TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

PEKKA HARTIKAINEN

DESIGN OF AN ELECTRONIC DRUM MODULE WITH AN AUDIO PLAYBACK FROM AN SD MEMORY CARD

Master's thesis

Examiner: professor Karri Palovuori
Examiner and topic approved in the Faculty of Computing and Electrical Engineering Council meeting on November 07, 2012

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Sähkötekniikan koulutusohjelma
HARTIKAINEN, PEKKA: Design of an Electric Drum Module with an Audio Playback from an SD Memory Card
Diplomityö, 46 sivua
Marraskuu 2012
Pääaine: Elektroniikan tuotesuunnittelu
Tarkastaja: professori Karri Palovuori
Avainsanat: Sähkörummut, SD muistikortti, tuotesuunnittelu, rumputriggeri, pietsosähköinen sensori.

Sähkörumpusovelluksissa käytetyt ääninäytteet ovat yleensä tallennettuina sähkörumpumoduulin sisäiseen muistiin. Olisi hyödyllistä jos sähkörumpumoduuleissa olisi mahdollisuus käyttää ulkoista laajennettavaa muistia ääninäytteille. Tässä työssä tutkitaan SD muistikortin soveltuvuutta ääninäytteiden tallennukseen sähkörumpusovelluksessa. Tätä tarkoitusta varten tässä työssä suunnitellaan yksinkertainen sähkörumpumoduuli SD muistikorttipaikalla, ja suunnitellun laitteen suorituskyky testataan.

Työ jakaantuu neljään osaan: Kirjallisuustutkimusosassa selvitetään sähkörumpusovelluksiin sekä työssä käytettyihin tekniikoihin liittyvät taustatiedot. Suunnitteluosassa sähkörumpumoduulin suunnittelu kuvataan yksityiskohtaisesti. Suunnitteluprosessi kuvataan piirikaavion, ohjelmiston sekä mekaanisten rakenteiden osalta erikseen. Prototyypin valmistus -osassa kuvataan käytetyt valmistusmenetelmät ja valmistusvaiheessa laitteeseen tehdyt muutokset. Tulokset-osassa testausmenetelmät ja tulokset selitetään erikseen sähköisille ominaisuuksille, moduulin reaaliaikaiselle suorituskyvylle sekä äänenlaadulle.

Tutkimus osoittaa, että SD muistikorttia voidaan käyttää ääninäytteiden tallennukseen sähkörumpusovelluksessa. Suurin saavutettu lukunopeus on 1.0MB/s SanDisk micro SD muistikortilla (väylänopeudella 7.5MHz, 4-bit siirtomoodissa). Mitä useampi ääninäyte pitää miksata yhteen, sitä enemmän aikaa kuluu miksaukseen. Jos vain yksi ääninäyte pitää toistaa, miksausta ei tarvitse tehdä ja korkeita näytteistystaajuuksia voidaan helposti käyttää. Ongelmia ilmenee kun monta ääninäytettä pitää miksata yhteen, jolloin prosessointiaika kasvaa huomattavasti. Kun neljä ääninäytettä miksataan yhteen samanaikaisesti, 23438Hz on suurin mahdollinen näytteistystaajuus jota voidaan käyttää ilman äänen vääristymistä tässä työssä suunnitellussa sähkörumpumoduulissa.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master’s Degree Programme in Electrical Engineering
HARTIKAINEN, PEKKA: Design of an Electric Drum Module with an Audio Playback from an SD Memory Card
Master of Science Thesis, 46 pages
November 2012
Major: Electronics Product Design
Examiner: Professor Karri Palovuori
Keywords: Electronic drums, SD memory card, product design, drum trigger pad, piezoelectric sensors.

Usually in electronic drum modules the used audio samples are stored in the internal memory. It would be beneficial to have external and expandable memory for the audio samples. The main focus of this thesis is to evaluate the suitability of an SD memory card for storing the audio samples in electronic drum applications. For this purpose a simple electronic drum module with SD memory card interface is designed and the performance of the device is tested.

The thesis is divided into four parts. In the background part, the theoretical aspects related to electronic drum applications and the techniques used in this thesis are explained. In the design part the design of the drum module is described in detail. Specifications are listed, and the design process is described separately for hardware, software and mechanical solutions. In the prototype manufacturing part the prototype manufacturing phase is described. Used manufacturing processes are explained, and problems and changes during the prototyping phase are examined. In the results, testing procedures and results are presented individually for hardware, real-time performance and audio quality.

Based on the results obtained in this thesis, SD memory card can be used as audio file storage in electronic drum applications. The highest obtained read speed is 1.0MB/s with SanDisk 2GB micro SD memory card (when bus speed is 7.5MHz with 4-bit transfer mode). The more audio files need simultaneous mixing, the more processing time is consumed. With just one file playing, no mixing is needed and high sampling frequencies can easily be used. Problems occur when many files need to be mixed simultaneously. With four simultaneously playing samples, 23438Hz is the highest possible sampling frequency that can be used without audio distortion in the electronic drum module constructed in this thesis.

PREFACE

This thesis was done as a part of Master's Degree in Electrical Engineering. The motive for this thesis comes from my own drumming background, and the interest for drum technology. The parts used in the construction of the electronic drum module were provided by The Department of Electronics and Communications Engineering in Tampere University of Technology. Most of the manufacturing process was also done in the same department.

I would like to sincerely thank Professor Karri Palovuori for guiding me in the process and leading me into the right direction. I would also like to thank my family and friends for providing support and listening my problems especially during the programming phase of the device. Finally, I would like to thank all my fellow musicians who have kept me inspired to keep on playing drums and rocking in the free world.

Tampere, May 8, 2013

Pekka Hartikainen
Email: pekkaharti@hotmail.com

CONTENTS

TERMS AND DEFINITIONS

ABDAC	Audio Bitstream Digital-to-Analog Converter. Converts audio sample values into digital bitstream.
ADC	Analog-to-Digital converter, converts analog voltages into digital values.
ASF	Atmel Software Framework.
CF memory card	Compact Flash memory card is a memory type with 16-bit parallel data transfer.
Cluster	Group of data blocks in FAT file system.
Contiguous allocation	File system where the data is stored as a contiguous sequence of disk blocks.
Drum trigger pad	Used in electronic drums to sense strokes.
Electronic drum module	An embedded device consisting of drum trigger inputs, a user interface and an audio output.
ESD	Electrostatic Discharge.
FAT	File Allocation Table. Contains the addresses of the data blocks in a FAT file system.
IC	Integrated circuit.
Inhibit	Specifies how much time has to pass between two subsequent drum trigger strokes of the same channel.
ISR	Interrupt Service Routine.
MCI	Multimedia Card Interface. Included in the AT32UC3A3256-microprocessor. Provides communication interface to be used with various multimedia cards, such as the SD memory card.
PCB	Printed Circuit Board.
PCM	Pulse Code Modulation. Uncompressed storage format to store the WAVE-files.
PDCA	Peripheral Direct Memory Access Controller. Transfers data between on-chip peripheral modules and on/off-chip memories without CPU intervention
Piezoelectric sensor	Structure that transforms mechanical stress into a measurable voltage.
SDHC	High Capacity SD memory card.
SD memory card	Secure Digital memory cards are NAND flash-based memories widely used in audio and video consumer electronic devices.
SDSC	Standard Capacity SD memory card.
SDXC	Extended Capacity SD memory card.
Slope	Speficies how fast the adaptive threshold decreases.
Speed class	Defines the minimum performance of an SD memory card.

SPI	Serial Peripheral Interphase. Provides serial data link between devices.
Threshold	Minimum value the measured drum trigger signal must have to cause audio sample to be played back.
USB	Universal Serial Bus. Communication protocol widely used in consumer electronics.
WAVE	Waveform Audio File Format. Used to store audio bitstream.

1 INTRODUCTION

Digital musical instruments are becoming more popular as manufacturers compete to make more and more realistic instruments. In electronic drum –applications, one of the most important aspects is the quality of audio samples used. Most of the affordable drum modules have built-in sounds without the option to change them. Some more expensive modules have the ability to load your own samples, such as the 2BOX DrumIt Five, but currently there is no electronic drum module supporting the affordable and commonly available SD (Secure Digital) memory card for the storage of the audio-samples.

In this thesis, an electronic drum module with micro SD memory card slot was designed. The suitability of SD memory card for audio storage in electronic drum applications was tested by copying audio files to the memory card and playing these back when a drum trigger pad associated to a certain audio file was stroked. Read speeds of three different micro SD memory cards were tested and these were compared to read speed of a flash IC-chip memory connected to Serial Peripheral Interphase (SPI). The ability to mix several audio files together in real time was also tested.

Chapter 2 focuses on the theoretical aspects related to electronic drum applications and the techniques used in this thesis. In chapter 3, the design of the drum module is described in detail. Specifications are listed, and the design process is described separately for hardware, software and mechanical solutions. Chapter 4 describes the prototype manufacturing phase. Used manufacturing processes are explained, and problems and changes during the prototyping phase are examined. In chapter 5, testing procedures and results are presented individually for hardware, real-time performance and audio quality.

2 BACKGROUND

Electronic drums are gaining popularity because they offer many benefits compared to traditional acoustic drums. For example, electronic drums produce less noise to the environment compared to acoustic drums. The volume levels of electronic drums can be adjusted and the samples used for each drum trigger pad can easily be changed. Tuning of acoustic drums is a time consuming task. With electronic drums this problem doesn't exist, the audio samples stay always in tune. In this chapter the principles behind electronic drums are explained. The used audio format and the operating principles of the memories tested in this thesis are also explained.

2.1 Electronic drums

Electronic drums transform digital signals into sounds, simulating acoustic drums. An electronic drum set consists of trigger pads which are connected into an electronic drum module via cables. When a drum trigger pad is hit, the drum module plays back a sound associated to that specific pad.

2.1.1 Electronic drum module

The electronic drum module is an embedded device consisting of drum trigger inputs, a user interface and an audio output. It is the most important part of the electronic drum set containing all the intelligence, which is why it is often called the "Drum brain". The audio samples used in an electronic drum module are usually built in, and they can be selected for each drum trigger pad separately. Often the drum modules have other features built-in, such as metronome and play-along songs.

Different properties of the most advanced electronic drum modules currently on the market are listed in Table 1. Most versatile drum module, when considering the ability to load your own samples, is the 2BOX DrumIt Five with its 4GB onboard memory. The most expensive device, the Roland TD-30, doesn't offer the possibility to upload your own samples, but the amount of samples and effects make it very versatile device. Even though the Alesis SamplePad is not an electronic drum module to be used with external trigger pads (it has four built-in pads), it was taken into this comparison because of the possibility to load your own samples via SD memory card. However this module doesn't support stereo-audio or the possibility to load several audio samples for the same channel (discussed in chapter 2.3).

Table 1 *List of properties of different electronic drum modules (TD-30, DTX950, DrumIt Five, DM10, SamplePad)*

Device	Memory capacity	Custom samples	Expandable memory	Notes	price range (20.3.2013)
Roland TD-30	N/A	No	No	Wide variety of sounds	2500e
Yamaha DTX900	N/A	Yes	512MB DIMM	Samples can be loaded via USB (from PC), or recorded via ‘Sampling In’ jack	1600e
2BOX DrumIt Five	4 GB onboard Flash	Yes	No	Samples can be loaded via USB (from PC)	1000e
Alesis DM10	128MB	Yes	No	Samples can be loaded via USB (from PC)	520e
Alesis SamplePad	14MB	Yes	SD, SDHC up to 32GB	Sample pad consisting of 4 trigger pads. Mono sounds.	150e

As can be seen from the table above, no high end drum modules offer the possibility to load the drum samples from an SD memory card (only the Alesis SamplePad has this option). Some of the drum modules have an USB (Universal Serial Bus) interface to PC for loading the samples, but no support for external storage. Using external storage would allow limitless expandability for the drum samples used in electronic drums, and a simple way to change the samples without the need for PC.

2.1.2 Drum trigger pads

The main component of a drum trigger pad is the piezoelectric sensor, which transforms mechanical force into a measurable electric charge. Piezoelectric sensors used in electronic drum triggers usually consist of two thin plates of a dielectric material arranged like a capacitor, resulting in a structure behaving like an electric charge generator. Electrodes are attached to the plates for signal-transfer. When mechanical stress is applied into the piezoelectric sensor, voltage difference can be measured between the electrodes. (Smith 2004)

In drum trigger pads, the piezoelectric sensors are built-in, as can be seen in Figure 1. Drum trigger pads can have several piezoelectric sensors built-in, for example one sensor for the drum head (the stroking surface attached to the drum shell) and one for the rim (which attaches the drum head to the shell). For simplification, the operation of a drum trigger pad with one piezoelectric sensor is explained.

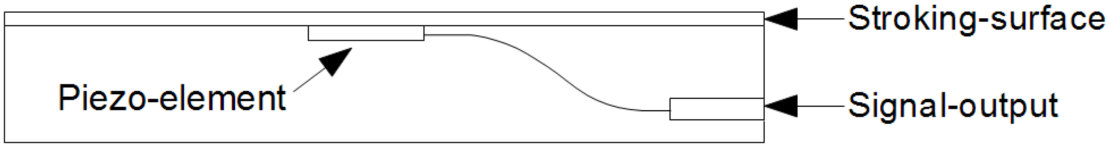

Figure 1 *Simplified illustration of a drum trigger pad with one built-in piezoelectric sensor.*

When the surface of a drum trigger pad is stroked with a drum stick, mechanical vibration is transferred into the piezo-element attached to the pad. The vibration causes a voltage change across the piezo-element. This voltage is transferred via cables to the output of the trigger pad and from there to the electronic drum module. The voltage formed in a piezoelectric sensor can have high amplitude and a lot of AC-components, so filtering has to be done before the signal can be read with an analog-to-digital -converter (ADC) and interpreted by a microcontroller. The piezo-signal filtering is described in chapter 3.1.2.

2.1.3 Real-time in electronic drums

For electronic instrument to sound realistic, it is important that the delay between stimulus (e.g. hit on a drum pad) and the resulting output (e.g. snare sound from a speaker) is minimized. If this delay is over 60ms (see chapter 5.2.1), the creation of the sound starts to have an unrealistic feel and playing of the instrument becomes challenging. This means that all the processing needed to start the wav-file playback has to be done in 60ms at most. To make this possible, the processor of the system should be as efficient as possible (high CPU-frequency, fast processor architecture).

The electronic drum module must also be able to recognize and play back subsequent strokes at a very fast pace. For example, playing 16th-notes at a 200bpm tempo means that there is 75ms between each stroke (for one beat there are four strokes if 16th-notes are played. So in one second 4*200bpm/60 = 13.3 notes are played, which means 1/13.3s = 75ms delay between each stroke). To recognize strokes at so fast pace, high ADC measurement frequency, fast processing of audio-samples and fast algorithms for peak detection are required.

2.2 Memory

Two types of memories were tested in this thesis: micro SD memory card and a flash integrated circuit (IC) -chip communicating through SPI-interface. Changeable micro SD memory card gives limitless expandability for the sounds used in electronic drums, which makes it an interesting alternative to internal memory of predetermined size. The serial flash was chosen for comparison.

2.2.1 SD memory card

In this project, NAND flash -based micro SD memory card was used to store the audio-sample files. According to SD Card Association, it is “a memory card that is specifically designed to meet the security, capacity, performance and environment requirements inherent in newly emerging audio and video consumer electronic devices” (SD Specifications Part I 2013). SD memory card is an interesting choice for electronic drum applications because it offers inexpensive expandability and easy-to-use interface to PC for transferring the audio files.

It is possible to communicate with SD memory cards via 1-bit protocol, via SPI mode of the SD memory card protocol, or via 4-bit protocol. Some microcontrollers offer a specific bus for the 1- and 4-bit protocols, such as the Multimedia Card Interface (MCI) of the AT32UC3A3256-microcontroller (AT32UC3A3/A4 Datasheet) used in this project. The SD communication is based on a 9-pin interface (including clock, command, four data lines and three power lines).

Currently there are three types of SD memory cards divided based on their capacities:

- Standard Capacity SD Memory Cards (SDSC) which can store up to and including 2GB
- High Capacity SD Memory Cards (SDHC) which can store more than 2GB and up to and including 32GB
- Extended Capacity SD Memory Cards (SDXC) which can store more than 32GB and up to and including 2TB.

The achievable bus speeds using 4-bit data transfer depend on the bus speed mode used, the host clock specifications and the PCB (Printed Circuit Board) -layout of the application. In default speed mode, frequencies up to 25MHz are possible (12.5MB/s, 3.3V signaling). In high speed mode, frequencies up to 50MHz are possible (25MB/s, 3.3V signaling). There are also Ultra High Speed Phase II Cards (UHS-II) supporting transfer speeds up to 195MB/s using specific signaling and lower operation voltages. (SD Specifications Part I 2013)

The speed class of a SD memory card defines the minimum performance of the card (the minimum transfer speed). Table 2 lists the minimum performance ratings for SD memory cards according to the speed class.

Table 2 *Minimum performance of the SD memory cards according to the speed class. (SD Specifications Part I 2013)*

Speed Class	Minimum performance
0	Not specified
2	≥ 2MB/s
4	≥ 4MB/s
6	≥ 6MB/s
10	≥ 10MB/s

SDHC and SDXC memory cards always have performance equal to at least class 2. The minimum performance indicates the minimum transfer speed the card is capable of, slower speeds are also possible if the host doesn't support as high transfer speeds as the speed class would allow.

2.2.2 FAT file system

FAT (File Allocation Table) file system is often used in SD memory cards. In FAT file system the data is stored in blocks of predefined size (usually 512 bytes). These blocks all have an address in the physical memory. If the file consists of several blocks, the addresses of these blocks have to be stored in a table. This table is stored in main memory and is called a FAT. Often in larger disks, the blocks are grouped into clusters (allocation units) to reduce the size of the allocation table. Default cluster size for FAT16 file system with 1GB – 2GB volume size is 32kB. For FAT32 file system and 256MB – 8GB volume size the default cluster size is 4kB (Default cluster size for NTFS, FAT, and exFAT). FAT32 allows larger volume sizes than FAT16 and it has to be used in SD memory cards over 4GB. When formatting the SD memory card, it is possible to change the cluster size and the file system. The smaller the cluster size, the longer it takes to find all the clusters of the file being accessed. However using small cluster sizes saves disk space if there are many small files on the memory. The FAT reserves some space on the disk, depending on the allocation unit size and the disk capacity. With larger allocation unit sizes, less space is reserved. (Tanenbaum 2001)

The FAT file system works great when there is a need for rewriting the disk, which is often the case in SD-cards; when a file is deleted from the memory, all the allocation units containing the file are freed and can be used by new data. Downside is a slower read speed because the allocation units have to be found from the memory using the FAT before they can be read. (Tanenbaum 2001)

2.2.3 Flash IC-chip memory

The other memory type used in this thesis (for testing) is a flash IC-chip memory communicating via SPI-interface (SST25VF064C, 8MB memory). This memory is used with the simplest file management method, called contiguous allocation.

Contiguous allocation is simple to implement and has high performance. File is stored as a contiguous sequence of disk blocks, and only the address of the first block and the size of the file have to be known to access it (Tanenbaum 2001). Downside of the contiguous allocation is the fragmentation of the disk space; when a file is removed, certain amount of space is freed from a certain location on the disk. If a new file is larger than the removed file, it cannot be written to this same location because there aren't enough contiguous memory units. Contiguous allocation is used in this project's flash IC-chip memory to optimize the read speed. When new samples are copied into the IC-chip, all files are rewritten and the start-addresses and lengths of the copied wav-files are stored at the beginning of the memory, which are loaded into program memory when the device is powered up.

2.2.4 Comparison of external memories

Properties of most common external memory types are listed in Table 3.

Table 3 *Properties of some of the most common external memory types (CF Cards, CF+ and CompactFlash Specification 2003, Jahed 2007, Universal Serial Bus 3.0 Specification 2011)*

Memory	Interface	Max. transfer speed
SD Memory Card (High Speed mode)	9-pin	25MB/s
SD Memory Card (UHS-II)	17-pin	195MB/s
Compact Flash	50-pin	120MB/s
USB 2.0 flash drive	4-pin	60MB/s
USB 3.0 flash drive	8-pin	625MB/s

As can be seen from the table above, highest transfer speeds can be theoretically obtained with USB 3.0 flash drives (625MB/s). Compact Flash (CF) memory cards have been faster than SD memory cards for a long time because they offer 16-bit parallel data transfer. Only recently SD memory cards have started to compete with the speeds of the CF memory cards by using UHS-II transfer mode (up to 195MB/s transfer speeds are possible). In this thesis, high speed mode of the SD memory card is used (maximum transfer speed 25MB/s).

The maximum transfer speeds rarely correlate to practical speeds obtainable; data fragmentation and design of the memory unit have an effect on the performance. If the host device uses 1-bit transfer protocol, transfer speeds are lower. The read and write routines and data handling of the host device have also an effect on the transfer speed. So the transfer speeds are always case sensitive which has to be taken into account when designing a system depending on external memory.

2.3 Audio and signal processing

Audio-files are used in electronic drum applications to store the sounds associated to the drum triggers. When an acoustic drum is stroked very gently, it generates different sound when compared to a loud stroke. Volume, tone and length of the sound vary depending on how hard the drum is stroked. For electronic drums this means that several audio-files are required for each channel (for example five different samples recorded with different stroke-intensities). This way when the trigger pad is stroked gently, the audio-file containing the gentle stroke is played. When the same pad is stroked with more force, the audio-file containing more intensity is played.

2.3.1 Audio file format

In this thesis, WAVE –file format is used to store and read the audio samples, because it is uncompressed and easy to process. Microsoft Pulse Code Modulation (PCM) format is used with 16bits per sample (stereo). CD-quality audio has 16bits per sample, with 44100Hz sample rate, and this is used as a reference quality for the audio playback of the device constructed in this thesis.

In the first bytes of a 16-bit stereo PCM WAVE –file, there is a format chunk about the specific wav file (information about the number of channels, coding type, resolution, sampling frequency and word alignment). After the format chunk the audio data is stored as sequence of four consecutive bytes (see Table 4). When reading a wav-file, these bytes have to be combined into 16-bit samples before any signal processing can be done (the “lowbyte” and “highbyte” have to be combined for each sample). (Multimedia Programming Interface 1991)

Table 4 *Data order for 16-bit Stereo PCM WAVE. Format chunk is at the beginning of the file. (Multimedia Programming Interface 1991)*

Format chunk	**Sample1 left lowbyte**	**Sample1 left highbyte**	**Sample1 right lowbyte**	**Sample1 right highbyte**	**Sample2 left lowbyte**	**Sample2 left highbyte**	**Sample2 right lowbyte**	**Sample2 right highbyte**

A higher sample rate offers better audio quality. But the higher the sample rate, the more data there is to be processed, which means more time is consumed by reading the file from the memory and mixing it into other files. So there is always a tradeoff between processing time and audio-quality. Processing time could be reduced by using one-channel audio (mono), but this is not preferred in electronic drum applications. Stereo imaging plays an important role in the simulation of acoustic drums because the spatial locations of different parts of the drum set differ (for example hi-hat is usually on the left side of the drummer and ride on the right side). Because of this, drum-samples are often panned to form a stereo-image mimicking the real-life situation.

2.3.2 Signal processing

If two or more audio-samples need to be played simultaneously, they need to be mixed. If the samples are simply summed together, overflowing occurs because the range of an integer is limited (for 16-bit integer the range is -32768 to 32767). For example, if two samples of a value 20000 are added, the most significant bits of the result get discarded which distorts the audio-signal. The simplest way to avoid overflowing would be to divide the result by two, but this reduces the dynamic range to half (if we sum values 20000 and 0 and divide the result by two, we get 10000 which is half of the original).

Overflowing and loss of dynamic range can be avoided by using the following equation:

$$y = \begin{cases} (a+b) - \frac{a \cdot b}{MIN}, & (a < 0) \land (b < 0) \\ (a+b) - \frac{a \cdot b}{MAX}, & (a > 0) \land (b > 0) \\ a+b, & else. \end{cases} \tag{1}$$

The equation (1) is presented by Toth (2000) and Tyson (2011). In the equation (1), *a* and *b* are the sample-values to be mixed, *MIN* is the lowest and *MAX* the highest possible value the sample can have. Because the equation is nonlinear, slight distortion of the audio signal occurs at points where the condition switches over. In the electronic drum module constructed in this thesis the algorithm worked sufficiently and no distortion was observed. (Toth, 2000 and Tyson, 2011)

2.3.3 Audio Bitstream DAC and Direct Memory Access

The microcontroller's Audio Bitstream Digital-to-Analog converter (ABDAC) converts two 16-bit audio sample values into two digital bitstreams which are fed into two I/O pins of the microcontroller (right and left channels). The average value of the bitstream is proportional to the sample value. Two cascaded 1st order RC- low pass filters are used to remove the high frequency -switching component from the output of the ABDAC (design of the low pass filter is discussed in chapter 3.1.2). (AT32UC3A3/A4 Datasheet)

The sampling rate of the ABDAC-module is generated from the main clock by division, using the equation

$$f_s = \frac{f_{main}}{256 * 2 * (div+1)}. \tag{2}$$

The *div*-parameter is set when the ABDAC-module is first initialized. With 60MHz main clock -frequency used in this thesis the following sampling rates are possible: 16741Hz, 19531Hz, 23438Hz, 29297Hz, 39062Hz and 58594Hz (lower sampling rates are also possible but audio-quality is poor). 46875Hz sampling rate is possible by using

48MHz main clock frequency. To obtain 44100Hz sampling rate, 11.2896MHz crystal could be used. (AT32UC3A3/A4 Datasheet)

ABDAC is used with peripheral direct memory access controller (PDCA). The PDCA transfers data between on-chip peripheral modules and on/off-chip memories without CPU intervention (in this case between on-chip memory and ABDAC). This improves the performance of the microcontroller as CPU is freed to do other processing tasks simultaneously. The PDCA transfers the data to the ABDAC via two alternating buffers. How often these buffers need to be reloaded depends on the buffer size and the sampling frequency of the audio to be played. With higher sampling frequencies and smaller buffer sizes the PDCA has to be reloaded with a new buffer more often. If the buffer isn’t updated before the PDCA tries to transfer it to the ABDAC, the correct samples aren’t played and the audio playback gets distorted. (AT32UC3A3/A4 Datasheet)

3 DESIGN

In this thesis an electronic drum module was designed to test the suitability for micro SD memory card for storing the audio-files containing the drum samples. The drum module was designed to be used with drum trigger pads using 6.3mm stereo plugs for connection. Simple user interface was designed to make the device as easy to use as possible. In this chapter the design of the drum module is described in detail.

3.1 Hardware

Hardware was designed around AT32UC3A3-microprocessor, which was chosen because it is designed for audio applications and it supports the 4-bit SD memory card transfer protocol. First the device was designed at a functional level (block diagram), and then hardware was designed based on the requirements set for the device.

3.1.1 Operational principle

Block diagram of the constructed electronic drum module can be seen in Figure 2.

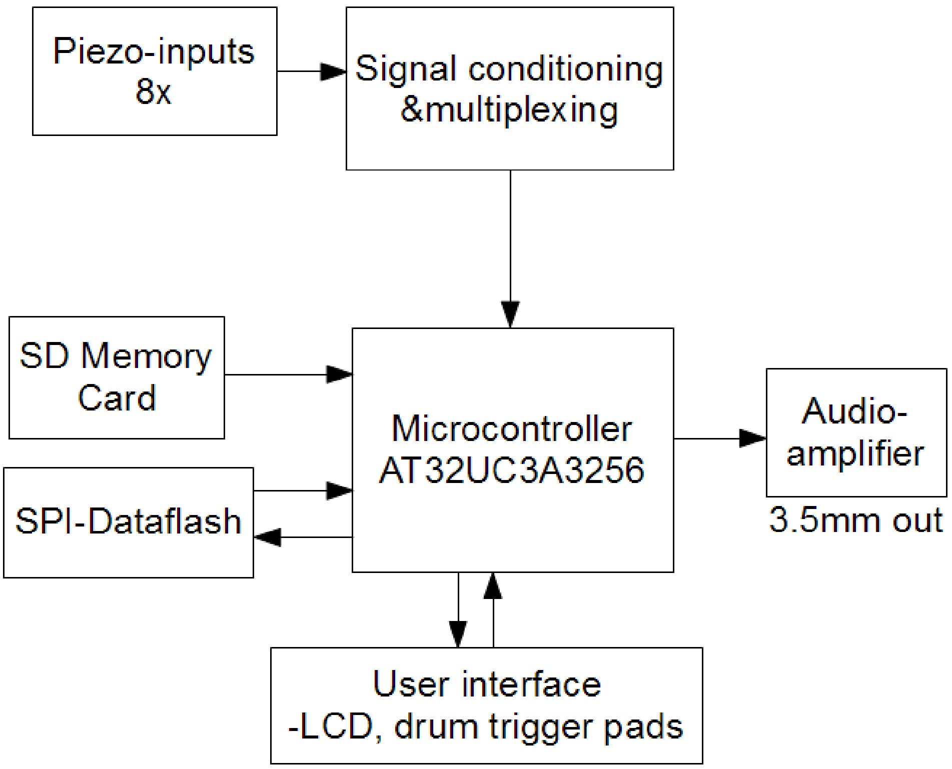

Figure 2 *Block diagram of the constructed electronic drum module.*

The drum trigger signals are clamped to appropriate level with a signal conditioning circuit seen in Figure 3. After the voltage clamping, eight different drum trigger inputs are multiplexed into two AD-channels of the AT32UC3A3256-microcontroller. Signal processing is done in the microcontroller. There are two external memories connected to the microcontroller:

- SD memory card communicating via MCI
- Flash IC-chip memory communicating via SPI.

The flash IC-chip memory is used only in testing phase. User interface consists of an alphanumeric 16x2 LCD-display, one button and the drum trigger pads. The audio signal is produced by the ABDAC-module of the microcontroller and external audio amplifier is used to amplify the signal. Stereo headphones or external audio-amplifier can be connected to the device via a 3.5mm stereo jack.

3.1.2 Hardware design

The device consists of two separate PCBs. One is the mainboard consisting of the microcontroller, voltage regulator, voltage reference, USB-connector, SD memory card slot, flash IC-chip memory and connectors for external components. The other PCB is for the drum trigger signal conditioning and multiplexing.

AT32UC3A3256 microcontroller is used in the electronic drum module to do the reading of the wav-files, detecting the drum trigger signals, signal processing etc. This specific microcontroller was chosen because it is specifically designed for audio applications (it has the ABDAC-module), and it has built-in interface for the SD memory card (MCI). USB-bus supplies 5V (VBUS), which is regulated to 3.3V (VCC) with LM1117 voltage regulator. ADE366 voltage reference is used to provide more accurate reference voltage for the AD-converter of the microcontroller. Midas 2x16 alphanumeric LCD is used in the device for setup and to display information. There is also a 64 Mbit SPI serial dual I/O flash memory SST25VF064C on the mainboard, and it is used for comparison in the testing-phase. TPA152D audio amplifier is used to amplify the signal coming from the ABDAC-outputs of the microcontroller. There is a 3.5mm stereo jack for audio-output.

The drum trigger pads can be connected to the back panel of the drum module with 6.3mm plugs. The drum trigger signal conditioning is done with BAT85 -Schottky diodes and series resistors seen in Figure 3.

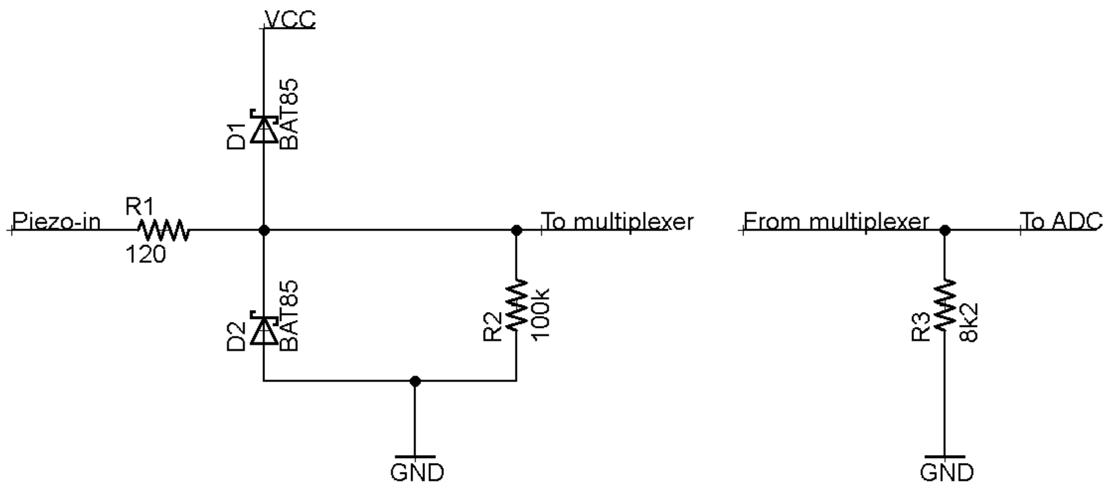

Figure 3 *The trigger signal clamping circuit used in the drum module.*

In Figure 3 the resistor R1 restricts the current coming from the piezo-element. The Schottky-diodes D1 and D2 clamp the voltage to appropriate level for the AD-conversion (-0.3…3.6V). The voltage clamping is done for each of the eight drum trigger inputs, which are then multiplexed into two channels going to the ADC-inputs of the microcontroller. Two MC4051B multiplexers are used, and three microcontroller GPIO-pins are used to control the switching of the multiplexers.

Stroking a drum pad causes a voltage peak on that piezo-channel. When it is multiplexed into an ADC-pin, the inner capacitance of the ADC-pin is charged. After the channel is measured, next channel is multiplexed into the same ADC-pin. Even if the next channel has no voltage, there is still some leftover voltage on the inner capacitance of the ADC from the previous piezo-channel, which could cause a false detection. Because of this, 8.2kΩ resistors are connected between the ADC-pins and ground to pull the ADC-pins low when no voltage is applied to them (R3 in Figure 3). Large 100kΩ resistors are connected between each of the piezo-channels and ground for similar reasons (R2 in Figure 3); if the piezo-channels are left floating when they are not multiplexed into the ADC, it could take too long after a trigger-stroke for the voltages to drop to zero.

The ABDAC output is filtered with two cascaded RC-filters seen in Figure 4.

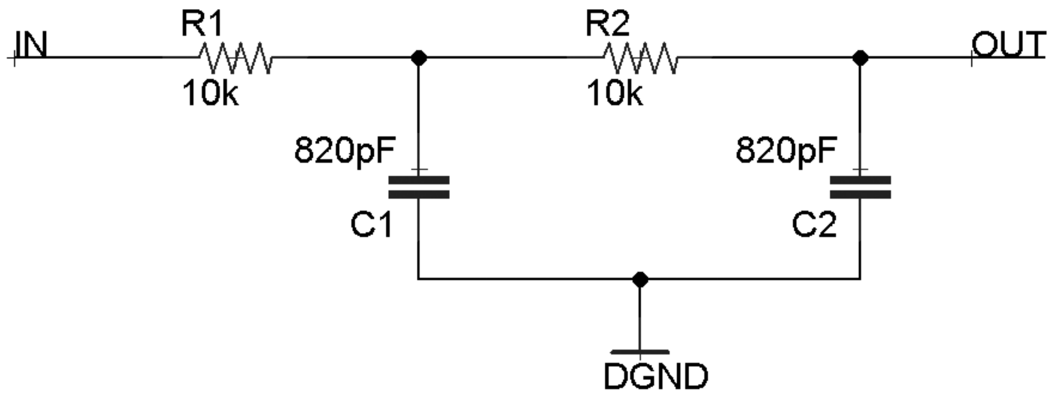

Figure 4 *The low-pass filter used for the ABDAC-output.*

Two RC-filter stages are used to provide more attenuation for the frequencies above the cutoff frequency. The cutoff frequency of this filter can be calculated from the equation

$$f_c = \frac{1}{2\pi RC}, \qquad (3)$$

if the both filter-stages have same resistor and capacitor values (Kuphaldt 2000). Using the equation (3) for the used RC filter we get cut-off frequency f_c = 19.4kHz. The human hearing range is usually 20Hz-20kHz (Grimnes et. al. 2000), so the designed filter is suitable for this application.

The schematic for the micro SD memory card socket can be seen in Figure 5.

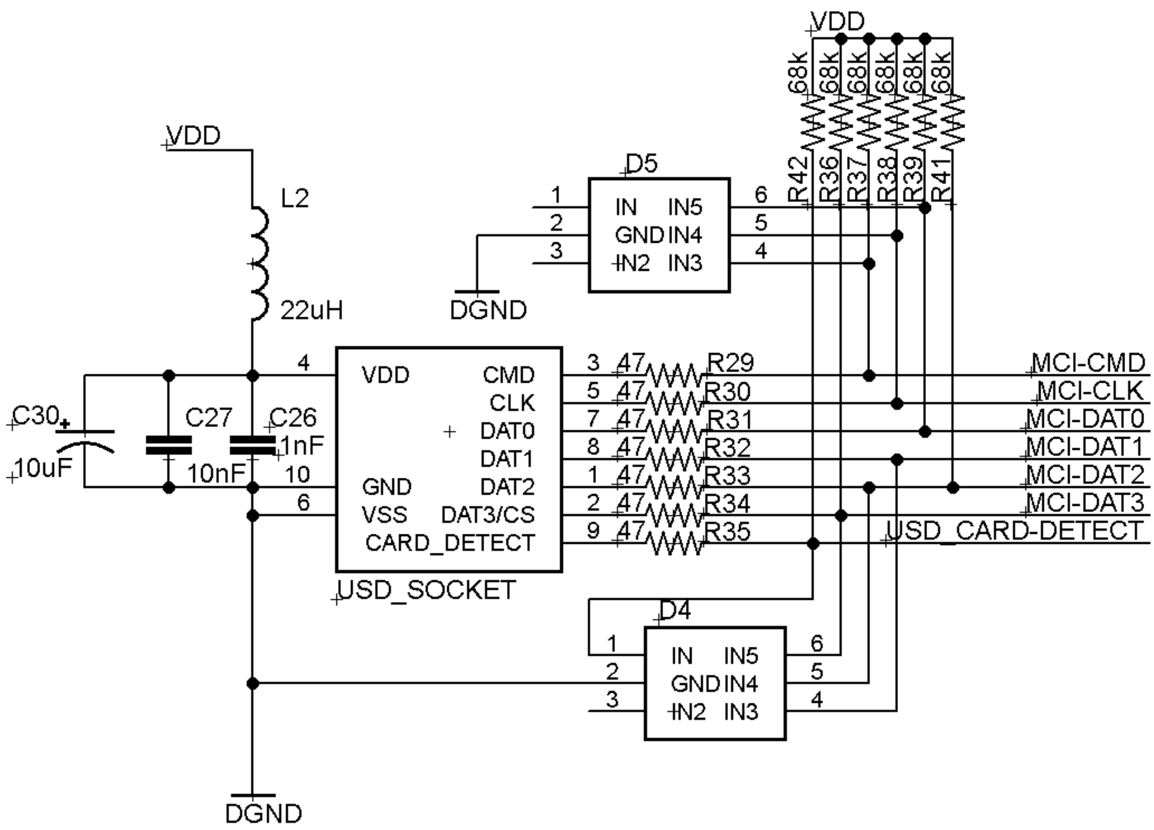

Figure 5 *Micro SD memory card socket*

SD memory cards have built in capacitors. When the memory card is being connected to the socket, the VDD voltage dips because the inner capacitor has to be charged. Because of this, the inductor L2 is connected between the main power supply and the SD memory card power supply. Together with the capacitor C30 they prevent incorrect operation caused by hot insertion/removal of the memory card (ChaN, 2013). Capacitors C26 and C27 filter the high frequency disturbances caused by SD memory card transfers. All the lines going to the microcontroller have 47Ω series resistors (R29…R35) to protect the microcontroller from electrostatic discharges (ESD) caused by the insertion of the SD memory card. The series resistors work together with TVS diodes D4 and D5 (SP1001 Unidirectional TVS Array) absorbing ESD strikes. All the lines except the CLK require pull-up resistors (R36…R39, R41, R42). (SD Specifications Part I 2013)

3.2 Software

The software of the drum module was written in C -programming language. Atmel Studio 6 was used as a programming environment and Atmel Software Framework (ASF) as a software library. Modules from the ASF library were used to communicate with different hardware interfaces such as the MCI, ABDAC, PDCA and ADC. JTAGICE mkii was used for programming and debugging the device.

3.2.1 Program operation

The program works by detecting the drum trigger peaks in an interrupt service routine (ISR) and reading and mixing the corresponding wav-samples in the main-loop. PDCA is used to transfer the mixed data into the ABDAC. The mixed samples are saved into a mix-buffer. When the PDCA finishes loading the previous mix-buffer, interrupt is triggered. In this interrupt service routine, the new mix-buffer is appointed to the PDCA. Figure 6 describes how the main-loop operates.

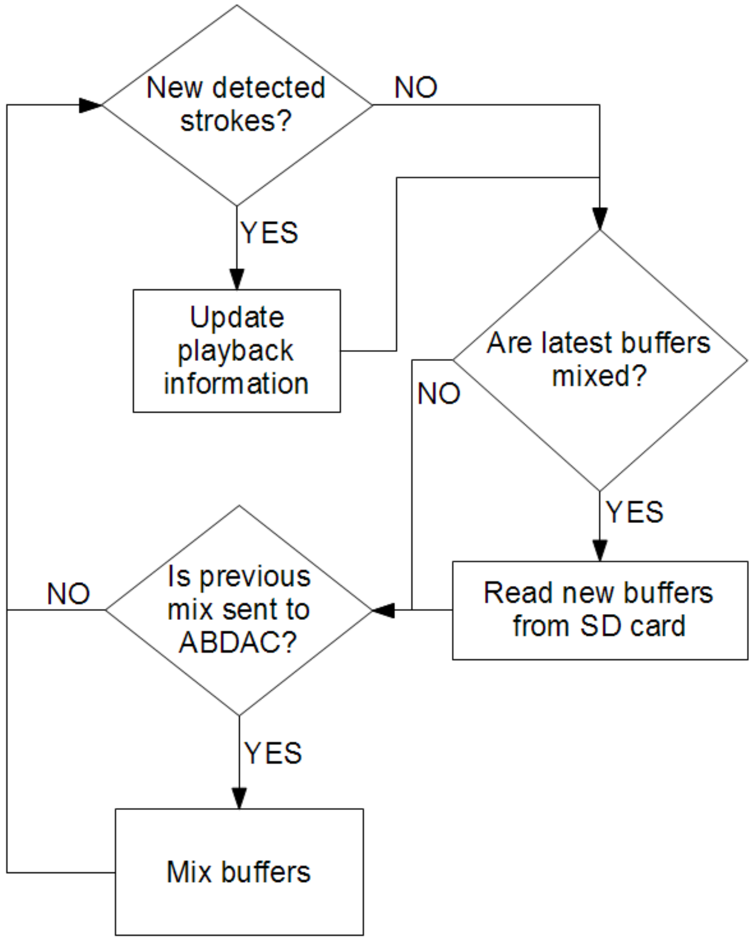

Figure 6 *Flowchart of the main-loop.*

When the drum module is active, the process described in Figure 6 loops forever (until the eject-button is pressed). First, if new drum trigger strokes are detected, playback information is updated. The playback information tells the SD memory card -read routine what files should be read. If the most recent read-buffers are mixed, the buffers can be updated with new samples from the SD-card (if there is something to be played back). After this, if the previous mix-buffer transfer is ready the next samples can be mixed. If there are no new samples to be mixed, zeros are sent to the ABDAC and nothing is played back.

3.2.2 Signal detection

The detection of the drum trigger signals is done in ISR every 0.2 milliseconds. Figure 7 describes the operation of the ADC ISR.

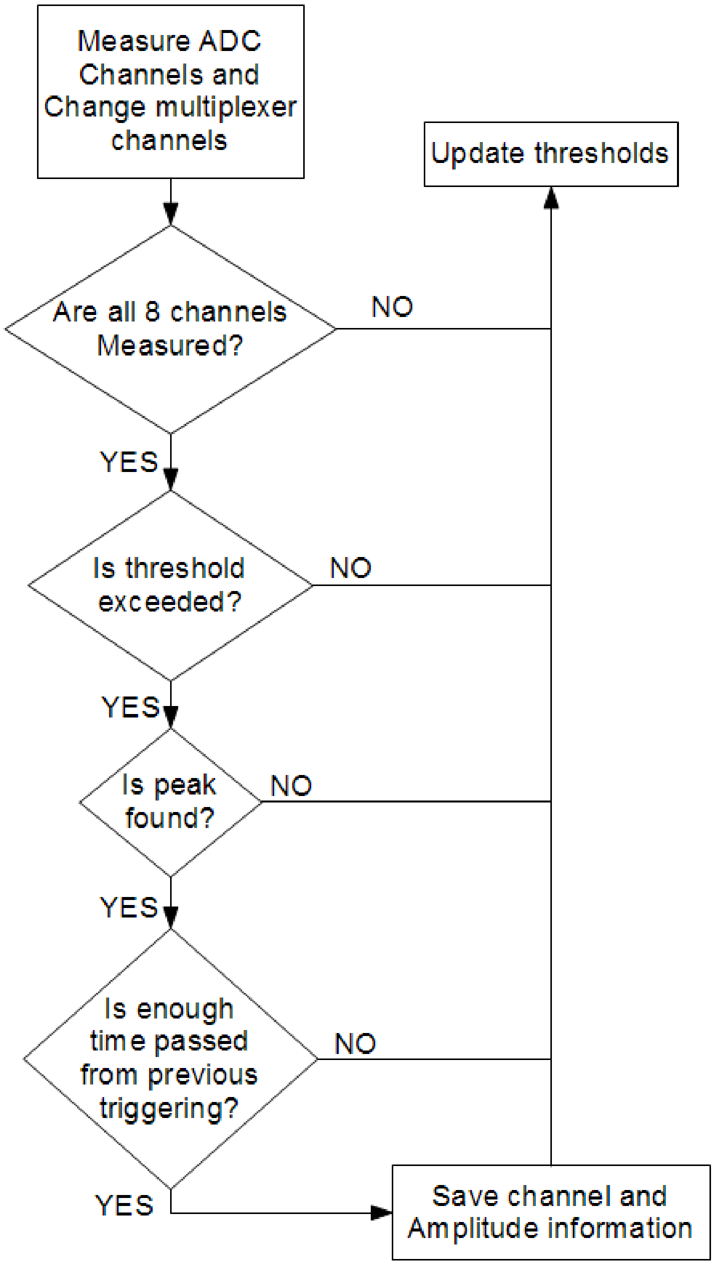

Figure 7 *Flowchart of the ADC ISR.*

Amplitudes of the two ADC channels are measured each time the ISR is executed. Then the multiplexer channels are changed so that next time different drum trigger channels are measured. It takes four ADC ISR executions before all the channels are measured, so each channel is measured every 0.8 milliseconds (4x0.2ms). Then for any of the channels, if threshold is exceeded, if peak is found and if enough time has passed from previous triggering of the same channel, amplitude and channel of the detected signal is saved. In main-loop, this information is used to determine which wav-file should be read and sent to the ABDAC.

Adaptive threshold is used in the detection of the trigger-signals. The adaptive threshold is illustrated in Figure 8.

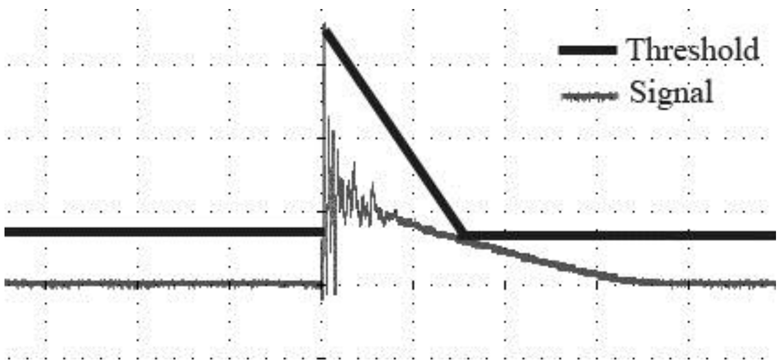

Figure 8 *Illustration of the adaptive threshold used in the drum trigger signal detection.*

Each time a peak is detected threshold is set to this peak-value. Each time the ADC ISR is executed, the threshold is reduced by a certain constant value. However the threshold cannot be lower than the noise-floor of the measured signal, so the threshold has a baseline value to which it decreases to after a certain amount of time. This baseline value can be changed in the setup of the drum module by adjusting the threshold-parameter. The slope of this adaptive threshold depends on how much it is decreased each round. The slope-parameter can be changed in the setup of the drum module.

Adaptive threshold doesn't always work, because there can be multiple peaks with same amplitude caused by the same stroke (for example due to vibration of the trigger pad), which results in multiple peak detections. Because of this, after a peak is detected from a certain channel it is blocked, so that detection is impossible for a predefined amount of time. This time period can be changed in the set-up by adjusting the parameter called inhibit. The inhibit-parameter is the value to which a specific counter value is compared; when a peak is detected on a certain channel, the play information for this channel is saved and the inhibit-counter for that channel is set to zero. The inhibit-counter value is increased by one each time a counter-ISR is executed. When detecting the next peak, the inhibit-counter value is compared to the inhibit-parameter. If the counter value is higher than the inhibit-parameter this means enough time has passed, the peak can be interpreted as a new stroke and the playback information can be updated. The counter-ISR is executed every 1ms.

If a new stroke is detected, the amplitude value of the corresponding channel is updated. The amplitude can have values between the threshold and 1023. Based on the amplitude value, one of the five audio files associated to that channel is played back (see Table 5).

Table 5 *Relation of piezo signal amplitude to the audio file to be played back.*

Amplitude value	Audio file to be played back
Threshold…49	CH0X-04.wav
50…119	CH0X-03.wav
120…249	CH0X-02.wav
250…499	CH0X-01.wav
500…1023	CH0X-00.wav

Table 5 tells which file is played when different amplitudes are measured. If the amplitude is below the threshold level, no audio file is played back. With the lowest amplitude values, the audio file containing the weakest stroke intensity sample is played back (CH0X-04.wav). With amplitude values 500-1024, the sample containing the strongest stroke intensity is played back (CH0X-00.wav). The naming of the audio files is described in detail in chapter 3.3.2.

3.2.3 Using the SD memory card

To access the SD memory card it has to be first initialized and identified. The card initialization and identification sequences can be found for example in SD Specifications Part 2, 2011. Modules from the ASF are used to communicate with the SD memory card. The initialization sequences are abstracted into the SD/MMC drivers of the ASF, which makes the MCI fairly easy to use.

Because FAT -file system is used on the SD memory card, file system management is needed to access the files. Atmel provides a file system management to be used with external memory cards. It consists of four modules: The navigation-module, the file-module, the fat-module, and the memory-module. The navigation-module provides routines to select navigator handles, to navigate in the file list, to change directories, to modify the directory tree and to get information about the files. Multiple files can be accessed using the navigation handles. The file-module provides routines to read and write files on the SD memory card. The fat-module is private and it decodes the FAT structure (it is used by other modules, not by the programmer). The memory-module is also used by other modules, and it provides an interface to the memory.

In the drum module constructed in this thesis, the navigation handles are used to open all the wav-files on the SD memory card (40 handles altogether, five for each channel). This way when a certain wav-file has to be accessed, it is already open and can be accessed quickly using the navigation handle for that file. The file-module from ASF is used in this thesis to read wav-files from the SD card into buffers (using the function ‘read_into_buffer()’ from the file-module). (AVR114: Using the ATMEL File System management, 2008)

3.2.4 Audio processing

When the program is started, lengths of all the wav-files in the SD memory card are read from the info chunks of the files and saved into a table to be used later on (the length-information is used to check how long each wav-file needs to be played back). The sample rate is read from the info chunk of the first wav-file, and the ABDAC sampling frequency is adjusted based on the sample rate. Same sample rate has to be used for each wav-file.

The audio files are stored in the SD memory card in wave-format, which stores the samples as sequence of four consecutive bytes (16bit stereo PCM WAVE). Before a specific wav-file can be sent to the ABDAC, it is read into 8-bit buffer 2048 bytes at a time (i.e. size of the buffer is 2048 bytes). There are four individual buffers that can each be filled with samples from different files. This enables maximum of four wav-files to be played back simultaneously.

Before two samples can be mixed together, the data has to be converted into the original 16-bit format. This is done simply by shifting bits and adding the low-order and high-order bytes together. After conversion into 16-bit integers, the samples are mixed with the method presented in equation (1). C-code implementation of this method can be seen in Figure 9.

```
int16_t MixSamples(int16_t a, int16_t b)
{
        int16_t x;

        if (a<0 && b<0)
        {
                x = (((int)a + (int)b) + (((int)a*(int)b)/32768));
        }
        else if (a>0 && b>0)
        {

                x = (((int)a + (int)b) - (((int)a*(int)b)/32768));
        }
        else
        {
                x = (a+b);
        }
        return x;
}
```

Figure 9 *The function used to mix two samples together*

In the mixing function seen in Figure 9, the 16-bit integers are first cast to 32-bit integers to prevent overflow when large values are multiplied. The result is cast back to a 16-bit integer after the operation. This mixing procedure is first done for the first samples of the first and second buffers. Then the first sample of the third buffer is mixed into this result. Finally, the first sample of the fourth buffer is mixed into the result, which gives us a value that has components from all of the four buffers. The buffers containing no samples to be played are filled with zeros, indicating silence. When the mixing is ready, the 16-bit right and left channel -samples are combined into a 32-bit sample by shifting the right channel bits left. This way the samples for both channels can be moved to the ABDAC via PDCA simultaneously by writing the samples into a 32-bit buffer (size of this buffer is 2048/4 = 512).

3.3 User interface

User interface of the device is very simple. When the device is powered up, it is possible to go into setup by hitting one of the trigger pads at the startup. In the setup, it is possible to change:

- The piezo-trigger threshold baseline
- The slope of the adaptive threshold of the piezo-trigger sensing
- The minimum time between subsequent strokes of the same drum trigger.

After the setup, device goes into play-mode and the user can begin to play the drums. Before shutting down the device the switch on top of the device should be pressed to end ongoing SD memory card transfers to prevent file corruption.

3.3.1 Mechanical specifications

The designed drum module is installed into a durable plastic enclosure. The eight drum trigger inputs (6.3mm stereo jacks) are installed on the back panel of the device (see Figure 10). The drum module has a micro SD memory card slot to be used with SDSC or SDHC cards, which can be accessed via the top panel (see Figure 11 and Figure 18).

At the right panel of the drum module (see Figure 12) there is a USB type A connector for power supply (5V), on/off switch, 3.5mm jack for audio output and 250mA/250V fuse for protection against electric faults. The device has a 16x2 alphanumeric LCD-display located at the front panel (see Figure 13), which is used for setup and displaying messages. There is an eject-button for safe removal of the SD memory card above the LCD-display.

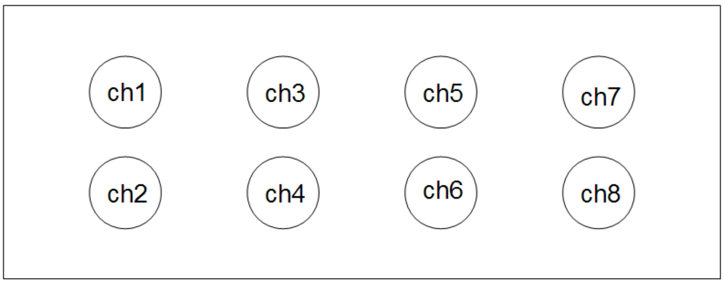

Figure 10 *Back panel of the electronic drum module.*

The drum triggers can be connected to the back panel of the device. The organization of the inputs can be seen in the picture above. The ch1-input is associated to audio-files CH01-00.wav…CH08-04.wav (see 3.3.2 for detailed description of the wav-file association). Both stereo and mono plugs (6.3mm) can be connected to these inputs (the tip of the plug is measured).

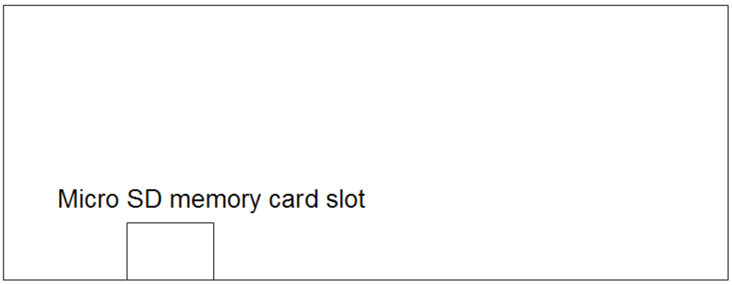

Figure 11 *Top panel of the electronic drum module.*

Micro SD memory card slot is soldered into the PCB, and it can be accessed via the top panel (see Figure 11). See chapter 4.2 for detailed description of the memory card slot usage.

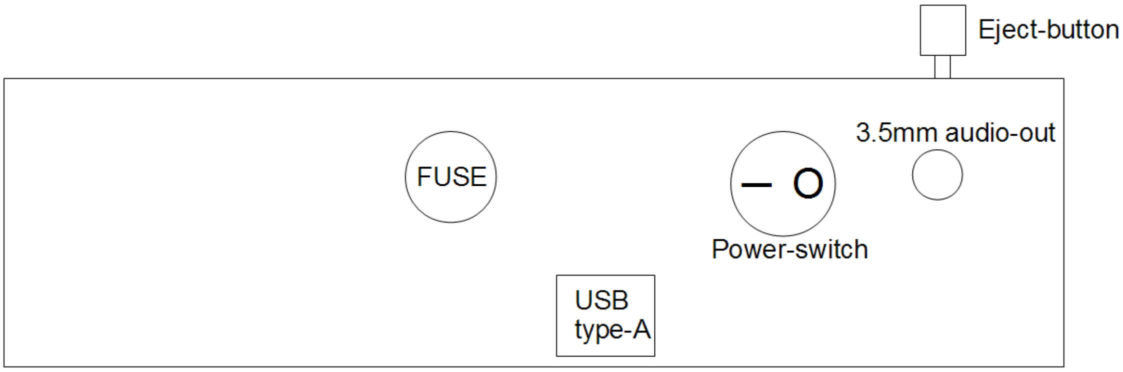

Figure 12 *Right panel of the electronic drum module.*

On the right panel there is

- The USB type-A slot, which is soldered into the PCB
- The power switch, switching the power coming from the USB supply
- The fuse, which is in series with the USB supply voltage
- The 3.5mm audio output (3.5mm stereo).

The device is powered from the USB type-A slot, providing 5V. USB data connection is not necessary. The power is on when the power switch is in I-position (and off in O-position).

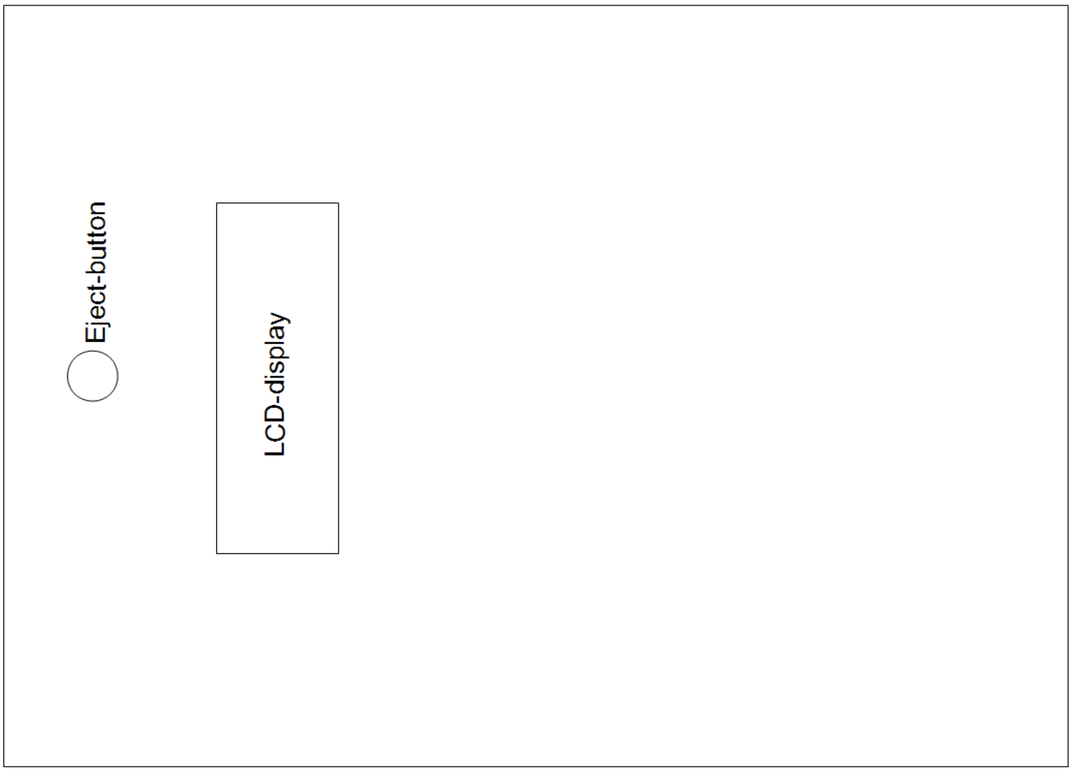

Figure 13 *Front panel of the electronic drum module.*

The LCD-display is mounted at the front panel of the device. Messages displayed on the LCD are described in chapter 3.3.3. The eject button above the LCD is used to safely remove the SD memory card from the system (when the button is pushed, any ongoing SD memory card transfers are stopped and the SD memory card can be removed from the memory card slot).

3.3.2 Wav-files

There are five different audio files associated to each channel (each file associated to different stroke intensity of the specific channel, see chapter 2.3). The way these audio files should be named on the SD memory card is predefined. For example, the wav-files associated to channel 1 are named CH01-00.wav, CH01-01.wav, CH01-02.wav, CH01-03.wav and CH01-04.wav. Different wav-file of the channel group is played back depending on how hard the drum trigger connected to that certain channel is stroked (CH01-00.wav contains the hardest stroke intensity sample and CH01-04.wav the weakest, CH01-01.wav…CH01-03.wav are in between). The wav-files should be in 16-bit stereo PCM –format, otherwise they are not played back correctly. Even if some of the channels are not used, all the wav files should exist on the SD card (CH01-00.wav…CH08-04.wav, in all 40 files).

All the wav-files on the SD memory card should have the same sampling frequency; otherwise the sampling frequency used for playback is not correct for every sample. The sampling frequencies supported by the device are: 16741Hz, 19531Hz, and 23438Hz (higher sampling frequencies are also tested, results are described in 5.2.3). If the sampling frequency of a wav-file is something else than the aforementioned, next supported higher sampling frequency is used, which results as slightly faster playback.

When designing the audio files to be used with the device, highest available sampling rate should be used when recording the audio. One could for example record five different audio files for each drum with sampling rate 44100Hz (first stroking the drum very gently, then increasing the stroking intensity for the next audio file, and finally for the last audio file stroking the drum with the maximum intensity). After recording, the audio files should be rendered into a suitable sampling rate with PC software (for example from 44100Hz into 23438Hz). Finally, the files should be renamed to match the desired channels on the device (CH0X-00.wav…CH0X-04.wav) and copied into the SD memory card.

3.3.3 Displayed messages

The drum module can print various messages onto the LCD screen related to the usage and error-situations (Figure 14).

Hit pad 3 for setup	Set threshold: 0010	Set slope: 0001
a)	b)	c)
Set trig inhibit 0080	Insert SD card	Card detected!
d)	e)	f)
Ready to play	File open error! Not found: 0001	File open error! File-open fail
h)	i)	j)
OK, Remove card		
k)		

Figure 14 *Messages displayed on the LCD-screen in different situations.*

When the device is powered up, it is possible to enter the setup by stroking the drum trigger pad connected to input 3 (Figure 14a). The setup is described in 3.3.4. If the setup is not entered or after the setup is done, the device asks to insert the SD memory card (Figure 14e). When card is successfully detected and initialized, the message in Figure 14f appears on the screen briefly. If all files are successfully found from the SD memory card and the program is able to successfully access them, the message in Figure 14h appears on the screen and the device is ready to be played. When the eject-button is pushed, the message in Figure 14k is displayed on the screen and the SD card can be safely removed from the device.

If one of the wav-files cannot be found, the message in Figure 14i is printed on the screen. The number after the “Not found:” –text specifies which file could not be opened (0001 for CH01-00.wav, 0002 for CH01-01.wav, 0040 for CH08-05). If one of the files is not found, the user should check that all wav-files exist on the SD memory card and the filenames are correct. If one of the files cannot be opened, the message in Figure 14j is printed on the screen. In this case the file-format and decoding of the wav-files should be checked.

3.3.4 Adjusting parameters

Some of the parameters related to the drum trigger detection can be adjusted by entering the setup. These parameters are threshold, slope and inhibit. Threshold-parameter is directly the value to which the ADC-measurement value of the drum trigger -input is compared (i.e. the minimum value the piezo signal should have to be interpreted as caused by a stroke). For the 10bit ADC used in this thesis, the resolution is 1024. So if the threshold is set to 10, the minimum voltage level that causes piezo-detection is 3.3V*10/1024 = 32mV. The slope-parameter defines how much the adaptive threshold is decreased each time the ADC-ISR is executed (see 3.2.2). If the slope is set to 1, the adaptive threshold decreases slowly. If the slope is set to 10, the decrease is 10-times faster. The optimal value for the slope-parameter depends on the drum trigger pad used, but for most trigger pads the default value should be suitable. The inhibit-parameter defines how fast subsequent detections can be made; if the inhibit-parameter is set to 75, the minimum time that has to go by between subsequent strokes for a certain channel is 75ms. This would allow 16^{th} notes to be played at 200bpm tempo. If this parameter is set too low, the vibration of the drum trigger pad could cause false detections right after the pad has been stroked.

The setup can be entered at the startup, by stroking the drum trigger pad connected to input 3 when the text in Figure 14a is shown on the LCD display. If the pad isn’t stroked within 2 seconds, default values will be used for these parameters. The default values are:

- Threshold = 10
- Slope = 1
- Inhibit = 80.

If the setup is entered, the message in Figure 14b is shown on the LCD screen. Threshold can be set to values 2…400. When the drum trigger pad connected to input 3 is stroked, threshold is increased by one. If threshold exceeds 400, it is reduced back to 2. When the threshold is set, the pad 5 should be stroked, which saves the chosen threshold value and the program moves on to the next parameter which is the slope (Figure 14c). The slope is adjusted the same way and it can be set to values 1…10, the value is saved by stroking the drum trigger pad connected to input 5. After the slope is set, inhibit-parameter can be adjusted between 5…100 (Figure 14d). To exit the setup, the drum trigger pad connected to input 5 should be stroked third time. These parameters have to be changed each time the drum module is powered up. Otherwise the default values will be used.

4 PROTOTYPE MANUFACTURING

The electronic drum module manufacturing is described in this chapter. The PCB was manufactured with etching process. The components were soldered manually and thorough testing was performed to ensure proper functioning of the device. Some changes were made during the prototyping phase to make the device to function properly, which are also described in this chapter.

4.1 PCB manufacturing and assembling

The constructed device consists of two two-sided PCB's manufactured by using etching process. The layouts were inkjet-printed on transparent substrates, and these substrate were used on top of the PCB's coated with a photoresistive material to confine the areas to be etched (the light sensitive coating is removed when UV light is applied). After the UV exposure, the PCB's were developed in 1% sodium hydroxide (NaOH) solution. Natriumpersulfat was used in the etching process to remove any unwanted copper from the PCB.

After the etching, the PCB was cleaned and flux was applied to both sides to prevent oxidation and to allow the solder to flow on the copper (Boothroyd et. al. 2001). Holes were then drilled manually to the PCB (1mm hole-diameter). After this, all the vias were soldered manually with a lead free solder using tinned copper wire for connections. All the components were also soldered manually, and insulated copper cable was used to make the connections between the two PCB's. Copper cable was also used to connect the LCD, the ON/OFF switch, the eject-button, the 3.5mm audio jack, the fuse holder and the 6.3mm jacks to the device.

After the soldering of the components, all the electrical connections were inspected with a digital multimeter to verify the conductivity. The device was then powered up and all the regulated voltages were tested. After this, test program was used to test the functioning of the microcontroller by flashing two LEDs on the PCB.

4.2 Assembly of the device

Simplified exploded view drawing of the constructed device can be seen in Figure 15.

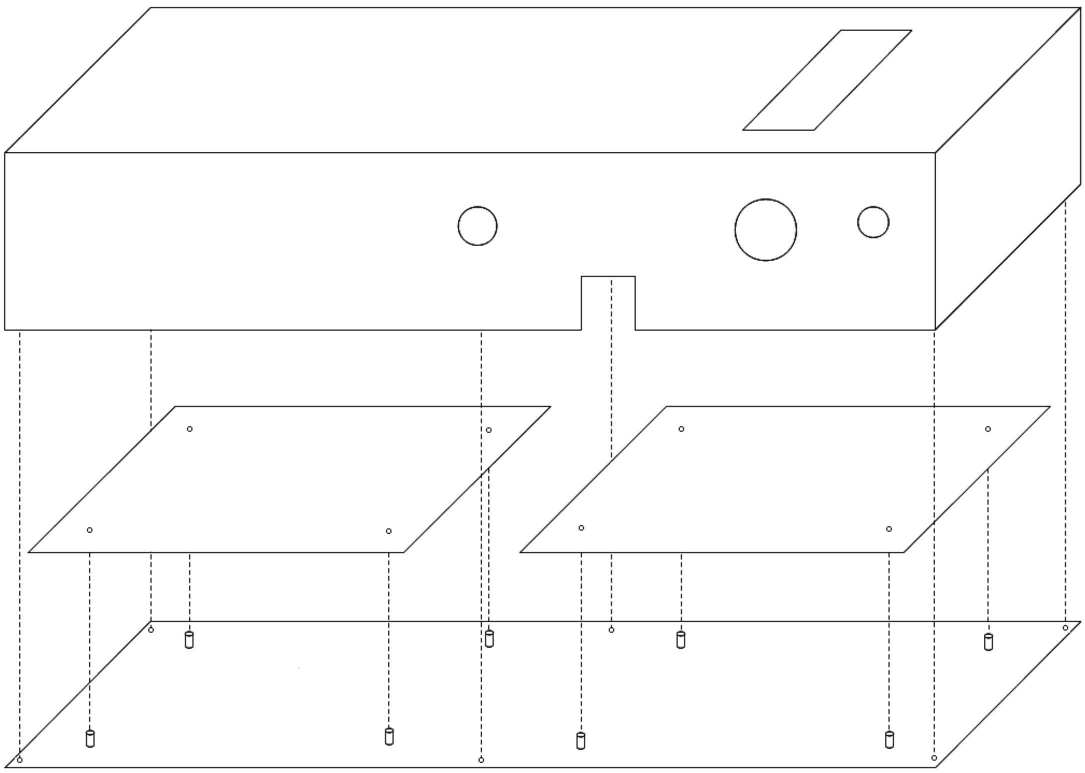

Figure 15 *Exploded view drawing of the electronic drum module.*

The back-lid of the device can be removed; it is attached with six screws. The two PCBs are attached to the back-lid of the device with 8 screws (4 screws each). The diameters of the holes drilled to the enclosure are listed in Table 6.

Table 6 *Hole diameters.*

I/O	Hole diameter
6.3mm stereo jack for drum trigger inputs	10mm
ON/OFF switch	19mm
3.5mm audio out jack	12mm
Eject-button	8mm
Fuse-holder	15mm

The LCD, USB and micro SD -outlets were carved to the enclosure manually by using an electrical mini grinder. The LCD was attached to the enclosure with hot-melt adhesive.

The finalized prototype can be seen in Figure 16, Figure 17 and Figure 18.

Figure 16 *Inside of the electronic drum module.*

The PCB containing the microcontroller and the main components can be seen at the foreground of the picture above. The PCB containing components for the drum trigger signal conditioning can be seen at the background. Figure 17 shows the enclosure of the drum module.

Figure 17 *Constructed electronic drum module.*

The electronic drum module consists of 8 drum trigger inputs, 16x2 alphanumeric LCD-display, micro SD memory card slot, 3.5mm Audio output (stereo), USB type-A connection for power, fuse holder, switch for safe removal of the SD-card and on/off switch. The SD memory card slot can be seen in Figure 18.

Figure 18 *Top panel of the device. The micro SD memory card slot can be seen at bottom of each picture. The upper picture shows the card before it is pushed into the slot. The lower picture shows the card in place.*

SD memory card is inserted into the slot by pushing it into the memory card slot, connectors facing up. The card locks on to the slot when a click is heard. To remove the card, it is pushed until a click is heard again. After the click the slot automatically pushes the card outwards and it can be pulled off from the device.

4.3 Changes during the prototyping phase

When testing the audio of the device, it was noticed that muting of the audio output could be beneficial. The used audio-amplifier (TPA152D) has a mute-pin which mutes the output when connected to VCC. NPN-transistor was added to the circuit to be used as a switch to connect the mute-pin either to ground or to VCC. Microcontroller’s I/O pin was used to control the transistor to allow the audio-amplifier to be muted with the software.

The audio-volume was designed to be adjusted with a linear 10kΩ potentiometer. Voltage division circuit was formed with the potentiometer, and the voltage division was measured with the ADC of the microcontroller. The volume-level was designed to be changed by adjusting every sample value fed to the ABDAC according to the measured potentiometer-voltage. However, the volume adjustment was left out from the finalized prototype because it consumed too much processing time. Better option for volume adjustment would have been to do the attenuation of the audio-signal with hardware. This would have taken no processing time, but because the device was already constructed the volume adjustment was left out.

The microcontroller has a built-in Hi-Speed USB Interface, which was planned to be used to make the electronic drum module to function also as a MIDI-device. This would have allowed the device to be used for example with VST-instruments in PC software (the sounds would have been generated by the PC from the MIDI-messages sent by the drum module). However, libraries for this microcontroller to be used in USB MIDI class weren’t readily available. It would have consumed too much time to construct the needed software from the scratch, so this feature was left out from the finalized prototype.

The drum module was first designed to be used with 16 channels (two drum-triggers can be connected to each of the 6.3mm stereo jacks). This feature could have been used with dual drum triggers which have two built in piezo-elements (for example a cymbal pad may have two piezo-elements, one for the bell (raised section around the hole of the cymbal) and other one for the bow (rest of the cymbal-surface). However when a dual-trigger pad is stroked, both of these piezo-elements create a measurable voltage. The recognition of which of the two channels is actually stroked caused to be problematic. Because the number of channels isn’t a key aspect in this thesis, it was decided that only eight channels would be used (one for each of the 6.3mm jacks).

5 RESULTS

The electronic drum module was thoroughly tested and the most important results are summarized in this chapter. The piezo signal conditioning circuit was tested, and the current consumption of the device was measured in different states (chapter 5.1). Real-time performance of the drum module, memory read speeds and highest obtainable sampling frequencies were tested and the results are shown in chapter 5.2. Because the electronic drum module plays sounds, audio quality was also evaluated (chapter 5.3).

5.1 Hardware testing

The hardware of the electronic drum module was tested first in the manufacturing phase (electrical connections, voltage supplies etc.). After the device was constructed, the piezo signal conditioning circuit was tested. The current consumption of the finalized device was also measured.

5.1.1 Piezo signals

Piezoelectric sensors can create large transient voltages that can break the measurement device. This is why the signal has to be restricted to a suitable voltage range before it can be fed to a microcontroller. Piezo-signals were measured with a digital storage oscilloscope, with and without signal conditioning.

First the waveform was measured straight from the output of a Roland PD-8 – drum trigger pad (no signal conditioning). The drum trigger pad was stroked with medium force and the output was measured. The result can be seen in Figure 19.

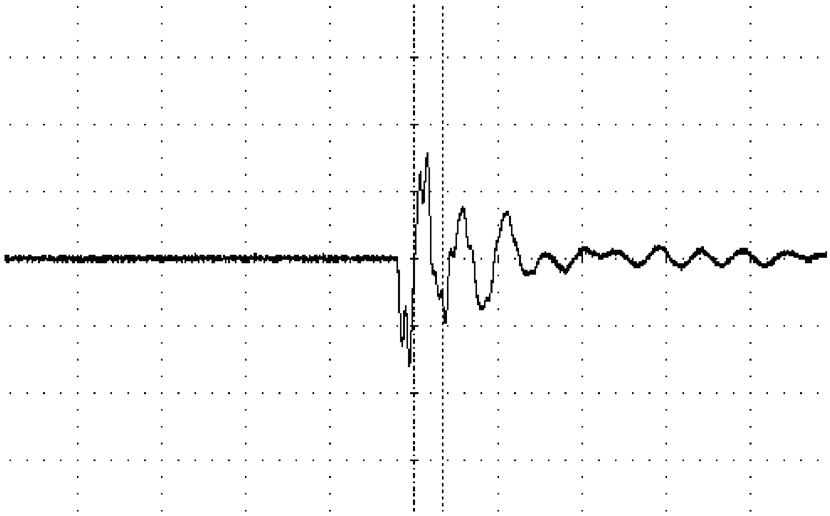

Figure 19 *Piezo-signal measured straight from the output of a Roland PD-8 –trigger pad when it was stroked with medium force. 2V/div***,** *5ms/div.*

There is an unwanted AC component in the unconditioned piezo-signal (in addition, some piezo elements produce as high as 50V voltage-peaks when a lot of stress is applied to them), so signal conditioning is needed.

The piezo signal conditioning circuit seen in Figure 3 was tested. Waveform of a piezo signal when the drum trigger pad is connected to the conditioning circuit can be seen in Figure 20.

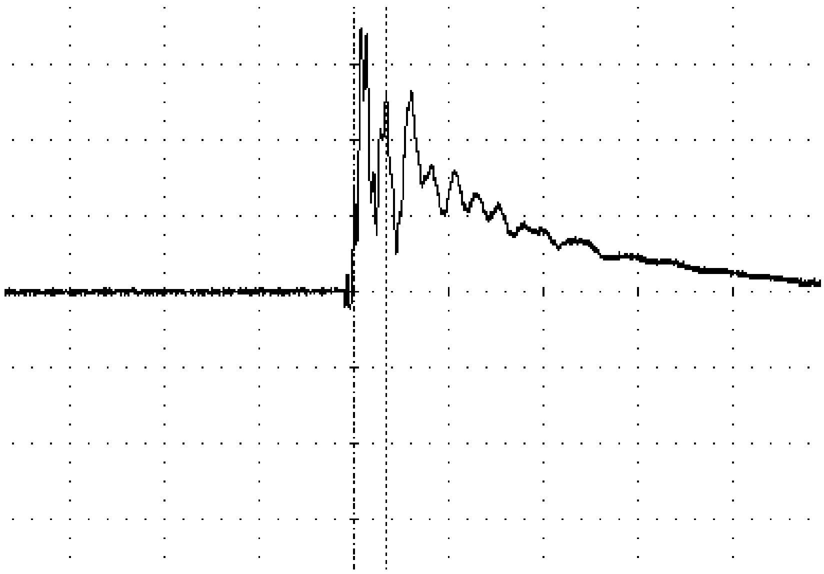

Figure 20 *Waveform of the piezo signal after voltage clamping. Roland PD-8 –trigger pad was stroked with medium force. 1V/div***,** *10ms/div.*

The waveform seen in Figure 20 is in suitable range for AD-conversion. It is still quite unstable because of the higher frequencies caused by the vibration of the piezoelectric sensor. These higher frequencies are filtered by the software, using gradually decreasing threshold for each input channel (see chapter 3.2.2).

5.1.2 Current consumption

Current consumption was measured with an adjustable power source with built-in current meter. Results can be seen in Table 7.

Table 7 *Current consumption of the electronic drum module in different states.*

Operation mode	**Current consumption**
Idle (active, no read or playback in progress)	75mA
Reading from SD memory card & playing audio	94mA
Reading from flash IC-chip memory & playing audio	80mA

In idle state, when the device is ready to be played, current consumption is approximately 75mA. When the SD memory card is being read and audio-sample is being played, the current consumption increases momentarily to 94mA. The flash IC-chip memory consumes less current than the SD memory card; the increase is only 5mA from the idle state.

5.2 Real-time performance and read-speeds

Electronic drum applications demand fast response times from the system. The maximum acceptable delays, SD memory card read speeds and highest obtainable sampling frequencies were tested to find out the performance of the constructed electronic drum module.

5.2.1 Delay between trigger-pad hit and sound playback

Audio-sample routed to a certain pad cannot be played back right away after the pad has been hit. The sample has to be first read from the memory and the required signal processing has to be done to get the all the simultaneously playing samples mixed. After this the resulting signal can be transferred to the ABDAC. This delay doesn’t need to be zero, because the human perception isn’t as fast as the processing in modern microcontrollers. So it is possible to do quite a lot of processing before the delay increases so much it can be perceived by the drummer playing the electronic drums.

This effect was measured with different delays between the trigger-pad hit and sound playback. The drum trigger-pad was stroked, at the same time evaluating whether the delay was perceptible or not. The delay was measured by simultaneously probing the trigger input and audio output with an oscilloscope. The results of this study are shown in Table 8.

Table 8 *Effects of different delays to the human perception in drum trigger playback.*

Delay between drum trigger pad -stroke and audio-playback	Perceived delay
50ms	No perceptible delay, playable
60ms	Barely perceptible delay, playable
80ms	Perceptible delay, playability decreased
90ms	Easily perceptible delay, barely playable
100ms	Notable delay, not playable

The table above shows that delays as high as 60ms can barely be sensed while playing electronic drums. This means that after the drum trigger pad has been hit, we have 60ms at most to do the required processing before the sound associated to that channel has to be played back.

5.2.2 Memory performance

The read performance of the SD memory card -interface was measured by software, using TC-interrupts (Timer/Counter). 16-bit counter was used to measure the amount of TC-interrupt execution times (counter was incremented each time the TC-interrupt occurred). The counter was first set to zero, and then a wav-file was selected (by selecting a specific navigator) and read into a buffer in 2048-byte chunks. After the whole file was read, the value of the counter was saved. Then the counter was set to zero again, another file was selected and read, and the counter value was saved. 10 different wav-files were read, and the read speed was calculated from the total time spent selecting and reading all these files. This way a good estimation of the average read speed of the SD memory card in this application could be obtained.

Maximum obtainable bus frequency for this application was 7.5MHz; when using higher bus frequencies, the reading of the SD-card always failed which was most likely caused by bad layout design of the SD-card interface (in higher frequencies, parasitic qualities start to disturb the signal quality due to long PCB-trails). The bus frequencies, theoretical read speeds and measured read speeds for different memories used are listed in Table 9. 32kB allocation unit size was used with all of the tested SD memory cards. FAT32 file system was used in the 4GB cards and FAT16 file system was used in the 2GB card. Changing the file system to FAT16 of vice versa didn’t affect the obtained read speeds significantly.

Table 9 *Obtained read speeds of the used external memories.*

Memory	Bus frequency	Theoretical read speed	Measured read-speed
SanDisk 2GB micro SD	7.5MHz	3.25MB/s	882-1.0MB/s
SanDisk 4GB micro SDHC class 4	7.5MHz	3.25MB/s	638-713kB/s
Kingston 4GB micro SDHC class 10	7.5MHz	3.25MB/s	425-620kB/s
Flash IC-chip memory (SPI)	24MHz	3MB/s	509kB/s

As can be seen from the table above, the measured SD card read speeds are lower than the maximum theoretical read speed 3.5MB/s at 7.5MHz. Reason for this is the FAT file system handling, which takes time considerably more than transferring contiguously allocated data. There was a high variety in the measured read speeds between different card-types; 882-1.0MB/s read speeds were obtained with SanDisk 2GB micro SD card, with Kingston 4GB micro SDHC class 10 card the obtained read speeds were between 425-620kB/s. There was also variety in the read speeds of the same card; with different wav-files on the card, different read-speeds were obtained. Changing the allocation unit sizes of the memory cards didn’t change the read speeds significantly.

Read performance of the flash IC-chip memory was measured with the same method using SPI bus frequency of 24MHz (highest supported read speed for the flash IC-chip memory used in this project was 33MHz). Theoretically highest possible transfer speed for the used bus frequency is 3MB/s, but the used memory only worked when sending one dummy byte between each read byte. So the read performance of this chip wasn’t optimal, resulting in a read speed of 509kB/s. Read speed of the flash IC-chip memory was constant regardless of the file being read because there was no file system handling slowing down the process.

5.2.3 Obtainable sampling frequencies

Maximum obtainable sampling frequency depends on many factors, such as CPU-frequency, amount of signal processing, and the read speed of the SD memory card. If several audio files are played at the same time the processing time increases considerably. Highest obtainable sampling frequencies for different numbers of simultaneously playing audio files are listed in Table 10. Sampling frequencies are derived from a 12MHz crystal. Cluster size 32kB was used in the tests.

Table 10 *Maximum sampling frequencies for different numbers of simultaneously playing samples and for different memory card -types.*

Audio files being played at the same time	Card type	Maximum sampling frequency
1	Micro SDHC class 10	>46875Hz (stereo/mono)
1	Micro SDHC class 4	>46875Hz (stereo/mono)
1	Micro SD	>46875Hz /stereo/mono)
4	Micro SDHC class 10	23438Hz (stereo) 29297Hz (mono)
4	Micro SDHC class 4	23438Hz (stereo) 39062Hz (mono)
4	Micro SD	23438Hz (stereo) 39062Hz (mono)
5	Micro SDHC class 10	14648Hz (stereo) 23438Hz (mono)
5	Micro SDHC class 4	19531Hz (stereo) 23438Hz (mono)
5	Micro SD	19531Hz (stereo) 23438Hz (mono)

As can be seen from the table above, the highest obtainable sampling frequency depends on the amount of audio files being played at the same time. For example, if 512 audio samples are being sent to the ABDAC with sampling frequency of 46875Hz, the buffers need to be updated every 10.9ms (512*(1/46875Hz)). This means that we have 10.9ms to read the next 512 audio samples from the memory and to do the necessary signal processing. When reading at 1MB/s, it takes about 0.1ms to read a buffer with 512 wav-samples. It takes about 8.3ms to mix five buffers with 512 samples together (with AT32UC3A3256 and processor frequency at 60MHz). So it takes about 9ms to read and mix 5 samples together, which is slightly under the 10.9ms. However, the drum trigger detection -interrupt running at the background reduces the processing time given to the mixing of the buffers, which results as a longer mixing time and a failure to update the ABDAC-buffers in time.

Bottleneck of this system is the speed of the mixing process, not the SD memory card read speed. When only one sample is played, sampling frequencies as high as 46875Hz can be used because no time is consumed by the mixing process. When five samples are being played at the same time, 19531Hz is the highest sampling frequency that can be processed without failures (using stereo samples). With higher CPU-frequencies and a processor designed for digital signal processing, this problem could be overcome.

5.3 Audio quality

With audio applications it is important to keep the supply voltages as noise-free as possible. Reading from an SD memory card draws a lot of current, and this should be considered when designing the power supply filtering. Effects of different noise sources to the audio quality were tested and the results are shown in this chapter.

5.3.1 Noise sources

Reading data from the SD card causes spikes to the supply voltage. These spikes can be seen in Figure 21.

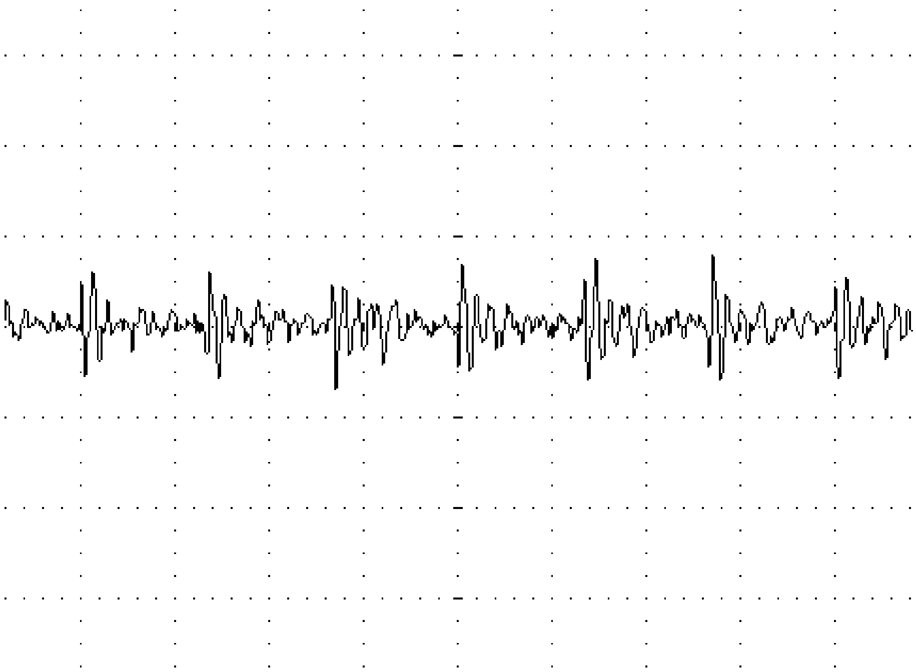

Figure 21 *Noise at the supply voltage when data is read from the SD memory card. 20mV/div, 100ns/div.*

Frequency of the noise is 15MHz and amplitude is approximately 20mV. Frequency is twice the SD bus frequency used. The spike seen in Figure 21 occurs every time clock-signal of the SD memory card interface changes state. The same noise component can be seen also at the audio output (Figure 22).

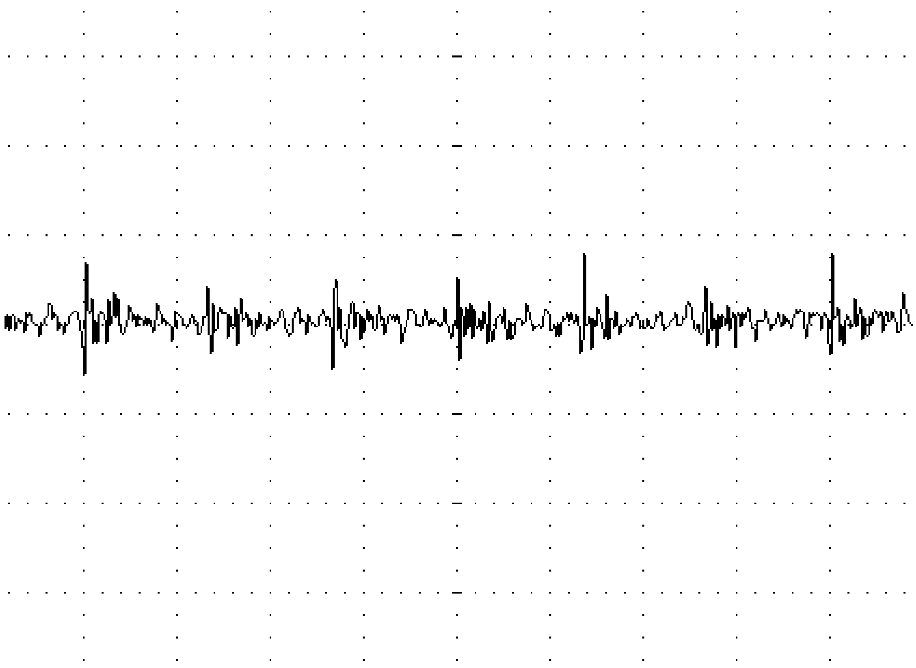

Figure 22 *Noise at the audio output when data is read from the SD memory card. 20mV/div***,** *50ns/div.*

The SD read -noise component is shown also at the audio output, slightly attenuated. This indicates that better power supply filtering is required for the audio amplifier. The audio amplifier is physically close to the SD memory card slot, which also contributes to the noise transfer.

Reading the SPI flash IC-chip memory causes also spikes to the power supply, which can be seen in Figure 23.

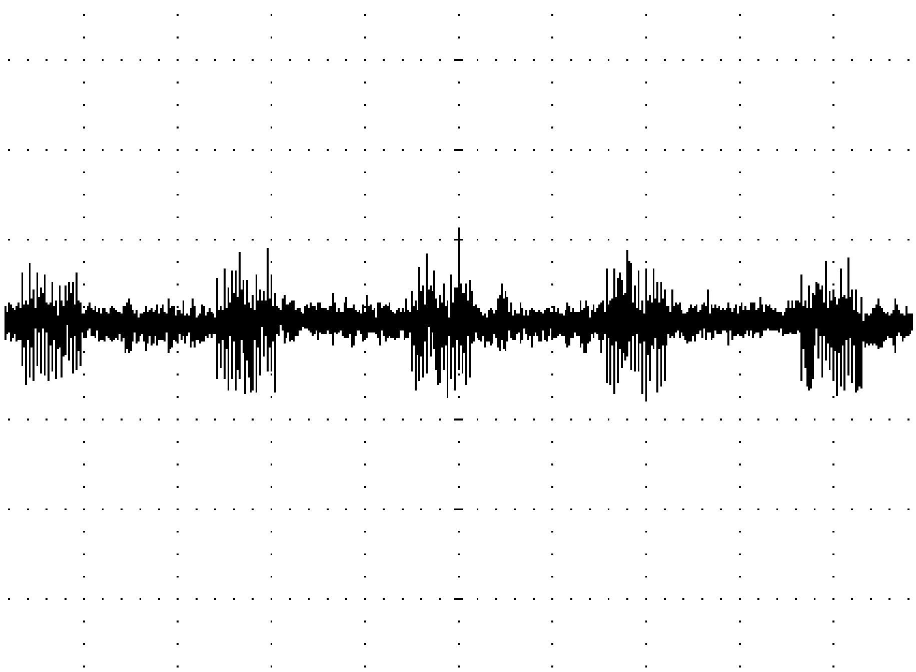

Figure 23 *Noise at the supply-voltage when data is read from the flash IC-chip memory via SPI. 20mV/div***,** *1us/div.*

When the flash IC-chip is read, high-frequency noise can be measured from the supply voltage. Frequency of this noise is 24MHz (same as the SPI bus frequency), and amplitude is approximately 30mV. This noise component is also transferred to the audio output (Figure 24).

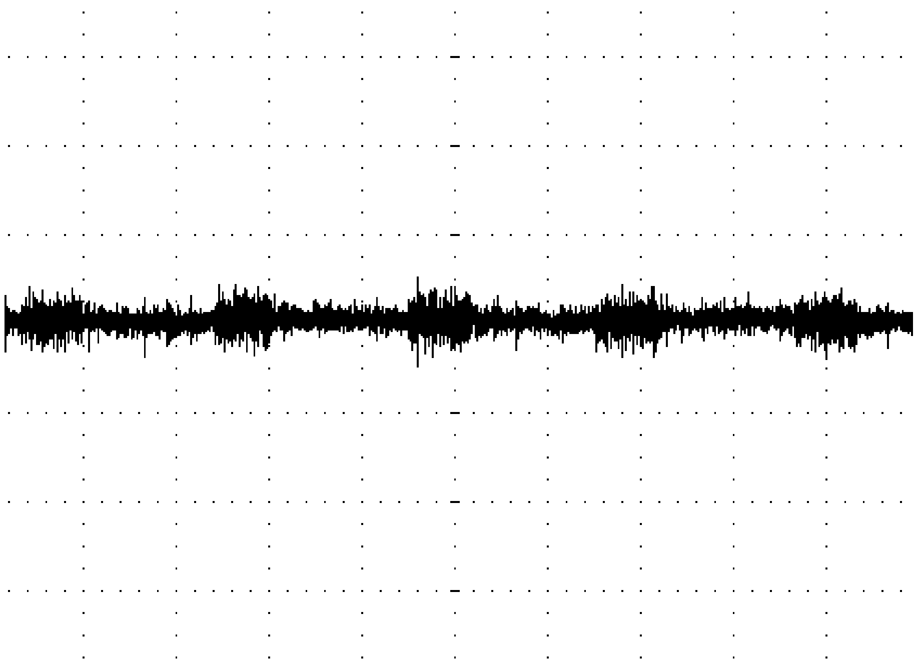

Figure 24 *Noise at the audio output when data is read from the flash IC-chip memory via SPI. 20mV/div*, *1us/div.*

Amplitude of this noise component is approximately 20mV at the audio output. The amplitude is slightly attenuated because there is large distance between the flash IC-chip memory and the audio amplifier, which decreases inductive coupling and coupling via ground plane.

Output of the ABDAC's low-pass filter, when silence is played back, can be seen in Figure 25.

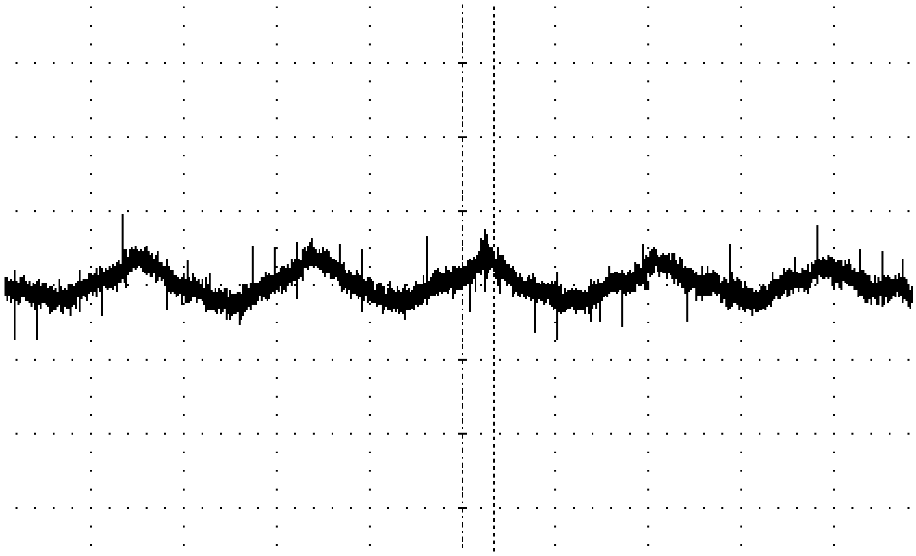

Figure 25 *Signal measured from the output of the RC-low-pass filter of the ABDAC without audio playback. 20mV/div, 2us/div.*

The signal above is unwanted noise coming from the switching of the ABDAC output (ABDAC is outputting 1.65V (zero-level of the audio) with PCM). The amplitude of this noise component is approximately 16mV and frequency is approximately 250kHz. As can be seen, the ABDAC-switching noise isn't totally filtered out by the RC-filter. Because this noise component is fed directly to the input of the audio-amplifier, it is seen also at the audio-output (Figure 26).

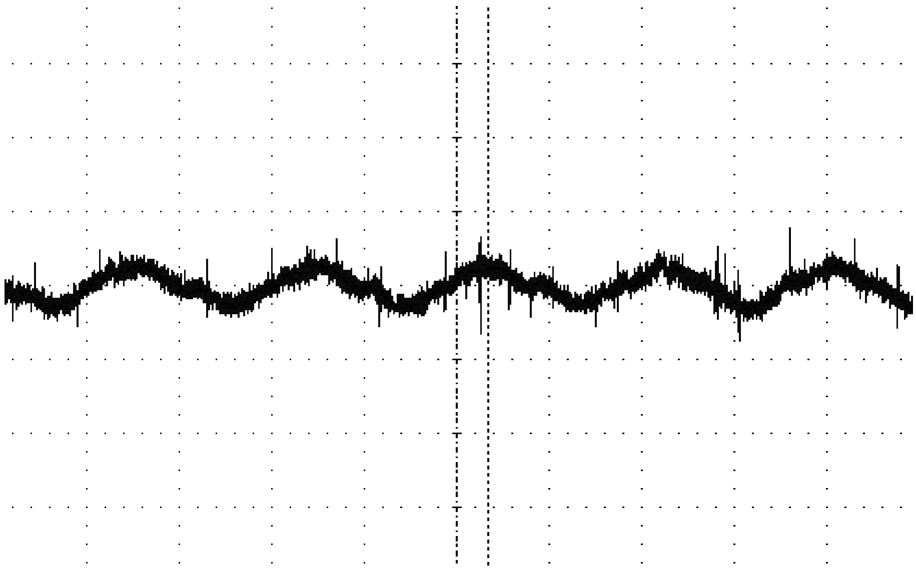

Figure 26 *Signal measured from the audio output without audio-playback. 20mV/div, 2us/div.*

As can be seen in the figure above, the ABDAC switching noise at the audio output is identical to the noise at the output of the low-pass filter. This is expected, because the gain of the audio amplifier is 1.

Reading of the SD memory card and the flash IC chip result in similar noise levels to the audio output, only the frequency changes. The switching of the ABDAC causes slightly smaller-amplitude noise to the audio output. All these noise sources have frequencies above the human hearing range, which makes them inaudible.

5.3.2 Audio quality

The noise showing on the power supply will also show on the output of the audio amplifier of the system. However, the noise sources described in previous chapter (the noise from reading the SD memory card, the ABDAC switching noise) have so high frequencies that they are inaudible by humans. Short noise peaks are usually also inaudible. (Roberts et. al. 1999)

When the device is at the play mode, slight hiss-sound can be heard from the output (using 32Ω headphones). However compared to the volume of the audio samples being played, this hiss sound is really quiet. The device doesn’t have any volume control, meaning that the volume is always at its highest. Lowering the volume for example with external amplifier makes the hiss almost imperceptible.

Better audio quality could be obtained by using the complementary digital outputs DATAn and DATANn of the ABDAC-module of the microcontroller. These outputs could be applied to a differential stage amplifier to increase the signal-to-noise ratio (SNR) and the total harmonic distortion. Class D amplifier approach could also be used to increase the SNR value. (AT32UC3A3/A4 Datasheet)

Audio gets distorted if the buffers going into the ABDAC aren’t updated with new samples in time. This happens when too many samples are mixed at the same time. The maximum sampling frequencies for different number of simultaneously playing samples are listed in Table 10. The maximum sampling frequency for each memory card type should be used because the quality of the audio decreases when the sampling rate is decreased. The sampling rate 23438Hz has a slightly reduced audio quality compared to 44100Hz. The difference can be heard mainly in higher frequencies; rendering a wav file into a lower frequency filters out some of the fast changes in the signal (i.e. the higher frequencies).

6 CONCLUSION

Suitability of a micro SD memory card for storing audio files containing the drum-sound samples in electronic drum applications was studied in this thesis. Electronic drum module was constructed with eight input channels for drum trigger pads and micro SD memory card slot for audio sample storage. 4-bit transfer protocol of the SD memory card was used to read wav-files and the read speeds were tested with different settings. Highest obtainable sample rates were also tested using different numbers of simultaneously playing samples.

Transfer speed of a SD memory card depends on many factors, such as the bus speed mode used, the host clock specifications, the SD memory card speed class, fragmentation of the files being read, and the PCB-layout of the application. In this thesis the read speeds were measured using FAT16 (SanDisk 2GB micro SD memory card) and FAT32 (SanDisk 4GB micro SDHC class 4 and Kingston 4GB micro SDHC class 10) file systems with 32kB cluster size. In electronic drum applications several audio files may have to be played back simultaneously. Because of this, the changing of the wav-file to be read was taken into consideration when measuring the read speeds. The highest obtained read speed was 1.0MB/s with SanDisk 2GB micro SD memory card (bus speed was 7.5MHz with 4-bit transfer mode). When higher bus speeds were used, data corruption occurred which was due to too long lines between the SD-card slot and the microcontroller. The higher capacity cards did not perform as well as the SanDisk 2.0 GB micro SD memory card.

Mixing of audio files is a demanding task for a processor. The more audio files need simultaneous mixing, the more time is consumed. With just one file playing, no mixing is needed and high sampling frequencies can easily be used. Problems occur when many files need to be mixed simultaneously. If mixing is still in progress when the samples should be transferred to the ABDAC, audio signal distortion occurs. In this thesis, sampling frequencies as high as or above CD audio quality (>44100Hz) could only be used when playing one sample at a time. With four simultaneously playing samples, 23438Hz was the highest possible sampling frequency without audio distortion.

Based on the results obtained in this thesis, SD memory card could be used as audio file storage in electronic drum applications. It is easy to use and gives limitless expansion possibilities for the audio storage. It is also affordable and widely used memory card type. When designing an electronic drum module, special care has to be taken when choosing the method for mixing the audio files. A digital signal processor designed for mixing audio-files could be used to ensure short processing times. With the combination of an efficient digital signal processor and a well-designed SD memory card interface, the needs of a demanding electronic drummer could be met.

SOURCES

AT32UC3A3/A4 Datasheet. 2011. Atmel Corporation. 1011p.

AVR114: Using the ATMEL File System management for AT32UC3x, AT90USBx and ATmega32U4. 2008. Atmel Corporation. 22p.

Boothroyd, G., Dewhurst, P., Knight, w. 2001. Product Design for Manufacture and Assembly, 2nd Edition. Taylor & Francis e-Library, Marcel Dekker Inc. 721p.

CF+ and CompactFlash Specification Revision 2.0. 2003. CompactFlash Association. 125p.

CF Cards [WWW]. Compact Flash Association. [referred 02.04.2013]. Available: http://compactflash.org/product-types/cf-cards/

ChaN. 2013. Ho to Use MMC/SDC [WWW]. [referred 30.4.2013]. Available: http://elm-chan.org/docs/mmc/mmc_e.html

Default cluster size for NTFS, FAT, and exFAT [WWW]. Microsoft Corporation. Article ID: 140365. [referred 3.5.2013]. Available: http://support.microsoft.com/kb/140365

DM 10 [WWW]. Alesis Corporation. [referred 02.04.2013]. Available: www.alesis.com/dm10

DrumIt Five – Electronic Drum System [WWW]. 2Box AB. [referred 02.04.2013]. Available: http://www.2box.se/World/pages/products/

DTXTREME IIS [WWW]. Yamaha Corporation. [referred 02.04.2013]. Available: http://usa.yamaha.com/product_archive/drums/dtxtreme_iis/?mode=model

DTX950K [WWW]. Yamaha Corporation. [referred 02.04.2013]. Available: usa.yamaha.com/products/musical-instruments/drums/el-drumkit/dtx900/dtx950k/

Grimnes, S., Martinsen, Ø. Bioimpedance and Bioelectricity Basics. 2nd ed. Oslo 2000, Elsevier Ltd. 471p.

Jahed, F. M. 2007. Semiconductor Storage Devices in Computing and Consumer Applications. In: Oklobdzija, V. G. Digital Design and Fabrication. Second edition. Boca Raton, Florida. CRC Press. Chapter 6.

Kuphaldt, T.R. 2000. Lessons In Electric Circuits Volume II – AC. 321p.

Multimedia Programming Interface and Data Specifications 1.0. 1991. Joint design by IBM Corporation and Microsoft Corporation.

Roberts, H.C., Liu, D. H. F. 1999. Noise Pollution. In: Liu, D. H. F., Lipták B. G. Environmental Engineers' Handbook. Second Edition. Boca Raton, Florida. CRC Press. Chapter 6.

SamplePad Multi-Pad Sample Instrument [WWW]. Alesis Corporation. [referred 02.04.2013]. Available: http://www.alesis.com/samplepad

SD Specifications Part 1 Physical Layer Simplified Specification (Version 4.10). 2013. SD Card Association. 186 p.

SD Specifications Part 2 SD Host Controller Simplified Specification (Version 3.00). 2011. SD Card Association. 146 p.

Smith, R.L. 2004. Sensors and Transducers. In: Dorf, R.C. (ed.). The Engineering Handbook, Second Edition. California, USA, CRC Press. Chapter 151.

Tanenbaum, A. S. 2001. Modern Operating Systems. Second edition. New Jersey, USA. Prentice-Hall Inc. 976 p.

TD-30 Drum Sound Module [WWW]. Roland Corporation. [referred 02.04.2013]. Available: http://www.rolandus.com/products/details/1204/488

Toth, V. T. 2000. Mixing digital audio [WWW]. [referred 22.4.2013] Available: http://www.vttoth.com/CMS/index.php/technical-notes/68

Tyson, M. 2011. A quick-and-dirty audio sample mixing technique to avoid clipping [WWW]. A Tasty Pixel. [referred 15.04.2013]. Available: http://atastypixel.com/blog/how-to-mix-audio-samples-properly-on-ios/

Universal Serial Bus 3.0 Specification, Revision 1.0. 2011. Hewlett-Packard Company, Intel Corporation, Microsoft Corporation, NEC Corporation, ST-Ericsson, Texas Instruments.

V-Drums [WWW]. Roland Corporation. [referred 02.04.2013]. Available: http://www.roland.com/V-Drums/