

# Design Sketch

Editor: Klas Gudmundsson

Version 1.0

## Status

Reviewed	CM	2016-09-16
Approved	MO	2016-09-23



## PROJECT IDENTITY

2016 HT

Linköping University, Dept. of Electrical Engineering (ISY)

### Group members

Name	Responsibility	Phone	Email
Linnea Faxén	Project Manager	0738296509	linfa440@student.liu.se
Klas Gudmundsson	Documentation Manager	0727140666	klagu863@student.liu.se
Eskil Jörgensen	Chief of Design	0762225372	eskjo325@student.liu.se
Stefan Lundström	Hardware Specialist	0703798079	stelu332@student.liu.se
Ema Becirovic	Test Manager	0735641573	emabe844@student.liu.se
Javier Preciado	Software Specialist	+34647926138	javpr698@student.liu.se

**Customer:** ISY, Linköpings universitet, 581 83 Linköping  
**Customer contact:** Mikael Olofsson, mikael.olofsson@liu.se  
**Course leader:** Danyo Danev, danyo.danev@liu.se  
**Supervisor:** Christopher Mollén, christopher.mollen@liu.se



---

## Contents

<b>Document History</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Aim and Goals . . . . .	1
1.2 Definitions . . . . .	2
<b>2 Overview of the System</b>	<b>2</b>
<b>3 Hardware</b>	<b>2</b>
3.1 Central Unit . . . . .	3
3.1.1 Computer . . . . .	3
3.1.2 Expansion Box . . . . .	3
3.1.3 Distribution Box . . . . .	4
3.1.4 Power Supply . . . . .	4
3.2 Signal Splitters . . . . .	4
3.3 L/M unit . . . . .	4
<b>4 Software</b>	<b>4</b>
4.1 API . . . . .	4
4.2 Application . . . . .	5



Document History

Version	Date	Changes	Sign	Reviewed
0.1	2016-09-15	First draft.	KG	KG,LF
0.2	2016-09-21	Fixed CM:s comments.	KG	KG,LF
1.0	2016-09-23	Updated version and history.	KG	KG,LF



# 1 Introduction

In the course TSKS05 a group of students are tasked with a project. This year, the group specified under Project Identity have been given the task to construct a Massive Audio Beamformer (MAB) that can be used as a demonstrator for Massive Multiple Input, Multiple Output (MIMO). This document will provide an outline of the MAB:s design and describe its functionality.

The MAB should be able to focus, or beamform audio to terminals. It will have an array that transmit signals and by beamforming it will direct these signals to specific terminals. Terminals will receive the signal intended for them along with noise but should not receive the other terminals signals. An example of how the MAB might look can be found in Figure 1. The group has decided to name this Massive Audio beamformer "BAM!" which is MAB backwards and then an exclamation mark at the end to make it sound more powerful.

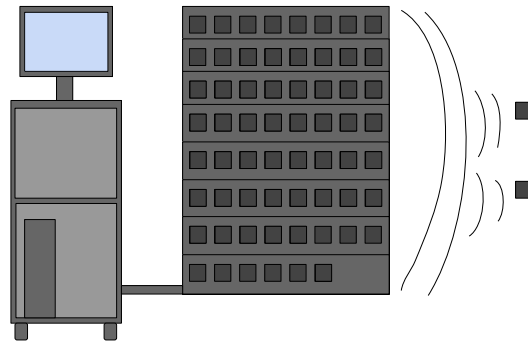


Figure 1: Example of finished product

A user will interact with the MAB via a user interface on a computer. Using this the user can control which antennas should be in the array or in the terminals and what data to transmit. The antennas in the MAB will be 64 combined loudspeaker and microphone units (L/M unit). Since the MAB will beamform audio, transmitting a signal corresponds to playing it in a loudspeaker. Receiving a signal corresponds to recording the sound, using a microphone.

To demonstrate the powers of massive MIMO in a pedagogic manner, two different sounds will be played at two different terminals in a room. The sounds are played from the array but through beamforming only one of the sounds is heard at one terminal and the other sound at the other terminal. This will demonstrate massive MIMO:s ability to direct signals to different users.

## 1.1 Aim and Goals

The main goal of this project is to build hardware and write software to create a MAB that can demonstrate massive MIMO at ISY at Linköping University.



## 1.2 Definitions

Throughout this document a number of abbreviations are used. All of them are defined in Table 1.

Table 1: Abbreviations used in the document

Abbreviation	Definition
A/D	Analog to digital
API	Application programming interface
D/A	Digital to analog
GUI	Graphical user interface
L/M unit	Loudspeaker/microphone unit
LED	Light-emitting diode
MAB	Massive Audio Beamformer
MIMO	Multiple input multiple output

## 2 Overview of the System

The system consists of two subsystems, hardware and software. Software is further divided into an application programming interface (API) and an Application. The user interacts with the Application which use methods from the API to control the hardware. The hardware contains all L/M units and the necessary electronics to control them. Figure 2 shows a basic overview of the system. There are three large areas of the project: assembling the hardware, making the API work and creating an Application. In the coming sections the hardware, API and Application will be described.

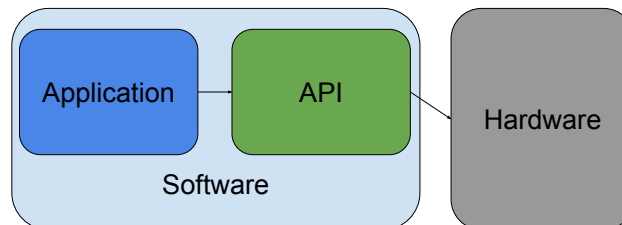


Figure 2: System overview

## 3 Hardware

The design of the whole system hardware has been created by the sponsor. What remains is to construct the hardware according to the design and control it via a computer. Figure 3 shows an overview of the hardware and its subsystems.

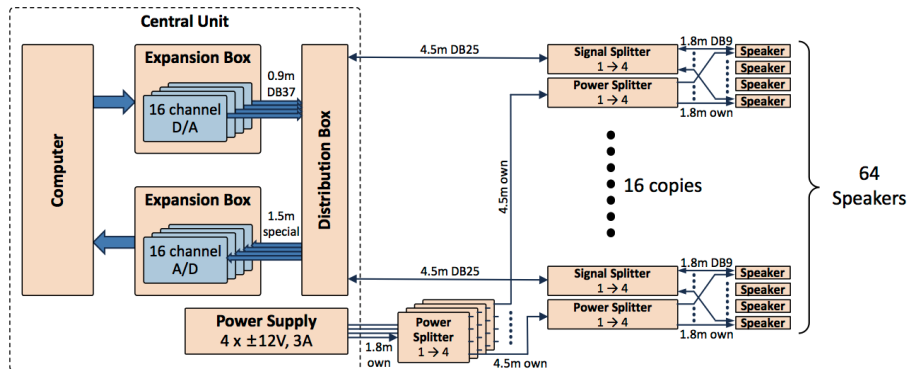


Figure 3: Hardware overview

As can be seen in Figure 3, the hardware will have a central unit that directs all signals from the computer to the L/M units (called Speaker in Figure 3) with the help of two expansion boxes and a distribution box. It will also direct the signals from the L/M units to the computer. From the central unit cables carrying the signals lead to the 64 speakers. There are 16 cables out from the distribution box which are split into 4 new channels each at the signal splitters. A power supply will also be located in the central unit, splitting the power with power splitters and distributing it to all L/M units. The rest of this chapter will describe each part of the hardware.

### 3.1 Central Unit

The central unit's purpose is to direct the signals to and from the cables leading to the specified L/M unit from and to the computer. It will consist of a computer, two expansion boxes, a distribution box and a power supply, as can be seen in Figure 3.

#### 3.1.1 Computer

The computer is a standard Windows PC tower. Each of the two expansion boxes will be connected via a DVI cable connected to a PCI-express card on the motherboard of the computer.

#### 3.1.2 Expansion Box

One expansion box will contain 4 16-channel A/D-converters and the other 4 16-channel D/A-converters. The D/A-converters are used to convert the digital signals from the computer to analog signals to be played at the L/M units. Similarly, the A/D-converters are used to convert the recorded analog signals at the L/M units into digital signals to be stored in the computer.

The A/D and D/A converter boards have a sample rate of 100 ksamples/s each. Splitting this over 16 channels gives a channel sample rate of 6250 samples/s. There are four 16-channel boards giving a total of 64 channels. The A/D converter is of type Contec AD12-16 (PCI) and the D/A converter is of type Contec DA12-16 (PCI)



### 3.1.3 Distribution Box

The distribution box contains electronics to distribute power and signals between the A/D and D/A converters and the L/M units. It also contains electronics (Arduino-units) to control the transmit/receive relays in the L/M units.

### 3.1.4 Power Supply

All L/M units are powered by the same power supply, which will be assembled by the sponsor. The power supply can be seen in Figure 3 as a part of the central unit. The power supply has four sockets, each consisting of three banana connectors: -12V, 0V and 12V. These potentials are then distributed to the L/M units by a series of power splitters.

## 3.2 Signal Splitters

To lessen the number of cables, boxes called signal splitters will be used between the distribution box and the L/M units. Each signal splitter distributes audio signals to and from four L/M units.

## 3.3 L/M unit

The L/M units are the combined microphone and speaker units used to receive and transmit sound. Each L/M unit has a printed circuit board that handles power, amplification of signals, and switching between transmit and receive mode.

# 4 Software

The software for the system will be written in MATLAB and used to control all speakers. It consists of two parts: API and Application. The API is the interface to the hardware and provides functions to use in the Application. The Application is the user interface and the signal processing needed to perform beamforming. The API is intended to be easy to use so that different applications can use the same API but perform different signal processing. This will make it easy for later projects to implement different signal processing projects on the MAB.

## 4.1 API

The API should provide basic functions for the Application to interact with the hardware. Table 2 provides an example of functions.





Table 2: API Functions

Function	Description	Input	Output
set_array	Sets the specified L/M units to group array	L/M units used for array	-
set_terminal	Sets the specified L/M units to group terminal	L/M units used for terminal	-
transmit_array	Transmits the specified signal on configured array	signal to transmit	received signals
transmit_terminal	Transmits the specified signal on configured terminal	signal to transmit	received signals

The L/M units can belong to one of two groups: array or terminal. The group array is the large number of L/M units that beamform signals. The group terminal contains the L/M units that should act as independent terminals, each receiving their own signal. To perform the described functions in Table 2 the API must interact with the A/D and D/A converters to receive and transmit the signals. The API must also control which group the L/M units belong to and whether they should transmit or receive. To receive and transmit data to the A/D and D/A converters MATLAB's Data Acquisition Toolbox will be used. Controlling which L/M unit belong to which group and its mode (transmit or receive) is handled by the distribution box using control signals from the A/D and D/A converters.

## 4.2 Application

The Application is the user interface and the needed signal processing to perform directive sound transmission. The aim of the Application is to play several different sounds to several different terminals. One way to do this would be using knowledge of the array's geometry. However, to demonstrate massive MIMO, the Application will estimate the channel using a pilot and then use this estimate to beamform the signal.

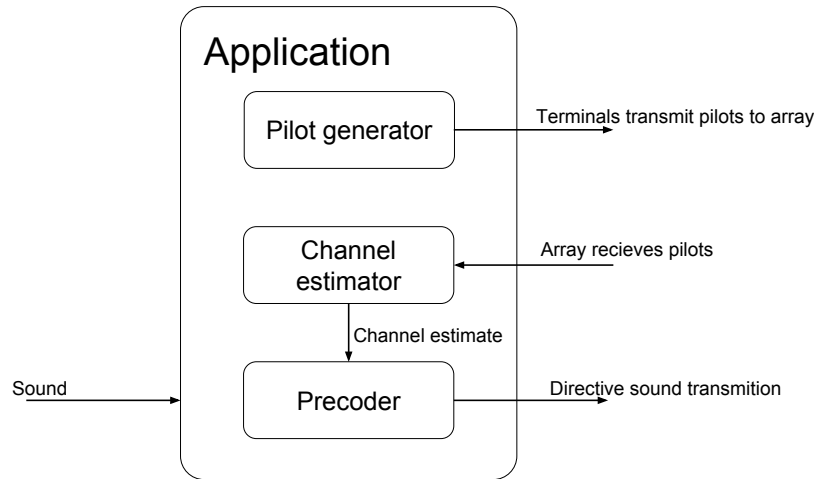


Figure 4: An overview of the Application

For the Application to be able to work, the channel has to be estimated. This is done by estimating the channel impulse response between each L/M unit in the base station array and each receiving terminal L/M unit. The estimation is performed on pilot-signals at the start of the transmission. The pilot is sent from the terminal to the array (uplink) and the received signal at the array is used to estimate the channel response for the uplink based on knowledge of the pilot-signals structure. Thereafter, we assume reciprocity, that the downlink channel will have the same channel response as the uplink. This can be done since since the only difference in the channel between uplink and downlink is that time has passed. Our array and users are stationary, not moving over time. Therefore, our channel is time-invariant.

After the channel is estimated, the sounds or songs we want to play are precoded so that the correct sound will be amplified at the correct place and canceled out at the other places. An overview of the Application is depicted in Figure 4, where it is possible to see the different parts of the Application.

The final user will be able to use the Application through a GUI. In the GUI, the user can choose what song to play at which user. The number of L/M units to use in the array can also be chosen.