SILESIAN UNIVERSITY OF TECHNOLOGY

FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND COMPUTER SCIENCE

Engineer thesis

Realization of digital audio effects on high-performance MCU platform.

Author: Michał Mieszczak

Supervisor: Dr Inż. Jerzy Fiołka

Gliwice, February 2020

Contents

Chapter 1

Introduction

The rapid development of technology lead to the moment, when anybody can afford a development board and learn how to program microcontrollers without even leaving house. Hobbysts all around the world create products in wide range of applications, that can easily compete with their professional counterparts in both price and quality.

One example of such a field is market of sound related equipment. More and more small companies are growing out of nowhere, providing butique studio recording and processing equipment with great customer service and support.

Chapter 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 while delivering decent sound quality. Following tasks are required to finish the project:

1. Choice of components
    - microcontroller
    - operational amplifiers
    - additional devices
2. List of features
    - list of audio effects
    - user interface
3. Creating the physical layer of device
    - block diagram
    - schematic
    - PCB
    - assembly

4. Developing software
    - choosing programming language
    - choosing IDE, text editor, toolchain and frameworks
    - writing and testing the program (user interface, effects)

Chapter 3

Topic analysis

This chapter is devoted to main concerns related to the project, that need to be considered at the beginning. These will determine all important features and parameters of the device.

3.1 Data acquisition

The nature of audio signal is analog. This means, that it should be conditioned in a proper way before it can be processed by digital circuitry. It is done by ADC and coresponding preamplifier.

Signal must be reduced from continuous-time to discrete-time. This process is called sampling and requires reading value of the signal in equal intervals of time. Range of audible frequencies is considered to be 20Hz to 20kHz and we need the whole range to acheive reasonable quality. According to the concept of Nyquist frequency, the sampling frequency should be at least twice as high as maximum frequency present in the signal. In case of audio signal, it must be at leasu 40kHz. There are some typical sampling frequencies preffered for audio signals such as: 44,1kHz, 48kHz, 96kHz, 192kHz and it would be a good idea to use one of them.

Furthermore ADC quantizes the value of the signal. This means, that signal can take one of finite number of values. The count of possible values is equal to 2^n, where n is the bit depth of the converter.

As stereo signal consists of two separate channels, there sould be two identical

sets of preaplifiers and converters.

3.2 Output generation

Processed digital signal must be converted back to analog. This is done using DAC and corresponding amplifier. DAC converts digital value into specific analog voltage, outputs it and holds the value until succesive sample is to be outputted.

3.3 Processing audio

Digital signals are usually processed in blocks. This approach allows to use FFT, which is necessary for some effects. Furthermore it can reduce time needed to process each sample. One drawback of this method is latency introduced. The whole block of samples needs to be collected before it can be processed. In case of blocks consisting of 512 samples, there is 1024 samples delay between collecting and outputting specific sample. In combination with 48kHz sampling frequency this results in 21.3 ms latency. Outputting, processing and collecting subsequent blocks off data must be performed simultaneously.

3.4 Effects

This section is dedicated to audio effects implemented in the devices as well as their features and parameters. It is important to fully understand nature of specific effect in order to implement it's entire functionality in a proper way.

3.4.1 Delay

Delay is the most basic time based effects. It creates echo of original signal, which is delayed by specified amount of time. This delayed signal is then mixed with original. Delayed signal can also be attenuated and then fed back into a delay block in order to achieve endless, fading reflections.

Because processed signal is quantized, delay time is described by number of samples. Digital implementation of this effect requires use of circular buffer, which

size is determined by delay time required. Each sample collected from input is stored in the buffer in order to be read later and mixed to the output.

3.4.2 Modulation

Modulation effects use LFO to modulate different parameters. Our attention will be focused on frequency modulation which is based on delay effect. Modulation effects usually use short delay times up to 30ms and LFO frequency up to several Hz. Different functions, such as sine, triangular or sawtooth can be used for LFO. There are a few distinct modulation effects:

- Chorus - uses delay time up to 20ms with LFO frequency up to 10Hz with medium depth.
- Flanger
- Vibrato
- Tremolo
- Phaser

3.4.3 Biquad filter

3.4.4 Drive

Chapter 4

External specification

4.1 Block diagram

4.2 Desctiption

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 with unity gain. 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 of the microcontroller with external pull-up resistors.

Additionally the board is equipped with 32MB SDRAM chip (of which only 16MB is accessible). It is not used by the program of this project, but allows for future improvements. SDRAM is handled by FMC 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 is preferred due to its performance benefits. Dedicated floating-point hardware greatly decreases time needed for floating-point calculations.

  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.

Chapter 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. This 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.

Another choice is PlatformIO IDE. It is free and open source IDE and comes as a plugin for Visual Studio Code and Atom text editors. It provides all libraries, frameworks, debugger server and compiler necessery for STM32 software development.

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:

- HAL drivers - Hardware Abstraction Layer driver is a library written in C language, provided by ST. It allows an easy use of microcontroller and code compatibility between different STM32 devices.
- STM32CubeMX - a graphical tool that allows an easy STM32 configuration. It generates configuration code for the microcontroller and desired peripherals in C language. STM32CubeMX uses HAL driver in generated initialization code.
- 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).
- OpenOCD - Open On-Chip Debugger provides programming and debugging support of boundary scan interface. It supports ST-Link, which was used for both programming and debugging of STM32 microcontroller.
- GNU Embedded Toolchain for Arm is a suite of tools for C, C++ and Assembly programming. It contains the GCC Compiler.

5.5 Software block diagram

5.6 Description of functions and objects

5.7 User manual

Chapter 6

Testing

Chapter 7

Conclusions

Bibliography

List of abbreviations and symbols

ADC - Analog to digital converter

API - Application programming interface

DAC - Digital to analog converter

DC - Direct current

DMA - Direct memory access

DSP - Digital signal processing

EDA - Electronic design automation

FFT - Fast fourier transform

FMC - Flexible momory controller

FPU - Floating-point uint

GCC - GNU Compiler Collection

GPIO - General purpose input/output

IDE - Integrated development environment

LCD - Liquid crystal display

LFO - Low frequency oscillator

PCB - Printed circuit board

RAM - Random access memory

ROM - Read only memory

SDRAM - Synchronous dynamic random access memory

SMD - Surface mounted device