使用紅外線傳輸的通用門禁系統

學生：曾律嘉　　指導教授：吳炳飛

國立交通大學電機與控制工程學系碩士班

摘　　要

人們使用傳統鑰匙進入各種場所的方式已經行之有年 ，然而，除了各種已經存在的破解方式對這種系統的威脅之外，功能性的難以擴充以及攜帶上的不便也令其在使用上受到了很大的侷限。為了解決上述的問題，本論文提出了一個稱為 IrGate 系統的解決方案。IrGate 系統為一個類似電子鑰匙的系統，使用電子與網路技術實現門禁控管的功能。也由於結合了電子科技與網路功能，使得本系統得以提供比傳統鑰匙更佳的便利性，並且能夠適用於各種場合，諸如住家、辦公大樓甚至是車子。IrGate 系統是以一個嵌入式系統作為主要架構。其終端裝置使用了內建 ARM 微處理器的 SNDS100 平台、ARM Logic Module 以及 ARM Interface Module 來實現，以作為控制門戶的裝置。而使用紅外線傳輸，一方面提供了本系統一個具有低層次物理隱密特性的無線連結，進而提升了系統的安全性，另一方面也使得系統具有容易使用的特性。最後，一個後端伺服系統以及客戶端手持式裝置也成功地開發出來，以清楚地展示 IrGate 系統的功能。

A Universal Entrance System Using IrDA Transmission

Student：Lu-Chia Tseng Advisor：Bing-Fei Wu

Department of Electrical and Control Engineering

National Chiao Tung University

ABSTRACT

The way people accessing various places by physical keys has been performed for years. However, the safety has always been challenged because the existence of numerous methods of breaking this system. Moreover, its functionality and convenience are limited. In this thesis, a solution, IrGate System, for the above problems was proposed. This system is an E-Key (electronic key) system that uses electronic and networked methods to get permission for entering specific places. Benefiting from the electronic design and the network capability, the system offers a great convenience which makes this new design applicable to many situations such as houses, office buildings and even cars. The system architecture was developed as an embedded system. The terminal device composed of ARM-based SNDS100 platform together with ARM Logic Module and ARM Interface Module plays a role as an electronic lock controlling physical doors. Lots of extra functionalities are brought by the high integration of this device. Besides, the utilizing of infrared data transmission also provides a low level physical security and even a convenient operation manner. Finally, a demo client and server for this system were successfully implemented as well so as to clearly demonstrate its functionality.

Table of Contents

Figure List

Table List

Abstract

The way people accessing various places by physical keys has been performed for thousands of years. However, the safety has always been challenged because the existence of numerous methods of breaking this system. Moreover, its functionality and convenience are limited.

In this thesis, a solution, IrGate System, for the above problems was proposed. This system is an E-Key (electronic key) system that uses electronic and networked methods to get permission for entering specific places. Benefiting from the electronic design and network capability, the system offers a great convenience which makes this new design applicable to many situations such as houses, office buildings and even cars.

The system architecture was developed as an embedded system. The terminal device composed of ARM-based SNDS100 platform together with ARM Logic Module and ARM Interface Module plays a role as an electronic lock controlling physical doors. Lots of extra functionalities are brought by the high integration of this device. Besides, the utilizing of infrared data transmission also provides a low level physical security and even a convenient operation manner. In this paper, a demo client and server for this system were successfully implemented as well so as to clearly demonstrate its functionality.

Terms, Abbreviations and Acronyms

IrGate System	The whole hardware and software system designed in this paper.
IrGate	Abbreviation for “Infrared Gate” – The terminal device of IrGate System as a gateway between IrGate Client and IrGate Server.
IrGate Client	Client device of IrGate System. Currently targeted devices are PDA or Mobile Phone.
IrGate Client AP	The application installed on IrGate Client, provides a graphical user interface for the communication with IrGate.
IrGate Server	Including Web Server, Database, IrGate Web Administration Page and IrGate Web User Page.
IrGate-IrPHY	IrDA Physical Layer hardware designed for IrGate System
IrGate Administration Web Page	The web page for administrators to manage the whole system.
IrGate User Web Page	The web page for users to manage his or her personal data or use the services provided by IrGate System.
SNDS100	An ARM7TDMI processor based development platform with UART and Ethernet interfaces.
Logic Module (LM)	An FPGA prototype board designed by ARM company.
Interface Module (IM)	A multiple hardware interface prototype board designed by ARM company.
PDA	Personal Digital Assistant
IrDA	Infrared Data Association, also usually the abbreviation of IrDA Data Link Protocol.
IrPHY	IrDA Physical Layer
SIR	Serial Infrared. Including data rates from 9600 bps to 115200 bps of IrDA Data Link Protocol.
MIR	Medium Infrared. Including data rates 0.576 Mbps and 1.152 Mbps of IrDA Data Link Protocol.
FIR	Fast Infrared. 4 Mbps data rate of IrDA Data Link Protocol
CRC32	32 bit IEEE 802.x Cyclic Redundancy Check Field

CHAPTER 1 Introduction

1.1. Thesis Architecture

This thesis begins with Chapter 1 which gives an introduction of IrGate System, includes motivation, system outline, function description and architecture. The part of IrGate is divided into 2 Chapters from Chapter 2 to Chapter 3. Chapter 2 describes the design of IrGate physical layer and Chapter 3 demonstrates the architecture and function of IrGate. Chapter 4 describes the design of IrGate Server, which describes its architecture, inner components and functions. And Chapter 5 gives a brief introduction of IrGate Client. Finally, Chapter 6 make a summary of results and gives a description about future development.

1.2. Motivation

In light of the internet environment nowadays, the security of personal information has became a significant issue. Combining with network technologies, payment process and information exchange have become more and more convenient. However, once a more convenient transferring method of information is adopted, a securer mechanism for protecting personal data must be accompanied to prevent stealing or misappropriating of valuable information.

At present time, people have been accustomed to practice payment or authentication by means of various kinds of plastic cards such as credit cards, IC cards or magnetic cards. In fact, most of these cards surely provide a high level security, while some of

them still have several faults. For systems employed by various service providers today, the back-end system indeed possesses a reliable authentication mechanism; nevertheless, valuable personal information may be stolen or eavesdropped at the card reader side. Moreover, the card reader also limits the convenience since the card should be taken out from one's wallet and slipped through or into the card reader to complete the process.

IrGate System is designed for the purpose of solving the above problems. This system does not intend to develop a new authentication system as an improvement of the back-end system, but put emphasis on solving the insecurity and limitation resulting from the hardware currently used instead.

1.3. Innovation

IrGate System is thought to have following innovations:

- **New type of entrance system**: IrGate System utilizes infrared data transmission which enables mobile phones or PDAs to play a role as an E-Key (Electronic Key). The low level security of infrared data transmission makes it hard to be eavesdropped thus highly increases the security.

- **Intelligent design**: IrDA protocol offers a bidirectional interaction capability between IrGate Client and IrGate Server. This feature makes it possible to cross-verify each other and also widen the application scope of the system. Thus, great achievements are made in security, functionality and scalability.

- **Personal universal key**: The authentication function can make IrGate Client the replacement of all kinds of keys in everyday life, such as keys for cars, home doors, apartment houses and office rooms. All traditional keys possessed by a person

could now become many authentication information within a small sized mobile phone.

- **Easy management**: System administrators could issue and cancel authentication directly via internet. Besides, operations of each user could be traced by the log information immediately. These benefits provide an easy way for system administrators to mange the whole system.

1.4. System Outline

IrGate System implements an electronic authentication device with high security, interoperability and integrity. As an embedded system integrating authentication mechanism, remote management and infrared data transmission technology, IrGate System provides an IrDA[1]-compliant handheld device with the ability to get all kinds of authentication through it.

IrGate System has following features:

- IrGate System utilizes the infrared data transmission technology. The nature of short range, narrow angle of the infrared signal presents a low level physical security. Thus the infrared data transmission has relatively better resistance of eavesdropping as compared with other wireless technologies.

- The “Point-And-Shoot” operation of the infrared data transmission provides a time-saving way of data exchanging. This feature makes IrGate System applicative to various kinds of situations such as shopping drinks at a vendor machine, buying tickets on a bus or opening car door with a pushing on the mobile phone button.

- The hardware resource of IrGate System is much more powerful than normal card readers since IrGate System uses an embedded system as its terminal. Therefore, an independent authentication core could be built into IrGate terminal device so as to carry out quick and direct authentication process.

- The mobile technology nowadays has made handheld devices more and more capable of doing lots of computing and storing considerable amount of data. Therefore, the handheld device is now ready for storing large amount of personal data and embedding more powerful security mechanism to protect valuable information. As for these applications, handheld devices are more favorable as compared with plastic cards.

1.5. System Function Description

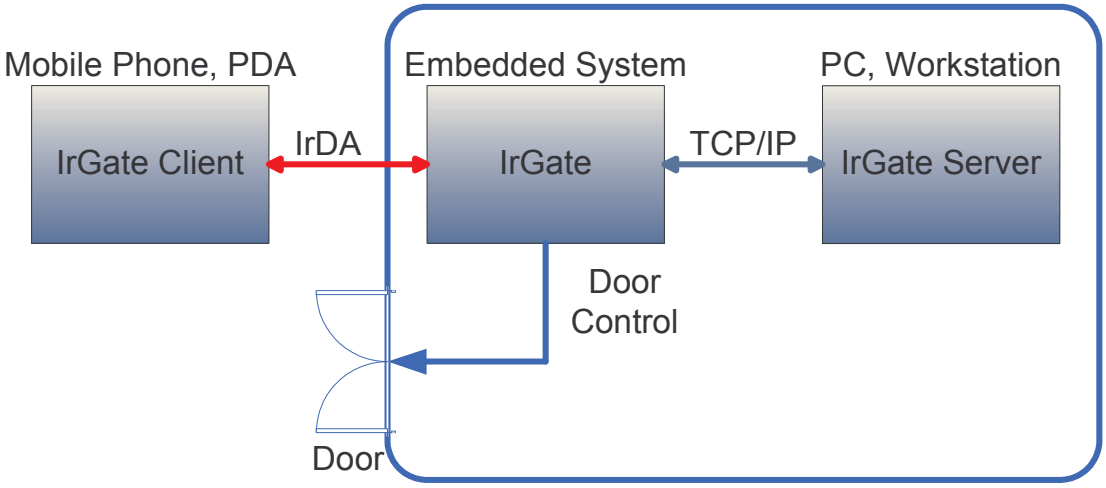

Figure 1. IrGate System function diagram

Figure 1 shows the function diagram of the IrGate System. IrGate System is mainly composed of 3 parts: IrGate Client, IrGate and IrGate Server. IrGate Client, which could be viewed as a key, is a handheld device like PDA or mobile phone which could connect with IrGate via IrDA connection. IrGate, which could be considered as a

keyhole, is a terminal device controlling a door or a gate in the proximity of an entrance. IrGate Server is the back-end system which contains databases, management system and authentication system. The operation of IrGate System is just like that of using a key to open a door, but here IrGate Client, a handheld device, plays the role as a key and IrGate, an embedded system in the proximity of an entrance, plays the role as a keyhole correspondingly. A user could get permission to enter specific places with a handheld device such as PDA or mobile phone by means of short range infrared transmission. Any IrGate device could connect to IrGate Server via internet connection so as to send authentication information, receive mail or update member data, etc. IrGate Message System enables managers to send news, announcements, notes, or just personal messages to members entering specific room or area via IrGate System.

1.6. System Architecture

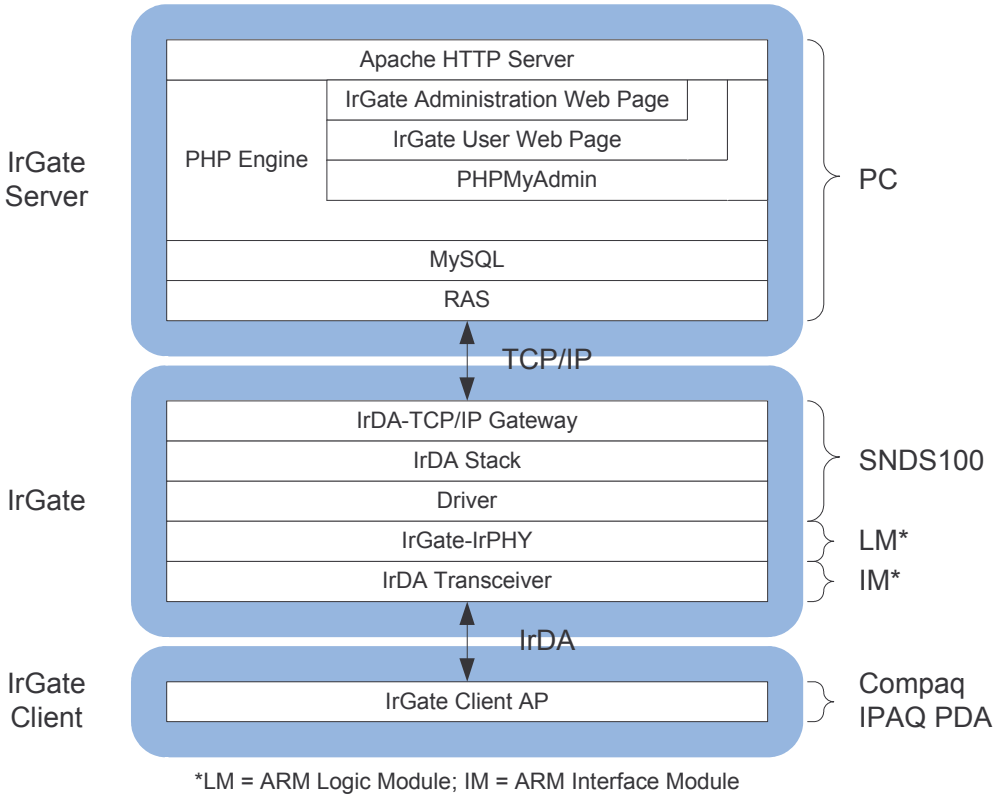

Figure 2. IrGate System Architecture

IrGate System employs many hardware devices, the functions of each hardware device and software component (or circuit design) inside it are briefly described below.

- PC: A personal computer utilizing RedHat[2] Linux 9.0 operation system with MySQL[3] database, Apache[4] HTTP server and PHP[5] server side scripting language.
  - **MySQL**: The MySQL database is used to storing member data and personal messages. This database is known to be a powerful opensource database and easily integrated with PHP.
  - **Apache HTTP Server**: IrGate Server utilizes Apache HTTP server, which is the most popular HTTP server in the world. With this HTTP server, members of IrGate System could get information they need via the web browser and system administrator could easily manage the whole system via internet.
  - **PHP Engine**: PHP is a free and popular server side scripting language. Cooperating with MySQL database on the server, the PHP programs (or web pages) could help to manage all kinds of data. The PHP is capable of accessing database and transforming database content into HTML pages. Therefore one can access the database content via web browsers with the aid of PHP programs.
- **SNDS100**: An embedded system platform with Samsung[6] S3C4510B01 microprocessor (with ARM7TDMI[7] processor core inside). This platform is used to realize an Embedded Linux OS, including Authentication System, Message System and IrDA Protocol Stack. The on-board Ethernet 10Base-T port provides the ability of internet connection with the server. Figure 3 shows the

appearance of the SNDS100 platform.

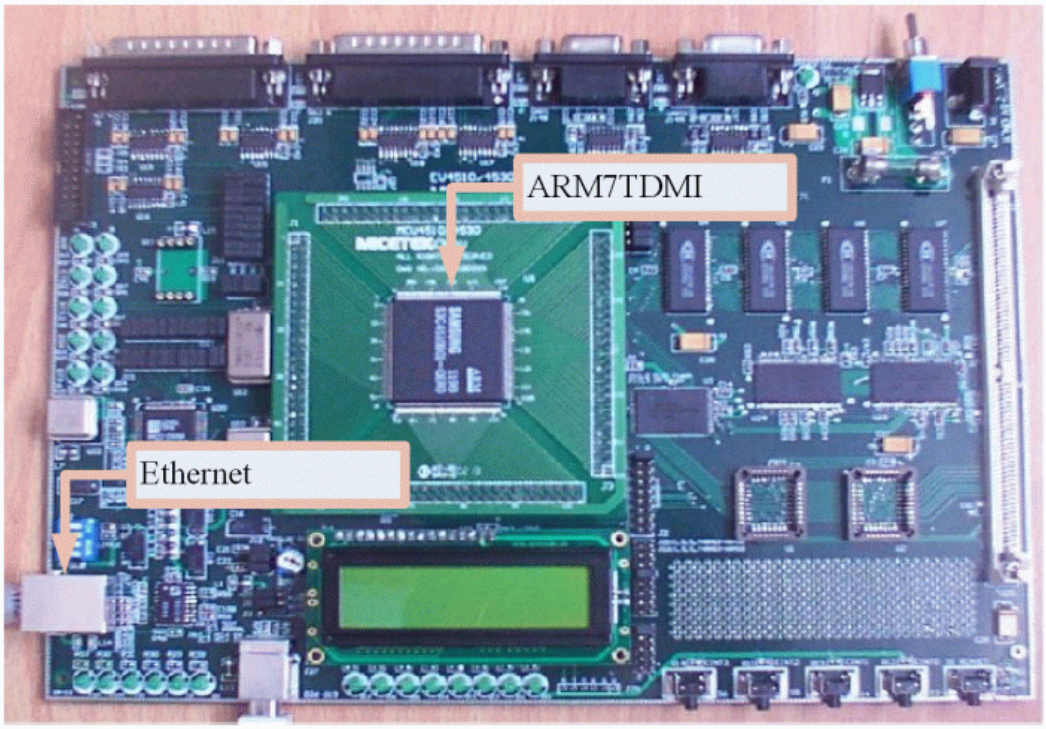

Figure 3. SNDS100 platform

- **IrDA-TCP/IP Gateway**: IrDA-TCP/IP gateway is designed to exchange IrDA packets and TCP/IP packets. When IrGate receives a packet from IrGate client via IrGate driver, IrDA-TCP/IP gateway would transform it into a TCP/IP packet and thus the packet could be sent to the internet using TCP/IP protocol. Oppositely, when IrGate receives a TCP/IP packet from internet, it could pass the packet to IrDA-TCP/IP gateway program and then the packet would become an IrDA packet, which could be sent to any IrDA device using IrDA protocol.

- **Driver**: The driver here is designed to cooperate with IrDA physical layer hardware.

- **ARM Logic Module (LM)**: The ARM[7] Logic Module has a Xilinx[8]

XCV2000E-6FG680 FPGA on it. Thus it is an ideal platform for the implementation of VHDL code for IrPHY circuit. Figure 4 shows the appearance of ARM Logic Module.

Figure 4. ARM Logic Module with Xilinx XCV2000E FPGA

The VHDL code of IrPHY is implemented totally on the ARM Logic Module. The designed IrPHY circuit uses the FPGA I/O pins to communicate with the SNDS100 platform. Besides, the ARM Interface Module plugged onto Logic Module gives the function of IrDA transceiver and also serves as a debug interface as well.

- **ARM Interface Module (IM)**: The ARM Interface Module integrates many useful hardware interfaces such as SD/MMC socket, Smart Card socket, USB port, VGA port, COM port and IrDA transceiver. Another benefit of Interface Module is its ability to be plugged onto the Logic Module. Hence it provides a

perfect physical connection between the Logic Module and IrDA transceiver on the Interface Module. Figure 5 shows the appearance of ARM Interface Module.

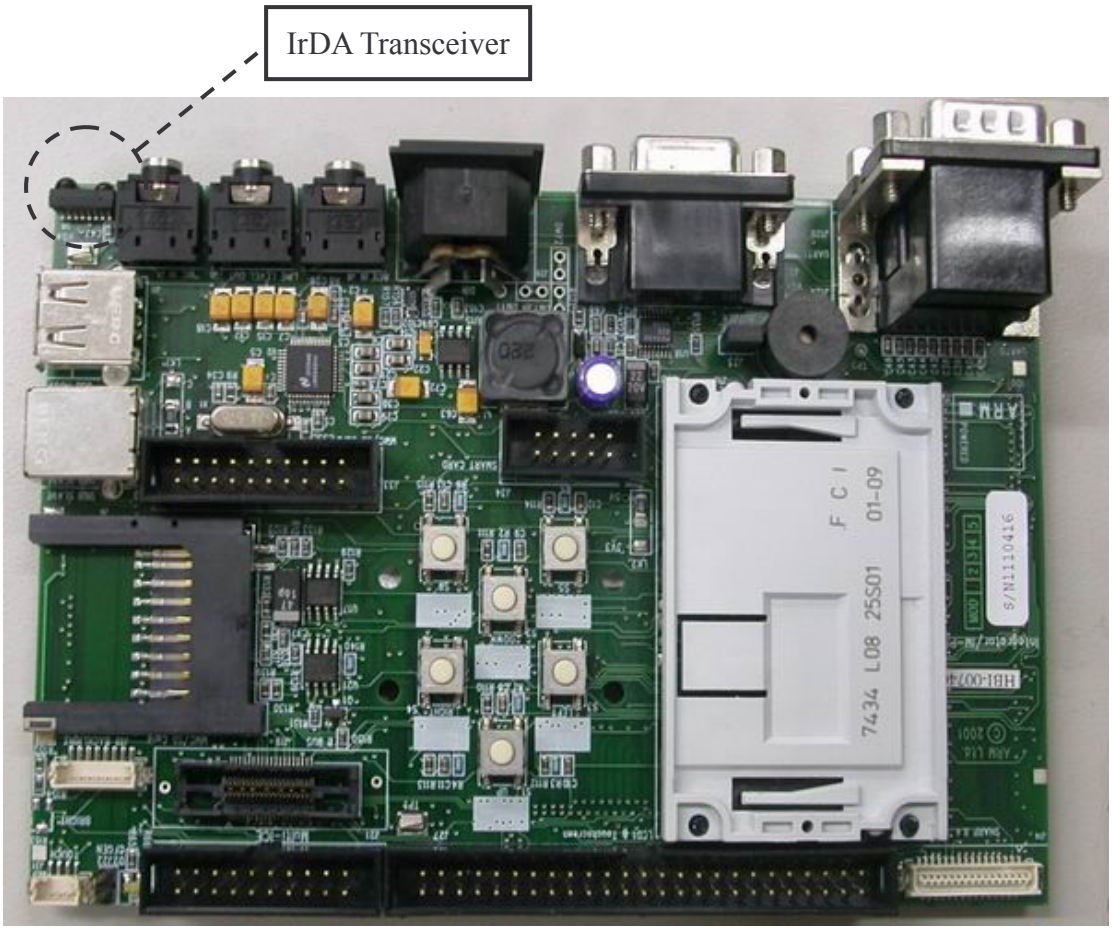

Figure 5. ARM Interface Module

The main role of the ARM Interface Module is to provide its IrDA transceiver's ability to the ARM Logic Module. However, designers may find its various hardware interfaces useful during development process since these interfaces help a lot on debugging and testing.

- **Compaq iPAQ PDA**: The Compaq[9] iPAQ PDA serves as a client platform for users to store personal information and communicate with IrGate Server with its IrDA ability.

- **IrGate Client AP**: IrGate Client AP is an application designed for end users to “talk” with the IrGate System. A valid member could use this program to login onto IrGate System, get authentication or send/receive personal messages.

CHAPTER 2 IrGate-IrPHY

The main difference between IrGate-IrPHY and standard IrPHY is their signaling rates. IrGate-IrPHY doesn't implement all allowable speeds defined in the IrPHY specification. The speed rates that are not implemented include 2.4 kb/s for SIR and 0.576 Mb/s and 1.152 Mb/s for MIR (which means the MIR is totally not implemented). In other words, IrGate-IrPHY is designed to communicated with other IrDA devices with speed ranging from 9.6 kb/s to 115.2 kb/s of SIR and 4 Mb/s of FIR.

2.1. IrGate-IrPHY Hardware

IrDA Physical Layer designed is called IrGate-IrPHY. The ARM Logic Module with Xilinx XCV2000EFG680-6 FPGA is utilized so as to realize the VHDL hardware design. However, an IrDA transceiver module to transmit or receive infrared signals is also needed. For this purpose, an ARM Interface Module is added and plugged onto the ARM Logic Module mentioned above. The ARM Interface Module facilitates many hardware interfaces such as Smart Card slot, SD/MMC slot, USB port, COM port, VGA port, buzzer, IrDA transceiver, push buttons, etc. With this add-on, the ARM Logic Module can easily obtain the ability of transmitting or receiving IrDA infrared signals.

2.2. IrGate-IrPHY Architecture

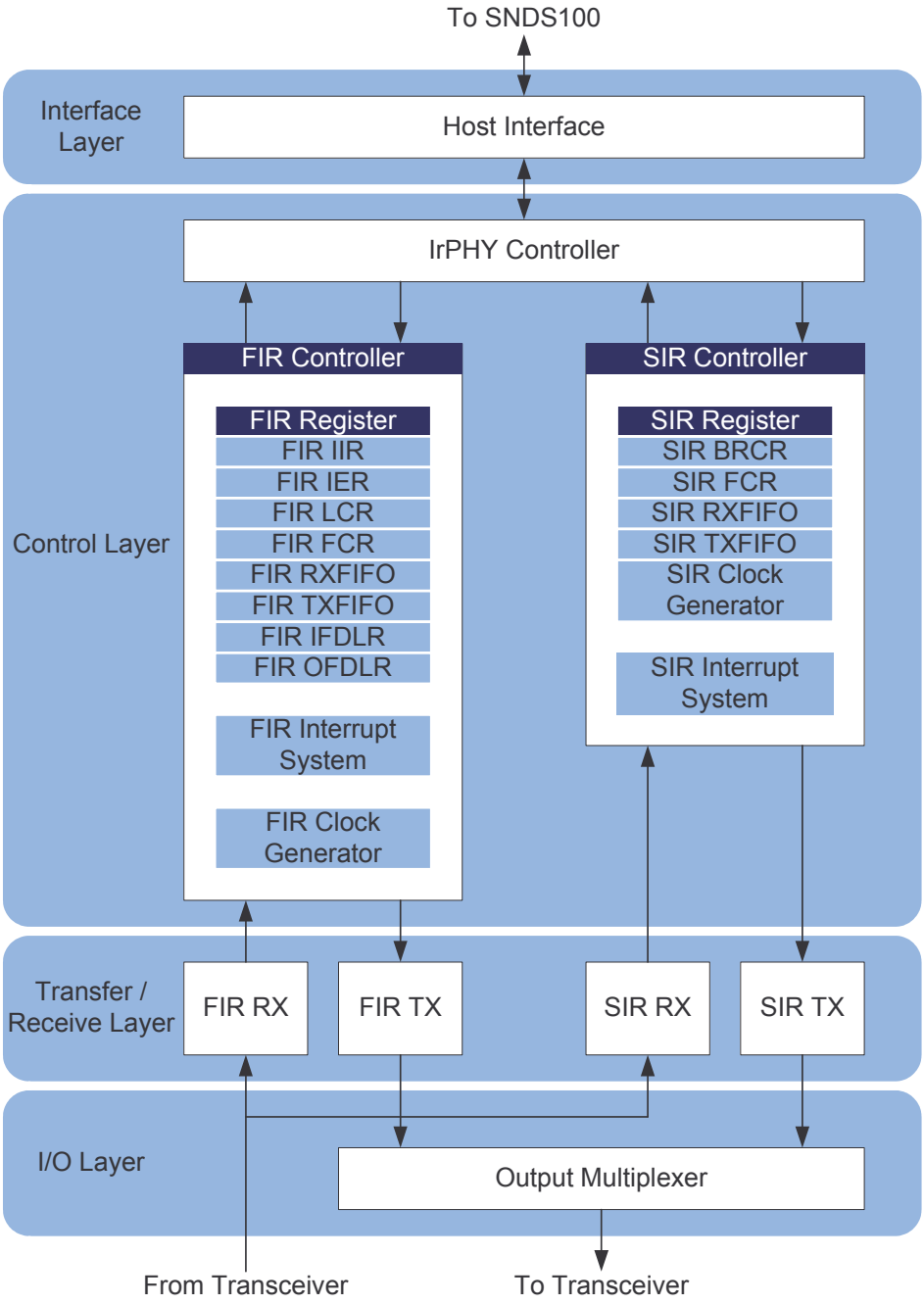

Figure 6. IrGate-IrPHY Block Diagram

Figure 6 shows the block diagram of IrGate-IrPHY. IrGate-IrPHY is mainly composed of 4 layers: Interface Layer, Control Layer, Transfer/Receive Layer and I/O Layer. The descriptions of each layer are listed below:

- Interface Layer: The Interface Layer contains circuits interacting with the SNDS100 platform. The pins connecting with SNDS100 include interrupt signal, DMA handshaking signals, address bus and data bus.

- Control Layer: The Control Layer contains IrPHY Controller, FIR Controller and SIR Controller circuits. These circuits are the main control units of IrGate-IrPHY. They control the data flow, interrupt signal generation, clock enable signal generation and speed rate.

- Transfer / Receive Layer: This layer is responsible for the modulation or the demodulation data frame. Data signals received from the transceiver are transformed into data bytes and then sent to upper layer and data bytes from upper layer are transformed to data signals and sent out via transceiver, each direction uses appropriate method according to different conditions.

- I/O Layer: This layer is responsible for the transmission, reception and routing of data signals. Data signals received from transceiver are directly sent into FIR RX module or SIR RX module while those signals sending out from FIR TX module or SIR TX module are sent to transceiver through multiplexing.

2.3. Clock Domain

IrGate-IrPHY has 4 clock domains. Clock domain 1 is for FIR controller and Interface Layer, and clock domain 2 is for SIR controller. The clock domain 1 uses a 32MHz clock, which is used by FIR controller, FIR RX module and FIR TX module. Also, the Interface Layer belongs to clock domain 1. The clock domain 2, which uses a 115200*16 Hz clock, covers the whole SIR circuit, namely, SIR controller, SIR RX module and SIR TX module. A figure showing different clock domains of

IrGate-IrPHY is given below .

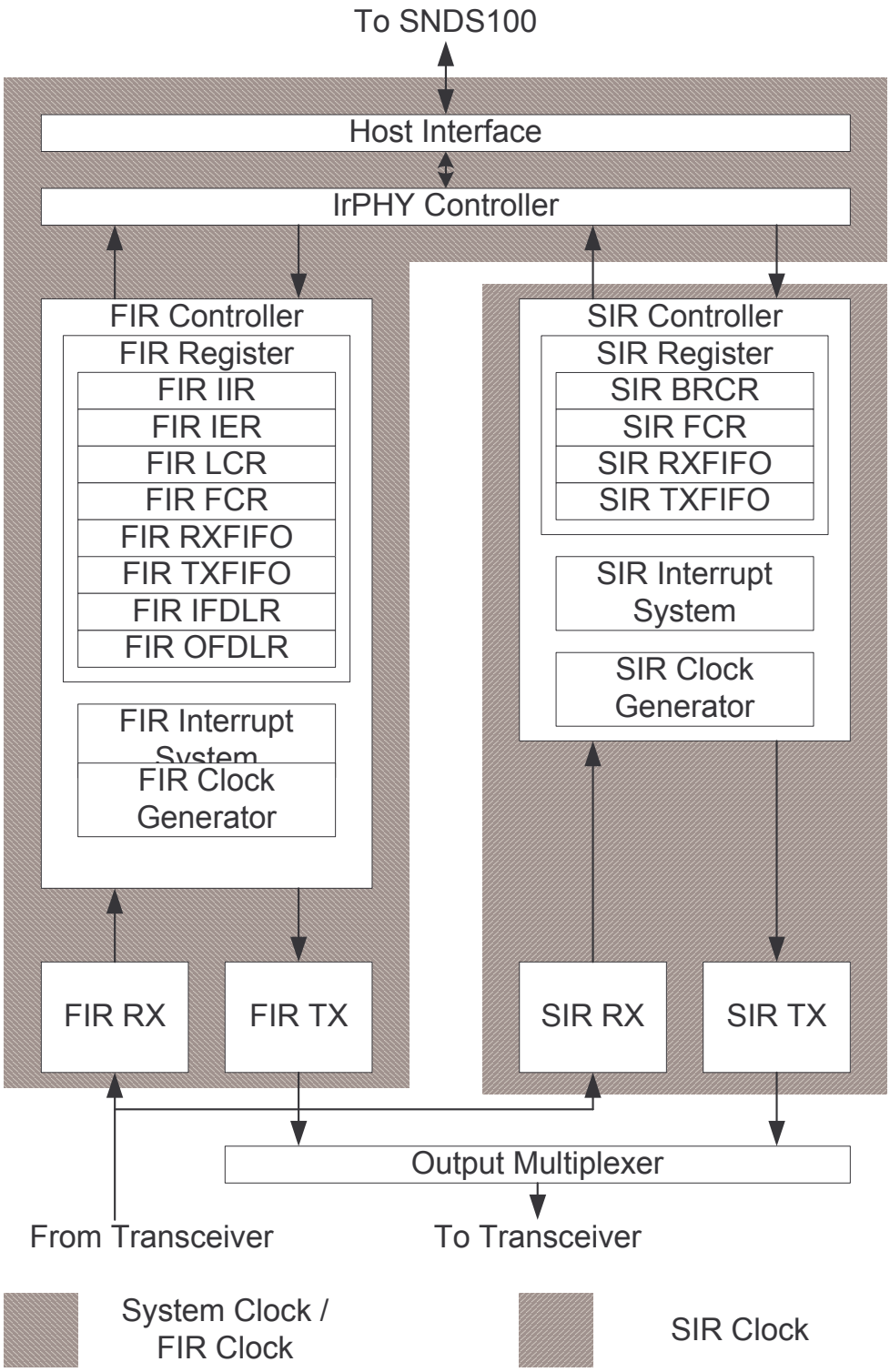

Figure 7. IrGate-IrPHY clock domain

2.4. SIR Circuits

2.4.1. SIR Controller

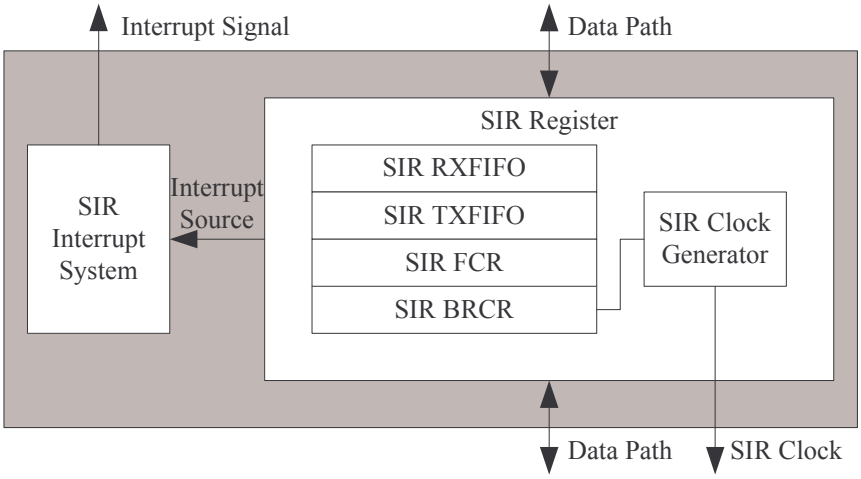

Figure 8. SIR Controller block diagram

SIR Controller is composed of 2 parts:

- SIR Registers:

Consisting of control registers such as SIR BRCR (Baud Rate Control Register) and SIR FCR (FIFO Control Register), SIR TXFIFO (Transmitter FIFO), SIR RXFIFO (Receiver FIFO) and SIR Clock Generator, SIR Registers Module serves as a role which connects the upper layer and the lower layer. SIR Clock Generator is combined into SIR Registers here for the sake of conveniently controlling of SIR BRCR.

- SIR Interrupt System:

The main goal of SIR Interrupt System is to generate interrupt signals according to specific conditions. These interrupt conditions are :

  - SIR TX FIFO empty: SIR Transmitter FIFO is empty.

- SIR RX FIFO full: SIR Receiver FIFO is full.

2.4.2. SIR RX Module

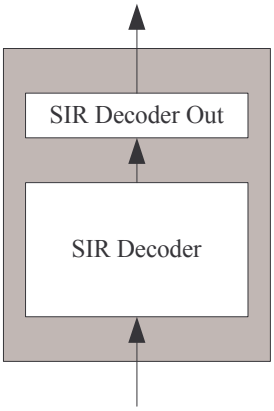

Figure 9. SIR RX module block diagram

SIR RX Module contains SIR Decoder and SIR Decoder Out circuits. The function of SIR RX Module is to transform data from RZI format (3/16 pulse width signal) into data byte, and then put the data into FIFO. The input data signals of SIR RX Module from IrDA transceiver are in RZI format. SIR Decoder first transforms the RZI format signals into the NRZ format signals like those received by normal UART RX pin, which starts with a start bit and ends with a stop bit, contains 8 bits of data in one frame. The second task of SIR Decoder is to decode the NRZ frame signal into a data byte, then send the data byte to SIR Decoder Out circuit. Consequently, SIR Decoder Out circuit takes the data byte as its input and generates proper control signals for SIR RX FIFOs so as to store the data byte just being sent out from SIR Decoder.

2.4.3. SIR TX

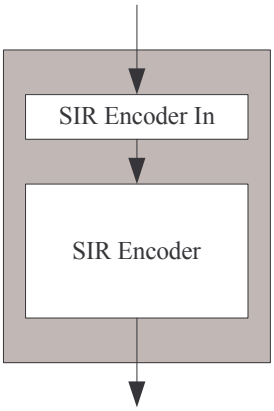

Figure 10. SIR TX block diagram

SIR TX Module is composed of SIR Encoder and SIR Encoder In circuits.

The function of SIR TX Module is to transform the data byte into RZI format signals, which is opposite to SIR RX Module.

SIR Encoder In circuit reads a data byte, which is done by generating proper control signals to SIR TX FIFOs, and then pushes the data byte to SIR Encoder as its input. After that, SIR Encoder circuit transforms the data byte into a normal NRZ frame and then modulates it into RZI format. Finally, the signals in RZI format are sent to IrDA transceiver, which generates infrared pulses according to the RZI format signals.

2.5. FIR Circuits

2.5.1. FIR Controller

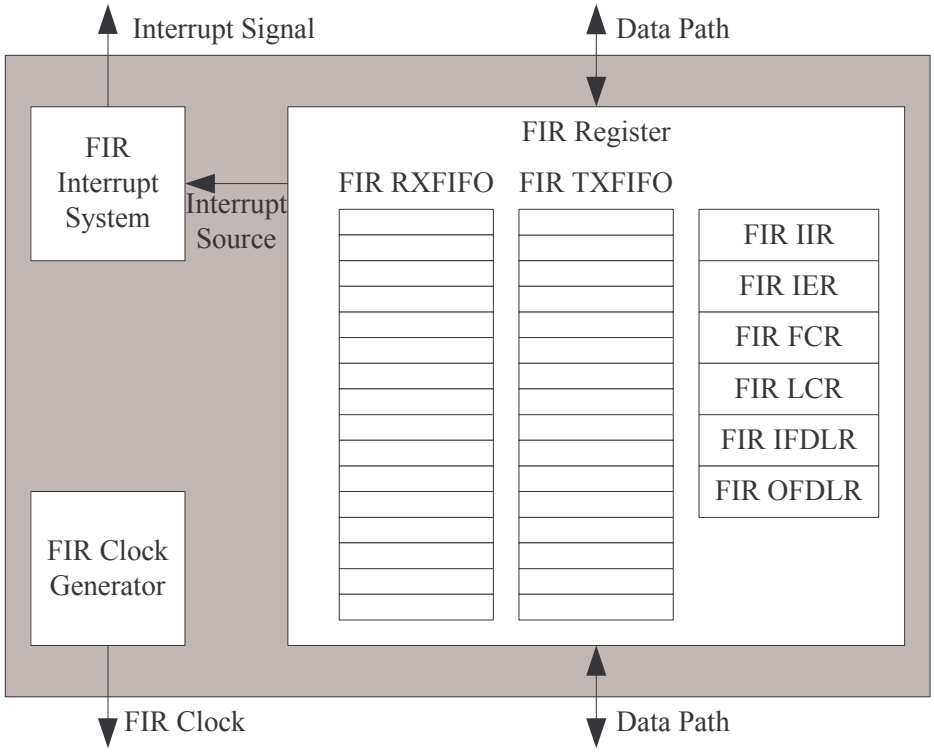

Figure 11. FIR Controller block diagram

FIR Controller, serves as a channel between the upper layer and the lower layer, is composed of three parts : FIR Registers, FIR Interrupt System and FIR Clock Generator.

FIR Registers contain all FIR Control Registers and FIFOs, including Transmitter FIFO (TX FIFO) and Receiver FIFO (RX FIFO) . The function of FIR Registers Module is like a data channel that transfers data from Control Registers and FIFOs.

FIR Interrupt System generates interrupt pulses according to several interrupt conditions, as indicated in the following list.

- Receiver FIFO trigger level reached
- Received End of Frame
- CRC check on input data failed
- Receiver FIFO overrun error occurred
- Receiver Error
- Transmitter FIFO is low
- Transmitter underrun
- Controller Busy

The term “CRC” mentioned above stands for “32-bit IEEE 802.x Cyclic Redundancy Check”. Detailed description of CRC32 is given in the following section.

Finally, the function of FIR Clock Generator is to generate clock enable signals for all FIR components.

2.5.2. CRC32

CRC32 is a 32 bit field that contains a cyclic redundancy check (CRC) value. The CRC is a calculated, payload data dependent field. It consists of the encoded data resulting from the IEEE 802 CRC32 algorithm for cyclic redundancy check as applied to the payload data contained in the packet. The CRC32 polynomial is defined as follows:

$$CRC(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

The CRC32 calculated result for each packet is treated as four data bytes, and each byte is encoded in the same fashion as is payload data. Payload data bytes are input to this calculation in LSB first format.

The 32 bit CRC register is preset to all "1's" prior to calculation of the CRC on the transmit data stream. When data has ended and the CRC is being shifted for transmission at the end of the packet, a "0" should be shifted in so that the CRC register becomes a virtual shift register.

Note: the inverse of the CRC register is what is shifted as defined in the polynomial.

An example of a verilog implementation follows to describe the process.

```
module txcrc32(clrcrc,clk,txdin,nreset,crcndata,txdout,bdcrc);
/*************************************************************/
// compute 802.X CRC
// x32+x26+x23+x22+x16+x12+x11+x10+x8+x7+x5+x4+x2+x+1
// on serial bit stream.
/*************************************************************/
/* bdcrc is input signal used to send a bad crc for test purposes */
/* note ^ is exclusive or function */
input clrcrc,clk,txdin,nreset,crcndata,bdcrc;
output txdout;
reg [31:0] nxtxcrc,txcrc;
/*************************************************************/
// XOR data stream with output of CRC register and create input stream
// if crcndata is low, feed a 0 into input to create virtual shift reg
/*************************************************************/
wire crcshin = (txcrc[31] ^ txdin) & ~crcndata;
/*************************************************************/
// combinatorial logic to implement polynomial
/*************************************************************/
always @ (txcrc or clrcrc or crcshin)
begin
if (clrcrc)
nxtxcrc <= 32'hffffffff;
else
begin
nxtxcrc[31:27] <= txcrc[30:26];
nxtxcrc[26] <= txcrc[25] ^ crcshin; // x26
```

```
nxtxcrc[25:24] <= txcrc[24:23];
nxtxcrc[23] <= txcrc[22] ^ crcshin; // x23
nxtxcrc[22] <= txcrc[21] ^ crcshin; // x22
nxtxcrc[21:17] <= txcrc[20:16];
nxtxcrc[16] <= txcrc[15] ^ crcshin; // x16
nxtxcrc[15:13] <= txcrc[14:12];
nxtxcrc[12] <= txcrc[11] ^ crcshin; // x12
nxtxcrc[11] <= txcrc[10] ^ crcshin; // x11
nxtxcrc[10] <= txcrc[9] ^ crcshin; // x10
nxtxcrc[9] <= txcrc[8];
nxtxcrc[8] <= txcrc[7] ^ crcshin; // x8
nxtxcrc[7] <= txcrc[6] ^ crcshin; // x7
nxtxcrc[6] <= txcrc[5];
nxtxcrc[5] <= txcrc[4] ^ crcshin; // x5
nxtxcrc[4] <= txcrc[3] ^ crcshin; // x4
nxtxcrc[3] <= txcrc[2];
nxtxcrc[2] <= txcrc[1] ^ crcshin; // x2
nxtxcrc[1] <= txcrc[0] ^ crcshin; // x1
nxtxcrc[0] <= crcshin; // +1
end
end
/*************************************************************/
// infer 32 flops for strorage, include async reset asserted low
/*************************************************************/
always @ (posedge clk or negedge nreset)
begin
if (!nreset)
txcrc <= 32'hffffffff;
else
txcrc <= nxtxcrc;    // load D input (nxtxcrc) into flops
end
/*************************************************************/
// normally crc is inverted as it is sent out
// input signal badcrc is asserted to append bad CRC to packet for
// testing, this is an implied mux with control signal crcndata
// if crcndata = 0 , the data is passed by unchanged, if = 1 then
// the crc register is inverted and transmitted.
/*************************************************************/
```

```
wire txdout = (crcndata) ? (~txcrc[31] ^ bdcrc) : txdin; // don't invert
                                                         // if bdcrc is 1
endmodule
/*****************************************************************/
```

To carry out a CRC check on a received packet, just send each bit of the payload data along with its CRC32 frame check sequence into a CRC32 circuit. Once the last bit was shifted in, the correct CRC32 value should be a 32-bit sequence equals to 1100 0111 0000 0100 1101 1101 0111 1011. Otherwise, the CRC check is failed.

2.5.3. FIR RX

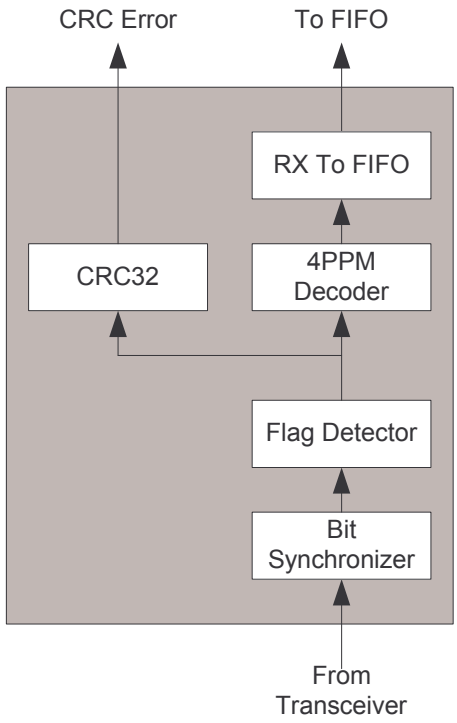

Figure 12. FIR RX block diagram

FIR RX module is responsible of the detection of the PA, STA and STO flags and the transformation of 4PPM format data into normal data byte. The CRC32 frame check sequence data is calculated when receiving payload data so as to check the its validity.

An interrupt condition is held when one of the following events occurs: wrong 4PPM format data received, wrong flag received and CRC32 check failed, which consequently results in the generating of an interrupt pulse.

When FIR RX module demodulates 4PPM data to normal data bytes, it must assure that the received data is a standard 4PPM format data, not PA, STA, STO or anything else. The demodulation starts upon each successful verification of the data. Once the data is demodulated, FIR RX Module finally puts them into FIR RX FIFO.

FIR RX Module is mainly consisted of 5 parts: RX to FIFO circuit, 4PPM Decoder, Flag Detector, Bit Synchronizer and CRC32 circuit.

1. **Bit Synchronizer**: The *Bit Synchronizer* detects the PA flag to achieve bit-synchronization between the original pulse sequence and the system clock, and correspondingly generates synchronized signals for the remaining circuits in *FIR RX*.

2. **Flag Detector**: The *Flag Detector* detects the PA, STA and STO flags and pass the payload data and the 4PPM formatted CRC32 data to the *4PPM Decoder* and *CRC32* circuit.

3. **4PPM Decoder**: The *4PPM Decoder* simply demodulates 4PPM data into normal bit stream and sends it to the *RX to FIFO* circuit.

4. **CRC32 circuit**: The *CRC32* circuit calculates the 32-bit frame check sequence while receiving the data, and generates CRC error signal when the check failed.

5. **RX to FIFO circuit**: The *RX to FIFO* circuit is like a serial to parallel circuit that transforms serial bit stream sent out from *4PPM Decoder* into 8-bit data byte and puts it to FIR *RXFIFO*.

2.5.4. FIR TX

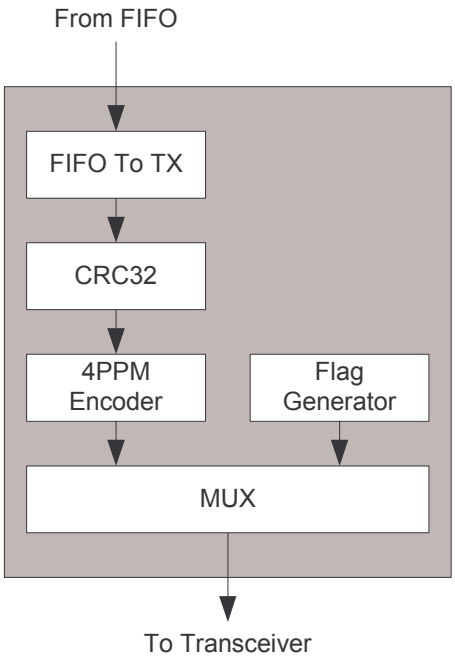

Figure 13. FIR TX block diagram

The task of *FIR TX* circuit is to divide each data byte read from *FIR TXFIFO* into a bit sequence and send it out after modulated as 4PPM format along with calculated CRC32 data. Flags like PA, STA and STO are not 4PPM modulated, they are sent out directly.

FIR *TX* circuit is mainly consisted of 6 parts: *FIFO to TX* circuit, *CRC32* circuit, *4PPM Encoder*, *Flag Generator* and *Output MUX*.

1. **FIFO to TX circuit**: The *FIFO to TX* circuit reads each data byte from *FIR TXFIFO* and transform it to bit stream which is then being sent to the *CRC32* circuit.

2. **CRC32 circuit**: The *CRC32* circuit is utilized for the calculating of a 32-bit frame check sequence according to the payload data. The calculation starts once the *FIFO to TX* circuit starts sending the bit stream into the *CRC32* circuit and stops at the end of the data. Afterward, the calculated 32-bit frame check sequence is cascaded to the end of the bit stream. Therefore, both the original bit stream and the 32-bit frame check sequence are sent to the *4PPM Encoder*.

3. **4PPM Encoder**: The *4PPM Encoder* simply modulates the bit stream received from *CRC32* circuit into 4PPM format and sends it to the *Output MUX*.

4. **Flag Generator**: The *Flag Generator* generates PA, STA and STO flags. The generated flag bit stream is sent to the *Output MUX*.

5. **Output MUX**: The *Output MUX* multiplexes the output to IrDA transceiver between the *4PPM Encoder* and the *Flag Generator*. A 4PPM packet contains a 16-bit PA flag, a 32-bit STA flag, variable length of 4PPM modulated payload data with frame check sequence, and a 32-bit STO flag. This multiplexer switches the output of the *Flag Generator* to the transceiver when sending the PA, STA and STO flags, else switches the output of *4PPM Encoder* to the transceiver when sending 4PPM modulated data with frame check sequence.

2.6. IrGate-IrPHY Register List

2.6.1. Master Control Register (MCR)

Bit #	Access	Description
0	R/W	Mode Switch. Set this bit to '1' when doing mode switching and clear this bit before next mode switching.
1	R/W	Mode, Select transmit/receive mode '0' – Receive mode '1' – Transmit mode
2	R/W	Reserved
4:3	R/W	Master_Speed, Select the type of modulation used based on speed group '00' – Low speed, SIR (9.6kbps to 115.2kbps) '01' – High speed, FIR (4Mbps) '10' – IR transmit mode '11' – Reserved
5	R/W	Reserved
6	R/W	Reserved
7	R/W	Reserved

Reset value: 00h

Note : Set address to "0000" to access the MCR.

2.6.2. SIR Mode Registers

Name	Address	Width	Access	Description
Master Control Register (MCR)	'0000'	8	R/W	Controls global settings of the core
Baud Rate Control Register (BRCR)	'0001' *	8	R/W	Setting baud rate
FIFO Control Register (FCR)	'0010' *	8	R/W	Control FIFO options
Transmitter FIFO (TxFIFO)	'0011' *	8	W	Transmit FIFO input

Receiver FIFO (RxFIFO)	‘0100’ *	8	R	Receiver FIFO output

* These addresses are valid when MCR[4:3] = “00”

- **Baud Rate Control Register (BRCR)**

Bit #	Access	Description
7:0	R/W	Define baud rate. value baud rate : ‘00000000’ 9600 ‘00000001’ 19200 ‘00000010’ 38400 ‘00000011’ 57600 ‘00000100’ 115200 others 9600

Reset value: 00h

- **FIFO Control Register (FCR)**

Bit #	Access	Description
0	W	Reserved
1	W	Writing a ‘1’ to bit 1 clears the Receiver FIFO and resets its logic.
2	W	Writing a ‘1’ to bit 2 clears the Transmitter FIFO and resets its logic.
7:3	W	Reserved

Reset value : 00h

2.6.3. FIR Mode Registers

Name	Address	Width	Access	Description
Master Control Register (MCR)	‘0000’	8	R/W	Controls global settings of the core
Interrupt Enable Register (IER)	‘0001’ *	8	R/W	Defines interrupt conditions
Interrupt Identification Register (IIR)	‘0010’ *	8	R/W	Describes pending interrupts
FIFO Control Register	‘0011’ *	8	R/W	Controls FIFO settings

(FCR)				
Line Control Register (LCR)	‘0100’ *	8	R/W	Break output and end of frame behavior control
Outgoing Frame Data Length Register 0 (OFDLR0)	‘0101’ *	8	R/W	Controls end of frame in another fashion (Lower byte)
Outgoing Frame Data Length Register 1 (OFDLR1)	‘0110’ *	8	R/W	Controls end of frame in another fashion (Higher byte)
Incoming Frame Data Length Register 0 (IFDLR0)	‘0111’ *	8	R	Length of received incoming data (Lower byte)
Incoming Frame Data Length Register 1 (IFDLR1)	‘1000’ *	8	R	Length of received incoming data (Higher byte)
Receiver FIFO (RxFIFO)	‘1001’ *	8	R	Read from this register to retrieve data
Transmitter FIFO (TxFIFO)	‘1010’ *	8	W	Write to this address to send data

* These addresses are valid when MCR[4:3] = “01”

■ Interrupt Enable Register (IER)

Bit #	Access	Description
0	R/W	Receiver FIFO trigger level reached interrupt
1	R/W	Received End of Frame
2	R/W	CRC check on input data failed
3	R/W	Receiver FIFO overrun
4	R/W	Receiver Error This interrupt will be requested when illegal data has been received or break was detected.
5	R/W	Transmitter FIFO low interrupt. This interrupt will be requested based on the Transmitter low trigger level.
6	R/W	Transmitter underrun. The transmitter wants to send more data but no data is available in the FIFO. In this case the controller will transmit a break for the receiving side to abort reception.
7	R/W	Reserved

Reset value : 00h

■ Interrupt Identification Register (IIR)

Bit #	Access	Description
0	R	Receiver FIFO trigger level reached. This bit will be set when the number of word in the receiver FIFO is above the high trigger level. It will be cleared when data is read from the FIFO and its count drops trigger level. The matching interrupt, if enabled, will be issued on the rising edge of this bit. If the DMA mode is enabled then a DMA request to read the data will be sent.
1	R	Received End of Frame. This bit will be set when the last byte in frame is within the receiver FIFO. An interrupt is generated on rising edge.
2	R	CRC check on input data failed. This bit is raised when after receiving the frame the CRC check on incoming data is failed. This bit is cleared after reading from the register.
3	R	Receiver FIFO overrun error occurred. This bit is raised when the FIFO is full and new data was received. In that case the new data won't be transferred to the FIFO but the bit will be set and any new data will overwrite the previous received byte. The bit is cleared upon reading from the register.
4	R	Receiver Error. This bit is set when an unexpected or illegal data has been received. This can be a break in transmission, illegal 4PPM chip values in 4Mbit mode or framing errors. This bit is cleared upon reading from the register.
5	R	Transmitter FIFO is low. This bit is raised and held while the count of bytes in the FIFO is below the lower trigger level. This is an indicator that new data can be written to the controller. If the DMA mode is enabled then a DMA request to transmit new data will be sent.
6	R	Transmitter underrun. The transmitter wants to send more data but no data is available in the FIFO. The controller will transmit a break. This bit is cleared upon reading from the register.
7	R	Controller Busy. This bit is set while the transmitter is busy transmitting or receiving data. It is clear when the controller is idle or transmits a Serial infrared Interaction Pulse (SIP) and the transmit FIFO is empty.

Reset value : 00h

■ **FIFO Control Register (FCR)**

Bit #	Access	Description
1:0	R/W	Set receiver FIFO trigger level (Number of bytes in FIFO that will set bit 0 in Status register). '00' – 8 bytes '01' – 10 bytes '10' – 12 bytes '11' – 14 bytes
2	W	Clear Receiver FIFO. Write '1' to this bit to clear receiver FIFO.
3	W	Reserved
5:4	R/W	Set transmitter FIFO trigger level – number of bytes in FIFO that an equal or smaller amount of bytes in FIFO will set bit 5 of the status register and request an interrupt and a DMA transfer. '00' – 2 bytes '01' – 4 bytes '10' – 6 bytes '11' – 8 bytes
6	W	Clear Transmitter FIFO. Write '1' to clear the FIFO. This will result in aborted frame and transmitting a break.
7	R/W	End of frame. Write to this bit to change behavior on transmitter FIFO underrun condition. '0' – The transmitter FIFO underrun will result in aborted frame and an interrupt will be raised if it is enabled. '1' – The FIFO underrun will result in normal termination of outgoing frame, i.e. the calculated CRC will be sent, followed by the frame stop sequence. The interrupt will be generated in this case, too, if enabled. Set this bit to '1' just before sending the last data to the controller if you don't use Count Outgoing Data mode. Reset this bit in response to FIFO underrun interrupt or when sending next frame.

Reset value: 00110011b

■ **Line Control Register (LCR)**

Bit #	Access	Description

Bit #	Access	Description
0	R/W	Force break. '0' – transmission is enabled '1' – transmission is disabled. The output pin is forced to break state (zero) regardless of what the transmitter is doing.
1	R/W	Count Outgoing Data mode select When enabled, the controller will count the bytes being transmitted in each frame and compare it to the number stored in Outgoing Frame Data Length register. When the count will have reached the designated number the controller will end the frame in normal fashion. Look at the OFDL register for more information. When this mode is disabled, the controller will know when to end the frame by using bit 7 of FIFO Control register. See bit 7 in FCR for more information on this mode of operation. '0' – Count Outgoing Data mode disabled '1' – Count Outgoing Data mode enabled
7:2	R/W	Reserved

Reset value : 00h

- **Outgoing Frame Data Length Register (OFDLR)**

OFDLR0

Bit #	Access	Description
7:0	R/W	Length of outgoing frame (Lower byte)

Reset value : 00h

OFDLR1

Bit #	Access	Description
7:0	R/W	Length of outgoing frame (Higher byte)

Reset value : 00h

- **Incoming Frame Data Length Register (IFDLR)**

IFDLR0

Bit #	Access	Description

Bit #	Access	Description
7:0	R	Length of incoming frame (Lower byte)

Reset value : 00h

IFDLR1

Bit #	Access	Description
7:0	R	Length of incoming frame (Higher byte)

Reset value : 00h

CHAPTER 3 IrGate

3.1. Introduction to IrGate

IrGate is the terminal device of IrGate System. The main purpose of IrGate is to provide an intermediate device between IrGate Client and IrGate Server. Through IrGate, an IrGate Client could use the services provided by IrGate Server.

3.2. IrGate Hardware

The hardware of IrGate is Samsung SNDS100 platform. With a built in S3C4510B01[6] microprocessor which has an ARM7TDMI processor core inside, this platform is capable of running an embedded operating system such as uClinux[10]. Its 10Base-T ethernet port provides a connection with other devices on internet or intranet. On the other hand, the GPIO and system bus together offer a control or data interface cooperating with self-designed hardware. The SNDS100 platform can utilize external interrupt and DMA transfer, both of these features help a lot in designing an effective control scheme and also a stable data transfer.

3.3. IrGate Architecture

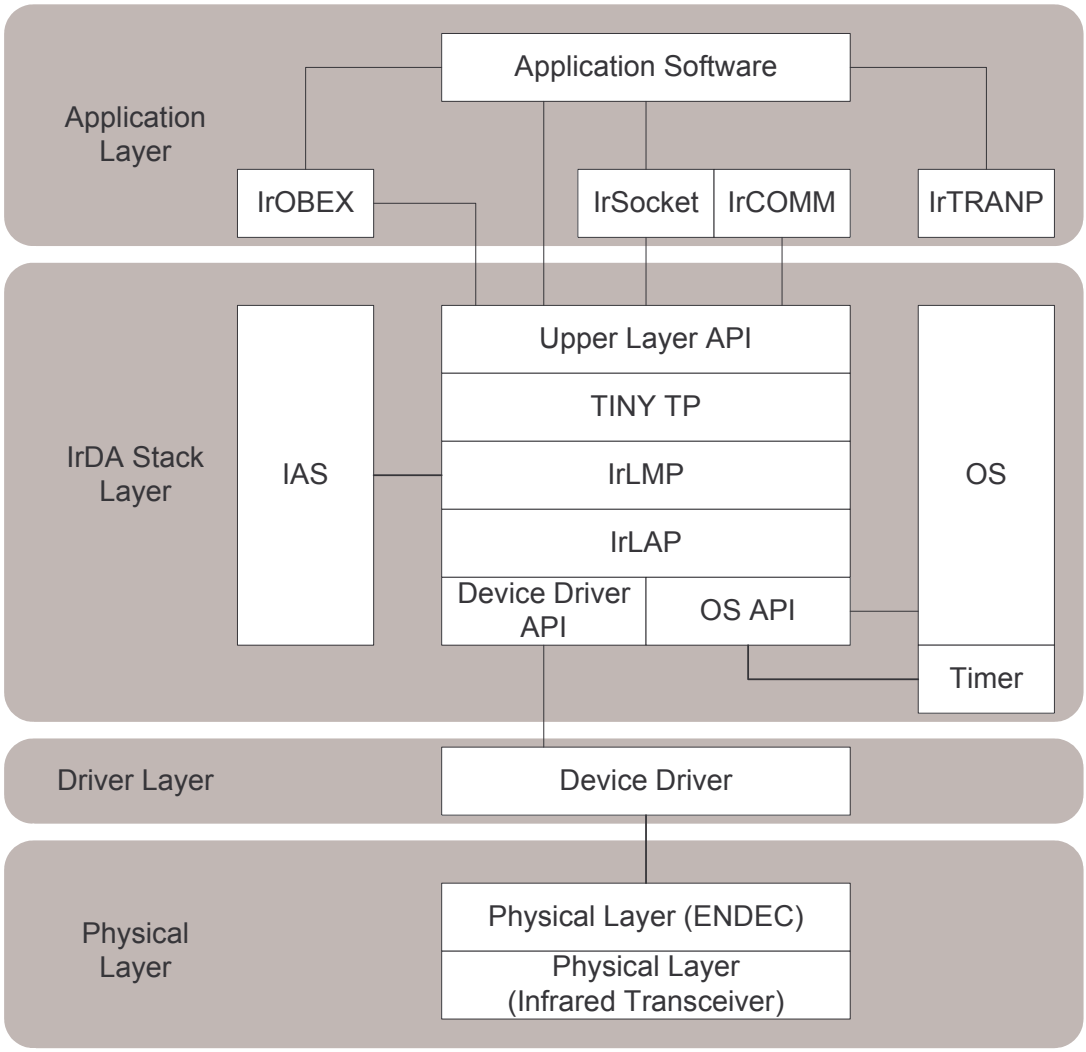

Figure 14. IrGate architecture

IrGate architecture is shown in Figure 14. Each layer and its relative blocks are described below:

- **Application Layer**: The application layer is consisted of 2 parts, one part is the optional upper IrDA protocol stack including IrCOMM[1], IrOBEX[1], IrSocket[11], IrTRANP[1] and so on, and another part is the application software. IrCOMM is the abbreviation for "Infrared Serial and Parallel Port Emulation over

Infrared", which is a wire replacement for serial and parallel port communication by infrared link. IrOBEX is the abbreviation for "Infrared Object Exchange protocol" which allows device independent, arbitrary transfer of data units. IrTRANP stands for "Infrared Transfer Picture" protocol which standardizes picture transfer between digital still cameras. IrSocket is a socket implementation of Linux IrDA stack that enables socket-based communication via an infrared transceiver. It is designed to support the industry-standard IrDA protocols. Applications implement IrSocket are much the same way as conventional socket. IrSocket is the protocol used by IrGate to link application layer and IrDA stack layer. Based on IrSocket, an application called "IrDA-TCP/IP Gateway" was implemented for IrGate to communicate with IrGate Server. This application uses IrDA socket for the interacting with IrDA stack layer.

- **IrDA Stack Layer**: IrDA stack is the primary part of IrDA protocol. The function of IrDA stack is to unwrap the data sending out from device driver and redirect it to application layer. There are 3 main components in this layer: IrLAP[1], IrLMP[1] and IAS[1]. IrLAP (Link Access Protocol) establishes the basic reliable connection, IrLMP (Link Management Protocol) multiplexes services and applications on the LAP connection, and IAS (Information Access Service) provides a "yellow pages" of services on a device.

- **Driver Layer**: The driver layer makes a connection between the physical layer and Linux kernel. The infrared data received by physical layer must be transmitted to IrDA stack which resides in Linux kernel for unwrapping and redirection. To make this work, there must be a linkage between the hardware – physical layer, and the software – IrDA stack, and this linkage is the device driver. The function description of driver layer could be divided into 2 parts:

receive and transmit. The descriptions of both functions are listed below:

- **Receive**: When physical layer received a complete set of infrared data and was ready to send it to the OS for further handling, it would generate an interrupt signal to inform the driver so as to trigger a receiving process in which the driver read each data byte from physical layer by means of DMA transfer and then wrote it into buffers in the OS. Once this had been done, the driver was ready to send an IrDA packet stored in the buffer to the upper layer – IrDA stack.

- **Transmit**: When IrDA stack invoked a driver API requiring device driver to transmit an IrDA packet out, the driver would start a process for transmitting this packet out. In the beginning of the process, the driver would write to a register in IrGate-IrPHY hardware for informing the physical layer to initiate a transmit process. Then, the physical layer would send an interrupt signal each time when its transmitter FIFO reached the low trigger level. Upon receiving this interrupt event, the driver would continuously write each byte of IrDA packet being transmitted out to the FIFO in IrGate-IrPHY hardware via DMA transfer until the transmitting task was completely carried out.

- **Physical Layer**: This layer contains an IrDA transceiver and IrGate-IrPHY hardware as mentioned in previous Chapter.

3.4. IrGate Operation Flow

In this section, main operation flows of IrGate are demonstrated in the form of sequence diagrams. Figure 15 shows the two operation flows of “discovery” operation and

"login" operation, and Figure 16 shows the another two operation flows of "get message" operation and "logout" operation. Short descriptions of each operation were listed below each figure to help for explaining each flow.

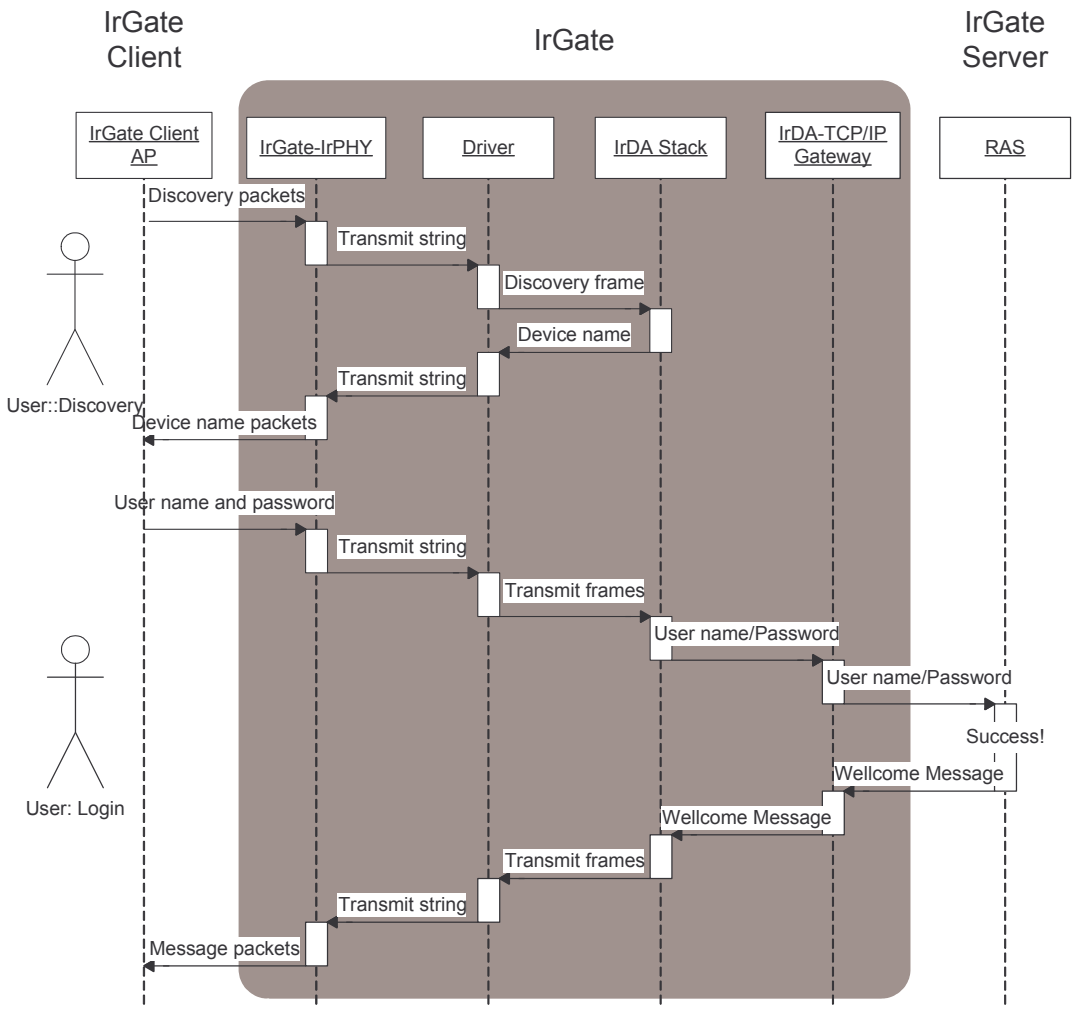

Figure 15. IrGate operation flow of "discovery" and "login"

- **Discovery (with Connection)**: A successful discovery operation should be done before any other operations. A discovery process started when user pressed the "connect" button on IrGate Client. After that, IrDA stack in the OS of IrGate Client began to search for any nearby IrGate device by sending out discovery packets several times. If any IrGate device was found through receiving its response packet by IrGate Client, a discovery process was then completed. After

the completion of the discovery process, a connection between IrGate Client and IrGate could be established by negotiating maximum speed, device name and other necessary information for building a stable, one-to-one and bidirectional infrared data channel on both sides. After a successful sequence of discovery and connection operations, an IrGate device found should be registered in the OS of IrGate Client. Thus, a connection channel was established between IrGate Client and IrGate device. With the existence of this connection, IrGate Client was then able to communicate with IrGate Server via the interconnections with IrGate. Note that the operation flow of discovery process illustrated in Figure 15 is a simplified model, a more detailed illustration of the discovery process and the connection process could be found in IrLAP specification[1].

- **Login**: A successful login process is required before performing other operations. Upon receiving a login request with username and password by IrGate Client, IrGate would transfer these request packets to IrGate Server for further authentication. Once the request was granted by IrGate Server, a successful login status with welcome messages would be returned to IrGate and then passed to IrGate Client. Consequently, messages showing a successful login operation along with welcome messages would appear on the screen of IrGate Client device. Now consider another case in which the login process was failed. In this case, the returned status value indicating a login failure event from IrGate Server to IrGate would cause IrGate to inform IrGate Client with a login failure message. Once this had happened, the connection between IrGate and IrGate Client would be disconnected and, as a result, another discovery process must be performed for a new login operation.

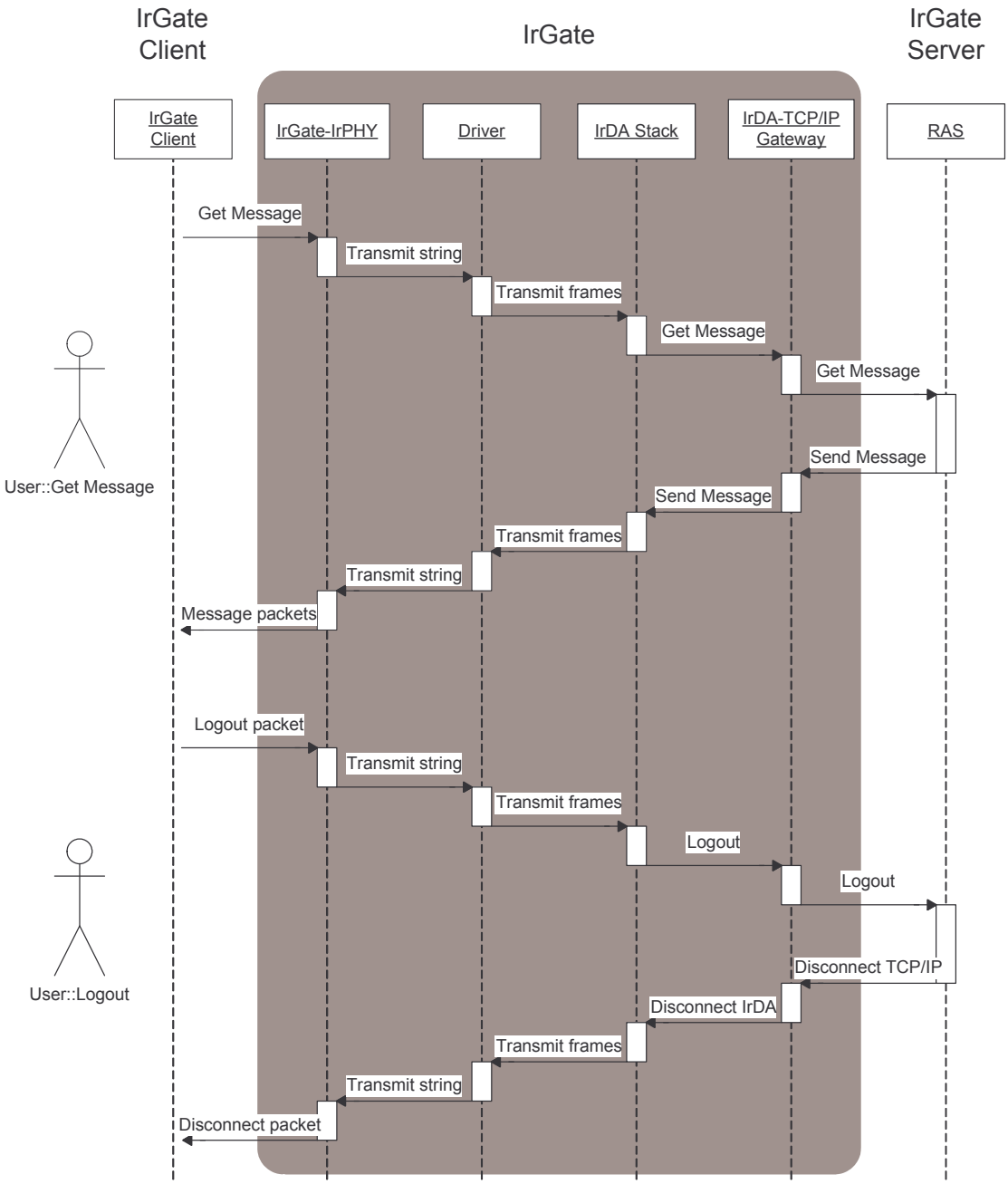

Figure 16. IrGate operation flow of "get message" and "logout"

- **Get Message (General Operation)**: After the login process was completed, messages including personal messages or system announcements stored in IrGate Server could be retrieved to IrGate Client by the performing of "get message" operation. The "get message" operation flow represents a common model for all general operations, such as "send message" or "open door". These operations

could be started by sending out an IrDA packet containing a command string in it from IrGate Client to IrGate. On receiving this packet, IrGate would unwrap this packet so as to obtain the command string inside and then send this string to IrGate Server via TCP/IP connection. Once this string was received by the Remote Access Shell (RAS) in IrGate Server, tasks would be executed correspondingly and message strings representing the results of executed tasks would be returned after the execution. These message strings would be packed (or wrapped) as one or several IrDA packets which were finally sent back to IrGate Client via IrDA connection. Figure 17 shows this kind of IrGate operation flow model for general commands.

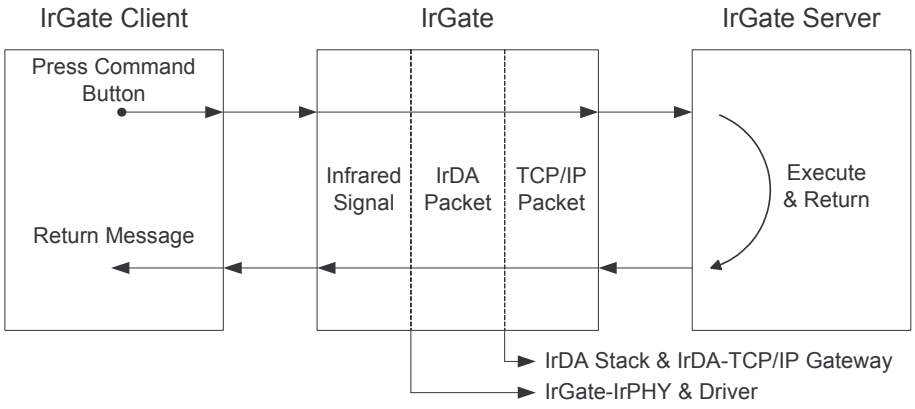

Figure 17. IrGate operation flow model

- **Logout**: The logout operation could be used to disconnect with IrGate and also with IrGate Server. By sending a command string representing the logout operation, IrGate Client could tell IrGate Server to refresh its status stored in the database as “disconnected” and IrGate Server would return a goodbye message on finishing updating the database. Once the message was sent to IrGate Client through IrGate, IrGate would soon afterwards cut off the connection with IrGate Client which consequently caused a disconnect message generated by IrGate Client to be shown on its screen.

CHAPTER 4 IrGate Server

4.1. IrGate Server Block Diagram

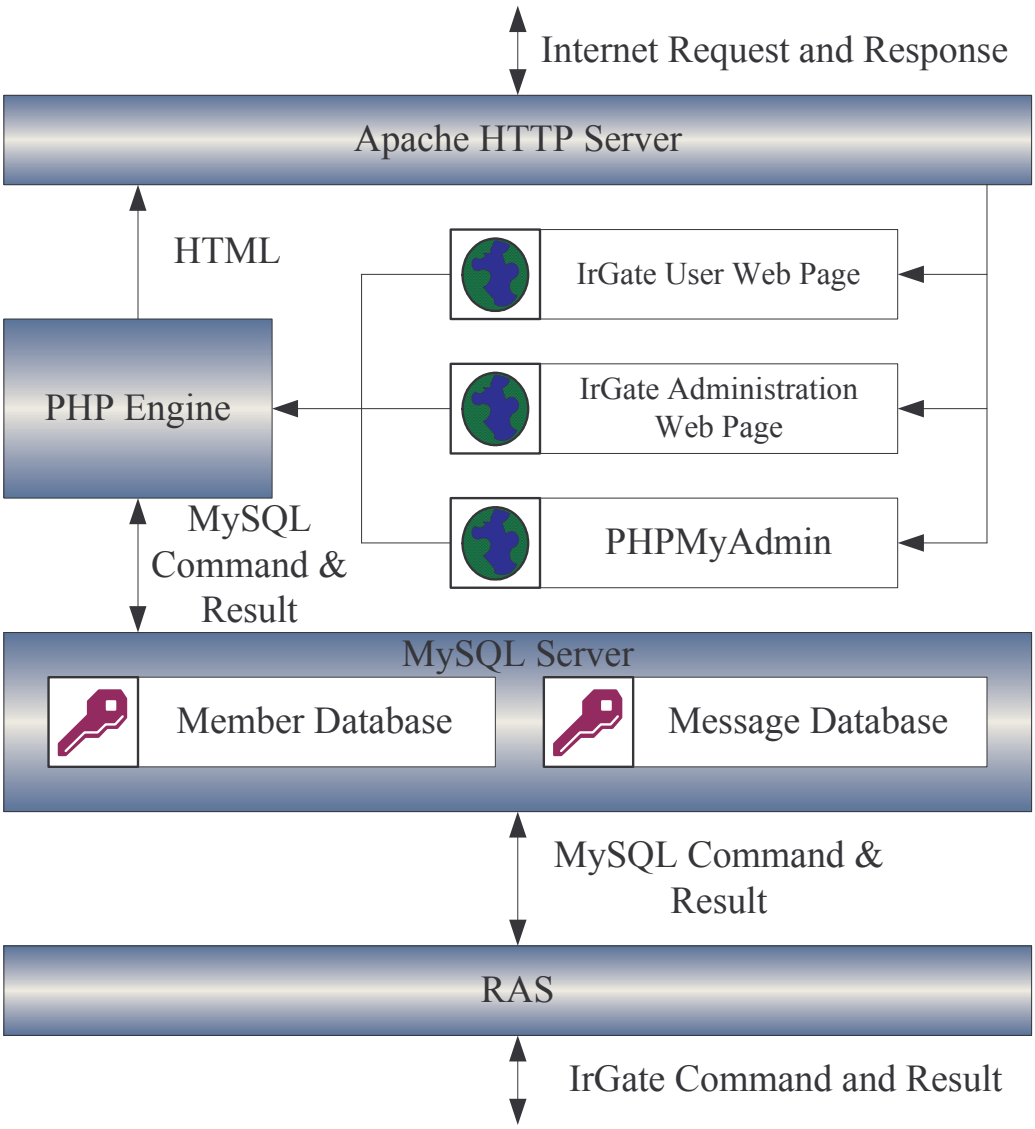

Figure 18. IrGate Server Block Diagram

The block diagram of IrGate Server is illustrated in Figure 18. The software components of IrGate Server are divided into 2 categories: Prerequisite **Software** and **Designed Software**. Descriptions of each software components in these 2 categories

are listed in the following sections.

4.2. Prerequisite Softwares

- **Apache HTTP Server**: The HTTP server used in IrGate System. The Apache Server is configured to support PHP so as to make both IrGate Web Administration Page and IrGate Web User Page work.
- **MySQL Database**: The database used in IrGate System. There are two main databases in IrGate Server: member database and message database. The member database is the database storing member data, and the message database is used to store the personal messages of IrGate System.
- **PHP**: PHP is a server side scripting language which integrates many powerful features. IrGate Server must have the PHP engine installed because it is necessary for both IrGate Web Administration Page and IrGate Web User Page to access the MySQL database.
- **PHPMyAdmin**: PHPMyAdmin[12] is an opensource software written purely in PHP which let site administrators to access MySQL database via internet. This software could aid the system administrator to manage the member database and the message database.

4.3. Designed Softwares

- **RAS**: A Remote Access Shell program that receives command strings from a remote device through TCP/IP network and executes tasks correspondingly. The RAS serves as a command shell for IrGate Client to communicate with IrGate

Server.

- **IrGate Administration Web Page**: A web page written in PHP and HTML for administrators to manage IrGate System on the internet. Administrators of IrGate System could login from the main page and then check the state of the system, manage user account and use all the system's services.

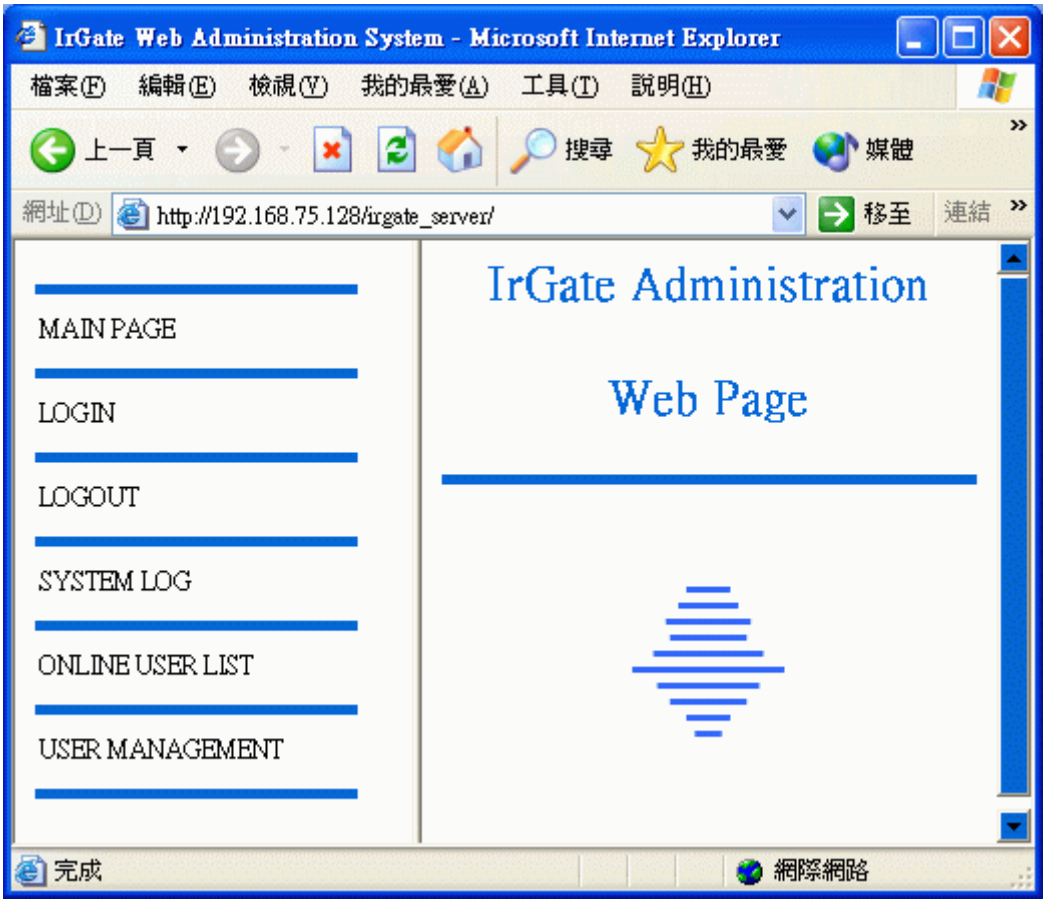

Figure 19. IrGate Administration Web Page

- **IrGate User Web Page**: Another web page written in PHP and HTML for the users of IrGate System to manage personal data and send personal messages on the internet. Users of IrGate System could login from the main page and then manage his or her own data. When someone loses his or her IrGate Client, this web page provides he or she an easy way to update the database so as to prevent somebody else from accessing IrGate System using his or her account.

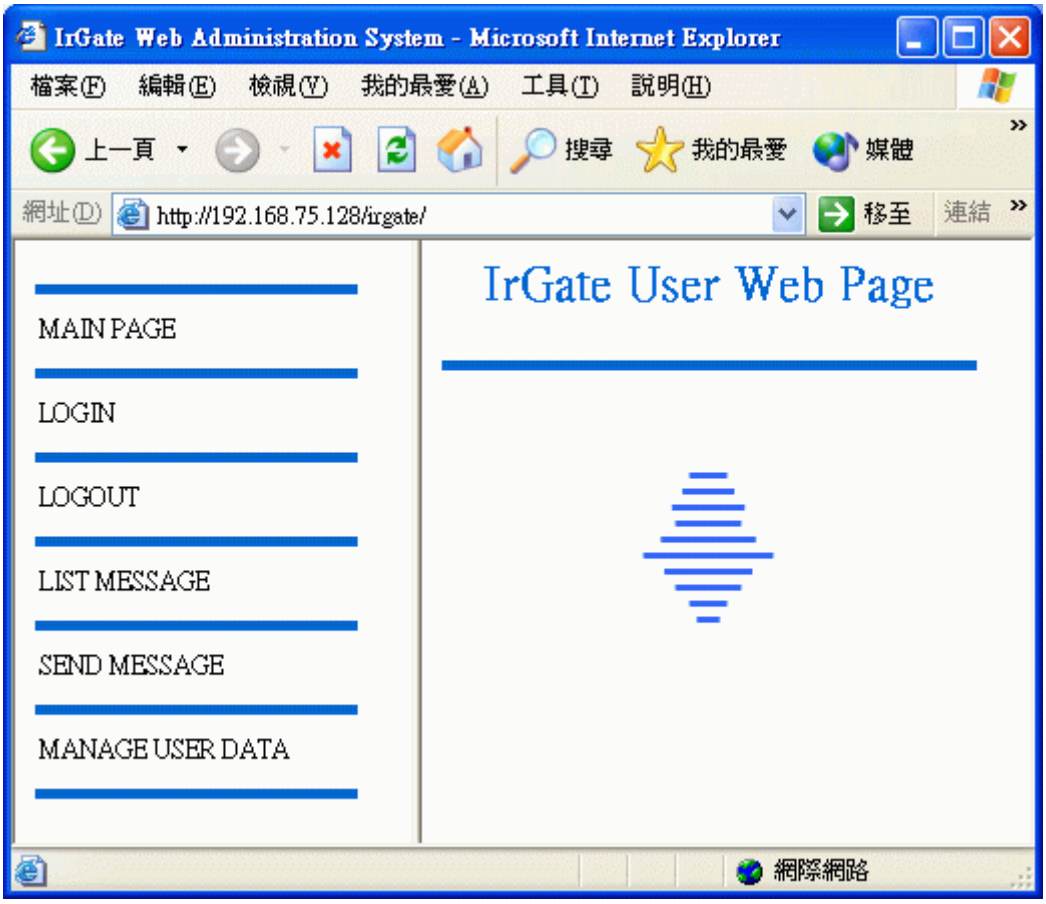

Figure 20. IrGate User We Page

- **Member Database**: The member database is used to store user information. The 2 tables in member database are shown below:

  user: stores member information of IrGate System

entry	description
id	user ID
password	user password
first_name	first name
last_name	last name
address	address
phone	phone number
sex	

  user_data: stores online user information

entry	description
id	user ID
ip	user IP
online	user online status

- **Message Database**: The message database is used to store personal message data. The 2 tables of message database are shown below:

  message: stores all information about a single personal message.

entry	description
msg_id	message ID
from_id	from user ID
to_id	to user ID
date	sending date
subject	message subject
text	message text content

  message_data: stores necessary data for management.

entry	description
msg_num	number of posted messages

  When a new message is posted, its msg_id is set to msg_num + 1, and msg_num increases by 1 too.

4.4. Operation Flow

The operation flow of IrGate Server is mainly composed of 2 parts: access via web browser and access via IrGate Client, both flows are indicated below:

- Access via web browser: As illustrated in Figure 21, this operation flow starts when the web browser sends a request to the HTTP Server and ends when the web browser gets the HTML result.

- Access via IrGate Client: As illustrated in Figure 22, this operation flow starts when IrGate Client sends a request to the HTTP Server and ends when IrGate Client gets the text result.

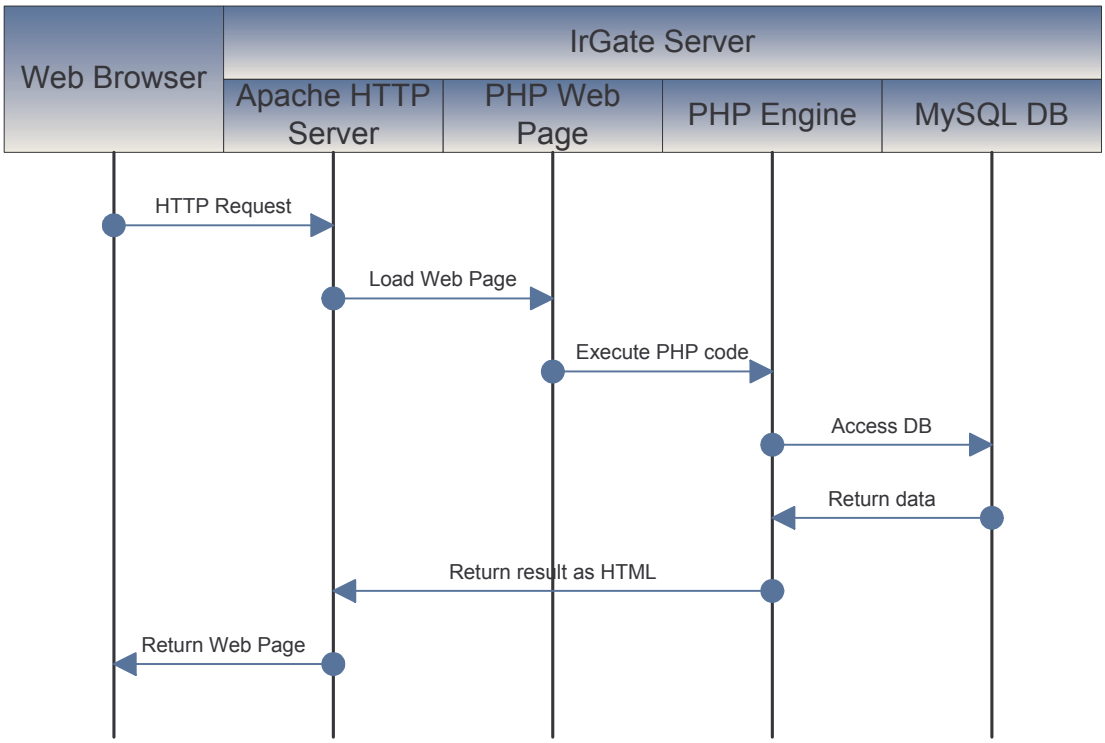

Figure 21. Access IrGate Server via web browser.

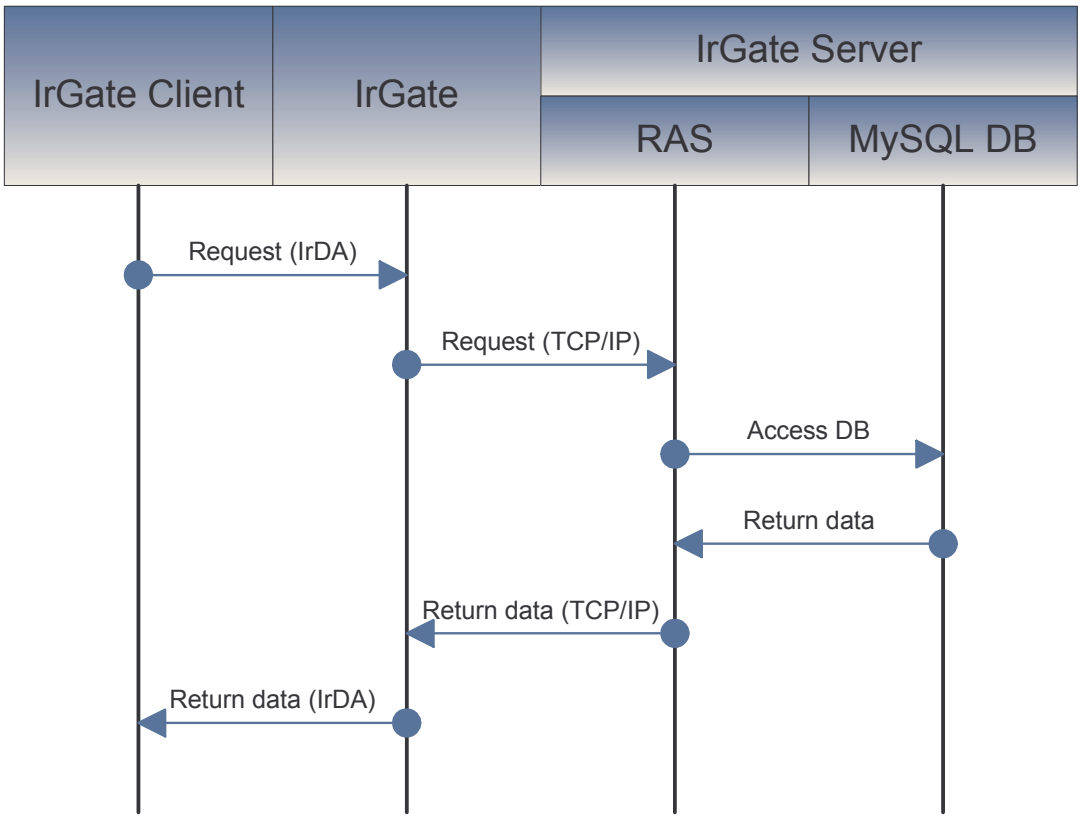

Figure 22. Access IrGate Server via IrGate Client.

CHAPTER 5 IrGate Client

IrGate Client is a handheld device running IrGate Client AP on it. The handheld device used is a COMPAQ iPAQ PDA with Pocket PC 2002[13] OS. This main point of this Chapter is to describes the function of IrGate Client AP. Detailed depiction of development process and the hardware and software environment are omitted in this Chapter.

5.1. IrGate Client AP

IrGate Client AP which was developed using Visual C++ provides a GUI for users to communicate with IrGate as illustrated in Figure 23. This program would communicate with IrDA-TCP/IP gateway in IrGate with IrDA ability provided by the PDA.

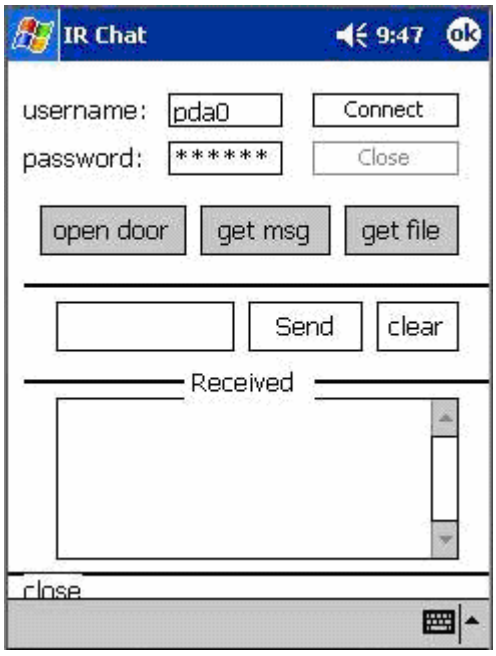

Figure 23. IrGate Client AP GUI

The operation flow of IrGate Client AP is illustrated in Figure 24:

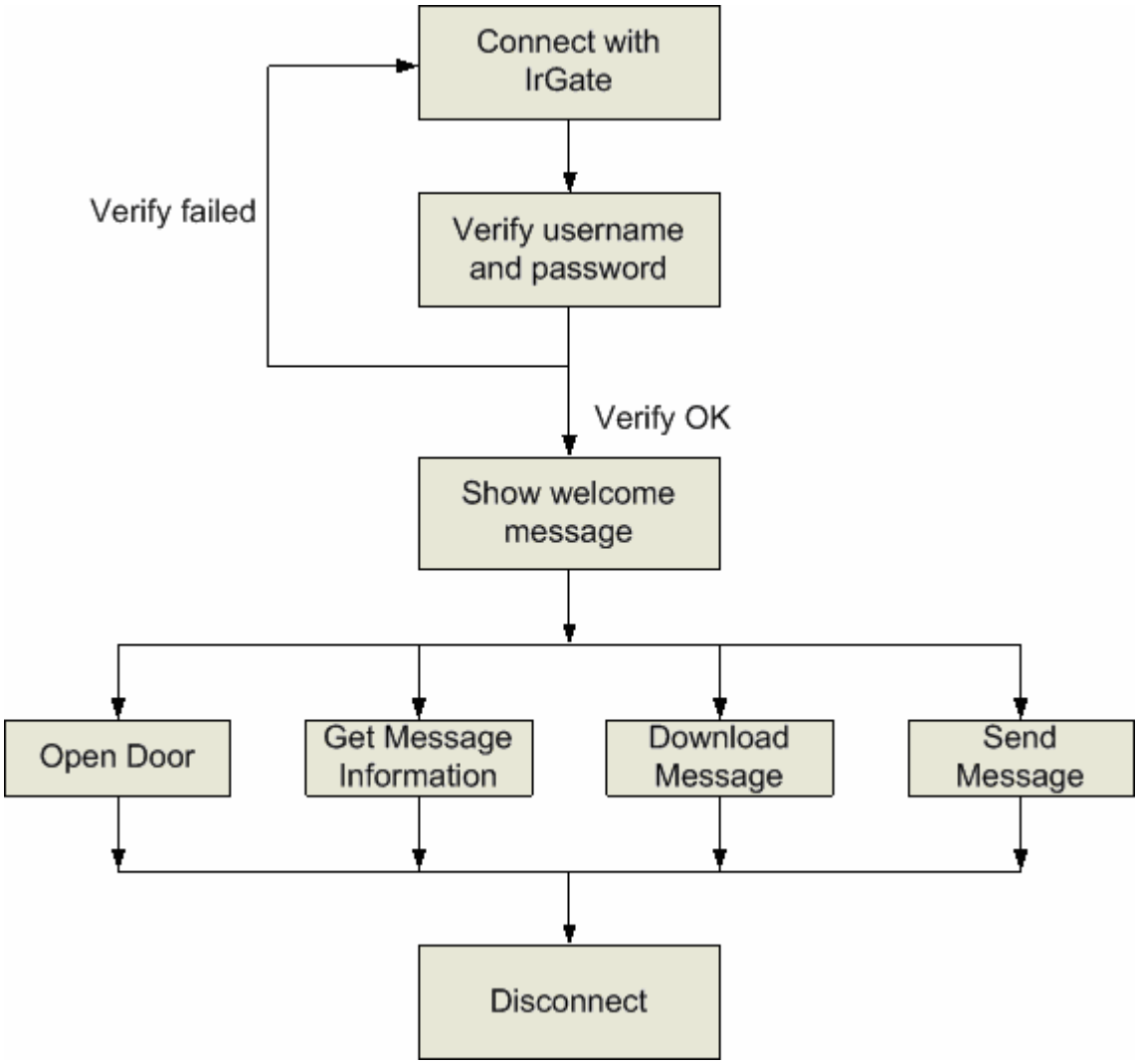

Figure 24. Operation flow of IrGate Client AP

The hardware device used to implement IrGate Client is a PDA. Figure 25 shows the appearance of the PDA running IrGate Client AP.

Figure 25. PDA running IrGate Client AP

5.2. IrGate Client GUI

IrGate Client GUI, as indicated in Figure 26, is mainly consisted of 4 parts:

- Connection area: Includes the text area for users to input username and password and a set of connect and close button for connection and disconnection.
- Operation area: The 3 buttons in this area could be used to send command to IrGate. When “open door” button is pressed, the door controlled by IrGate will be opened in a few seconds. When “get msg” button is pressed, a text will be shown in the received text field indicating how many personal messages the user

currently has in IrGate Server. When “download” button is pressed, all personal messages will be downloaded from IrGate Server to IrGate Client.

- Transmission area: Used for the transmission of personal message.
- Reception area: The text field in this area shows all messages received from IrGate, including welcome message, operation acknowledgement, personal message and system status.

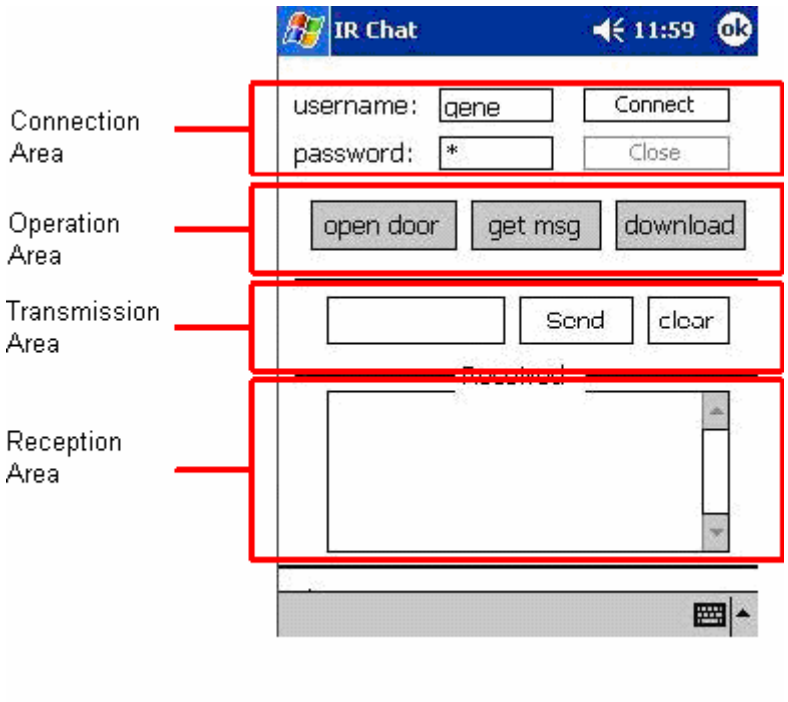

Figure 26. IrGate Client GUI

Once the user pressed the “Send” button, another window will appear to let user input the “recipient”, “subject” and “message content” of the personal message, as shown in Figure 27.

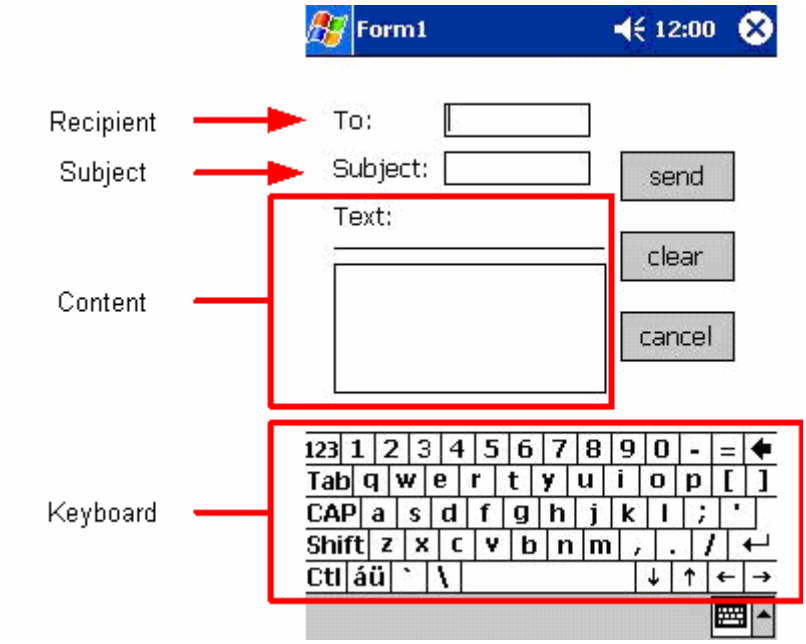

Figure 27. Send message window

CHAPTER 6 Conclusions and Future Works

6.1. Using IrGate System

The usage of IrGate System is as easy as other entrance systems. Moreover, the high integrity of IrGate System also produces many additional functions. When IrGate System is installed in an office building, staffs could enter specific area and send/receive bulletin, announcement of personal messages with handheld devices like mobile phones or PDAs.

Normal use case of IrGate System would start with adding member data into member database via IrGate Administration Web Page by administrators. Afterward, authorized member could enter IrGate-controlled rooms or areas with a “point-and-shoot” operation toward IrGate using their mobile phone. Once the membership expires or is cancelled, the system would reject the request of this member when he or she wants to login into IrGate System.

Detailed descriptions of each use case of IrGate System are listed below:

- Building member database

Building of member database could be accomplished using IrGate Administration Web Page. Administrators could use this tool to add member into member database on IrGate Server. And to cancel a membership of a user, the administrator just remove the authorization the member data of that user.

- Identification of member

The identification of member would decide whether to let the user pass an entrance or not. An identification process starts upon the pressing on the “connect” button in IrGate Client AP. After that, IrGate Client interacts with IrGate Server via IrGate and get identified if the information sent out from IrGate Client is valid. With a successful identification, a member could then get into places controlled by IrGate.

- Message system

After a member logins into IrGate System, he or she could use the message system to send or receive messages. Normally a welcome message would be received by IrGate Client just after a successful connection with IrGate Server. Additional information about the entered place or significant announcement may accompanied with the welcome message. Besides, personal messages could be received or sent using the message system.

6.2. Operation flow

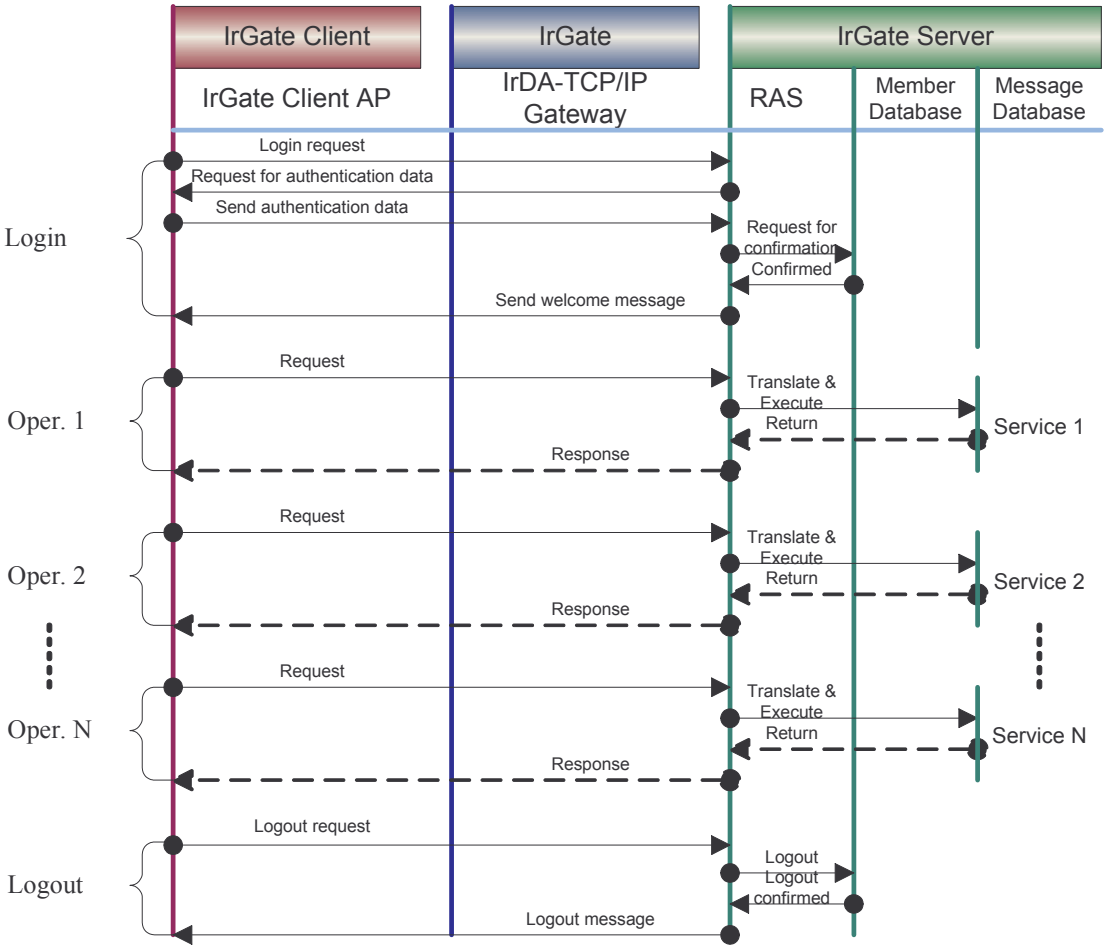

Figure 28. General operation flow of IrGate System.

Figure 28 shows the general operation flow of IrGate System. The operation flow is mainly divided into 3 groups: “Login”, “Operation” and “Logout”. At the beginning, a user must login into IrGate System first. After a successful login, the user could do a sequence of operations supported by IrGate System. In the end, the user must logout the system for the termination of all operations. The start point of any operation group is initiated from IrGate Client. The signals is sent to IrGate via IrDA connection and then transformed into TCP/IP packets which finally go into IrGate Server. The end point should be any software components such as RAS, Member Database, Message Database or any other services provided by IrGate Server.

For IrGate System, a most common use is to login into the system, download messages, open the door and then logout the system. The operation flow that follows the process sequence of “login”, “download message”, “open door” and “logout” is illustrated in Figure 29.

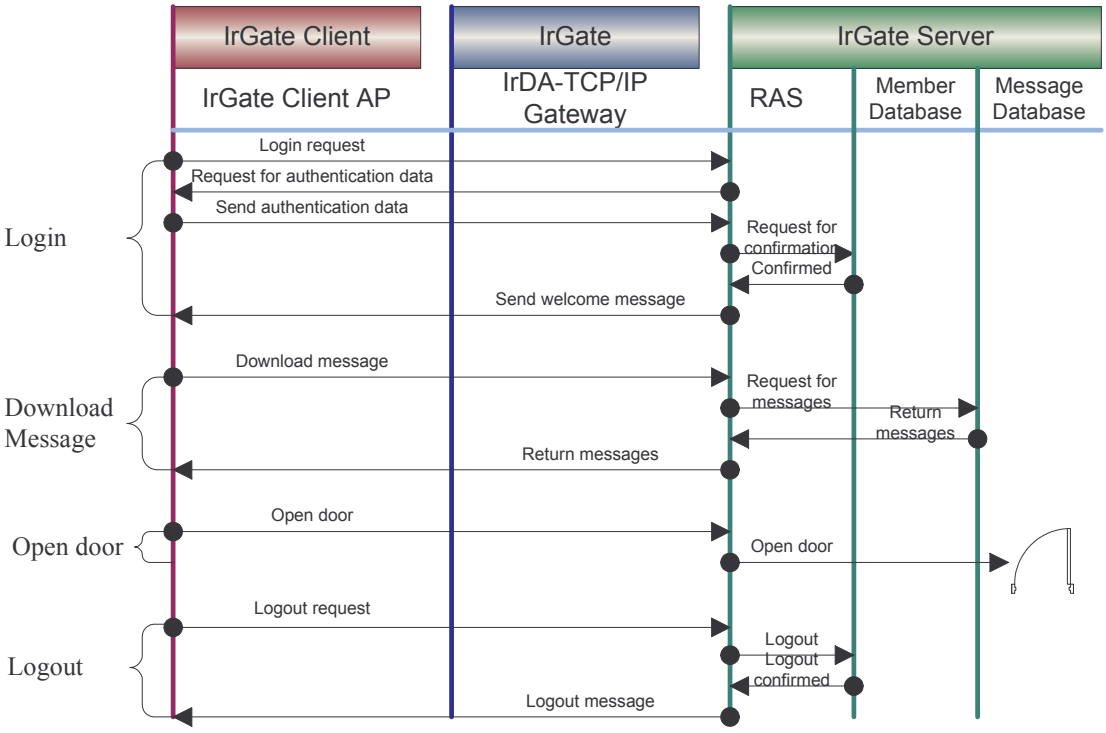

Figure 29. Normal operation flow of IrGate System.

6.3. Comparison

6.3.1. IrGate VS. traditional entrance systems

Being a new type of entrance system with wireless transmission, the IrGate system obviously overwhelms systems like IC card, magnetic stripe card specifically in functionality, scalability and interactibility.

A simple comparison table of various types of present entrance systems is given in

Table 1. IrGate System is observed to be comparatively better than other systems especially in functionality, scalability and interactibility. The security of IrGate System could be as good as that of IC Card because employing the same security system of IC Card on IrGate System is entirely possible. However, the convenience is the main weakness of IrGate System, since carrying a handheld device like mobile phone produces lots of inconvenience.

Table 1. Comparison of entrance systems

	Security	Functionality	Scalability	Interactibility	Convenience
Traditional Key	■□□	□□□	□□□	□□□	■■■
IC Card	■■■	■■□	■□□	■□□	■■■
Magnetic Stripe Card	■□□	□□□	□□□	□□□	■■■
Fingerprint Identification	■■□	□□□	□□□	□□□	■■■
Retina Identification	■■■	□□□	□□□	□□□	■□□
IrGate	■■■	■■■	■■■	■■■	■■□

6.3.2. IrDA VS. RFID

RFID stands for Radio Frequency Identification, which has the ability of allowing contactless, bidirectional storage of small amounts of data on trivial circuits powered by the reader infrastructure itself. The range of RFID varies quite a lot because of the non existing standard. RFID devices are created with the maximum range that the application requires. Typical readers only provide a few inches range.

As for application about access control and security, the new RFID is indeed a competitive technology of IrDA due to its reliable short-range wireless capability and ease of use. Thus, a comparison on security of both technologies is given here to give

reasons for why IrDA is better than RFID in secure applications. Note that the comparison was made on a premise that there are no security improvements for both technologies.

- Power Attack: The RFID systems are powered by the outside world. Given a cooperative RF field, the chip emits the same bits, over and over and over again. In contrast to RFID, IrDA has no such weak point since one must actively point-and-shoot to send out infrared signals.

- Sniff Attack: Both technologies could suffer from sniffing. However, since the narrow angle of infrared transmission, sniffing on IrDA is more difficult than that on RFID.

Various techniques have been exploited for solving the above problems of RFID. A technique that requires a person to pinch somewhere on the card to activate RFID functionality could be used to avoid power attack. Also, there are some works to embed cryptographic constructs into Passive RFID systems. Hash algorithms can be made using very little silicon, so having the card read some value from the badge reader and return a the hashed value with a shared secret can be a valid solution.

Nevertheless, the security of RFID chip is still an extraordinarily difficult problem to solve, because the chips are necessarily trivial since they're powered by the sensors. Not only is it nearly impossible to build any kind of cryptosystem into a chip that small and weak, but the system itself would remain completely defenseless against electrical hacking. Manipulating a chip's power source is one of the definitive ways of divining its cryptographic secrets.

On the contrary, the nature of IrDA makes it nearly invulnerable to power attack. And with the exploiting of a cryptosystem, sniffing attack may take no effects. Thus IrDA

is comparatively better than RFID in security.

6.4. Improvements

Following is the list of practicable improvements of IrGate System:

- **SSL**: IrDA Data Link Protocol has no encryption for its infrared signals sending out from IrDA transceivers. Thus the signals could be eavesdropped and valuable information carried in these signals could be extracted from IrDA packets. Adding SSL encryption to IrDA packets could be a significant improvement to this problem. The SSL protocol runs above TCP/IP and below higher-level protocols such as HTTP or IMAP. It uses TCP/IP on behalf of the higher-level protocols, and in the process allows an SSL-enabled server to authenticate itself to an SSL-enabled client, allows the client to authenticate itself to the server, and allows both machines to establish an encrypted connection. This encrypted connection could make up the lacking of encryption of IrDA transmission and also greatly improve the security of IrGate System.

- **Camera Module**: As an entrance system, real-time image monitoring is a helpful function for IrGate System. This could be brought to practice by adding a camera module on IrGate. The powerful hardware capability of IrGate makes it possible to add such a camera module and continuously send the captured image to IrGate Server via internet. System administrators or operators could take advantage of these images to identify the user and monitor the entrances.

6.5. Potential applications of IrGate System

Considering the properties of infrared data transmission:

- Short distance (0~1m)
- Narrow angle (30 degree cone)
- Hard to be eavesdropped
- Easy to use (Point-and-Shoot)

According to these properties, IrDA is suitable for proximity and security applications. As a result, the following applications are thought to be potential applications for IrGate System:

- **E-Key**: IrGate System is actually a kind of E-Key system which electronically integrates all kinds of applications in which people are accustomed to use keys. Traditionally, people use physical keys for home, cars, apartment houses, and so on. However, carrying all the keys in one's pocket is indeed not a good experience. The E-Key system helps to release people from this annoying situation. With E-Key system, a mobile phone can be a substitute for all keys in everyday life.
- **E-Wallet**: The E-wallet system allows users to store payment information in their mobile phones and use the wireless communication system such as IrDA to make payments. First, it allows wireless operators to not only issue digital cards but to immediately disable the payment function of stolen or lost mobile phones, and makes it easier to keep track of users who have defaulted on payment and simplifies re-issuance procedures while creating a digital card that is constantly linked to the credit card company. Second, it can relay personal payment information over a distance sufficient to make payments in such places as tollgates. In addition, two-way communication makes it possible to carry out various transactions necessary in everyday life such as digitally downloading

tickets and coupons, which are then used via infrared or fixed-wired and wireless Internet. As such, information necessary to issue or cancel cards, disabling of stolen or lost cards, as well as for users who have defaulted on payment, can be carried out on a mobile phone without having to be linked to a credit card company's database. Furthermore, the addition of proximity payment via infrared has made it possible for consumers to use their mobile phones to make payments which until now were only possible with real-world cash or credit cards.

Three application schemes, according to the potential applications of IrGate, are described below:

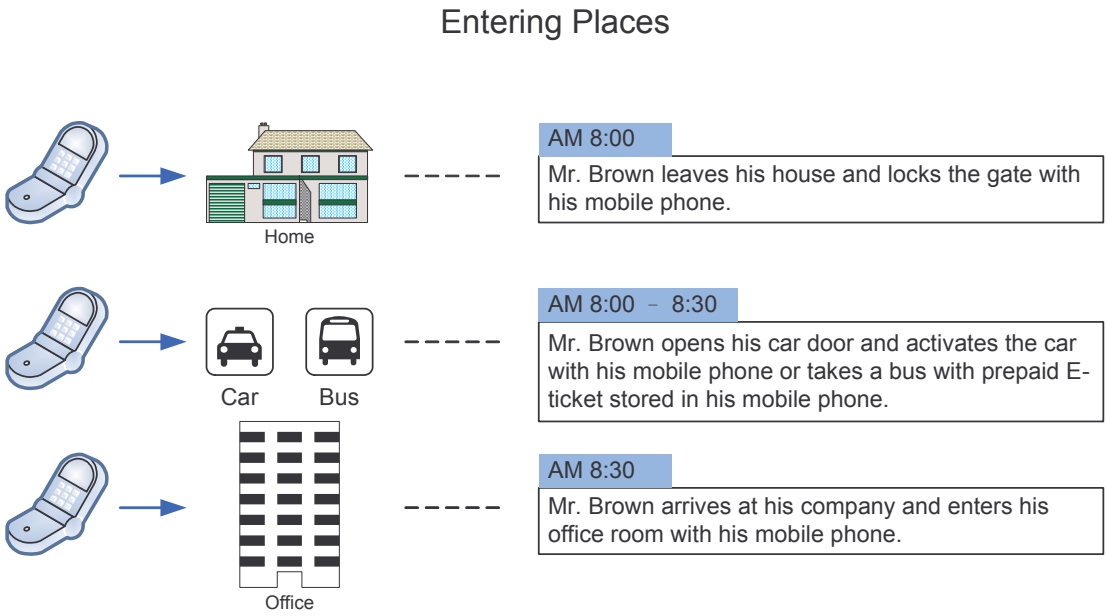

Figure 30. IrGate Application Scheme : Entering Places

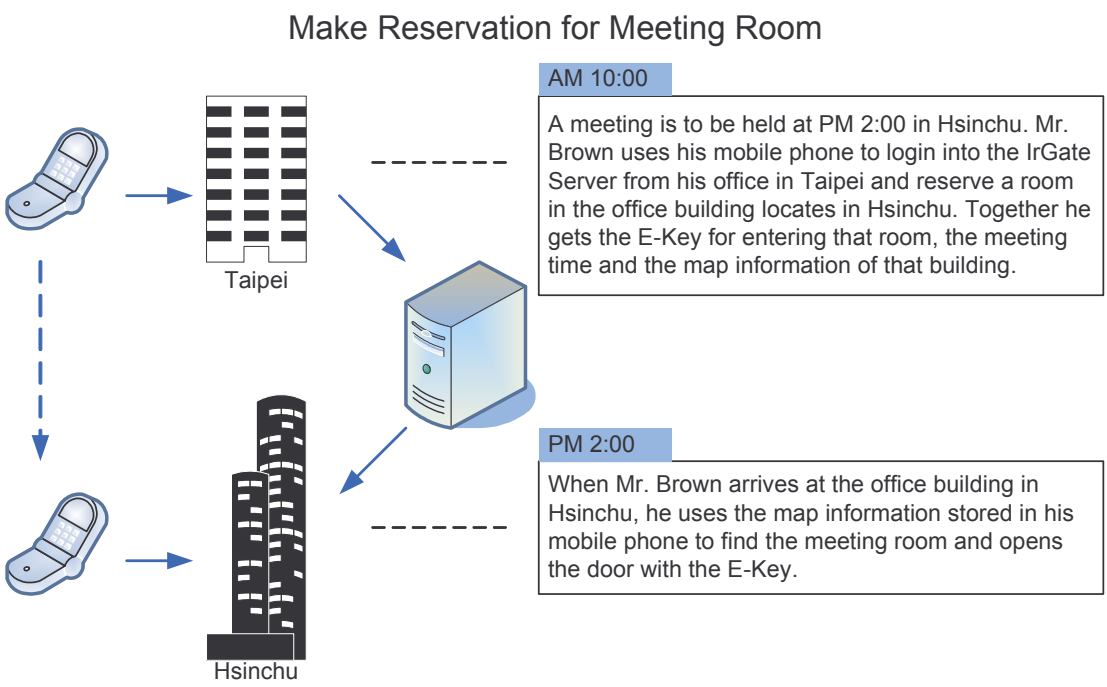

Figure 31. IrGate Application Scheme : Make Reservation for Meeting Room

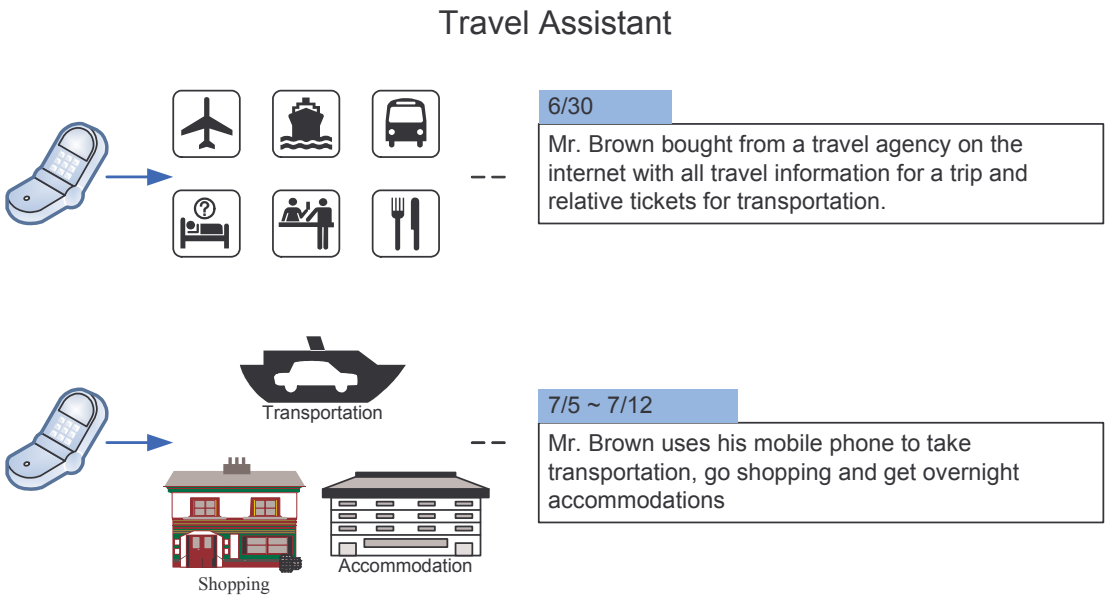

Figure 32. IrGate Application Scheme : Travel Assistant

APPENDIX A Introduction to IrDA

This appendix presents an overview of IrDA, and the features and requirements of IrDA compliant devices.

A.1. Overview of IrDA

The Infrared Data Association ,IrDA, is an independent organization whose charter is to create standards for interoperable, low cost IR data interconnection. Setting standards for IR communication is key to effortless communication between various types and brands of equipment. It is the goal of IrDA to set standards and protocols, which can be reasonably and inexpensively implemented in order to promote the usage of IR communication. first version of IrDA data link Physical Layer Specification (IrPHY) 1.0, provided for communication at data rates up to 115.2 kbps. Version 1.1 extended the data rate to 4 Mbps, while maintaining backward compatibility with Version 1.0 products. Version 1.2 defined a low power option for data transmission speed up to 115.2 kbps. Version 1.3 extended the low power option operation to 1.152 Mbps and 4 Mbps.

Without a communication protocol, such as that provided by IrDA, a non-cabled link is inherently not robust. Unlike a cable, which is semi-permanently attached, the ends of an IR link may move freely in and out of range. The link may even be broken in the middle of a transmission. IrDA defines a set of specifications, or protocol stack, that provides for the establishment and maintenance of a link so that error free

communication is possible. IrDA Standards include three mandatory specifications: the Physical Layer (IrPHY), Link Access Protocol (IrLAP), and Link Management Protocol (IrLMP).

While a new revision provides for additional options, it does not mean that the device has to meet it. For example, while IrPHY 1.3 supports 4 Mbps at low power, it does not mean that an IrPHY 1.3 compliant device has to support 4 Mbps at low power.

A.2. IrDA Datalink Protocols

IrDA Datalink Protocols are organized as a series of layers, each built upon the one below it, with the lowest layer being the physical layer. They may be visualized as a protocol stack. The function of each layer is to offer certain services to the next upper layer, shielding those layers from the details of how the offered services are actually being implemented. The three mandatory layers as mentioned above are the Physical Layer (IrPHY), Link Access layer (IrLAP), and the Link Management layer (IrLMP). A typical implementation of IrDA protocol stack within an operating system is shown in Figure 33.

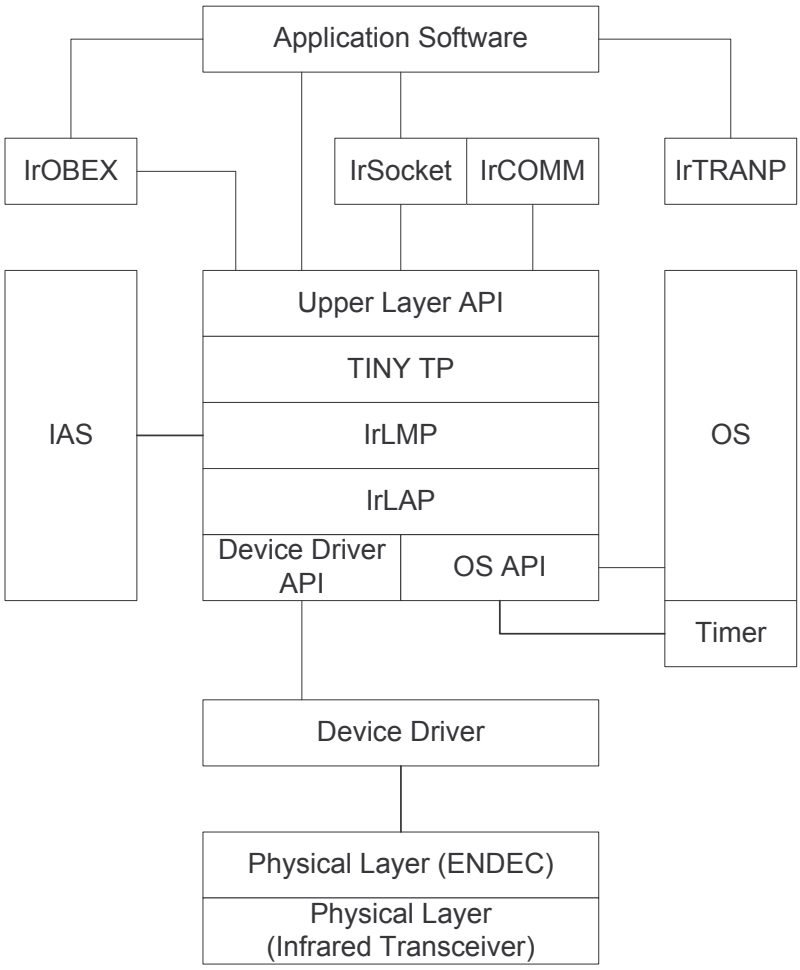

Figure 33. Example of a Typical IrDA Implementation in an Operating System.

A.3. IrPHY – IrDA PHYSICAL LAYER

A.3.1. Overview of IrPHY

The Physical Layer Specification provides guidelines for point-to-point communication between equipments using IR. The current specification (Version 1.3) supports the two options: standard power and low power. The standard power option ensures error free communication from a distance of 0 to 100 cm, at an off axis angle of 0 to at least 15° (Figure 34). The low power option, intended for handheld devices

and the telecommunication industry, ensures error free transmission from 0 to 20 cm, at an off angle of 0 to at least 15°. Included are specifications for modulation, viewing angle, optical power, data rate, and noise immunity in order to guarantee physical interconnectivity between various brands and types of equipment. The specifications also ensure successful communication in typical environments where ambient light or other IR noise sources may be present, and minimize interference between IR participants.

The specifications for optical intensity for the transmitter and sensitivity for the receiver were chosen to guarantee that the link will work from 0 to 100 cm for standard power devices, and 0 to 20 cm for low power option devices. The receiver sensitivity was chosen so that a minimum intensity emitter will guarantee the minimum link distance as specified.

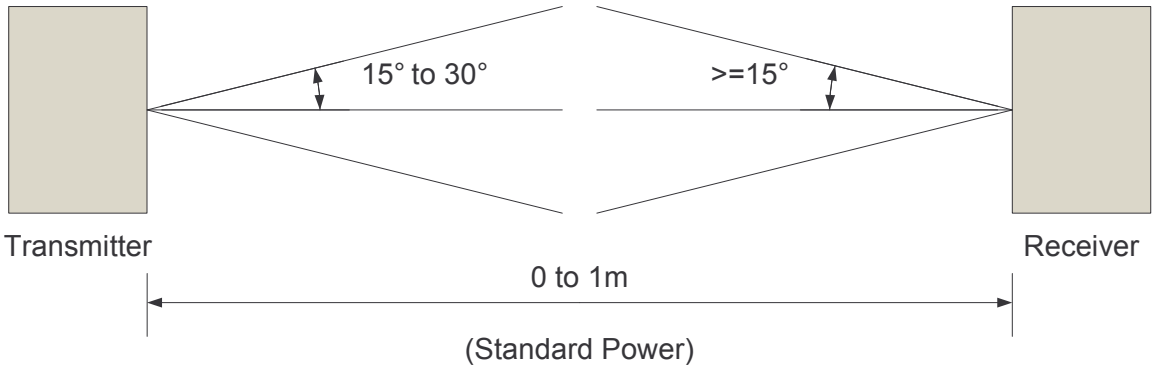

Figure 34. IrDA Physical Layer Viewing Angle and Distance.

Figure 35 shows a block diagram of the physical layer for data rates up to 115.2 kbps. This was conceived as a link that would work readily with conventional UARTs, such as the NS 16550. Thus it is a straightforward extension of the serial port. Note in Figure 35, however, that the data is first encoded before being transmitted as IR pulses. This is required because UARTs and serial ports use NRZ (non-return to zero) coding where the output is the same level for the entire bit period and can stay at one

level for multiple bit periods. This is seen in Figure 36 as the data labeled “UART frame.” This is not optimal for IR data transfer since a continuous string of bits could turn on the LED transmitter for an arbitrarily long time. Thus the power in the LED would need to be limited, which would then limit the working distance. Instead, IrDA standard requires pulsing the LED in a RZI (return to zero, inverted) modulation scheme so that the peak to average power ratio can be increased. The maximum pulse width is required to be 3/16 of the bit period. The minimum pulse width can be as little as 1.41 μs, which is derived from 3/16 of the highest data rate of 115.2 kbps. The effect of the encoding can be seen in Figure 36 as the data labeled “IR Frame.” A 16x clock is conveniently available on many UARTs, so it is easy to count three clock cycles to encode the transmitted data, and to stretch the received data with 16 clock cycles. Note that this scheme requires an encoder/decoder (endec), either embedded in the I/ O chip or as a discrete component.

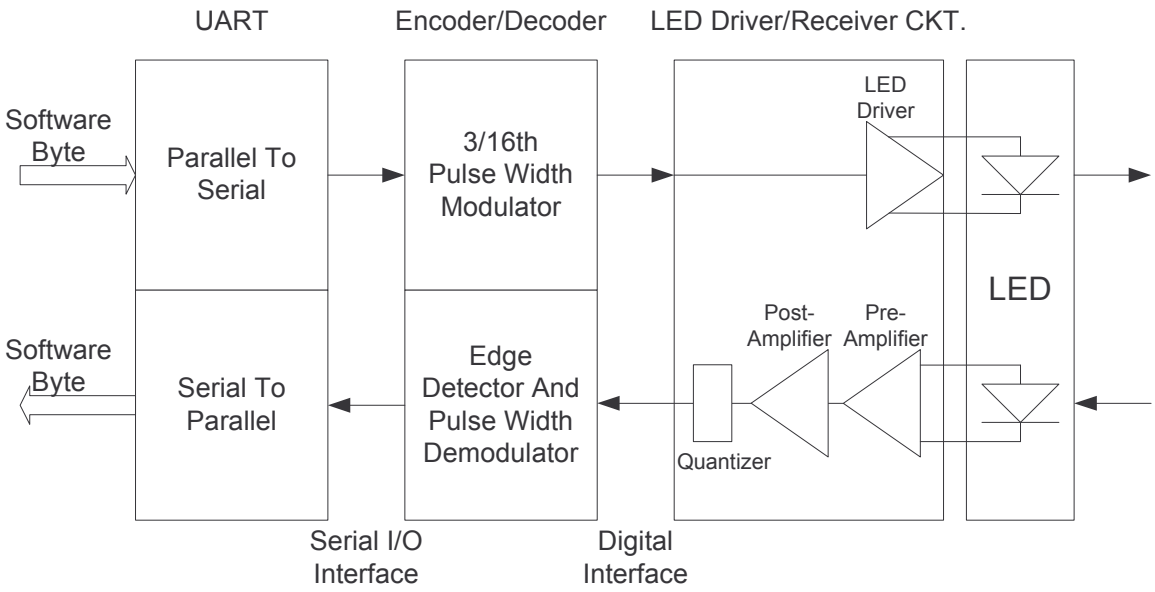

Figure 35. IrDA Version 1.0 Physical Block Diagram.

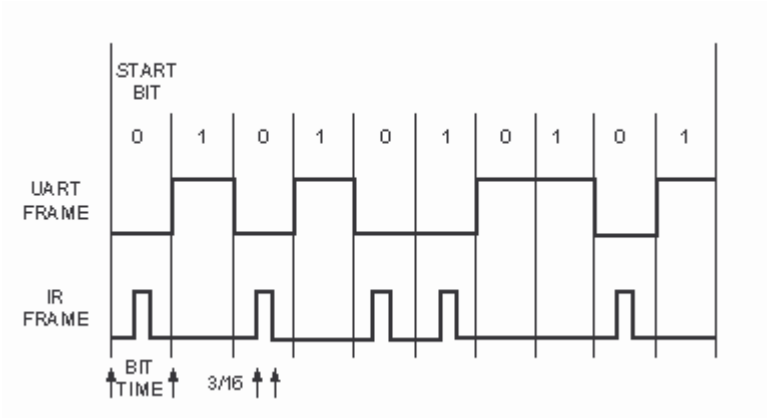

Figure 36. IrDA 3/16 Data Modulation.

A.3.2. 576 kbps and 1.152 Mbps Data Rate

IrDA Physical Layer Specifications also support the intermediate data rates of 576 kbps and 1.152 Mbps. These speeds use an RZI modulation similar to the 3/16 modulation used at 115.2 kbps and slower, but use a nominal 25% pulse width.

A.3.3. 4 Mbps Data Rate

Beginning with Version 1.1 of IrDA physical layer specification, a 4 Mbps data rate is supported. IrLAP specification requires all links to begin negotiation at 9.6 kbps and then negotiate to higher data rates, if supported at both ends. Therefore all devices that support a 4 Mbps data rate will have to be capable of supporting a lower data rate of 9.6 kbps at the minimum. Thus a 4 Mbps device will be able to communicate with a device that only supports 9.6 kbps. This ensures backward compatibility.

A 4 Mbps IrDA link uses a modulation scheme known as 4 PPM (Pulse Position Modulation), instead of the 3/16 modulation used for slower data rates. With 4 PPM, information is transferred by the position of a pulse within a time slot. In the 4 PPM

scheme (Figure 37) two data bits are combined to form a 500 ns “data bit pair” (DBP). This DBP is divided into four 125 ns time slots or “chips”. The two bits to be encoded will have one of four states 00, 01, 10, 11. Depending upon which of these states is present, a single pulse is placed in either first, second, third or forth 125 ns time slot. Thus a demodulator, after phase locking on the incoming bit stream, can determine the data pattern by the location of the pulse within the 500 ns period. The demodulator phase locks with a string “preamble” field. A preamble consists of 16 bits, and is transmitted 16 times.

Bit Pattern	Pulse Position
00	1000
01	0100
10	0010
11	0001

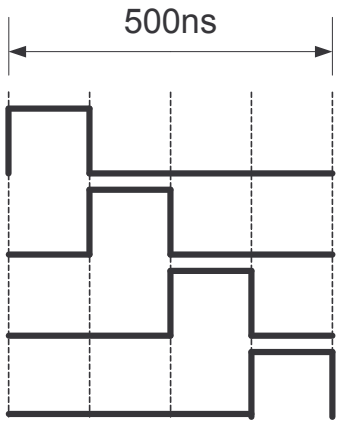

Figure 37. 4PPM Modulation.

The block diagram for the Version 1.1 (4 Mbps) Physical Layer (Figure 38) looks similar to the Version 1.0 block diagram, except that the UART and the Encode/ Decode circuitry are replaced with an I/O device that is designed for 4 Mbps IrDA data communication. This device performs the encoding and decoding of both the 3/16 and the 4 PPM modulation.

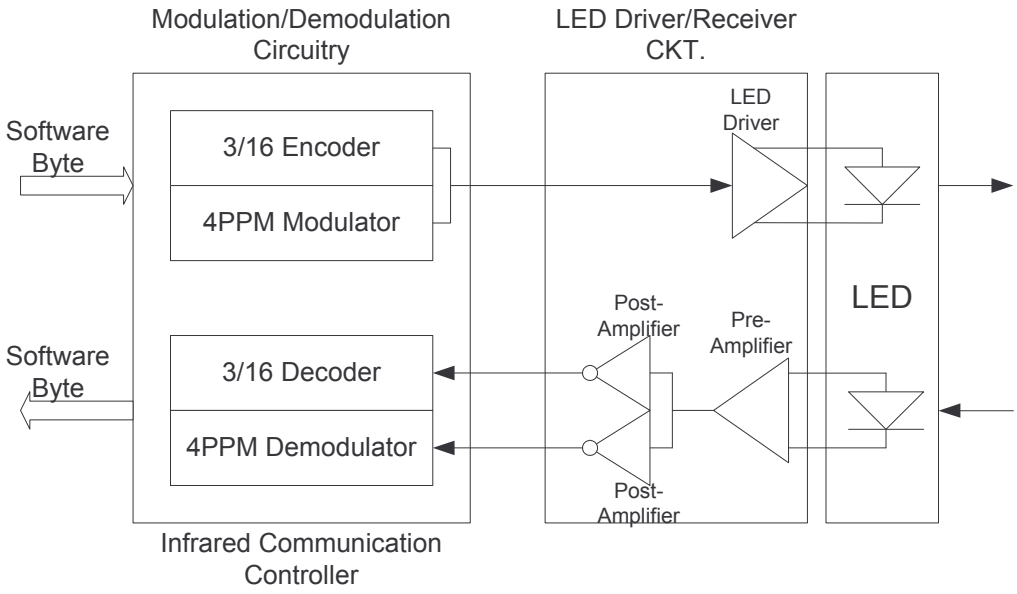

Figure 38. IrDA Version 1.1 Physical Layer Block Diagram.

A.3.4. Signaling Rate and Pulse Duration

An IrDA serial infrared interface must operate at 9.6 kb/second. Additional allowable rates listed below are optional. Signaling rate and pulse duration specifications are shown in Table 2.

Table 2. Signaling Rate and Pulse Duration Specifications

Signaling Rate	Modulation	Rate Tolerance % of Rate	Pulse Duration Minimum	Pulse Duration Nominal	Pulse Duration Maximum
2.4 kb/s	RZI	+/- 0.87	1.41 us	78.13 us	88.55 us
9.6 kb/s	RZI	+/- 0.87	1.41 us	19.53 us	22.13 us
19.2 kb/s	RZI	+/- 0.87	1.41 us	9.77 us	11.07 us
38.4 kb/s	RZI	+/- 0.87	1.41 us	4.88 us	5.96 us
57.6 kb/s	RZI	+/- 0.87	1.41 us	3.26 us	4.34 us
115.2 kb/s	RZI	+/- 0.87	1.41 us	1.63 us	2.23 us
0.576 Mb/s	RZI	+/- 0.1	295.2 ns	434.0 ns	520.8 ns
1.152 Mb/s	RZI	+/-0.1	147.6 ns	217.0 ns	260.4 ns

4.0 Mb/s (single pulse) (double pulse)	4PPM	+/-0.01	115.0 ns	125.0 ns	135.0 ns
	4PPM	+/-0.01	240.0 ns	250.0 ns	260.0 ns

A.3.5. Key Physical Layer Parameters

IrDA physical layer specification defines the requirements for a serial, half-duplex IR link that will communicate with another IrDA device at distances from 0 to 100 cm (standard power device). The key physical layer parameters are shown in Table 3.

Note: More information may be obtained from IrDA Serial Infrared Physical Layer Specification (currently Version 1.3).

Table 3. Standard Power Key Physical Layer Parameters

ACTIVE OUTPUT (TRANSMITTER) SPECIFICATIONS	Data Rates	Minimum	Maximum
Peak Wavelength, Up, um	All	0.85	0.90
Maximum Intensity In Angular Range, mW/Sr	All	-	500
Minimum Intensity In Angular Range, mW/Sr	115.2 kb/s & below	40	-
	Above 115.2 kb/s	100	-
Half-Angle, degrees	All	+/-15	+/-30
Signaling Rate (also called Clock Accuracy)	All	See Table 2	See Table 2
Rise Time Tr & Fall Time Tf, 10-90% , ns	115.2 kb/s & below	-	600
	Above 115.2 kb/s	-	40
Optical Over Shoot, %	All	-	25
Pulse Duration	All	See Table 2	See Table 2
Edge Jitter, % of nominal bit duration	115.2 kb/s & below	-	+/-2.3

Jitter Relative to Reference Clock, % of nominal bit duration	0.576 &1.152 Mb/s	-	+/-2.9
Edge Jitter, % of nominal chip duration	4.0 Mb/s	-	+/-4.0
ACTIVE INPUT (RECEIVER) SPECIFICATIONS			
Maximum Irradiance In Angular Range, mW/cm^2	All	-	500
Minimum Irradiance In Angular Range, uW/cm^2	115.2 kb/s & below	4.0	-
	Above 115.2 kb/s	10.0	-
Half-Angle, degrees	All	+/-15	-
Receiver Latency Allowance, ms	All	-	10
LINK INTERFACE SPECIFICATIONS			
Signaling Rate (also called Clock Accuracy)	All	See Table 2	See Table 2
Minimum Link Length, m	All	0	0
Maximum Link Length, m	All	1	-
Bit Error Ratio, BER	All	-	10^-8
Receiver Latency Allowance, ms	All	-	10

A.3.6. Optical Requirements

Figure 39 and Figure 40 show IrDA requirements for output intensity and input irradiance vs. angle. For the output (Figure 39), the intensity at any point within a cone of half angle 15° with respect to the optical axis, must fall between the minimum and the maximum values. The intensity at any point outside of a cone greater than 30° with respect to the optical axis, must fall below the minimum value.

For the optical input (Figure 40), the receiver must be able to recognize a signal between the minimum irradiance (depending upon data rate) and the maximum of 500 mW/cm2, at any point within a cone of 15° with respect to the optical axis.

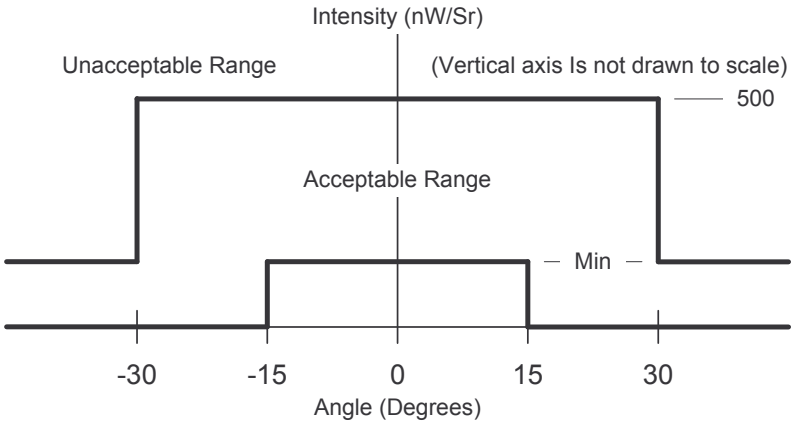

Figure 39. Acceptable Optical Output Intensity Range.

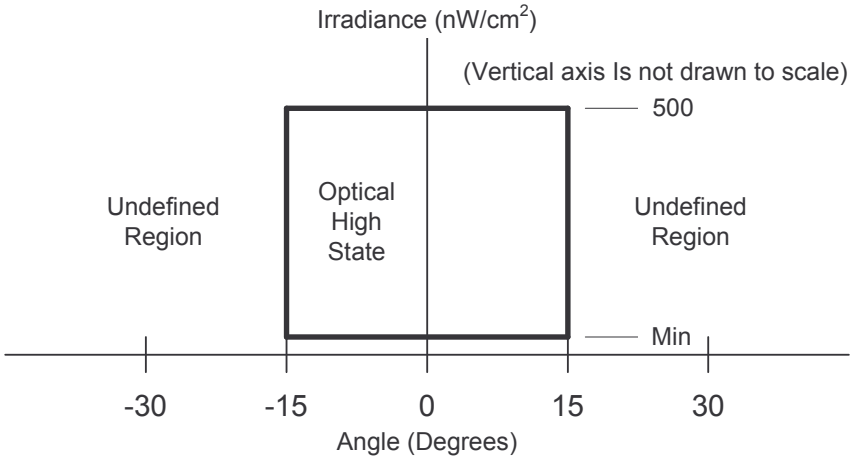

Figure 40. Optical High State Range.

A.3.7. Low Power Option

The low power option specification was added to IrDA 1.2 physical layer specification to cater to Telecom applications, where low power operation is more important than the link distance. Under the low power option, the minimum link distance when two low power option devices communicate is 0 to 20 cm. However, when a standard power device communicates with a low power option device, the minimum link distance range is increased to 0 to 30 cm. Note that under version 1.2,

the maximum supported data rate is 115 kbps.

Version 1.3 of IrDA Physical Layer specification extends the low power option to all data rates up to the maximum of 4 Mbps.

Table 4 shows the key physical parameters for the low power option.

Table 5 gives a quick comparison of physical layer specifications for standard power (IrPHY 1.1), SIR low power option (IrPHY 1.2) and FIR low power option (IrPHY 1.3).

Table 4. Key Low Power Option Physical Parameters

	Applicable Data Rates	Min.	Max.
Active Output (Transmitter) Specifications			
Intensity in Angular Range, mW/Sr	All	-	72
	115.2 kbps and below	3.6	-
	Above 115.2 kbps	9	-
Active Input (Receiver) Specifications			
Irradiance in Angular Range, mW/cm2	All Speed	-	500
	115.2 kbps and below	0.009	-
	Above 115.2 kbps	0.0225	-
Receiver Latency Allowance, ms	All	-	0.5

Table 5. Comparison of key parameters between IrPHY 1.1 and 1.3

	SIR/FIR Std. Power	FIR Low Power	Remarks

	(IrPHY 1.1)	(IrPHY 1.3)	
Link distance, cm			
Lower limit	0	0	
Upper limit (low power to low power)	-	20	} Range per Telecom
Upper limit (low power to std power)	-	30	} SIG use model
Upper limit (std power to std power)	10	-	
Data rate, bps			
Minimum	9.6K	9.6K	
Maximum	4M	4M	
Intensity, mW/Sr			
Minimum (≦115.2 kbps)	40	3.6	IrPHY 1.2/1.3 uses <10% of std power LED drive current, allows small eye safe devices
Minimum (>115.2 kbps)	100	9	
Maximum (all data rates)	500	72	
Irradiance			
Minimum, μW/cm2 (≦115.2 kbps)	4	9	IrPHY 1.2/1.3 allows use of less sensitive receiver, hence higher minimum irradiance
Minimum, μW/cm2 (>115.2 kbps)	10	22.5	
Maximum, mW/cm2 (all data rates)	500	500	
Latency, ms	10		Lower latency required to prevent voice clipping

A.3.8. Half Duplex and Latency

IrDA link is half-duplex, and there is a time delay allowed from the time a link stops transmitting to the time when it must be ready to receive. IrDA link cannot send and receive at the same time because the transmitter and receiver are not optically isolated,

and the transmitted signal can interfere with the incoming signal. When the transmitter is emitting light it may even saturate its own receiver, and disable it from receiving data from another source. IrDA specifications allow a period of 10 ms (Standard Power Option) after transmitting, for the receiver to regain its full sensitivity. Shorter times may be negotiated when the link starts up. This delay, from the time the transmitter stops sending light pulses to the time the receiver is guaranteed to be ready to receive data, is termed latency. Latency is also known as receiver setup time.

A.3.9. Ambient Light

There are requirements for ambient light rejection to ensure proper working of IrDA datalink under a wide range of environmental conditions. IrDA specification specifies the test methods for measuring the data integrity of the link under electromagnetic fields, sunlight, incandescent lighting and fluorescent lighting. An IrDA receiver must be able to reject up to 10K lux of sunlight, 1K lux of fluorescent light and 1K lux of incandescent light. These values were chosen as typical of what may be encountered under normal use conditions. Please refer to IrDA Physical Layer Test Specification for test methodologies.

A.4. IrLAP - LINK ACCESS PROTOCOL

IrLAP protocol specification corresponds to the OSI layer 2 (Data Link Protocol), and is a mandatory layer for IrDA protocols. IrLAP is based on the pre-existing HDLC and SDLC halfduplex protocols, with some modifications to cater to the unique features and requirements of infrared communications. IrLAP provides guidelines for

the software which looks for other machines to connect to (sniff), discovers other machines (discover), resolves addressing conflicts, initiates a connection, transfers data and cleanly disconnects. IrLAP specifies the frame and byte structure of IR packets as well as the error detection methodology for IR communications. Figure 41 shows the block diagram for IrDA IrLAP function.

IrLAP defines three different framing schemes corresponding to the three types of data rates (9.6 kbps - 115.2 kbps, 0.576 Mbps and 1.152 Mbps, 4 Mbps). The wrapper types for the three physical layer schemes are:

- Asynchronous (ASYNC) Framing (9.6 kbps - 115.2 kbps)
- Synchronous (SYNC) HDLC Framing (576 kbps and 1.152 Mbps)
- Synchronous 4 PPM Framing (4 Mbps)

Figure 42 shows the three different types of frame structures. IrLAP Payload data includes the address, control field and information data. To implement IrLAP layer, please refer to IrDA IrLAP specification.

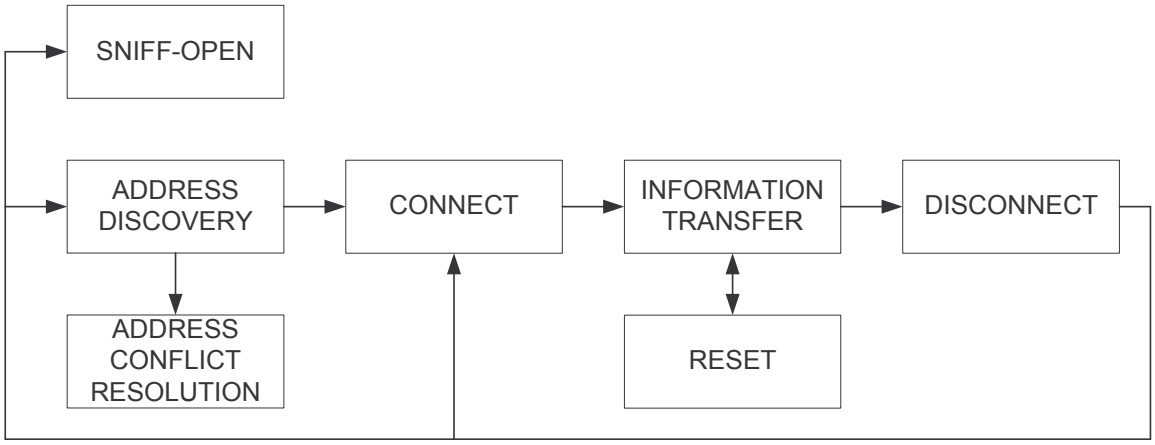

Figure 41. IrLAP Block Diagram.

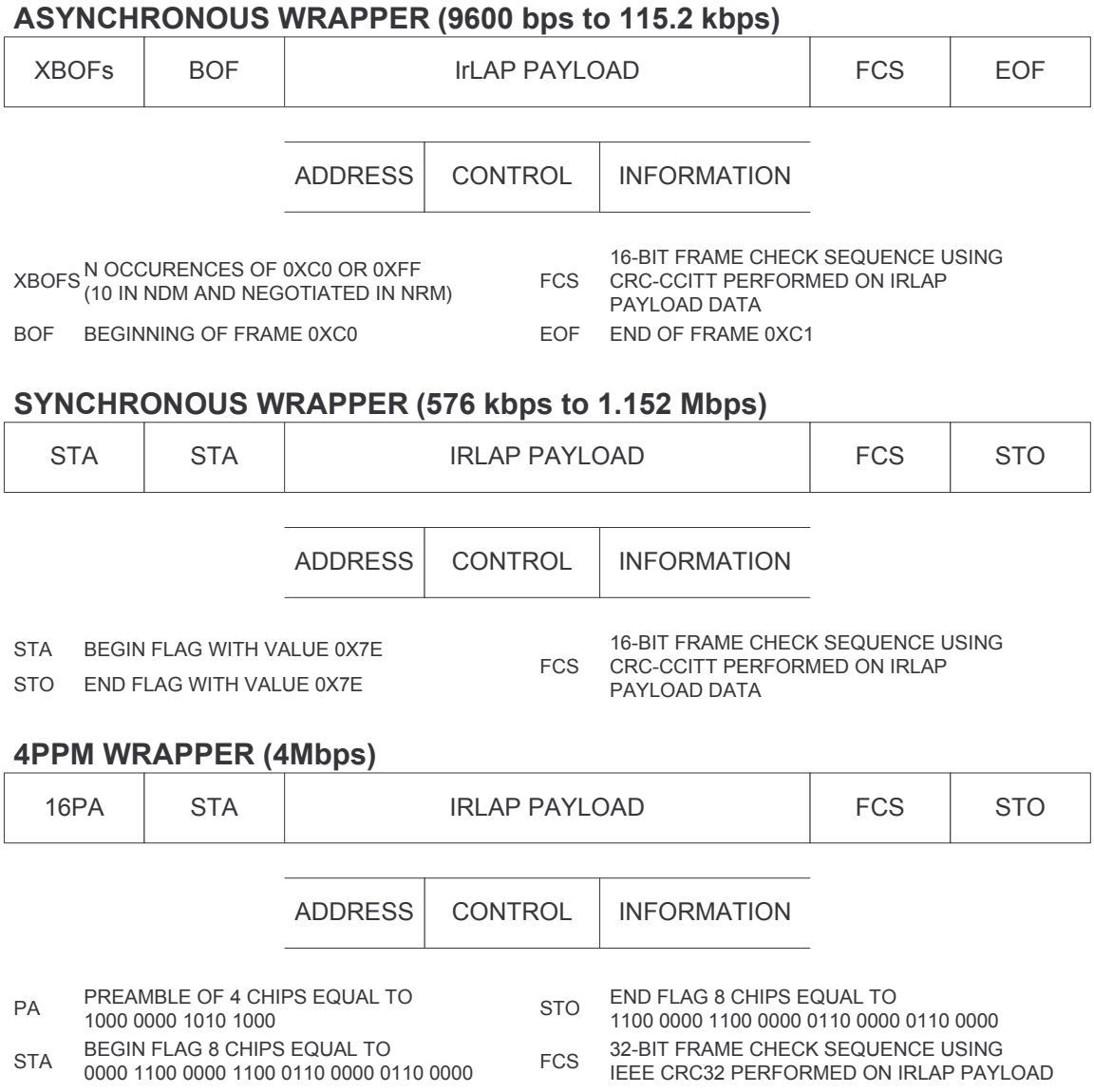

Figure 42. IrLAP Frame Structure.

A.5. IrLMP PROTOCOL

IrLMP (Link Management Protocol) is a layer that sits above IrLAP layer. It provides services to both the Transport layer and directly to the application layer. IrLMP consists of two components, LM-IAS (Information Access Service) and LM-MUX (Link Management Multiplexer). LM-IAS maintains a database of IrDA devices discovered as well as providing information on what services IrDA compliant devices

offer. LM-MUX provides services to both the local LM-IAS and also to the transport entities or applications that bind directly to the LM-MUX layer. LM-MUX provides a mechanism for linking multiple devices over IrLAP, as well as sharing control of a single IrLAP connection between a pair of stations.

Figure 43 shows the Link Management Architecture. IrLMP adds a two-byte header to IrLAP frame, as shown in Figure 44.

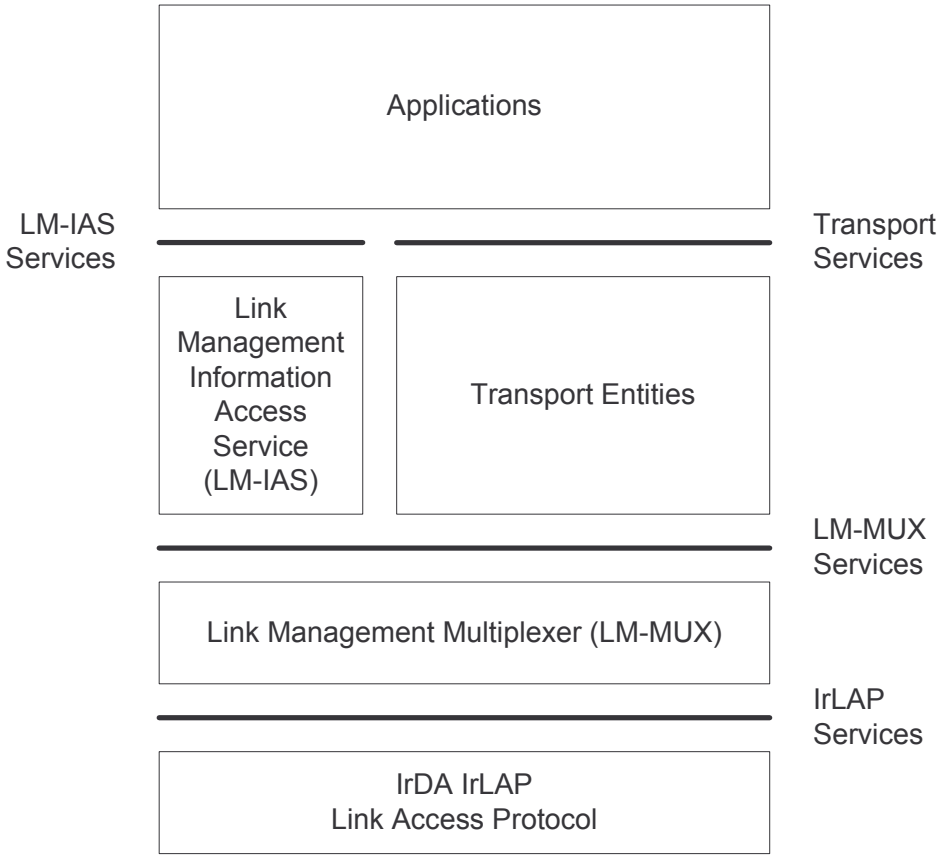

Figure 43. IrDA Link Management Architecture.

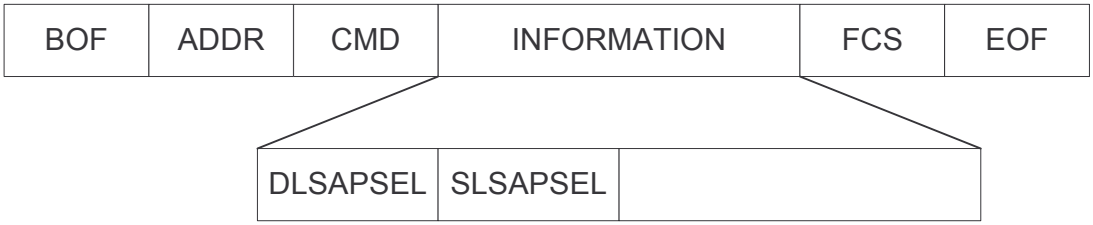

Figure 44. IrDA LMP Frame Structure.

Reference

[1] IrDA Specifications and Technical Notes. [Online]. Available: http://www.irda.org/displaycommon.cfm?an=1&subarticlenbr=7

[2] Red Hat Linux 9.0. [Online]. Available: ftp://linux.nctu.edu.tw/iso/redhat/9/i386/

[3] MySQL 3.23. [Online]. Available: http://dev.mysql.com/downloads/mysql/3.23.html

[4] Apache HTTP Server 1.3. [Online]. Available: http://httpd.apache.org/download.cgi

[5] PHP 4.3.8. [Online]. Available: http://www.php.net/downloads.php

[6] SNDS100 Board Specification. [Online]. Available: http://www.samsung.com/Products/Semiconductor/SystemLSI/Networks/PersonalNTASSP/CommunicationProcessor/S3C4510B/snds100.pdf

[7] ARM documentation of boards and firmware. [Online]. Available: http://www.arm.com/documentation/Boards_and_Firmware/index.html

[8] Xilinx Virtex E FPGA Data Sheets. [Online]. Available: http://www.xilinx.com/xlnx/xweb/xil_publications_display.jsp?sGlobalNavPick=&sSecondaryNavPick=&category=-18776&iLanguageID=1

[9] COMPAQ(HP) iPAQ h5555 Pocket PC. [Online]. Available: http://www.hp.com/country/us/en/prodserv/handheld.html

[10] uClinux. [Online]. Available: http://www.uclinux.org

[11] Existing IrDA Protocol Implementations. [Online]. Available: http://tuxmobil.org/Infrared-HOWTO/infrared-howto-s-irda-protocols.html#id2887275

[12] PHPMyAdmin. [Online]. Available http://sourceforge.net/projects/phpmyadmin/

[13] Windows Mobile-based Pocket PCs Homepage. [Online]. Available: http://www.microsoft.com/windowsmobile/pocketpc/ppc/default.mspx

[14] 林家弘，『一全數位式紅外線傳收系統之設計與 FPGA 實現』，國立交通大學，碩士論文，民國 92 年

[15] Jacob Gorban, IrDA Core Specification, [Online]. Available: http://www.opencores.org/projects.cgi/web/irda/overview

[16] Bob Zeidman, "*Designing with FPGAs and CPLDs"*, Kansas: CMP books, 2002.

[17] K.C. Chang, "*Digital Systems Design with VHDL and Synthesis"*, IEEE Computer Society, 1999.

[18] Douglas E. Comer and David L. Stevens, *"Internetworking With TCP/IP"*, New

Jersey: Prentice-Hall, 1996.