A modular platform for a subtractive synthesizer on Arduino Due

Rodolfo Pedó Pirotti[1], Marcelo Johann[1], Marcelo Pimenta[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

rodolfo.p.pirotti@gmail.com, {johann, mpimenta}@inf.ufrgs.br

Abstract. *In this paper we present an initial design to implement a platform of a subtractive synthesizer by using the Arduino Due board. We intend to develop a modular system, which can be used with a real musical keybed or as a MIDI device, to play and generate musical notes by using subtractive synthesis, including different types of waveforms, envelope generators, filters and configurations. To achieve a high quality output signal, we intend to attach simple but effective external DACs and analog output stages. By reaching this goal, we will be able to provide a powerful low-cost platform for subtractive synthesis.*

1. Introduction

Subtractive synthesis is a method for sound generation that became very popular in the 60's and 70's, with the original implementation being done by using electronic analog circuits. Through the years, "Virtual Analog" became available on digital synthesizers to simulate the sound resulting from the old subtractive analog synthesizers, but most of these machines were and are still expensive, obtained by significant design efforts, making use of the most up-to-date technologies [Urban 2002] [Wiffen 1997].

There are some open and Do-It-Yourself (DIY) projects which implemented or investigated subtractive synthesis on low cost platforms, such as Mozzi [Barrass 2015], Scalable Polyphony-MIDI Synthesizer [Huovilainen 2010], AVRSynth [Biddulph et al 2015], and others. But we intend to combine a high quality audio output on a wide available and low cost popular platform (in this case, Arduino Due), which we did not find in the above examples.

The Arduino Due is a widely available low cost board with a 32-bit ARM processor running at 84 MHz, introduced in 2012. Its usage nowadays is becoming very popular for applications that require interaction with the real world and people, including audio applications [Johann 2015].

1.1 Objective

With the goal to create a flexible, modular and powerful platform for subtractive synthesis, achieving a high quality output signal on a low cost and wide available platform, we started the design and development of a subtractive synthesizer using the Arduino Due board.

In this paper, we present the initial design and development, and our goals for the finished platform. This is an ongoing project for which only initial tests and code is already completed.

2. Subtractive Synthesis

Subtractive synthesis is a type of synthesis based on an input signal containing a lot of harmonic content. Examples are the square and sawtooth waves. On this signal, a filter is applied, which selectively removes some of the harmonic content. And so, we have the name of "subtractive synthesis".

On a subtractive synthesizer, the first stage is an oscillator which generates the waveform at the desired frequency. Usually, this oscillator is called VCO (voltage controlled oscillator). On the output of the oscillator, a filter is connected. This can be a low-pass filter, high-pass filter, band-pass filter or a band-stop filter. The filter is usually called VCF (voltage controlled filter). A low-frequency oscillator (LFO), connected to VCO, VCF or Amplifier blocks can also be used with the goal to create amplitude and frequency modulation on the output signal [Urban 2002].

3. Hardware Architecture

To implement a powerful platform for subtractive synthesis, special care is needed on all hardware blocks. Real time input controls are needed to modify the sound parameters on real time. In the output stage, a high quality digital to analog converter (DAC) was chosen, to achieve a high quality audio output signal.

For musical note input, the hardware can be used with two different options: with a musical keybed, or as a MIDI host. Figure 1 shows a block diagram of the hardware prototype.

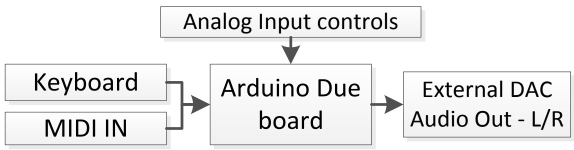

Figure 1. Hardware prototype blocks

The Audio Out block is implemented with an external DAC to achieve a higher quality than using Arduino's internal DAC. Like in [Johann 2015] a TDA1543 16-bit DAC is connected to Arduino board SAM3X8E processor, which has an I2C interface.

4. Software modules and design

The software design must use good software engineering approaches, such as low coupling, high cohesion and well defined interfaces, in order to achieve the goal of a modular platform, which includes the possibility of easily change some blocks of the synthesizer, such as the filter type, effect, etc., but keeping the code fast enough to run properly on Arduino Due processor.

4.1 Waveform sample and oscillator

This block is responsible for the generation of the desired audio signal, at the desired frequency, on the digital domain. It uses a numeric controller oscillator (NCO) and a phase to amplitude converter (PAC) to generate the output data.

The NCO uses a phase accumulator to increment the phase of a waveform during time. The phase value is then used to index a second block, the PAC, which is implemented with a lookup table containing the waveform sample. This approach allows us to easily change the waveform sample, by changing the lookup table. A fixed point approach was chosen instead of a floating point approach for faster processing.

4.2 Envelope generator

The envelope generator module is responsible for generating an ADSR curve (attack-decay-sustain-release), which can be applied to any stage of the synthesizer. As an independent module, the envelope generator (EG) produces and output value between 0 and 1 (using fixed point format), which should be multiplied by the output of the block. For an example, if we want to apply EG on the output of the filter signal, the output value of EG is multiplied by the filter output data. As a separate module, more than one EG can be instantiated and applied to different software blocks.

4.3 Filter

The filter block consists of a low-pass filter, with cutoff frequency and resonance as input parameters, which can be changed in real time using input analog controls.

4.4 Effects

On the final stage of the synthesizer, an effect block is implemented. The first prototype has only a delay with feedback. Other types of effects will be analyzed in the future.

4.5 Other features

A low frequency oscillator (LFO) is implemented, which can be connected to modulate the frequency of the main oscillator (adding a vibrato); a soft pitch slide when changing notes (portamento) is implemented and can be turned on, off and its time configured; the selection of the note priority (last, highest, lowest) and trigger mode, to behave like old classic analog synthesizers [Reid 2000], is also available.

4.6 Module connections

All blocks are implemented as independent modules. This allows us to continuously develop different modules for each stage, changing and adding different characteristics to our synthesizer. Figure 2 shows a basic module block diagram.

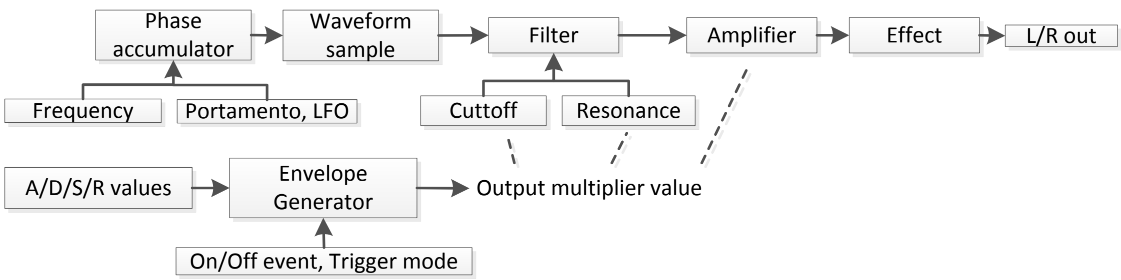

Figure 2. Software module blocks

5. Next Steps

As of current development, an aliasing is easily noticeable on higher frequency notes, requiring us to implement an antialiasing method for waveform generation, such as on [Pekonen et al 2008] and [Valimaki et al 2007]. Faster algorithm for the note priority selection feature (sort and find algorithm) and the low pass filter is also required.

A higher order low-pass filter with a higher resonance is desired. We intend to research mathematical models of low pass filters used on classic analog synthesizers and implement on our platform. To improve control capabilities, when used as a MIDI host it is possible to implement the decoding of some MIDI messages to change internal parameters, as an option to increase real time control with no need to add external electronic components to the board.

As of sound quality, a deep analysis of audio output using instruments such as oscilloscope and spectrum analyzer, plus some ABX testing is also planned [Johann et al 2013].

6. Conclusions

In this paper we presented an initial design and prototype of a modular, powerful and low-cost subtractive synthesizer using an Arduino Due. Initial prototype tests have demonstrated that it is feasible to implement this type of synthesizer using a low-cost Arduino Due board.

We intend to continue the development of this platform, including other features, such as: different waveform sample selection, different filter selection, more effect types, and also the possibility for different block connections between modules, just like old analog synthesizer used to work.

7. References

Barrass, Tim (2015) “Mozzi sound synthesis library for Arduino”. <http://sensorium.github.io/Mozzi/>

Biddulph, L., Ziembicki, J. (2015) “AVRSynth – monophonic virtual analogue synth”. < http://www.elby-designs.com/contents/en-us/d5.html>

Huovilainen, Antti (2010) “Design of a Scalable Polyphony-MIDI Synthesizer for a Low Cost DSP”. Master’s Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology. Aalto University School of Science and Technology. Espoo, Finland.

Johann, M., Folle, L., Gomes, W., Cordoni, R. (2013) “Estabelecendo uma Metodologia de Avaliação Subjetiva de Qualidade de Áudio”. IV Ubimus. Porto Alegre, Brazil.

Johann, Marcelo. (2015). “An additive synthesis organ with full polyphony on Arduino Due”. VI Ubimus. Växjö, Linnaeus University, Sweden.

Pekonen, J., Valimaki, V. (2008) “Filter-based alias reduction for digital classical waveform synthesis”. IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV. 2008.

Reid, Gordon (2000). “Synth Secrets, Part 18: Priorities & Triggers”. <http://www.soundonsound.com/sos/oct00/articles/synthsec.htm>

Urban, Ondrej. (2002). “Short overview of methods of sound synthesis.” The 32nd International Acoustical Conference - EAA Symposium "Acoustics Banská Štiavnica 2002”. September, Slovakia.

Wiffen, Paul. (1997) “Synth School, Part 1: Analogue Oscillators, Filters & LFOs”, <http://www.soundonsound.com/sos/1997_articles/jun97/synthschool1.html.>

Valimaki, V., Huovilainen, A. (2007) “Antialiasing Oscillators in Subtractive Synthesis”. Signal Processing Magazine, IEEE (Volume:24 , Issue: 2).