MHS

MOBILE HARDWARE SAMPLER

PROBLEM ANALYSIS

Fast and **reliable audio sampling** and playback is mandatory for most music production workflows!

1. How is the audio processed to achieve this?

2. How to structure a user friendly, intuitive Interface?

PROJECT GOAL

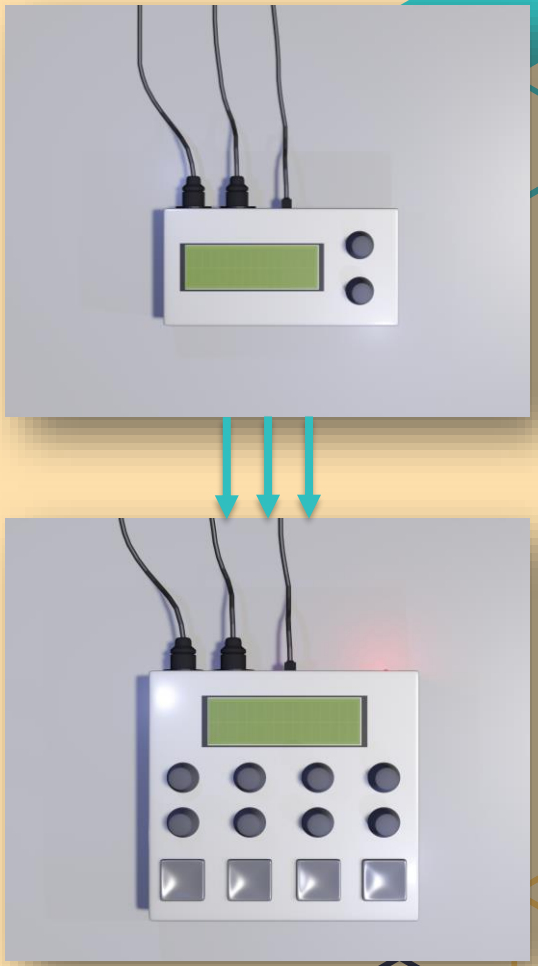
- Building a funktional hardware sampler

Minimum

- Read audio from a SD Card and load it into the RAM
- Provide connectivity to other devices over MIDI and MIDI over USB
- Design a simple but effective User interface
- 2-8 tone polyphony

Optional

- Digital effects (algorythmic reverb, filters, delay ...)
- A line Input to directly sample on the device
- Recording of a MIDI loop
- Velocity sensitive pads



THE STAGES

01

STAGE 1
MINIMAL FEATURES

02

STAGE 2
OPTIONAL FEATURES

03

STAGE 3
BUILDING A CASE

TECHNICAL ENVIRONMENT

HARDWARE

TEENSY 4.0

TEENSY AUDIO SHIELD

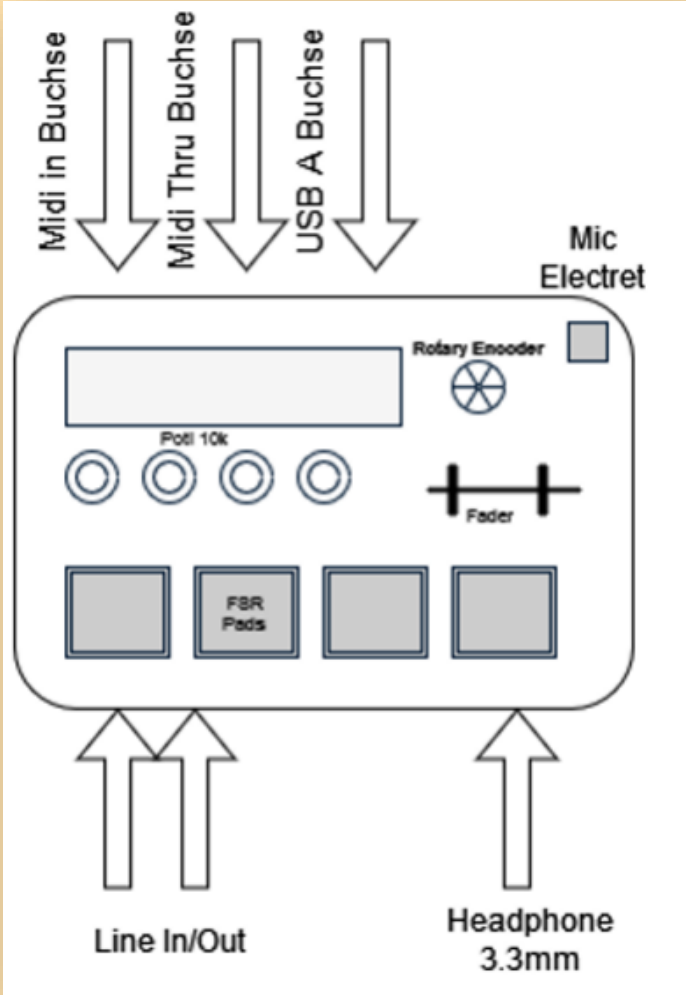
SOFTWARE

C++

TEENSY AUDIO LIBRARY

DESIGN IDEA

- **Rotary encoder** to select between samples and to navigate through the menu
- **Screen** to see the Name of the current sample pack and parameter changes.
- **Potentiometers** to change effect values
- **Velocity sensitive pads** to play samples live



PROJECT ORGANIZATION

Discord for communication

JetBrains Space for project structure and task management

Git to share and distribute files and code

To communicate on the go and wake up sleeping team members

TEAM STRUCTURE

TEAM MIDI

DAVID MERTENS

LENA WILBERTZ

TEAM DISPLAY

ALEXANDER KOSTENKO

DENNIS OBERST

TEAM AUDIO

DENNIS OBERST

LUCAS HAUPT

TEAMLEADER: DENNIS OBERST

TIMESCEDULE

Stage one finished	Calender Week 20
Stage two finished	Calender Week 28
Case finished	Calender Week 32
Final product tested	Calender Week 35