

Development of a low power hand-held device in a low budget manner

Master thesis in Electronic Systems

by

Anders Kagerin, Michael Karlsson

LiTH - ISY - EX - - 06 / 3797 - - SE

Linköping 18 March 2006

Development of a low power hand-held device in a low budget manner

Master thesis in Electronic Systems

by

Anders Kagerin, Michael Karlsson

LiTH - ISY - EX - - 06 / 3797 - - SE

Supervisor: **Kent Palmkvist**,
Department of Electrical Engineering, Linköpings Universitet

Examiner: **Kent Palmkvist**,
Department of Electrical Engineering, Linköpings Universitet

Linköping: **18 March 2006**



LINKÖPINGS UNIVERSITET

Avdelning, Institution

Division, Department

Institutionen för Systemteknik
581 83 LINKÖPING
SWEDEN

Datum

Date

18 March 2006

Språk

Language

- Svenska/Swedish
 Engelska/English

Rapporttyp

Report category

- Licentiatavhandling
 Examensarbete
 C-uppsats
 D-uppsats
 Övrig rapport

ISBN

ISRN

LiTH - ISY - EX - - 06 / 3797 - - SE

Serietitel och serienummer ISSN

Title of series, numbering

URL till elektronisk version

URL to electronic version

<http://www.es.isy.liu.se>
<http://www.diva-portal.org/liu/>

Titel

Title

Development of a low power hand-held device in a low budget manner

Författare

Author

Anders Kagerin, Michael Karlsson

Sammanfattning

Abstract

The market of portable digital audio players (DAPs) have literally exploded the last couple of years. Other markets has grown as well. PDAs, GPS receivers, mobile phones, and so on. This resulted in more advanced ICs and SoCs becoming publically available, eliminating the need for in-house ASICs, thus enabling smaller actors to enter the markets.

This thesis explores the possibilities of developing a low power, hand-held device on a very limited budget and strict time scale.

This thesis report also covers all the steps taken in the development procedure.

Nyckelord

Keyword

Embedded, Hand-held, Device, Codec, Decoder, MP3, Ogg, WMA, FLAC, Development, Hardware, Software

Abstract

The market of portable digital audio players (DAPs) have literally exploded the last couple of years. Other markets has grown as well. PDAs, GPS receivers, mobile phones, and so on. This resulted in more advanced ICs and SoCs becoming publically available, eliminating the need for in-house ASICs, thus enableing smaller actors to enter the markets.

This thesis explores the possibilities of developing a low power, hand-held device on a very limited budget and strict time scale.

This thesis report also covers all the steps taken in the development procedure.

Keywords: Embedded, Hand-held, Device, Codec, Decoder, MP3, Ogg, WMA, FLAC, Development, Hardware, Software

Acknowledgements

We would like to thank our supervisor and examiner Kent Palmkvist and all the staff at ISY for all the help and support they have provided.

We would also like to thank our families, and our friends who have proof-read this thesis.

Finally we would like to thank all the Rockbox developers, especially Linus Feltzing Nielsen for all the help on getting the development environment working.

Last, but not least, we want to pay our tribute to the open source community.

Nomenclature

Most of the reoccurring abbreviations and symbols are described here.

Notation

In this thesis kilobytes (kB) means 1024 bytes, megabytes (MB) means $1024 * 1024$ bytes, and so on.

Signals are written as *SIG* or, should the signal be used as active-low, \overline{SIG}

Symbols

f_s	Sample frequency.
f_{clk}	CPU frequency.
f_{bus}	System bus frequency.
V_{dd}, V_{cc}	Supply voltages.

Terminology

Active-low	The associated function is performed when it is logical 0.
Bootloader	A program that loads and executes the operating system.
Codec	In the context of hardware: refers to CODEC for digital/analog conversion. In the context of software: refers to a program that encode or decode compressed audio data.
Driver	A collection of routines for hardware control and abstraction.
Firmware	Software in embedded systems.
Operating system	The main software that handles hardware and programs.

Abbreviations

AAC	Advanced audio coding, audio compression standard
ADC	Analog to digital converter
ARM	Advanced RISC machine, processor family
ASIC	Application specific IC
ASIP	Application specific processor
ATA	AT Attachment, also known as IDE
BDM	Background debug mode, debug protocol
BGA	Ball grid array, chip package
BSD	Berkeley software distribution, software licence
CODEC	Coder/decoder
CPU	Central processing unit
DAC	Digital to analog converter
DAP	Digital audio player
DC	Direct current
DSP	Digital signal processor
EEPROM	Electrically erasable programmable read-only memory
FAT32	File allocation table 32 bit version, data storage format
FM	Frequency modulation
FSF	Free software foundation
GCC	GNU Compiler Collection
GDB	GNU Project Debugger
GNU	GNU's not Unix, collection of software
GPIO	General purpose input/output
GPL	General public licence, software licence
GUI	Graphical user interface
HDD	Hard disk drive
I ² C	Inter-IC, communication protocol
I ² S	Inter-IC sound, communication protocol
IC	Integrated circuit
IDE	Integrated drive electronics, communication protocol
IEC	International electrotechnical commission, standardization group
IEEE	Institute of electrical and electronics engineers, inc., standardization group
LCD	Liquid crystal display, monitor technology
LED	Light emitting diode

m68k	Motorola 68000, processor family
MCU	Micro controller unit
mil	Mili-inch, 1000th of an inch
MP3	MPEG-I Layer 3, audio compression standard
MPEG	Motion picture encoding group
NAND	Not and, logical operations
NDA	Non-disclosure agreement
PCB	Printed circuit board
PHY	Physical layer
PMU	Power management unit
QFN	Quad footprint no-lead, chip package
QFP	Quad footprint, chip package
RAM	Random access memory
RDS	Radio data system, data protocol for radio
RISC	Reduced instruction set computer, processor category
RMS	Root mean square, mathematical formulae
RS232	Recommended standard 232, communication protocol
SDK	Software development kit
SDRAM	Synchronous dynamic RAM
SNR	Signal to noise ratio
SO	Small outline, chip package
SoC	System on chip
SPI	Serial peripheral interface, communication protocol
SRAM	Static RAM
TFT	Thin film transistor, monitor technology
THD	Total harmonic distortion
UART	Universal asynchronous receiver/transmitter, communication protocol
USB	Universal serial bus, communication protocol

Contents

1	Introduction	1
1.1	Background	1
1.2	Purpose	1
1.3	Realisation	2
1.4	Limitation	2
1.5	Requirements	2
1.6	Topics covered	3
2	Audio coding & sound quality	5
2.1	Lossless coding	5
2.2	Lossy coding	5
2.2.1	MP3, MPEG-1 Layer 3	6
2.2.2	Ogg/Vorbis	6
2.3	Sound quality	6
2.3.1	SNR, Signal-to-noise ratio	7
2.3.2	THD, Total Harmonic Distortion	7
3	Component selection	9
3.1	Audio processor	9
3.1.1	Philips Nexperia	9
3.1.2	PortalPlayer	9
3.1.3	SigmaTel D-Major STMP35xx	10
3.1.4	Freescale ColdFire MCF5249	11
3.1.5	Conclusion	12
3.2	USB interface	13
3.2.1	FTDI Chip FT245R & FT232R	13
3.2.2	Philips ISP1582	13
3.2.3	Cypress EZ-USB AT2LP	13
3.2.4	Conclusion	13
3.3	Audio codec	14
3.3.1	Wolfson WM8750L	14
3.3.2	Philips UDA1380	15

3.3.3	Conclusion	15
3.4	External memory	15
3.4.1	File storage	15
3.4.2	RAM	17
3.5	FM tuner	17
3.6	Display	18
3.7	Power supplies	18
3.8	Battery charger	19
3.9	Bluetooth	19
4	Hardware design	21
4.1	Component packages	21
4.1.1	BGA, Ball grid array	21
4.1.2	QFP & SO, Quad footprint & Small outline	22
4.1.3	QFN, Quad footprint no-lead	22
4.1.4	Conclusion	22
4.2	Interconnections	22
4.2.1	I ² C, Inter-IC	22
4.2.2	SPI, Serial peripheral interface	23
4.2.3	I ² S, Inter-IC sound	23
4.2.4	IEC958	23
4.2.5	UART	23
4.2.6	System bus	24
4.2.7	BDM, Background debug mode	26
4.2.8	GPIO, General purpose input/output	26
4.3	Peripheral components	26
4.3.1	Passive components	26
4.3.2	Active components	27
4.3.3	Electro-mechanical components	27
4.4	PCB design	27
4.4.1	Cost vs Functionality	28
4.4.2	PCB manufacturers	28
4.4.3	Conclusion	29
5	Software	31
5.1	Firmware	31
5.1.1	Rockbox	32
5.1.2	Bootloader	33
5.2	Development environment	33
5.2.1	GCC & Binutils	34
5.2.2	GDB	34
5.2.3	BDM software	34

6	Problems and mishaps	37
6.1	Manufacturers	37
6.2	Development	39
7	Results and future work	41
7.1	Results	41
7.1.1	Power consumption	41
7.2	Future work	42
7.2.1	Hardware	42
7.2.2	Software	43
A	GNU General Public License	47

Chapter 1

Introduction

This text is written as a master of science final thesis at Linköpings Universitet by Anders Kagerin and Michael Karlsson with Kent Palmkvist as supervisor and examiner. The document is written in \LaTeX in 2006.

In this first chapter some background about embedded systems, SoCs and their availability will be given. Also some main questions this project seeks to answer will be formulated, and topics covered will be described.

1.1 Background

The use of digital audio and its distributions over the Internet have become more and more common the last years. This have made the market of portable digital audio players (DAPs) literally explode.

Other markets has grown as well. PDAs, GPS receivers, mobile phones, and so on. This resulted in more advanced ICs and SoCs becoming publically available, eliminating the need for in-house ASICs, thus enableing smaller actors to enter the markets.

However, the development effort can still be very large, even through the use of highly integrated systems. Many design features need to be decided upon and many development tasks are time exhausting.

1.2 Purpose

Considering the availability of ICs and SoCs would it be possible to develop a small low power hand-held device on a very limited budget and time scale?

The purpose of this thesis is to try to develop an embedded system on a low budget and in that way evaluate the possibilities of

doing so on a low budget and a very limited time scale.

Different kinds of problems may occur while developing such a system because of the low budget. These problems has to be considered too.

Things as power consumption, power management and energy saving needs to be considered during the development as well.

Is it possible to get a fully functional device in less than 40 man weeks? What are the main difficulties in developing a system like this?

1.3 Realisation

The development of an embedded device involves many tasks, these are usually divided into hardware vs. software. Hardware tasks include architecture planning, selection and evaluation of ICs, and PCB layout. Software tasks include architecture design, implementation and testing.

All of these tasks are highly interdependent, where design decisions and tradeoffs have to be made based on several factors, such as size, performance, power, compatibility and project resources.

1.4 Limitation

This thesis will focus on developing a working prototype. Many optimizations will be done, but due to time limitations many optimizations are left outside of this thesis.

1.5 Requirements

The basic requirements of the system is the following:

Requirement 1: It should be able to run on a battery.

Requirement 2: It should be able to start a program from a non-volatile memory.

Requirement 3: It should be able to output audio to earphones.

Requirement 4: It should have a display.

Requirement 5: The software should be upgradeable.

Requirement 6: It should be able to decode at least two common lossy audio format.

Except for the basics requirements some goals are set, these are:

Goal 1: It should be able to act as a USB 2.0 mass storage device.

Goal 2: It should have Bluetooth connectivity.

Goal 3: It should include an RDS capable FM tuner.

1.6 Topics covered

There are seven chapters (including this introduction) and one appendix. Main topics dealt with are:

Chapter 2: Will give some basics about audio coding and sound quality.

Chapter 3: Presentation of the component selection.

Chapter 4: Here the hardware design procedure will be presented.

Chapter 5: Presentation of the software integration.

Chapter 6: Will give a summary of problems and mishaps during the project.

Chapter 7: Summary of the project results, and suggestions of future work to improve the player.

Appendix A: The full GNU General Public License text.

Chapter 2

Audio coding & sound quality

This chapter covers the basics in audio coding. It's not going to be a detailed guide on how to code audio data, but rather a short introduction to different coding techniques. This is to have a basic idea of what the device should be doing. For a more detailed introduction to audio coding see [1]. This chapter will also be a brief introduction to measures of sound quality.

2.1 Lossless coding

Lossless coding is one principle of audio coding. Lossless compression algorithms produce compressed data that can be decoded to output that is identical to the original, according to [2]. Due to the complexity of audio it is hard to get good compression rates using lossless coding.

For use in audio applications there are a couple of lossless formats, FLAC among others.

2.2 Lossy coding

Lossy encoding algorithms discards data to gain good compression rates. In opposite to lossless coding the original data cannot be perfectly restored from the compressed data.

Since good compression rates are possible to get, lossy coding are often used in audio coding. The basic idea is to try to avoid coding sound that are outside of what can be heard by the human ear.

Examples of lossy audio formats are MP3, Ogg/Vorbis and AAC.

2.2.1 MP3, MPEG-1 Layer 3

MP3 is the most commonly used lossy audio format today. The MP3 format uses technology patented by Fraunhof Group and Thomson Inc. The distributors of every MP3 encoder and decoder have to pay a compulsory license fee to be allowed to use those patented technologies. This is to be considered when developing a DAP. Not only manufacturers of encoders and decoders have to pay fees. Artists, record companies, and so on, also have to pay fees to be allowed to distribute material coded in the MP3 format.

The MP3 format is a part of the MPEG-1 standard. There are specifications of how the encoder should encode the signal, as well as how the decoder should decode it.

2.2.2 Ogg/Vorbis

There have been forces in the Free Software community to develop a format that should be as least as good as MP3, but it should be free to use in any manner. Ogg/Vorbis was to be that format. Ogg/Vorbis is a patent and royalty free audio compression format developed by the Xiph.org Foundation. Since the format is royalty free nobody has to pay anything to use the format or to implement encoders and decoders.

The Ogg/Vorbis format does not specify how to encode an Ogg/Vorbis stream. The specification only tells how to implement the decoder. All streams that can be correctly decoded in such a decoder are to be considered as Ogg/Vorbis streams. This gives the opportunity to let encoders use any technique to encode the stream. In this way the format isn't stuck with a technique if better techniques arise, as would be the case with MP3.

Decoding an Ogg/Vorbis stream is more demanding than MP3 decoding, in terms of computing time. But at the other hand the Ogg/Vorbis format have a lower transparency. The transparency is a measure of the bit rate needed so that the human ear cannot hear the difference between the decoded signal and the original uncompressed signal.

2.3 Sound quality

In this section the methods of measuring sound quality will be discussed. Focus will be on two measurements that have been chosen since they are the most common.

2.3.1 SNR, Signal-to-noise ratio

The signal-to-noise ratio is a term of the power ratio between a signal (meaningful information) and the background noise. The SNR is defined as:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

Since many signals have a very wide dynamic range, SNR is usually expressed in terms of the logarithmic decibel scale:

$$SNR(dB) = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) = 20 \log_{10} \left(\frac{A_{signal}}{A_{noise}} \right)$$

P is average power and A is RMS amplitude. Important to note is that signal and noise power are measured only within the system bandwidth.

SNR measures the ratio between noise and an *arbitrary* signal on the channel, not necessarily the most powerful signal possible. Hence, measuring SNR requires the selection of a representative signal.

SNR is usually taken to indicate an average signal to noise ratio. It is possible that instantaneous SNR values are considerably different. Higher signal to noise is better, i.e. cleaner sound.

For more information about the SNR measurement, see [3].

2.3.2 THD, Total Harmonic Distortion

The defining properties of any linear and time-invariant system are, of course, linearity and time invariance:

- **Linearity** means that the relationship between the input and the output of the system satisfies the scaling and superposition properties.
- **Time invariance** means that whether an input is applied to the system now or t seconds from now, the output will be identical, except a time delay of that t seconds.

In most cases the ideal transfer function is linear and time-invariant. However the real-life system is not ideal. Whenever the signal passes through the non-linear system additional content is added at the harmonics of the original frequencies.

The total harmonic distortion, or THD, is a measurement of the extent of that distortion.

The THD is defined as the sum of the powers of the harmonics above the fundamental frequency to the power of the fundamental:

$$THD = \frac{\sum \text{harmonics powers}}{\text{fundamental frequency power}}$$

or as:

$$THD = \frac{P_2 + P_3 + \dots + P_n}{P_1}$$

The THD is usually expressed in dB. A meaningful measurement must include the number of harmonics included (and should include other information about the test conditions). For more information about THD, see [3].

Chapter 3

Component selection

In this chapter the process of selecting components will be presented. The choices made are presented and which facts those choices were made upon.

3.1 Audio processor

To get the most possible flexibility out of the audio player in terms of supported file formats and upgrade possibilities there should not be any specific hardware decoder/encoder. Instead they should be software based. This leads to a number of choices.

In principle any DSP or MCU with enough capacity to encode and decode compressed audio data, in real-time, would suffice. A few different alternatives have been studied.

3.1.1 Philips Nexperia

The Philips Nexperia series of multimedia processors covers almost everything regarding audio and video decoding [4]. Unfortunately it was hard to get any specific details about their line of audio DSPs without signing Philips NDA. At this point that did not seem to be necessary since there were many others options regarding DSPs.

3.1.2 PortalPlayer

The PortalPlayer website [5] states that the PortalPlayer chip features dual ARM7 microprocessors with a zero wait state memory architecture delivering an equivalent of a 160 MHz ARM7.

According to [5] the chip has connectivity options for USB 2.0

PHY¹ and FireWire (IEEE 1934a). This does not mean that the chip has an on-chip USB 2.0 PHY, just that it can interface to one.

A feature that the chip does have on-chip, is a codec². This codec is built upon technology from Wolfson Microelectronics. Wolfson is known to build high quality audio codecs that can deliver high SNR.

The PortalPlayer series of audio processors is used in the iPod players from Apple.

3.1.3 SigmaTel D-Major STMP35xx

The D-Major from SigmaTel is a series of audio SoCs. The STMP35xx have an integrated USB 2.0 PHY, battery charger, 18 bits/channel audio codec, and more. For a more complete summary of features see [6].

The DSP core in the SigmaTel STMP35xx is a Motorola DSP56004 core with a filter coprocessor for FIR and IIR filters. The DSP core is specified to run at as most 75.6 MHz at $V_{dd} > 1.92$ V.

According to the data sheet [7] power dissipation is about 80 mW when running at 36 MHz (typical MP3 decoding in power save mode).

Software

SigmaTel supply a massive SDK with among others decoders for MP3 and WMA. The SDK also comes with a C compiler.

E-mail correspondence with a SigmaTel representative showed that the SDK license comes with a one time fee of \$12,000 and that SigmaTel do not support customers using third party software such as the GNU C compiler for Motorola DSP56k.

The SigmaTel D-Major pros and cons can be reviewed in table 3.1.

Pros	Cons
Much integrated hardware allowing for small designs. Good interfacing possibilities.	Very expensive SDK. No support from manufacturer if one does not buy the SDK.

Table 3.1: SigmaTel D-Major pros and cons

¹PHY (Physical Layer) is the lowest level of the implementation, i.e. the hardware.

²The codec contains the digital-to-analog converter and headphone amplifier.

3.1.4 Freescale ColdFire MCF5249

The ColdFire MCF5249 from Freescale Inc. [8] is a MCU based on the Motorola 68000 architecture. The MCF5249 is a variable instruction length RISC processor that incorporates a MAC unit which makes it suitable for signal processing. It also have good interfacing possibilities thanks to the I²C, Philips IIS (I²S) and its external parallel system bus.

The MCF5249 doesn't have any internal audio codecs, but thanks to the I²S bus it is easy to interface external codecs.

According to [8] the MCF5249 is capable of decoding a MP3 file at frequencies as low as 20 MHz, using only the on-chip SRAM. But since the file to be decoded needs to be stored, there is still need for an external memory. Thanks to the system bus this should not imply any significant problems.

Freescale [8] also states that the power dissipation is less than 1.3 mW/MHz. That means that MP3 decoding consumes about 26 mW. To be added to this value is power dissipation in the audio codec³.

Freescale technical support works well, they normally answers inquiries within 48 hours.

The MCF5249 is available in two different packages, 160-pin BGA or 144-pin QFP. See figure 3.1 for physical differences between the two package options. The QFP package is missing some of the pins, those pin are used for example in the flash media interface. For a more specific specification of what pins are missing, see the datasheet [10].

Software

There is a free⁴ firmware for the SCF5249 based iRiver MP3 players called Rockbox [9]. (According to [8] SCF5249 is a MCF5249 manufactured by Samsung rather than Motorola.)

Rockbox is released under the GNU General Public License (GPL). In short terms this means that the source code of the Rockbox project can be freely used and modified. The full license text can be found in Appendix A.

The Rockbox project website [9] states that the Rockbox firmware have functions for reading and writing a FAT32 filesystem. They

³Since external memory is needed always this does not need to be accounted for when comparing the processors.

⁴As defined by FSF ("Free as in speech, not beer"), often referred to as open source.

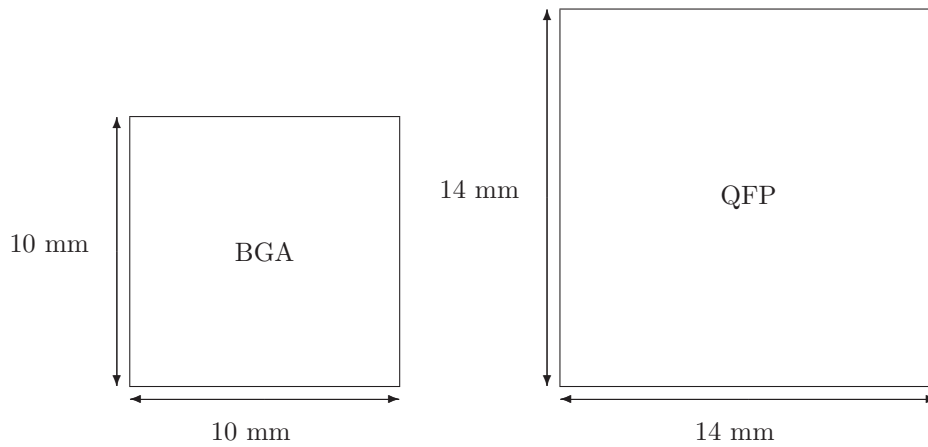


Figure 3.1: Physical size differences between BGA and QFP packages

also claim that they have a multi-codec architecture supporting decoding of among others MP3, Ogg/Vorbis and FLAC. It is also stated that the Rockbox firmware have a simple thread handler and fully functional memory management. This should be a solid code base to get started from. With this solution no codecs need to be implemented (nor any existing ported). Thus, the software coding part will consist more of device driver programming for the chosen peripheral devices.

The ColdFire MCF5249 pros and cons are summarized in table 3.2.

Pros	Cons
Good support from manufacturer. Existing open source software. Good interfacing possibilities. Easy to receive sample chips.	Needs many external components.

Table 3.2: ColdFire MCF5249 pros and cons

3.1.5 Conclusion

The SigmaTel chip offers small designs because of the integrated components such as USB 2.0 PHY and power management units. However their SDK fee of \$12,000 makes it impossible to use their chip in a low budget design.

The most lucrative option seems to be the ColdFire MCF5249 microcontroller from Freescale Inc. Offering good support, fast deliveries of samples and an existing open source DAP firmware, even though it is not from Freescale, this option seems better than the

PortalPlayer. The ColdFire processor does not come with an on-chip USB 2.0 PHY, but neither does the PortalPlayer. The loss in size of the design needing an external audio codec for the ColdFire is compensated by not needing to pay any fees for SDKs.

3.2 USB interface

3.2.1 FTDI Chip FT245R & FT232R

The FT245R & FT232R chips from Future Technology Devices International Ltd. (FTDI) [11] are made to be easy to use. The FT245R is a USB to parallel FIFO interface, and the FT232R is a USB to serial UART interface with optional clock generator output.

The simplicity of these chips however is their weakest side. There is no support for USB Mass Storage Class.

3.2.2 Philips ISP1582

According to the datasheet [12] the ISP1582 USB 2.0 controller from Philips Semiconductors fully complies with Universal Serial Bus Specification Rev.2.0, supporting data transfer at high-speed (480Mbit/s) and full-speed (12Mbit/s).

It connects through a parallel buss interface, hence it could be connected to the processor through the external system bus. This would also require some external logic for address decoding.

3.2.3 Cypress EZ-USB AT2LP

The AT2LP from Cypress Semiconductor is a fully featured USB 2.0 to ATA bridge supporting the USB 2.0 Mass Storage Class. It is a fixed function ATA controller that needs no special firmware to function.

The chip is able to set its data pins to high impedance, thus enabling it to share the IDE data bus with a processor or micro controller.

For more information about the AT2LP chip see the datasheet [13].

3.2.4 Conclusion

The FTDI chips are easy to use, but since they do not support USB Mass Storage Class they cannot be used in this design.

Both the ISP1582 from Philips and the AT2LP from Cypress can be used, both chips support USB 2.0 high-speed and full-speed. The ISP1582 does not directly support USB Mass Storage Class, but since it fully complies with USB 2.0 the Mass Storage Class can be implemented in software, however since the AT2LP already supports Mass Storage Class this will be the chip of choice. This allows development to be concentrated on other tasks since there is no need to implement the Mass Storage Class.

3.3 Audio codec

Since the chosen processor does not offer an on-chip audio codec there is a need for an external codec.

One of the most important things to consider, not to say the most important thing, when choosing a codec is sound quality. However, some tradeoffs may be done, for example in a portable device the power consumption is an important factor.

One other important thing is the interface. The ColdFire supports different standards for audio interfacing so this should not be a problem.

3.3.1 Wolfson WM8750L

Wolfson Microelectronics plc. is a United Kingdom based manufacturer known for its high quality audio codecs [15]. Offering a 24bits/channel DAC with a possibility to automatically over-sample the signal by a interpolation factor of 64 or 128 it can deliver a SNR of 98 dB(A) and a THD of -84 dB at 48 kHz, according to [16].

The codec have an internal headphone driver supporting capless⁵ mode of operation, eliminating the need for external DC decoupling capacitors at the output.

All parts of the codec can be shut down while not in use to reduce power consumption. When running at lowest power settings ($V_{dd} = 1.8V$) the codec consumes only 7 mW while still delivering a SNR of 90 dB(A).

The WM8750L has a feature that Wolfson calls *USB Mode*. When running in this mode, the codec is capable of generating all common sample rates from a USB application clock frequency. (For a more specific list of these sample rates, se the data sheet [16].) The *USB Mode* supports f_{clk} of 12 MHz to 24 MHz, these are common USB application clock rates.

⁵Some amplifiers needs DC decoupling capacitors to work.

Wolfson provides the possibility to order free samples from their website. However it was not possible to order samples of the WM8750L. After being in contact with a Wolfson sales office they said that the WM8750L were exclusive to some of their customers. However they had another chip, the WM8973, that was identical. When the sample chips arrived both the WM8750L and the WM8973 were included for some unknown reason.

3.3.2 Philips UDA1380

The Philips UDA1380 codec is commonly used in many MP3-players on the market [9]. The datasheet [17] states values for the SNR and THD of the codec, these are almost identical to the values of the WM8750L codec.

According to the datasheet the minimum power consumption during playback is a little more than 10 mW. The supply voltage must be in the range 2.4 - 3.6 volts.

3.3.3 Conclusion

The Wolfson WM8750L will be the codec of choice. The WM8750L was able to generate all common sample rates from USB clock rates. This allows the use of only one external clock circuit since the codec and CPU then can be run at the same frequencies as the USB chip.

3.4 External memory

There will always be a need for an external memory in these types of applications. Sometimes there is only need for some storage space, sometimes there might be need of some kind of extra RAM⁶.

3.4.1 File storage

The file storage media needs to be large, in terms of storage space, since the device should be able to store a lot of music. There are basically three different options available; memory card, flash memory or a hard disk drive. As a matter of fact memory cards are flash based so it should be enough to compare general flash memories to HDDs.

The memory card and flash memory options are still limited in storage size. The hard disk drive option allows for much greater storage space, but instead it requires more power. Due to the power

⁶For example as a data buffer in hard disk based players.

requirements, the hard disk drive option requires a more advanced power management.

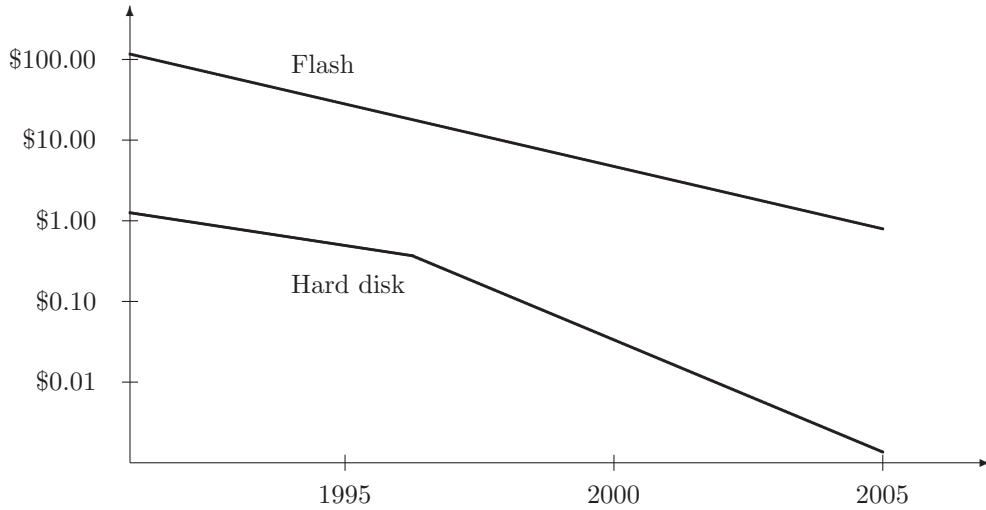


Figure 3.2: Price/MB (trends) [18]

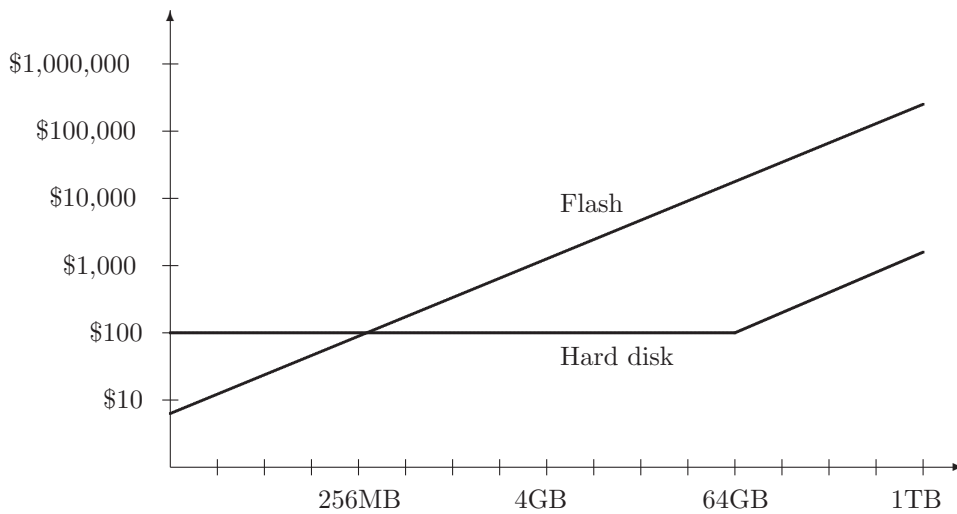


Figure 3.3: Total system costs as a function of total storage space [18]

As seen in figure 3.2 HDDs have always been much cheaper per megabyte than flash based memories, and they still are. It is clear from the chart in figure 3.3 that HDDs have a floor price. Small capacity flash memories are cheap, the cross over appear at approximately 256 MB. Since the MP3-players of today usually have at least 512 MB or 1GB of storage space, everything less than this would be unacceptable.

Since the processor has no support of booting directly of a HDD it needs some kind of boot memory, preferably a flash memory for the ability to upgrade the bootloader. It then seems like there are two possible solutions to the storage space issue; either one large flash memory or a HDD for storage space with a tiny flash based boot memory.

Conclusion

HDDs have the benefit of being much more cost effective than flash memories. Since at least 1 GB of storage space is wanted HDDs will be cheaper than flash, even though there is a need for a tiny boot flash memory.

Hence, a HDD was chosen for storage and a small flash memory as boot memory.

3.4.2 RAM

When using a HDD as storage media the HDD should not have to be powered on at all times. Therefore some kind of RAM to buffer the files, to allow the HDD to spin down, is needed.

To get high performance the software should run from a high speed memory, such as RAM, rather than from flash.

According to the MCF5249 data sheet[10] the internal RAM of the processor is no greater than a total of 96 kB. This is far less than enough for this application, especially when using a HDD.

SRAMs are expensive in large sizes, and since the MCF5249 have a built-in SDRAM controller it would be intuitive to use a cheaper SDRAM.

The MT48LC8M16A2 (2 Meg x 16 x 4 banks) module from Micron was selected since the Philips retailer, that contact already was established with, also was a Micron retailer and provided free samples.

3.5 FM tuner

Currently there are just two decent FM tuners with RDS capability available at the market. The TEA5764 from Philips Semiconductors and the Si4701 from Silicon Laboratories [19]. However, the Si4701 is covered by a NDA that is not compatible with the use of open source software. Hence the Philips tuner TEA5764 was to be used.

3.6 Display

Sound is not everything. The player must have some sort of graphical user interface (GUI). Some players on the market (such as the iPod Shuffle) does not have a GUI, this is not acceptable on a player with several gigabytes of music. Almost all players on the market today feature colour displays.

As it turned out, it was not easy to find colour displays in small volumes. Eventually a 2.5 inch TFT supporting up to 262,144 colours was found. The display manufactured by a Chinese company, Tianma, was not well documented, the data sheet [20] was very incomplete. However the datasheet for the controller IC used [21] was available on the Internet.

From these two datasheets the conclusion was drawn that the TFT was easy to connect to the processor through its external system bus without any extra glue logic.

3.7 Power supplies

The player should run on batteries according to the requirements in section 1.5. Batteries are not capable of delivering constant voltages.

For example, the voltage of a lithium polymer cell varies from about 2.7 V (discharged) to about 4.23 V (fully charged) [14]. Since stable voltages at 3.3 V and 1.8 V are needed no matter what the current battery voltage is, there is need for voltage regulators. For high efficiency these regulators should be of switching DC/DC type.

At first there was some hope of being able to use a power management unit (PMU) from Philips Semiconductors. It had multiple individually adjustable voltages outputs. However using this PMU was not going to be possible since the smallest ordering volume was about half a million pieces.

Since no other comparable PMU was available on the market at the time it was decided to use separate DC/DC converters for the different required supply voltages. The DC/DC converters chosen were TPS62005, TPS61031 and TPS61061, all from Texas Instruments. They were chosen mainly because of their availability (they were all available as free samples from Texas Instruments web page [22]) and because they have the right properties.

One major concern for the TPS61031 was to make sure it was capable of delivering enough current to power the HDD during spin-up. The TPS61031 is capable of delivering up to 1A at 3.3V so there is no problem driving the HDD at any time.

For more detailed information about these ICs, see the Texas Instruments web site [22].

3.8 Battery charger

There are many battery charger ICs available on the market, most of them do not differ much from each other. Where they differ, is mainly on how the charging monitoring works, temperature monitoring, and so on. They may also differ in what battery types they are capable of charging.

The BQ24023 from Texas Instruments is a simple lithium polymer battery charger with no temperature monitoring. This was chosen because it was one of the most basic chargers available and the current leakage in its off-state was almost none.

The charging sequence used is to apply a constant current until the charged voltage is reached at the battery. After that, the voltage is kept constant until the charging current reaches about 6-7 percent of the initial current.

3.9 Bluetooth

AirLogic Co. Ltd. [23] is a manufacturer of bluetooth devices. They have a couple of interesting bluetooth ICs. A try to establish contact with their sales office was done, but resulted in no response. Due to this it was decided to go without bluetooth since bluetooth connectivity was not a requirement, and since too little time for the implementation of this feature was available.

Chapter 4

Hardware design

In this chapter steps taken in the hardware design process will be described. Also an account of the design choices made and the underlying facts will be given.

4.1 Component packages

When the main components have been selected the task of assembly begins with the choice of packages. Here several factors which affect the choice of package for each component must be taken into account. Some of the most important factors when choosing packages are availability, package type, physical size, performance and functionality.

4.1.1 BGA, Ball grid array

Many of the components were available in BGA packages. These packages are surface mounted, small, of high performance and usually have a lot of pins allowing more functionality.

The downside with BGA is the assembly requirements. BGA packages do not have physical pins but solder pads arranged in grids under the package, thus the package cannot be soldered with a soldering iron¹. This grid also makes routing complex, requiring 4 or more layers and very small vias situated between the footprint pads on the PCB. These vias, often referred to as micro-vias, are rare and expensive to manufacture.

¹BGA packaged chips have to be factory mounted or mounted through the use of ovens or other elaborate heating devices.

4.1.2 QFP & SO, Quad footprint & Small outline

QFP and SO packages are available for almost all components and are the most common alternative to BGA when a high pin count is needed. The pins are arranged along the sides of the package (SO uses two sides and QFP all four) making them fairly easy to solder and route depending on the pin spacing (pitch).

These are larger than BGA equivalents and therefore usually are functionality-reduced to lessen the number of pins. They may also be slower than their BGA counterparts.

4.1.3 QFN, Quad footprint no-lead

The QFN package is a blend between BGA and QFP package types. QFN packages use solder pads instead of pins but these are arranged along the sides of the package. This makes them fairly small and possible to solder using a soldering iron.

These packages are not as common as QFP and are not available with as many pins, but are much more compact than QFP equivalents.

4.1.4 Conclusion

Using BGA packages is compelling when designing for small size as well as performance but the problematic assembly and complex routing increases costs² as well as development and prototyping time. Thus using QFN, QFP and SO packages prioritized after size and availability seems to be a better choice.

4.2 Interconnections

Components must be connected to create a working system. For this purpose there are several data busses & bus protocols available. The ColdFire MCF5249 [10] has dedicated bus hardware for several protocols; I²C, SPI, I²S, IEC958, UART, BDM and a system bus. It also has hardware support for the IDE protocol and a SDRAM controller, both using the system bus.

4.2.1 I²C, Inter-IC

The I²C protocol uses a two wire serial bus for low speed communications which is most suitable for controlling peripheral de-

²See chapter 4.4.3, PCB design - conclusion.

vices [27]. This protocol is very common for this purpose and is implemented in the Wolfson WM8750L audio codec [16] and the Phillips TEA5764HN radio tuner [29]. It is also used by the Cypress CY7C68301B USB to ATA bridge [13] to communicate with its configuration EEPROM.

The ColdFire MCF5249 [10] has one dedicated I²C and one shared I²C/SPI bus interface. The first solution was to use I²C for the radio and SPI for the audio codec but this was changed to I²C only, with minor hardware modifications, due to existing software support for I²C.

4.2.2 SPI, Serial peripheral interface

SPI is a faster alternative to I²C using three wires (plus chip-selects) [10].

4.2.3 I²S, Inter-IC sound

Philips IIS, I²S, I is a special protocol for audio transfers [28]. This is supported by both the ColdFire MCF5249 [10] and the Wolfson WM8750L [16] and was thus used without much debate. Both components also claim support for the Sony EIAJ protocol using the same bus hardware, but there is no obvious reason to use the EIAJ protocol instead.

A design choice to note is that the audio codec was chosen to act as bus master. The audio sample rate timings are then handled by the audio codec rather than the CPU.

4.2.4 IEC958

IEC958 is also known as S-PDIF and is also an audio transfer protocol but is intended for digital transfers to other devices rather than for direct DAC/ADC communication like I²S. This is not used since there is no component that supports this interface, I²S is used instead.

4.2.5 UART, Universal asynchronous receiver/transmitter

The UART protocol is commonly found in the form of RS232 (PC COM-port). Since this hardware was available it was connected to a RS232 physical interface intended for debugging.

4.2.6 System bus

The ColdFire MCF5249 CPU system bus is a large bus using 16 data pins, 23 address pins plus control pins; read/write (R/\overline{W}), Output enable (OE) and chip-selects ($\overline{CS\#}$). It features programmable chip-selects, wait-states and bus termination. It supports 16 and 8 bit port size but can only address 16 bit due to lack of address 0 ($A0$) pin [10].

Chip-select 0

The CPU itself uses the system bus and chip-select 0 for boot purposes, so it has to be connected to a boot memory, in this case a Flash memory.

Chip-select 1

The chip-select 1 pin is unconfigured at start-up and is connected to the TFT screen. The TFT screen does not have a compatible bus interface [21], instead it uses separate read (\overline{RD}) and write (\overline{WR}) pins. This usually results in the usage of so called glue logic, but in this case additional address pins could be used as control pins. The TFT interface does not trigger events without an active chip-select so unrelated activity of address pins is harmless.

For this solution address 1 ($A1$) is connected to the 1 bit address on the TFT interface (RS), address 2 ($A2$) is connected to read (\overline{RD}) and read/write (R/\overline{W}) is connected to write (\overline{WR}). Effectively this enables reads and writes using special memory addresses. Using “Base address + 0x2” or “Base address + 0x6” for writes and “Base address” or “Base address + 0x4” for reads. Writing or reading from these addresses in any other way results in unknown operation or read of invalid data.

Another problem with the TFT interface is that it is 8 bit and requires two consecutive read or write operations to the same address since it internally has 16 bit registers. Fortunately the CPU converts 16 bit writes to two 8 bit writes and since address 0 ($A0$) is missing the address incrementation remains unseen. (Writing is only allowed on even addresses for 16 bit words.) Thus using the TFT interface in the connected way works without any problems.

SDRAM, Synchronous dynamic RAM

The ColdFire MCF5249 has a built-in SDRAM controller [10] which shares address and data pins with the system bus. Control pins are

dedicated for this purpose and only the bus clock ($BCLK$) is shared, though no other component uses the bus clock in this design. This is used to connect the SDRAM component in the design.

IDE, Integrated drive electronics

The CPU also has dedicated hardware for IDE interfacing. This interface also shares address & data pins with the system bus and uses dedicated control pins. A problem with this is that adding a hard drive in addition to the flash, SDRAM and TFT already on the system bus will add excessive load and compromise signal integrity. As a solution to this, the ColdFire MCF5249 [10] includes a buffer enable 1 pin ($\overline{BUFEN1}$). Unfortunately this pin is intended to drive a buffer situated after the SDRAM and before the flash. The buffer enable 2 pin ($\overline{BUFEN2}$) intended for IDE buffering does not exist on the QFP package variant posing a troublesome problem.

Trying to work around this problem several solutions were considered, the most prominent solution was to use a GPIO pin as buffer enable. Though this would be impossible since the system bus would have to be exclusive to the IDE when active and this is very hard to achieve and almost impossible to guarantee. For example a soft reset (interrupt or exception) could leave this pin active which would cause a bus collision on the first read when booting, the system bus is also used when running the firmware since it is stored in SDRAM at runtime. These problems may be avoidable by using the internal SRAM for all interrupt & exception handlers but would require a lot of work fitting the code into the SRAM and making sure no code is executed from SDRAM accidentally.

Thus the most transparent solution was to add some glue logic, in this case an OR gate. The gate performs “or” on buffer enable 1 ($\overline{BUFEN1}$) and a GPIO pin that defaults to high at reset, the output is used as the new buffer enable. At boot this would lock the buffer enable high, disabling the buffer, and when the GPIO is driven low the buffer would follow buffer enable 1 ($\overline{BUFEN1}$). This only poses one restriction; even if buffer enable 1 ($\overline{BUFEN1}$) is configured to follow the IDE signals, it always follows chip-select 0 ($\overline{CS0}$). Thus, the flash must not be read when the IDE buffer is active since that would cause a bus collision. This is not a severe problem since the flash is only used at boot, uploading all executable code to SDRAM before starting IDE transfers.

4.2.7 BDM, Background debug mode

The BDM protocol is intended for debugging but is also used for programming software to the system flash. This protocol is very useful for development, allowing code to be uploaded to RAM and debugged, during runtime, remotely on a PC. The Freescale MCF5249 [10] allows single-stepping, breakpoints, etc. for precise code & data debugging.

4.2.8 GPIO, General purpose input/output

Protocols not supported by the CPU can be implemented in software and use GPIO pins as configurable busses. Additional protocols & busses were not needed since all connected components had dedicated hardware for these purposes. This left GPIO pins for simple input/output signals which is their ideal usage.

There were quite many signals that needed to be connected in this way; battery charge control, power management, USB detection & bridge control, push-button inputs and LED outputs, to name a few. Since GPIO pins often are shared with dedicated logic, these pins may exhibit random behaviour after reset. This meant short-circuits³ and unwanted activity⁴ had to be taken into account until these pins had been configured properly. Only a handful of the GPIO pins had defined reset behaviour requiring careful planning of signal assignment.

4.3 Peripheral components

The component selection process results in a handful of primary components, but these are rarely sufficient for a working system. Most components require peripheral components for purposes such as power supply filtering, analog signal processing and input sources.

4.3.1 Passive components

A lot of components are simple resistors, capacitors, inductors and diodes. These are used mostly for power supply filtering but have many other uses such as pull-up/down on signals to prevent unintended operation and signal processing (voltage dividers, filters and

³Short-circuits appears when two outputs are driving the same signal to different levels, also a signal not driven may float to invalid voltages causing large current flows at input gates.

⁴Some GPIO output clock signals which are, for example, not suitable for power management.

feed-back circuitry). Diodes are quite useful for voltage regulation and current rectifiers but also for cosmetic and user interface purposes in the form of LEDs.

4.3.2 Active components

Often more complex components than passive are needed such as logic gates or discrete devices like transistors. Logic gates are useful as glue logic or for simple functionality such as de-bouncing push-buttons and bistable latches (multivibrators). Transistors are suitable for controlling power supplies or very simple digital signal operations.

In this project a bistable latch is used for power supply on/off control, and is directly connected to the battery power. This latch consists of two cross-connected NAND gates with inputs from a power button and CPU. Using this, power can be disabled for all other circuits when the player is in an off state and still react to events on the power button.

4.3.3 Electro-mechanical components

Some components are needed for physical properties as well as electrical, mainly push-buttons and connectors. Push-buttons are used for user interaction (power, play, stop, etc.), connectors are plentiful and come in many forms (power, RS232, USB, stereo plug, IDE, BDM, etc.).

4.4 PCB design

Selecting a PCB for the mounting of the components was a straightforward decision. The only viable alternative to PCBs are prototyping boards and those are mostly intended for hole mounted components. Even though there are surface mount capable prototyping boards, finding one capable of mounting the many different footprints needed seemed impossible. The majority of components chosen for this design were not available as hole mounted, rendering these boards useless for this project. There are cutting edge technologies that provide alternatives to PCBs such as using conductive weave for fabric “boards” [25] and conductive ink [24], none of these are practical for this project or readily available.

4.4.1 Cost vs Functionality

Even if all PCBs are based on the same manufacturing technology they are very configurable. Functionality- and cost-wise the most interesting parameters are; board size, minimum trace/drilling sizes and number of layers. Physical size primarily depends on the routing and component requirements. To reduce board size, components can be mounted on both sides of a PCB as well as using multiple layers to reduce routing space.

Minimum trace and drilling sizes also have an influence on board size but are more important when fulfilling component requirements. Minimum trace size defines the smallest signal separation and width, this must be small enough to fit the smallest pin pitch needed. A small trace width can be beneficial when minimizing board size but most benefit is gained from the simplified routing process. This makes a small trace width a sought-after feature. Drill size has similar properties to trace width, usually minimum drill sizes are far smaller than hole mounted component requirements but they are still relatively large in comparison to trace widths. This complicates routing since drill holes are used for vias⁵. Depending on number of layers and component layout vias can quickly amount large quantities of drill holes, increasing cost and board size.

Using multiple layers can decrease board size, lessen routing effort and provide better electrical properties. 2-layer⁶ PCBs are a necessity when designing for almost anything but the simplest circuits⁷. They also allow for components to be placed on the bottom side of the board. 4- and 6-layer PCBs are also available and are a very interesting alternative to 2-layer but come at increasing costs. As discussed in chapter 4.1.4, Component packages - conclusion, the use of BGA packages was omitted largely based on the difficulty of mounting them, thus using more than 2 layers was not mandatory.

4.4.2 PCB manufacturers

There are several PCB manufacturers available, from these Beta LAYOUT Ltd. was selected because of their low costs and high performance PCB boards. Their website [26] specifies their technology limits to 6mil (0.15mm) minimum trace gap/width and minimum drill size to 12mil (0.3mm). This satisfies the estimated trace

⁵Vias are used to route a signal from one layer to another.

⁶Also known as double-sided PCBs.

⁷On a single layer PCB signals can not cross paths without extra components, using two layers theoretically allows arbitrary routing though trace/drilling sizes provide a natural limit.

width minimum of 8mil (0.2mm) and provides drill sizes that are far smaller than required.

4.4.3 Conclusion

To reach a conclusion the pricing of PCB layers according to table 4.1 were studied. As can be seen the price increases steeply from 2- to 4-layers, thus making 4-layer PCBs rather expensive, though the decrease in size that results from an increase of layers must be taken into account. A decrease in size of 50% seems very unlikely. Thus, the pricing can still be considered to be over €100 making the price difference noticeable. After much thought a 2 layer PCB was selected due to pricing and the ability to do post-manufacturing changes⁸, considering of course that this board would not have hard size-constraints that a final encapsulated version would have.

In hind-sight using a 4-layer PCB for this project would have been more time efficient and resulted in a higher quality board. The current design led to some very poor signal routing as a consequence of the large number of signals needing high performance. A few signals were even routed by additional wires to save time and effort in trying to route them on the PCB.

Layers	Price/dm ²
1	€48.75
2	€61.25
4	€164.00
6	€205.00

Table 4.1: PCB technology prices⁹

⁸PCBs can be modified by cutting the copper traces and soldering wires to the correct connections, this cannot be done on the internal layers of 4- or 6-layer PCBs.

⁹Prices taken from Beta LAYOUT Ltd. website [26].

Chapter 5

Software

In this chapter the software used, any problems that may have arisen and their solutions will be described.

5.1 Firmware

Embedded systems are rarely comprised of purely hardware. This is mostly because, in comparison with reusing configurable ASIPs or programmable ICs like MCUs, DSPs and CPUs, designing pure hardware ASICs is a time-consuming and costly ordeal. Thus software is used to implement functionality at a faster and more cost-effective way. This software may more or less be transparent to the user of the system and is therefore generally referred to as firmware.

The solution chosen for this project, as described in chapter 3.1, was to use a MCU and implement needed functionality in software. This software would have implemented a lot of features, some of which are:

- User interface
- Audio encoding & decoding
- File management
- Power management
- Peripheral control
 - Hard disk drive
 - TFT LCD monitor
 - Audio codec
 - Radio tuner

- USB interface
- Push-button interface
- Battery monitor & charger

Implementing all this functionality would be too time-consuming so it was decided that existing software should be used as much as possible. When using existing software, care has to be taken due to copyright and licensing limitations. Although, there are licenses that allow for relatively unhindered use.

The most notable sources of software are the growing communities of users and developers working on free¹ and open source software. These communities develop software for almost all purposes and provide the source code through licences such as GPL or BSD-licenses.

5.1.1 Rockbox

Since there was an existing free firmware known as Rockbox (See Software section in chapter 3.1.4) already ported to the MCF5249 MCU, this was a perfect choice.

Rockbox consists of an operating system and an application layer. The operating system layer implements a cooperative multithreading kernel with message passing ability, drivers for hardware and dynamic CPU frequency control. The application layer implements functionality that does not directly depend on hardware implementation, only on the functionality it presents. Applications may, for example, be a GUI or audio encoding/decoding.

From studying the Rockbox source it was concluded that the iRiver H100 was the most similar target to this projects hardware design, though many modifications were still needed.

GPIO configuration and pin-mapping

Even though there are large similarities between the system designed and the Rockbox target almost no GPIO-pins were used for the same purpose. This meant all usage of GPIO pins had to be modified to match this project's design.

Wolfson WM8750L

Since no Rockbox target utilized a Wolfson WM8750L compatible IC, a driver had to be written from scratch. This driver handles

¹As defined by FSF (“Free as in speech, not beer”), often referred to as open source.

configuration, power management and runtime operation of the IC. Changes were also made in the code outside of the driver like I²C & I²S setup on the MCF5249 and extended feature support.

A lot of issues were experienced with the hardware during development, the most notable was getting I²S communication to work properly. This was finally solved using oscilloscope probing and configuration experimentation. The conclusion from this issue was that the MCF5249 User Manual [10] lacked important information about the I²S interface when used in slave mode.

TFT monitor

It was found that the TFT monitor used the same controller IC as the monitor used by iRiver's H300 player. Though this player was not officially supported by Rockbox at the time, development was underway and a driver for the controller IC had already been written. Unfortunately Rockbox's GUI was not originally intended for colour monitors, thus uses only simple foreground/background colour schemes. This leaves room for ample visual improvements, but still give a perfectly functional interface.

5.1.2 Bootloader

The original Rockbox bootloader was designed to boot from the flash memory, initialize basic hardware functions and load the firmware from HDD to the SDRAM and execute it from there.

As mentioned in section 4.2.6, it is not possible with this hardware to access the flash and the IDE at the same time. Hence, there was need for some other way to boot the firmware than the original bootloader.

The basic idea with the new bootloader is to boot from the flash memory, load a program (that resides in the flash) into the SRAM and execute that program. Thereafter that program will enable the IDE² and load the firmware from the HDD to the SDRAM and execute it from there.

5.2 Development environment

To successfully develop such a complex system as this there is need for some kind of development environment. The development tools used for this project consisted of a set of GPL'd software.

²After that, flash cannot be accessed without disabling IDE.

5.2.1 GCC & Binutils

The GNU Compiler Collection, GCC, includes front ends for C, C++, Objective-C, Fortran, Java, and Ada, as well as libraries for these languages.

The GCC compiler generates assembler code from a high level language source code. Since assembler code cannot be executed directly on the target processor it needs to be assembled into a processor specific binary code.

Binutils is a set of tools to make binary executables. It contains the assembler to generate the binary code that can be run on the target processor.

To have support for the m68k architecture and the ColdFire MCF5249 these tools had to be compiled with m68k support. This was fairly easy and worked as described in the installation instructions on the Rockbox web page[9].

Both GCC and Binutils is part of the GNU Project and can be downloaded from the GNU Project web site[30].

5.2.2 GDB

Debugging software can be a very time consuming and nerve wrecking ordeal. To be able to debug a complex program such as a full firmware one must have some kind of tool. This is where the GNU Project Debugger, GDB, comes in handy.

GDB is a debugging tool that lets the user load their program and enter breakpoints, examine memory contents, single step the program, and many more features.

The GDB utility is a part of the GNU Project, and can be downloaded from the project web site[30].

5.2.3 BDM software

The BDM has been mentioned earlier, in section 4.2.7. To be able to use all the features of the BDM interface some software was needed. There are a set of proprietary software for this purpose, but there also is a set of tools that indeed is free software.

This set of tools included a utility to flash the boot memory through the processor, and more important, it included a device driver making GDB working over the BDM protocol.

The install procedure of this toolset was not as easy as it seemed at first. However, lots of help was given from one of the leading developers in the Rockbox development team. Also a set of useful

scripts were retrieved from him that helped setting up the debugging environment.

The set of BDM tools is available from SourceForge [31].

Chapter 6

Problems and mishaps

This chapter will briefly give an account of the problems and mishaps experienced during the project.

6.1 Manufacturers

Problems encountered with manufacturers were not that many, most manufacturers were very professional and accommodating, though unfortunately there were some mishaps. It must be made clear that, though the following text may reflect negatively upon the manufacturers at hand, these were isolated experiences and do not represent a comprehensive review of their respective services.

AirLogic Co., Ltd

AirLogic Co., Ltd [23] did not respond to e-mail inquiries. This is also mentioned in Chapter 3.9. The reason for this is unknown and it was not followed up later due to loss of interest in their product offering.

Silicon Laboratories, Inc.

SiLabs [19] had very appealing products but unfortunately they required a NDA to be signed, see chapter 3.5. This is problematic since an NDA will conflict with the requirements of the GNU GPL software license, thus prohibit distribution of any software, source or binary.

SigmaTel, Inc

SigmaTel [6] was originally a target of interest due to their D-Major chipset (See chapter 3.1.3) but it was clarified that their SDK costs \$12,000 and that they do not support third-party SDKs. This prohibited further development for this chipset.

Royal Philips Semiconductors

Philips Semiconductors [4] has a wide range of high quality products, one of these being a PMU. Unfortunately this PMU is factory configured and thus Philips requires a minimum order of 500,000 pieces. This is far too many for the PMU to be a viable option in this project.

Philips also provided a FM-radio chip. This chip was acquired uneventfully but problems were encountered when acquiring the special varicap diodes that were specified as peripheral requirements. These were not possible to acquire from normal suppliers and thus were ordered as samples from Philips itself. This was followed by a long delivery time inhibiting incorporation of the FM-radio chip.

Micron Technology, Inc

Micron is a SDRAM vendor and was requested for samples (See chapter 3.4.2) of a low power chip, but due to a “company reorganization” delivery was severely delayed. This forced the search for a temporary alternative and SDRAM chips that were physical and functional equivalents were found. Unfortunately these chips were not electrically equivalent, specifically they were not low power and will therefore have a negative affect on the power consumption.

Cypress Semiconductor, Corp

Cypress was requested for samples of a USB chip, see chapter 3.2.3, though their sample arrived too late to be incorporated into the system. This functionality was therefore temporarily left out but should be easily included in future development.

Tianma Microelectronics Co., Ltd.

Tianma is a LCD monitor manufacturer. A sample monitor was acquired (See chapter 3.6) from this manufacturer but unfortunately the datasheet [20] was far from complete. The only available datasheet

lacked important information and was unclear on electrical properties. Fortunately a datasheet for the LCD controller IC [21] incorporated into the monitor was available and proved very useful.

6.2 Development

Developing a product without encountering problems is impossible¹, fortunately no severe problems were encountered.

Components

Some issues with hardware was encountered in a late stage of development. The Wolfson WM8750L codec and ColdFire MCF5249L had some compatibility problems due to clock speeds and protocol configurations over I²S, see chapter 5.1.1. This was solved by finding the correct configuration parameters through careful reading of the datasheets, and using the trial & error methodology.

Other issues with components were difficulties during soldering. The ColdFire MCU's pins were very easily bent, even when nudging the chip into place on the PCB. The Wolfson codec was very difficult to solder due to it's very small size and the pin design of the QFN package. The QFN packages required a very sharp soldering tip, which are inherently harder to solder with, and the leadless pins inhibit the solder to flow out and make a good connection with the PCB. This was overcome with perseverance and the use of various tools such as a microscope.

PCB layout

As mentioned in chapter 4.4.3, the design choice to use a 2-layer PCB is in hind-sight not the best when dealing with routing requirements of this complexity. Routing the large system bus to the many components using it proved time-consuming and required some elaborate solutions, rarely being electrically optimal. Using a 4-layer PCB would have been more time-efficient but would have been more expensive, so this is still not a clear-cut choice.

The routing problems were solved by routing long, performance undemanding connections using wires soldered to vias. This method was also used to modify the board when the change from SPI to I²C was definite, see chapter 4.2.1 and 4.2.2.

¹Providing Finagle's law is proven, "Anything that can go wrong, will."

A true design error was made when placing the connectors for the TFT monitor. Even though the positioning was meticulously done, the two connectors did not line up with the monitor's. This was most likely due to a miscalculation or a simple offset direction mishap, which could have been caught by printing out the PCB board layout in real-size on paper and visually comparing connections.

Software

Software “bugs” are probably the most common problems by far. Fortunately these are often minor and simple to fix, though finding them can be a daunting task. Some of the problems encountered were spelling mistakes, misconfiguration of hardware or other absent-minded blunders.

Chapter 7

Results and future work

This chapter presents the results of this thesis, as well as some suggestions of how to further improve the device.

7.1 Results

A device that fulfilled all the requirements were implemented. However, the additional goals were not fulfilled because of chips showing up too late and due to the strict timeline.

Some problems was experienced during the project, such as components never showing up, and chip manufactures that did not reply to requests.

However, it have been proven that it is indeed possible for just two persons to develop a complete, working DAP in just 20 weeks and on a very limited budget.

The player is capable of playing a number of audio formats, Ogg/Vorbis and MP3 beeing two of them.

Open source software is the key to the success. It saved a lot of time since it was not necessary to develop an firmware from scratch, even though many task were to be done to get a functional firmware.

7.1.1 Power consumption

The power consumption of the resulting player was measured for some different music files. (Ogg/Vorbis and MP3 files in different bit rates.)

In table 7.1 the results from the measurements are summarized. According to these figures an approximate battery life of about 16 hours¹ can expected. Note that the hard drive power consumption

¹Approximated with a 2000mAh Li-pol battery.

is not taken into account in these figures. About 30 % less battery life can be expected when taking the hard drive into account. There was no disk drive of the right type available so no real measurements could be done. Average power consumption for the hard drive were approximated out of figures at the Hitachi web site [32].

Format	Bit rate	Power consumption
MP3	128 kbps	486 mW
MP3	160 kbps	493 mW
Ogg/Vorbis	141 kbps	508 mW
Ogg/Vorbis	168 kbps	532 mW

Table 7.1: Power consumption measurements

7.2 Future work

There are a number of possible improvements to be done in the future. In this section some of the possible improvements are presented.

7.2.1 Hardware

Improvements of the hardware is mainly concentrated to PCB size and power consumption.

Low power SDRAM

Since Micron did not send the low power SDRAM that was preferred, it was necessary to use an ordinary SDRAM to get the hardware working. This is far from optimal from a power consumption point of view. Once the low power SDRAM arrives it can replace the current one on the PCB. This will make the player run at significantly less power consumption.

Power supplies

Significantly more than halfway through the project, Texas Instruments released a power management unit similar to the one made by Philips mentioned in section 3.7. This should be a suitable replacement for some of the ICs used and would have saved space on the PCB.

PCB size

The main thing to improve is the PCB. Currently the PCB is quite large due to a number of reasons:

- A two layer PCB were used
- Full sized electromechanical components (IDE connector, BDM connector) were used
- Extra debugging functionality (RS232) were added to the PCB
- The PCB have other superfluous components² as well

By removing the debugging connectors and the superfluous components the PCB size can be reduced by a fair amount.

Using a four layer PCB will decrease the PCB size even more since a higher component density is achievable when signals can be routed on the inner layers of the PCB.

7.2.2 Software

There is still som unimplemented features in the software, such as a driver for the radio tuner and recording feature in the WM8750L driver. Also, the battery charger control is not implemented. Support for the USB mode is implemented but is untested.

Other things to do includes improvements of the GUI and much optimization of the software codecs. Optimization of the codecs will reduce power consumption even more since it will allow for the processor to run in a lower frequency for greater periods of time.

²For instance decoupling capacitors that might be unnecessary.

Bibliography

- [1] Bosi M et. al, *Introduction to digital audio coding and standards*, (2005), Dordrecht
- [2] Xiph.org Foundation, <http://www.xiph.org>, (September 2005)
- [3] Löwenborg P, *Mixed-Signal Processing Systems, Second edition*, (2006), Linköping
- [4] Royal Philips Semiconductors, <http://www.semiconductors.philips.com>, (September 2005)
- [5] PortalPlayer, Inc., <http://www.portalplayer.com>, (September 2005)
- [6] SigmaTel Inc., <http://www.sigmatel.com>, (September 2005)
- [7] SigmaTel Inc., *Product Data Sheet STMP35xx*, (2004)
- [8] Freescale Inc., <http://www.freescale.com>, (September 2005)
- [9] Rockbox Project, <http://www.rockbox.org>, (September 2005)
- [10] Motorola/Freescale Inc., *MCF5249 ColdFire Integrated Micro-processor Users's Manual*, (2003)
- [11] Future Technology Devices International Ltd., <http://www.ftdichip.com>, (September 2005)
- [12] Royal Philips Semiconductors, *ISP1582 Hi-Speed Universal Serial Bus peripheral controller*, (2005)
- [13] Cypress Semiconductor Corp., *EZ-USB AT2LPTM USB 2.0 to ATA/ATAPI Bridge*, (2005)
- [14] Wikipedia, The free encyclopedia, <http://www.wikipedia.org>, (October 2005)
- [15] High-Quality Low-Voltage Audio, <http://www.rane.com/hilovolt.htm>, (September 2005)

-
- [16] Wolfson Microelectronics plc., *WM8750L Stereo CODEC for Portable Audio Applications*, (2005)
 - [17] Royal Philips Semiconductors, *UDA1380 Stereo audio coder-decoder for MD, CD and MP3*, (2004)
 - [18] Handy J, Semico Research *Flash Memory vs. Hard Disk Drives - Which Will Win?*, (June 6, 2005)
 - [19] Silicon Laboratories, <http://www.silabs.com>, (September 2005)
 - [20] Tianma Microelectronics Co., Ltd., *Specification for LCD module TM0276ANFWG*, (2004)
 - [21] Renesas, *HD66789 528-channel, One-chip Driver for Amorphous TFT Panels with 262,144-color display RAM, Power Supply Circuit, and Gate Circuit*, (2003)
 - [22] Texas Instruments, <http://www.ti.com>, (October 2005)
 - [23] AirLogic Co., Ltd., <http://www.airlogic.co.kr>, (October 2005)
 - [24] Conductive Circuits Inc., <http://www.conductivecircuits.com/>, (January 2006)
 - [25] Post O et. al, MIT Media Laboratory, *E-broidery: Design and fabrication of textile-based computing*, (May 9, 2000)
 - [26] Beta LAYOUT Ltd., <http://www.pcb-pool.com>, (October 2005)
 - [27] Royal Philips Semiconductors, *The I²C-bus specification*, (2000)
 - [28] Royal Philips Semiconductors, *The I²S-bus specification*, (1996)
 - [29] Royal Philips Semiconductors, *TEA5764HN FM radio + RDS*, (2005)
 - [30] The GNU Operating System, <http://www.gnu.org>, (January 2006)
 - [31] SourceForge.net, <http://www.sf.net>, (November 2005)
 - [32] Hitachi Global Storage Technologies, <http://www.hitachigst.com/portal/site/en>, (January 2006)

Appendix A

GNU General Public License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your

cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent

infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years from the date of publication barring exceptional circumstances. The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: <http://www.ep.liu.se/>

Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår. Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art. Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart. För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

© 2006, Anders Kagerin, Michael Karlsson