**Examensarbete**

LiTH-ITN-KTS-EX--02/21--SE

# Audio over Bluetooth and MOST

Peter Ekström &
Fredrik Hoel

2002-03-07

**Department of Science and Technology
Linköping University
SE-601 74 Norrköping, Sweden**

**Institutionen för teknik och naturvetenskap
Linköpings Universitet
601 74 Norrköping**

LiTH-ITN-KTS-EX--02/21--SE

# Audio over Bluetooth and MOST

## Examensarbete utfört i kommunikationssystem vid Linköpings Tekniska Högskola, Campus Norrköping

Peter Ekström
Fredrik Hoel

Handledare: Thomas Söderqvist
Examinator: Johan M Karlsson

Norrköping den 2002-03-07

**Avdelning, Institution**
Division, Department

Institutionen för teknik och naturvetenskap
Department of Science and Technology

**Datum**
Date

2002-03-20

**Språk**
**Language**

☐ **Svenska/Swedish**
☒ **Engelska/English**

☐ _______________

**Rapporttyp**
**Report category**

☐ **Licentiatavhandling**
☒ **Examensarbete**
☐ **C-uppsats**
☐ **D-uppsats**
☐ **Övrig rapport**

☐ _______________

**URL för elektronisk version**

**ISBN**
________________________________

**ISRN LiTH-ITN-KTS-EX--02/21--SE**

**Serietitel och serienummer** **ISSN**
Title of series, numbering ________________________

**Titel** **Ljud över Bluetooth och MOST**

Title Audio over Bluetooth and MOST

**Författare**
Authors
Peter Ekström and Fredrik Hoel

**Sammanfattning**
I detta examensarbete studeras möjligheten att ansluta standardprodukter trådlöst till MOST, ett multimedianätverk för fordon. Den trådlösa tekniken som analyseras är Bluetooth. Rapporten beskriver teoretiskt hur MOST ska integreras med Bluetooth via en gateway och tar även upp olika framtida scenarier som möjliggörs med hjälp av denna gateway. Lösningen beskriver hur en förbindelse kan upprättas och ljuddata överföras från en ljudkälla till MOST-nätet med hjälp av Bluetooth-teknik.

Abstract
In this Master Thesis the possibility of connecting standard products wirelessly to MOST, a multimedia network for vehicles, is investigated. The wireless technique analysed is Bluetooth. The report theoretically describes how MOST could be integrated with Bluetooth via a gateway. Future scenarios that are made possible by this gateway are also described. The solution describes how a connection could be established and how the synchronous audio is transferred from a Bluetooth sound source to the MOST network.

**Nyckelord**
**Trådlös, Bluetooth, MOST, samplingsfrekvenskonvertering, interpolation**

Keywords
Wireless, Bluetooth, MOST, sample rate conversion, interpolation

# Abstract

In this Master Thesis the possibility of connecting standard products wirelessly to MOST, a multimedia network for vehicles, are investigated. The wireless technique analysed is Bluetooth. The report theoretically describes how Bluetooth could be integrated with MOST via a gateway. Future scenarios that are made possible by this gateway are also described. The solution presents how a connection could be established and how the synchronous audio is transferred from a Bluetooth sound source to the MOST network.

As a sound source equipment supporting the Bluetooth Headset Profile is used. It communicates with the MOST network via a gateway. As the recipient of the system, a speaker module connected to MOST is used.

The gateway task when transmitting audio, using synchronous data, is to convert the sample rate of the audio stream from 8 kHz used in the Bluetooth system to 48 kHz used in MOST. This is done by interpolation and filtering. Several different methods for this are described and compared.

The key issue in this report is the sample rate conversion between the two systems sample frequencies.

# Sammanfattning

I detta examensarbete studeras möjligheten att ansluta standardprodukter trådlöst till MOST, ett multimedianätverk för fordon. Den trådlösa tekniken som analyseras är Bluetooth. Rapporten beskriver teoretiskt hur Bluetooth ska integreras med MOST via en gateway och tar även upp olika framtida scenarier som möjliggörs med hjälp av denna gateway. Lösningen beskriver hur en förbindelse kan upprättas och ljuddata överföras från en ljudkälla till MOST-nätet med hjälp av Bluetooth-teknik.

Som ljudkälla används utrustning som stöder ’Bluetooth Headset Profile’. Den kommunicerar via en gateway med MOST-nätet. Som mottagare i systemet finns en högtalarmodul ansluten till MOST.

Vid överföring av ljud, i form av synkron data, är gatewayens uppgift att samplingskonvertera ljudströmmen från 8 kHz som används i Bluetooth-delen till 48 kHz som används på MOST. Detta sker med interpolation och filtrering. Flera olika metoder för detta redovisas och jämförs.

Huvuduppgiften i rapporten är samplingskonverteringen mellan de olika systemens samplingsfrekvenser.

# Preface

This report presents the results of our Master Thesis performed at Volvo Technological Development Corporation.

We would like to thank all of the people supporting us during our work in the department of Infotronics at Volvo Technological Development. We would also like to thank our examiner Johan M Karlsson at Linköping University of Technology. A special thanks goes to Thomas Söderqvist for his invaluable support and his great knowledge of both the Bluetooth and the MOST technologies.

Göteborg, Mars 1, 2002

Peter Ekström and Fredrik Hoel

# Contents

# 1 Introduction

The aim of this Thesis is to design a gateway between Bluetooth and MOST, Media Oriented Systems Transport. The gateway shall be able to handle control data to initiate a synchronous link and to route synchronous audio sent from Bluetooth to MOST.

## 1.1 Background

MOST is already an established technology in the automotive industry and Bluetooth is becoming more and more interesting. Combining those techniques would bring the possibilities further to another level where the applications in the vehicle can connect wirelessly. As a step in this evolution Volvo Technological Development is interested in developing competence in this area.

## 1.2 Method

The method used throughout this Master Thesis has been an iterative process. The first thing done was a target plan. It contained the different goals, a time plan etc. Then a literature study was made in which the Bluetooth and MOST technologies were studied. Previous reports in the area of Bluetooth, performed at Volvo Technological Development, were read [20, 21]. The task was separated into smaller tasks where the solution ideas were backed up by more thorough studies of specifications, books and articles. In the later part of the Master Thesis the area of Digital Signal Processing (DSP) was investigated. Finally the results and the solutions along with the descriptions of the different technologies were written down in this Thesis.

## 1.3 Limitations

There is no implementation description of the achieved solutions to this task. The work is done on a theoretical level were the possibility of interconnecting Bluetooth with MOST is investigated.

Security and error handling are areas that have to be considered more careful in a real system implementation.

## 1.4 Structure of the Thesis

Chapter 2, 3 and 4 are descriptions of the technologies of Bluetooth, MOST and DSP respectively. Those chapters could be skipped if the reader is familiar with those areas.

Chapter 5 describes the system design of the audio source.

In chapter 6 a specific gateway for this Master thesis is discussed as well as the descriptions of how control data and synchronous audio is supposed to be handled by this gateway. A general Bluetooth-MOST gateway and some future scenarios for this are also presented in this chapter. It ends by a description of a simplified implementation done.

Chapter 7 contains the conclusions and includes the future works of this report.

The appendix consists of frequency spectrums from interpolation simulations.

# 2 Bluetooth

This chapter will describe an overview of the Bluetooth history, organization and technology. The Bluetooth specification 1.1 has been the main source of this chapter.

## 2.1 Introduction

The Bluetooth technology has quite an exiting history though the idea of Bluetooth came up as late as 1994. Because of its origin, Bluetooth is strongly associated with Scandinavian culture and history.

### 2.1.1 The name

The engineers named the technology to honour the tenth century Viking king of Denmark. His name was Harald Blåtand, which translates into English as Harold Bluetooth. Harold became known as the king who united Denmark and Norway and christened the Vikings in his kingdom. Due to Harold's talent for diplomacy the Ericsson engineers thought it would be a good name for a technology that will unite the data- and telecommunications industry. Bluetooth has gathered multinational companies into the Bluetooth SIG (Special Interest Group).

### 2.1.2 The product

The technology was primary meant to be a cable replacement between different kinds of devices. Ericsson had 1994 a short link radio vision where they desired a power efficient and platform independent radio module. This cable replacement technology was going to have the following preferences:

- Be perfect for mobile devices (small, low power, low cost, low weight)
- Have short range distance
- Guarantee interoperability
- Open specification
- Ad hoc connectivity

The engineering work started 1995 but it was not until 1997 when Ericsson realized that they had to collaborate with other large companies if this technology was going to be a widespread success. Ericsson established Bluetooth SIG founder group together with Intel, IBM, Nokia and Toshiba. Gradually the SIG grew and the SIG promoter group was formed with the founders, 3COM, Lucent, Microsoft and Motorola [24]. The collaboration between that many large companies has made Bluetooth an open standard.

## 2.2 Organisation

Bluetooth SIG today consists of different member levels. From the beginning the Bluetooth name was a trademark owned by the Ericsson telephone company. Nowadays the Bluetooth name is owned by the Bluetooth SIG. The membership is divided into four levels:

- Promoters
- Associates
- Adopters
- Early Adopters

The tasks of the different member levels are described in Figure 1.

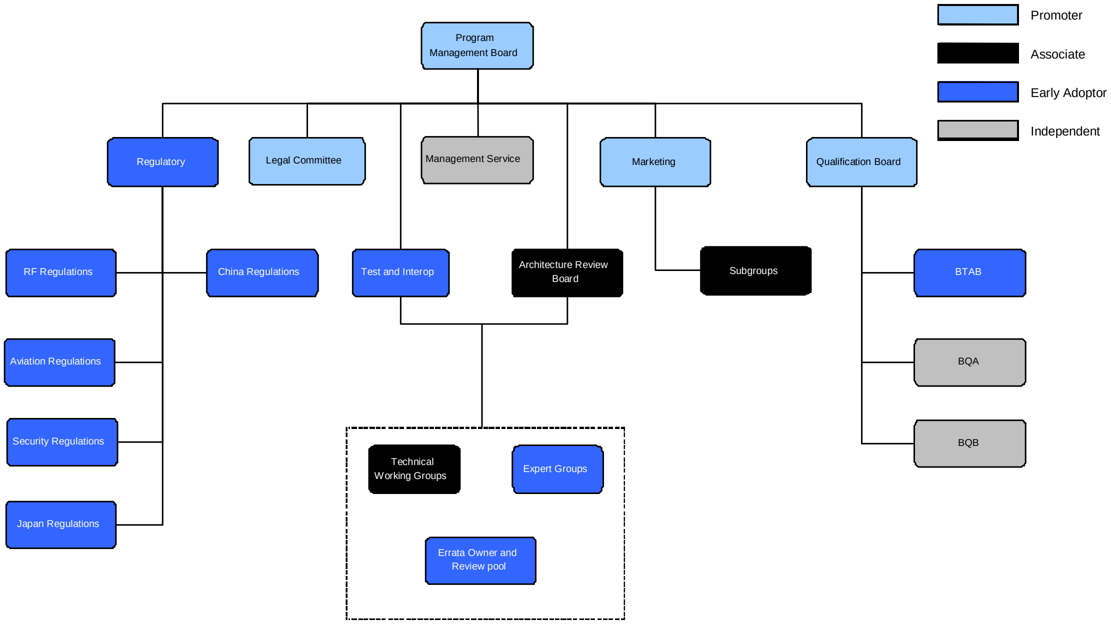


*Figure 1: Bluetooth SIG structure*

## 2.3 Characteristics

The Bluetooth radio is transmitting on the globally unlicensed frequency starting at 2,402 GHz and stopping at 2,480GHz. Bluetooth uses a frequency hop technology with 79 hops displaced by 1 MHz. The maximum hopping rate is 1600 hops per second. The frequency hopping procedure follows a scheme generated by the master (see 2.3.1). Frequency hopping helps the Bluetooth radio to avoid interference with other devices [1].

It is easy to implement Bluetooth everywhere without complications with governmental, military or other kind of frequency restrictions. One complication is that Bluetooth transmits on the same frequency as other products like microwave

ovens and WLAN adapters. This could lead to undesired interference [22].
In a Bluetooth network 8 devices can be simultaneously active. This is called a piconet. To connect more devices it is possible to connect up to 10 piconets into a scatternet [25]. The gross data rate is 1 Mbps but the net data rate is maximum 432,6 kbps symmetric duplex and 723,2 kbps asymmetric duplex. For voice it is possible to have 3 simultaneous synchronous duplex connections per piconet. The Bluetooth technology specifies 3 power classes presented in Table 1.

| Power Class | Maximum Output Power | Range |
|---|---|---|
| 1 | 100 mW (20 dBm) | ~100 meters |
| 2 | 2.5 mW (4 dBm) | ~35 meters |
| 3 | 1 mW (0 dBm) | ~10 meters |

*Table 1:Bluetooth Power Classes*

The range values in Table 1 depend on the antenna construction and if the devices are in line of sight or not. It is only the power that is specified in the specification [13].

When Bluetooth is transmitting voice it does not resend lost or corrupt packages. There are three formats supported for the air-coded signal: A-Law, μ-Law and CVSD. The logarithmic encodings (A-Law and μ-Law) are not yet supported by any Profile but are implemented for eventual future use. The audio subsystem is described in chapter 5.2.

CVSD is the most common encoding format in Bluetooth devices. It is a robust voice encoding format that follows the waveform of the signal and is very resistant to bit errors. The bit-errors are noticeable as background-noise. When the bit-error rate increases there will be more background-noise. Figure 2 shows a CVSD encoded signal trying to follow the original continuous signal.

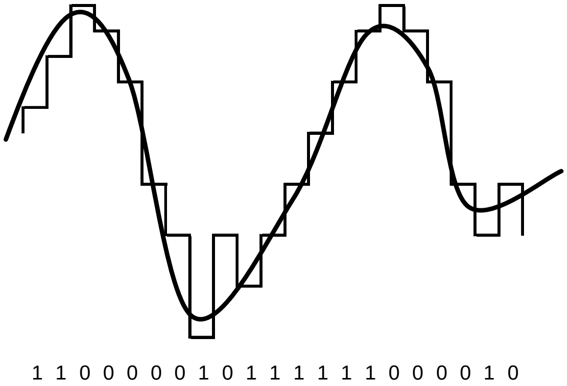


*Figure 2: CVSD encoded signal*

### 2.3.1 Network topology

The Bluetooth devices within the range of communication can build up a so-called ad hoc network. In difference to many other wireless systems, which have stationary transceivers, all Bluetooth devices in the network are equal except from the Master, which provides the clock etc. The different topologies are shown in Figure 3.

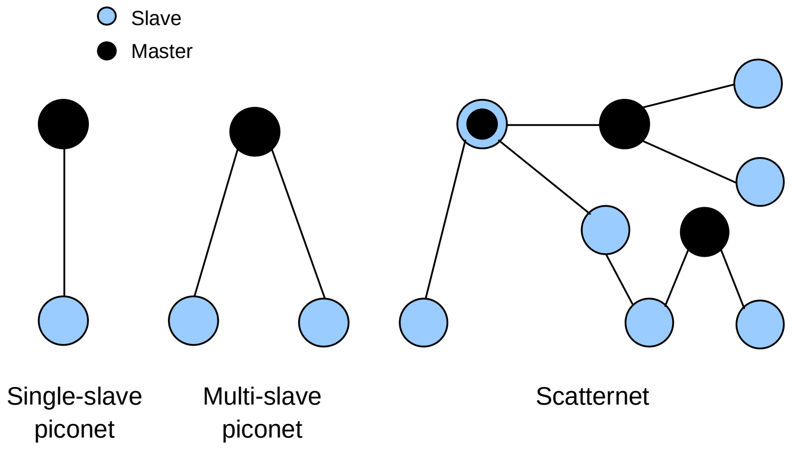


*Figure 3: Bluetooth topology structures*

## 2.4 The Bluetooth Protocol Stack

The Bluetooth specification contains a protocol stack that defines how the devices are supposed to locate, connect to and exchange data with each other. Figure 4 shows the Bluetooth stack mapped towards the OSI (Open Systems Interconnect) reference model [2].

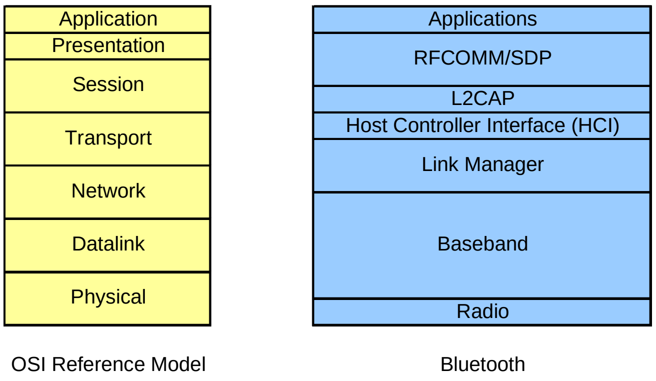


*Figure 4: OSI -Bluetooth*

In the following chapter the Bluetooth stack components will be described, starting with the Bluetooth Baseband and ending with the Bluetooth Profiles. The radio properties is mentioned in the chapter 2.3.

### 2.4.1 Baseband

The Baseband supports both synchronous and asynchronous data [9]. The synchronous link can contain both audio and data while the asynchronous link carries data and coded audio and video. Data packets can be provided with different kind of error correction. HEC, FEC and CRC are previously explained and they are mainly used for asynchronous data. The SCO (Synchronous Connection Oriented) links allows point-to-point communication between the slave and the master and the ACL (Asynchronous ConnectionLess) links also allows point-to-multipoint communication between the master and the slaves in the piconet. There are numerous functions that the Baseband handles. The main function is to control the link. Some of the other functions are:

- Clock supplying
- Frequency hop selection
- Paging and inquiry
- Security algorithms

As mentioned above there are two different kinds of links: ACL and SCO links. These two link types have different kind of packets in the Baseband protocol. When an SCO link is established the audio data is put in the voice field. The voice field has fixed length and no header. The voice can be High-quality voice (HV) or Data Voice (DV). The HV field has a length of 240 bits and the DV field has 80 bits. The asynchronous data field is divided into three segments: payload header, payload body and a CRC code. The Baseband packet can also mix ACL and SCO packets. The Baseband packet structures are illustrated in Figure 5.

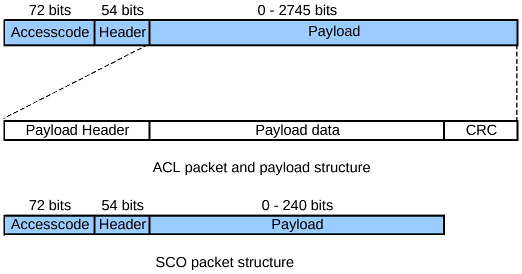


*Figure 5: Bluetooth Baseband packet structure*

The data- and voice transfer is designed to be robust. The data packets can have the following error correction and detection:

- ARQ (Automatic Repeat reQuest) that automatically resends corrupt packages.
- FEC (Forward Error Correction), which is a technique that is used to obtain optimal performance. It provides more bandwidth efficient ways to improve the bit error rate.
- CRC (Cyclic Redundancy Checksum) for detecting bit errors.
- HEC (Header Error Correction) a CRC function performed on the header.

### 2.4.2 Link Manager

To manage a Bluetooth link between the devices in the network the Link Manager is used. It handles the establishment of the ACL and the SCO links. The Link Manager governs also things like security, power, QoS (Quality of Service), transmission scheduling etc.

### 2.4.3 Host Controller Interface (HCI)

The HCI provides a command interface to the Baseband and the Link Manager. It also makes the hardware status and control registers accessible. In some systems where there are two main processors the HCI is the link between the systems. If a system has an embedded stack the HCI is not present. The interface is meant to provide a uniform way to access the Baseband capabilities.  This simplifies the integration for different manufactures. The important issue is to create a driver that handles the communication between the hardware integrated layers and the layers above such as L2CAP etc.

### 2.4.4 L2CAP

The Bluetooth specification includes a Logical Link Control and Adaptation Protocol (L2CAP). The L2CAP provides higher-level protocols with multiplexing and packet segmentation and reassembly (SAR). L2CAP permits higher-level protocols and applications to transmit and receive data ACL packets up to 64 kilobytes. SCO links are not supported in L2CAP. They are supported by the Baseband.

One of the most important L2CAP functions is protocol multiplexing. The protocol multiplexing is essential because of the separation of the upper layers. The data packet has to go through L2CAP because the Baseband protocol does not support a type field to identify higher protocols like SDP (Service Discovery Protocol), RFCOMM and TCS (See section 2.5.6). The other main function is SAR that divides the higher-level packets before transmission and then reassembles the packages after reception. The SAR is used to improve efficiency by supporting a maximum transmission unit (MTU) using larger packets than the largest Baseband packet.

Another function that L2CAP handles is QoS. During the connection process L2CAP allows exchange of information regarding QoS between the Bluetooth devices. The L2CAP ensures that the QoS contracts are enforced.

### 2.4.5 RFCOMM

The Bluetooth is a standard for replacing cables. The RFCOMM protocol emulates a serial cable using the RS-232 nine-circuit serial port standard [2]. RFCOMM relies on the Baseband to provide reliable sequenced data streams. The data stream rate will be limited where there are physical serial ports involved. If there are just Bluetooth devices in the network, RFCOMM will deliver the highest possible data rate. RFCOMM is included in many of the Bluetooth Profiles.

## 2.5 Profiles

To make the Bluetooth standard universal the Bluetooth SIG identified various usage models. The different usage models are implemented in so called “profiles”. A profile defines specific messages and procedures used to implement a feature. Some features are mandatory, others are optional and some may be conditional. In the Bluetooth SIG there are many workgroups, in which new profiles are developed. In Figure 6 the Bluetooth profile hierarchy is illustrated [14].

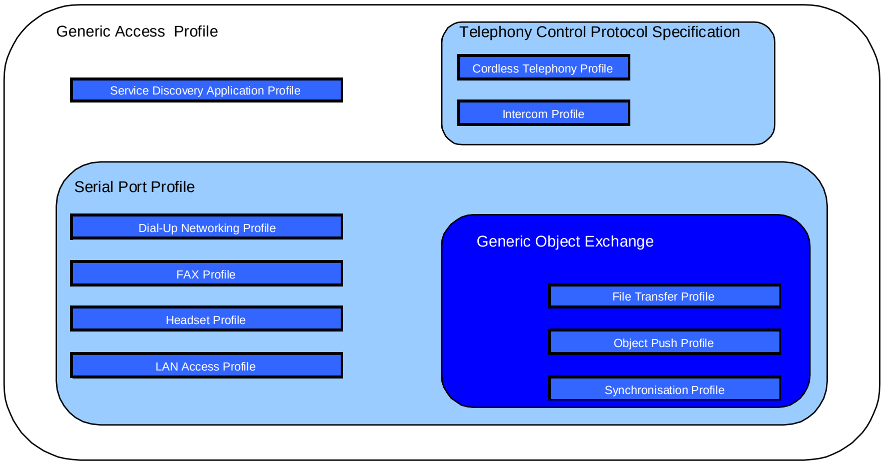


*Figure 6: Bluetooth Profile Hierarchy*

### 2.5.1 Generic Access Profile (GAP)

This profile is the most basic of the Bluetooth profiles. All the other profiles are based upon it and use its facilities. The GAP defines procedures for Bluetooth devices when they want to connect, discover identities, set up security etc. There are four different modes of operation defined [10]:

- Discoverability (non-discoverable, limited discoverable and general discoverable)
- Connectability (connectable and non-connectable)
- Pairability (pairable and non-pairable )
- Security (non-secure, service level enforced security and link level enforced security)

There are also other parameters that are governed by this profile. Preferences like Bluetooth device name, Bluetooth PIN and Class of Device are set to simplify the communication and the user interface. The parameters set in the GAP are called common parameters and must be supported in every Bluetooth device in order to work in a Bluetooth network.

### 2.5.2 Serial Port Profile (SPP)

Bluetooth was from the beginning and is still a technology for cable replacement. The serial port profile provides RS-232 cable emulation and is based upon GSM standard GSM 07.10 which allows multiplexing of numerous serial connections over one link. It is not just computers, PDA's and cellular phones that uses the SPP. Profiles like Headset, Dial Up Networking and Generic Object Exchange are built upon the SPP.

To provide a virtual serial port the SPP depends on the lower layers in the Bluetooth stack hierarchy. RFCOMM is used to create an L2CAP channel. There are numerous steps to be done where the initiator has different possibilities when setting up the connection.

### 2.5.3 Service Discovery Application Profile (SDAP)

This profile describes feature and procedures used to discover the services on other Bluetooth devices. SDAP retrieves information about the services that are supported and what features the services provide. For retrieving the service information the SDAP uses the SDP (Service Discovery Protocol), which is a protocol that is unique for the Bluetooth wireless technology.

### 2.5.4 Generic Object Exchange Profile (GOEP)

The GOEP defines the OBEX layer within Bluetooth. OBEX is a remainder from the IrDA protocol and is a standard for exchanging virtual vCards, vCalendar data etc. GOEP also defines how the link layer sets up client/server communications. As shown in Figure 6, there are three Profiles that depend on the GOEP and those are:

- File Transfer Profile: Defines simple file transfers from terminal to terminal
- Synchronisation Profile: Provides a standard way to synchronize personal data
- Object Push Profile: The primary task is to exchange business cards.

### 2.5.5 Headset Profile (HP)

One of the most trivial profiles is the Headset Profile. It defines the facilities required to make it possible to receive hands-free calls from a cellular phone. The Bluetooth headset is made simple in order to minimize the device's size, power consumption and processing power. The headset is controlled via buttons on the headset.

The Bluetooth headset is based upon other profiles to work, the Generic Access Profile and the Serial Port Profile. The Serial Port Profile emulates a serial connection where the control messages are sent. The Headset Profile is described more in chapter 5.

### 2.5.6 Telephony Control protocol Specification (TCS)

TCS is based upon the existing ITU-T recommendation Q.931. It is a binary encoding for packet based telephony control and is often shortened TCS-BIN. There are two Profiles that use the TCS: Cordless Telephony Profile and Intercom Profile. The Cordless Telephony Profile is a way to let Bluetooth devices interconnect with the PSTN and the Intercom Profile is meant to work as a walkie-talkie, i.e. half-duplex. The TCS and its profiles could make a Bluetooth equipped cell phone into a so-called "three in one phone".

### 2.5.7 Dial Up Networking Profile (DUNP)

The Dial Up Networking Profile defines protocols and procedures used by devices like modems and cellular phones for connecting to computer networks. There are two roles defined for the DUNP: the Gateway (GW) that provides the access to the network and the Data Terminal (DT) that is the device that uses the dial up service.

### 2.5.8 LAN Access Profile (LANAP)

To connect multiple Bluetooth devices to Local Area Network (LAN) the LAN Access Profile is used. The LANAP defines Data Terminals that connect to the LAN Access Point (LAP) via the Point-to-Point Protocol (PPP). Protocols like TCP/IP and IPX are supported but LANAP does not need to use any particular protocol.

### 2.5.9 Fax Profile (FaxP)

The Fax Profile defines the procedures for sending and receiving faxes wirelessly. It can be considered a special case of the DUNP. In many aspects fax and data transmissions are similar. As in DUNP and LANAP the Data Terminals need a Gateway. The Gateway provides the access to the PSTN and can be a cellular phone, a cordless phone or a modem.

## 2.6 Future

In the approaching future the Bluetooth standard will be applied with new Profiles. Some of the new Profiles are developed by Car, Printing and Local Positioning Profile Working Groups. The Car Profile Working Group will be interesting since it works with Profiles that will support audio and other suitable functions for implementation in cars.

The Bluetooth development groups work on the Bluetooth 2.0 release. The 2.0 version will preserve backwards compatibility with the 1.0 version which could mean that the devices has to be able to handle two modulation types. Except from new Profiles substantial improvements will be on the Baseband and the Radio [2]. One of the largest improvements will be the increase of data rate. It is possible that Bluetooth 2.0 will support data rates as high as 12 Mbps.

# 3 MOST

This chapter will describe an overview of the MOST history, organization and technology. The MOST specification revision 2.1 has been the main source of this chapter.

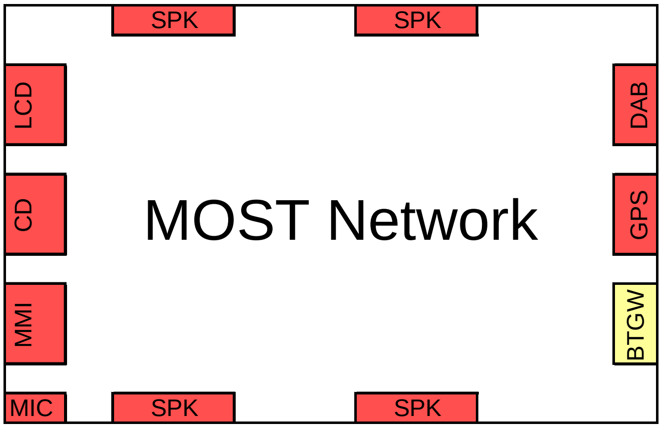
## 3.1 Introduction

Today there are a lot of multimedia products in the car and the market is growing. Audio, video and information products are becoming a more and more important factor at a car purchase.

Along with the fast expanding market on multimedia products, the need for an automotive multimedia network grows stronger. The idea with a network like this is to reduce the cables, and therefore also the cost, and at the same time interconnect all of the multimedia devices as well as other communication devices. The interconnection would make the use of products easier due to the possibility to use the same interface such as a keypad and a display. The same interface towards the system increases the flexibility and minimizes the product dependent design.

MOST, Media Oriented Systems Transport, is an open standard which gives the developers freedom to design products of their own that still would be compatible with the MOST system. Carmakers supporting the standard are above all European but also Asian and American. The MOST standard is since 2001 represented in serial manufactured cars.

The system supports up to 64 units connected to the network. Those units could be cameras, TV sets, navigation systems, video players, Hi-Fi audio equipment or multimedia computers. They are typically controlled via the same Man Machine Interface (MMI) i.e. a keypad and a display. A MOST network is illustrated in Figure 7.



| | |
|---|---|
| SPK | Speaker |
| BTGW | Bluetooth Gateway |
| DAB | Digital Audio Broadcast |
| GPS | Global Positioning System |
| LCD | Liquid Crystal Display |
| MMI | Man Machine Interface |
| CD | Compact Disc |
| MIC | Microphone |

*Figure 7: A MOST network in vehicle*

## 3.2 Organisation

In 1997 the MOST Cooperation began as an informal cooperative effort. The organisation was officially founded 1998 by BMW, DaimlerChrysler, Harman/Becker and OASIS SiliconSystems and today there are 17 car makers and 50 key component suppliers joined in the cooperation.

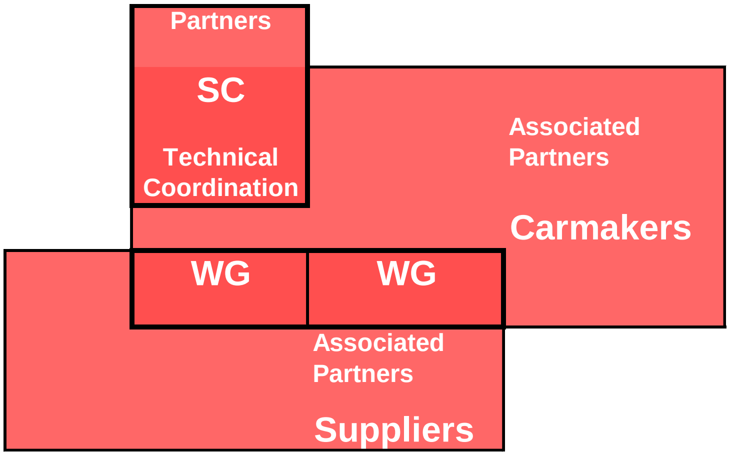


*Figure 8: MOST Cooperation structure*

There are three different member levels in the MOST cooperation. They are the *Steering Committee (SC)*, the *associated partners carmakers* and the *associated partners suppliers*. The SC runs the technical coordination and the members are from the companies who started MOST adding Ford and Audi. Some of the carmakers are Volvo, Porsche, SAAB along with companies from the SC and others. The suppliers are component suppliers and companies such as Nokia, Siemens and Matsushita are represented in this group. The Working Groups are made of people represented from the associated partners on initiative from the carmakers who invite the suppliers to join. The Working Groups develop standardisations and recommendations in different areas. Some of the groups are WG Telephony, WG DAB and WG Bluetooth, which is a subgroup out of WG Telephony. The organization is illustrated in Figure 8.

## 3.3 Characteristics

The MOST network is based on the medium of optical plastic fiber. The bandwidth capacity of the system is 24,5 Mbps. The topology of the system could be star, ring or combined topology but the ring version is the commonly accepted solution. MOST has the possibility to transfer combined synchronous and asynchronous data at the same time as control data is transferred. Figure 9 describes the MOST Frame [16].

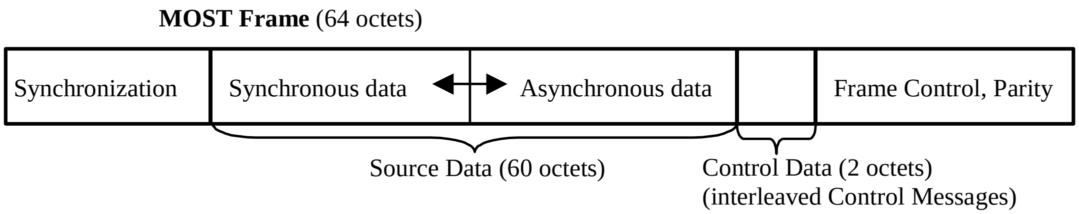


*Figure 9: MOST Frame structure*

### 3.3.1 Bandwidth

The arrow in Figure 9 indicates the possibility to shift the number of octets between the asynchronous and the synchronous area. The maximum limit of 60 octets give the maximum speed of 24,5 Mbps in the MOST network. The control data is locked to 2 octets that limit the speed to 768 kbps. A single control message needs a bandwidth from 10-100 kbps. Table 2 describes the demand of bandwidth for different types of data transfer.

For data transfer a bandwidth between 1 and 10 Mbps is required. Audio demands up to 4 Mbps uncompressed and up to 0,5 Mbps compressed. Video is more demanding and needs a bandwidth of 2-50 Mbps. In compressed representation video requires between 1 and 12 Mbps [17].

| **Data** | **Bandwidth (Mbps)** |
|---|---|
| Information (Internet, pictures, GPS) | 1-10 |
| Audio | < 4 |
| Compressed audio | < 0,5 |
| Video | 2-50 |
| Compressed video | 1-12 |

*Table 2:Bandwidth required for different data types*

To put the numbers in comparison a DVD movie that is compressed with MPEG2[1] needs a bandwidth around 4 Mbps.

Different types of information can be transmitted over MOST and the frame structure separates the data into three different sections: Control data, synchronous data and asynchronous data.

---

[1] MPEG2 is a standard for compressing video and audio.

### 3.3.2 Control data

The control data is transported to a certain address and is secured by CRC. Specific addresses make it possible to group- and broadcast data. Control data has an *ack/nak mechanism*[2] with automatic retry. It is suitable for transmissions of short packets and for use of low bandwidth around 10 kbps [16].

### 3.3.3 Asynchronous data

Asynchronous data is sent in a burst manner. Connections are administered via the control channel. MOST can handle transmission in need of a high bandwidth and with large packets [16].

### 3.3.4 Synchronous data

Continuous stream of data can be transmitted even if it may need a high bandwidth. The connection established to transmit synchronous data is done via the control channel. Different applications can request bandwidth in the system. The requests for bandwidth are recommended to be handled by a central unit, even though it is not necessary. The Function Block *ConnectionMaster* handles the administration of synchronous connections.

In the MOST Frame it is possible to change the boundary between the synchronous and the asynchronous areas. This can be done during the initialisation of the system. If it is done in an active network the synchronous connections must be rebuilt.

---

[2] Resend mechanism to ensure the quality of the transmission

## 3.4 MOST System Services

The MOST System Services provides all the basic functionality of the MOST system operability. MOST System Services are divided into different segments as shown in Figure 10 [16].

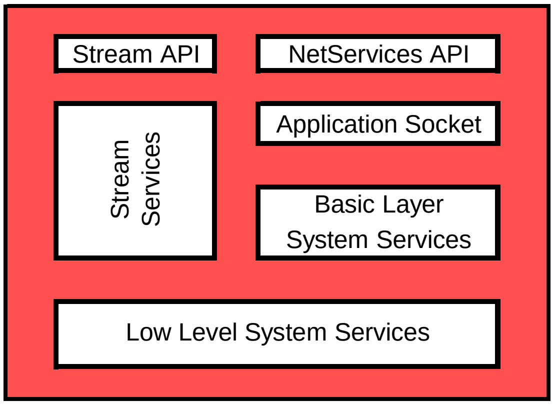


*Figure 10: MOST System Services*

Those segments includes different services and some of the most important ones are described below:

- The *Application Socket* handles addressing and describes what functions the device has.
- The *Basic Layer System Services* handles error checks, power management, segmentation and bus channel allocations.
- The *Low Level System Services* handles the clock, physical interface and the format conversion of different connections to and from the MOST device.
- The *Stream Services* handles the streaming media of the MOST network.

The *Application Socket*, the *Basic Layer System Services* and the *Stream Services* are implemented in the MOST Netservices. The *Low Level System Services* is implemented in the MOST Transceiver.

### 3.4.1 NetServices

MOST NetServices is organized into two layers where Layer 1 is the Basic Layer System Services and Layer 2 is the Application Socket. A mapping of the MOST system towards the OSI reference model is done in Figure 11 [23]. NetServices provides services to simplify the handling of the MOST Transceiver and is an intermediate layer between the MOST Transceiver and MOST FBlocks. See Figure 12.

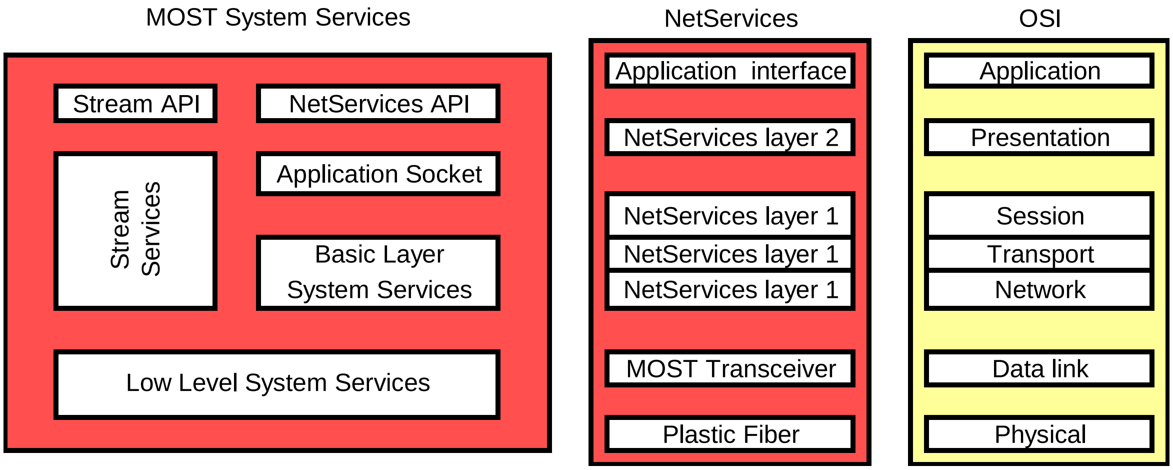


*Figure 11: MOST - OSI*

### 3.4.2 FBlocks

An application, such as a CD player, is controlled via an MMI. For the communication with the application the MMI uses Function Blocks (Fblocks) that is an interface towards the NetServices. The FBlocks describe the functions of an application such as “play” on a CD or “mute” on a loudspeaker. The MMI then controls the application using the FBlocks that transforms into communication via the Netservices and then gets recreated on the receiving side. A MOST device could contain multiple FBlocks. Each device has a special Function Block called NetBlock that describes the functions of the entire device. A MOST Device is shown in Figure 12.

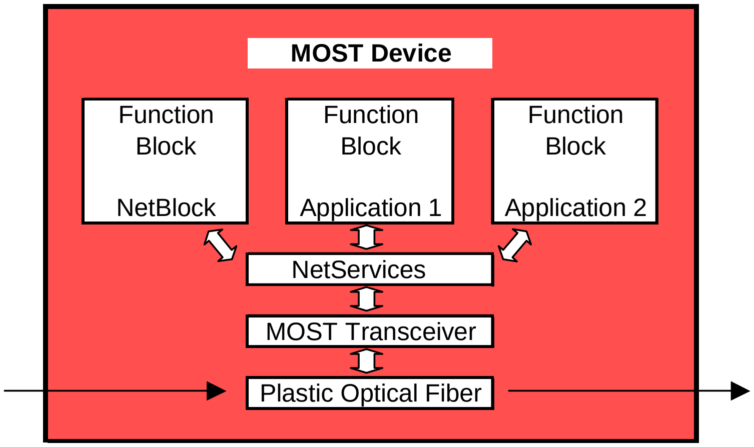


*Figure 12: Function Blocks in MOST*

Slaves, Human Machine Interfaces (HMI’s) and controllers are the three types of FBlocks existing. The slaves are always controlled FBlocks. HMI’s work as interfaces towards humans. Controllers combine multiple functions of different Fblocks and they can control and also be controlled.

#### 3.4.2.1 FBlock functions

Functions can be divided into two types. Those are methods and properties. Methods could be used to control other FBlocks and the definition of Methods is a function that can be started and which leads to a result after a definable period of time. Properties are functions used to read and set variables like limits or status. The variables could be heat, speed or volume.

Events are properties that change values without explicit requests. It can be used to notify that limits has been reached or the changed of values in function blocks. An example of this is the change of the time elapsed in a music track played by the CD player. Using Events eliminates the need of cyclical reading of the properties (polling) and reduces the communication between Function Blocks.

#### 3.4.2.2 Addressing an FBlock

A function is addressed with the FBlock ID, the Function ID (Fkt ID) and the operation type (OPType). This gives the following address structure:

FBlock ID. Fkt ID. OPType

Fblocks and Functions are identified with Fblock ID and Fkt ID respectively. The *OPType* specifies the operation of the function. In Table 3 the function *Connect* in FBlock *AudioAmplifier* is described. The parameters of the Functions are presented under *Parameter* and the operation types under *OPType.*

| Fblock | Fkt | OPType | Parameter |
|---|---|---|---|
| AudioAmplifier<br>(0x22) | Connect<br>(0x111) | StartResult | SinkNr, SrcDelay, ChannnelList |
| | | Processing | |
| | | Result | SinkNr |
| | | Error | ErrorCode, ErrorInfo |

*Table 3: Function connect of Fblock AudioAmplifier*

### 3.4.3 Low Level System Service

The transceiver described is the OS8104 from Oasis Silicon Systems. It is a low cost transceiver with low power consumption. Some of the features are the onboard Network Management that includes:

- Node position and delay detection
- Error reporting
- Automatic multimedia channel allocation
- Automatic wake up

The transceiver can handle a data rate of over 24,5 Mbps and can handle both synchronous and asynchronous data. It also has an independent 768 kbps control channel [15]. It is possible to connect to other applications via the clock manager, the data source port and the control port. Figure 13 shows the different interfaces of the MOST Transceiver.

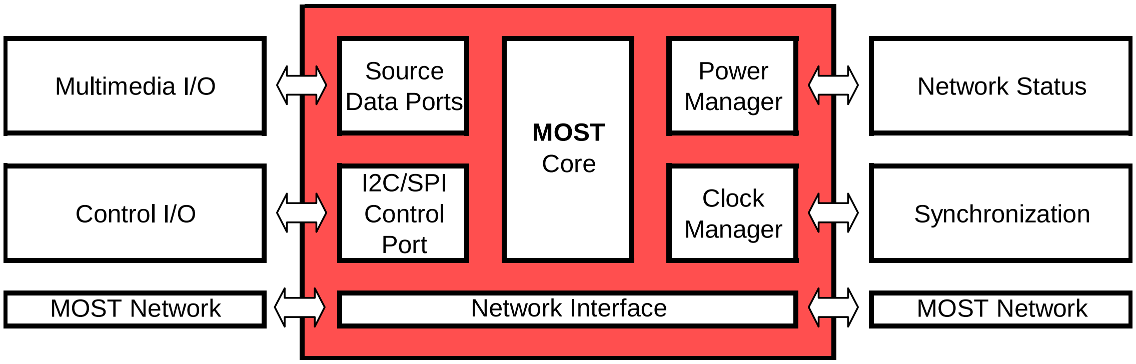


*Figure 13: MOST transceiver*

### 3.4.3.1 Clock

Each device can generate the network clock but only one acts as the *TimingMaster* and generates the clock and the frame structure of a network. All the other nodes in a network are slaves that synchronises to the *TimingMaster*. Various A/D and D/A converters, digital signal processors (DSP), and media players such as CD players and DVD players can be synchronized to the system.

### 3.4.3.2 Interface

The transceiver has several different interfaces. It can send and retrieve information via the Control Port, the Source Data Port or via the network interface. It is also possible to get the network status out of the Power Manager and to synchronize to other systems using the Clock Manager. The interfaces are shown in Figure 13.

### 3.4.3.3 Source data ports

Source data is data transmitted, transported and received in a continuous stream and in real time. The hardware interface for this is called source data ports. It can operate in serial or parallel mode. The Source Data Ports are typically connected to multimedia sources and sinks that handle audio and video streams. Some of the serial formats supported are $I^2S$, Matsushita, Sony and S/PDIF.

## 3.5 Future

The first cars with MOST implemented were the BMW 7-series 2002 models, which were shown on the market in the year of 2001. Other carmakers like Audi and SAAB have plans to release their first car models vehicles with MOST in the year of 2002. Volvo presents their city jeep XC90 in Q3 2002 and this is their first using MOST.

The standard is continuously developing and an improvement will soon be introduced. It is a development from MOST and will be backwards compatible. The newer standard will be able to handle much higher data rates.

MOST is the first standard for optical networks in vehicles and has shown to be an effective way to improve the multimedia interconnections. Ford, Toyota, BMW, DaimlerChrysler and more along with the major companies of car electronics supports MOST. Therefore it has every possibility to become the standard multimedia network for vehicles. The advantages of MOST, in form of unified interface for all applications towards the network, are probably one key issue when it comes to the speed of the growth of MOST. This brings design independent solutions.

# 4 Digital Signal Processing

In most cases digital signal processing is compared to classical analogue signal processing. This is actually not a good comparison because analogue signals and systems can take any value in the specific boundary. The digital signals are discrete in both amplitude and time (see Figure 14), which means the signals cannot get every single value between the boundaries. The advantages with digitally represented signals are that they are flexible, reproducible and are easy to process. The disadvantages are that they have a finite word length and resolution.

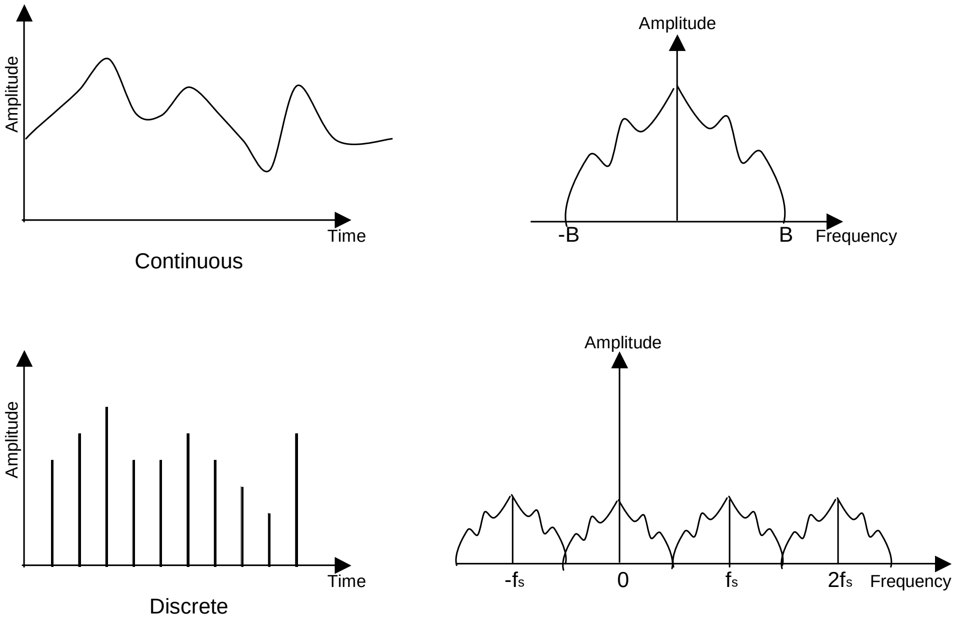


*Figure 14: Signal representations*

## 4.1 Sampling

In order to go from an analogue to a digital representation the signal has to be sampled. The sampling is done by an A/D converter. The A/D converter is connected to a sample clock, which decides the sampling frequency. To decide the sampling frequency the Nyqvist theorem is used.

*"A continuous-time signal with the bandwidth B Hz can without any loss be represented by values sampled with the frequency 2B Hz or higher."*

In this case all of the bandwidth energy is gathered in the frequency interval, which is

*B* Hz wide. The sampling frequency $f_s$ is 1 divided by the time between the samples *T* ($f_s$ =1/T). The desired sampling frequency *2B* Hz is called the Nyqvist frequency. If $f_s$<2B the sampled signal will be distorted and the original signal can not be regained, even with an ideal filter. If $f_s$>2B there will not be an improvement of the signal. The only reason to over sample is to make it easier for the non-ideal filter to regain the original signal.

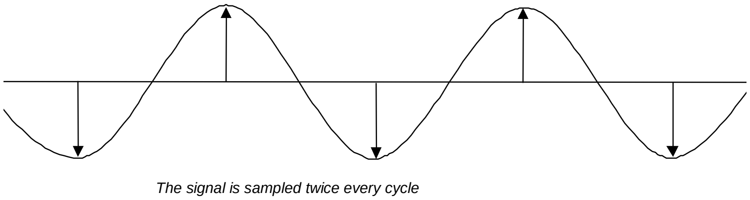


*Figure 15: The Nyqvist theorem*

### 4.1.1 Aliasing

Aliasing occurs when a sampling is performed with less than double the frequency of the highest frequency component of the analogue signal. As described before in the Nyqvist theorem samples have to be taken twice per cycle (see Figure 15). This means that if the frequencies are higher than half the sampling frequency they have to be removed before sampling to avoid aliasing. It is important to consider this when making CDs. In this case you have to remove frequencies higher than 22 kHz before sampling the music since the CD format has a sample rate of 44.1 kHz [6].

## 4.2 PCM

The most common way to digitalize audio is by Pulse Code Modulation (PCM) [7]. There are different PCM formats for different signals. Some are international standards and some are widely spread so called de facto standards. PCM does not just represent signals in linear form. For the PSTN (Public Switched Telephony Network) the signals are often represented in logarithmic form.

Our ear can in best cases hear frequencies from 20 Hz to 20 kHz [7]. To guarantee real reproduction of the signal it is essential to operate in this interval. Voice has most of its energy concentrated in a closer interval, 300-3400 Hz [7]. That is why Bluetooth and other telephony devices, due to the Nyqvist theorem, samples with the frequency 8000 Hz. Ordinary PSTN telephones quantifies with 8 bits logarithmic PCM when Bluetooth quantifies with 13, 14 or 16 bits/sample. 16 bits/sample is most common because Bluetooth mostly uses 64 kbps CVSD encoding for the air coding.

When it comes to High Fidelity there is a de facto standard for the compact discs. The CD’s has a sample rate of 44,1 kHz and quantifies the signal to 16 bits. MOST follows the PCM-standard for DVDs, 48 kHz 16 bits.

PCM is commonly used in synchronous systems such as CD players, MOST synchronous channel, Bluetooth voice etc. To transmit a sampled PCM signal from one system to another system with a different sampling frequency, a sample rate conversion of the signal would be required.

## 4.3 Sample rate conversion

To convert the signal from a sample rate of 8 kHz to 48 kHz it is possible to convert the signal from digital to analogue and to digital again. This method is applicable for any ratio of source sample rate to sink sample rate. See Figure 16.

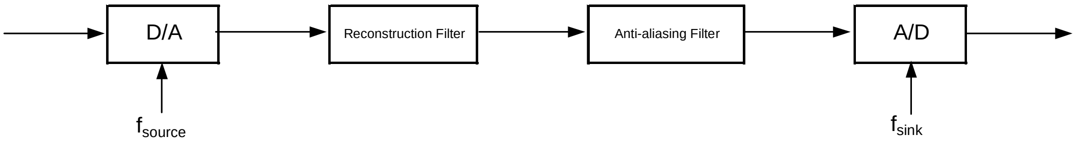


*Figure 16: A D/A-A/D sample rate converter*

This method has several limitations and problems. To retain large signal-to-noise ratio (S/N) of the digital audio signal, expensive and high quality A/D- and D/A-converters have to be used [19]. Another thing is that the output signal from the D/A-converter has to be reconstructed by an analogue filter of high precision with cut off frequency $f_{source}/2$.To fulfil the Nyquist theorem the input signal to the A/D-converter has to be band limited to $f_{sink}/2$ with an anti-aliasing filter. Besides those limitations, the need of expensive analogue filters and A/D- and D/A-converters, is that any jitter on the sampling clocks will translate into signal distortion.

In multirate systems the sampling frequency rate is changed during signal processing. It could be used to interface two systems with different sampling frequencies like MOST and Bluetooth. The problem is to do the sample rate conversion without destroying the information contained in the original signal. To digitally manipulate the signal there must be digital filters to retain the vital information of the original signal. The sample rate conversion is done with interpolation followed by anti-aliasing filtering, see Figure 17.

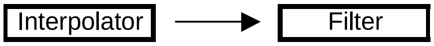


*Figure 17: Interpolation procedure*

The decimation procedure is reversed and the anti-aliasing filtering comes first and then the decimation, see Figure 18.

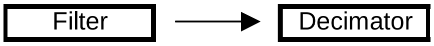


*Figure 18: Decimation procedure*

### 4.3.1 Converting with arbitrary numbers

Combining the decimation with interpolation makes it possible to convert frequencies with arbitrary numbers. To interpolate a signal with for example the factor of 5,5, it is possible to first interpolate with the factor 11 and then decimate with the factor of 2. To do this procedure the interpolation has to be done first so that none of the vital basic information gets destroyed. The filters between the interpolator and the decimator could be combined into one filter as shown in Figure 19.

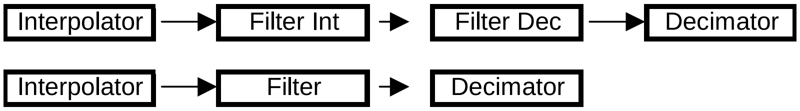


*Figure 19: Combined filter*

The stopband edge (see chapter 4.4) should then be pi(π) divided by the largest interpolation *(L)* or decimation *(M)* factor. See Equation 1.

Stopband edge: $\omega_s T = \min(\pi/L, \pi/M)$ [11]

*Equation 1*

It is difficult to perform a conversion between the CD standard of 44,1 kHz and the DVD standard of 48 kHz. The ratio is between two very large integers and the workload is high. This makes it very expensive. To solve this problem more advanced techniques are used.

### 4.3.2 Polyphase structure

This method is used to reduce the workload. Dividing the interpolation into several steps can do this. As an example the interpolation by 6 can be done by first interpolate by 3 and then by 2. A general polyphase structure has the same number of operations as a non-polyphase structure but the number of operations per second is reduced by the conversion factor [11].

## 4.4 Filter

When designing digital filters, it is common to use the great knowledge and experience of analogue filter design. This can be done by first create an analogue filter and then design a digital filter with the same characteristics.

### 4.4.1 Filter characteristics

The filter is used to separate desired from undesired frequencies. For most cases it is not possible to separate the different frequencies completely. Instead the signal (desired) to noise (undesired) ratio is tried to be made as large as possible. To achieve this, different parameters of the filter must be taken under consideration. Some of those parameters are the three sections of the filter and their relationship to each other. The three sections are the passband, the stopband and the transition band as shown in Figure 20.

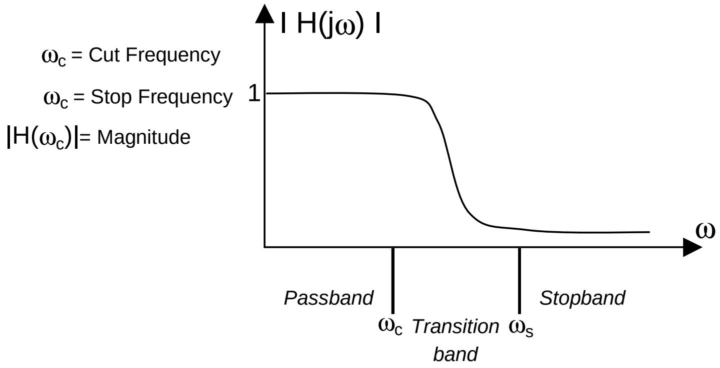


*Figure 20: A lowpass filter*

#### 4.4.1.1 Passband

The passband is the frequency band that is occupied by the desired signal. To transmit the desired signal in the passband with no distortion, the filter has to provide constant loss and group delay[3]. The passband frequency for an analogue lowpass filter is from 0 to $\omega_c$ , see Figure 20 [11,12].

#### 4.4.1.2 Stopband

The stopband is the frequency band occupied by the undesired signals. In a digital lowpass filter the stopband begins at $\omega_s$ and goes to Pi($\pi$). It is important not to use more stop band loss then necessary when designing a filter. This is due to the costs of making a narrow complex stop band and also the fact that the passband will be affected as the stopband changes. It is not always possible to know how to set the minimum stopband for all frequencies and therefore a constant loss for all frequencies is set. This constant level is considered to be a safe level.

---

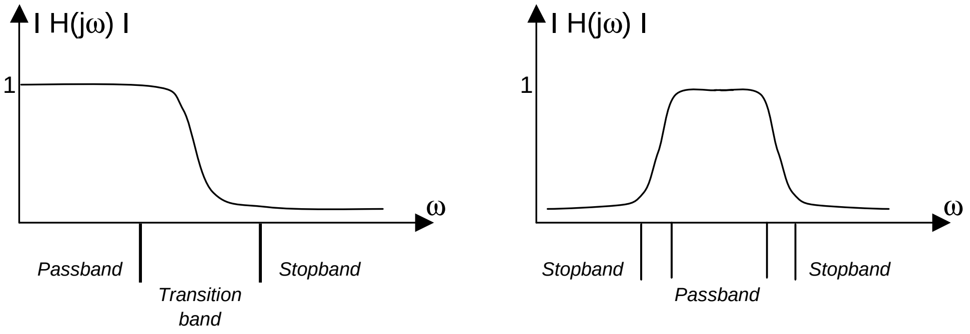
[3] Group delay is the frequency delay associated with the processing of the signal [11]

### 4.4.1.3 Transition band

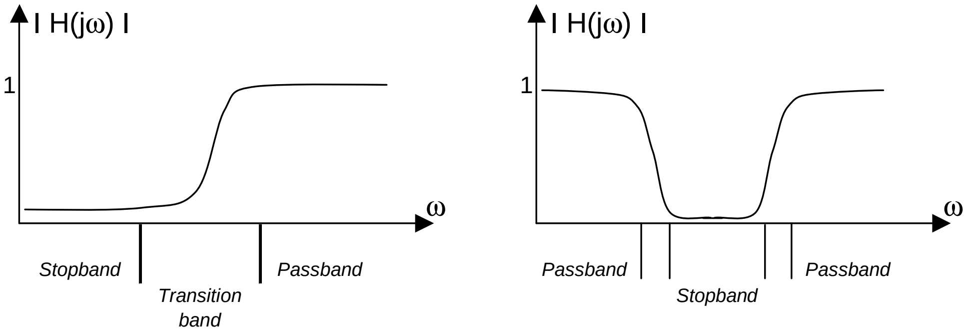
The frequency band between $\omega_c$ and $\omega_s$ is called the transition band , see Figure 20. This band separates the edges between the stopband and the passband. The bandwidth between those edges is one of the main factors when designing the size of the filter. Although it is better with a narrow transition band, the complexity of the filter grows as the transition band is narrowed. The slope of the curve must also be very steep towards the edge of the passband to avoid loss.

## 4.4.2 Different types of filters

There are five different kinds of filters [12]. Lowpass- (LP), highpass- (HP), bandpass- (BP), bandstop-(BS) and allpass- (AP) filters. Filters are used in all kinds of electrical equipment. A lowpass filter allows frequencies up to the cutoff ($\omega_c$) to pass. All other frequencies are rejected. This is used on a receiver to set the treble. The bass control corresponds to a highpass filter that rejects the frequencies below the cutoff frequency. The tuning procedure in an AM radio is an example of a variable bandpass filter [8]. Those are some examples of filters used in every day equipment. The different types of filters are shown in Figure 21 and Figure 22.



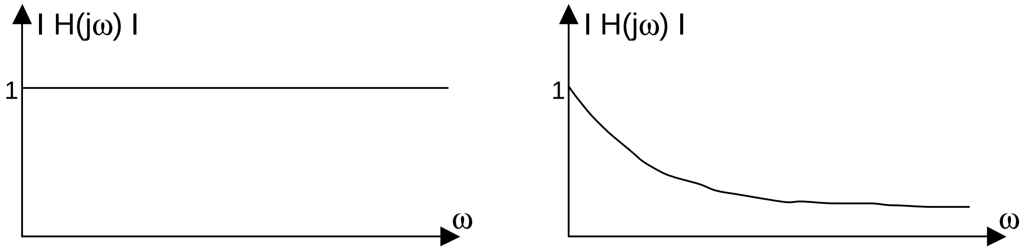
**Lowpass filter** **Bandpass filter**



**Highpass filter** **Bandstop filter**

*Figure 21: Different filter types*

The allpass filter lets all the frequencies pass and most often with no absorption. The filter is still used since the phase response will affect the signals and different frequencies will be shifted differently [4]. The allpass filter and its phase curve is shown in Figure 22.



**Allpass filter** **Allpass filter, phase curve**

*Figure 22: Allpass filter*

When designing filters the hardest thing is to approximate the ideal response depending on the requirements set for the filter. Most of the work with approximations has been put into the design of analogue lowpass filter. This is because it is possible to make other filters by frequency transformation [12]. It is also possible to make digital filters from those results. Some of the most famous standard filter types are Chebychev, Cauer and Bessel [12].

## 4.4.3 Digital filters

To use a digital filter the signal which is filtered has to be digital. This means that the signal is sampled into a discrete time signal before filtering. When it comes to audio the signals are often represented by a PCM signal. Digital filters are added to remove noise, distortions etc. The advantages with digital filters versus analogue are that they are more flexible, sensitive and robust because of their digital structure.

### 4.4.3.1 Filter types

The most common filters are the frequency selective filters. They are causal, linear and time invariant systems [6]. Digital filters are categorized as finite (FIR) or infinite (IIR). The discrete-time filters can be described by difference equations like in Equation 2.

$$\text{output}(t)=a_0*\text{input}(t)+a_1*\text{input}(t-1)+a_2*\text{input}(t-2)$$

*Equation 2: A ” three tap” FIR filter*

The “input” values are sample values fed to the filter, t is the time and $a_0$, $a_1$ and $a_2$ are filter coefficients.

#### 4.4.3.2 FIR (Finite-time Impulse Response)

FIR filters are well suited for applications with multirate systems like interpolation etc. The advantages with FIR versus IIR filters are that they are guaranteed to be stable and have a linear phase-response. Fast Fourier Transforms are often used when realizing FIR filters. FIR filters use previous output values to calculate the present.

#### 4.4.3.3 IIR (Infinite-time Impulse Response)

IIR filter are developed from analogue prototypes. The analogue prototypes are mapped into digital filters by linear transformation. An IIR filter is generally more effective than a FIR filter. The number of multiplications, additions, subtractions and delay elements are often less [12].

## 4.5 Interpolation methods

When choosing an interpolation method computation time, quality and memory usage are considered. The choice is often a compromise between these three aspects. Another aspect is to see what kind of filter that has to cut of unnecessary frequencies in the frequency domain.

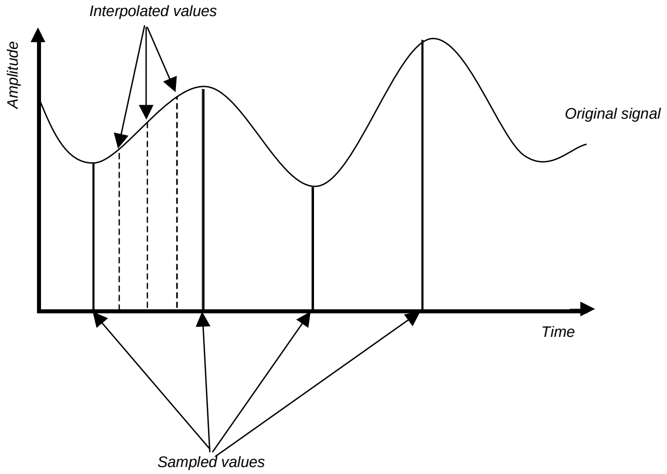


*Figure 23: Interpolation of a signal*

Figure 23 shows an ideal interpolation where the interpolated values follow the original signal. One of the most important issues when it comes to interpolation of speech is the quality. Much of the distortion added in the interpolation can be removed with a filter.

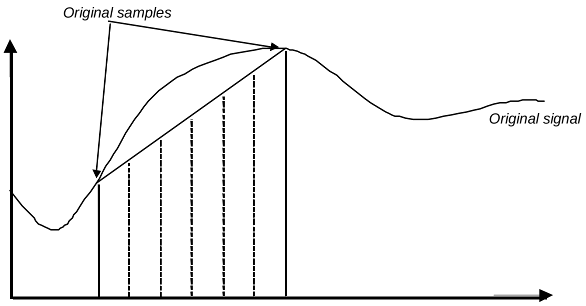
### 4.5.1 Linear interpolation

Linear interpolation (see Figure 24) is one of the simplest methods. Straight lines are drawn between the two adjacent sample points and the new sample points are put on the line. The interpolation becomes a little edgy but will be acceptable for speech. It is easy to interpolate between frequencies with a rational quotient.

$$r = f_{sink} / f_{source}$$

*Equation 3*

In Equation 3 $f_{source}$ is the source sampling frequency and $f_{sink}$ the sink sampling frequency. If the quotient r is rational it is just to put r-1 samples between the original samples.



*Shows a signal interpolated with the quote 6 (the broken lines are the samples put in after the interpolation)*

*Figure 24: Linear interpolation*

When converting a signal sampled with 8 kHz to 44,1kHz the quote is non-rational and interpolation becomes a little trickier. The most common way to solve this problem is to first interpolate to a higher value and then decimate the signal so it represents 44,1 kHz. This is described more in chapter 4.3.1.

### 4.5.2 Sinc interpolation

Sinc[4] interpolation is said to be a perfect interpolation method [18]. Replacing all the sample points with correctly scaled sinc curves (see Figure 25) makes the interpolation almost perfect. A sinc curve is infinite, therefore you need all the sample points to calculate one interpolated value. In practice the number of sample points is set to around 1000, which gives an acceptable interpolation [18]. That is still to heavy for a real-time application but it will give a result with high accuracy. The sample points have to be quite few but at least six if it is implemented in a real-time implementation [18]. It is common to use linear interpolation to provide virtual continuity when performing sinc interpolation.

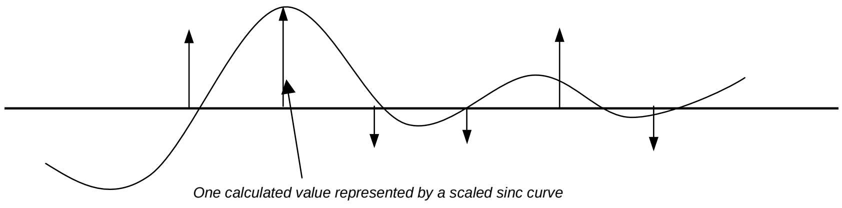


*Figure 25: Sinc curve*

### 4.5.3 FFT interpolation

Interpolation with help from Fast Fourier Transform (FFT) is a very flexible way to interpolate a signal. FFT is often used in digital filters and can in quite an easy way be used for performing interpolation as well. The interpolation algorithm transforms the discrete-time signal from time domain to frequency domain. In the frequency domain the spectrum is analysed and calculated for the new sampling frequency. After that the

---

[4] sinc(x)=sin(x)/x

algorithm retransforms the signal back to the time domain. The signal now has the number of samples needed for playing it in the higher frequency.

In order to interpolate the signal with FFT interpolation a large number of samples is needed to get a good result.

### 4.5.4 Zero-filling interpolation

An easy way to interpolate is to insert samples with zeros between the original samples. As in linear interpolation this is a good method for interpolation between sampling frequencies with a rational quotient. Combining interpolation with decimation makes it possible to convert frequencies with any arbitrary quotients. After interpolation the spectrum of the sequence do not only contain the base band of the original signal, but also repeated images of the base band (see chapter 4.1.1). The repeated images will distort the outgoing signal. To remove undesired images a digital lowpass filter is used . The procedure of zero-filling interpolation and filtering is shown in Figure 26.

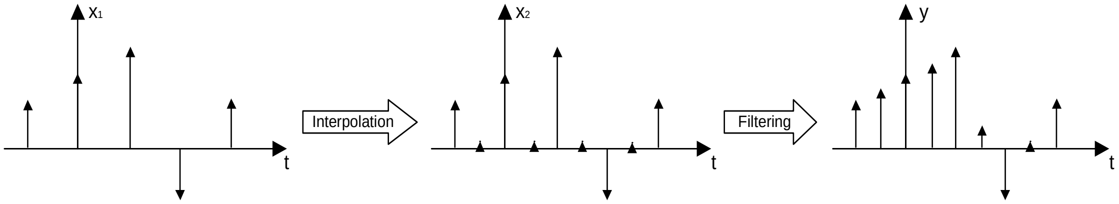


*Figure 26 : Zero-filling procedure*

# 5 System design of audio source

For the possibility of the implementation of the Bluetooth-MOST gateway the audio source have to be designed first. This is because the gateway needs to know the preferences of the audio source. According to the task, one property of the audio source should be the possibility to send control data that establishes a synchronous connection for transferring voice data. The Headset Profile of the Bluetooth specifications fulfils this demand and fits as an audio source.

In order to establish a Bluetooth headset connection two roles are defined, the Audio Gateway (AG) and the Headset (HS). The Audio Gateway can be a personal computer or a cellular phone and the headset is the actual wireless headset that remotes the AG. Both devices have both input and output audio.

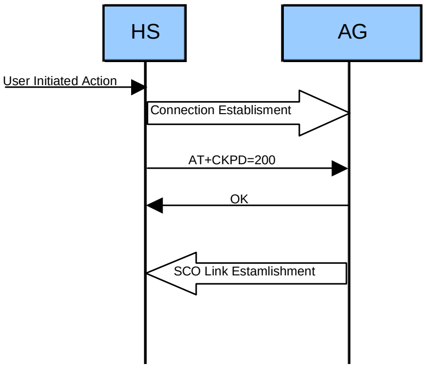


*Figure 27: HS - AG connection establishment*

In Figure 27 the headset initiates a connection. There are several other AT commands that perform different operations such as volume control and close connection. The AT commands are reused from the ITU-T recommendation V.250. Some of the AT commands are presented in Table 4.

| AT capability | Syntax | Values |
|---|---|---|
| RING | RING | - |
| Microphone gain | +VGM=<gain> | 0-15 |
| Speaker gain | +VGS=<gain> | 0-15 |
| Headset button press | +CKPD=200 | - |

*Table 4: HS Profile AT-commands*

## 5.1 Functional requirements

An easy description of this source should have a control button, a microphone and a speaker. The control button will be a button on the headset that will initiate the connection with the MOST system. When pushing the headset button during a transmission the connection should be closed down.

## 5.2 Audio requirements

As mentioned above the Bluetooth Headset Profile is chosen as the audio source. The Headset Profile uses the CVSD codec as the air transmission format. In order to ensure the quality of the sound the digital audio has to fulfill some requirements. The signal level of the 16 bit linear PCM signal fed to the CVSD encoder is defined to be 3dBm0. The spectral power density of the PCM signal must be above 4 kHz.

The spectral power density in the 4-32 kHz band of the output signal of the CVSD decoder should be more than 20 dB below the maximum amplitude in the 0-4 kHz band. In Figure 28 a scheme of the Bluetooth audio subsystem is shown.

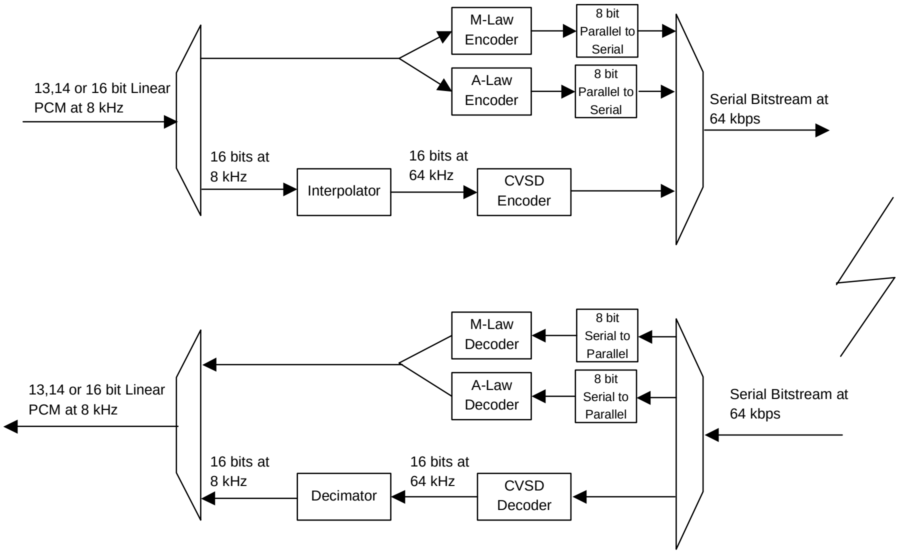


*Figure 28:Audio subsystem*

# 6 System Design of Gateway

The design of the gateway is divided into two sections. The first section is a specific gateway describing a gateway dedicated for transferring audio from the Bluetooth headset to the MOST speaker node. The second section is a description of a general gateway that could be used for several different areas using a Bluetooth device with MOST.

## 6.1 Specific Gateway

There are many things to take under consideration when designing a gateway. Should it be a repeater that just copies the information from one physical network to another, a router that forward packages with information or an interpreting gateway that totally changes the information into formats that the networks can understand?

To find the approach of the solution the two systems are compared and the differences are presented in Table 5.

| **Comparison** | **MOST** | **Bluetooth** |
|---|---|---|
| *Medium* | Fiber | Radio |
| *Bandwidth* | 24,5 Mbps | 723 kbps |
| *Topology* | Ring | Piconet |
| *Nodes* | 64 | 8 |
| *Sample rate* | 48 kHz | 8 kHz |

*Table 5: Comparison of MOST and Bluetooth*

The differences of importance for this gateway are the medium of radio for Bluetooth and fiber for MOST and also the sample rate frequency of 8 kHz for Bluetooth and 48 kHz for MOST.

The systems are also mapped against the OSI reference model to more easily see were to find the gateway design problems [3]. For synchronous audio the transmission is in the link layer but for the control data the approach is to design an interpreter on the application level. In Figure 29 the mapping towards the OSI reference model is shown.

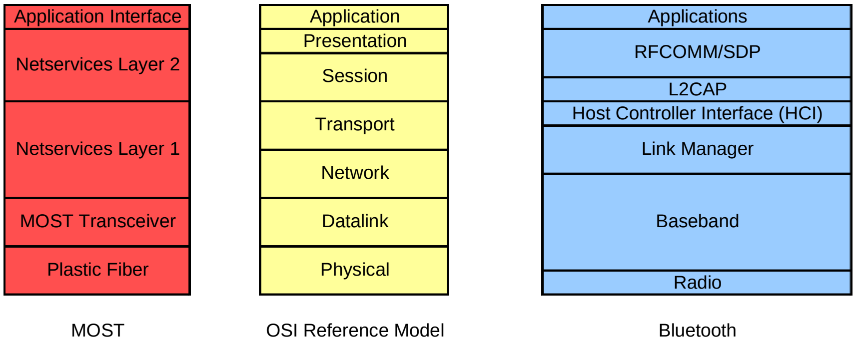


*Figure 29: Mapping towards the OSI Reference Model*

The conclusions are that two totally different networks are to be connected, one wireless and one fibre optical network. This makes it harder to connect the networks on a lower level like some kind of lower level router so the higher-level approach is most suitable for this gateway.

### 6.1.1 Higher Level Gateway approach

To let the system be unaware of one another's existence is the idea of this approach. This transparent system would need an interpreter that handles the used Bluetooth profile procedure and converts them to the corresponding MOST procedures. This would make it easy to implement nodes on both sides of the gateway but would make the gateway more complex with a Bluetooth stack and support for the MOST System Services.

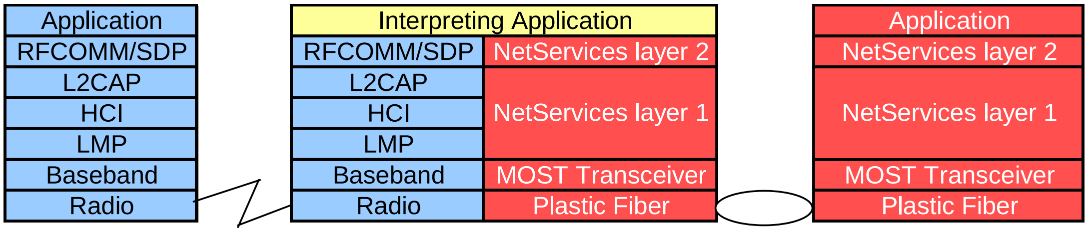


*Figure 30: Bluetooth – MOST Gateway structure*

As seen in Figure 30 the drivers for the gateway is called upon by the application. The application in the gateway handles the actual Bluetooth profile and interprets it to functions and methods understandable by the MOST network nodes. There are two different kinds of data transferred between the systems when sending audio from Bluetooth to MOST. The connection has to be established using control data. When this is done the stream of synchronous audio is transferred. The data in both cases has to be modified to fit the other system. The handling of data transformation is described separately for control data and synchronous audio.

## 6.2 Control data

The establishment of the connection should take place as if the connection were between two Bluetooth sources or between two sources in MOST. This approach of gateway connection problem makes it possible to connect as usual with standard procedures. Procedure of connection and disconnection of establishment and error handling of those procedures are presented below.

To establish a connection there are three different procedures to take under consideration:

A) The communications from Bluetooth headset to the Gateway(GW)
B) The Gateway interpretation of the message
C) The communications from the Gateway to the MOST amplifier

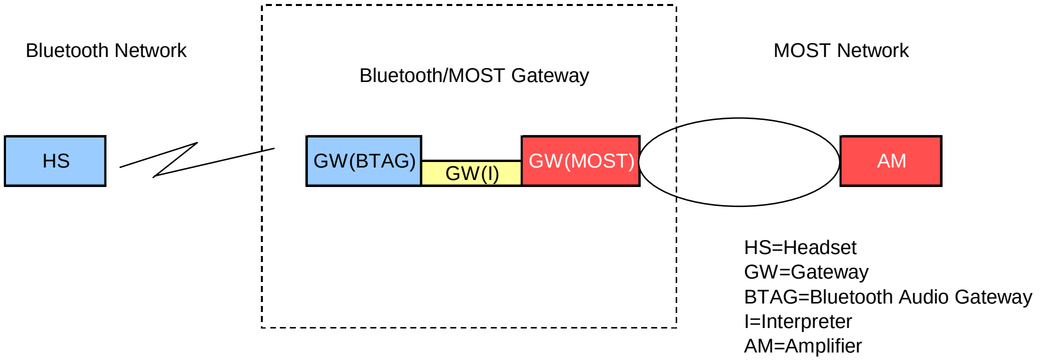


*Figure 31: Bluetooth – MOST Gateway*

In the A procedure the communication is handled as a transparent Bluetooth connection. The headset profile in the headset talks to the headset profile in the gateway as if it were a Bluetooth Audio Gateway (AG) (see Figure 31). The gateway interprets the commands so that the MOST system makes the connection independently of what happens on the Bluetooth side.

### 6.2.1 Establishing synchronous connection from Bluetooth headset

The steps in each procedure are shown in Figure 32 and explained in chronological order:

A)

1. User Initiated Action (Button push)
2. Connection Establishment
3. AT- command “AT+CKPD=200” is sent HS  GW (AG)
4. Audio Gateway replies “OK” HS  GW (AG)
5. SCO link establishment

B)

1. Gateway (GW) receives the HS control message as in A.3
2. Gateway interprets the received AT-command to MOST method “BuildSyncConnection”
3. “BuildSyncConnection” is sent to GW (MOST)
4. “BuildSyncConnection OK” is replied
5. Mute is sent to activate the speakers (mute activates or deactivates the speakers depending on the state their in)

C)

1. GW (MOST) starts the connection procedure with the Amplifier
2. The property “Mute” is used by the GW (MOST) to mute off the amplifier

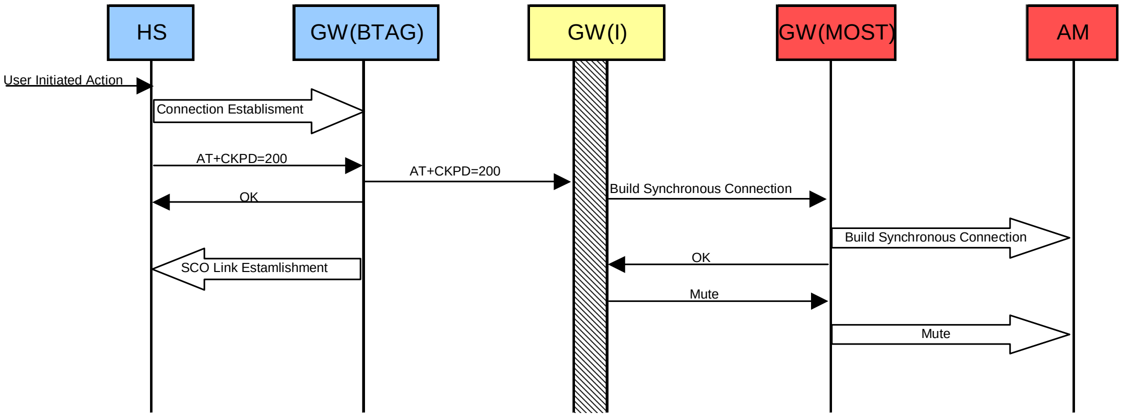


*Figure 32: Procedure of establishing connection.*

The chosen solutions are built from the assumption that the connection establishment is handled independently for each system. This assumption is based upon the theory that Bluetooth cannot wait the time it takes for MOST to establish a synchronous connection, to establish a connection of its own. The scenario describes the way of the communication when no errors are reported.

### 6.2.2 Disconnecting from Bluetooth headset

When disconnecting a connection there are three stages to be considered. The tree stages are:

A) The communications from Bluetooth headset to the Gateway(GW)
B) The Gateway interpretation of the message
C) The communications from the Gateway to the MOST amplifier

The procedures are described in Figure 33.

A)

1. User Initiated Action (Button push)
2. AT command "AT+CKPD=200" is sent from headset HS GW (BTAG)
3. GW Bluetooth Audio Gateway(BTAG) answers "OK" GW HS
4. SCO release procedure is initiated by the GW(BTAG) GW HS
5. The Connection release procedure is initiated by the GW(BTAG) GW HS

B)

1. Gateway Interpreter (I) receives the HS control message sent in A.2
2. GWI checks status on connection.
3. GWI interprets the received AT command to MOST method "RemoveSyncConnection" and MOST property "Mute"
4. OK is replied from GW (MOST) to GW (I)
5. "RemoveSyncConnection" is sent to GW(MOST)
6. Property "Mute" is send to GW(MOST)

C)

1. The Method "RemoveSyncConnection" is initiated by the GW(MOST)
2. The property "Mute" is sent to mute the amplifier

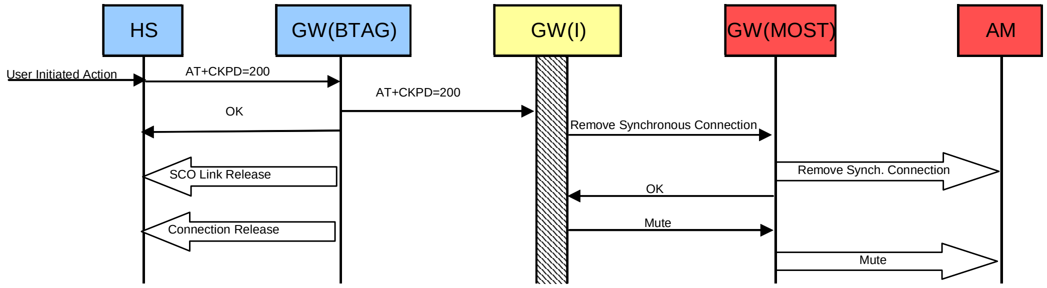


*Figure 33: Procedure of closing the connection*

### 6.2.3 Establishing synchronous connection from MOST

The procedures are described in Figure 34. The steps in each procedure are explained in chronological order:

A)

1. Command ring is received by GW (BTAG)
2. Connection Establishment GW HS
3. AT Command Ring is sent from GW HS
4. User answers with button push
5. AT command “AT+CKPD=200” is sent from headset HS GW (BTAG)
6. Gateway answers “OK”
7. SCO link establishment

B)

1. Gateway receives Ring command and interprets it into AT command
2. The command for Ring is sent to GW (BTAG)

C)

1. MOST initiated action
2. Command for ring is sent to GW (I)
3. GW (MOST) starts the connection procedure with the Amplifier
4. The property “Mute” is sent by the GW (MOST) to mute off the amplifier

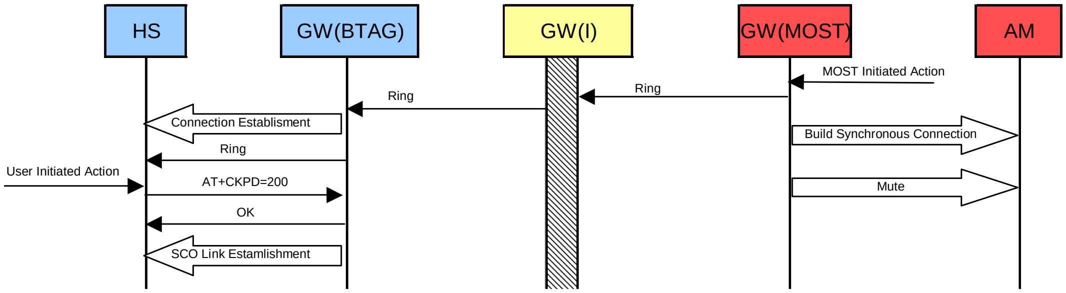


*Figure 34: Procedure of establishing connection.*

### 6.2.4 Disconnecting from MOST

When disconnect a connection there are three stages that has to be considered. The tree stages are:

A) The communications from Bluetooth headset to the Gateway(GW)
B) The Gateway interpretation of the message
C) The communications from the Gateway to the MOST amplifier

The procedures are described in Figure 35.

A)

1. Release command is received by GW (BTAG)
2. SCO release GW HS
3. Connection Release GW HS

B)

1. Gateway receives release command and interprets it into AT command
2. The AT command for release is sent to GW (BTAG)

C)

1. MOST initiated action
2. Command for release is sent to GW (I)
3. The Method "RemoveSyncConnection" is initiated by the GW(MOST)
4. The property "Mute" is sent to mute the amplifier

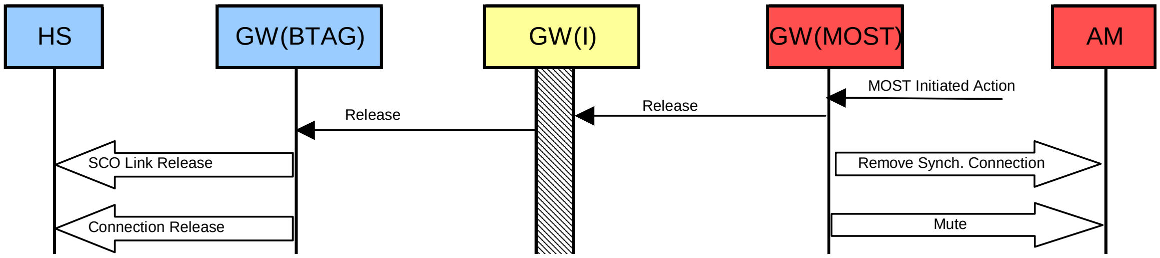


*Figure 35: Procedure of closing the connection*

# 6.3 Synchronous Audio

The main assignment of this Master Thesis is to transfer synchronous audio from a Bluetooth device to a MOST speaker node. In order to describe a solution for this the way of the audio signal through the entire system is presented. The system is illustrated in Figure 36.

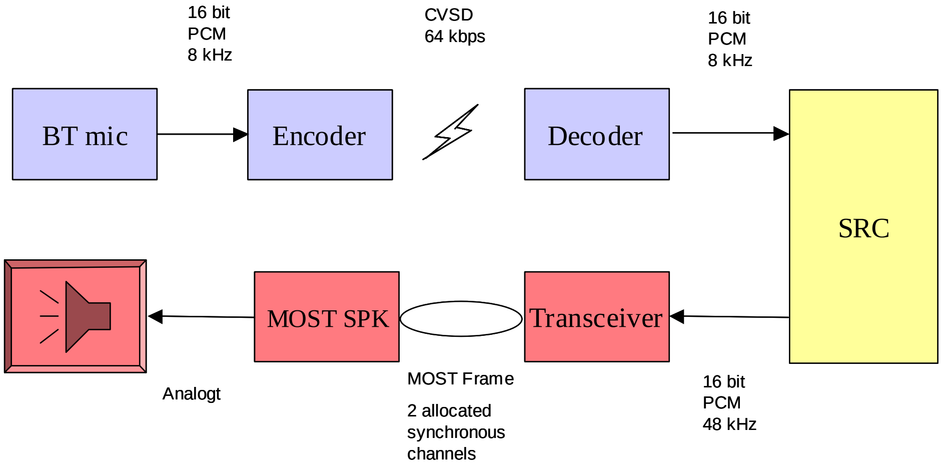


*Figure 36: The audio signal through the entire Bluetooth – MOST system*

## 6.3.1 Bluetooth part

The first part of the system is a Bluetooth device using the Bluetooth Headset Profile that is chosen as an audio source for the system. This was the first part of the assignment and is described in chapter 5. The Bluetooth microphone samples the analogue audio signal with 8 kHz 16 bits PCM. The sampled digital PCM signal is converted in the headset encoder to 64 kbps CVSD format and transmitted through the air. The decoder, which is a part of the Audio Gateway, decodes the CVSD signal back to the original format, 8 kHz PCM.

## 6.3.2 SRC part

The SRC part is divided into the three separate procedures of interpolation, parallel to serial conversion and synchronization (see Figure 37).

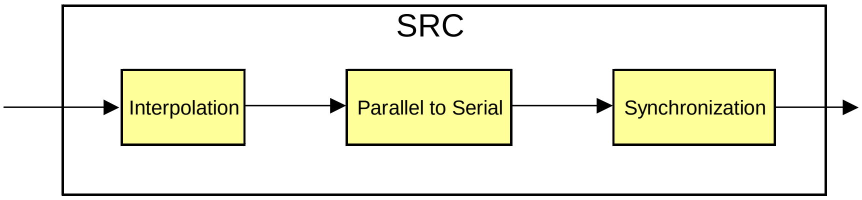


*Figure 37:SRC part*

### 6.3.2.1 Interpolation

Since the two systems use different sample rates a sample rate conversion is required. MOST uses both 44,1 kHz and 48 kHz, which demand either a flexible solution or a static sample rate on MOST. Solutions for both sample rates have been investigated but the focus has been on 48 kHz as the MOST sample rate. This is because 48 kHz is the standard sample rate of DVD and is generally accepted as MOST standard.

In the first solution the signal is converted from 8 kHz to 48 kHz. This means that the audio signal has to be interpolated 6 times. This is done using zero filling as the interpolation method, see chapter 4.5.4, and a lowpass FIR-filter that reduces aliasing effects and overtones. The lowpass filter cuts off the signal’s component frequencies above 4 kHz. Depending on what is set as maximum delay time, different orders of the filter are used. A higher order gives a sharper edge between the passband and the stopband but demands more complex calculations. Zero filling is most suitable for this solution since this interpolation do not require heavy calculations and because the sound of the audio signal is acceptable after the filtering process. It is possible to use this method for converting between 8 and 44,1 kHz using conversion between arbitrary numbers, see chapter 4.3.1.

The second solution is an interpolation with FFT. This will not require a filter because the interpolation is performed in the frequency domain. It has several advantages since there is no need for filtering and it is more flexible due to that it does not require fixed sample rate and can convert between any arbitrary numbers. The calculations are though very heavy and this interpolation method is not to recommend in a real implementation.

### 6.3.2.2 Parallel to serial conversion

When the signal is converted to the correct sample rate it has to be transformed into a serial format that is compatible to the MOST Transceiver. The PCM signal from Bluetooth is in parallel form from the beginning. It is possible to set the transceiver in parallel or parallel combined mode but since it is synchronous audio that is transferred between the systems, the serial mode has been found the most suitable. The parallel combined mode would make the gateway more flexible and able to handle both synchronous and asynchronous data. The gateway design would be different compared to the gateway described in this Master Thesis but is recommended in a real implementation since it could be used for any type of data.

There are several serial source port formats supported by the transceiver and it is also possible to set different parameters to create own formats. The standard format of $I^2S$ has been chosen for this solution since it has been used in earlier projects at Volvo Technological Development and because it is the serial format best described in the MOST specifications. Other formats would work as well. The Transceiver reads the signal and then places the correct bits at the right place in the MOST Frame. Different source data port formats place the bits of information differently in the signal. This is why the signal has to be in a special, for the transceiver, known format. The $I^2S$ format is shown in Figure 38 and some of its parameters are described below.

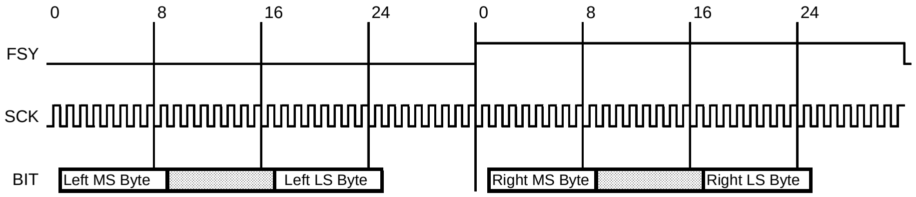


*Figure 38 $I^2S$ source data format*

The *SCK* is the bit clock. The *FSY* is the frame sync and the *Bit* is bits containing the information. In $I^2S$ the FSY, which could be high or low, is low when reading the information of the left channel and high when reading the right channel. The most significant bit is read first and the least is read last. This format is also shifted one bit relative to the FSY. Possibly SCK modes for $I^2S$ is 8/16/24/32/48/64/128/256 times the sampling frequency, Fs.

In other formats the different information bytes are located in another place in the signal. The shifting and the FSY could also vary between different source port formats.

#### 6.3.2.3 Synchronization

The two systems are both able to handle synchronous data. To keep the system synchronous there has to be clocks to synchronize the networks. The gateway between the systems has to consider both clocks in order to make the sample rate conversion.

The clock on the Bluetooth part is generated and supplied by the master in the Bluetooth network and is used by the synchronous 8 kHz PCM bit stream. The DSP device[5] uses the Bluetooth clock to synchronize the bitstream. When the bit stream has been converted into 48 kHz it has to be synchronized to the MOST network. The synchronization clock on the MOST Gateway can be obtained by the network. To synchronize the bit stream the MOST synchronization clock is obtained from the transceiver by the DSP device. To get the data in phase a buffer is used after the conversion procedure. The design of the gateway must see to that the buffer never gets empty and that the stream stays synchronized without losses. When the transceiver then gets the samples, the bit stream is in phase and can be transmitted with the correct bit rate into MOST. The synchronization procedure is illustrated in Figure 39.

---

[5] A general digital signal processing device. Could be a DSP, FPGA etc.

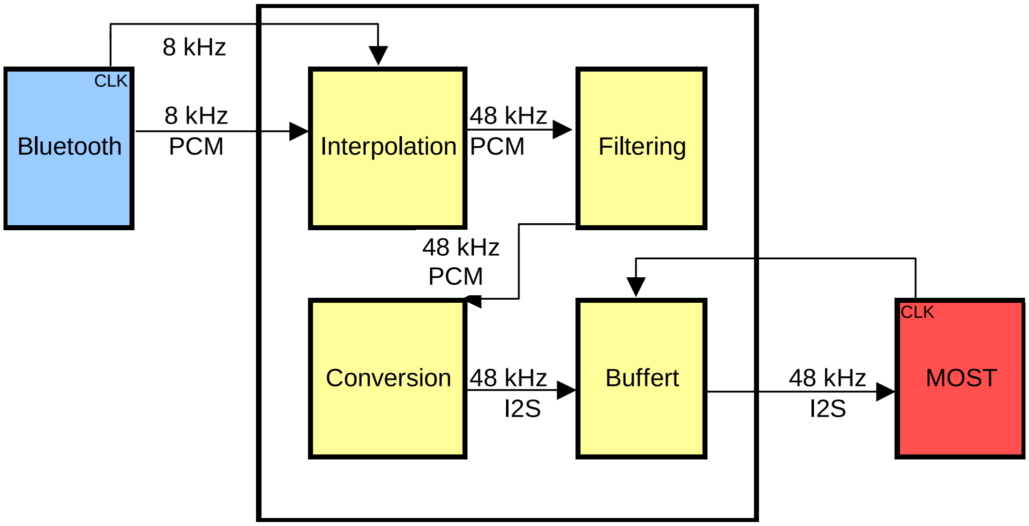


*Figure 39: Synchronization of Gateway*

### 6.3.3 MOST part

The transceiver in the MOST system reads the $I^2S$ signal and puts the 16 bit samples into the allocated synchronous channels in the MOST Frame. There are four synchronous channels allocated between the transmitting and the receiving node in which the audio is transmitted. The synchronous audio is from the beginning sampled in mono but then transformed into stereo. Information in a stereo signal needs 32 bits and 4 MOST channels are allocated since one channel corresponds to 8 bits. Figure 40 shows how the synchronous data could be placed in the MOST Frame.

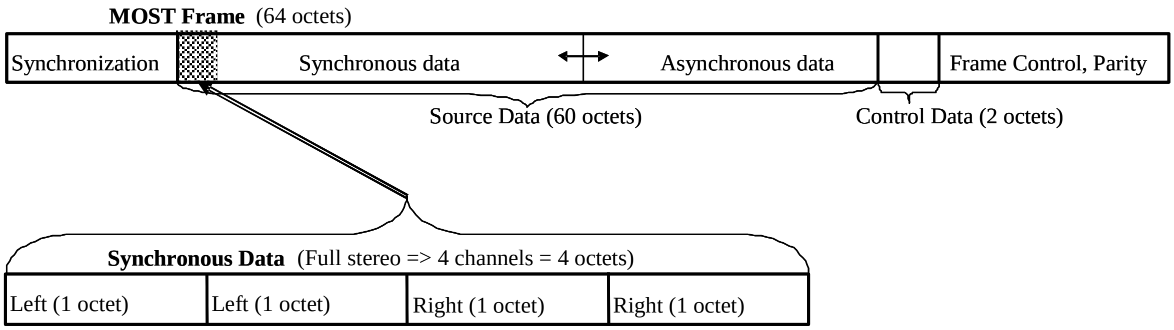


*Figure 40: The placement of audio bits in MOST*

In the MOST speaker node the digital audio stream is received and made analogue before it is sent to the speaker.

## 6.4 General Gateway

A general gateway should be flexible and be able to handle conversions between any different sample rates. It should also be able to support different types of data transfer and in both serial and parallel mode and support full duplex data exchange.

### 6.4.1 Scenarios

To connect Bluetooth and MOST it is adequate to have a gateway for several functions. Finding and describing those functions is found by rational discussions. We have looked upon existing as well as future systems to determine our use cases. The following chapters will describe those and how Bluetooth can improve the usability of the MOST system.

The general gateway should be designed to fit all use cases. There are three different kinds of data to manage:

- Control data
- Audio (synchronous transmission)
- File transmission (asynchronous transmission)

Those types of data need to be handled differently in the gateway and therefore it must be designed to manage all of the data types. The connection is also differently established depending on if it is synchronous data or asynchronous data that are being transferred. The control data has to be translated in the gateway from MOST language to Bluetooth language and vice versa. The gateway has to be programmed so that the command sent from Bluetooth gets interpreted correctly on MOST and that a Bluetooth command from MOST gets interpreted to the correct command.

The audio in the two systems is sampled with different sample rates so there has to be some sort of sample rate conversion. The gateway should be able to handle both interpolation and decimation. MOST uses both the sample rate of 44,1 kHz and 48 kHz and Bluetooth use 8 kHz. The gateway should therefore be able to handle interpolation and decimation between both 8 - 44,1 kHz and 8 - 48 kHz. The Audio format type also has to be in a form that both the systems recognizes, see 3.4.3.2.

Packet data transferred between MOST and Bluetooth also has to be handled by the gateway to fit the frames of both the systems. This area has not been investigated in this Master Thesis and is therefore only mentioned.

#### 6.4.1.1 Mobile phone interconnection

Today you have to have specialised docking stations to interconnect the phone with the hands-free system in the car. Many car producers have tried to solve this problem by integrating a phone in the car. This brings another problem since there has to be two SIM-cards[6], one for the mobile phone and one in the car. With Bluetooth integrated those problems can be eliminated. In a future scenario it is possible for the

---

[6] A SIM card acts as a key used to connect to the Mobile operator.

user to just enter the car without physically docking the phone and still be able to use the built in multimedia system. This means that the user will be able to use the car phone hands-free, display and keypad to control and interconnect with the mobile phone. The user could have the phone packed in a bag in the trunk of the car and still not miss a single call. The phone connects wirelessly to MOST via the Bluetooth gateway and is independent of the phone brand.  This is standardized by the Bluetooth SIG Car Profile Working Group. In MOST the WG Bluetooth works with the mobile phone interconnection.

#### 6.4.1.2 Voice control

The second use case is the improvement of voice control that Bluetooth can make possible. The progress in the field of voice recognition has made it possible to control the car applications by just using commands. This helps the driver who can use both hands to control the vehicle and still manage the multimedia products of the car by using the voice. A Bluetooth interconnection could expand those possibilities so that the vehicle could be controlled from outside the coupe´. Turning on and off lights and music, closing or open windshields and sunroof from outside the car are some of the possibilities made possible with a Bluetooth extension of the system.

#### 6.4.1.3 Driver profiles

Some carmakers of today have implemented driving profiles in the key to the car. This means that different drivers have their own key with a built in memory chip. On this chip information about the car settings is stored. The settings of the mirrors, seats and radio channels among other things could be stored here. With a channel allocated via Bluetooth this information could be sent. This would improve the performance and the car could start the setting as soon as the driver approaches so that they are done by the time the driver has reached the car. Different profiles could also be stored in a device so that the settings could be saved for different occasions. It would then be possible to have one device for different vehicles or different places in the car. A passenger in the backseat could watch a certain TV channel and have one seat setting on a Monday and a different seat setting and a prepared video game on Tuesday. Those settings could be programmed on the software level.

#### 6.4.1.4 Data transmission

Today there are a lot of handheld devices with Bluetooth attachments. With a Bluetooth gateway in MOST it would be possible to exchange information like addresses (OBEX) and documents (files) between the onboard computer and the handheld device. For the user this would mean an easy way to store contacts and other files. Contacts then easily could be looked up and contacts from different user and different handheld devices would be gathered in the same place. The possibility to synchronize data is made possible without the use of any physical cradle.

In a future scenario the Bluetooth capacity of data transfer would have reached around 12 Mbps. That allows applications in need of a greater bandwidth and the possibilities expand. To download a map to the navigation system or games and music files to a media source would not take long. This could be done simultaneously as the gas refill takes place at a gas station.

Another improvement is the area of various payment systems. The need of road taxes increases and one possible solution for this could be a system connected via Bluetooth. This would be a smooth and secure system. The car only has to pass a certain point by the toll plaza and then a question with the fee appears on the screen in the car. With a few button pushes the procedure is over. Since Bluetooth uses frequency hops and time synchronization it is very difficult to bug. The security could also be added on the software level and by using encrypted messages in the Bluetooth transfer.

This idea could also be used in other payment systems. Imagine a fast-food restaurant with a drive-in line. The buyer in this case only has to pass a certain point in the line to have the menu presented by both sound and picture in the coupe. With a few button pushes the purchase would be done and no extra stop for ordering would be needed.

#### 6.4.1.5 Connection standard

There is a need for a standardization concerning the procedure of the connection. Is it possible to connect more then one Bluetooth product like a headset, a computer, a phone or a palm computer to the gateway at the same time? If it is, then there has to be some sort of priority of who has first access to the speakers, the microphone and so on.

Another problem to solve is how the Bluetooth device should connect to MOST. Who will be the master in the Bluetooth net, the gateway or the device? And who has the possibility to request a connection? In Bluetooth it is possible to make devices trustable which means that it could connect without needing to go through the authorization procedure each time. This is very comfortable but not particularly safe. Let us say a trusted device gets in the wrong hands. If there is sensitive information accessible from this device then the damage could be great. Adding a code to sensitive information, making the user enter a code to reach this information, easily solves this but then the ease of use will be affected in a negative way. It is a trade off that has to be done but a possibility is to let the user decide on his own if there should be a password protection in the system [5]. MOST *Working Group Bluetooth* handles some of those questions.

## 6.5 Implementation

Models in Matlab/Simulink have been made to simulate the interpolation procedure. To perform this simulation in reality a Bluetooth headset connected to a PC has been used. A demonstrator is set up where the audio is processed. Using the Ericsson HBH-20 Bluetooth headset and the PC card Digianswer Democard II made this possible.

The Bluetooth connection consisted of the headset HBH-20 and the Bluetooth PC card connected to a PC. Both devices are compatible with the Bluetooth headset profile 1.1. In this solution the PC acts the audio gateway and the software *Bluetooth Neighbourhood* is used to set up the connection between the devices. The PC is configured so that the Bluetooth PC card becomes the audio source.

The interpolation procedure is simulated in Matlab/Simulink. A model of the system has been made in the program and different interpolation methods and filter system has been simulated.  Matlab algorithms were created and inserted in the model to accomplish this. The interpolation model is shown in Figure 41. Some of the frequency spectrums after the interpolation are shown in Appendix A.

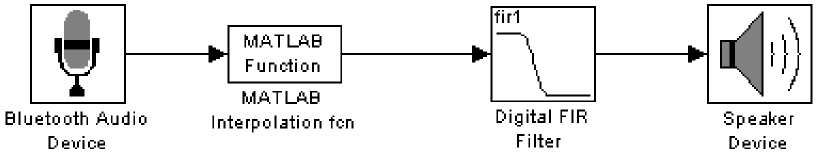


*Figure 41: Interpolation simulation*

The source sound is sampled with 8 kHz 16 bit mono PCM. The sound out of the filter procedure is 48 kHz 16 bit mono PCM. The parallel to serial conversion and the MOST network are not simulated.

### 6.5.1 Results

By empirical research the interpolation methods have been evaluated. The attributes considered in the evaluation have been computational workload, audible quality and memory usage. The solution will have to be a trade off between speed, quality and cost. An interpolation method with low computational workload is fast but the quality of the sound will be poor. With a filter this quality can be improved but the level of the improvement depends on the filter design. A good filter is expensive. The results of the interpolation methods without filters are presented in
Table 6.

| **Method** | **Workload** | **Memory** | **Quality** | **Filter** |
|---|---|---|---|---|
| *Zero filling* | Very low | Very Low | Poor | Yes |
| *Linear* | Low | Low | Fair | Yes |
| *FFT* | High | High | Good | No |
| *Sinc* | Very High | Very High | Excellent | No |

*Table 6: Interpolation comparison*

Using the methods zero filling and linear interpolation will require a filter to remove the noise that have been added in the interpolation procedure. The filter has to be better for zero filling since there is more noise added. The filter parameters set are the type of filter, the filter order and the cut frequency. The filter type used is a lowpass filter cutting off the frequencies that add noise to the signal. The cutoff frequency has been set so that the filter cuts off the frequencies above 4 kHz. Depending on the demands set for the gateway like quality and speed, the filter order may vary. In this simulation a filter order of around 20 is found acceptable according to the frequency spectrums and the audible result. This filter order does not cause a noticeable delay for this simulation.

The recommended solution for the interpolation is to use zero filling with a FIR filter. The results of the simulations for this method give a good audible result without a too heavy computational workload. This gives good conditions for a cheap implementation.

# 7 Conclusion

MOST and Bluetooth are two standards, which both have future in consumer products. By combining those techniques the expansion of integrated multimedia car applications are simplified. The gateway solution of this Master Thesis could be a part of a system interconnecting a cell-phone via Bluetooth to MOST. This would simplify the interconnection compared to the systems used today. A Bluetooth access point would be a universal interface for all Bluetooth enabled cell-phones.

## 7.1 Control data

The gateway has to interpret and convert the Bluetooth control data commands so that they are understandable by MOST. In the gateway design the Bluetooth Headset Profile is chosen as the audio source. The gateway has to translate the Bluetooth AT commands defined in the Headset Profile specification into routines that establish the connection on MOST. The commands are executed and the connections are established parallel on both sides to get a smooth and general solution where the two networks are unaware of each other.

## 7.2 Synchronous Audio

The sample rate on Bluetooth is 8 kHz and the sample rate on MOST is 44,1 or 48 kHz. The focus of this solution has been on 48 kHz since that sample rate is generally accepted. This simplifies the conversion since it is between rational numbers. The recommended implementation method is to interpolate the digital signal with zero filling. The interpolation has to be followed by a FIR filter that improves the audio by removing noise. To synchronize the audio to the MOST network a buffer has to be used to get the samples in phase. The MOST clock is used for the synchronization with the buffer. The audio is in parallel form after the interpolation and has to be converted to a serial format in this solution since the MOST Transceiver is set in serial mode. To design a more general gateway that can handle both asynchronous and synchronous data, it is recommended to use the MOST transceiver in a parallel combined mode.

In a complete communication system the total delay of audio can be no more than 250 ms [25]. This has to be taken under consideration when designing the gateway. This gateway is only a part of the whole system and the delay of other systems has to be considered and included in the total delay.

## 7.3 Future work

A future work of the results in this Master Thesis would be to implement a complete gateway. This could be done by using Ericsson Bluetooth Development Kit (EBDK) and the MOST General Node (MGN). Both cards have onboard FPGAs on which it is possible to implement applications. The MGN also has a microprocessor that can handle special application requirements such as interpreting Control commands from Bluetooth to MOST and vice versa. The FPGAs can be used for the interpolation application, the filter process and the serial to parallel conversion. There are also implemented clocks that can be used for synchronizing the data stream between the cards. When a success of this is made an investigation concerning cost and preferences of real parts for the gateway could take place.

# References

## Books

[1] Ahlin , Lars and Frank, Christer and Zander, Jens, *Mobil Radio kommunikation*, Studentlitteratur, 1995

[2] Bray, Jennifer and Sturman, Charles F , *Bluetooth – Connect Without Cables* , Prentice Hall PTR , 2001

[3] Comer, Douglas E, *Internetworking with TCP/IP*, Prentice Hall, 2000

[4] Glad, Torkel and Ljung, Lennart, *Reglerteknik Grundläggande teori*, Studentlitteratur, 1989

[5] Gollman, Dieter, *Computer Security*, Wiley, 1999

[6] Kamen, Edward W. and Heck, Bonnie S, *Fundamentals of signal and systems using MATLAB*, Prentice Hall, 1997

[7] Karlsson, Gunnar, *Modern telekommunikation*, Studentlitteatur, 1997

[8] Lancaster, Don, *Active filter Cookbook*, SAMS, 1988

[9] Miller, Brent A. and Bisdikian Chatschik , *Bluetooth Revealed* , Prentice Hall PTR , 2001

[10] Muller, Nathan J, *Bluetooth – Demystified*, McGraw – Hill , 2000

[11] Wanhammar, Lars and Johansson, Håkan, *Digital filters*, Linköping University, 1999

[12] Wanhammar, Lars and Sikström, Björn, *Elektriska filter*, Liber, 1986

## Specification documents

[13] Bluetooth SIG, *Specification of the Bluetooth System*, Core, version 1.1, 2001

[14] Bluetooth SIG, *Specification of the Bluetooth System, Profiles, version 1.1*, 2001

[15] MOST Cooperation , *OS 8104MOST Network Transceiver*, 2000

[16] MOST Cooperation, *MOST Specification Rev. 2.1*, 2001

[17] MOST Cooperation, *MOST Specification Framework Rev 1.1*, 1999

## Reports

[18] Niemitalo, Olli, *Yehar's digital sound processing tutorial for the braindead!*, University of Oulu, DSPSTUFF, 2001

[19] Rohthacher, Fritz Markus, *Sample Rate Conversion: Algorithms and VLSI Implementation*, Swiss Federal Institute of Technology, ETH No.10980,1995

[20] Trusch, Heiko, *A Bluetooth – CAN – Interface*, Chalmers University of Technology, EX043/1999, 1999

[21] Zobec, Robert, *Bluetooth for diagnostics and software download application*, Chalmers University of Technology, Ex033/2001, 2001

## URLs

[22] Derfler Frank J. Jr., *Crossed Signals: 802.11b, Bluetooth, and HomeRF*, *http://www.zdnet.com/products/stories/reviews/0,4161,2470132,00.html*, 2002-02-01

[23] MOST Cooperation, *MOST Cooperation*, *http://www.mostcooperation.com/*, 2002-03-04

[24] Bluetooth SIG, *Bluetooth Wireless Infosite*, *http://www.bluetooth.com/*, 2002-03-01

[25] Cisco, *Cisco White Paper – Understanding Delay in Packet Voice Networks*, *http://www.cisco.com/warp/public/788/voip/delay-details.pdf*, 2002-03-02

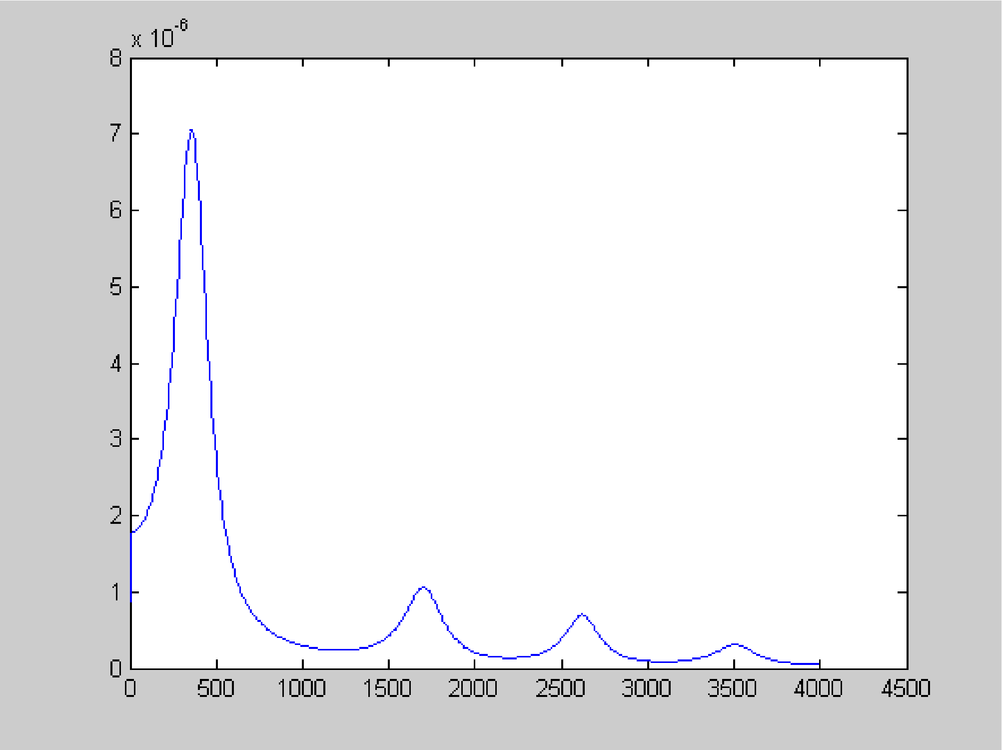
# Abbreviations

| | |
|---|---|
| **A/D** | *Analogue to Digital* |
| **ACL** | *Asynchronous Connectionless* |
| **AG** | *Audio Gateway* |
| **AM** | *Amplitude Modulation* |
| **ARQ** | *Automatic Repeat Request* |
| **CRC** | *Cyclic Redundancy Checksum* |
| **CVSD** | *Continuous Variable Slope Delta* |
| **D/A** | *Digital to Analogue* |
| **DSP** | *Digital Signal Processor* |
| **DV** | *Data Voice* |
| **DVD** | *Digital Versatile Disc* |
| **FEC** | *Forward Error Correction* |
| **FPGA** | *Field Programmable Gate Array* |
| **FSY** | *Frame Synchronization* |
| **GAP** | *General Access Profile* |
| **GOEP** | *Generic Object Exchange Profile* |
| **GSM** | *Global System for Mobile communication* |
| **HCI** | *Host Controller Interface* |
| **HEC** | *Header Error Correction* |
| **HS** | *Headset* |
| **HV** | *High quality Voice* |
| **IrDA** | *Infrared Data Association* |

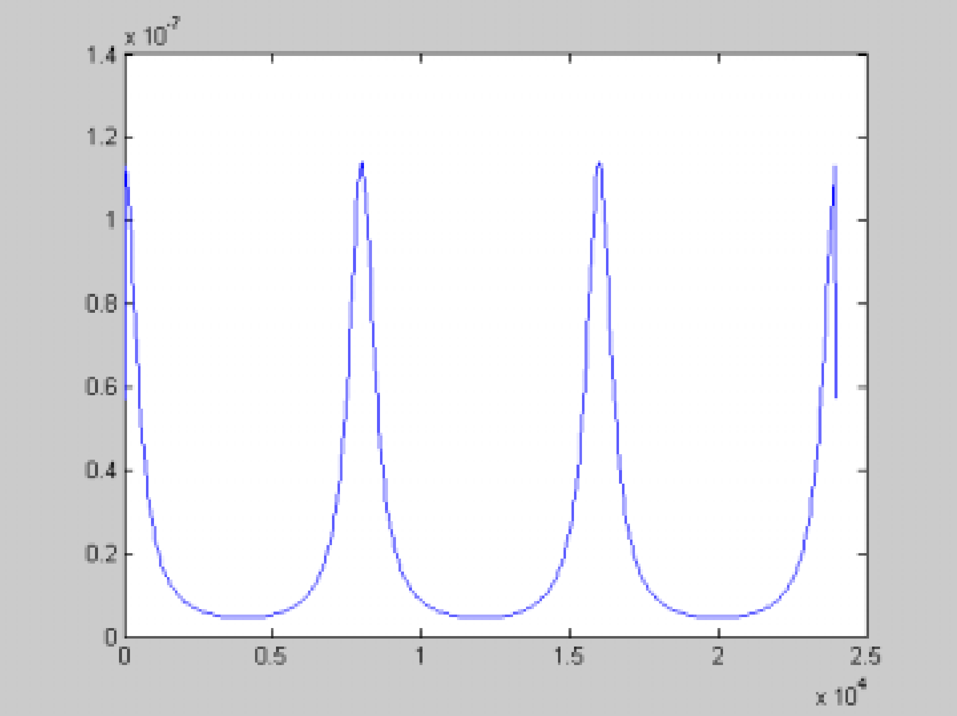
| | |
|---|---|
| **ITU-T** | *International Telecommunication Union – Telecommunication Standardization Sector* |
| **L2CAP** | *Logical Link Control and Adaptation Protocol* |
| **LAN** | *Local Area Network* |
| **MOST** | *Media Oriented System Transport* |
| **MPEG** | *Motion Pictures Expert Group* |
| **MTU** | *Maximum Transfer Unit* |
| **OBEX** | *OBject EXchange Protocol* |
| **OSI** | *Open System Interconnection* |
| **PCM** | *Pulse Code Modulation* |
| **PDA** | *Personal Data Assistant* |
| **PIN** | *Personal Identification Number* |
| **PSTN** | *Public Switched Telecommunication Network* |
| **QoS** | *Quality of Service* |
| **RFCOMM** | *Radio Frequency COMMunication* |
| **S/N** | *Signal to Noise ratio* |
| **SAR** | *Segmentation And Reassembly* |
| **SCK** | *Synchronization ClocK* |
| **SCO** | *Synchronous Connection Oriented* |
| **SDAP** | *Service Discovery Application Profile* |
| **SDP** | *Service Discovery Protocol* |
| **SIM** | *Subscriber Identity Module* |
| **SPP** | *Serial Port Profile* |
| **TCS-BIN** | Telephony Control Specification Binary |
| **WLAN** | *Wireless Local Area Network* |

# Appendix A: Frequency spectrums

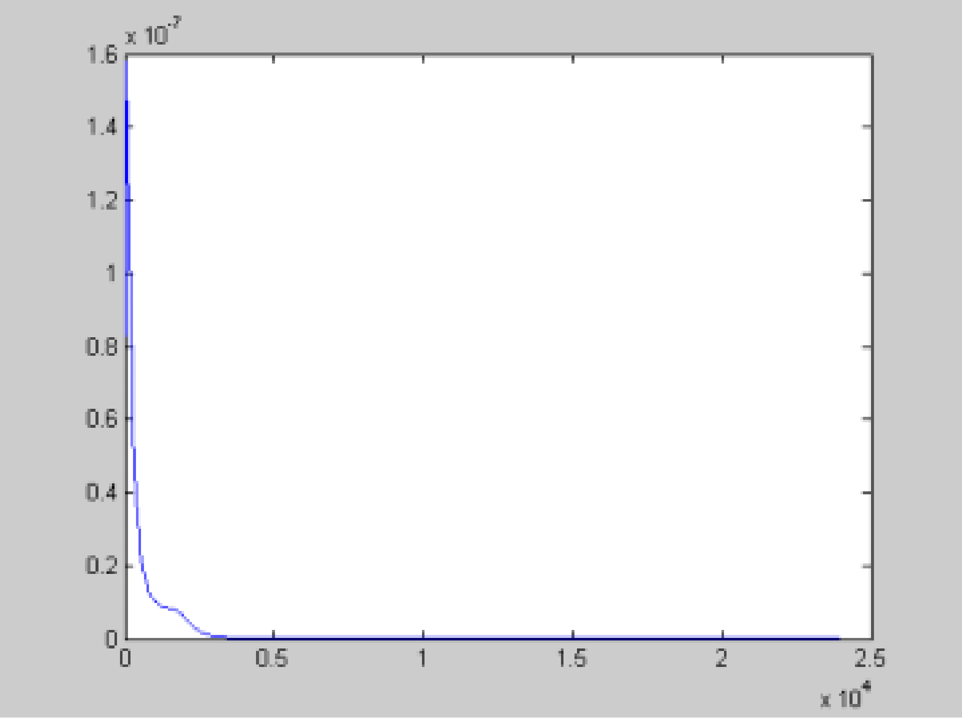
The original data used in the laboratory experiment is sampled with 8kHz 16 bits mono PCM and stored in a file. It lasts 2,7 seconds and contains ordinary telephone speech. The plots show the frequency spectrums of the original signal and the interpolated signals. The x-axis is the frequency in Hertz and the y-axis is the magnitude.



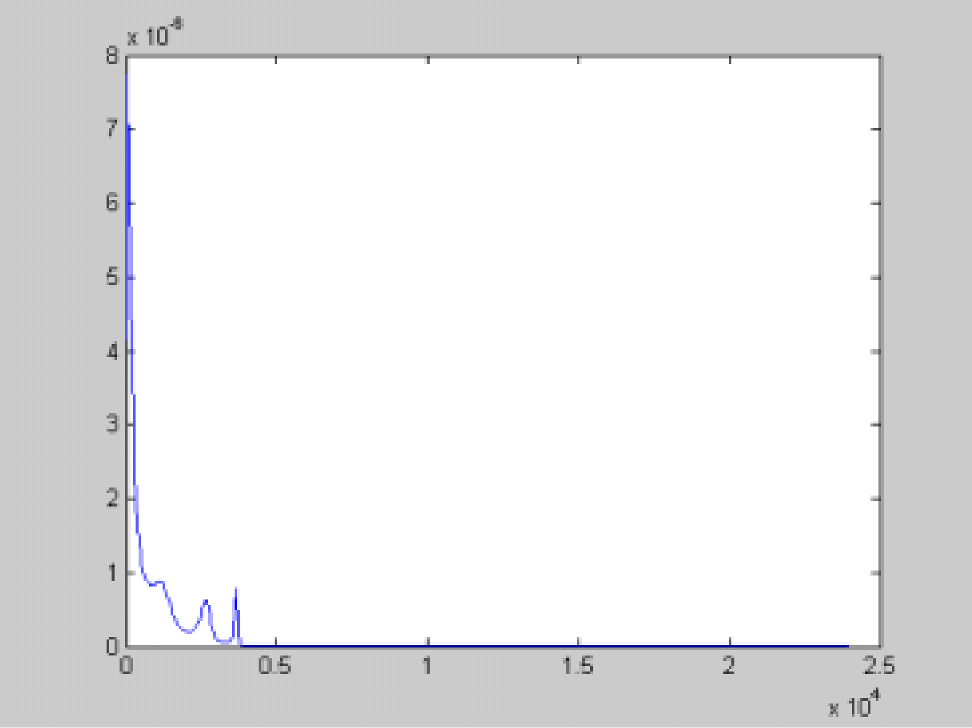
Frequency spectrum of the original signal (16 bit 8000 Hz PCM)



Frequency spectrum of the zero-fill interpolated signal (16 bit 48000 Hz PCM)



Frequency spectrum of the FIR filtered zero-fill interpolated signal (16 bit 48000 Hz PCM)



Frequency spectrum of the FFT interpolated signal (16 bit 48000 Hz PCM)