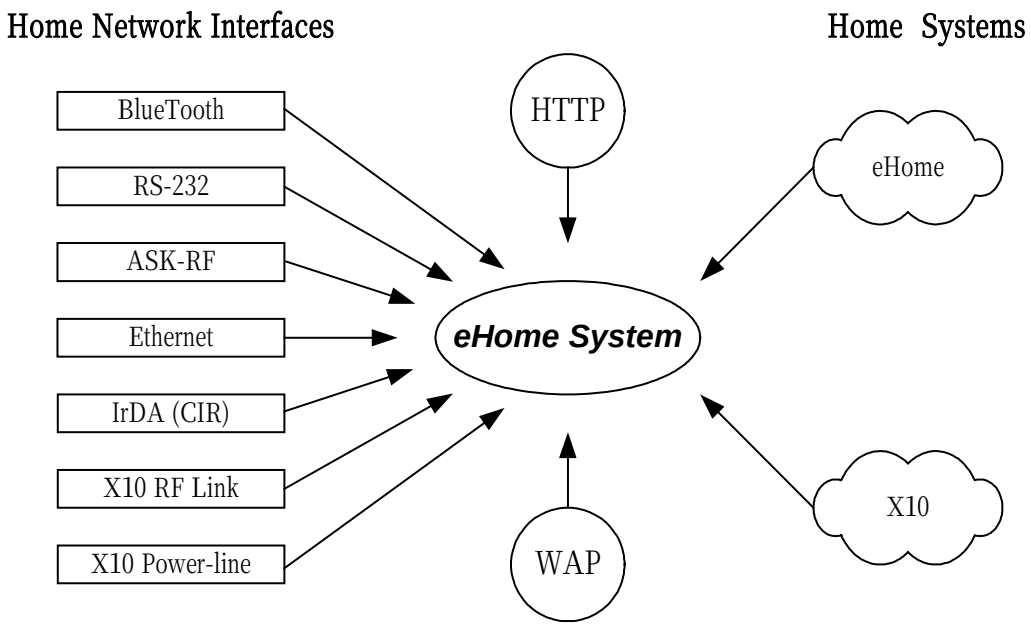


Figure 6.1: Current Integrated eHome System Prototyping

6.1 Summary

We have successfully implemented a tentative prototyping with low cost to verify our proposed eHome system. The prototyping is satisfactory and confirms to the minimal requirement of the future home system. The current prototyping have integrated with multiple technologies and systems including multiple home network interfaces and systems, and other network technologies as shown in Figure 6.1 Besides we have

integrated HTTP and WAP into current eHome system to provide the remote access capability. Hence, we can using WAP-enabled portable device to connect our eHome RG via HTTP protocol to access any service provided by home appliances of eHome system when we outside house. Moreover, we have integrated portable devices to support the remote capability inside home. Now users are able to control the home appliances through BlueURC, the portable device with friendly user interface and BlueTooth, whenever they are at home.

6.2 Future Work

Although we have implemented a tentative prototyping of eHome system with many technologies, but there are still some problems needed to solve. For instance, we just integrate one existed home system - X10 system into the current prototyping, but different home systems will cause different issues and implementation approaches. Perhaps, it is necessary for us to add more software components to eRC.

Moreover, the current prototyping does not support the multimedia service. We plan to support the multimedia service by integrated with HAVi home system. HAVi is a home system dedicated to support multimedia services for AV equipments based on IEEE 1304 network. To integrate HAVi into eHome may cause many unknown issues. Some related researches [84][85] devote to propose the suitable solution for these issues. For example, [84] proposes a gateway-based architecture, which integrates with IEEE 1394 network, a popular home system in Japan, and Internet, to support the capability of remote access of multimedia services.

The current prototyping does not support the multimedia service. Therefore, we intend to integrate a multimedia home system - HAVi with current prototyping to provide the capability of multimedia services in the future. Besides, we also try to implement a more complete and convenient mechanisms for eURC. For instance, we can implement a mechanism to support java program downloading. Thereby, we can download the needed java control program according what kind of services of home appliance we want to access through potable device wherever you are. Moreover users are able to access the service instinctively due to the java control program with friendly user interface. Some related researches are proposed recently. For instance, [82] has proposed a great scheme, the autonomous and generic remote control scheme

and implemented a prototyping, the generic remote controller (GRC), successfully. GRC utilized XML as representation of service control user interface. Such scheme is appropriate to be the future implementation of eURC due to XML is the popular markup language for documents containing structured information. We will strive to overcome the problems described above in the future.

References

[1] Du Guangdong, Doctor, Shenzhen REBO Modern Network Equipment Co., Ltd.,"Digital Home", on-line: http://www.eurochina2002.com/Conference_Presentations/Du_Guangdong.pdf

[2] Kevin W. Spurling, "Home Networking A Comparison of Modern Technologies", on-line: http://faculty.ed.umuc.edu/~meinkej/inss690/spurling.pdf

[3] Kumar Shankar Vellal, Parveen S. Jamooji and Anjan Kumar J.V., "Trends in Consumer Electronics", on-line: http://www.tcs.com/0_whitepapers/htdocs/Trends_in_Consumer_Electronics.pdf

[4] IEC Organization, "Home Networking", on-line: http://www.iec.org/cgi-bin/acrobat.pl?filecode=65

[5] Bill Rose, WJR Consulting, Inc., "Home Networks: A Standards Perspective", IEEE Communication Magazine, vol 39, issue 12, pp. 78-85, December 2001

[6] Sandy Teger and David J. Waks, System Dynamics Inc.,"End-User Perspectives on Home Networking", IEEE Communication Magazine, vol 40, issue 4, pp.114-119, April 2002

[7] IEEE 802.3 Ethernet Standard, on-line: http://grouper.ieee.org/groups/802/3/index.html

[8] Amit Dhir, Saeid Mousavi, "Home Networking Using "No New wires" Phoneline and Powerline Interconnection Technologies", Xilinx corp. WP133

[9] Charles Spurgeon's Ethernet Web Site, on-line: http://www.ethermanage.com/ethernet/

[10] Cygnet Technologies, "White Paper", on-line: http://www.homepna.org/docs/CYG01-WP-101901.pdf

[11] Edward H. Frank, Jack Holloway, "Connecting the home with a phone line network chip set", on-line: http://www.homepna.org/docs/paper500.pdf

[12] HomePNA Alliance, on-line: http://www.homepna.org

[13] HomePlug Alliance, on-line: http://www.homeplug.com

[14] Steve Gardner, Brian Markwalter, and Larry Yongem "HomePlug Standard Brings Networking to the Home", on-line: http://www.commsdesign.com/main/2000/12/0012feat5.htm

[15] Intellon Corporation, "PowerPacket Whitepapers", on-line: http://www.intellon.com/docs/papers/PowerPacketPrimerv10.pdf

[16] Intellon Corporation, "PowerPacket Tutorial", on-lien: http://www.intellon.com/products/powerpacket/documents_open/PowerPacketTutorialMarketingBriefRev20.pdf

[17] HomeCNA Alliance, on-line: http://www.homecna.org

[18] Home Cable Network Alliance, "HomeCAN Overview", on-line: http://www.homecna.org/HomeCNA%20Overview.pdf

[19] IEEE 1394 Trade Association, on-line: http://www.1394ta.org

[20] USB Implementers Forum, on-line: http://www.usb.org

[21] USB v1.1 Specification, USB Implementers Forum, Inc., on-line: http://www.usb.org/developers/data/usbspec.zip

[22] USB v2.0 Specification, USB Implementers Forum, Inc.,

on-line: http://www.usb.org/developers/data/usb_20.zip
[23] “A technical introduction to USB 2.0”, USB Implementers Forum, Inc., on-line: http://www.usb.org/developers/data/usb_20t.pdf
[24] USB related information collection by EEdesign website, on-lien: http://www.eedesign.com.tw/spotlight/usb2/
[25] On-The-Go 1.0 specification, USB Implementers Forum, Inc., on-line: http://www.usb.org/developers/data/otg1_0.pdf
[26] Kosta Koeman, Cypress Semiconductor, “Understanding USB On-The-Go”, Cypress Semiconductor, on-line: http://a330.g.akamai.net/7/330/2540/8f9fc5f56f30ae/www.e-insite.net/ednmag/contents/images/181883.pdf
[27] Electronic Industries Association, on-line: http://www.eia.org/
[28] IEEE 802.11 Working Group, on-lien: http://grouper.ieee.org/groups/802/11/
[29] Wi-Fi Alliance, on-lien: http://www.wi-fi.org/OpenSection/index.asp
[30] The Collection of Wireless LAN, http://www.eedesign.com.tw/spotlight/wlan/
[31] Chris Heegard, Jogh T. Coffey, Srikanth Gumman, Peter A. Murphy, Ron Provencio, Erc J. Rossin, Sid Schrum, and Mattew B., Showmake, Texas Instruments, “High-Performance Wireless Ethernet”, IEEE Communication Magazine, vol 39, issue 11,pp 64-73, November 2001
[32] Bluetooth SIG, Inc., http://www.bluetooth.org
[33] K. V. S. S. S. S. Sairam, University of Madras, Dr. M. G. R. Engineering College, N Gunasekaran, and ohthesAnna University, S. Rama Reddy, Jerusalem College of Engineering, Dr. M. G. R. Engineering College, ”Bluetooth in Wireless Communication”, IEEE Communication Magazine, vol 40, issue 6,pp 90-96, June 2002
[34] Bluetootl v1.1 Specification, on-line : http://www.bluetooth.org/docs/Bluetooth_V11_Core_22Feb01.pdf
[35] Bluetootl v1.1 Profile Specification, on-line: http://www.bluetooth.org/docs/Bluetooth_V11_Profiles_22Feb01.pdf
[36] HomeRF Working Group, http://www.homrf.org
[37] Amit Dhir, Xilinx Corp., “Home Networking Middleware”, on-line: http://www.xilinx.com/publications/whitepapers/wp_pdf/wp136.pdf
[38] Kenneth Wacks, “Home System Standards: Achievements and Challenges”, IEEE Communication Magazine, vol 40, issue 4, pp 152-159, April 2002
[39] Kenneth Wacks, “The success and failures of standardization in home systems”, pp 77-88, Standardization and Innovation in Information Technology, 2001 2nd IEEE Conference , 2001
[40] Konnex Association, http://www.konnex.org/
[41] Open Service Gateway initiative, http://www.osgi.org
[42] OSGi Specification overview 1st, on-line: http://www.osgi.org/resources/docs/specoverview.pdf
[43] OSGi Specification overview 2nd, on-line: http://www.osgi.org/resources/docs/specoverview52001.pdf
[44] Timo Honkanen, “OSGi – Open Service Gateway initialtive”, on-line: http://www.automationit.hut.fi/julkaisut/documents/seminars/sem_s01/honkanen.pdf
[45] OSGi Open Source – OSCAR, http://oscar-osgi.sourceforge.net/
[46] OSGi Open Source – Mister Bundles, http://mbundle.sourceforge.net/
[47] OSGi Open Source – Oxygen, http://oxygen.sourceforge.net/
[48] Sun Java Embedded Server, http://wwws.sun.com/software/embeddedserver/

[49] IBM VisualAge Micro Edition, http://www.ibm.com/software/ad/embedded/
[50] Ericsson OSGi Solution, http://www.ericsson.com/mobilityworld/sub/open/technologies/osgi/index.html
[51] Gatespace's OSGi Development tools, http://www.gatespace.com/userarea/login/
[52] Pavlin Dobrev, Prosyst, David Famolari, and others, "Device and Service Discovery in Home Networks with OSGi", IEEE Comunication Magazine, vol 40, issue 8, pp 86-92, August 2002
[53] Dimitar Valtchev and Ivalio Frankov, Prosyst Sofrware AF, "Service gateway architecture for a smart home", IEEE Comunication Magazine, vol 40, issue 8, pp 126-132, August 2002
[54] Chen-Bang CHEN (陳邦正), " OSGi Based Embedded LINUX Network Service Gateway Implementation and Design", Mater Thesis, National Taiwan University, June 2001
[55] Hung-Cheng SHIH (施宏政), " Design and Implementation of a Residential Gateway for Flexible Service Provisioning in Wireless Home Networks", Mater Thesis, National Taiwan University, June 2002
[56] Home Electronics System Working Group 1, http://sc25wg1.metrolink.com/
[57] Kenneth P. Wacks, "International Home Automation Standards", on-line: http://sc25wg1.metrolink.com/documents/WG1-OVER.PDF
[58] HAVi Organization, http://www.havi.org
[59] HAVi Whit paper, http://www.havi.org/pdf/white.pdf
[60] Kuo-Wei Hsu (徐國偉), "Design and Implementation of a Software Gateway between HAVi and Jini" , Mater Thesis, National Taiwan University, June 2001
[61] Jini Technology, http://www.sun.com/jini
[62] Universal Plug and Play Organization, http://www.upnp.org
[63] UPnP White Paper, on-line: http://www.upnp.org/download/UPNP_UnderstandingUPNP.doc
[64] X10 Ltd., http://www.x10.com
[65] Phil Kingery, "Digital X-10 - Which One Should I Use, Part XIII", on-line: http://www.hometoys.com/htinews/feb99/articles/kingery/kingery13.htm#Digital%20X-10
[66] Eiji Tokinaga, Hiro Ishikawa, Makoto Kurahashi, and others, "A Framework for Connecting Home Computing Middleware", IEEE Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (ISORC'02'02), 2002
[67] Daiki Ueno, Eiji Tokinaga, and others, "Connecting Object-Oriented Middleware for Home Computing with Virtual Overlay Networks", IEEE Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (IDCDSW'02), 2002
[68] Tatsuo Nakajima, Daiki Ueno, and others, "A Virtual Overlay Network for Integrating Home Appliance", IEEE Proceedings of Symposium on Applications and the Internet (SAINT'02), 2002
[69] Takeshi Saito, Ichiro Tomoda, an others, "Home Gateway Architecture and Its Implementation", IEEE Transactions on Consumer Electronics, Vol. 46, No 4, November, 2000
[70] Koon-Seok Lee, Hoan-Jong Choi, and others, "A New Control Protocol For Home Appliance – LnCP", IEEE ISIE, 2001
[71] Sun CDC profile,
[72] Sun JNI Interface Standard,
[73] BlueZ – BlueTooth Protocol Strack, http://bluez.sourceforge.net/

[74] QT GUI Toolkit, http://www.trolltech.com/
[75] QT Embedded, http://www.trolltech.com/products/embedded/
[76] LIRC Open Source, http://www.lirc.org
[77] Familiar Embedded Linux Open Source, http://familiar.handhelds.org
[78] BottleRocket Open Source, http://mlug.missouri.edu/~tymm/
[79] X10 FireCracker Product, http://www.x10.com/automation/firecracker.htm
[80] Wen-Hung HSIAO (蕭文宏), "Development of an Integrated Wireless Network Application Execution Environment", Mater Thesis, National Taiwan University, June 2001
[81] 粘宇村, “The Integrated Environment of MMS and WAP 2.0 Browser on Embedded Linux”, Mater Thesis, National Taiwan University, June 2002
[82] Yu-Chi CHEN(陳禹齊), “Java Application Environment on Embedded Linux Based on WAP”, Mater Thesis, National Taiwan University, June 2002
[83] Y.-C. Yang and F.-T. Cheng, "An Autonomous and Universal Remote Control Scheme," in Proc. of The 28th Annual Conference of IEEE Industrial Electronics Society (IECON’02), Sevilla, Spain, November 2002, pp. 2266-2271
[84] Takeshi Saito and others, “Gateway Technologies for Home Network and Their Implementations”, Distributed Computing Systems Workshop, 2001 International Conference, April 2001, pp.175 -180
[85] Tai-Yeon Ku and others, “A Java Based Home Network Middleware Architecture Supporting IEEE 1394 and TCP/IP”, IEEE Transactions on Consumer Electronics, Vol. 48, No 3, August, 2002, pp. 496-504
[86] ECHONET Consortium, http://www.echonet.gr.jp/english/index.htm
[87] CEBus Industry Council, Inc., http://www.cebus.org/
[88] Echelon Corp. http://www.echelon.com/