Universidade de Aveiro
Ano 2011

Departamento de Electrónica, Telecomunicações e Informática

Marcin Józef Janiszewski

Unidade de Efeitos de Áudio

Audio Effects Unit

Universidade de Aveiro
Ano 2011
Departamento de Electrónica, Telecomunicações e Informática

Marcin Józef Janiszewski

Unidade de Efeitos de Áudio

Audio Effects Unit

Układ do Generowania Efektów Dźwiękowych

Dissertation submitted to the University of Aveiro as part of its Master's in Electronics and Telecommunications Engineering Degree. The work was carried out under the scientific supervision of Professor Telmo Reis Cunha of the Department of Electronics, Telecommunications and Informatics of the University of Aveiro.

Mr. Marcin Janiszewski is a registered student of Technical University of Lodz, Lodz, Poland and carried out his work at University of Aveiro under a joint Campus Europae program agreement. The work was followed by Professor Marcin Janicki at Technical University of Lodz as the local Campus Europae Coordinator.

o júri

presidente

Professor Doutor Dinis Gomes de Magalhães dos Santos
Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Professor Doutor Carlos Miguel Nogueira Gaspar Ribeiro
Professor Adjunto do Departamento de Engenharia Electrotécnica da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

Professor Marcin Janicki
Associate Professor of Department of Microlectronics and Computer Science of Faculty of Electrical, Electronic, Computer and Control Engineering,Technical University of Lodz, Lodz, Poland

Professor Doutor Telmo Reis Cunha
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

palavras-chave

processamento de sinais audio, microcontroladores, conversão analógico-digital e digital-analógico, desenvolvimento de circuitos impressos, conexão USB.

resumo

O objectivo principal da presente tese de mestrado centrou-se no desenho e construção de uma unidade de efeitos de áudio (Audio Effects Unit -AEU), cuja função consiste em processar sinais áudio em tempo real. O propósito central foi desenvolver uma unidade de processamento áudio genérica, cuja função de processamento, implementada no domínio digital, pode ser facilmente especificada pelo utilizador via uma aplicação de software implementada num computador.

A primeira etapa deste projecto consistiu na implementação completa do hardware que constitui o AEU. É importante acrescentar que esta concepção teve em conta a inclusão desse hardware numa caixa apropriada. Este método de projecto e implementação constituiu uma experiência muito interessante e útil.

A próxima etapa consistiu no desenvolvimento de algoritmos matemáticos a ser implementados no microcontrolador do AEU e que geram os efeitos sonoros desejados por processamento dos sinais áudio originais. Estes algoritmos foram inicialmente testados através do Matlab.

Para controlar os efeitos sonoros produzidos foi ainda criada uma aplicação de computador que permite a intervenção, de forma muito simples, do utilizador. A referida aplicação assegura a comunicação entre o microcontrolador do AEU e o computador através de uma ligação USB.

O dispositivo, na sua versão final, foi testado em laboratório e através do Matlab. Cada bloco do dispositivo, e o dispositivo completo, foi testado individualmente. Com base nessa avaliação foram desenhadas as respectivas características na frequência e analisada a qualidade do dispositivo de áudio.

Para além da experiência adquirida em concepção de hardware, este projecto permitiu-me alargar o meu conhecimento em programação de microcontroladores e na optimização de código, um requisito do processamento de sinal em tempo real. Também me deu a oportunidade de utilizar a ferramenta comercial MPLAB para programação de microcontroladores.

keywords audio signal processing, microcontrollers, analog-digital and digital-analog conversion, printed circuit boards development,USB connection.

abstract

The main aim of this master thesis was to design and build an Audio Effects Unit (AEU), whose function is to process, a particular audio signal in real time. The objective was to develop a general purpose audio processing unit where the processing function, implemented in the digital domain, can be easily specified by the user by means of a software application running on a computer.

The first stage of this project consisted on the full design and implementation of the hardware that constitutes the AEU. It is worth adding that such design also considered that the layout could be placed in an enclosure. Such way of designing was a great new experience.

The next stage was to prepare the mathematical algorithms to be implemented in the AEU microcontroller which create the sound effects by processing the original audio signal. These algorithms were first tested in MatLab.

To control the produced sound effects a computer program was created which allows the user intervention in a straightforward way. This program ensures communication between the AEU microcontroller and PC software using an USB connection.

The completed device was tested in laboratory and with Matlab. The individual blocks of the AEU, and the whole device, were tested. On the basis of these tests frequency characteristics were drawn and the quality of the audio device was analyzed.

Besides acquiring expertise in hardware design, this project has broadened my knowledge on microcontroller programming and code optimization, a requirement for real time signal processing. It also gave me the opportunity to use the commercial MPLAB programming environment.

słowa kluczowe

przetwarzanie sygnału dźwiękowego, mikrokontrolery, konwersja analogowo cyfrowa i cyfrowo analogowa, budowa obwodów drukowanych, połączenie USB.

streszczenie

Głównym celem tej pracy magisterskiej było zaprojektowanie i zbudowanie układu do generowania efektów dźwiękowych (Audio Effects Unit - AEU) służącego do przetwarzania sygnału dźwiękowego w czasie rzeczywistym. Zadaniem autora było skonstruowanie ogólnego zastosowania układu przetwarzającego sygnał dźwiękowy, w którym funkcja przetwarzania, zaimplementowana w sposób cyfrowy, może być łatwo określona przez użytkownika poprzez zastosowanie odpowiedniego oprogramowania komputerowego.

Pierwszy etap projektu polegał na szczegółowym zaprojektowaniu i zbudowaniu warstwy sprzętowej tworzącej AEU. W projekcie przewidziano tez możliwość umieszczenia układu w obudowie, co było dla autora nowym doświadczeniem projektowym.

Kolejnym etapem było opracowanie algorytmów matematycznych, zaimplementowanych w mikrokontrolerze AEU, które tworzą efekty dźwiękowe poprzez przetwarzanie oryginalnego sygnału dźwiękowego. Te algorytmy zostały najpierw przetestowane w programie MatLab.

Do kontrolowania wytworzonych efektów dźwiękowych, został napisany program komputerowy, który pozwala na prostą interakcje z użytkownikiem. Ten program zapewnia komunikację między mikrokontrolerem AEU i oprogramowaniem komputerowym poprzez złącze USB.

Gotowe urządzenie zostało zbadane w laboratorium oraz za pomocą programu Matlab. Poszczególne bloki AEU jak i całe urządzenie zostały przetestowane, co pozwoliło na wykreślenie charakterystyk częstotliwościowych i umożliwiło analizę jakości wykonanego urządzenia audio.

Oprócz zdobywania doświadczenia w projektowaniu sprzętu, udział w projekcie poszerzył moją wiedzę o programowaniu mikrokontrolerów i optymalizacji kodu, potrzebną dla przetwarzania sygnału w czasie rzeczywistym. Ponadto miałem możliwość zapoznania się z komercyjnym środowiskiem programistycznym MPLAB.

Table of contents

List of Figures

List of Tables

Chapter I:
Introduction and Objectives

Signal processing for audio applications has been a topic of thorough research in the last decades, generating several solutions at different levels of the audio field. A search through the commercially available audio equipment reveals the existence of several devices whose purpose is to process, or transform, an input audio signal into another audio signal with added properties. Examples are amplifiers, filters, equalizers and audio effects devices. A particular class of audio devices is dedicated to transform the signals that are produced by musical instruments, giving them more flexibility in terms of the variety of sound characteristics they can produce. A typical example is that of the electric guitar which whose generalized use throughout most of the spectrum of musical genders was only made possible by the audio processing devices that process the electrical signal it generates.

Traditionally, these processing devices are based on analog implementations, but in recent years, with the development of digital technology, digital processing units are proving to be very competitive with the former analog solutions. There seems to be a tendency for every analog circuit to be realized through a digital system. This can be justified by the extra flexibility that digital processing has over analog solutions, since the very same hardware can be used to process the audio signals in many different ways, just by changing the digital algorithm that performs such processing.

With this in mind, the work here presented consists on the development of a digital-based electronic device whose objective is to process the electric signal generated by a musical instrument (in fact, it may be generated by any device that produces voltage signals in the audio frequency range), and to send out the transformed voltage signal so that it can be amplified by an audio amplifier, or captured by an audio sampling system. Even though this device, denominated Audio Effects Unit (or simply, AEU), was more focused on processing

algorithms commonly used to create sound effects for electric guitars, it can be used with any other audio signal source that makes available the source signals as a voltage wave.

Even though there can be found in the market several audio devices which can generate a specified set of sound effects, the idea behind this work was to develop a sound processing device whose processing algorithm could be designed by the user, in a computer, and uploaded into the device. This way, the number of sound effects that can be produced by the developed device is limited only by the imaginative ability of the user.

Moreover, this work had also as objectives to allow the acquisition of knowledge and practice in different, but important, fields of electrical engineering, such as: analog circuit design, digital circuit design, PCB design and elaboration, device design, digital signal processing, USB and I2C communication, among others.

The basic diagram for the AEU implementation is depicted in Figure 1. As shown, the signal generated by the musical instrument is processed by the AEU, which sends the transformed signal directly to the audio amplifier. By means of an USB2.0 connection, the user can specify and upload into de AEU the algorithm that performs the AEU signal processing. Preceding the design stage of the AEU, a set of basic requirements was specified for the AEU, which include:

- Guaranteeing a good quality of the output audio signal;
- Good linearity characteristics (i.e., low undesirable distortion);
- Low noise addition.

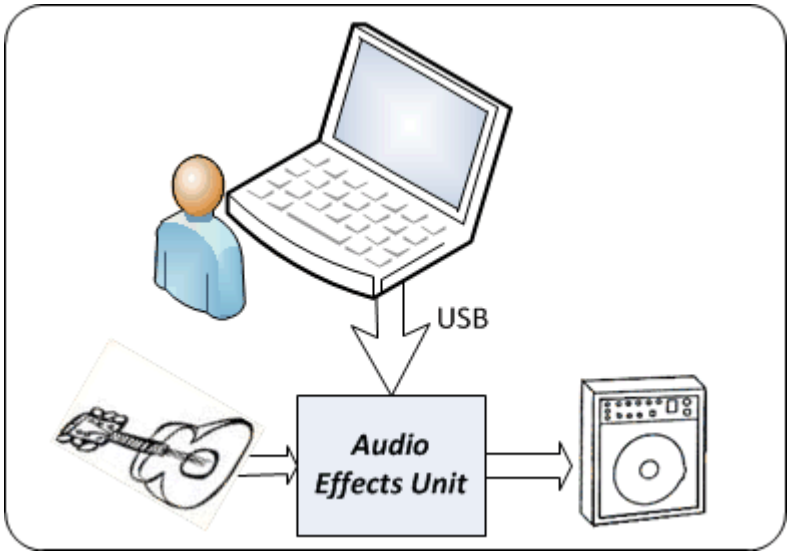


Figure 1 - General AEU implementation diagram.

To design and upload the AEU processing algorithms, a user-friendly MS-Windows based application is to be developed, providing the user with different examples on how to perform this task. Among other objectives, this interface (and the ability that the AEU has to be programmed in such an easy manner) allows the user to test the impact that variation of the algorithm parameters has on the produced sound effects.

To provide the user with more flexible sounds, the commercially available sound processing devices usually allow different effects to be simultaneously active (on moderate cost devices, the number of simultaneous effect is usually three). Thus, it was established as an additional requirement that the developed AEU allows the simultaneous use of four independent effects, being set active (or deselected) by the user by means of buttons placed on top of the AEU device. If active, these effects are executed, in series, by the digital processing unit of the AEU.

Besides the hardware design and respective implementation, this project involves the challenging task of on-the-fly processing the audio samples in a very efficient way so that information is not lost. To maintain a high level of audio signal quality it is required to have simultaneously a high sampling rate and a high resolution of the analog to digital conversion. Thus, the processing algorithms which transform the original audio signals into those enhanced with sound effects must be carefully controlled in terms of computational efficiency since, with common microcontrollers, the time available to process will impose a significant restriction.

Chapter II:
State-of-the-art Survey

II.1. Properties of audio signals

Sound (acoustic wave) is a mechanical wave that is an oscillation of pressure. It can be transmitted through a solid, liquid or gas medium. This third propagation medium is most common for our ears. Sound, as heard by humans, is composed of frequencies on the approximate range of 20 Hz to 20 kHz [Kahr 4], with a level sufficiently strong to be heard, as we can see in Figure 2. Even though most authors define the lower limit of the audio frequency as 20 Hz, there are other scientific manuscripts that state that it is 16 Hz [Body 285].

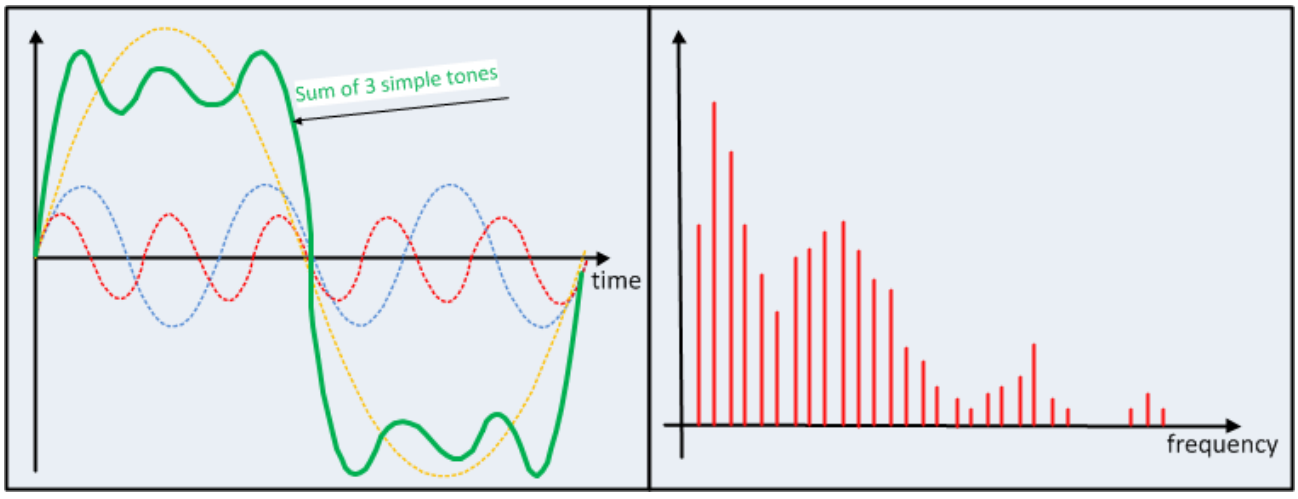


Figure 2 Audio signal in: a) time domain b) frequency domain.

Sound waves are classified in three main groups: infra sounds, audible sounds, and ultra sounds. In Figure 3 is shown the distribution of such groups in the frequency scale.

As we can see, the frequencies of audible sound lie within the 20 Hz to 20 kHz range. This range is, naturally, just an indication, because not every human ear can hear through all of this range. Some people can only hear part of this range. Also, when we grow old our capability to hear becomes diminished.

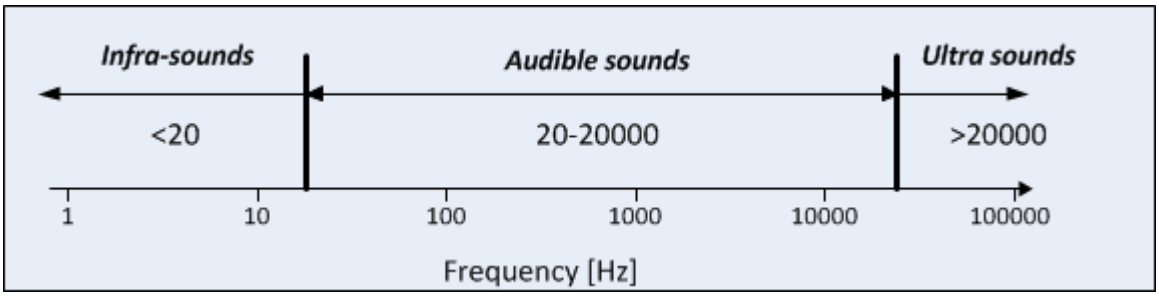


Figure 3 Main classification of sound waves according to their frequency.

Vibrations from 20 to 300 Hz are known as low frequencies and from 3 kHz to 20 kHz as high frequencies. Human ear is most sensitive for frequencies in range of 300– 3000Hz.

Even though the audio effects device developed in this work can be intended for general purpose audio, thus processing signals generated from distinct equipment, its common application is to modify the properties of sound waves produced by musical instruments. One of the most common uses is to process the electric analog signal generated by the pick-ups of a guitar instrument. So, the characteristics of these signals need also to be analyzed in the scope of the development of the audio effects device.

For the case of a typical guitar instrument (and also of others musical instruments), the range of generated frequencies is narrower than the full audio range. In Table 1 [Proj] are shown the musical tones and theirs corresponding frequencies (the frequencies for the six unconstrained cords of a standard guitar are highlighted in red). The highest string has generates a 330 Hz tone if freely played, but it can go as high as its fourth harmonics, yielding a maximum frequency of 1320 Hz.

Frequencies [Hz]													
Name of octave	Typical name for tones												Typical designation for voices
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	
1st	16.4	17.3	18.4	19.4	20.6	21.8	23.1	24.5	26.0	27.5	29.1	30.9	C_2
2nd	32.7	34.6	36.7	38.9	41.2	43.7	46.2	49	51.0	55.0	58.3	61.7	C_1
3th	65.4	69.3	73.4	77.8	82.4	87.3	92.5	98.0	103.8	110.0	116.5	123.5	C
4th	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9	c
5th	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9	c^1
6th	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8	c^2
7th	1046.5	1108.7	1174.7	1244.5	1318.5	1396.9	1480.0	1568.0	1661.2	1760.0	1864.7	1975.5	c^3
8th	2093.0	2217.5	2349.3	2489.0	2637.0	2793.8	2960.0	3136.0	3322.4	3520.0	3729.3	3951.1	c^4
9th	4186.0	4434.9	4698.6	4978.0	5274.0	5587.7	5919.9	6271.9	6644.9	7040.0	7458.6	7902.1	c^5
10th	8372.0	8869.8	9397.3	9956.1	10548.1	11175.3	11839.8	12543.9	13289.8	14080.0	14917.2	15804.3	c^6

Table 1 Frequency of the musical single tones.

The Nyquist-Shannon theorem defines the sampling frequency limit in a sampling process. It is known that if we want to precisely reconstruct a continuous signal from discrete one, without losing any information from the continuous signal that initially originated the sampled signal, then the considered sampling frequency must be higher than at least two times the highest frequency within the original signal. So, for a general audio signal, the sampling frequency must be higher than 40 kHz, otherwise aliasing-effects will occurs. However, if the limit is imposed by the musical instrument that generates the audio signal to be processed, then the sampling frequency can be significantly reduced (for instance, a frequency of 2640 Hz would be the lower limit for sampling the signals generated by a guitar instrument).

In Figure 4 is shown the usual sampling frequencies (given in kHz), and also the respective bit resolution, used in audio applications.

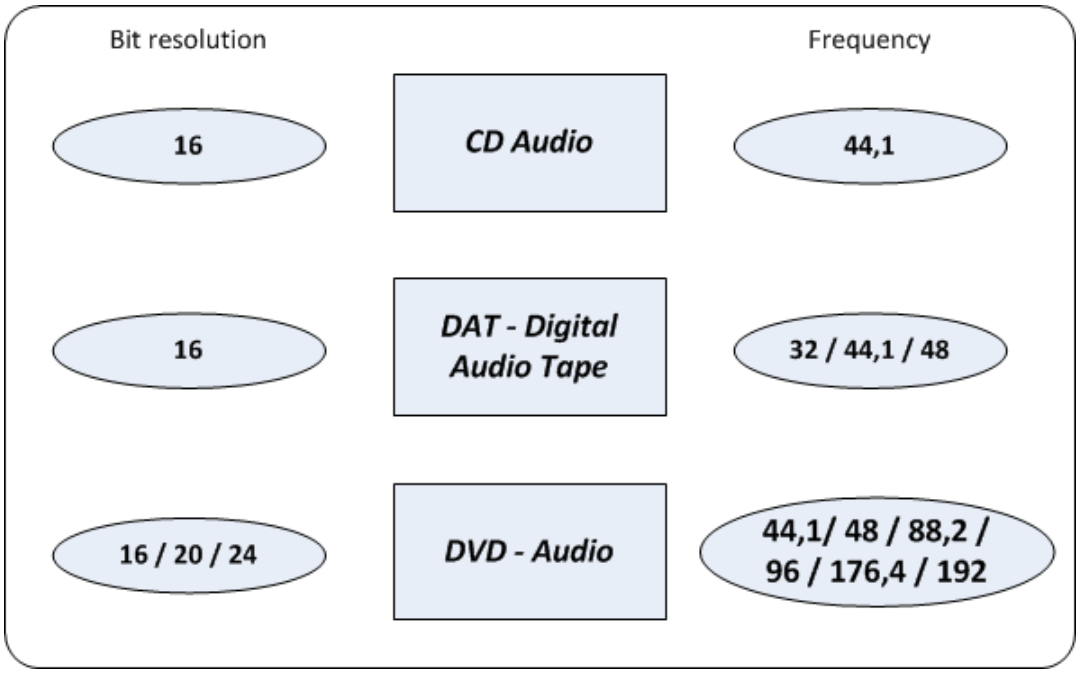


Figure 4 Bit resolution and frequencies (in kHz) in Audio common audio devices.

II.2. Typical architecture of digital processing units for audio signals

The typical topology used in real-time signal processing units for audio applications is depicted in Figure 5. The main block, in the middle, consists on a digital processing unit (such as a micro-processor or micro-controller) which receives the digital samples of the input audio signal, generated by an analog-to-digital converter (ADC), and processes them according to specific algorithms. These algorithms change the sampled input signal according to the device objectives – for instance, this can be a simple filter, an equalizer, or, as is the case in the work here presented, it can add effects to the original audio signal such as echo, reverb, distortion, or others commonly encountered in commercial devices.

The modified samples are then sent by the processing unit to a digital-to-analog converter (DAC). Since the conversion from analog domain to the digital domain introduces quantization error, a low-pass filter is usually used to interface the DAC output with the device output. Moreover, DC level adjustment and signal amplitude amplification is usually also necessary at this point. These filtering and signal adjustment operations are included in the block generally denominated as output signal conditioning block.

Also common in real-time signal processing devices is the input signal conditioning which, again, filters the input signal (usually to eliminate higher frequency noise) and adjusts the DC and amplitude levels to the specifications of the ADC input.

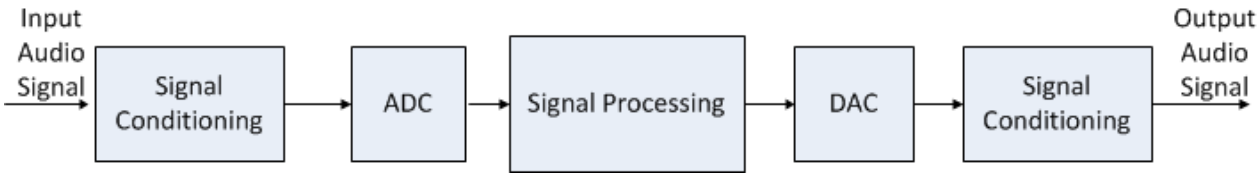


Figure 5 Typical architecture of a real-time audio signal processing device.

II.2.1. Digital Signal Processing Units

Digital signal processing units present several advantages over those based on analog signal processing. The most significant of these is that microprocessor systems are able to accomplish tasks inexpensively that would be difficult or even impossible using analog electronics. Digital systems are also insensitivity to environment changes. For example, analog circuits' behavior depends on its temperature, while digital system delivers the same response independently of the temperature (naturally, within the device's safe range of operation). Additionally, they are almost insensitive to component tolerances. They are also reprogrammable, which makes them useful not only for one concrete functionality but for a large number of applications (maintaining the same hardware). Another advantage of the use of microprocessors is the size reduction of devices, when compared with analog ones. In fact, as the microprocessor concentrates in a single (small) chip the input-output behavior of the signal processing system, it avoids the necessity of using a considerable number of discrete components which would be necessary to be considered in an analog implementation. However, there are also important disadvantages associated with microprocessors such as finite memory and limited processing time. This last issue is of high importance, being many times the bottleneck of many applications. In fact, as the microprocessor operates in a clocked way, processing every program instruction in a step-by-step mechanism, the number of instructions that are possible to be executed between two consecutive samples acquired by the external ADC is limited. Moreover, this limitation is further restricted as the sampling frequency increases, limiting the flexibility of the processing algorithm.

Digital signal processing unit can be generally characterized into three groups:

- Digital Signal Processors
- Microcontrollers
- Digital Signal Controllers

All of them are implemented according to the Harvard architecture (shown in Figure 6), which makes use of separate program and data memories.

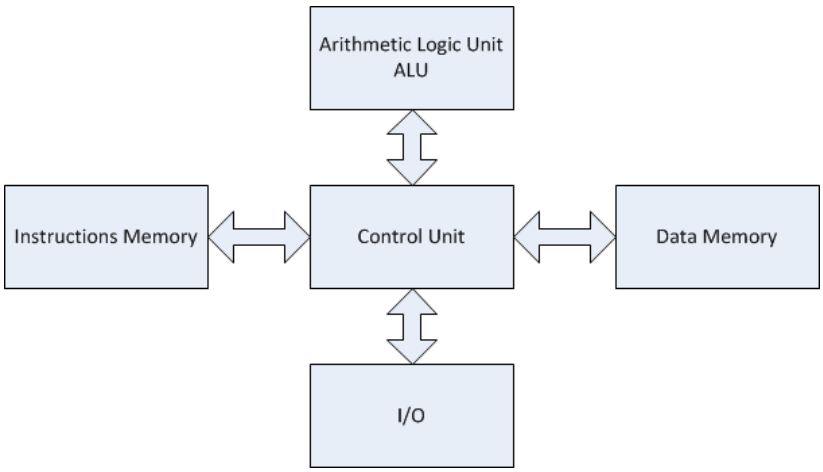


Figure 6 Harvard architecture.

The first group is Digital Signal Processors (DSP), sometimes called also Programmable Digital Signal Processors (PDSP). These are microprocessors that are specialized to achieve high performance in digital signal processing-intensive applications. DSPs are sometimes found in applications requiring high frequency operation like: Radio Signaling and Radar, High Definition Television, Video [Phil 4] etc. Usually their clock rates can be up to 100 MHz for common use, with faster rates can be found in some high-performance products [Phil 5]. Hardly ever, they are used in low frequency systems, such as: Seismic Modeling, Financial Modeling and Weather Modeling. In this three modeling systems, algorithms are complex and microprocessor need time to execute functions. It is possible to say that, if quantity of sample rate is growing, algorithm complexity is decreasing.

In audio applications it is common to use DSPs to implemented specific functionalities, like: speech synthesis, Hi-fi audio encoding and decoding, noise cancellation, audio equalization, sound synthesis and more.

The DSP market is very large and growing rapidly. According to the market research, sales of user-programmable DSPs in 1996 were about two billion dollars and in 2002 were around sixteen billion dollars. In Table 2 [Phil 10], are shown the most common DSP processors.

Vendor	Processor Family	Data Width
Analog Devices	ADSP-2Ixx	16
	ADSP-210xx	32
AT&T	DSP16xx	16
	DSP32xx	32
Motorola	DSP56xxx	16
	DSP563xx	24
	DSP96002	32
Texas Instruments	TMS320Clx	16
	TMS320C2x	16
	TMS320C3x	32
	TMS320C8x	8|16

Table 2 A selection of commercially available DSPs.

There are a lot of advantages in using DSPs. They are adapted for working with high frequency signals and for complex algorithms. Second feature is the ability to complete several accesses to memory in a single instruction cycle. This allows microprocessor to fetch simultaneously instruction, operand and write to memory.

One disadvantage of DSPs is that they usually have a small number of peripheral circuits included. They have memory, and sometimes in expensive versions, ADCs and DACs.

The second group of microprocessors here considered is that of microcontrollers. Microcontrollers can be considered as self-contained systems with processor, memories and embedded peripherals so that, in many cases, all that is needed to use them is to add software. They include one or more timers, an interrupt controller, and last but definitely not least general purpose I/O pins. They have been designed in particular for monitoring and/or control tasks [Gunt 7]. Microcontrollers also include bit operations which allow changing one bit within a byte without changing the other bits.

Following is a list of microcontroller manufacturers, and a selection of respective microcontroller series:

- Atmel (ATtiny, ATmega, ATxmega, AVR,AT91SAM)
- Intel (MCS-48, MCS-51, MCS-96)
- Microchip (PIC10, PIC12, PIC16, PIC18, PIC24, PIC32)
- NXP Semiconductors (80C51, ARM7, ARM9, ARM Cortex-Mx)
- STMicroelectronics (STM32, STM8, ST10)
- Texas Instruments (TMS370, MSP430, TMS320F28xx)
- Freescale Semiconductor (M-CORE,68HCxx, MPC 860)

In microcontrollers are available peripherals like:

- USB 2.0-compliant full-speed device
- CAN module
- UART modules
- SPI modules
- I2C™ modules
- Parallel Master and Slave Port
- Timers/Counters
- Compare/PWM outputs
- External interrupt
- I/O pins

The last group of microprocessors considered is Digital Signal Controller (DSC). These are built on a single chip that incorporates features of microcontrollers and DSPs. By combining the processing power of DSPs with the programming simplicity of a microcontroller, DSCs can provide high speed and high flexibility at low costs A further advantage is that there are relatively easy to program in programming language like C or Assembly. They include many peripherals (not as many as microcontrollers). A selection of DSC commercial devices is presented in Table 3.

Vendor	Device	Clock Speed MHz	Flash (kB)	PWM channels
Microchip	dsPIC30F	30	6-144	4-8
	dsPIC33F	40	12-256	8
Texas Instruments	TMS320F280x	60-100	32-256	16
	TMS320LF240x	40	16-64	7-16
Freescale	MC56F83x	60	48-280	12
	MC56F80x	32	12-64	5-6
	MC56F81x	40	40-572	12

Table 3 DSC Devices from the three different vendors.

There are also dedicated chips for audio purposes, like the TMS320C672x [Tex 3], which are characterized by a particular architecture. They have implemented three multichannel audio serial ports and operations, which make them ready for working with stereo audio signals. For communication with external devices, these are prepared with two SPI serial

ports, and two inter-integrated circuit (I2C) ports. For additional memory (if necessary, because internal size is about 256K-byte for RAM, and 384K-byte for ROM) it is used an external memory interface for a 100MHz SDRAM (16- or 32-bit). They are also equipped with real-time interrupt counter/watchdog.

II.2.2. Analog to Digital Converters

Analog to Digital Converters are devices which are used to convert analog signals to an equivalent digital form. In electronic literature these are usually referred by their abbreviation: ADC or A/D converter (sometimes only A/D). Since the ADC handles both analog and digital signals, its internal hardware is divided in two parts. In the analog part, the input voltage signal (which is referred to the V_{SS} analog ground reference) is sampled by a sample-and-hold block. This sample-and-hold is controlled by a clock signal that dictates when each sample must be taken – that is, it imposes the sampling frequency. Each sampled voltage value is then converted to the digital domain by a specific mechanism (which usually requires a reference voltage V_{REF} for such conversion). The digital samples are then ready to be sent to the controlling device (usually, a microprocessor) by means of different communication protocols (parallel or series – such as I2C, I2S, and others).

The history of first A/D converters begins at the middle of the twentieth century. The first one was called “Flash ADC” whose construction was very simple. It contains only a voltage divider (made of resistors), comparators and a decoding circuit (made from transistors). The advantage of this solution is its ultra-low conversion time. On the other side, there is the severe disadvantage that the number of required comparators raises dramatically with the bit resolution. In fact, the number of resistors (for the voltage divider) and comparators is 2^N-1 for each, where *N* is the number of bits of the ADC. Moreover, the number of required transistors (in the decoding block) is $\frac{N \cdot 2^N}{2}$. Nowadays, Flash converters are built for a maximum 5 - 6 bit resolution.

Generally, in high quality audio application it is needed to have a 16 bit resolution or more, and a reasonably fast speed of conversion. There are three main groups of ADCs, which are suited for these requirements: Sub-ranging especially with Pipeline architecture; Successive approximation; and Sigma-Delta.

The first group is characterized by a block diagram which is shown in Figure 7 [Wal 63].

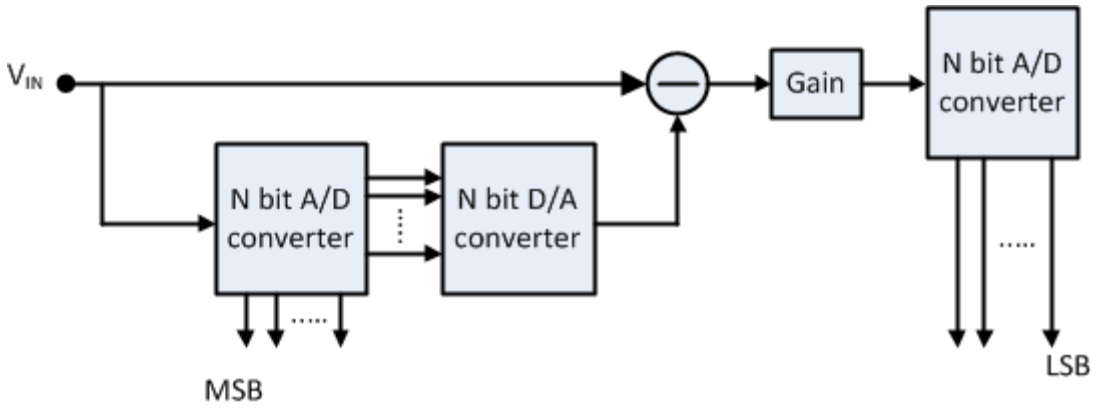


Figure 7 N-bit Two-Stage Sub-ranging ADC.

The input is first converted by an N-bit A/D converter (it can be used a Flash ADC when N is small enough), resulting on the N most significant bits (MSB) of the output register. Then, such digital value is converted back into analog format by an N-bit DAC and subtracted from the input, this gives a residue for next stage. Afterwards, with an appropriate gain operator, the residual analog signal is again converted by another ADC with the same resolution (N). This produces the least significant bits (LSBs) of the output register, forming a complete 2N resolution conversion. Although this description considered only a two-stage conversion, more stages can be considered (which, for the same final resolution, requires an ADC at each step with lower resolution N).

The pipelined architecture allows the previous stage to process the next sample at the same time that the current stage is still processing the current sample. At the end of each phase of a particular clock cycle, the output of a given stage is passed on to the next stage using the T/H (Track and Hold) functions and new data is shifted into the stage.

This method is shown in Figure 8 [Wal 71], where converters with identical stage are presented. There is also the possibility to combine non-identical stages, which is required in odd resolution converters.

The effect of “pipeline delay” in the output is shown in Figure 9 [Wal 69]. Delay is a function of the number of stages and blocks. Advantage of this architecture is that in one time are converted more than one samples of the signal, which makes these converters really high speed. For example, the 12-bit AD9235 (from Analog Devices) has efficiency up to 65 Mega Samples per Second (MSPS).

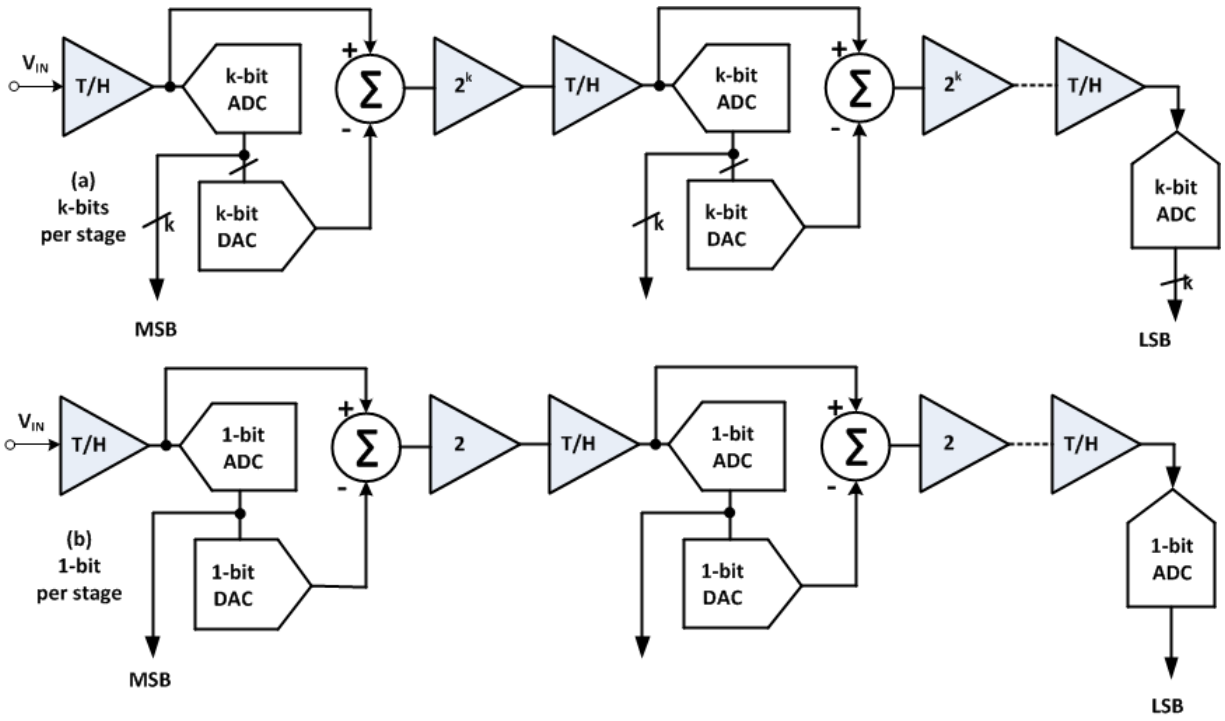


Figure 8 Basic Pipelined ADC with Identical Stages: a) k-bits per stage b) 1-bit per stage.

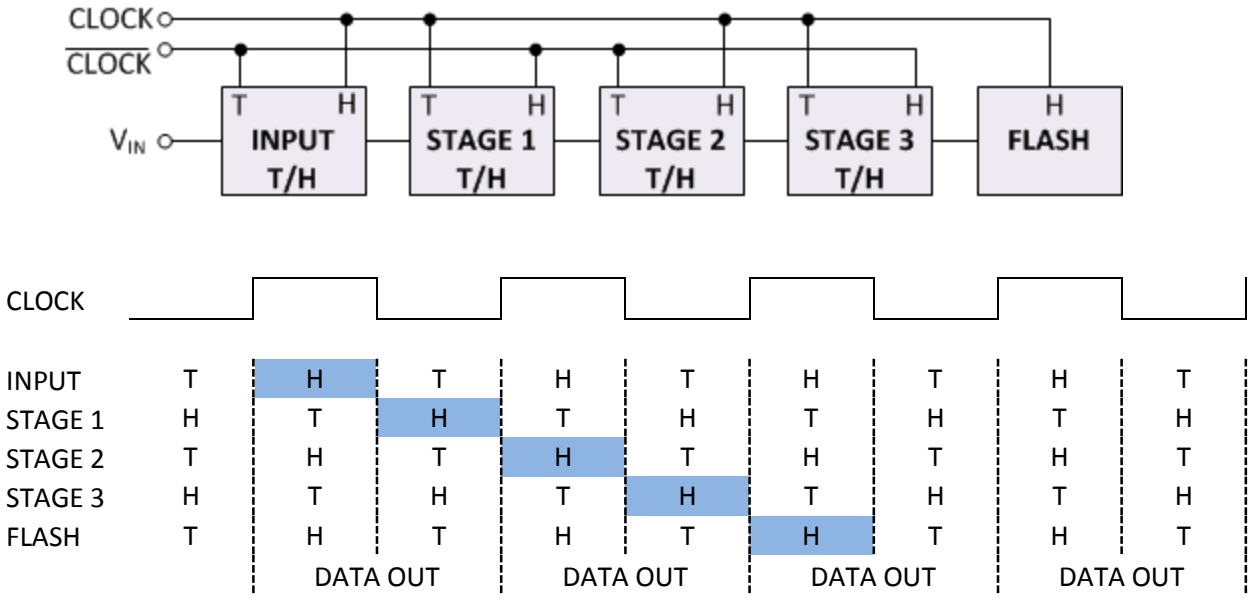


Figure 9 Example of operation in Pipelined ADCs.

The second group of ADCs here considered is the Successive Approximation ADC, sometimes called SAR ADC. The operation principle is shown in Figure 10 [Wal 54].

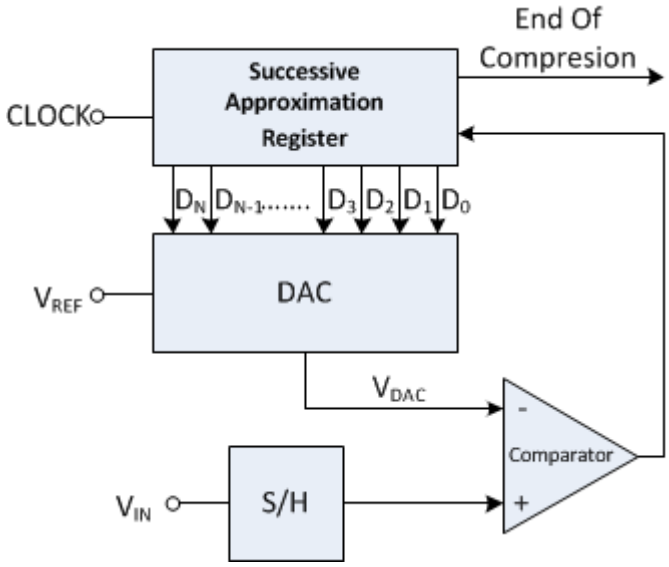


Figure 10 Successive Approximation ADC Block Diagram.

This device makes use of the successive approximations algorithm. The idea is simple. Firstly, the input signal passes through the Sample-and-Hold (S/H) block, where it is sampled and latched. For better explanation, let us analyze a particular example where V_{IN}= 43 V, V_{REF}= 32 V and N=5 (6 bit conversion). Firstly, SAR sets the MSB and input of the DAC is:

D_{543210}=100000 → V_{DAC}= 32 →V_{COMP}=1, because V_{IN}> V_{REF} and this value is latched in SAR register as D_5, and the next bit is then set to "1",

D_{543210}=110000 → V_{DAC}= 32+16=48 →V_{COMP}=0 and this value is latched as D_4,

D_{543210}=101000 → V_{DAC}= 32+8=40 →V_{COMP}=1 and D_3=1,

D_{543210}=101100 → V_{DAC}= 32+8+4=44 →V_{COMP}=0 and D_2=0,

D_{543210}=101010 → V_{DAC}= 32+8+2=42 →V_{COMP}=1 and D_1=1,

D_{543210}=101011 → V_{DAC}= 32+8+2+1=43 →V_{COMP}=1 and D_0=1.

After conversion SAR sets a signal EOC (End of Compression) and the output of this register is the final result, which in this situation is 43_{10} = 101011_2.

In successive approximation ADCs, most often Binary-Weighted DACs, are used, which are described in Section 2.2.3. Nowadays for this solution, manufacturers produce Capacity Binary-Weighted DACs, whose architecture is shown in Figure 11 [Wal 56]. The main reason

for using capacitors instead of resistors is the possibility of making smaller Integrated Circuits.

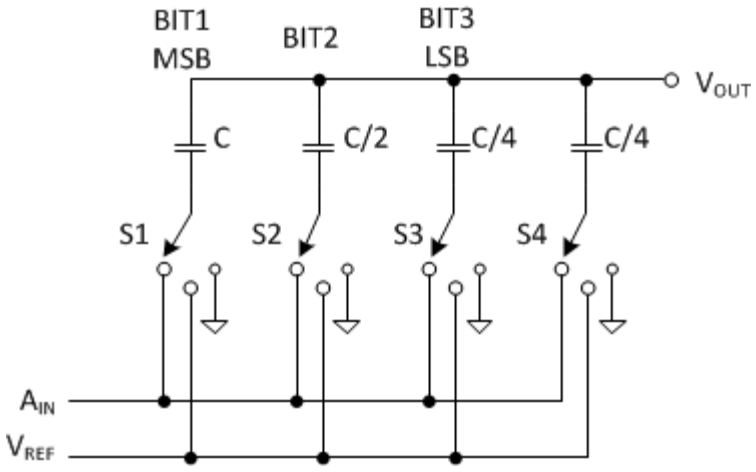


Figure 11 Capacitive Binary-Weighted DAC in Successive Approximation ADC.

The third group is Delta Sigma converters, which sometimes are called over sampling converters. Sigma-delta converters, which are shown in Figure 12[Wal 111], consist of 3 major blocks: modulator, digital filter and decimator. The modulator includes an integrator and a comparator (or N-bit flash ADC) with a feedback loop that contains a 1-bit (or N-bit) DAC. The modulator is oversampling the input signal, transforming it to a serial bit stream with a frequency K times above the required sampling rate. The output digital filter reduces bandwidth and then decimator converts the bit stream to a sequence of parallel digital words at the sampling rate. Decimation does not cause any loss of information.

Biggest advantages of them are high resolution (up to 24 bit) and low cost. Even if key concepts are easy to understand, mathematical calculation is complex. The penalty paid for the high resolution achievable with sigma-delta technology has always been speed. These converters have traditionally been used in high-resolution low frequency applications (such as speech, audio, precise voltage and temperature measurements).

Another advantage of these converters is that they shape the quantization noise, so that most of it falls outside digital filter pass band. When number of order sigma delta converter is rising, SNR is also rising. Number of sigma-delta blocks, defines order of converter.

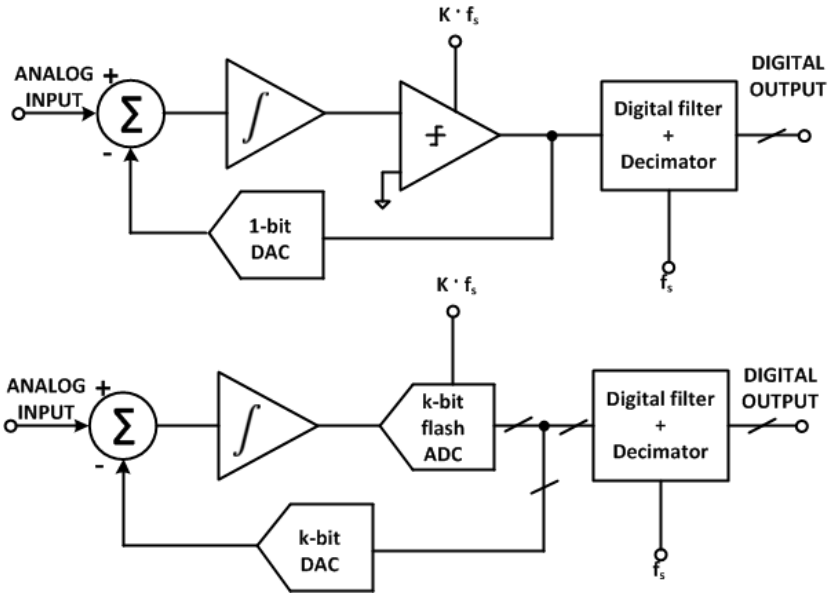


Figure 12 First order single and multibit Sigma-Delta ADC block diagram.

II.2.3. Digital to Analog Converter/ DAC

The simplest DAC structure of all is the divider or string DAC [Wal 4]. An N-bit version of this DAC simply consists of 2^N equal resistors in series (they are located between V_{REF} and V_{SS}) and 2^N switches (usually CMOS), one between each node of the chain and the output. The output is taken from the appropriate tap by closing just one of the switches. Relation between N-bit digital value and chosen switch bases on principle of operation N to 2^N decoder.

Exponential growing number of required resistors and switches, in this type of DAC, is big disadvantage for higher N-bit resolution. It is possible to say that evolution of them was binary-weighted DACs. They need only N resistors and N switches, as it is shown in Figure 13 [Wal 10]. Change of switch position, supply resistor adequately V_{REF} or ground for corresponding resistor. Values of resistors are successive power of two. To MSB is connected resistor with highest value. Output voltage is a sum of particular resistors.

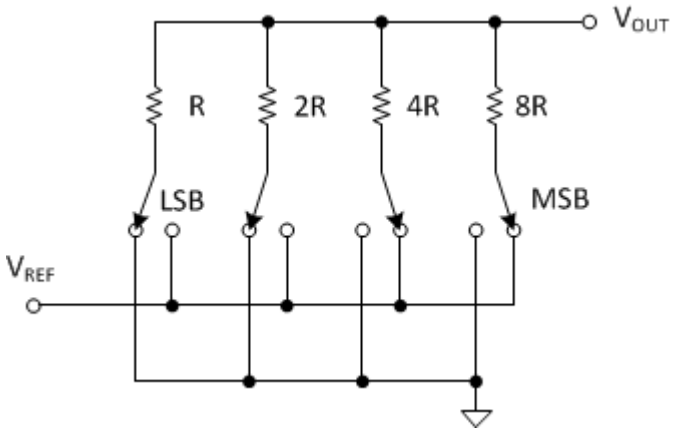


Figure 13 Voltage-mode Binary-Weighted Resistor DAC.

Binary–weight resistor DAC can be manufacture in current-mode. There is another modification of this type DACs structure, which has recently become widely used. It is Capacitive Binary-Weighted DAC, which is shown in Figure 11.

Another group D/A converters are segmented DACs. They consist of two or more String DAC. This solution gives possibility to obtain required performance. In Figure 14 is presented 6-bit segmented DAC composed of two 3-bit string DACs. To understand this clever concept better the actual voltages at each of the taps has been labeled.

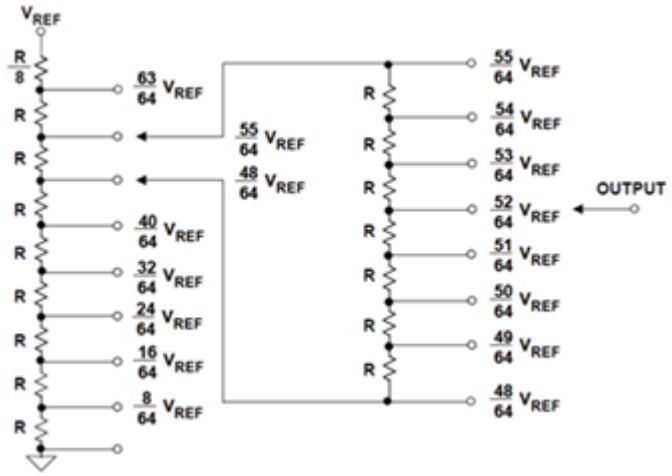


Figure 14 Segmented Unbuffered String DACs.

The resistors in the two strings must be equal. One resistor placed at the top first String DAC must be smaller—$1/2^K$ of the value of the others. Number of resistor responsible for MSBs must be equal 2^N, but number of resistors responsible for LSBs is equal 2^N-1. At two extreme points of second String DAC occurs voltage taken from output of first. Output signal is taken from adequate resistor.

Sigma-Delta DAC is a next presented group of D/A converters. In Figure 15 [Wal 133] is shown theirs block diagram. It consists of four blocks. First, interpolation filter is a digital circuit which accepts data at a low rate, inserts zeros at a high rate. Then it applies a digital filter algorithm and at output occurs high rate data. After is a digital Sigma-Delta modulator, which effectively works as a low pass filter to the signal but as a high pass filter to the quantization noise. Third block is an M-bit DAC, which convert digital stream to 2^M analog levels. After that output is filtered in an analog low pass filter to remove high frequency noises, which come from digital part of device.

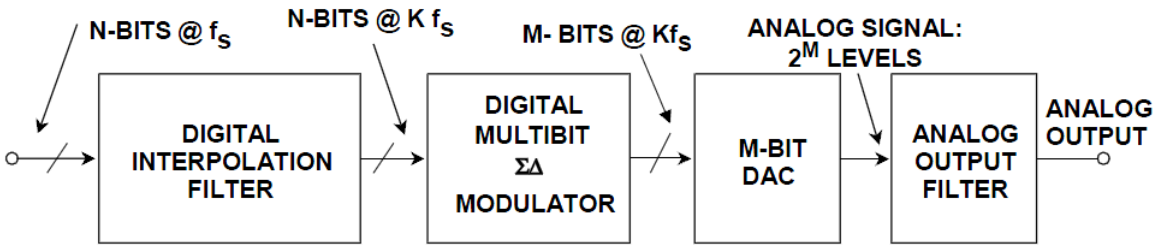


Figure 15 Delat Sigma DAC.

II.3. The usual communication protocol between Microcontroller and ADC/DAC

There are two possibilities of data transmission between microcontrollers and ADCs or DACs:

- Serial transmission –bits are sent sequentially, one bit after another, in one channel (wire). This makes it really cheap in design, but on the other hand it is a slow process.
- Parallel transmission – bits are sent simultaneously on different channels (wires). This makes this a fast process, but designing is expensive.

There are two types of serial transmission: synchronous and asynchronous. In this first one, groups of bits are combined into frames and frames are sent continuously. In the second one, groups of bits are sent as independent units with start/stop flags. There is no synchronization. In both groups sometimes between frames are sent bits of acknowledgment.

In audio signal systems, in serial synchronous transmission, there are three main standards:

- I^2S
- I^2C
- SPI

II.3.1. I^2S

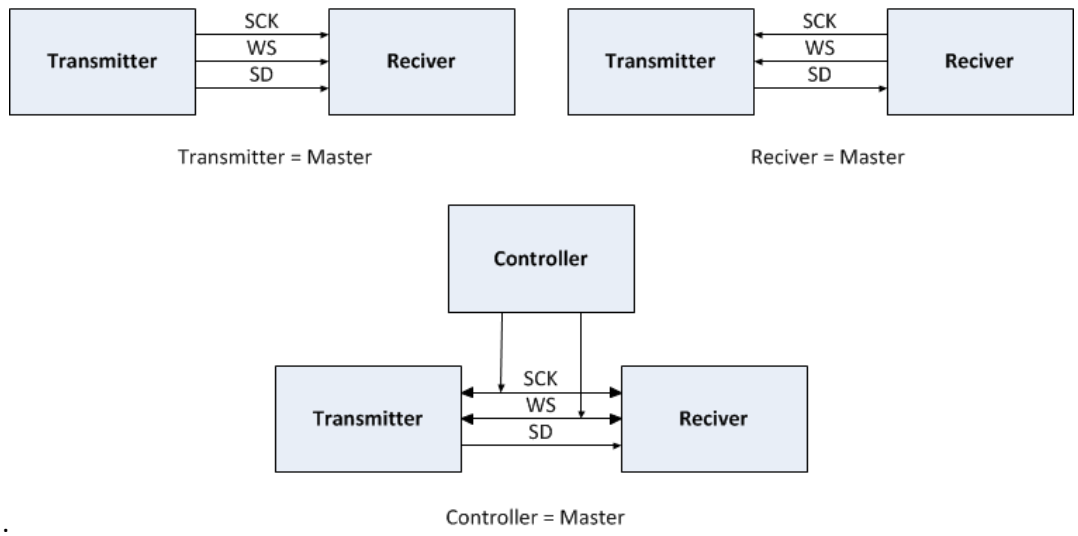
This is a standardized communication structure for exchanging information between two digital devices. Inter-IC sound (I2S) bus is a serial link especially for digital audio. It is used in Audio Signal Microprocessors, Analog-to-Digital converters, and Digital-to-Analog converters. They are also used in professional audio applications like: mixers, effects boxes, audio synthesis, audio conferencing, [I2S 1] etc. This standard is compatibility with stereo signals.

To minimize the number of pins required and to keep wiring simple, a 3-line serial bus is used consisting of a line for two time-multiplexed data channels, a word select line and a clock line.

The I2S protocol allows different connection configurations between devices. Figure 16 presents some examples of such configurations.

The I2S specification considers 3 lines [I2S 2]:

- continuous serial clock (SCK)
- word select (WS) - indicates which channel is being transmitted:
- WS = 0; channel 1 (left)
- WS = 1; channel 2 (right)
- serial data (SD)



.

Figure 16 I^2S system configuration.

When the transmitter and receiver have the same clock signal for data transmission, and if transmitter is the master, the latter has to generate the bit clock, word-select signal and data. But if the receiver is the master, it has to generate only the bit clock and the word-select.

In complex systems, where there may be several transmitters and receivers, it is difficult to define the master. In such systems, there is usually a system master controlling the digital audio data-flow between the various ICs. It generates only SCK and SD.

Serial data is transmitted in two data frames, one for left channel and one for right channel, which each one of them starts with MSB. MSB is transmitted first because the transmitter and receiver may have different word lengths. It is not necessary for the transmitter to know how many bits the receiver can handle, nor does the receiver need to know how many bits are being transmitted.

In that situation, when word length is not the same:

- transmitter sends more bits than receiver can handle, bits after LSB are ignored

- transmitter sends less bits than receiver needs, this needed bits are padded with ‘0’

For being sure that the most important bits are sent and received, the transmitter always sends the MSB of the next word one clock period after the WS changes.

Serial data sent by the transmitter may be synchronized with either the trailing (HIGH-to-LOW) or the leading (LOW-to-HIGH) edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal.

II.3.2. I²C

This specification for data transmission needs only two wires:

- SDA; serial data

- SCL; serial clock

Both SDA and SCL are bi-directional lines, connected to a positive supply voltage via a current-source or pull-up resistor. When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function. A typical connection is shown in Figure 17 [I2C 8].

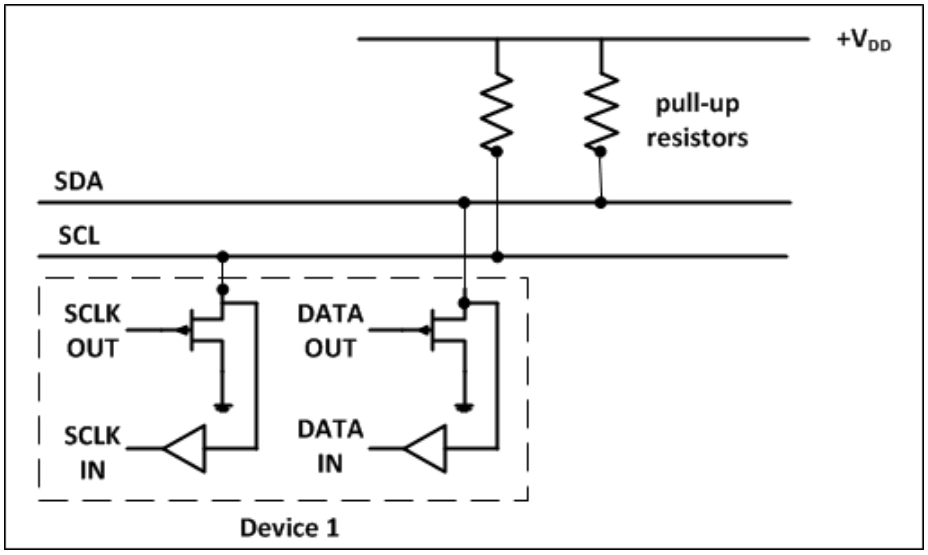


Figure 17 Recommended connection for I^2C.

Like in I^2S, in I^2C there are also the distinct concepts of: transmitter, receiver, master and slave. There are also more two important definitions: multi-master and arbitration [I2C 7]. The first means that the bus can be controlled by different devices at different times. The second means that if more than one master tries to control the bus, only one is allowed to do that

.Messages, which were not serviced, are waiting to control the bus. In Figure 18 is shown an example where more than one master are present.

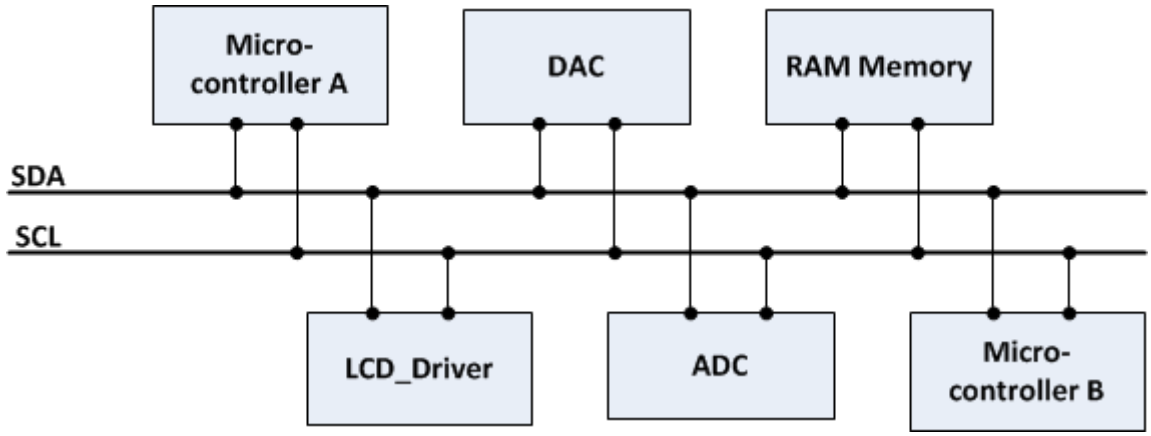


Figure 18 I^2C connection between 6 devices.

Another difference between I^2S and I^2C is that in the second one we can have more than one receiver and more than one transmitter. But this generates a problem. Which device will be transmitting and which will be receiving at a particular moment?

Each device is recognized by a unique address. When the master (in this case, a microcontroller) starts a session, it sends on the bus the address of the device which will be working in accordance with its purpose. As we can see in Figure 19 [I2C 10], after this address, it sends a bit R/(~W) which indicates that master will be writing on reading. The ninth bite is an acknowledgment.

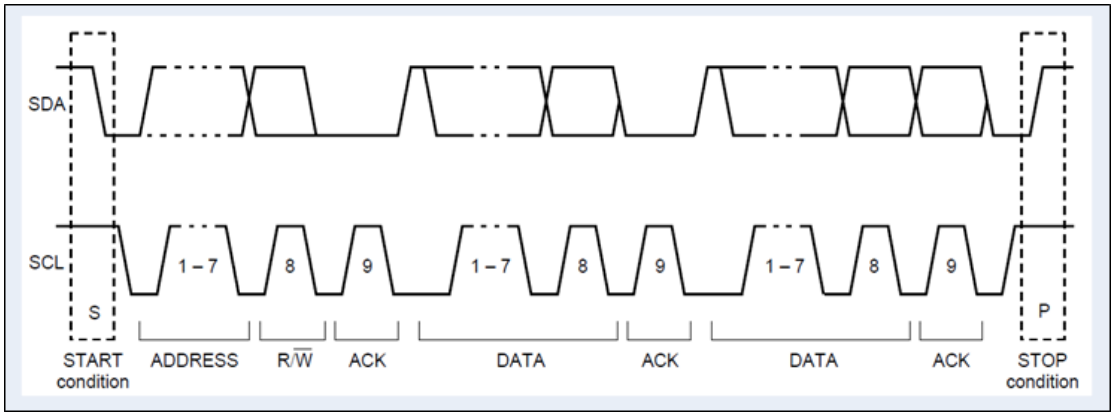


Figure 19 A complete data transfer with START and STOP conditions in a I2C bus.

II.3.3. SPI

This standard needs 4 wires [Ser 3]:

- SCK – Continuous clock
- SI – Data input
- SO – Data output
- CS – Chip select (active low)

It is possible to divide this line into two groups: control lines (SCK, CS) and data lines (SI, SO).

In Figure 20 is shown an example of a possible configuration with 1 master and 2 slaves.

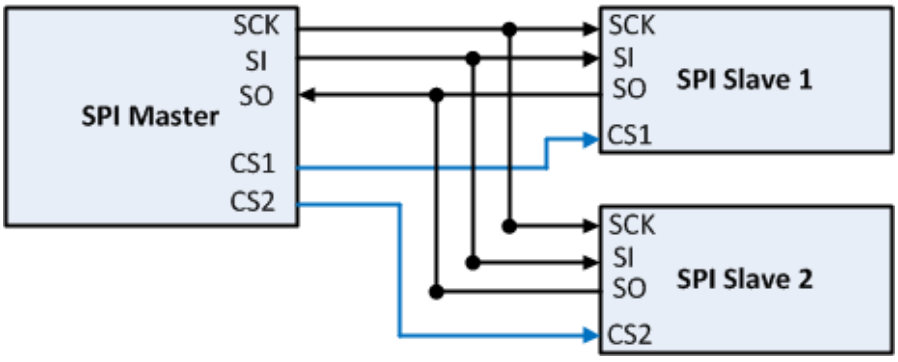


Figure 20 Multiple slave SPI implementation example.

The master decides with which peripheral device it wants to communicate. In this simple way a system can be organized with several slave devices. Another difference between I^2C and SPI is that the SPI bus does not need any external pull-up resistors.

In Figure 21 [Pic 23] is shown typical SPI transmit/receive mechanism. As we can see, at one time instant, a device can only get or send information. Reading and writing services use corresponding for them buffers.

The sizes of buffers are usually: 8 / 16 / 32 bits. If our data resolution is between these three values, we always choose the next higher value. We have also the possibility to decide on what edge of clock signal data will be transmitted.

A pair of parameters, called clock polarity (CKP) and clock edge (CPHA), determines the edges of the clock signal on which the data are driven and sampled. Each of the two parameters has two possible states, which allows for four possible combinations, all of which are incompatible with one another. So a master/slave pair must use the same parameter pair values to communicate. These are shown in Figure 22.

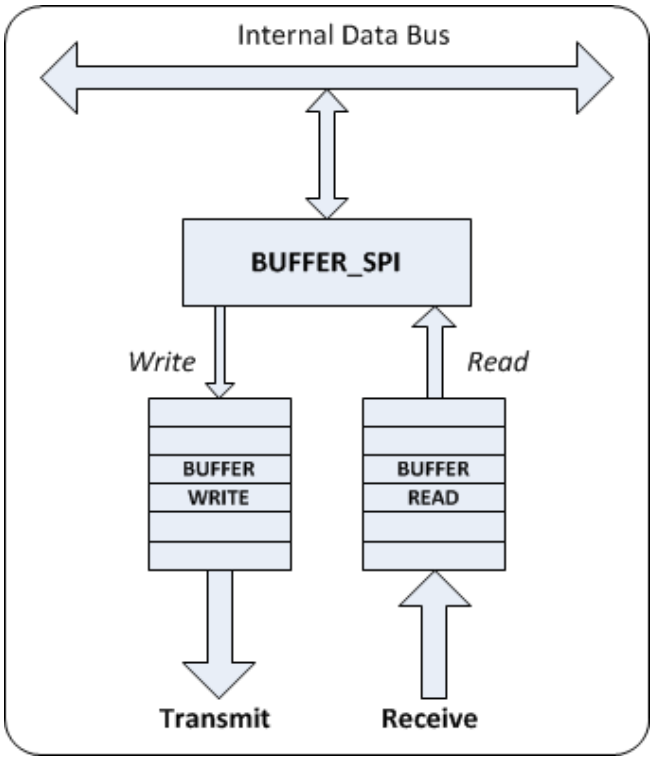


Figure 21 SPI transmit and receive mechanism.

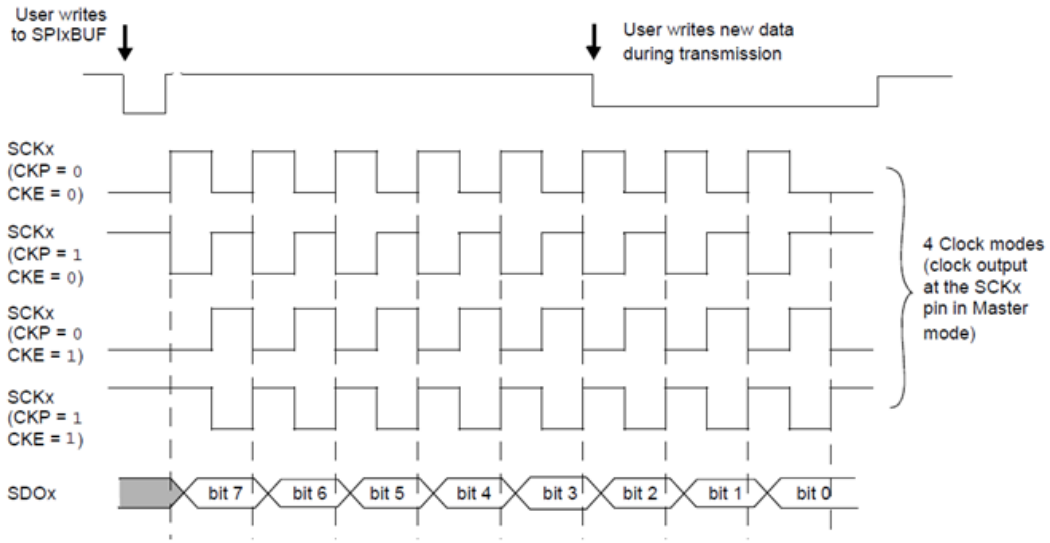


Figure 22 Typical SPI timing in sending information.

SPI	***I^2C***
Three bus lines are required: -SCK -SI -SO for more devices is also need : CS	Two bus lines are required: -SDA -SCL
No official specification (created by Motorola, but every device can have his own specification)	Official specification (I^2C was created by Philips Semiconductor)
Higher data rates (even up to 10MHz)	Lower data rates (400kHz)
Efficient in configuration: single master and single slave	Efficient in configuration: multi-masters and multi-slaves
Master is not using address to connect with device	Master is using address to connect with a specific device
Does not have an acknowledgement mechanism to confirm data reception	Has an acknowledgement mechanism to confirm data reception

Table 4 General comparison between SPI and I2C protocols.

In Table 4 [Ser 11] is presented a comparison between the SPI and I2C communication protocols. Contrary to I2S, which is very specific to audio dedicated devices, these two protocols are the most commonly encountered in microcontrollers and peripheral devices (as ADCs and DACs).

II.4. Signal conditioning of input and output channels of audio devices

Conditioning systems prepare an analog signal in such a way that it meets the requirements of the next stage for further processing. They normally consist of electronic circuits performing functions like: amplification, level shifting, filtering, impedance matching, modulation, and demodulation.

In Figure 5 in Section 2.2 it is shown where input and output signal conditioning is placed in the signal processing chain. The input signal conditioning block manipulates the input signal so that the ADC works with maximum range. The voltage waveform generated from a musical instrument of from any audio device (such as an MP3 player) has positive and negative voltage values, usually oscillating around zero Volt. Moreover, the maximum amplitude of such waveform may also vary depending on the input device. Since the ADC has a full voltage range from V_{SS} to V_{DD}, which is usually not centered on zero Volt, the signal conditioning block must shift the dc value of the input waveform and also perform gain adjustment to take the profit of the ADC resolution.

The output signal conditioning block ensures voltage level matching to next stage component, which can be, in audio applications, a speaker, an audio amplifier, a mixing table, or any other audio processing device. In some cases, like the one of the work described in this work, the audio signal processing unit changes the properties of the signal originated from the source (like a musical instrument) and must deliver an output signal that has the same electrical characteristics of the input signal (same voltage range, same dc value). Is this case, the output signal conditioning block implements the reverse operations from those of the input signal conditioning block.

The input and output signal conditioning blocks should also filter the frequency components of the signals being processed which are not to be considered in the application. In audio applications, the input block should filter frequencies outside the audio range (or, if the device is specific to a musical instrument, for instance, it could have a narrower filter to limit the pass-band to that of such instrument. This will contribute to noise reduction and, consequently, to a better quality processed signal. The output signal conditioning block

should also implement a similar filter, especially to attenuate the quantization noise generated by the analog to digital conversion.

II.5. Audio effects

Nowadays, there are a lot of audio effects which are used in sound production. They arose from the moment of origin instruments and from the beginning of music. Some of them are for general purpose, others are destined for specific instruments. In this section, are presented only the most common audio effects encountered in audio processing devices.

Delay/echo –the original audio signal is followed closely by a delayed repeat, just like an echo. The delay time can be as short as a few milliseconds or as long as several seconds. It depends where the virtual sound barrier, which reflects the audio signal, is. A delay effect can include a single echo or multiple echoes.

Multiple echoes occur, everywhere where more than one sound barrier is present. Delay also forms the basis of other effects such as chorus, reverb, flanger and phaser (flanging and phasing).

Chorus effect was designed to make a single person's voice sound like multiple voices singing the same thing. It can also be used for musical instruments. The main idea is making a soloist into a chorus. The original signal is duplicated, by adding multiple delays, which are short. This creates the effect of many sources, and each source is very slightly out of time and tune. This makes an illusion which we are thinking it is like in real life. There are some common parameters of this effect:

- The number of source, that we want to hear
- Delay between sound and its duplication , which is typically 20 to 30 milliseconds
- Modulation waveform for generating time-varying delayed samples.

The delays which are added to original signal are not constant. Because they are changing, it makes the effect more real.

Flanger (or flanging) is an effect which is technically very similar to chorus. It is a mixture of the original and delayed signal. There are two differences between chorus and flanger. First concerns about speed of changing delay, which in flanger it is slower than in chorus. Second one is that delayed sample are taken form output, not form input like in chorus. It is worth to add that this effect produce a large number of notches (in frequency characteristic), which create musically related peaks, between those notches.

Phaser(or phasing) is an audio effect which takes advantage of the way sound waves interact with each other when they are out of phase. By splitting an audio signal into two signals and changing the relative phase between them, a variety of interesting sweeping effects can be created. When the signal is in phase (at 0 degrees, 360 degrees and 720 degrees) the signals reinforce, providing normal output. When the signals are out of phase (180 degrees and 540 degrees), they cancel each other. Number of producing notches is relatively small in comparison with flanger. A phaser is characterized as being harder and more pronounced than a chorus, but gentler than a flanger

Vibrato- an effect which is especially known in stringed (only with neck) and wind instruments, relies on regular pulsating change of pitch. Musicians make it by special playing technic. For instance, guitarists stop the finger on a string, and wobbled on the fingerboard, or actually moved up and down the string for a wider vibrato. Players of wind instruments generally create vibrato by modulating their air flow into the instrument. There are two main factors characterizing this effect. First is the “extent of vibrato” - the amount of pitch variation, and second is “rate of vibrato” – speed with which the pitch is varied.

Reverb– short from reverberation, is an effect which simulates what happens with sound in different areas. It refers to the way sound waves reflect off various surfaces before reaching the listener's ear. The most common are: hall reverb, concert hall reverb, small room reverb, etc. There are also deviations from these, like: empty hall reverb and occupied hall reverb.

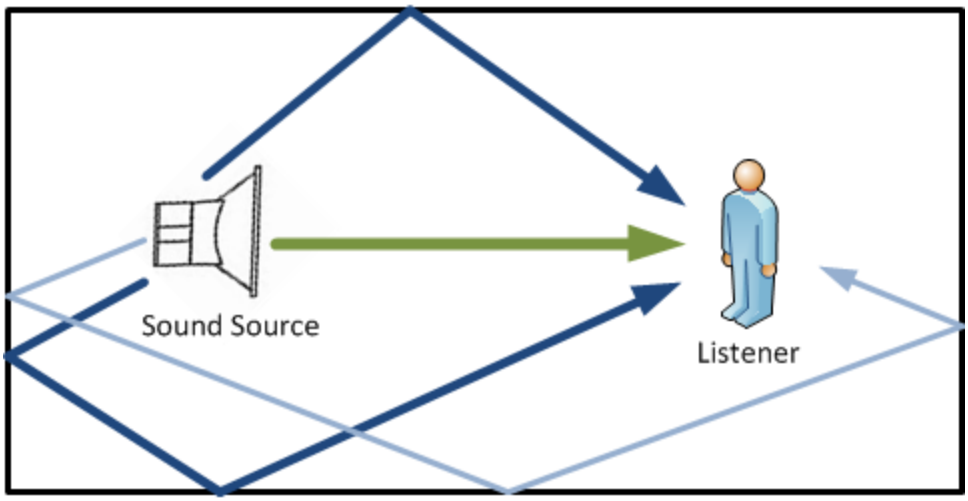


Figure 23 Way of propagation waves in a closed room.

In Figure 23 is shown how is created the reverb effect. The lines are a graphical representation of waves which are fetched up to the listener’s ears. The green one is called

main sound, which has the loudest level. Others (shown as blue), called lateral sounds, are weaker. When sound waves reflect off walls, they need longer time to reach the listener. With every bounce they lose energy, so become less audible.

Overdrive and distortion- these two effects are of the most common effects for electric guitars. The main idea is to compress the peaks of sound waves, and add overtones. The sound is created as fuzzy. Distortion is a little harder sound, good for rock music, while overdrive gives a more natural compression sound. In Figure 24 is shown the process of altering an audio signal called clipping. Differences come from that overdrive is formed by soft clipping, while distortion is formed by hard clipping.

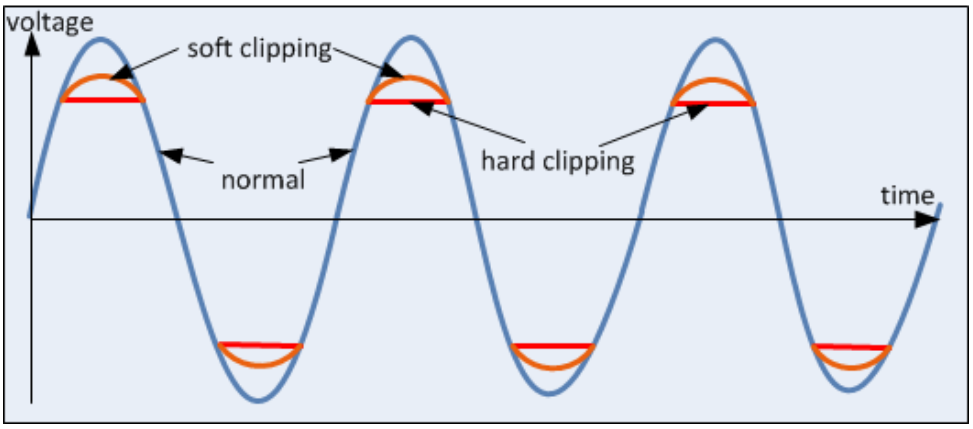


Figure 24 Soft and hard clipping.

There are three general parameters to operate this effect:

- Drive (Gain) controls the amount of overdrive
- Mix level – show how much is mixed original and distorted sound
- Volume of distorted sound

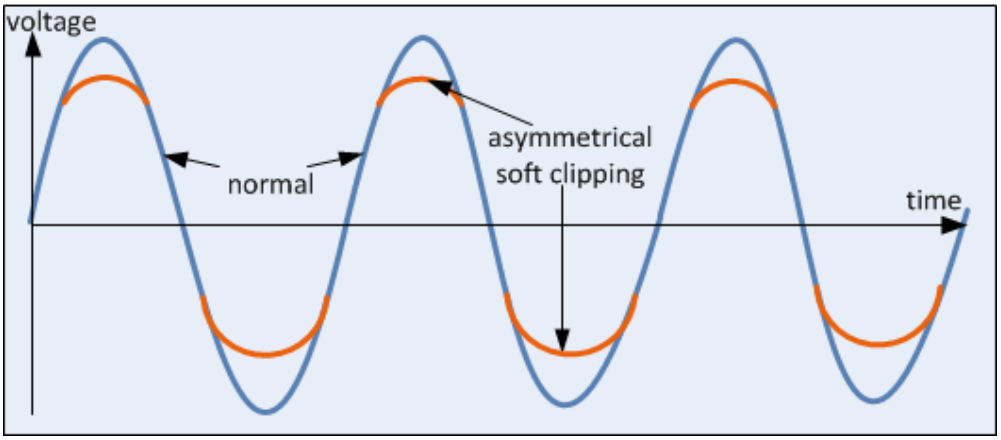


Figure 25 Asymmetrical soft clipping.

There is the possibility to create some variations from these effects. One of them (shown in Figure 25) is to make asymmetrical overdrive. For this purpose there is another control parameter. The difference between them is that, symmetrical is smoother and less toothy, while asymmetrical has more aggressive sounding

Tremolo modulates the volume, like rapidly turning up and down the volume to give pulsating effect. In almost all situations is used for more than one musical instrument or audio channel. For example, it can be used to give the impression of one instrument to be appearing and disappearing against the background of other instruments. For example, guitarists used tremolo mixed with reverb to produce the sound typical of the 60's and 70's (many times denominated as surf music).

There are two common parameters in tremolo control:

- Frequency with which volume is turning down and up
- Depth is the maximum and minimum level of volume variation.

Panning is the consolidation of two tremolo effects, one for the left channel, and another for the right channel (or sometimes one for one instrument and the second for another one). They are linked together and when left volume channel is high, right volume channel is low, and vice-versa. When input signal is in stereo format, the listener can feel that the sound "moves" from one side to the other.

WahWah effect was made popular by guitarists like Jimi Hendrix and Eric Clapton. It imitates human voices saying "wawa". This effect has two parts: in the first on the speaker hears something similar to an open "a", while in the second human ear hears something

between a “u” and a “w”. It is implemented by moving a peak in the frequency response up and down the frequency spectrum, as shown in Figure 26.

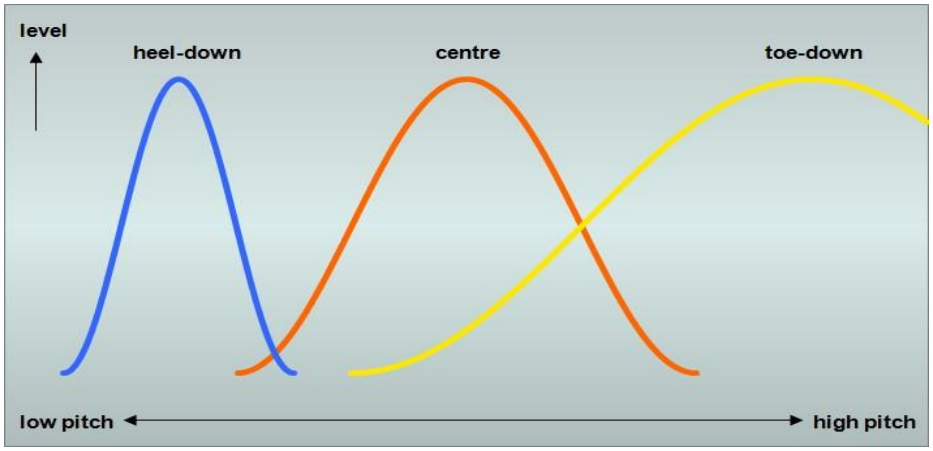


Figure 26 WahWah effect illustration.

Equalization – this effect gives the possibility to control tunes by shaping the audio spectrum, by enhancing some frequency bands and attenuating others. It is realized by a series connection of first or second order shelving and peak filters, which are controlled independently. Schematic representation is shown in Figure 27 [Udo 51].

Shelving filters boost and cut low and high frequency bands, with cut-off frequency f_C and gain G controlling parameters, while peak filters are destined for mid frequencies, being controlled by the cut-off frequency f_C, bandwidth f_b and gain G. Peak filters are defined by the Q factor, which is given by $Q = \frac{f_b}{f_c}$.

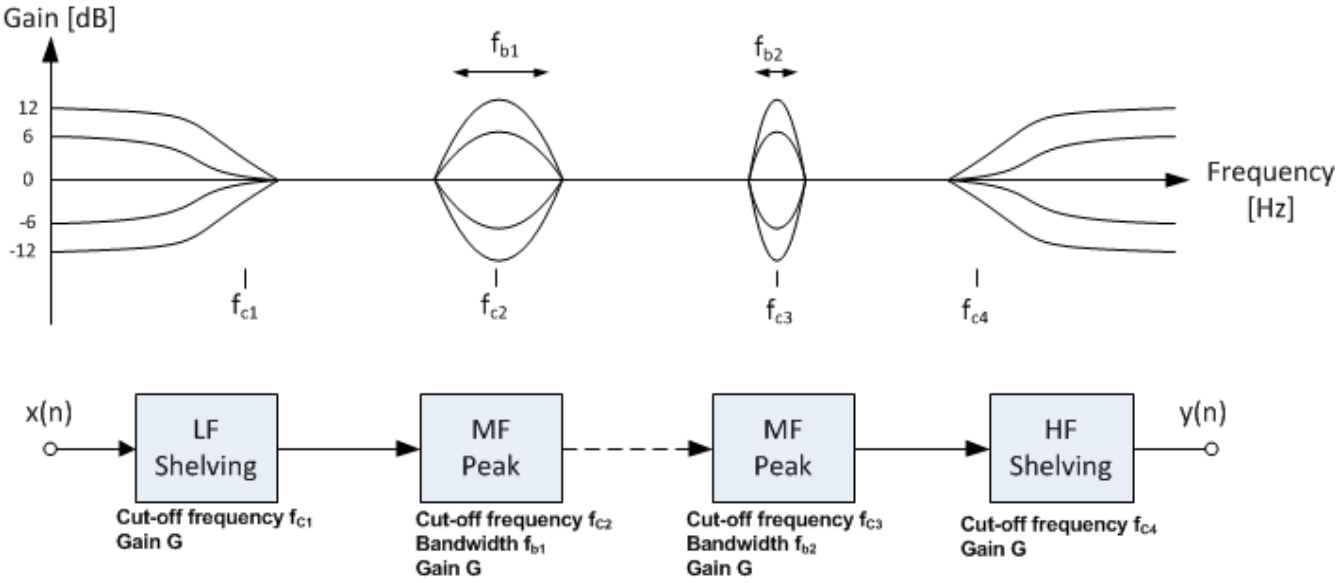


Figure 27 Series connection of two shelving and two peak filters.

Octave– a simple effect that generates a signal one or two octaves below or above the original signal. Some effect units are mixing the normal signal with the synthesized octave signal, which is derived from the original input signal by halving or doubling the frequency.

Only in field of guitar effects there are over one hundred manufacturers of audio effects devices. Their history began at the middle of the twentieth century. They initially based their devices on the vacuum tube, but with the development of semiconductors, they moved to transistors. The first vendors were Warwick Electronics and Univox, which started to produce stompboxes, which are known as “effects pedals”. In the 1980s, these units began being digitalized, because of increased flexibility.

Chapter III:
Description of the Designed Part of the Work

Designed part of work can be divided into two sections. First section applies to description of PIC32 Module, which is a board included microcontroller from Microchip manufacturer, made in 32bit architecture. It includes indispensible components which are needed to start working with microcontroller. Second section is about Audio Effects Unit and shows individual schematic of each block and connection between them.

III.1 PIC32 Module Design

PIC32 Module is designed for general purpose for all students, which are interested in programming microcontrollers. From educational point of view there is no necessity to use 100 pins microcontroller, which has more peripheries. Microchip company offers easily programmed USB connection, which makes this board more universal for all fields of electronic. It will be used in Audio Effect Unit and Wireless Audio Unit Projects.

III.1.1. Choice of microprocessor (why the PIC32MX795F512H)

Microcontroller, that will be used, has to include:

- Two channels SPI, one for receiving data from Analog to Digital converter, and another one for sending data to Digital to Analog converter. Length of SPI buffer must be minimum 16bit
- Implemented USB 2.0 On-The-Go Peripheral with integrated physical layer. Device will be working as a host.

- Possibly the highest working speed, for process audio effects algorithms.
- Four external interrupts, to ensure possibility of changing effects, when device has been already programmed.
- Data memory equals 17,4kB. 15kB is necessary to realize delay effect and 2,4kB for store samples, which are used in chorus effect.
- Timer, with throwing interrupt, for set up sampling frequency.

There are available five 32bit Microchip microcontroller families. Only three of them perform above-mentioned brief foredesign: PIC32mx5xx, PIC32mx6xx and PIC32mx7xx. Differences between them are in number of peripherals. If prices of them are similar (which take place), it will be best to choose microcontroller with the highest number of included circuits and with the biggest size of memory. PIC32MX795F512H is most developed one.

III.1.2. Schematics

In **Błąd! Nie można odnaleźć źródła odwołania.** is shown schematic of PIC32 Module. Pins VDD must be supplied by voltage range of 2.3V to 3.6V, and pins VSS must be connected to (digital) ground. Main components that are used, to ensure correctly started with 32bit microcontroller:

- Y1 – is an external main oscillator (OSC) which value equals 8MHz. PIC32 has single cycle multiply and hardware divide unit, which gives possibility to set up clock frequency to 80MHz. Oscillator pins are required to connect to capacitors in 22 pF to 33 pF range. (C8=22pF and C10=22pF). This value of oscillator ensures possibility of using PICkit3 Programmer.
- Y2 – secondary oscillator (SOSC) is designed specifically for low-power operation with an external 32.768 kHz crystal. It can also drive Timer1 and the Real-Time Clock and Calendar (RTCC) module for Real-Time Clock (RTC) applications. Capacitors C11 and C12 have the same values - 22pF.
- S1- switch is set between VSS and Master Clear Pin, which provides two specific device functions: device reset and device programming and debugging. This input is level sensitive, it requires a low level to create reset.
- C6 – capacitor, which value equals 10µF, is required on the VCAP/VCORE pin, which is used to stabilize the internal voltage regulator output.
- C1, C2, C3, C4 – decoupling capacitor which must be connected to all VDD pins, Typical value of them is 100nF.
- C5 – decoupling capacitor which is connected to AVDD, even if the ADC module is not used.

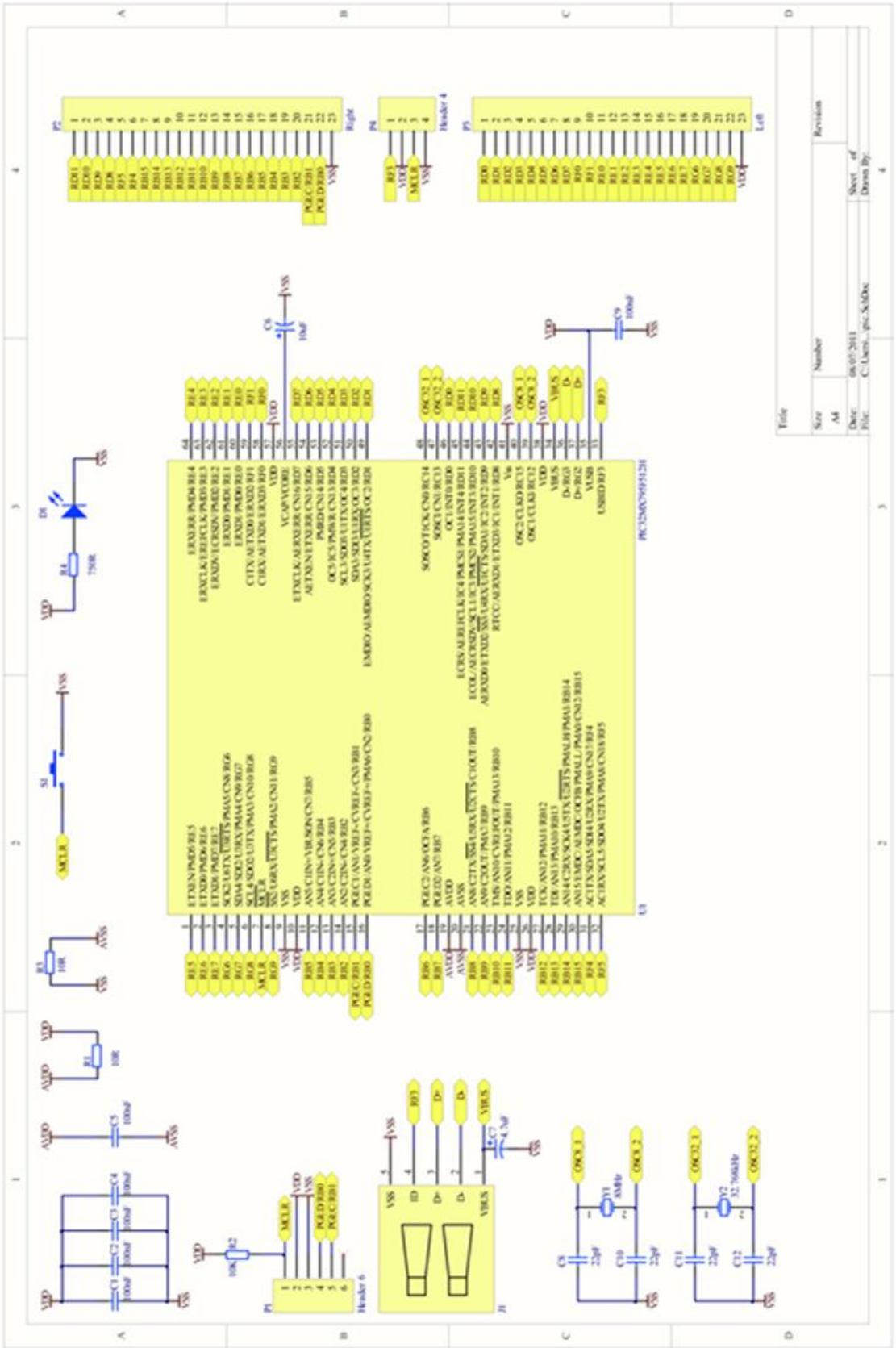


Figure 28 Schematic of PIC32 Module.

- R1, R3 – resistors which are recommended between analog and digital voltage supply. Also between analog and digital ground.
- D1 – LED diode (with appropriately match resistor R4) is signalizing that circuit is supplied

Right and left headers (P2, P3) are connected to microcontroller pins in such way: I/O ports are ordered in growing or descending sequence. That gives the opportunity of easy connection between associate devices and PIC32 Module. Header P4 is destined for simply integrated circuits, which require voltage supply, ground and one signal pin. It will be useful for testing them without additional components. They can be supplied from programmers which are connected to computer by USB.

Header P1 is a connector between PICkit 3 Programmer and microcontroller. Pins PGED/B1 and PGEC/B0 are responsible for transmitting information between them. It is necessary to put pull-up resistor R2=10k to MCLR pin. Sequence of pins is taken from instruction of programmer.

USB2.0 implemented in PIC32mx795f512, allow using PIC32Module like a host, device and “On the Go” device with bidirectional data transfer. This last type requires five wire connector [Pic 27], which is common, called Mini A-B. On wires D- and D+ data are transmitted. Pin USBID is used to detect OTG transmission. Pin VBUS is analog input and serves power monitoring.

III.1.3. PCB

During designing board the following rules were kept:

1. Decoupling capacitor was placed closely to microcontroller.
2. Capacitors, responsible for correct working of oscillators, were placed closely to its pins.
3. Wires feeding oscillations to microcontroller have almost the same length.
4. Under the oscillators, others wires were not leaded.
5. Signal with high frequency were leaded without using vias.
6. Polygon was removed from areas, where pins occur close to each other.
7. Ground wires were not leaded, because polygon was placed.
8. Spacing, between header pins, must be a multiple of 100 mil. With this rule, PIC32 Module board can be easily place at testing board.

Design rules for board manufacture are:

1. Width for power wires: typically 20 mil , 9.5 mil (for microcontroller output pins)
2. Width for signal wires: typically 10 mil, 9.5 mil (for microcontroller output pins)

3. Minimum clearance for all signal: 10 mil
4. Minimum hole size (diameter): 18 mil
5. Minimum hole to hole clearance: 10 mil

PIC32 Module PCB layout can be found in appendixes.

III.2 Audio Effects Unit Design

III.2.1. General Architecture

General architecture of Audio Effects Unit is shown in Figure 29. Instead of "Microcontroller PIC32" block, PIC32 Module is put. Expected peak to peak voltage range of input audio signal will be 2 V, centered at 0V. First block must add DC value equaled reference voltage, lead to ADC, divided by two, and amplify signal to maximum range of converter placed in next stage. Output signal conditioning block must do opposite things to Input signal conditioning block: cut the DC value and amplify signal to expected input of stage after Audio Effects Unit.

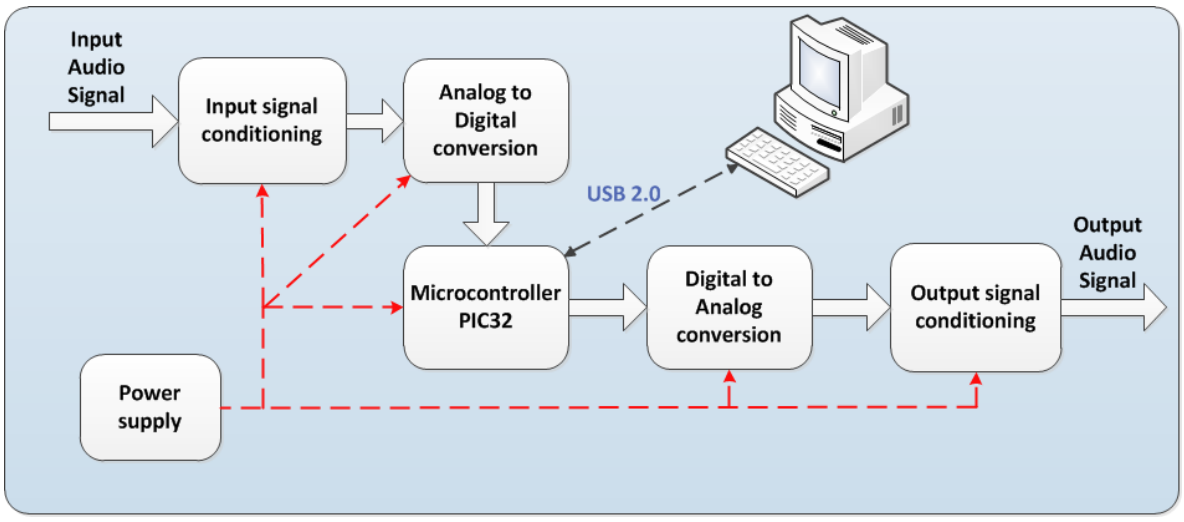


Figure 29 Audio Effects Unit block diagram.

III.2.2. Choice of Components

To create first and last blocks presented in Figure 29, were used operational amplifiers AD8531. It comes from Analog Devices manufacturer and need single supply voltage

from 2.7 V to 6 V. It has high output current: ±250 mA and low supply current: 750µA. Wide bandwidth up to 3 MHz ensures passing audio signal [AD 8531].

In both conditioning blocks were used lowpass filters from Linear Technology Company, LTC1569. They can set up cut off frequency, by one external resistor, up to 150 kHz on a single 3V Supply and up to 300 kHz on a single 5V supply. The LTC1569 is a 10th order filter featuring linear phase and DC accurate. It has good Signal-to-Noise ratio, equals 80dB. Figure 30 shows its frequency response [LTC 1569], for different cut off values.

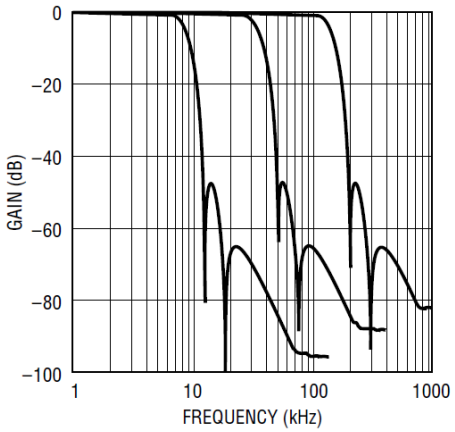


Figure 30 Frequency response, f_{cutoff}=8/32/128 kHz.

As Analog to digital converter, LTC1864L was chosen. Its maximum sampling frequency is 150 kHz with 16 bit resolution. It operates on a single 3-7V supply with maximum supply current 450µA (for 150 kHz, with 3V) [LTC 1864]. Reference voltage range is from 1V to voltage supply. Conversion takes minimum 2µs, and after this time ADC goes into sleep mode drawing only leakage current. It has compatible SPI I/O pins, which give possibility to use SPI channel, and not concentrate at acquiring data from ADC. It sends only 16 bit result of conversion without any additional bits, which allows using 16 bit SPI buffer. Receiving data takes less time, which gives more time to process mathematical algorithms. Maximum value of SPI clock frequency is 8 MHz, so storing data takes theoretically, 16 * ($1/f_{SCK}$) = 16*(1/*MHz) = 2µs time.

From the same company, LTC2641 was chosen as digital to analog converter. It can do operations from 2.7V to 5.5V single supply, with low current 120µA. Reference voltage range is from 2V to voltage supply. It has fast 1µs settling time to 16 bits, and maximum 16-Bit INL error: ±1LSB, over temperature [LTC 2641]. The LTC2641 uses a simple SPI compatible 3-wire serial interface which can be operated at clock rates up to 50MHz. But minimum working SPI clock frequency is 10MHz.

Similarly to ADC, DAC also requires only 16 bits without any additional bits. With this resolution, and this clock frequency, sending data takes 16*(1/f_{SCK}) = 16*(1/10MHz) = 1.6 µs. Unbuffered DAC has low offset error of ±1LSB.

As buffer of DAC it was used LTC6078, which is low noise operational amplifier with low power consumption 56µA (for 3.3V). It can be supplied voltage from 2.7 to 5.5V.

To produce 3.3 voltage supply DC/DC converter was used, with 91% efficiency. TSR 1-2433 is step-down switching regulator from TRACO POWER Company. Its input voltage range is from 4.75 to 36 V, with maximum output current 1A [TSR 1]. This component has short circuit protection, and his pins are compatible with LMxx linear regulators.

III.2.3. Schematics

To demonstrate a process, that is usual in engineering design - the first design is usually not perfect and needs improvements or corrections; this sub-chapter is divided to three parts. At the beginning first version of schematics will be described. Second part will include the list of changes, which were made, with justifications. At the end, final version of schematics will be presented.

On schematic nr 1 (Figure 31) power supply is shown, which produces voltage supply equals 3.3V. For guarantee, correct working of TSR 1-2433, on its input was placed electrolytic capacitor C1=22µF, for voltage rating 25V. Task of LED diode is to signal, that circuit is supplied.

Schematic nr 2 (Figure 31) presents block of input signal conditioning. After Jack socket, was placed capacitor, for blocking DC value. Resistor R2 and potentiometer R5 are responsible for setting correct gain value. From voltage amplification equation for operational amplifier working in inverting configuration (1), was calculate value of R3.

$$V_{OUT} = -\frac{R5}{R2} * V_{IN} \qquad (1)$$

Amplitude of input signal equals 1V, and expected output voltage must be 1.65V. Because R2 has value 1kΩ, R5 must be set to 1.65kΩ. Resistor R3 and potentiometer R4 ensures adding DC value which must be half of ADC reference voltage. After matching signal amplitude, active filter cuts unexpected frequencies. Resistors R6, R7 and capacitor C3 are responsible for biased internal ground for LTC1569. Their values are successively: 3.48kΩ, 2kΩ and 1µF. Setting cut off frequency, was calculated from equation (2).

$$f_{CUTOFF} = \frac{128\ kHz\ (10k/R_{EXT})}{1{,}4\ or\ 16} \qquad (2)$$

When the internal oscillator is enabled, pin 5 (DIV/CLK) can be used to engage an internal divider. The internal divider is set to:

- 1:1 when DIV/CLK is shorted to GND.
- 4:1 when DIV/CLK is allowed to float, with connection to 100pF capacitor.
- 16:1 when DIV/CLK is shorted to voltage supply.

This pin was connected to 3.3V, which makes that numerator will be divided by 16. R_{EXT} = 4kΩ was enumerated for 20 kHz audio range.

On ADC schematic nr 3 (Figure 31), can be seen, that reference input was shorted to voltage supply. The span of the A/D converter was defined to 3.3V, which makes value between levels of quantization equal 50.35µV. Pins IN+ and IN- are differential inputs of converter.

Input analog signal was leaded to positive pin, while negative pin was shorted to ground. Others pins are connected to microcontroller. Pin CONV is an input pin, and when high level will be on it, the ADC starts conversion process. A low level on this input, enables the SDO pin, allowing the data to be shifted out, with clock frequency, which appears on SCK pin.

On DAC schematic nr 4 (Figure 31), can be found sub-circuit responsible for digital to analog conversion without errors. Reference voltage was set to value of 3.3V. /CS pin serves for control working mode of D/A converter.

When /CS is low, SCLK is enabled for shifting in data on DIN. When CS is taken high, SCLK is disabled, the 16-bit input word is latched and the DAC is updated. Pin 5 is input asynchronous clear. Low voltage on it, gives at DAC output zero.

Schematic nr 5 (Figure 31) presents output signal conditioning block. The same filter, with the same values of components, was used in schematic 2. Filtered, from interfering frequencies become from working frequency of digital devices, signal must be amplified to expected value. Operational amplifier works as a voltage follower. Because amplitude must be reduced from 1.65V to 1V, resistor R20=1kΩ and potentiometer R21 make voltage divider.

$$V_{wy} = V_{we} * \frac{R_{21}}{R_{20}+R_{21}} \qquad (3)$$

Value of R_{21} can be easily calculated from equation (3), it is 1.52kΩ. Capacitor before jack socket is responsible for cut DC value.

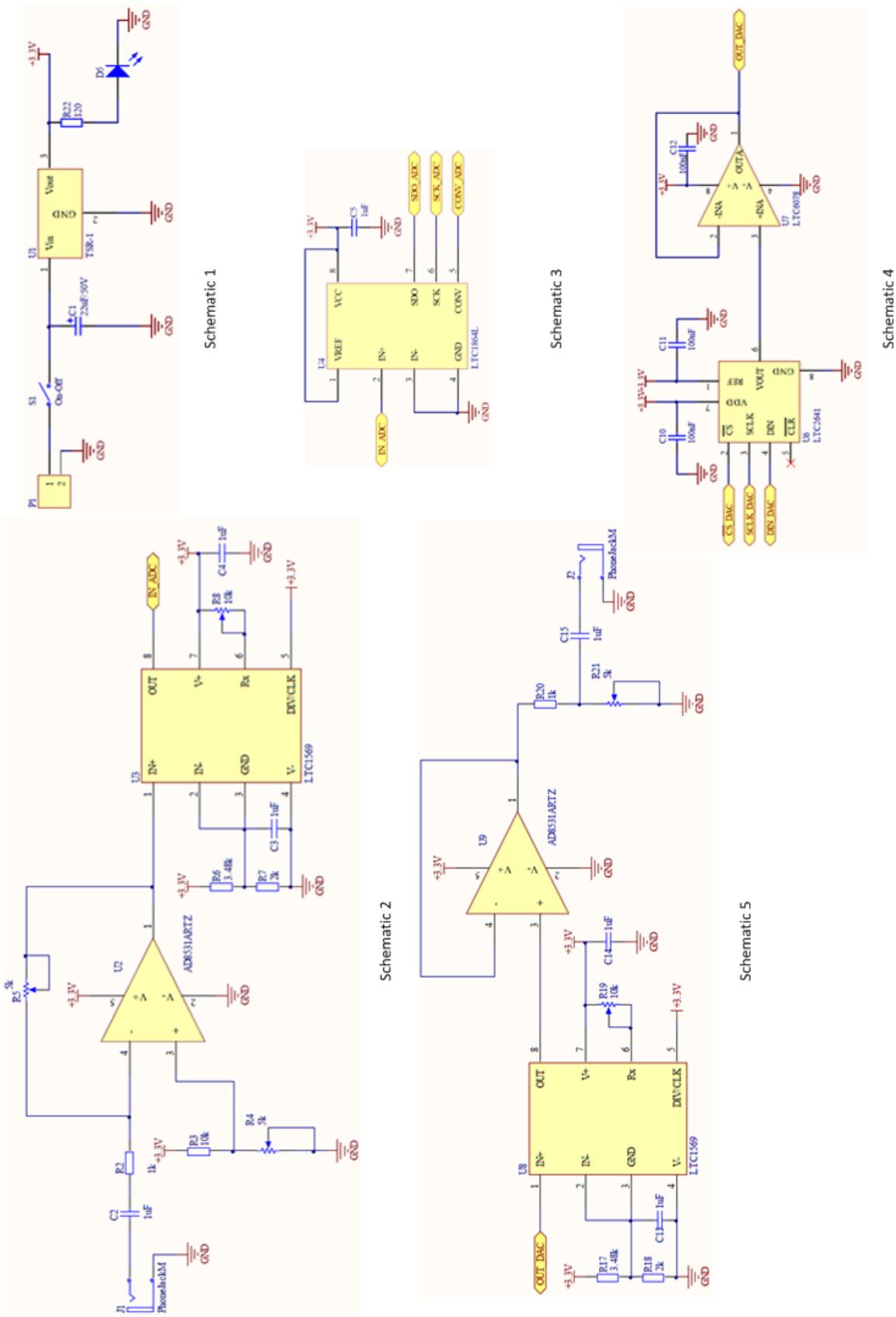


Figure 31 First Audio Effects Unit Schematics.

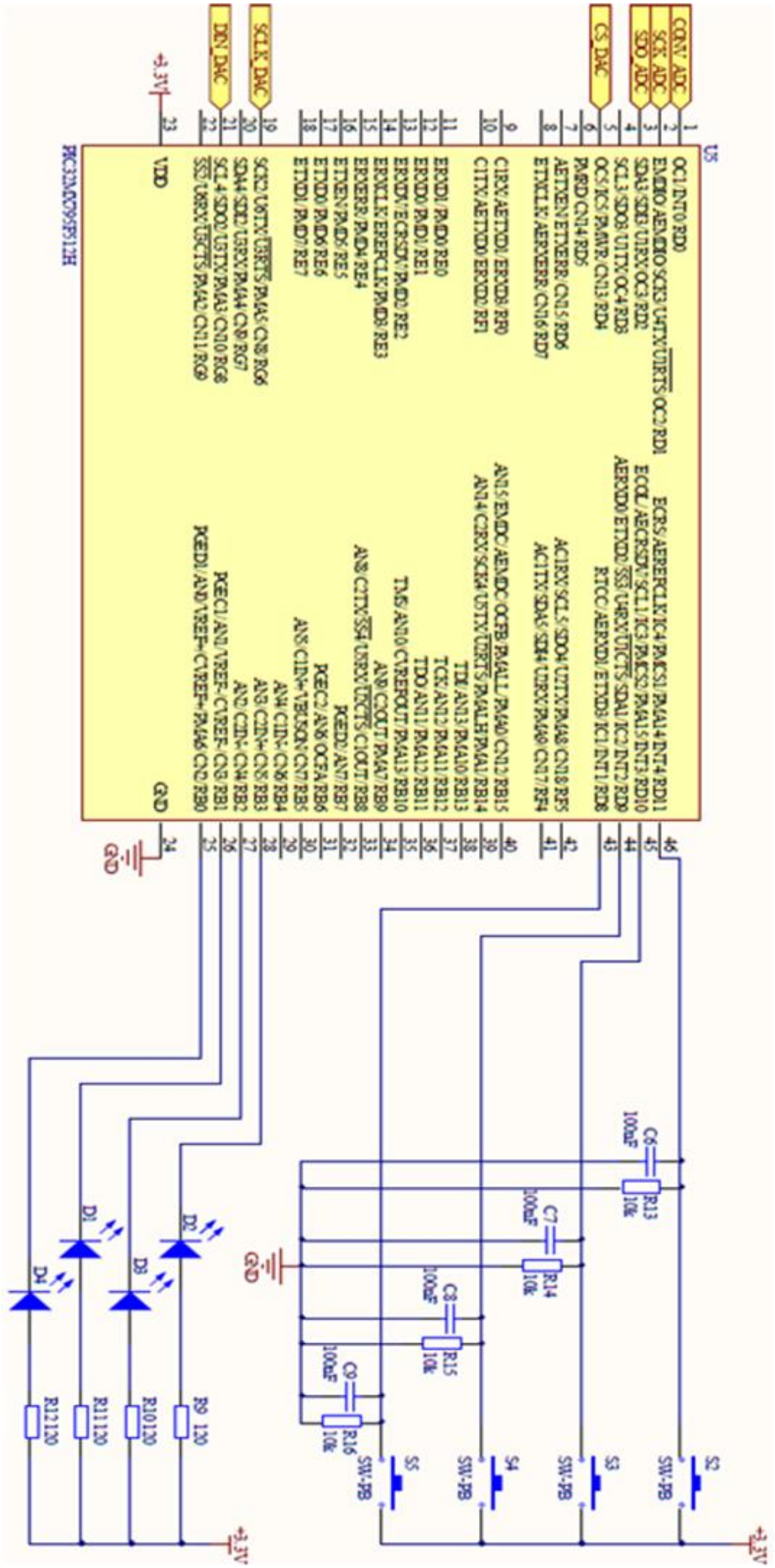


Figure 32 Connection between PIC32 Module and Audio Effects Unit.

Schematic nr 6 (Figure 32) presents PIC32 Module with input and output signals, which come from ADC and DAC. First four pins of Port B are used for switching on and off LED diodes, which will inform user about turned up effects. Four pins of Port D are used for capture external interrupts. They will come from switches S1 - S4. R13 – R16 are pull-down resistors. C6 - C9 are capacitors which are responsible for eliminating contact vibration of switches.

List of changes which were made is presented below.

- Capacitor C2 was removed. This one with potentiometer R4 made unexpected low pass filter, and signals with frequency below 400Hz were attenuated.
- Resistor R3 was replaced by new one with value 4.7kΩ, because, with old, was impossible to obtain required 1.65 V DC value on positive input of operational amplifier
- Both filters LTC1569 are supplied by 5V. Justification comes from the filter output voltage range, which is typically 50mV to (VDD – 0.8V). When filter was supplied by 3.3V, there was no possibility to prepare signal to maximum span range of ADC. This change applies to both filters used in Audio Effects Unit.
- Filter LTC1569 pin 5 (DIV/CLK), was connected by capacitor 100pF to ground. With previous connection maximum pass band was around 14 kHz. With new configuration is possible to set cut off frequency to 20 kHz.
- Analog to digital converter SDO and SCK pins became changed. When schematic of component was drawn in Altium Designer, numbers of pins were incorrectly attributed to names of pins.
- Digital to analog converter pin named /CLR was connected to voltage supply. Without this correction, ADC was working properly for 30 seconds, and after this time, low level (zero volts) was observed at the output. Reason of that integrated circuit behavior was incidental appearance low level of voltage on this pin.

In Figure 33, are presented corrected schematics of Audio Effects Unit. In connection between microcontroller and PIC32 Module there are not any changes. Also both versions of schematics can be found in appendixes.

III.2.4. PCB

During designing board the following rules were kept:

1. Decoupling capacitor was placed closely to integrated circuits.
2. Signal with high frequency were lead without using vias.
3. Polygon was removed from areas, where pins occur close together.
4. Ground wires were not lead, because polygon was placed.

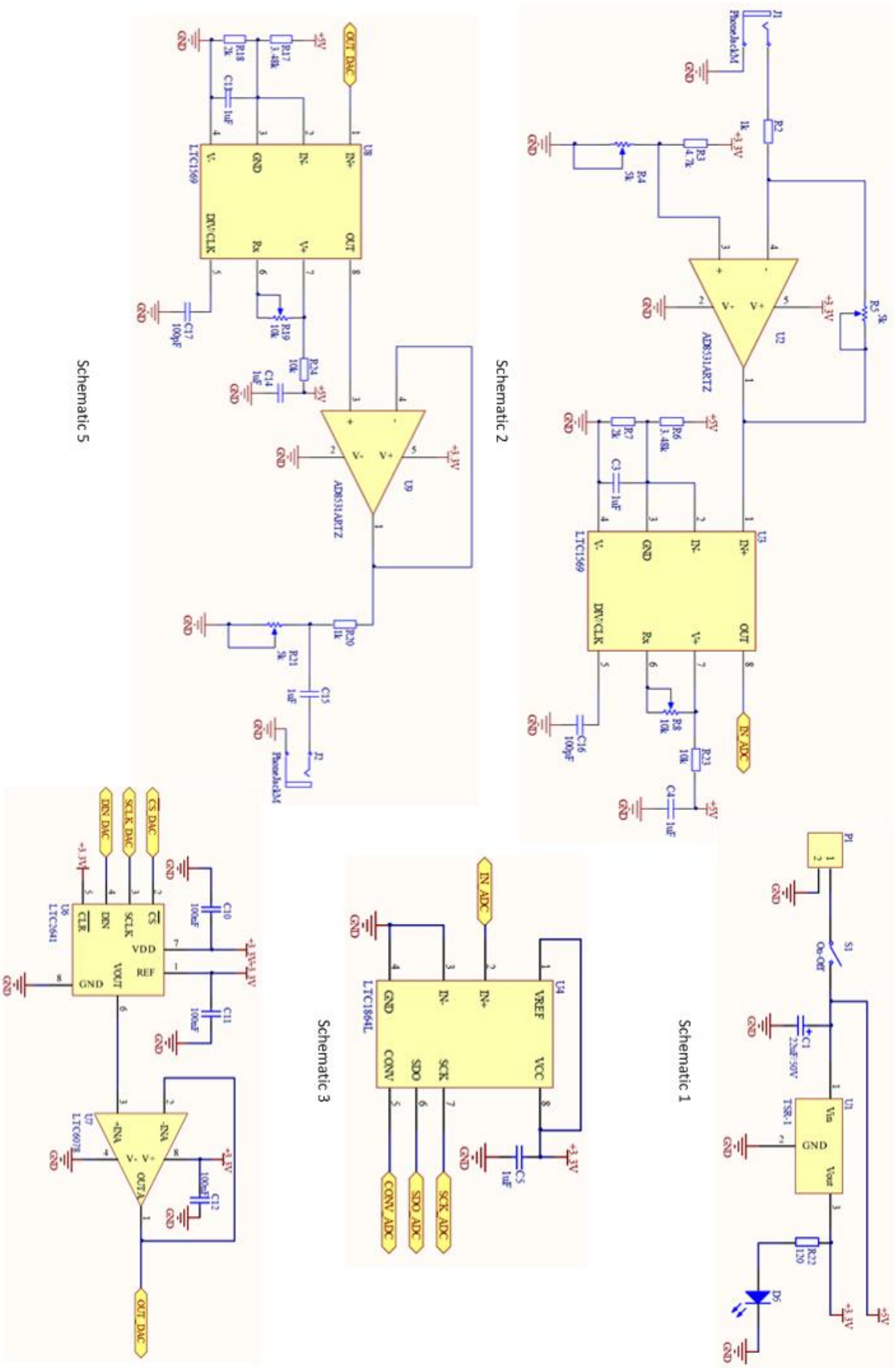


Figure 33 Corrected Audio Effects Unit Schematics.

5. For not soldering PIC32 Module pins, mainframe, consisted of two male to female headers, was used.
6. Many elements, connected with audio signal, were placed under the PIC32 Module, for making smaller dimensions of AEU Board.
7. Potentiometers were placed in way that designer has an easy access to them.
8. On board corners, holes for spacers which are used for mechanical assembly were placed.

Design rules for board manufacture are:

1. Width for power wires: typically 15 mil, 8 mil (for DAC output pins)
2. Width for signal wires: typically 10 mil, 8 mil (for DAC output pins)
3. Minimum clearance for power: 12 mil , 10 mil (for DAC and LTC 1569 filters)
4. Minimum clearance for signal: 10 milMinimum hole size (diameter): 22 mil
5. Minimum hole to hole clearance: 10 mil

Audio Effects Unit PCB layout can be found in attachments.

III.2.5. Final Assembly

Figure 34 presents connection between two boards: PIC32 Module Board and AEU Board. In second board two male to female headers are soldered, so PIC32 Module can be easily insert to its. For elements which are placed on enclosure, screwed joint were soldered to AEU board.

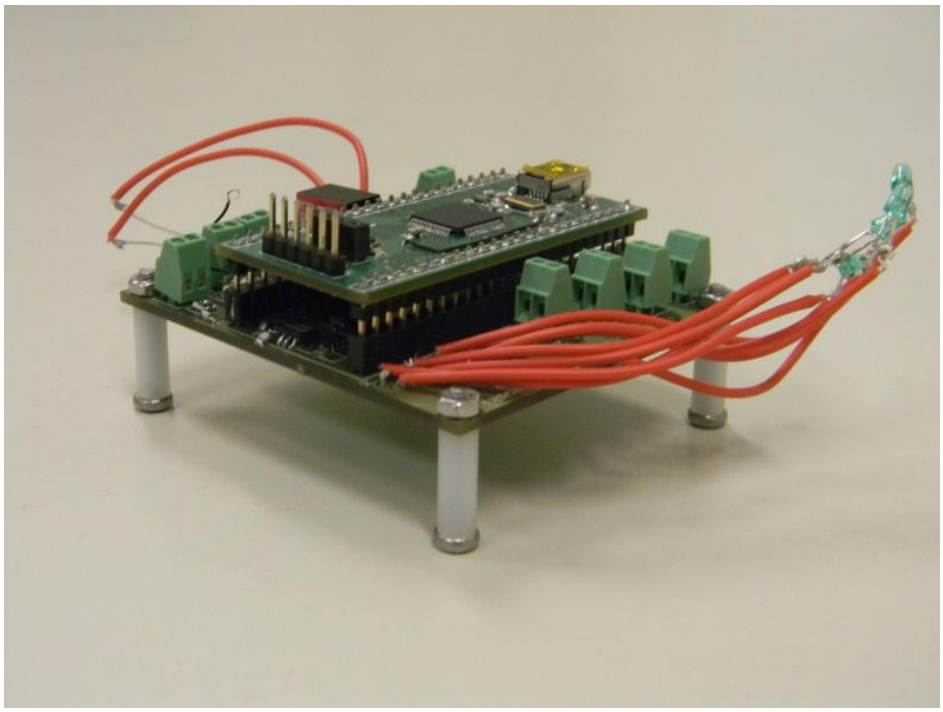


Figure 34 Executing circuit of AEU.

Figure 35 shows final assembly of AEU.

Figure 35 Final appearance of AEU.

Chapter IV:
Audio Effects Algorithms

In this chapter will be presented a way of implementation audio effects algorithms and their functioning results on audio signals, which will be illustrated in Language of Technical Computing – Matlab.

IV.1. Equalization effect

In Figure 27 was shown typical series connection of shelving and peak filters, which are common uses in equalizers. As well on the same figure, can be found expecting frequency spectrum of implemented filters. It is worth to add that when all filters have gains equal zero, signal is passing thoroughly without changes. In comparison to usually used filters, which after cut-off frequency attenuate signal to -60db (average value), shelving and peak ones dampen to 0dB.

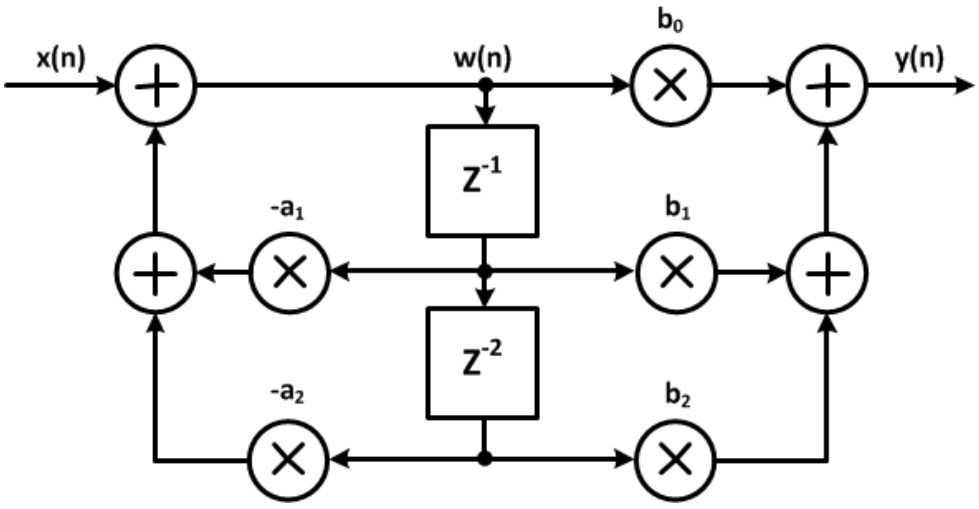


Figure 36 Canonical second-order digital filter.

In Figure 36 is shown second-order digital filter, which represents model of IIR filters [Udo 34]. Symbols b0, b1, and b2 are called non-recursive parameters and a1 and a2 recursive parameters. Output sample is a sum of input sample, two previous output samples and two previous input samples. Every sample is multiplied by corresponding coefficient. Equation (4) describes this model:

$$y(n) = b_0 * x(n) + b_1 * x(n-1) + b_2 * x(n-2) - \\ -a_1 * y(n-1) - a_2 * y(n-2) \qquad (4)$$

Computing each y(n) requires five multiplications and four additions. In sum, four samples of previous input and output need to be saved for each filter. If project requires low memory usage, it is possible to use equations (5) and (6), without any difference of functioning.

$$w(n) = x(n) - a_1 * w(n-1) - a_1 * w(n-2) \qquad (5)$$

$$y(n) = b_0 * w(n) + b_1 * w(n-1) + b_2 * w(n-2) \qquad (6)$$

These equations required exactly the same number of multiplications and additions, but only two states w(n-1) and w(n-2) are remembered on every loop.

In shelving and peak filters, there is no necessity to be distressed about gain of particular ranges, because filter coefficients are dependent on gain G, which must be defined in dB. In Table 5 [Udo 53] and Table 6 [Udo 55] are equations for calculate filter coefficients. For both is valid dependence (7).

$$K = \tan(\frac{\pi fc}{fs}), \qquad (7)$$

where: fc – cut-off frequency

fs – sampling frequency

low-frequency shelving (boost $V_0 = 10^{G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1+\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(V_0K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$
low-frequency shelving (cut $V_0 = 10^{-G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1+\sqrt{2}K+K^2}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{2(V_0K^2-1)}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{1-\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2V_0}K+V_0K^2}$

high-frequency shelving (boost $V_0 = 10^{G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{V_0 + \sqrt{2V_0}K + K^2}{1 + \sqrt{2}K + K^2}$	$\frac{2(K^2 - V_0)}{1 + \sqrt{2}K + K^2}$	$\frac{V_0 + \sqrt{2V_0}K + K^2}{1 + \sqrt{2}K + K^2}$	$\frac{2(K^2 - 1)}{1 + \sqrt{2}K + K^2}$	$\frac{1 - \sqrt{2}K + K^2}{1 + \sqrt{2}K + K^2}$
high-frequency shelving (cut $V_0 = 10^{-G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1 + \sqrt{2}K + K^2}{V_0 + \sqrt{2V_0}K + K^2}$	$\frac{2(K^2 - 1)}{V_0 + \sqrt{2V_0}K + K^2}$	$\frac{1 - \sqrt{2}K + K^2}{V_0 + \sqrt{2V_0}K + V_0K^2}$	$\frac{2(K^2/V_0 - 1)}{1 + \sqrt{2/V_0}K + K^2/V_0}$	$\frac{1 - \sqrt{2/V_0}K + K^2/V_0}{1 + \sqrt{2/V_0}K + K^2/V_0}$

Table 5 Shelving filter design's equations.

For peak filters, exists Q factor, which is bandwidth to frequency cut-off ratio, $Q = \frac{f_b}{f_c}$

peak (boost $V_0 = 10^{G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1 + \frac{V_0}{Q}K + K^2}{1 + \frac{1}{Q}K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{1}{Q}K + K^2}$	$\frac{1 - \frac{V_0}{Q}K + K^2}{1 + \frac{1}{Q}K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{1}{Q}K + K^2}$	$\frac{1 - \frac{1}{Q}K + K^2}{1 + \frac{1}{Q}K + K^2}$
peak (cut $V_0 = 10^{-G/20}$)				
b_0	b_1	b_2	a_1	a_2
$\frac{1 + \frac{1}{Q}K + K^2}{1 + \frac{V_0}{Q}K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{V_0}{Q}K + K^2}$	$\frac{1 - \frac{1}{Q}K + K^2}{1 + \frac{V_0}{Q}K + K^2}$	$\frac{2(K^2 - 1)}{1 + \frac{V_0}{Q}K + K^2}$	$\frac{1 - \frac{V_0}{Q}K + K^2}{1 + \frac{V_0}{Q}K + K^2}$

Table 6 Peak filter design's equations.

The series connection of these filters can be done very easily, by putting output sample of one stage to input next stage. It's good to take into consideration that every filter needs store in memory its own four (or two) temporary values. For projects which has slow execution of mathematical computation, and does not possess enough memory, it is possible to use "filter selector", which is only one filter with the same equations and coefficients.

Equalizer, which was implemented, uses equation (4) presented in this sub-chapter. It was composed of two shelving filters (one low-frequency and one high-frequency) and three peak filters. Cut-off frequencies for them will be in succession: (200, 400, 800, 1600, 3200 Hz). Q factor for all peak filters will equal one. Equalization effect required twenty five multiplications, twenty additions and twenty four samples were stored. Internal for loop is responsible for equalization effect, while external loop process all stored samples.

In Figure 397 (in blue color) audio signal which was recorded for testing algorithms (this signal will be using for next algorithms) is presented. The stream was rescaled to the scale of [-32768, 32768] bits, centered at 0, which will suit to conditions which take place in microcontroller after Analog-to-digital conversion.

In Figure 37 results of working equalizer for different gain values is presented.

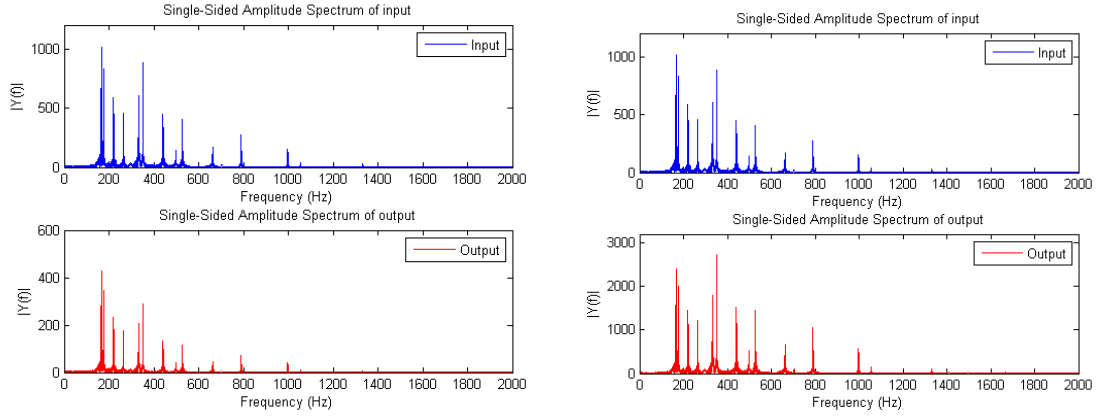


Figure 37 Amplitude spectrum for -6dB and 6dB.

IV.2. Delay

Main idea of this effect is to hold samples of input signal, store it in memory and produce at the output mix of input and delayed sample. All of these procedures must be done in one cycle. More developed algorithm store output signal. Figure 38 shows delay effect block diagram, with additional feedback gain [Tex 4]. Both this and forward gain are in zero to one range. If gain value is more than one, it will be unreal situation that previous signal is stronger than present. Forward loop ensures that to memory will be send output signal. Usually feedback and forward gain are equals, which give “clear” effect. D in this schematic signify, which previous sample will be taken to adder.

Delay effect algorithm can be described by equations (8) and (9).

$$w(n) = x(n) + f_g * w(n - D) \quad (8)$$

$$y(n) = x(n) + g * w(n - D) \quad (9)$$

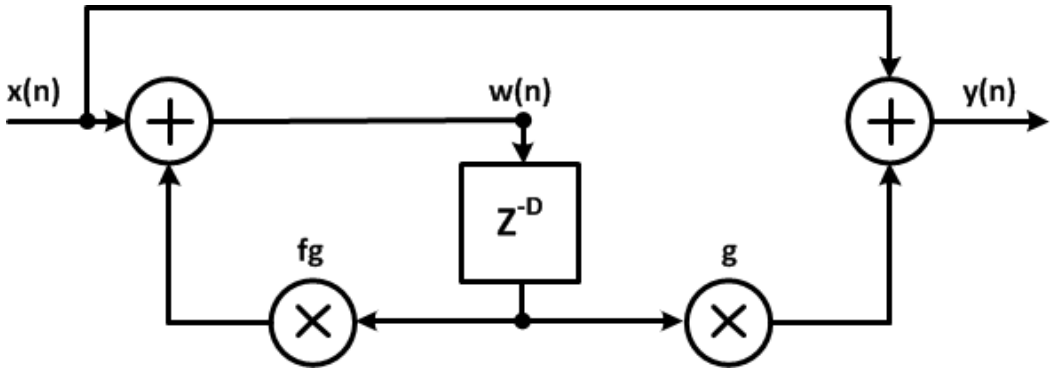


Figure 38 Delay effect algorithm block diagram.

Computing each y(n) requires two multiplication and two additions. Parameter D also defines how much space in memory is necessary to allocate.

Because output samples, which include information about previous samples, are stored, it is really easy to obtain result, which can be audible. By setting up gains parameters at high admissible values, and allocate more memory for sample, it is possible to get long audible delay effect. This time depend also on sampling frequency. If it is low, effect give bigger time. Sampling value has an effect on quality of signal.

To this algorithm must be add initial condition concerning first iterations. At the beginning there are no previous samples, so it would be inappropriate adding random values from memory. There are four solutions to prevent situation like this. Firstly, fill all memory with constant value - zero. Secondly, transmit input signal to output, concurrently save samples. When conditions will be sufficient to expected work, switch on whole algorithm. Thirdly, also transmit input signal to output, but simultaneously save samples with feedback gain. The fourth solution is more complex, because consists of many steps. With first sample behave like in second and third solution. With second one mix input and previous stored sample, not forgetting hold this value in memory. For next steps, procedure is the same, but from store is taken the oldest sample, till the moment that conditions will ensure correct working.

Algorithm of delay, add present to previous value multiplied by gain. In worst case we can make additions of two highest ones. To prevent this situation input signal must be divided. Divisor strictly depends on gain value, but it can possible to accept making signal two times lower. This procedure is negligible when, at input, occurs signal with small amplitude.

Figure 39 presents result of working delay effect. Values, used in this example, are: fg=0.8, g=0.8 and D= 2500.

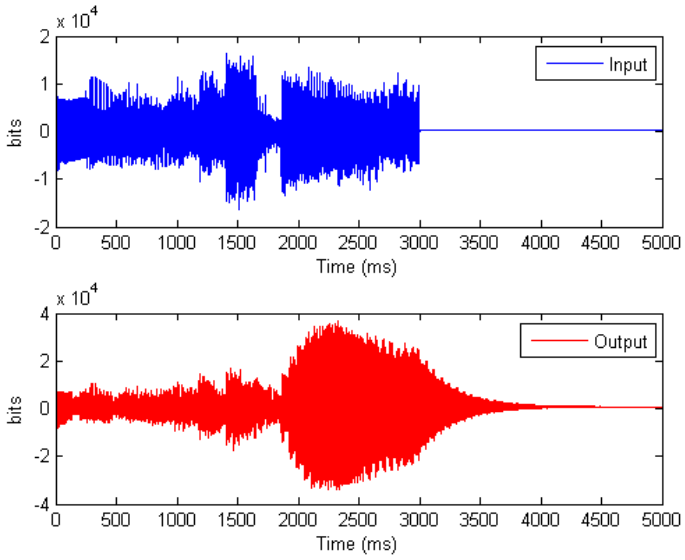


Figure 39 Result of delay algorithm.

IV.3. Chorus

Algorithm of this effect bases on the same idea as delay. Difference between them, comes from work principle. In delay, there was possibility to store samples from output or input. In chorus stored samples can only derive from input. This situation imitates delays between playing musicians. Figure 40 presents block schematic of implemented effect. As it can be seen, chorus returns reality of more than two musicians. Gains can be from zero to one range.

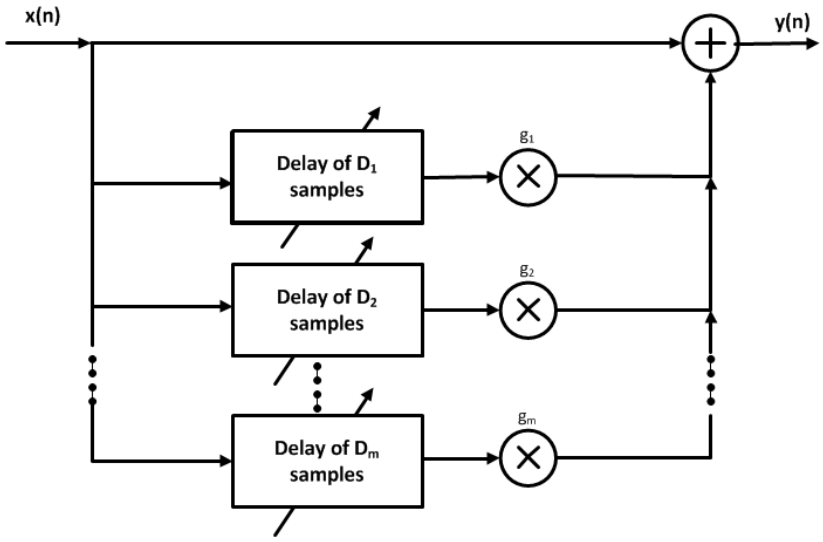


Figure 40 Chorus effect algorithm block diagram.

Above algorithm can be described by equations (10) and (11).

$$w(n) = x(n) \tag{10}$$

$$y(n) = x(n) + g_1 * w(n - D_1) + g_2 * w(n - D_2) + \cdots + g_m * w(n - D_m) \tag{11}$$

Computing each y(n) requires m multiplication and m additions, where m is a number of additional musicians, whose playing the same melody like main person. Parameter D defines how much space in memory is necessary to allocate. Samples are taken from the same buffer. Initial condition and dividing input signal is the same like in delay effect.

For making effect more real, delay values D_m of individual musicians lines, must change periodically. Typically modulation can be sine, square, triangle or sawtooth. One parameter call modulation frequency also must be taken into consideration. It determinates change velocity of delay value D_m. Typical value of it is between 0.1 – 1 Hz [Udo 71].

As it was written in sub-section 2.5 “Audio effects” value of delay length is typically 20 to 30 milliseconds. For execute delay like that, is necessary calculate maximum value of D parameter. For sampling frequency f_S=40 kHz, storing sample take places on every twenty five microseconds. For thirty millisecond delay, must be reserved space for 1200 samples.

Figure 41 presents result of working chorus effect for two musicians. Values, used in this example, are: g=0.2 and D= 1200. This algorithm used triangle modulation with frequency 0.83 Hz.

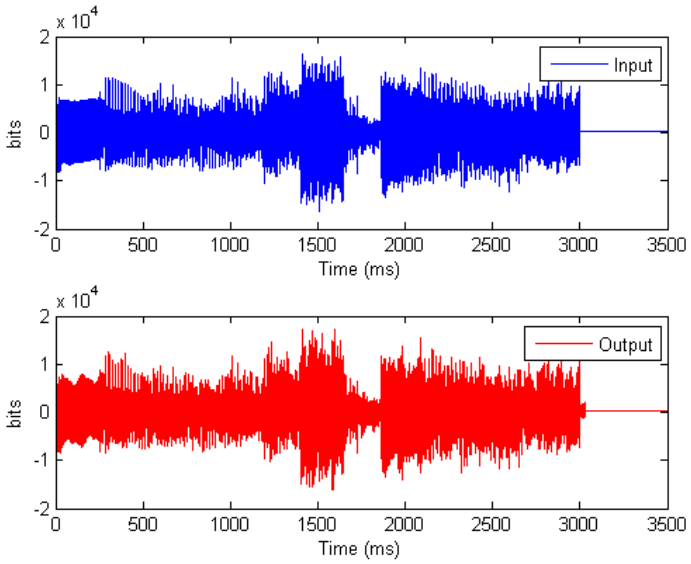


Figure 41 Result of chorus algorithm.

IV.4. Distortion

In Figure 24 is presented input and output signals after distortion and overdrive effects. Hard clipping is more easily to implement than soft clipping. First one only needs two conditions. If value of input sample is higher than distortion top level, at output appears value of top level. Analogically if signal is lower than distortion bottom level, at output occurs value of bottom level. In last case (when signal is between top and bottom level) input sample is transmitted to output without changes.

The second stage of distortion (overdrive) effect is multiplication output signal to full maximum range. Without this stage, sound becomes quieter. At the bottom of Figure 42, deformed shape of signals is shown.

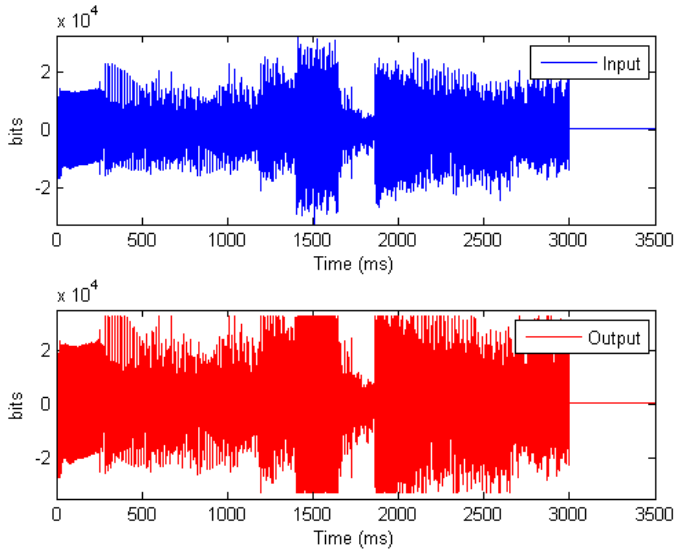


Figure 42 Result of distortion effect.

Chapter V:
Software Design

This chapter presents the flowchart of the developed software. The developed codes are placed in a CD accompanying the report. An organization of this section is divided to two parts. First part applies to PIC32 Module firmware, while second one concerns about describing window application, which is used to control effects' parameters.

V.1. PIC32 Module Firmware

PIC32 Module firmware was written in C programming language using software environment, which is adapted to working with Microchip microcontrollers, called MPLAB. With the aid of this environment, and PICkit3 programmer exists possibility of debugging and using additional tools like: Logic Analyzer, Simulink or Segmented Display Designer, which are helpful during writing the code.

Flowchart (shown in Figure 43) represents working process of the whole program, shows generic processing steps and connection between them. At the beginning of all algorithms, become four initializing subprograms.

First subprogram sets diodes pins as digital outputs, and buttons pins as digital inputs. These buttons are connected to external interrupts capture, so second function of this subprogram is to configure and enable interrupts, which are high-to-low edge sensitive.

After this block program is initializing SPI channels, which are responsible for acquiring data from ADC, and sending data to DAC. Also pins, answerable for control modes of working, are set like as output pins. The last function of this block is inscribing appropriate values to register SPIxCON. Both SPIs work in master configuration, in 16bit mode.

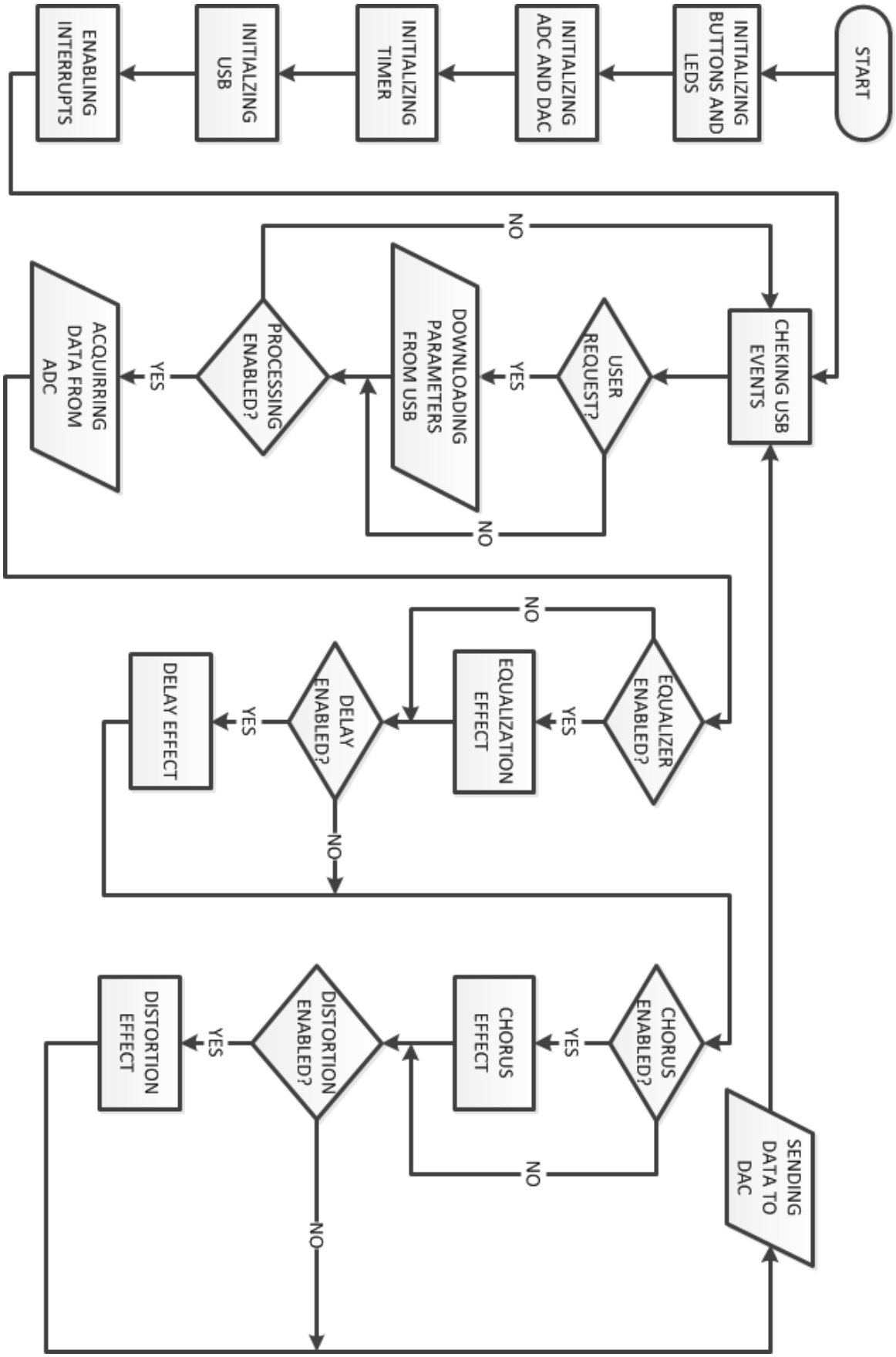


Figure 43 PIC32 Module firmware flowchart.

ADCs' SPIxBRG register is set to work with 8MHz frequency and DACs' works with 10MHz clock. Equation (12) defines the SCKx clock frequency as a function of SPIxBRG settings

$$F_{SCK} = \frac{F_{PB}}{2*(SPIxBRG+1)} \quad (12)$$

Third initializing block switches 16bit timer up. To ensure 40 kHz to PR3 the value 1999 was written (it was calculated from the equation (13)). After every end of timer counting, every 25µs is throwing interrupt signaled processing enabled.

$$Timer\ Period = [(PR+1) * T_{PB} * (Timer\ Prescale\ Value)] \quad (13)$$

Fourth block is an interface, which initializes the variables of the USB host stack. This block contains a software code, which was taken from sources proposed by Microchip.

After these four subprograms, instruction INTEnableSystemMultiVectoredInt occurs. It activates interrupts handling mechanism. After switching on the program, first five blocks occur only one time. It is worth to add, that before these subprograms are two loops responsible for clearing previous allocated memories for delay and chorus effects.

Next blocks concern about main tasks of Audio Effects Unit, which are placed in endless main processing loop. In this loop, function called USBTasks must be placed. This function executes the tasks for USB host operation. It must be executed on a regular basis to keep everything functioning. When an USB request from user comes to microcontroller, this function handles them appropriately. After this subprogram, request from user is being executed. It means that from PC computer are downloaded parameters, which are used in effects algorithms.

When downloading is finished, program goes to block connected with ADC (of course if processing is enabled). To acquiring data third SPI channel is used. The writeSPIx() is a truly bidirectional transfer function to service SPI. Function writeSPI3, which is connected to ADC, must firstly inscribe dummy word (consisting of two bytes) to transfer buffer. Next waiting for the end of transmission takes place. After completed transfer, value from receive buffer must be read. This function returns digital representation of measured sample.

Next eight blocks are connected with audio effects. Four of them are condition blocks checking if concrete effect is enabled. Changing accessibility of each effect takes place in corresponding interrupt handler function. This function also sets voltage value on pins responsible for Led shinning. Functioning of each audio effect block was presented in Section IV. Succession of their appearance is: equalizer, delay, chorus and distortion. Each algorithm uses parameters which were downloaded from PC. If there was not downloading process, algorithm will use primary values written in memory. All of effects can function in one cycle.

PIC32 Module firmware avoids doing mathematical operation on real numbers represented by float or double. Operating on them requires much time. Idea of audio effects algorithms is the same, but all decimal numbers were changed to integers. Even if algorithms need one or more additional operations required to conversion numbers, the process time is seven times smaller than in case of using real numbers. At the end of every effect block value of processed sample is converted to number in range from –32768 to 32768, to prepare sample for last stage presented in flowchart.

At the end of main loop function sending processed sample to DAC appears. It uses second SPI channel, and adequate function writeSPI2. Instead of writing to transfer buffer dummy word, is writing digital representation of output sample. This function returns received buffer, but it is not used in code. It is worth to add that value going to D/A converter must be unsigned, represented on 16bit, integer. Before that block mathematical operation takes place, which scales number to range from 0 to 2^{16}, which is 65536.

V.2. PC Software for AEU Effects Generation

PC Software was written in C programming language using software environment, called Borland C++ Builder 6.0. This environment was chosen to avoid problems with adding library and DDL file to project, which come from Microchip sources. They were written and compiled also in the same version of Borland. Software for PC is a window application served to control parameters of audio effects. For building this application, GUI creator was used. With this tool, there was no need to spend time writing code responsible for graphical appearance of program.

Window application consists of five overlaps and main menu. On first page block diagram of audio effects succession can be found. Below every effect are labels with name of parameter and value which will be sent to microcontroller. It is created to re-check set parameters and it is useful when all effects are used. In Figure 44 shows appearance of this page. As it can be seen in Main Menu are two positions. First is File and after open it, we can stop program. Second is Help, and in its can be found information about program (version, programming environment and purpose). Also there are two buttons: UPLOAD and EXIT. The second one is for quitting from application. First one is used to upload parameters to microcontroller. After clicking it, its function collects all effects parameters and writes them in a send buffer. Situation like that takes place with parameters from delay, chorus and distortion, whereas with equalizer values, is different. To microcontroller are sent filters coefficients, and it is good when they are accurate to four numbers after coma. In one byte can be send value from 0 to 255, so coefficient must be divided to two bytes and in PIC32Module

firmware, this two bytes are integrated. For five filters (each one requires five coefficients), fifty bytes must be sent.

In second window, presented in Figure 45 (on the left side), is picture showing serial connection between filters and cut off frequencies of each filter. Users do not worry about calculating filters coefficients, because they are computing in program on the basis of gains, which are set. For setting gain parameters serve five sliders (one for one frequency). Their values are given in decibels, and they can change from minus six to six.

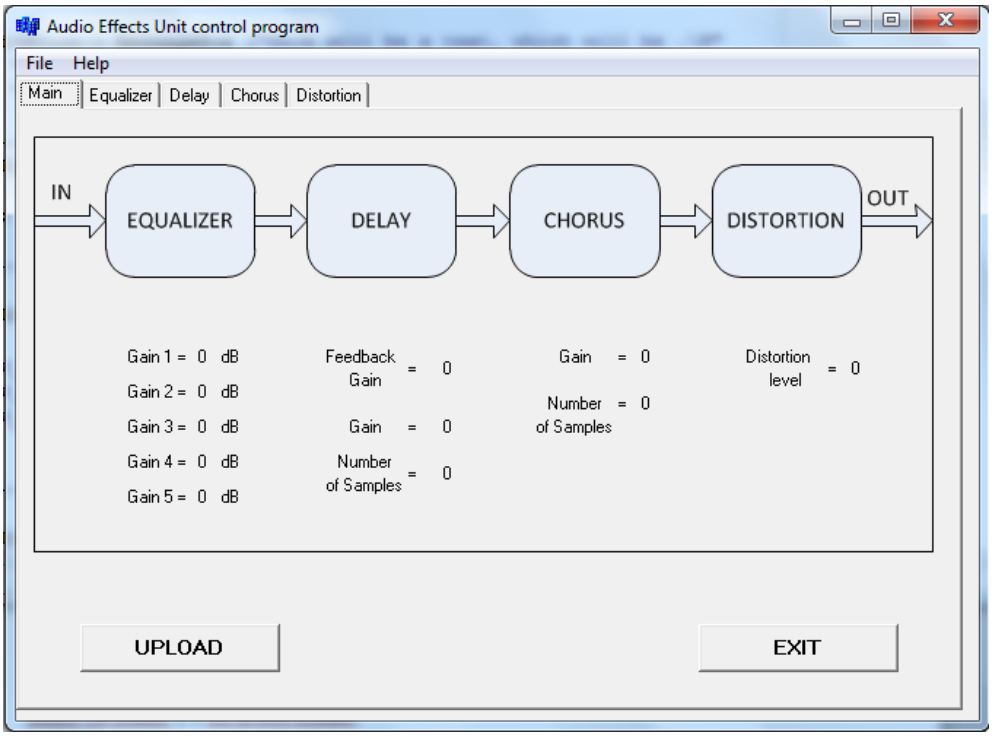


Figure 44 Main Window.

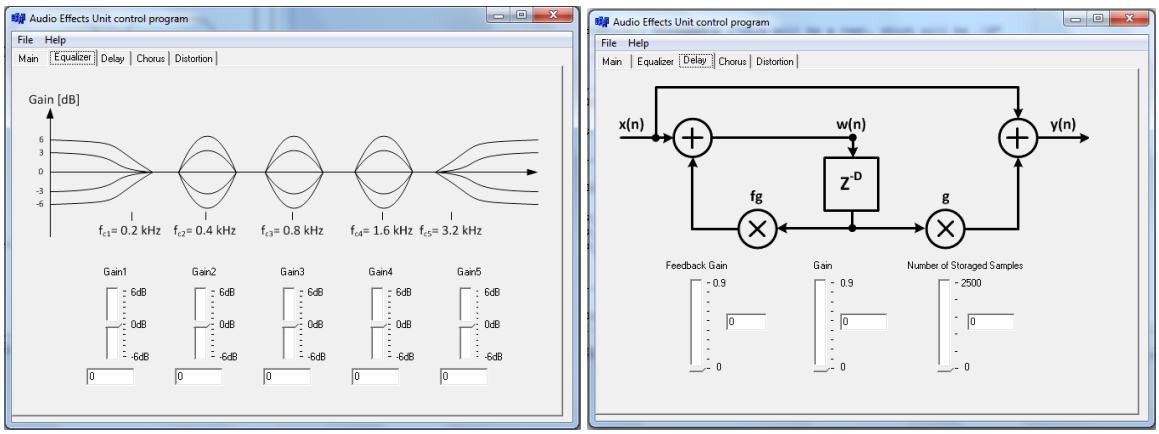


Figure 45 Equalizer Window and Delay Window.

Third control page, shown in Figure 45 (on the right side), is used for setting parameters for delay effect. Also there is presented an algorithm of this effect, which gives look of its functioning. Values of first two parameters can change from 0 to 0.9, and value, which is responsible for number of delay samples, is in range to 2500. As it was said, microcontroller is operating on integer values so feedback gain and gain are multiplied by ten, and third parameter is divided by ten, to be sending in one byte. In PIC32 Module firmware this last value is multiplied by ten.

Fourth window presented in Figure 46 (on the left side), gives opportunity to control parameters connected with chorus effect. This control page looks really similar to delay one. It includes algorithm, two sliders, and multiplying and dividing parameters is exactly in the same way like in previous effect. Difference between them is only one that this window have not got feedback gain track bar, but it comes from principle of working this algorithm.

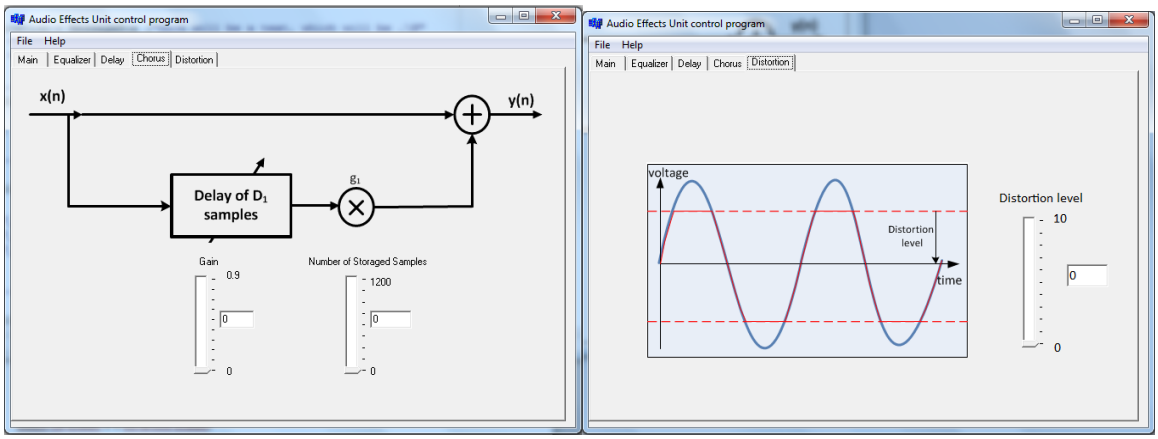
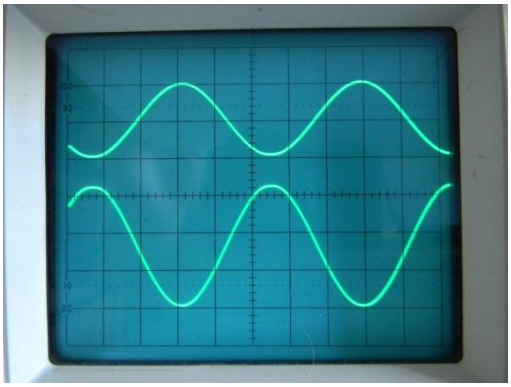


Figure 46 Chorus Window and Distortion Window.

Last window applies to distortion effect. It is presented in Figure 46 (on the right side), which consists of: picture presenting dependence between distortion level and output signal, and slider to setting value of parameter. Its range can change from zero (inaudible changes in sound) to ten (effect is very good audible, but sound is degenerated).

Chapter VI:
Audio Effects Unit testing

This section presents results of working AEU device, and it consists of pictures, taken form oscilloscope, proving correct hardware and firmware performance. First tests were done on generated signal, which had 2 Volts peak to peak amplitude. This value (2 V) is expected on real audible signals, which come from audio devices. In Figure 47 top waveform is input signal (for all hardware tests it is placed on the same positions), and bottom one represent outcome after first operational amplifier stage. Amplified signal has required 3.3 Volts amplitude. Next aspect that proves correct work of amplifier stage is that output is in an opposite phase, because operational amplifier works in inverting configuration.



PROBE x	10
TIM/DIV	0.2 ms
VOLTS/DIV A	0.1V
VOLTS/DIV B	0.1V

Figure 47 Results after operational amplifier stage with 1 kHz signal.

Second stage of input signal conditioning block is active filter, with 20 kHz cut off frequency. Result of working this stage is shown in Figure 48. Three decibel reduction is followed at 20 kHz. This frequency characteristic was made on the basis of pictures taken from oscilloscope, which can be found in appendixes on CD.

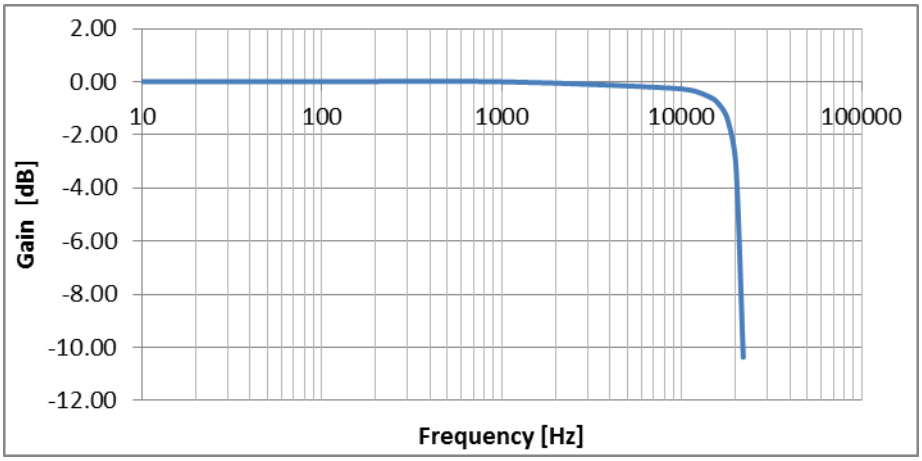


Figure 48 Frequency characteristic of input active filter.

After this block signal is converted by analog to digital converter, and goes to microcontroller. Received sample (without any changes) is transmitted from digital to analog converter. After conversion signal is represented by stepped waveform and gets to output active filter. Its frequency characteristic is presented in Figure 49, which was created in the same way like Figure 48.

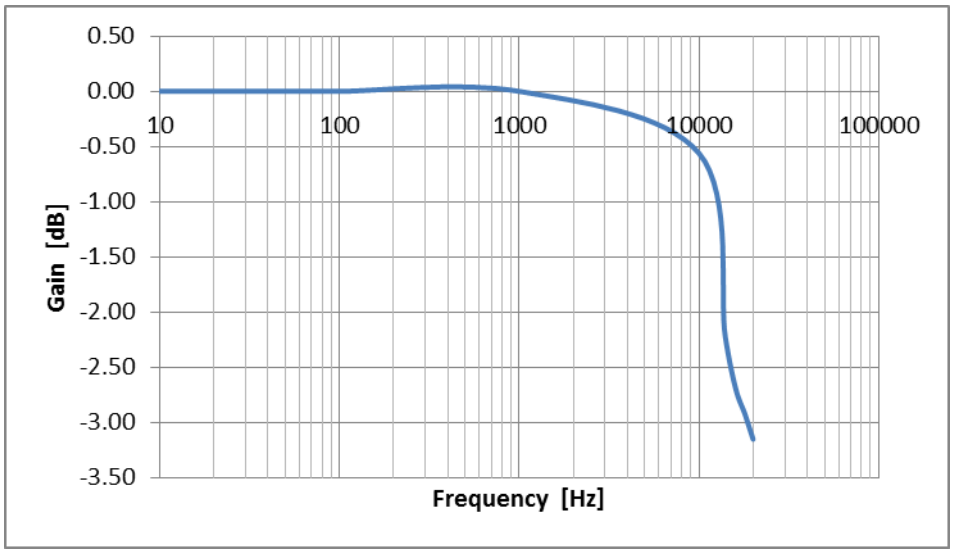


Figure 49 Frequency characteristic of output active filter.

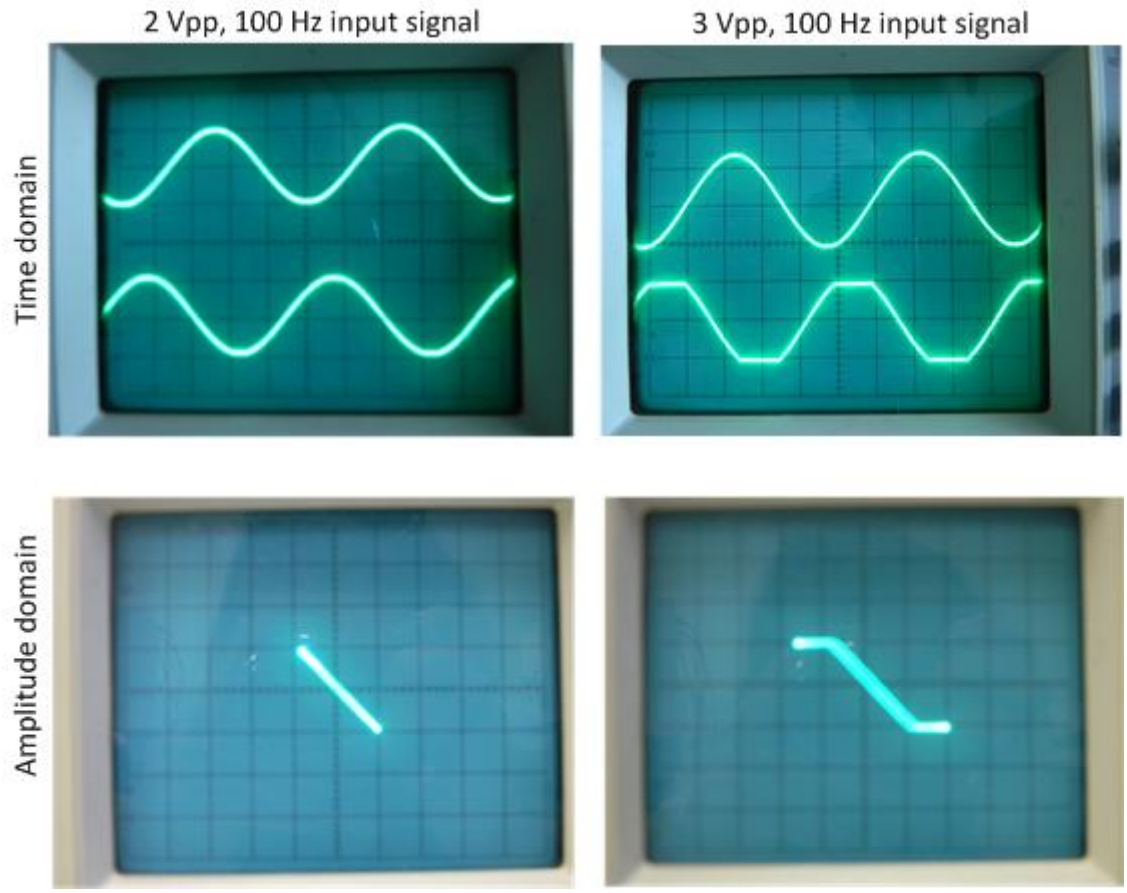


Figure 50 Input vs. output in time and amplitude domains.

In Figure 50 results for two different input voltages are presented. With 2 Volts on input, circuit works correctly and output signal is not distorted. Form X-Y characteristic it can be seen that output signal is sifted in phase. Reason of these different phases is that first operational amplifier works in inverting configuration. Also time required to process all algorithms has influence on phase of second signal. On the right side is situation when input signal has higher voltage value, and output signal is distorted. Potentiometers in both operational amplifiers are set to amplify expected audio signal level, which equals 2 Volts.

Figure 51 presents result of working of implemented equalizer. For better demonstration, it consists of two frequency characteristics. Cut-off frequencies of filters included in equalizer, are 200, 400, 800, 1600 and 3200Hz. For peak filters, Q factor equals 1. Gain of each filter can change in -6 to 6 dB range. On top of Figure 51, is situation when one of filter is set to 6dB and others equal zero. At the bottom is the same situation but one filter is set to -6dB.

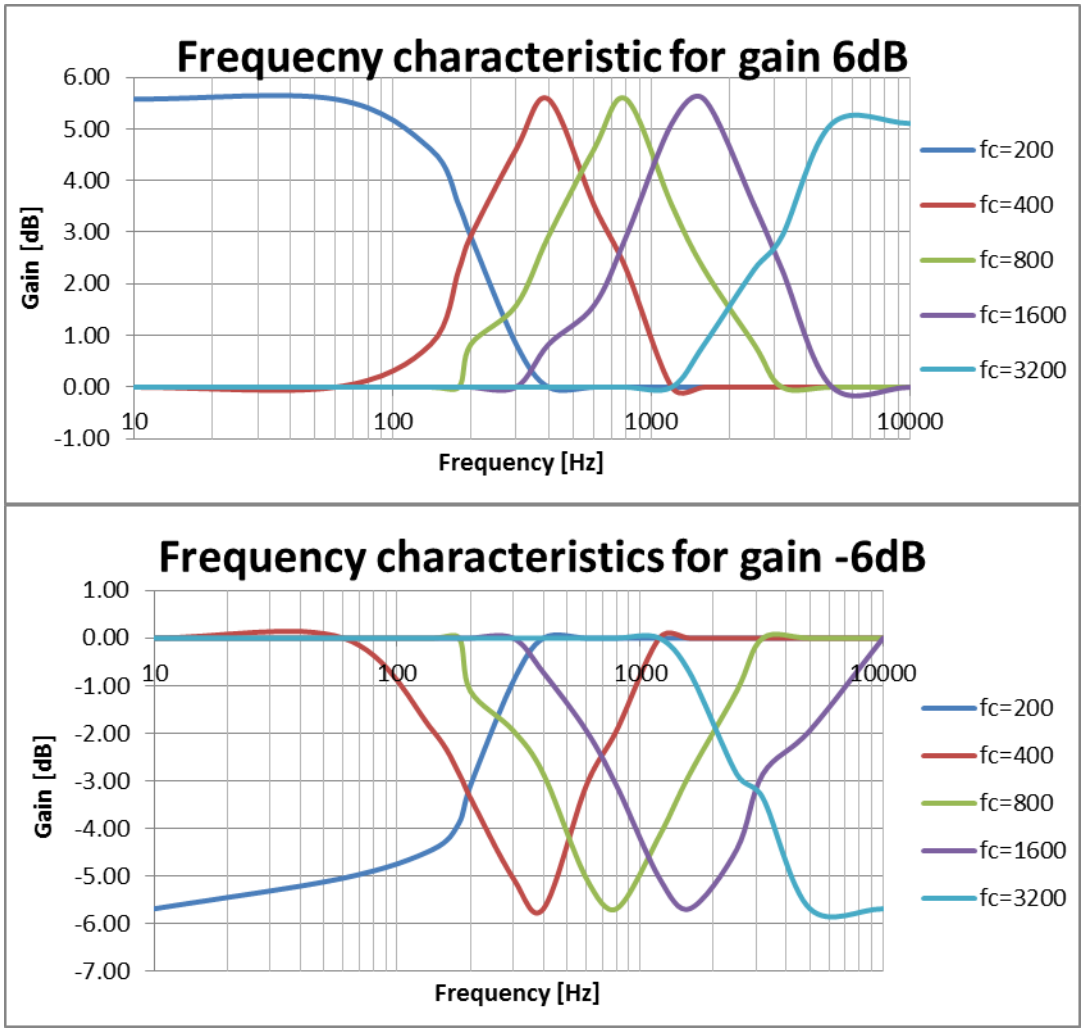


Figure 51 Equalizer frequency characteristics.

It is seen that signal, in both cases, does not achieve exactly expected values 6 or -6 dB. Maximum gain, when device is working, is in range from -5.68 to 5.57 dB. Difference between expected and real value can come from rounded filters coefficients and measurement inaccuracy. Results don't deviate much from expected values, so it can be admitted that equalizer is working correctly.

Other three effects cannot be presented in graphical way. When delay effect is working, in speakers two-second delay is audible. When chorus effect is switched on, there is an impression that two musicians are playing. When device is using distortion effect, cutting signal is heard.

In all audio devices is important to demonstrate that the AEU meets high quality audio processing. This consists in measuring the amount of distortion that the AEU hardware itself adds to the signal that passes through it. Total Harmonic Distortion (THD) parameter is a standard way to evaluate the amount of distortion that is added by the channel where the signal flows.

Microcontroller sends to the DAC, the sample that was received from the ADC, without adding any effects. AEU input is connected to the sound-card output (headphones) of one computer, and AEU output is connected to the sound-card input (microphone) of another computer. First computer generates a sine-wave signal with 500 Hz frequency. Second computer records the signal from the AEU output, in MATLAB and calculates FFT of the signal.

THD is a ratio between a sum the power that signal has in its harmonics and power of the fundamental component. Equation nr (14) describes above dependence. Equation nr (15) is also commonly used and from this equation was calculated THD ratio of AEU.

$$THD = \frac{\sum_{i=2}^{\infty} P_n}{P_1} \tag{14}$$

$$THD = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + V_5^2 + \cdots}}{V_1^2} \tag{15}$$

Result of this test was THD ration equals 0.17%, which in decibel range is -56dB. These values classify AEU as high quality audio device, but not well as in Hi-fi technology.

Chapter VII:
Conclusions and Future Work

Upon the conclusion of the work that served as the basis for the current text, several conclusions need to be registered. Starting with the research made on Audio Effects Systems, it appears that these systems are widespread, and at every time will be demand on consumer market for device like that. This work gives good results and possibility to circuit development. However, there are some things that need to be more thoroughly analyzed.

Digital processing of audio signal gives an opportunity to implement effects algorithms. These, which required storing previous sample, can be easily controlled by changing size of allocated memory. Nevertheless microcontroller memory is limited, so it is suggested to add additional external one. It will increase number of possible algorithms based on delays working at one process cycle.

Digital filters, which are used to create equalizer, required many multiplications and additions. For creating expanded equalization effect it is necessary to use a microprocessor working with higher frequency. AEU shows that is possible to create device which includes described effects, and they are working in parallel. However, if quality which occurs in recording studios is needed, signal microprocessor should be used.

Audio Effects Unit includes USB connection which is very important for controlling effects parameters from the computer. USB connection is only one of many advantages which make it possible for commercial consumers to use this device. Such simple connection is very important for typical user, and it is taken into consideration at first. Another advantage is that setting of parameters is very easy; it can be done by changing their value in PC Software. It is also small, what can be very helpful, when the device is used at home. What is more, musical instrument can be connected to the device, without using preamplifiers or other circuits, which usually occur in such situations.

Sampling frequency and resolution bit are comparable to such which are used in CD Audio. Good quality is the result of the fact that the device does not introduce distortions. This is confirmed by the fact that THD rating is at the level of 0,1% what is considered to be in high-fidelity and inaudible to the average human ear.

In spite of advantages for average users, Audio Effects Unit has also pluses for programmers. The most important one is that it can be reprogrammed, what means that the new audio effects can be implemented. Features of this device allow implementing almost every algorithm, which make this circuit widespread.

References

[Kahr 4] Mark Kahrs, Karlheinz Brandenburg, *"Applications of Digital Signal Processing to Audio and Acoustic"*, Kluwer Academic Publishers, 2002, page 4.

[Body 285] Body, Human." *The New Book of Knowledge"*, New York Grolier, 1967, page 285.

[Proj] http://projewski.wordpress.com/2008/05/15/lekcja-2-struny-gitary/#Czestotliwosc

[Phil 4, 5, 10] Phil Lapsley, Jeff Bier, Amit Shoham, *"DSP Processor Fundamentals*", The Institute of Electrical and Electronics Engineers, New York, page 4,5,10.

[Gunt 7]Gunther Gridling, Bettina Weiss,*"Introduction to Microcontroller"*, Vienna University of Technology 2006, page7.

[Tex 3, 4] Texas Instruments *"How to Create Delay-based Audio Effects on the TMS320C672x DSP"*, 2005, page 3, 4. Available: focus.ti.com/elit/an/spraaa5/spraaa5.pdf

[Wal 4, 10, 54, 56 63, 71, 111, 133] Walt Kestr, *"Analog-digital conversion"*, Analog Device Incorporation Central Applications Department, 2004, page 3.4, 3.10, 3.54, 3.56, 3.63, 3.69, 3.71, 3.111, 3.133.

[I2S 1,2] *I^2S bus specification*, Philips Semiconductors, 1986, page 1, 2. Available: www.nxp.com/acrobat_download2/various/I2SBUS.pdf

[I2C 7, 8, 10] *The I2C-bus specification*, Philips Semiconductors, 2000, page 8. Available: www.nxp.com/documents/other/39340011.pdf

[Ser 3, 11] *Serial Peripheral Interface & Inter-IC (SPI_I2C)*, Renesas Electronics Corporation, 2003, page 3. Available: http://documentation.renesas.com/eng/products/region/rtas/mpumcu/apn/spi_i2c.pdf

[Udo 34, 51, 53, 55]Udo Zolzer, Editor,*" DAFX - Digital Audio Effects",* John Wiley & Sons, Ltd, 2002, page 34, 51, 53, 55

[Pic 23*] PIC32 Family Reference Manual, Sect. 23 Serial Peripheral Interface*, page 3. Available: ww1.microchip.com/downloads/en/DeviceDoc/DS-61106F.pdf

[Pic 27] *PIC32 Family Reference Manual, Sect. 27 USB On-The-Go,* page 47. Available: ww1.microchip.com/downloads/en/DeviceDoc/61126F.pdf

[LTC 1569] - *LTC1569-7 Datasheet*. 1998, Linear Technology Corporation, page 1.

[AD 8531] -*AD8531 Datasheet*. 2008, Analog Devices, page 1.

[LTC 1864] - *LTC1864L Datasheet*. 2001, Linear Technology Corporation, page 1-3.

[LTC 2641] - *LTC2641 Datasheet*. 2007, Linear Technology Corporation, page 1-2.

[TSR 1] - *TSR-1 Datasheet*. 2009, Traco Power, page 1-3.