

Final Project

Michał Mieszczak

January 23, 2020

Contents

1	Introduction	3
2	Goals and tasks of the project	4
3	Topic analysis	5
3.1	Data acquisition	5
3.2	Processing audio	5
3.3	Effects	5
4	External specification	6
4.1	Block diagram	6
4.2	Description	6
4.3	Choice of components	7
4.4	Schematic	7
4.5	PCB	7
4.6	Assembly	8
5	Internal specification	9
5.1	Programming language	9
5.2	IDE	9
5.3	Text editor	9
5.4	Frameworks and tools	9
5.5	Software block diagram	10
5.6	Description of functions and objects	10
5.7	User manual	10
6	Testing	11
7	Conclusions	12
8	Bibliography	13
9	Attachments	14

1 Introduction

2 Goals and tasks of the project

The main goal of the project is to develop a platform based around modern, high-performance microcontroller. The device should process stereo audio signal to achieve a selection of different effects. To achieve decent audio quality at least 48kHz sampling frequency should be chosen.

Following tasks are required to finish the project:

1. Choice of components
 - microcontroller
 - operational amplifiers
 - additional devices
1. List of features
 - list of audio effects
 - user interface
1. Creating the physical layer of device
 - block diagram
 - schematic
 - PCB
 - assembly
1. Developing software
 - choosing programming language
 - choosing IDE, text editor, toolchain and frameworks
 - writing and testing the program (user interface, effects)

3 Topic analysis

3.1 Data acquisition

3.2 Processing audio

3.3 Effects

4 External specification

4.1 Block diagram

4.2 Description

Device was designed in a form of a “shield” for NUCLEO development board. The NUCLEO board provides a ST-Link programmer/debugger and voltage regulators required to power the board from 5V USB. The shield is populated with analog and digital parts as well as two audio jacks for analog input and output. Separate power lines and ground planes are used for digital and analog supply.

Device relays on both onboard digital to analog and analog to digital converters present in the microcontroller, so external preamplifiers are the only thing needed on the audio path. Input preamplifier consist of two stages. First stage acts as a buffer and provides about 14. 5dB boost. Boost is added to improve performance of converters. At this point DC offset is added, because unipolar voltage supply was used. The second stage is a basic phase inverter. Outputs of both stages are fed into differential inputs of ADC and are sampled by microcontroller. Output preamplifier consist of single inverting operational amplifier. It provides -20dB attenuation necessary to achieve line level on output. After output stage there is a 1. 5Hz high-pass filter, whose task is to eliminate DC offset. Both input and output preamplifiers are stereo and introduce 20kHz low-pass filter to prevent aliasing and noise. All 3 operational amplifiers are decoupled using a 100nF capacitor.

TFT LCD is used to provide necessary information to the user. It is connected to the board using gold pin headers and sockets. Display communicates with microcontroller through full duplex SPI connection.

To allow user to interact with the device, a rotary encoder was introduced. It is connected to microcontroller and is handled by internal timer. Push button of the encoder and additional user button are connected to GPIOs (General Purpose Input/Output) of the microcontroller.

Additionally the board is equipped with 32MB SDRAM chip (only 16MB is accessible). It is not used by the program of this project, but allows for future improvements. SDRAM can be handled by FMC (Flexible Memory Controller) present in the microcontroller.

4.3 Choice of components

- Microcontroller The microcontroller should meet certain parameters to be chosen. One of the most important parameters is the clock speed. Digital signal processing requires a lot of processor instructions per each audio sample, so high clock speed is preferred.

The second important criterion is amount of memory. The microcontroller should have at least 128kB of ROM to fit the written program. Size of RAM should be as high as possible, because delay effects require a lot of memory.

Presence of FPU (floating-point unit) is preferred due to its performance benefits.

To reduce the total cost of the device, proper A/D and D/A converters could be included in the microcontroller.

All of above criterion is met by STM32H743ZI microcontroller. It is based on Arm Cortex-M7 32-bit RISC core. Most important parameters of this microcontroller are listed below:

- 480MHz maximum clock frequency
- 1MB of RAM and 2MB of Flash memory
- single and double precision FPU and set of DPS instructions
- 16-bit ADC and 12-bit DAC
- 140 I/Os and selection of peripherals such as hardware SPI, I2C, I2S, UART, SPDIF, USB OTG

The STM32H743ZI comes one the NUCLEO-H743ZI evaluation board. It includes ST-LINK debugger/programmer, which simplified development of the device.

- Operational amplifiers
- Additional devices

4.4 Schematic

Schematic was created using KiCad EDA, which is both cross-platform and open source software. It provides rich library of components and very intuitive user interface.

4.5 PCB

PCB layout was prepared using KiCad EDA as well. All elements except for gold pin headers, rotary encoder and audio jacks are SMD (surface mounted).

To reduce cost of the project, the PCB was designed as 2 layer board only. This decision resulted in increased difficulty of routing process.

4.6 Assembly

The PCB was assembled by hand using 50W soldering iron and hot air rework station.

5 Internal specification

5.1 Programming language

C language is the most popular choice for STM32 development. It is widely supported by ST and other companies providing frameworks and libraries.

Since the object oriented approach was preferred in this project, C++ language was chosen. It allows to benefit from libraries written in C and gives advantages of using classes with inheritance and polymorphism.

5.2 IDE

The popular environments for STM32 development are Keil uVision, IAR and SW4STM32. ST also provides its own STM32CubeIDE, which is based on Eclipse.

But this time PlatformIO IDE was chosen. It is open source IDE and comes as a plugin for Visual Studio Code and Atom text editors. It provides all necessary libraries, frameworks, debugger server and GNU embedded toolchain for Arm.

5.3 Text editor

Visual Studio Code is a free code editor developed by Microsoft. It was chosen because it is supported by PlatformIO IDE. VSCode provides linting and word completion extensions, which simplify the process of software development.

5.4 Frameworks and tools

List of frameworks and additional tools used for software development:

- STM32CubeMX - a graphical tool that allows an easy STM32 configuration. It generates configuration code for the microcontroller and desired peripherals in C language.
- HAL drivers - (Hardware Abstraction Layer) driver is a library provided by ST. it allows an easy use of microcontroller and code compatibility between different STM32 devices.
- CMSIS - CMSIS (Cortex Microcontroller Software Interface Standard) is a hardware abstraction layer provided by Arm. It is compatible with all Arm Cortex microprocessors and microcontrollers. It includes API for Cortex-M and Cortex-A core and peripherals as well as additional tools (DSP library, neural network kernels, real-time operating systems).

- 5.5 Software block diagram
- 5.6 Description of functions and objects
- 5.7 User manual

6 Testing

7 Conclusions

8 Bibliography

1. Udo Zölzer, DAFX: Digital Audio Effects, 2002 John Wiley & Sons Ltd, ISBN: 0-470-84604-6
2. Udo Zölzer, Digital Audio Signal Processing, 2008 John Wiley & Sons Ltd, ISBN: 978-0-470-99785-7

9 Attachments