İstanbul Bilgi University
LAUREATE INTERNATIONAL UNIVERSITIES

FACULTY OF ENGINEERING AND NATURAL SCIENCES

ANALOG SYNTHESIZER

by

117202074 Metehan Malakçı
115202025 Berkay Erdoğan

Supervised
by

Asst. Prof. Baykal Sarıoğlu
Asst. Prof. Okan Zafer Batur

A report submitted for EEEN 4992 senior design project class
in partial fulfillment of the requirements for the degree of
Bachelor of Science
(Department of Electrical and Electronics Engineering)
in Istanbul Bilgi University

June 15th, 2021

ACKNOWLEDGMENTS

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude to our esteemed advisors, Asst. Prof. Baykal Sarıoğlu and Asst. Prof. Okan Zafer Batur, who guided and supported us with their knowledge and experience during the graduation project.

We would like to thank our families who have never left us alone with their financial and moral support throughout our project.

We also would also like to thank all our teachers in Istanbul Bilgi University Electrical and Electronics Engineering Department, who have supported us with their guidance throughout our undergraduate education.

ABSTRACT

As electricity became more widely available, the early 20th century saw the invention of electronic musical instruments. The rapid spread of electronic music and increasing demand for unique and innovative sounds led to innovation of synthesizers. In the mid 1960s, when Bob Moog introduced the Moog modular synthesizer, it was the first newly popularized instrument invented since the saxophone. Unlike traditional instruments, which haven't changed much in hundreds of years, electronic instruments like synthesizers have evolved almost continuously through recent history.

Grammy Award Winning Masterpieces such as Switched On Bach (1968) which is the remake of classical music pieces created with Moog modules by Walter Wendy Carlos contributed to making musicians understand the infinite possibilities offered by the new instruments. In a flourishing of experiments, the synthesizers soon burst into all musical genres especially 1960s psychedelic rock and 1970s progressive rock and 1980s disco music.

This report presents the construction stages of the sytnhesizer that we built to achieve our own unique sounds. In the report, the structure and properties of the materials we use while building our synthesizer, and the software we use in the PCB circuit design phase will be presented. The report also provides information such as the brief history of analog synthesizers, some differences between analog and digital synthesizers, the transition process from analog to digital synthesizers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

Appendix

1. INTRODUCTION

1.1 General Information

As electricity became more widely available, the early 20th century saw the invention of electronic musical instruments. The use of electricity for musical purposes actually dates to 1759, when the Clavecin Electrique, or Electric Harpsichord, was made by the Parisian Jean-Baptiste de La Borde. Under keyboard control, its static electricity-loaded clappers hit bells, much like a carillon, except that carillons are purely mechanical.1 At the same time when Moog developing his synthesizer in a small factory at New York, the muse of invention was affected by a different person named Don Buchla. Both Moog’s and Buchla’s synthesizers used a varying voltage to control the sound-shaping features of the components of the synths. Buchla, Moog and every other synth manufacturer after them put efforts to produce the modern-day synthesizer.

This project seeks to be a short and easy-to - understand approach to the world of those captivating instruments, named 'synthesizers,' to make a general presentation of their typology and structure, and attempt to explain several of the important aspects underneath their extraordinary flexibility, versatility and power. Our project is to produce an analog synthesizer, the cornerstone of today's electronic music, with the lowest possible budget, and to obtain sounds at different frequencies for the electronic music works that we want to make. Today, although some of them will comment that the use of analog synth is an old school progressive rock music habit, we believe that this instrument can capture some different sounds that we cannot get through digital synth applications, and we can break new ground in electronic music as in every period

1.2.Methodology

In the scope of our search for various resources and our meetings with our professors who will supervise us on this project, Labolida Synthesizer’s “Tiny Synth” project was our main guide. Also Ray Wilson's book “Make: Analog Synthesizer” and the web site "Music From Outer Space", also owned by the author, used as an important source in our project. While building this instrument, which has infinite examples in the Internet and various books, we considered both the common interests of us who made the project, the expectations of our supervisors, and the fact that it is a project that can be developed.

1.3. Goals

Our primary goals in this project is to build an analog synthesizer, the cornerstone of the world of electronic music, at a lower cost than the serial production devices in our university, to obtain different sounds that we cannot obtain from digital synthesizers, as we briefly mentioned in the introduction. Another goal of ours is to specialize in this instrument while building each circuit from scratch during the construction phase of this instrument, combining each circuit element correctly, and having fun with the relaxing spirit of music while working throughout our project term. Considering today’s exchange rates and customs duties, it seems impossible to buy an instrument with such a history, considering that the lowest models start at 5000TL and the prices of keyboard Moog synths rise to 15000TL. We believe that as our group members, the young musician candidates, music lovers who has idled Keith Emerson from 70’s progressive rock music or Giorgio Moroder, the virtuoso of EDM music of the same period, should have access to this instrument. In that sense, our main goal is to build this instrument, which has inspired people who have devoted themselves to different musical genres in these different generations, with the possible lowest budget.

2. BRIEF HISTORY OF SYNTHESIZERS

2.1 Defining the Instrument

What is the analog synthesizer? We should answer this question individually. Firstly, what is a synthesizer? Basically, it is an electronic device for creating and modifying sounds. Types of synthesizers differ from traditional analog to digitally controlled, pure digital and, more lately, software emulation. In order to explain the synthesizer, we should first explain briefly regarding sound and how traditional instruments work.

Sound is made up of vibrations which directly cause fluctuations in air pressure. Our eardrums pick up these rapid fluctuations when these vibrations, rarefaction and air compression fluctuate between a frequency of 20 Hz and 20 kHz and we perceive them as sound. When you strum a guitar or hit a piano key, a string quickly vibrates to produce a sound. The faster it vibrates, the higher the pitch . The harder we strike or strum the string, the louder the sound appears. If we orient that string to another framework made of a material that can pick up these vibrations, such as a guitar body or piano soundboard, it amplifies the sound by vibrating with the string and, like a drum head, introduces its own tonal qualities from the wood , metal or membrane.

With the emergence of electronics and radio around the turn of the last century, there were some developments in amplifying and transmitting sound via transducers. A transducer is a device for turning mechanical energy into electronic signals and back again, microphones and speakers. A guitar pickup is a transducer, just like a needle for a record player and a microphone. A synthesizer substitutes an electronic sound source for the first transducer, that "synthesizes" the vibrations that a guitar string or other source would have generated.2 This new source of sound was named an "oscillator," which is an electronic sound synthesizer 's essential first key component.

After the brief explanation of synthesizer, let's complete our first question. What is an analog synthesizer? Analog synthesizer is a synthesizer which electronically generates sound using analog circuits and analog signals. Analog synthesizers are constructed using Op-amp integrated circuits, potentiometers, and variable resistors. They also use low and high pass filters to reshape the sound.

2.2. Continuous Evolution of Synthesizers

Tinkerers and electronics pioneers have made a huge impact on modern music, creating complex sounds out of simple electronic circuits. Unlike traditional instruments, which haven't changed much in hundreds of years, electronic instruments like synthesizers have evolved almost continuously through recent history.

Early synths

The seeds of the modern electronic synthesizer were planted at the turn of the 19th century when an American inventor called Thaddeus Cahill applied for a patent to protect his idea for a Dynamophone. It was played into the public telephone network because there was no such thing as a loudspeaker or public address system. After this attempt, Russian inventor, Leon Theremin created the Theremin in 1919, it was much smaller than dynamophone but famously difficult to play, but still its eerie sound has found its way into dozens of horror movie soundtracks.

The term synthesizer was first used to describe an instrument in 1956, with the RCA Electronic Music Synthesizer Mark I. It was developed by Americans Harry F. Olson and Herbert Belar and it generated sound with 12 tuning forks that were stimulated electromagnetically.

Moog synths and beyond

One of the first synthesizers that would be recognised as such by modern musicians was created in 1964 after Bob Moog met Herbert Deutsch, and the former was inspired to create a voltage-controlled oscillator and amplifier module with a keyboard – but it wasn't until 1967 that Mr Moog called his diverse modular system a 'synthesizer'.

After releasing a few successful models, Moog realised that there was a problem. In 1969, he had 42 employees working round the clock to create two or three complete modular systems each week for studios. He concluded that these synths were too big, complex and expensive to be sold directly to the public through music retailers, so he set about trying to create a compact, portable and affordable synth that was easier to use and had close-to universal appeal. After a few early prototypes, The Minimoog Model D was released in the summer of 1970. The first fully integrated synthesizer, the Minimoog represents a crucial development in electronic music.

Digital's day

After changing pop music in the 1960s and driving disco in the 1970s, synthesizers became more widely available in the 1980s. But in many ways, these were entirely different instruments. They were digital synthesizers. Unlike analogue synthesizers, which produce music using real analogue circuitry, digital synthesizers emulate analogue sounds with digital signal processing techniques.

Tribute to the Past

Digital synthesizers are still incredibly popular. And the technology has improved considerably over the last few decades. But analogue synthesizers have gone through something of a revival in recent years. In the same way that vinyl was rescued from the depths of obscurity over the last ten years or so, throwback analogue synths have got back in the groove of things too.

Famous German composer Hans Zimmer uses sounds from analogue synthesizers in many of his famous movie soundtracks including The Dark Knight Rises and Blade Runner 2049.

In an interview, Hans Zimmer explained the difference with using an analogue synthesizer. "It's so the opposite of how we make music in the modern world with a mouse on a screen," Zimmer said.

2.3. Some Remarkable Models

Once the brief history of synthesizers and synthesis technologies have been introduced, we'll look in some detail at some of the most remarkable synthesizer models.

The Minimoog (Moog Music, 1970)

Fig. 1. The Moog Minimoog, model "D". Image obtained from Wikipedia

The Minimoog was a very innovative synthesizer in many aspects, besides being one of the very first "commercial successes" in the synth world, with a total production (1970-1982) of over 12,00 units, which was very relevant considering the age and the situation of the synthesizer market at that time.

The Prophet 5 (Sequential Circuits, 1978)

Fig. 2. The Sequential Circuits Prophet 5. Image obtained from Wikipedia

This synthesizer stands out for several reasons; among them, that of being one of the first polyphonic synthesizer models and, being the first synth with programmable memory, what was an important step in the history of those instruments. Although the sound generation technology is still analog, this board is digitally controlled by a microprocessor.

The DX-7 (Yamaha, 1983)

Fig. 3. The Yamaha DX-7. Image obtained from Wikipedia

The Yamaha DX-7 is, as all the other models included here, a synthesizer that stands out for different reasons. Among them, we can emphasize that it’s one on the first totally digital synths, that it has a (then) ample polyphony of 16 voices, that it’s based on an advanced FM synthesis model as its only sound source, that it was one of the first models that implemented MIDI (the –then- new standard of communication between electronic

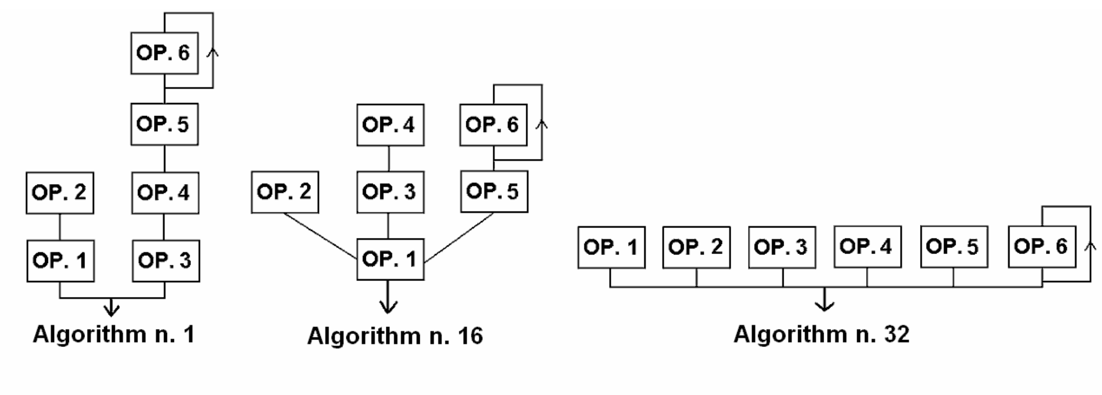


Fig. 4. Some of the algorithms of the Yamaha DX-7

3. THEORETICAL INFRASTRUCTURE

3.1. Kinds of Synthesizers

Depending on some of their features, it's possible to classify synthesizers in diverse categories.

Monophonic and polyphonic synthesizers.

A synthesizer is monophonic if it's only able to reproduce one single voice (sound) at once, whereas it's polyphonic if it can play two or more voices or notes at once. Generally, the polyphonic capabilities of synthesizers have increased with the evolution of technology.

Monotimbral and multitimbral synthesizers

The classification responds to the ability or inability or reproducing different timbres at once (such as a bass sound and a strings ensemble sound). This classification is relatively independent from the former one, because although a monophonic synthesizer can only be monotimbral, there are both polyphonic and monotimbral synths and polyphonic and multitimbral ones.

Analog and digital synthesizers

All synthesizers are electronic instruments, but some of them use analog electronic components (electric currents or voltages) to generate the sound, whereas others use digital technology (the whole process is done using numerical sequences, which at the end are converted into an analog signal by an electronic circuit called DAC – Digital-to-Analog Converter -, which is the signal that is sent to the amplifier and then to the speakers. There are also some hybrid models, most of them from late 70s to mid 80s, in which some components are analog but they are digitally controlled by a microprocessor.

3.2. Types of Synthesis

3.2.1. Subtractive synthesis

Nearly all analog synthesizers and a great number of digital ones are included in this category. The main idea is that the sound is formed through the subtraction or elimination of a part of the harmonics from the main sound generator. Using an analogy, it would be like the process used by a sculptor who, from a marble bock, removes part of the material in order to shape it in the desired way.

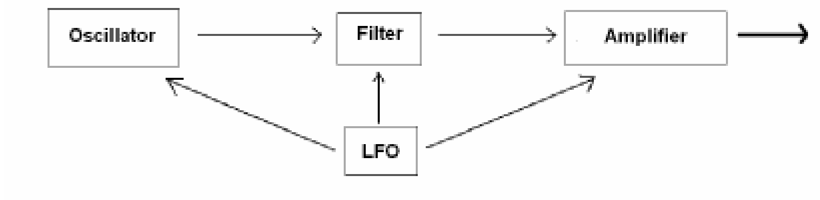


Fig.5. Basic model of subtractive synthesis

The Oscillator

The oscillator is an electronic circuit that generates a waveform in the desired frequency or pitch. In acoustic instruments, the acoustic signal can be generated by an air current by the friction or plucking of a string or by a physical hit on a surface. In synthesizers, that task is carried out by the oscillator. In analog synths, they are usually called VCOs (voltage-controlled oscillators), whereas in digital ones they are usually called DCOs (digitally-controlled oscillators). Usually, those oscillators are able to generate only a few different basic and simple waveforms.

Some of the most common waveforms that most synths can generate shown below.

1.Sinus

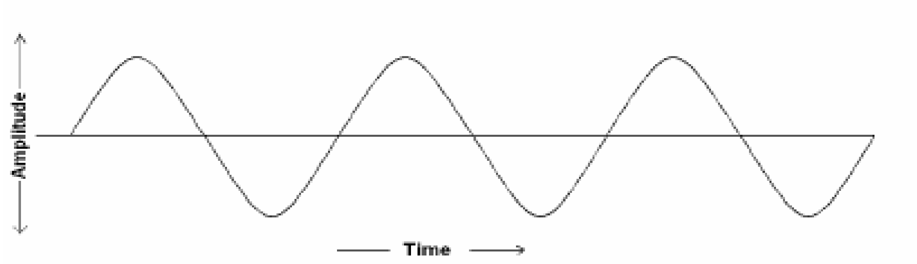


Fig. 6. Graph of a sinus waveform

2. Triangle

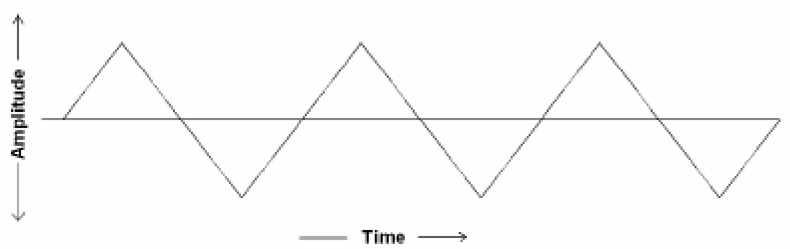


Fig. 7. Graph of a triangle waveform

3. Sawtooth

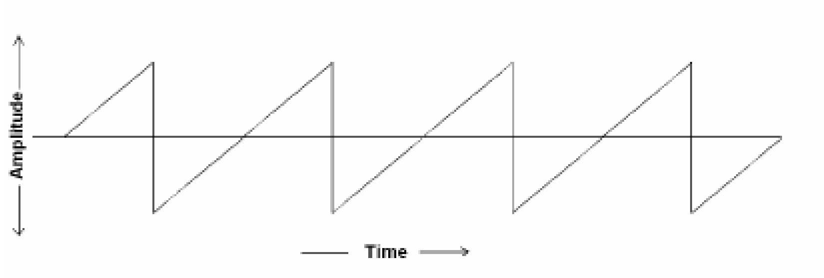


Fig. 8. Graph of a sawtooth waveform

4. Square

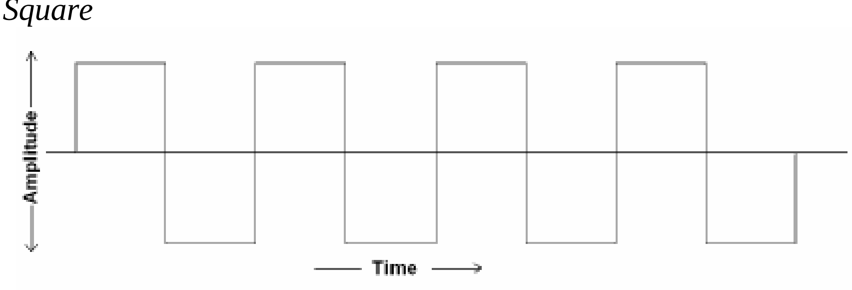


Fig. 9. Graph of a square waveform

5. Pulse

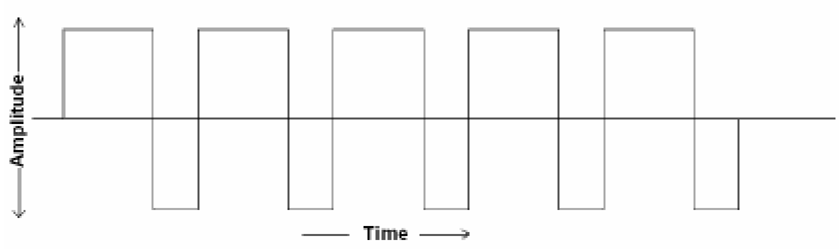


Fig.10. Graph of a pulse waveform

6. Noise

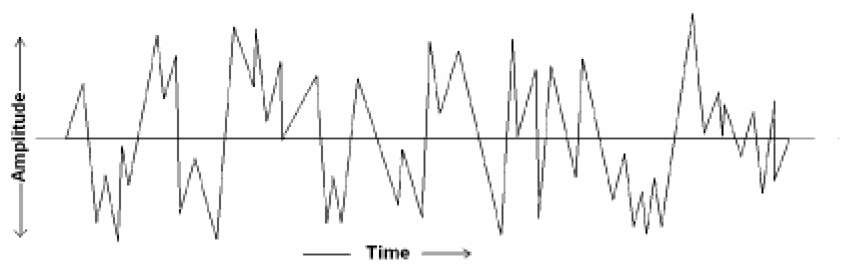


Fig.11. Graph of a noise waveform

The Filter

The filter or VCF (voltage-controlled filter) is a fundamental piece in subtractive synthesis synths, and actually they are the main reason why this synthesis model is called "subtractive", given that the main function of filters is to lower or to cut (subtract) the amplitude of certain frequencies, modifying the harmonics of the original waveform and changing the timbre.

There are several different kinds of filters, with different characteristics. The most common one, which is used in nearly all synthesizer models, is the low-pass filter. This filter attenuates all frequencies that are higher to the cutoff frequency, determined by the user, leaving, on the other hand, lower frequencies intact. The following pictures show the effect of the filter.

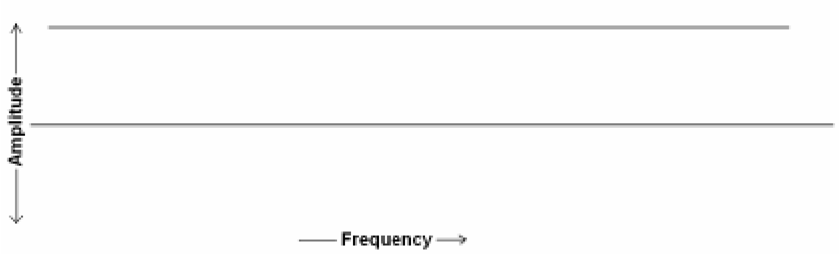


Fig. 12. Original non-filtered waveform

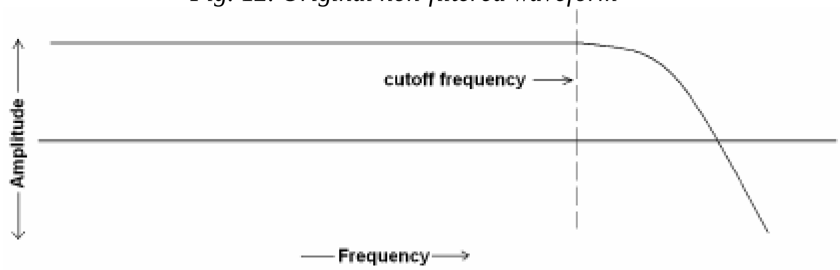


Fig. 13. Action of a low-pass filter

Less common but also quite usual among synthesizers. Among the more common ones, there's the high-pass filter, which attenuates frequencies below the cutoff frequency, leaving the higher ones intact; the band-pass filter, which attenuates both higher and lower frequencies, leaving only the ones close to the cutoff frequency; and the notch filter, which attenuates only the frequencies close to the cutoff frequency.

Also, both the band-pass and the notch filter can be obtained by means of a combination of a low-pass and a high-pass filter.

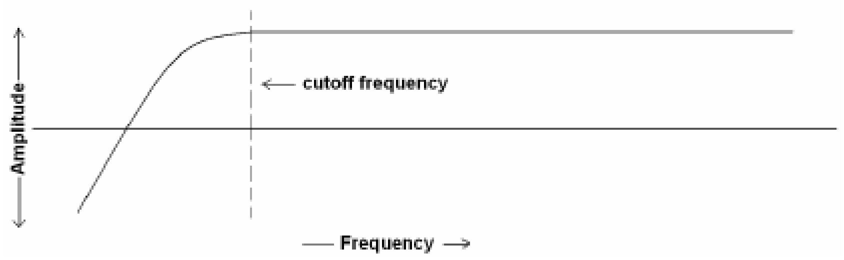


Fig. 14. Action of a high-pass filter

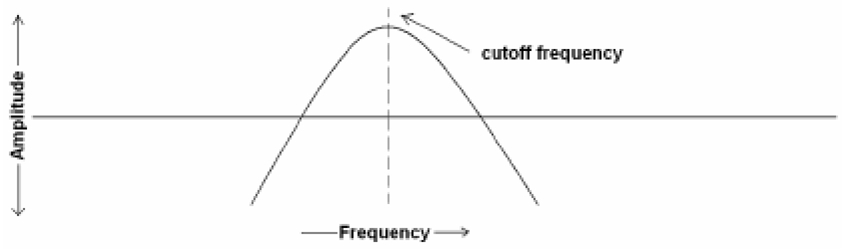


Fig. 15. Action of a band-pass filter

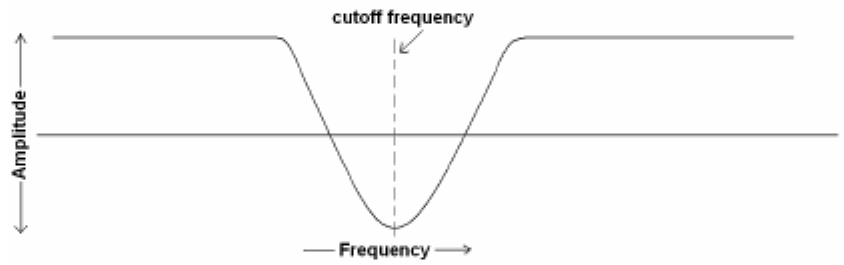


Fig. 16. Action of a notch filter

The amplifier or VCA (voltage-controlled amplifier), is, at its name suggests, a circuit that controls the amplitude (volume or intensity) of the signal from the oscillator and modified by the filter. In order to allow greater flexibility and expression, this amplification is not linear or static, but it can be dynamically controlled through an adjustable envelope. The most common structure of the VCA envelope is that of four segments or ADSR (attack, decay, sustain and release)

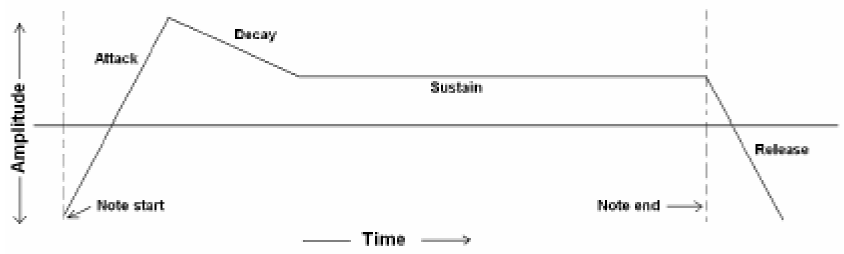


Fig. 17. An ADSR envelope

The Low Frequency Oscillator

The LFO is, as its name suggests, an oscillator, which, as such, produces certain waveforms like triangle, sawtooth or pulse, but at a very low frequencies usually under 20 Hz. The reason for this is that it's not aimed to generate sound, but instead to interact cyclically with other elements of the synthesis, such as the oscillator, the filter or the amplifier. Usually, it is routable and the results will vary depending on the LFO frequency, the amplitude and, above all, the element that it affects. Hence, if it's used to control the main oscillator, it will be useful to create a vibrato effect when the LFO waveform is a sinus or a triangle, or to create an alternance between two different notes.

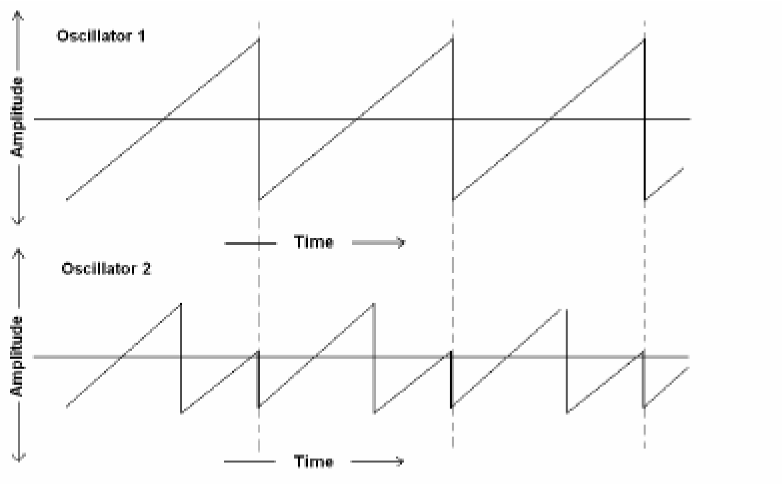


Fig. 18. Waveforms of two synchronized oscillators

Ring modulation consists in the combination of two waveforms but not in a simple mixing, but in a "modulation" of one waveform by the other, giving rise to a new waveform which is different from the two original ones, and which is very rich in harmonics.

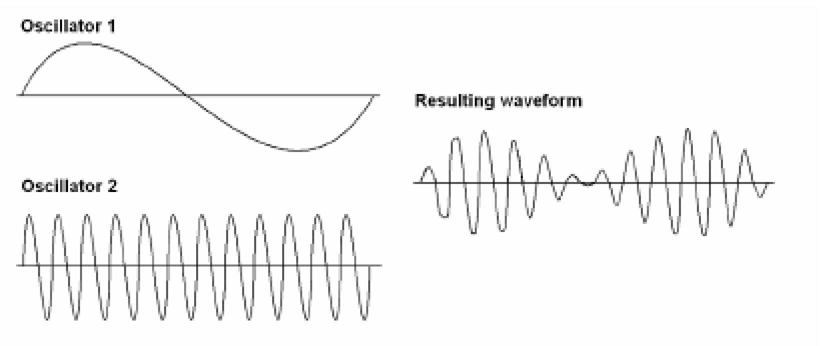


Fig. 19. Ring modulation of two sinus waveforms

3.2.2. Additive Sytnhesis

When compared to subtractive synthesis, additive synthesis operates through the opposite principle, although results can be quite similar. Whereas, in subtractive synthesis, the basic principle is the elimination of harmonics in order to obtain the desired sound, in additive synthesis new harmonics are added to configure the final sound.

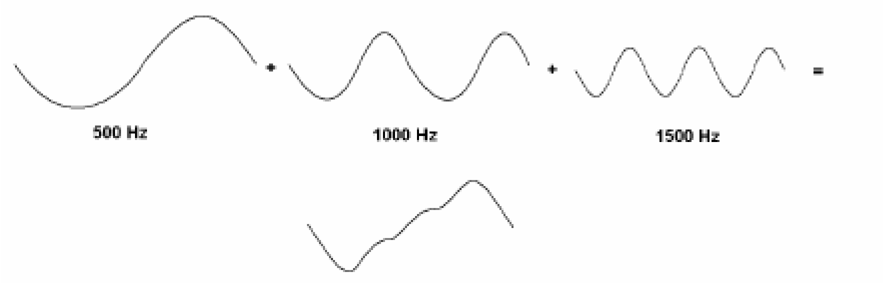


Fig. 20. Construction of a sawtooth wave using additive synthesis

3.2.3. Frequency Modulation Synthesis

Frequency modulation synthesis usually called FM, has a quite curious history. It was discovered nearly by chance in the 60s by John Chowning, a researcher from the Stanford University, when he was working on vibrato techniques. The theoretical operation of the frequency modulation is actually very similar to the vibrato, hence the context in which it was discovered. Vibrato consists in a cyclical variation of the waveform frequency, which slightlychanges its pitch around a certain base frequency at a certain speed, leading to that sensation of "vibration". In a synthesizer, the vibrato effect can be achieved using a LFO which affects the signal of the main oscillator. The wave generated by the oscillator is usually called the carrier, whereas the LFO signal is called the modulator.

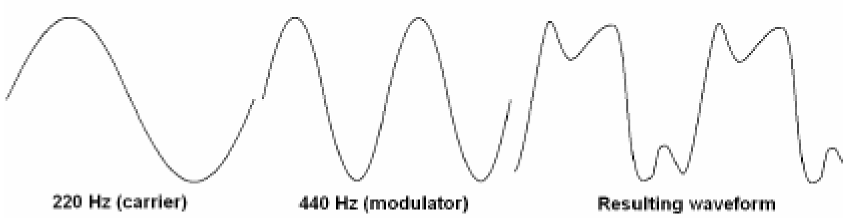


Fig. 21. Basic scheme of the frequency modulation synthesis

3.3 Filter Basics

Filters are devices (or algorithms) which change the spectrum of signals - their most prevalent action on signals is to boost, attenuate or completely block frequencies. In the tutorial about sinusoids, we saw that any signal can be seen a sum of sinusoids, each of which having its own frequency f, amplitude A and phase ϕ. In general terms, filters modify the amplitudes and phases of incoming sinusoids according to their frequency. Assume that the input signal to our filter is a single sinusoid with frequency f, amplitude A and phase ϕ. The output signal will again be a sinusoid of frequency f but possibly with different values for the amplitude and phase. And this is already the essence of a filter: all a (linear) filter can do to a sinusoid is to multiply its amplitude by some factor and to add some offset to its phase; but a filter will never change the general shape of the sinusoid, nor will it change its frequency. So, let's call that multiplication factor for the amplitude G (for gain) and let's call that phase shift θ (a greek lowercase theta). Both, gain and phase-shift depend on the frequency of the incoming sinusoid, so both G and θ can be expressed functions of the frequency f:

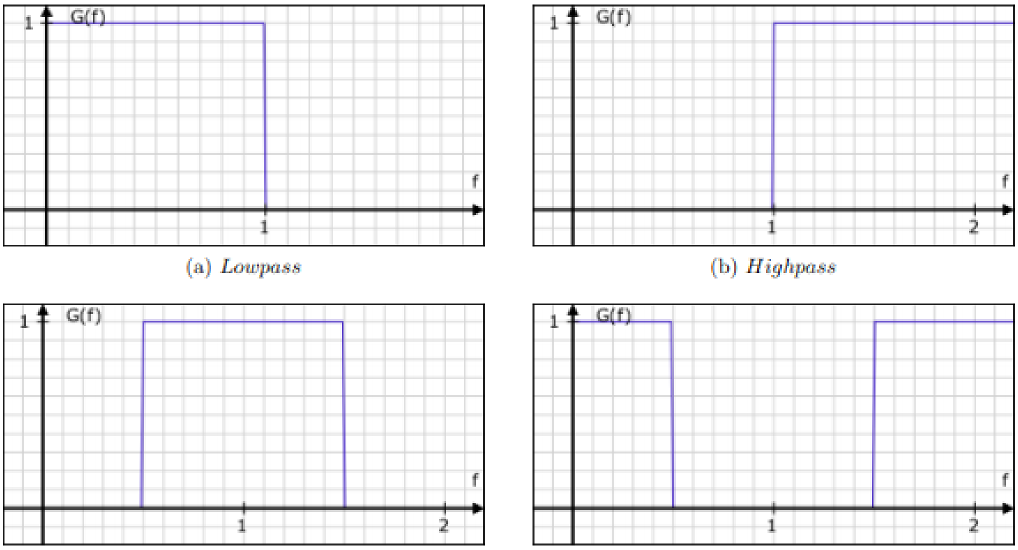
$G = G(f)$ and $\theta = \theta(f)$

These two functions are called the magnitude response and the phase response of the filter respectively. Taken together, these two functions form the frequency response of the filter - in a somewhat sloppier slang, one also often hears the term frequency response when the magnitude response alone is meant because this part of the frequency response often counts more. When we know these two functions, we can can predict the output signal of the filter for any arbitrarily shaped input signal by viewing the input signal as a sum of sinusoids.

Ideal Filters

The classical purpose of a filter is to let certain frequencies pass unchanged and to block others - hence the name filter. An idealized lowpass-filter for example would pass all frequencies up to some cutoff-frequency fc and block all frequencies above that cutoff-frequency. The idealized lowpass magnitude response GLP (f) would be therefore: $G_{LP}(f) = \begin{cases} 1 & \text{for } f \leq f_c \\ 0 & \text{for } f > f_c \end{cases}$

We would like the phase response to be identically zero for the filter types as well in the ideal case. In figure 1 we see the ideal magnitude responses for such filters. The lowpass and highpass filters in these plots are normalized in the sense that they have a cutoff frequency of unity. The physical unit in which frequency is measured is actually irrelevant for this discussion, but if you prefer to deal with something concrete and practical, feel free to suppose it to be kHz. The bandpass and bandreject filters have a lower cutoff of 0.5 and an upper cutoff of 1.5. Magnitude responses of idealized filters can be drawn as;



(a) *Lowpass* (b) *Highpass*

(c) *Bandpass* (d) *Bandreject*

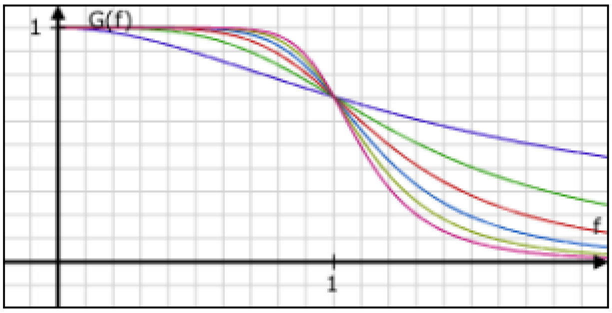
Real Filters

Unfortunately , we do not live in a perfect world and these ideal frequency responses as requested above are not attainable in the real world. Real filters can only approximate these requirements and various filter design techniques exist to obtain different kinds of approximations. As it turns out from the mathematics, the squared magnitude responses of realizable filters are always ratios of even polynomials (also called rational functions), that is functions of the form: $G^2(f) = \dfrac{B^2(f)}{A^2(f)}$ where B(f) and A(f) are both polynomials of their argument f. So, in order to design a filter which approximates any desired magnitude response, we must find a rational function which best approximates the desired magnitude response.

This is the so called approximation problem in filter design. For the lowpass-, highpass-, bandpass- and bandreject-responses, standard solutions to this design problem exist and these are are known under the names Butterworth, Chebyshev, inverse Chebyshev, Bessel, elliptic, etc. The simplest of these is the Butterworth approximation. In the case of a lowpass filter with unit cutoff frequency the approximant is chosen as; $G^2(f) = \frac{1}{1+f^{2N}}$

Where N is some positive integer number which is called order of the filter. To obtain the (non-squared) magnitude response, we must take the square root of this; $G(f) = \sqrt{\frac{1}{1+f^{2N}}}$

Magnitude responses of Buttweworth lowpass filters of orders 1 to 6 can be drawn as;



The graph for N = 1 has the shallowest slope with lowest values G(f) for frequencies above the cutoff frequency (fc = 1) and the highest values above the cutoff frequency - thus, compared with our ideal response it performs worst. As the filter order N increases, the actual frequency responses approach our ideal better and better in the sense that they come closer to unity below the cutoff frequency and closer to zero above.

The Slope of the Filter

Our perception of frequency and amplitude for audio signals spans a huge range and is (roughly) proportional to the logarithm of the quantity in question (frequency or amplitude). That's why we often draw the magnitude response of a filter in a coordinate system in which both axes are scaled logarithmically - for example the frequency axis in octaves and the magnitude axis in decibels. In the case of Butterworth lowpass filters this leads to the plot in figure 3. This time, the plot has been drawn with actual physical units on the axes and the cutoff frequency has been tuned to 1000Hz. We observe, that the filters gain in decibels is approximately 0dB below the cutoff-frequency, −3.01dB at the cutoff frequency and drops off approximately linearly above the cutoff frequency. This linear behavior of the filter gain in a doubly logarithmic plot makes it meaningful to associate a slope with the filter. The slope defines, by how many decibels the filter gain drops off per (logarithmic) frequency interval above the cutoff frequency. The interval is usually measured either in octaves (frequency ratio 1/2) or in decades (frequency ratio 1/10) where in the former case the unit of the slope is decibels per octave (dB/oct) and in the latter case decibels per decade (dB/dec).

Resonant Filters and Subtractive Synthesis

Another type of filter which the audio engineer is likely to stumble across, is the resonant filter and in particular the resonant lowpass filter. Filters of that kind are ubiquitous in subtractive synthesizers. A resonant lowpass filter is generally lowpass in its nature but it exhibits a resonant peak in the vicinity of the cutoff frequency. The peakiness of this peak (its height and narrowness, that is) is determined by an additional user parameter, most often called 'resonance' but sometimes also 'Q', 'emphasis' or 'feedback'.

A word about the definition of 'cutoff-frequency' is in order: for technical applications, one most often defines the cutoff frequency as the half-power frequency or equivalently, the −3.01 dB point. Here in this case, it makes more sense to define the cutoff frequency as the frequency at which the filter resonates and this is the frequency where the gain is at −12.04 dB here. Why is this? The filter consists of a series connection of 4 identical first order lowpass filters, each contributing a drop in magnitude of −3.01 dB and a phase shift of -45° at the cutoff frequency. The combined effect is therefore a magnitude drop of −12.04 dB and a phase shift of -180° . A sine wave which is shifted by 180° and inverted is the same sine-wave again, thus feeding back the filter output to the input will lead to an interference which is maximally constructive at the cutoff frequency (because the interfering sines are exactly in phase at that frequency. This explains why the resonance occurs where it occurs. The drop of −12.04 dB at this frequency on the other hand, explains why the resonance runs into self oscillation with a feedback gain of 4: −12.04 dB corresponds to a factor of 1/4 and the factor 4 cancels this out to yield unity gain in the overall feedback path at the resonant frequency. Theoretically, feedback gains from 4 upwards would lead to a buildup of a resonance with infinite amplitude. However, because all real filter saturate internally at some level, this won't happen in practice. What we see instead is a stable oscillation but due to the saturating nonlinearities inside the filter, this oscillation is not exactly sinusoidal but also has some overtones. Strictly speaking, when driving the filter into self oscillation and/or saturation, we cannot really apply the theory of linear filters anymore.

Nonlinear Filters

Filters are generally assumed to be linear which is equivalent to the assumption that they multiply the amplitude of an incoming sinusoid by some factor and shift the phase of the incoming sinusoid by some offset - and nothing else. a nonlinear (or non-linear) filter is a filter whose output is not a linear function of its input. That is, if the filter outputs signals R and S for two input signals r and s separately, but does not always output $\alpha R + \beta S$ when the input is a linear combination $\alpha r + \beta s$.

Both continuous-domain and discrete-domain filters may be nonlinear. A simple example of the former would be an electrical device whose output voltage $R(t)$ at any moment is the square of the input voltage $r(t)$; or which is the input clipped to a fixed range [a,b], namely $R(t) = \max(a, \min(b, r(t)))$. An important example of the latter is the running-median filter, such that every output sample R_i is the median of the last three input samples r_i, r_{i-1}, r_{i-2}. Like linear filters, nonlinear filters may be shift invariant or not.

3.4. Circuitry

We used Labcenter Electronics Ltd's Proteus Design Suite software for PCB circuit design of the synthesizer. The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards. The Proteus Design Suite is a Windows application for schematic capture, simulation, and PCB (Printed Circuit Board) layout design. It can be purchased in many configurations, depending on the size of designs being produced and the requirements for microcontroller simulation.

Schematic capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project. It is therefore a core component and is included with all product configurations. All PCB Design products include an autorouter and basic mixed mode SPICE simulation capabilities. The PCB Layout module is automatically given connectivity information in the form of a netlist from the schematic capture module. It applies this information, together with the user specified design rules and various design automation tools, to assist with error free board design. PCB's of up to 16 copper layers can be produced with design size limited by product configuration.

The 3D Viewer module allows the board under development to be viewed in 3D together with a semi-transparent height plane that represents the boards enclosure. STEP output can then be used to transfer to mechanical CAD software such as Solidworks or Autodesk for accurate mounting and positioning of the board.

4. DESIGN AND CONSTRUCTION

During the design phase of the project, we decided to use the circuit design of Tiny Synth, which we determined as the main source in line with the consensus we reached with our consultants. Our Synth is a simple, may be minimum, compact and integrated synthesizer. It was developed around a single MIDI controlled digital oscillator, including too a VCF, VCA, LFO, and AR ASR envelope generators.

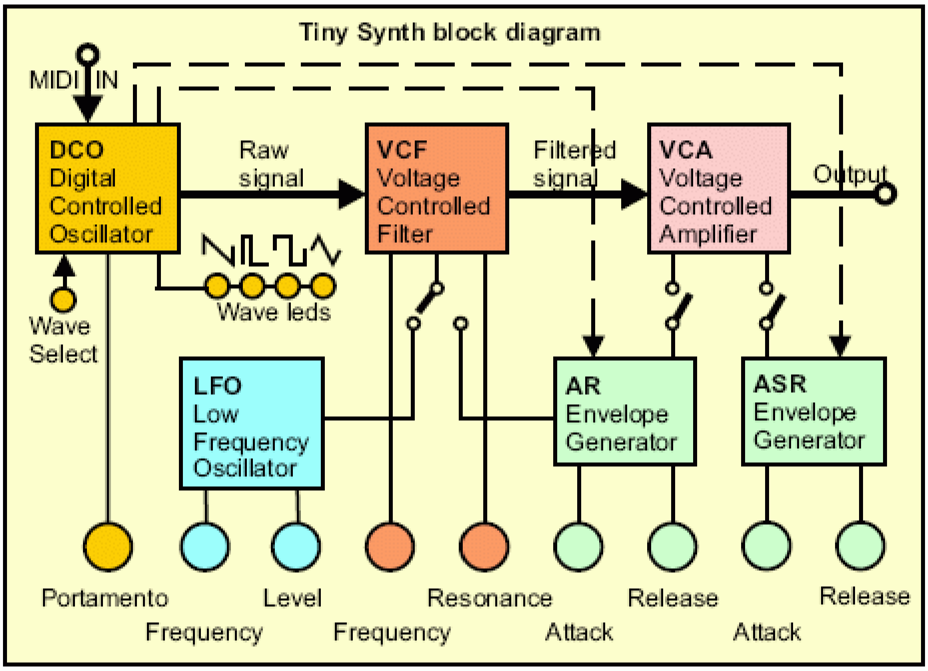


Fig. 22. Block diagram for hybrid synthesizer

The heart of the system is the AT89C2051 microcontroller. There are no uncommon components at all, the integrated circuits includes operational amplifiers LM324 an TL084, an octal buffer 74HC245, a CD4007 MOSFET dual complementary transistor pair plus inverter, the traditional 555 and regulators 7805 and 79L05.

There are three switches for controlling the interconnection between the internal modules. The S1 switch connects the VCF frequency control to LFO or AR signals. The S2 switch connects the VCA to the AR envelope or left it unconnected. The S3 switch select the VCA gain control to ASR or unconnected. When both S2 and S3 are selected to AR and ASR positions, the result is a sum of both signals giving similar result as ADSR envelope generator. When both S2 and S3 are unconnected, VCA let flow the wave signal with an attenuation of about 10dB.

4.1. PCB Design

We have used Proteus Design Unit for PCB Design as aforementioned. Our PCB guide from our main source site is shown below. The Printed Circuit Board is single layer and includes all potentiometers and external connectors on board, avoiding wiring.

The circuit layout maintain enough wire separation that make non critical the construction, giving the possibility of drawing it by hand.

The source power is simply an external 9V to 12V AC/DC converter, avoiding undesired interferences with an internal transformer. This external voltage is reduced to +5V with a traditional 7805 regulator and created negative voltage of minus 5V with a DC/DC converter voltage implemented with an oscillator associated to a voltage duplicator.

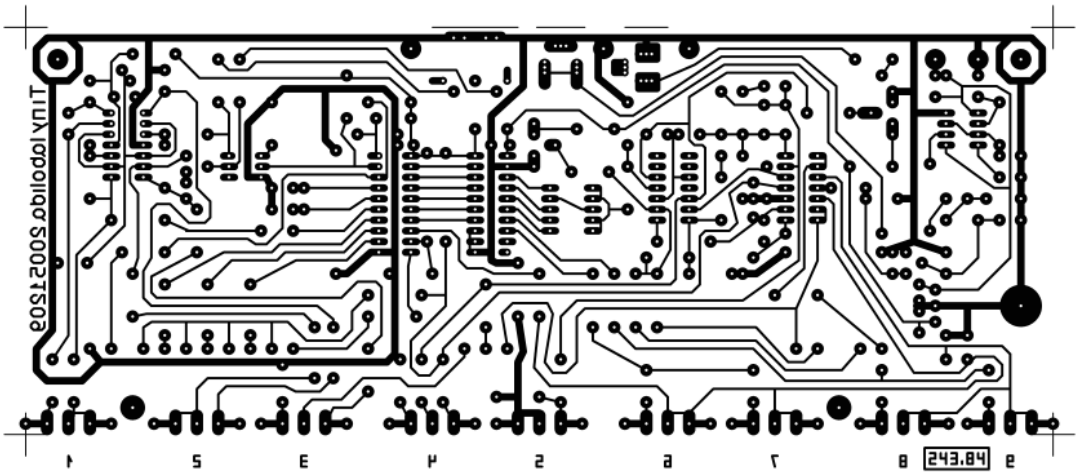


Fig 23. PCB design guide of the synthesizer

First we draw our schematic capture shown below.

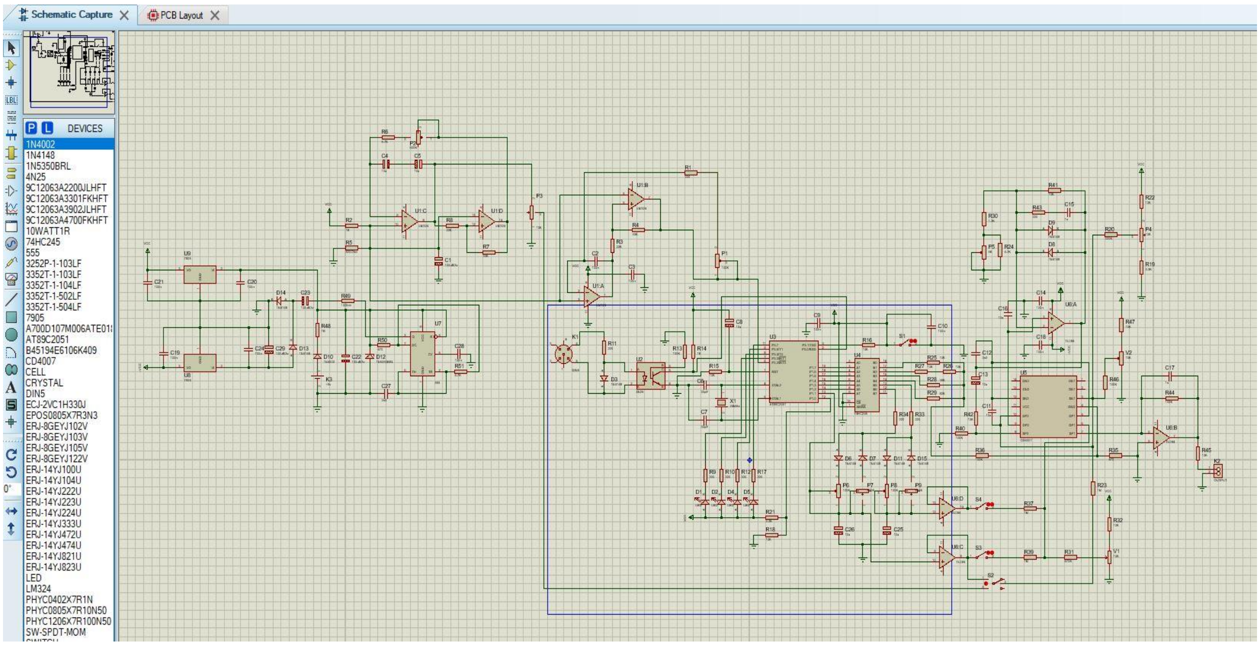


Fig. 24 Schematic capture of the synthesizer

After drawing of the scheme of the circuit, we started PCB Layout design. For the location of components we have used that guide from our main source shown below.

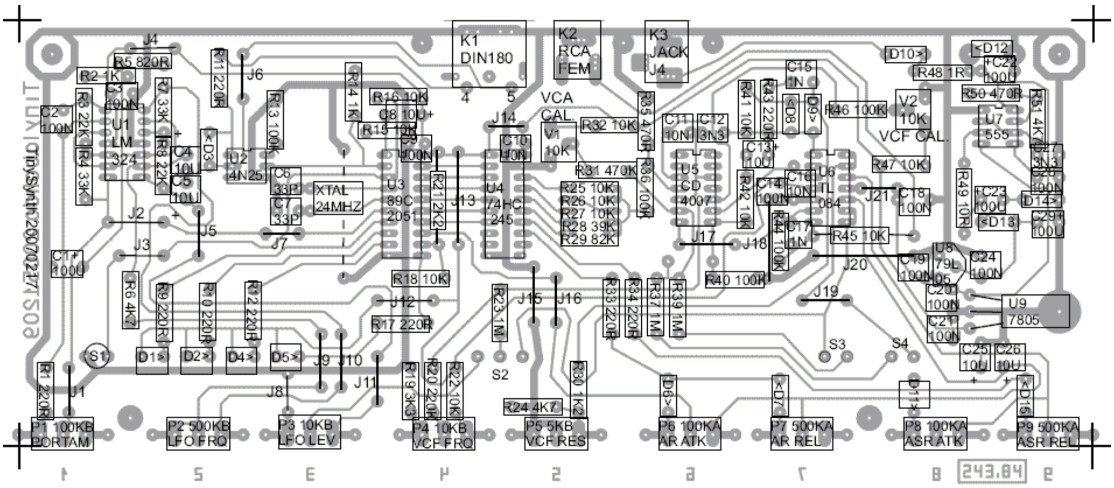


Fig.25. Components guide for PCB design

In the light of the above diagram, we completed our PCB Layout drawing shown below and printed our PCB in a professional PCB printing store.

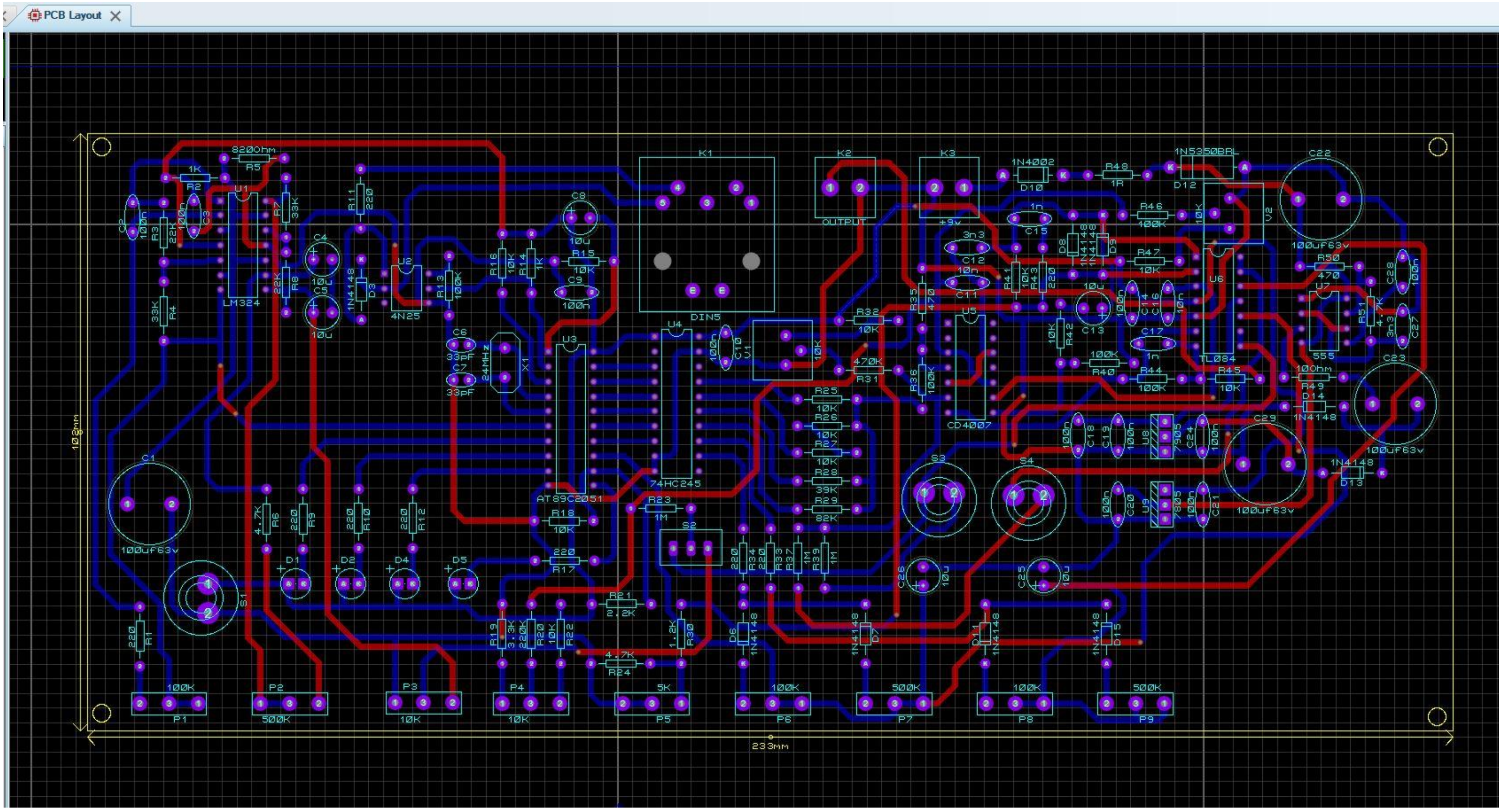


Fig.26. PCB Layout from Proteus

With the 3D Viewer module of Proteus, our circuit design was look like:

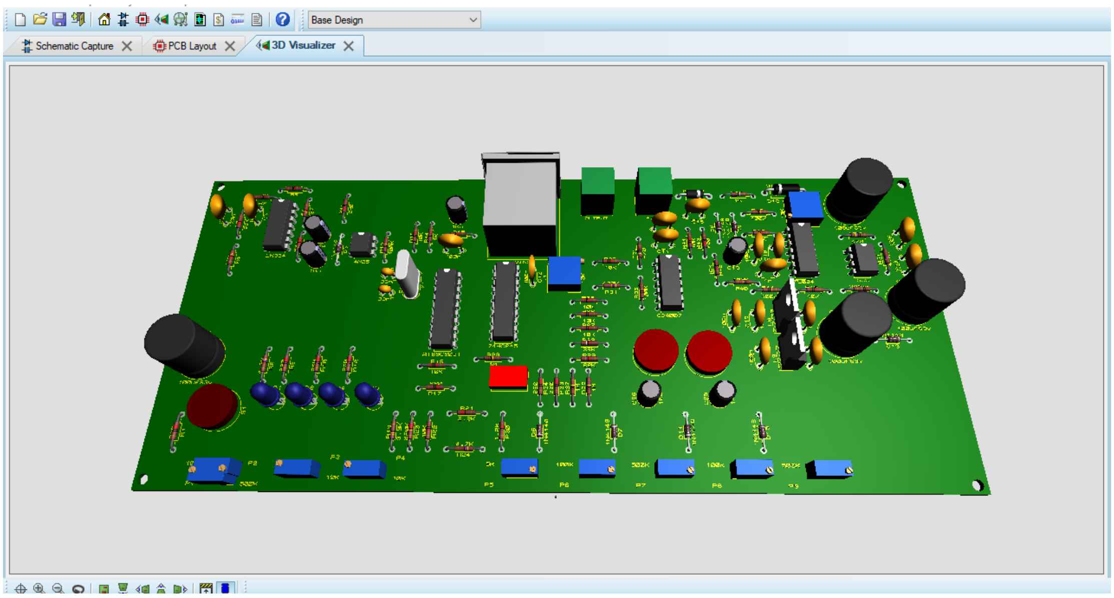


Fig.27. 3D visualized version of the circuit

4.2. Construction

The Digital Controlled Oscillator is implemented with an AT89C2051 microcontroller, where are processed MIDI messages, generate waveforms and envelope signals. The wave has 4 bits resolution, using a passive D/A converter through resistors with weights of 1, 2, 4, and 8.

The glide is obtained via software in successive halftone steps controlled through a variable frequency oscillator. The program was developed in C language and compiled with SDCC, Small Devices C Compiler, available as Open Source.

The Low Frequency Oscillator sends a triangular wave that can control the VCF, with a range of 0.2 to 20 Hz.

The Voltage Controlled Filter is based in the Sallen-Key structure, using MOSFET N-channel transistors of CD4007 integrated circuit as variable resistors, to take advantage of coupled pairs inside. The CD4007 is a MOSFET dual complementary transistor pair plus inverter.

The frequency control is about 800mV per octave, in the range of 100 Hz to 5 KHz, whit an offset of 1V approximately, depending of device used. Out of this range has a non-exponential behavior. The behavior of this filter is satisfactory when the linearity of control voltage is not a critical requirement.

The Voltage Controlled Amplifier uses a MOSFET N-channel transistor of CD4007 as variable input resistor in an inverter operational amplifier. A capacitor included in the feedback path reduces the noise in high frequency band.

The AR envelope generator is done using an internal comparator of AT89C2051. At the NOTE ON message a logic 1 signal is send to timing capacitor through an attack potentiometer. When the comparator detects the high limit level in the capacitor, turn the signal to logic 0. It discharges the capacitor through the release potentiometer.

The ASR envelope generator is implemented through the logic 1 signal during NOTE ON and NOTE OFF period, charging and discharging a capacitor through attack and release potentiometers respectively.

The calibration of VCF initial frequency is done through V2 preset that maintain a positive bias to the gate of MOSFET transistor. This frequency must be maintained around 10Hz in order to avoid an undesirable transition to off condition of the transistors.

The calibration of VCF initial frequency is done through V1 preset that maintain the bias of transistor gate in order to reaches attenuation around 80dB when te envelope generators are inactive.

After we expressed our materials technicallcy, now we will present our construction stages by pictures.

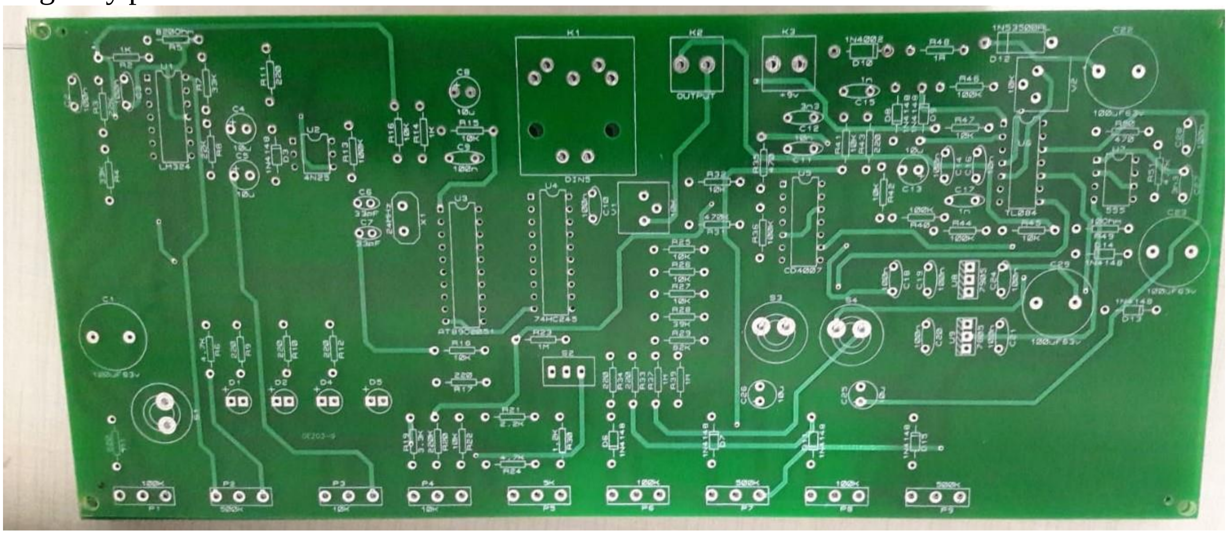


Fig.28. Printed Circuit Board of the Synthesizer

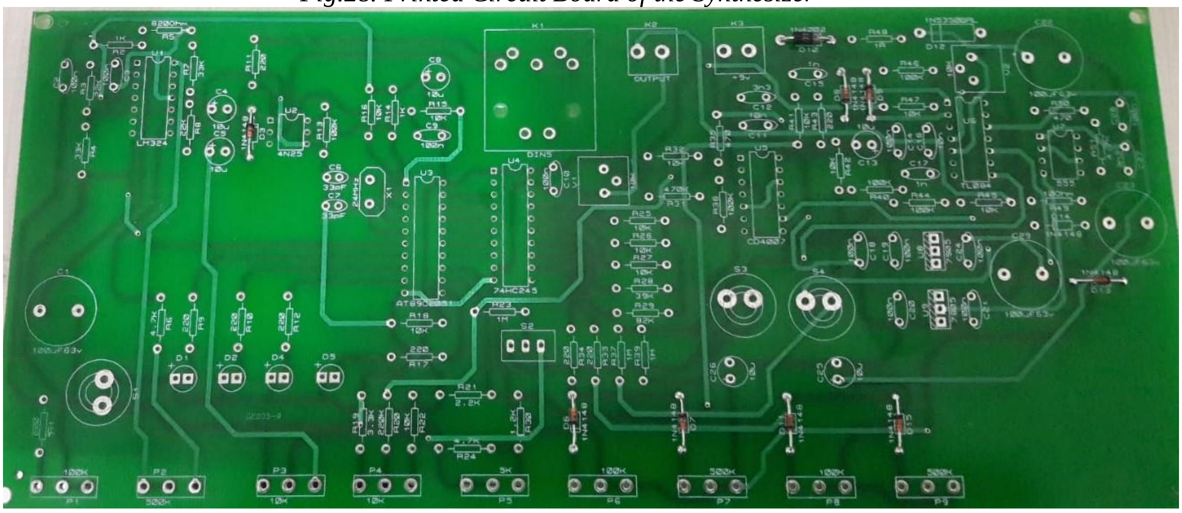


Fig.29. Construction Stage 1

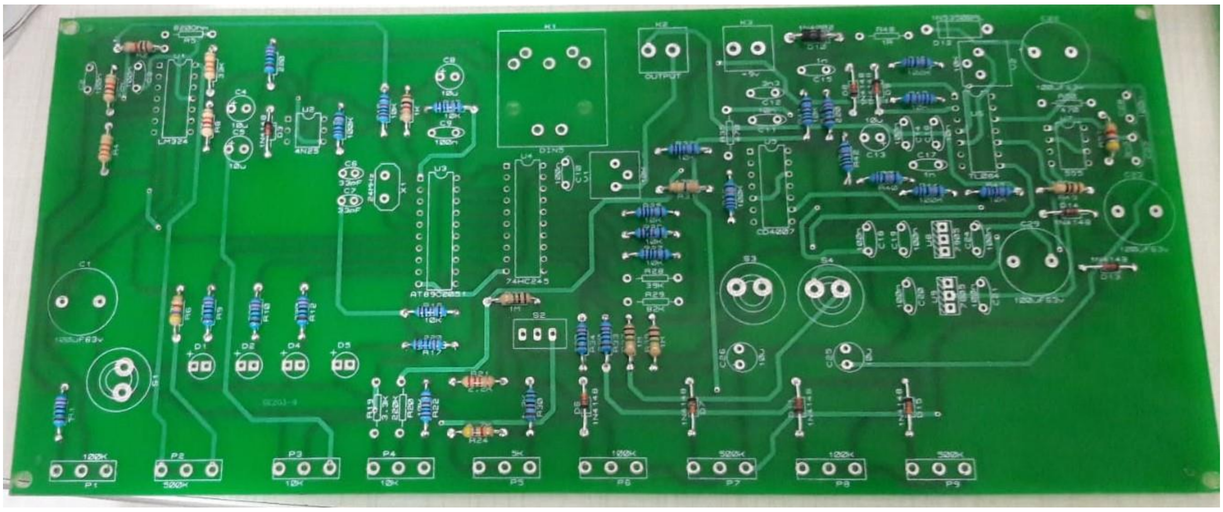


Fig.30. Construction Stage II

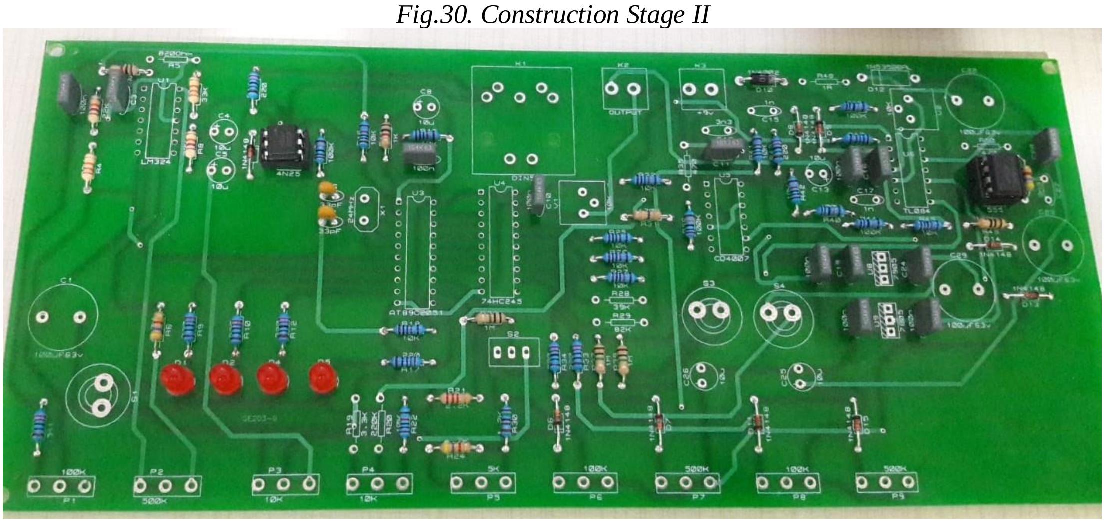


Fig.31. Construction Stage III

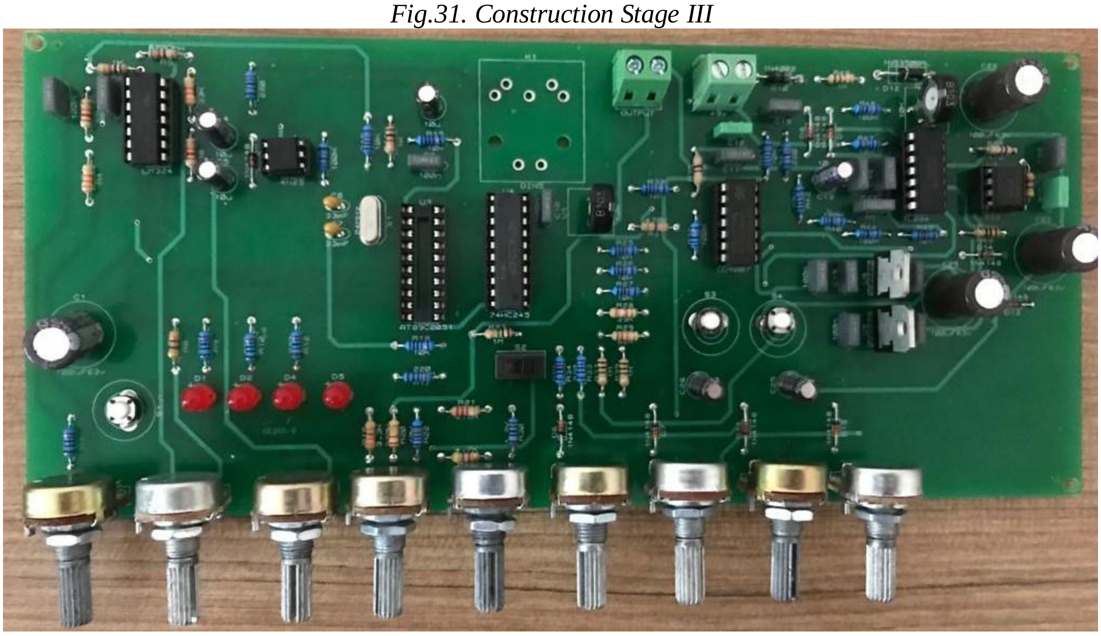


Fig.32. Final look of the circuit

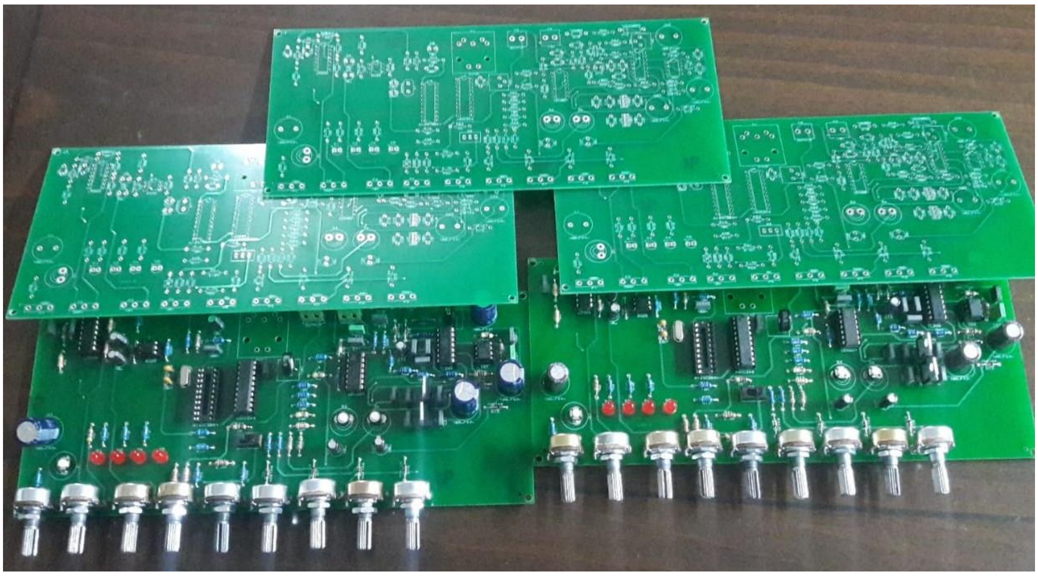


Fig.33. All work together

4.3. Materials

S2,S3,S4-Switch

P2-P5P6-P8-P3-P4-The property of the potentiometer is that it has controllable resistance. It is one of the basic elements of electronics and is found in many circuits that require control.

C20-C21-C24 Ceramic capacitors contain ceramic substances such as barium titanate or titanium dioxide among the conductive sheets as dielectric material. The capacity values of these types of capacitors are easily affected by temperature and humidity. Because the energy losses of ceramic capacitors are low, they can be used in high-frequency circuits.

V1-V2(Trimpot) When used as a voltage divider, usually one leg of the trimpot is connected to the supply voltage, another leg is connected to the ground, and the moving leg is also used to receive output. If the trimpot is to be used as a variable resistance, the variable end is connected to the ground end.

C2-C28-C11-C16-C15-C17-C12-C27-Polyester capacitors use polyester as an insulating material. Areas of use they are used in circuits up to 50-1000 Volt.

C13,C25,C26,C4,C5,C8-C1,C22,C23,C29-C1,C22,C23,C29 -Polarized type (polarized) capacitors. They can provide high capacity values (mostly 1µF and above). They are cylindrical in structure.

X1-A resonator is an electrical component that exhibits a series of resonances and a parallel resonant center frequency. It exhibits a piezoelectric property that allows the ceramic material to generate minimal electrical energy when subjected to electromechanical expansion and compression.

K3- Rca Connector- An RCA connector is a type of connector organized for coaxial cables to be used from several megahertz to low frequency ranges. it is used to carry audio and video signal.

U1-U4-U6-Y7-U8-U9 An integrated circuit can sometimes be called a chip or microchip. Integrated on miniaturized active devices (transistors and diodes) and passive devices is an electronic circuit manufactured as a single unit is a set of.

4.4 Microcontroller Coding Part

The main part of our system is AT89C2051 microcontroller. We used Xgecu's TL866II Plus Universal programmer in programming part.

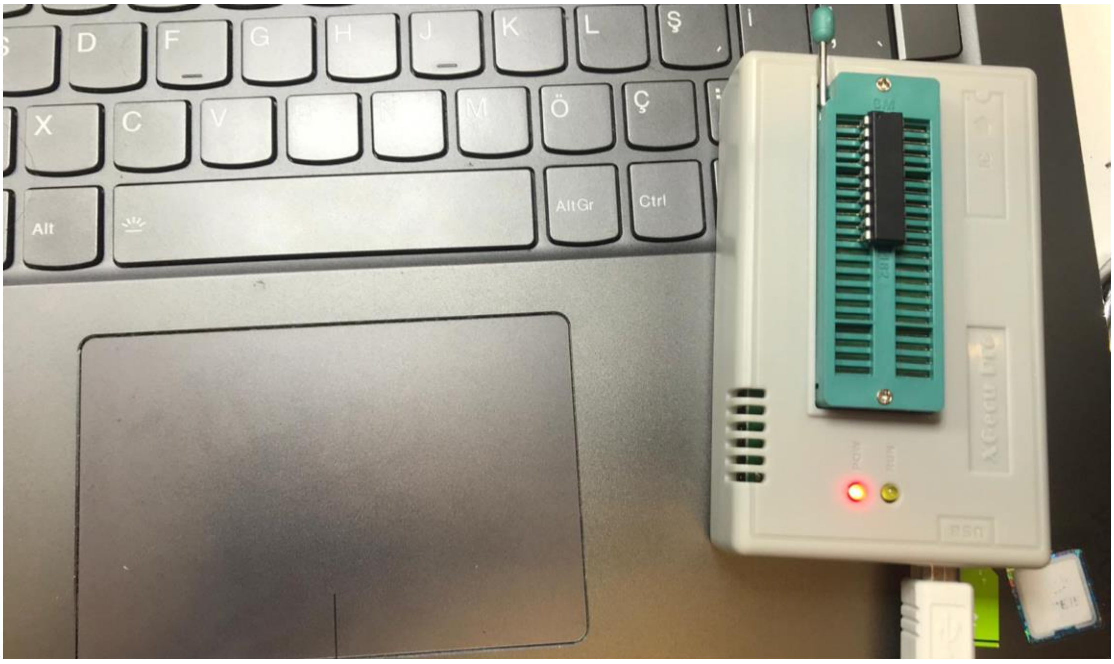


Fig. 34. XGecu TL866II Universal Programmer

When we started the coding part of our project, we realized that we had drawn some points wrong in the circuit and we worked on it by thinking how we could solve it. Although it may seem like minor problems, we encountered new problems when we dug deeper, and we tried to identify the problems and fix the ones we could fix manually, with the help of cables.

When these problems were over, this time we encountered problems with the circuit elements. We detected the broken circuit element and replaced it. Later, our circuit was getting very hot and we were afraid that it would burn, so we changed the circuit elements by reviewing them from the beginning. After we finished our coding and connected the power, we obtained the signals by renting a workshop over the oscilloscope as much as we corrected our error.

Then we connected the circuit to an amp and we showed in the video that we can get different sounds with the button and play on it.

The steps we performed in TL866II Plus Programmer software are shown below.

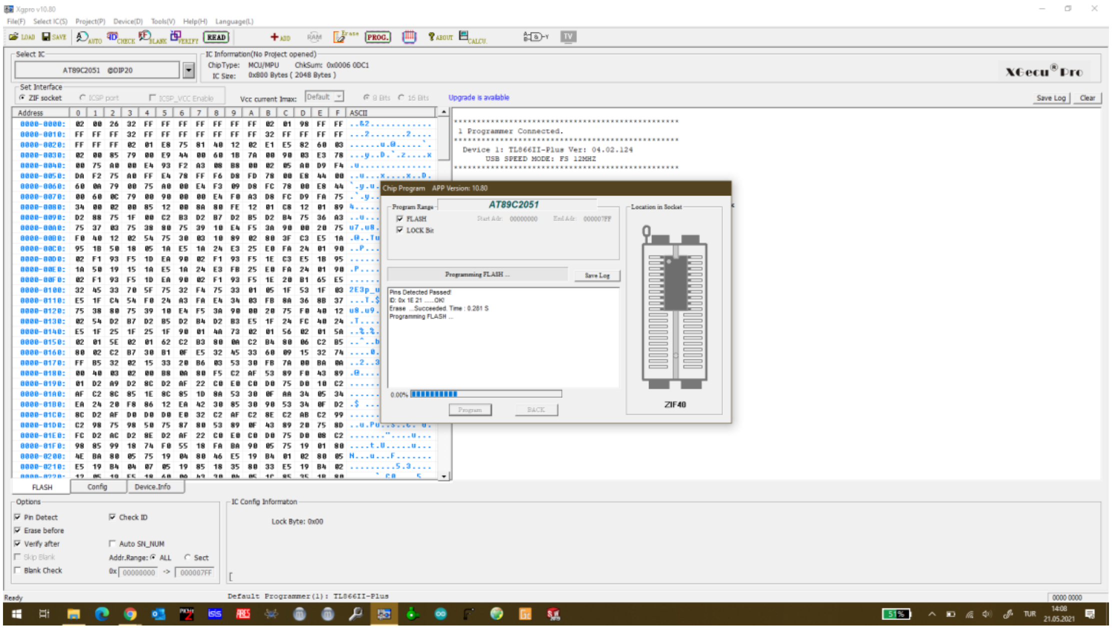


Fig.35. Programming Step 1

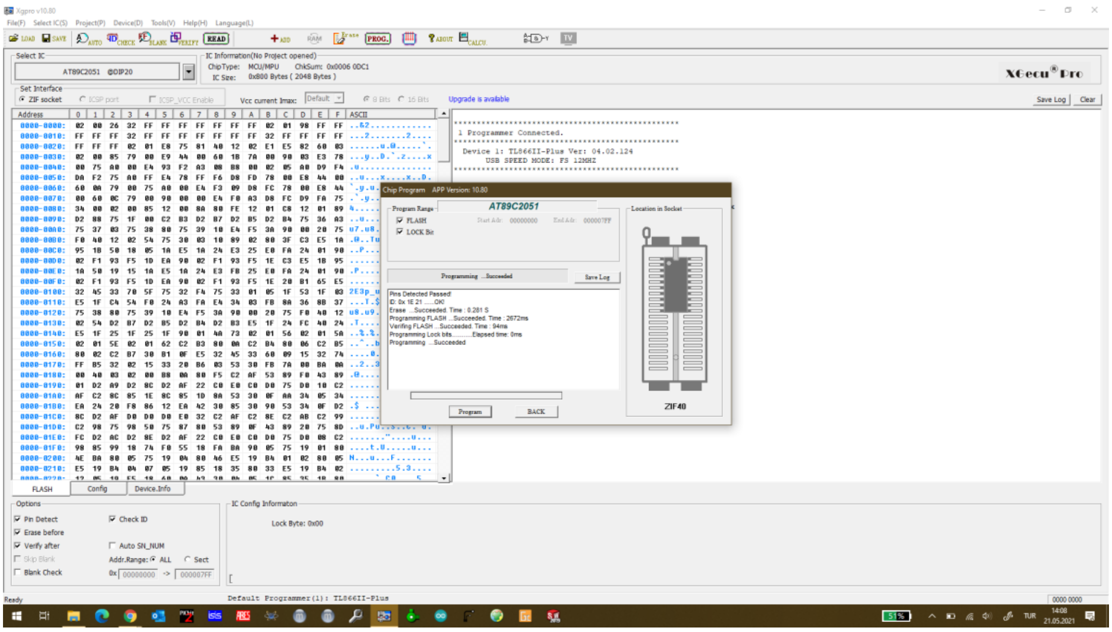


Fig.36. Programming Step 2

The main code presented as appendix at the end of the report.

5. CONCLUSION

5.1. Results

As a result, we built a synthesizer that produces different types of waves in our project. The oscilloscope outputs of the synthesizer, whose construction stages have been mentioned above, are presented below.

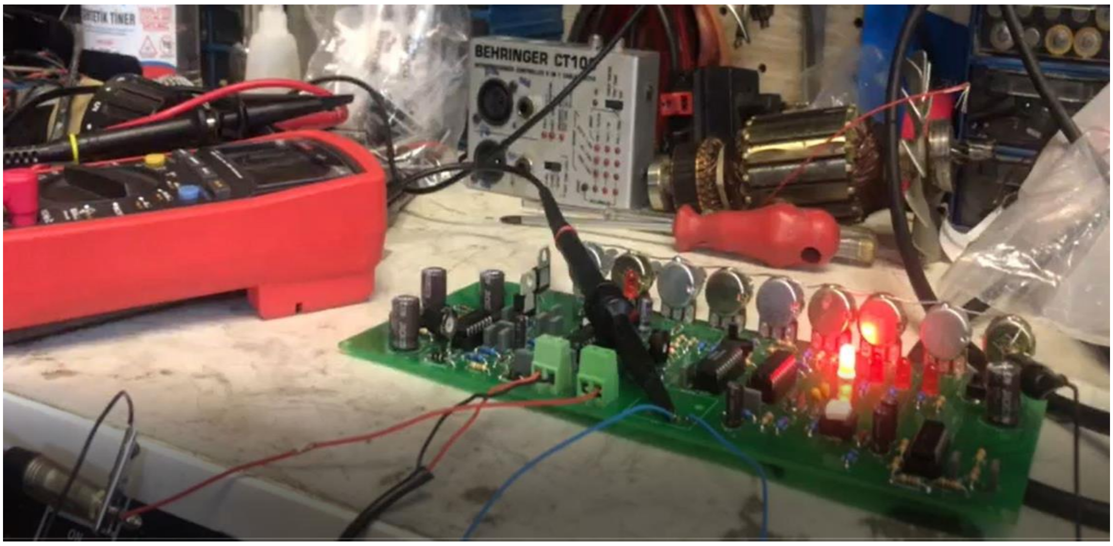


Fig.37. Oscilloscope Instalisation in Lab

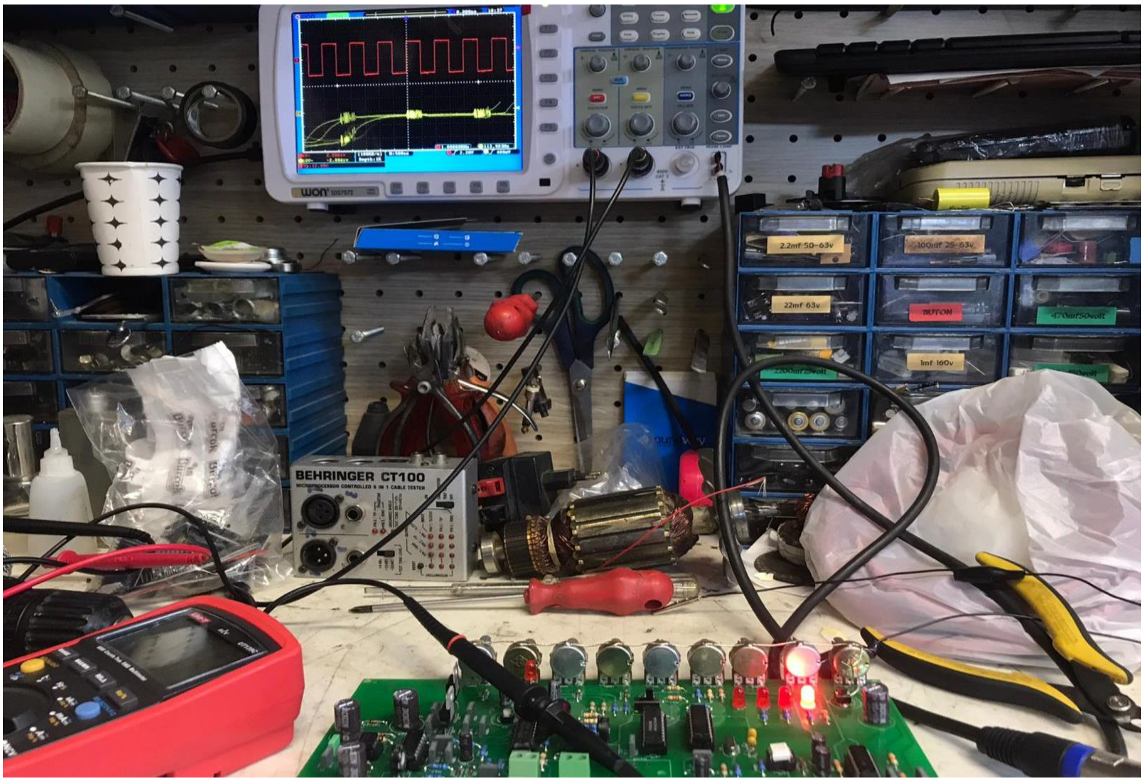


Fig.38. Oscilloscope Output and Circuit

If we are asked why our output is taken as a square wave, we input a square wave for analogy to an audio signal, since signal generators are usually square wave output.

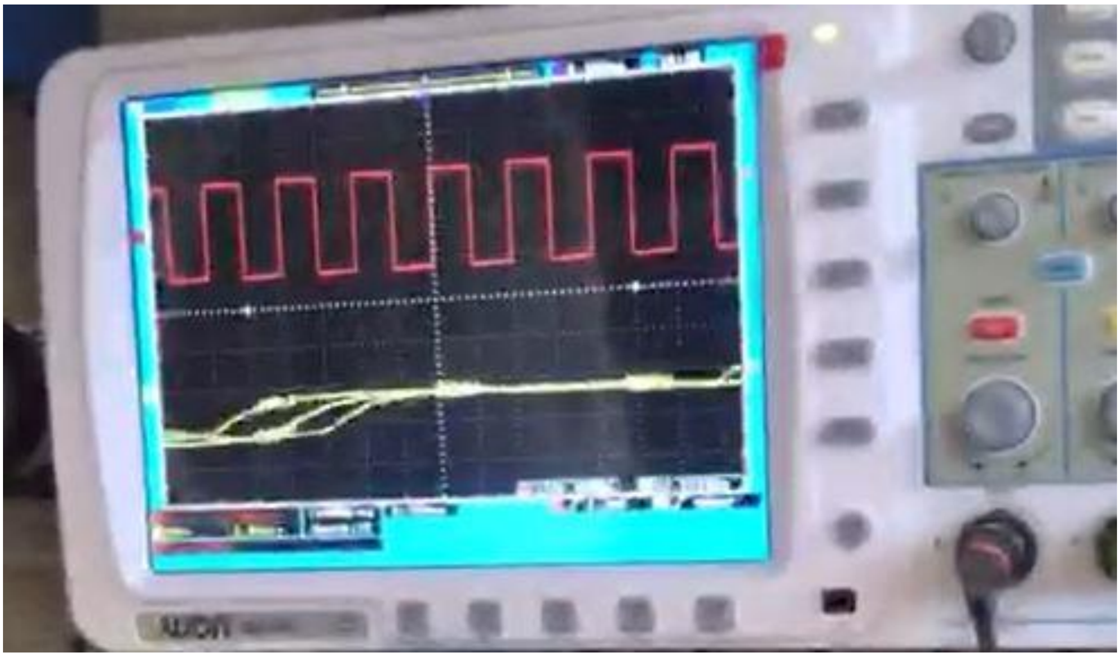


Fig.39. Output Graph of a square waveform 1

Fig.40. Output Graph of a Square waveform 2

5.2. Discussion and Future Work

We aimed to run the circuit we have created by power integration and to obtain a compact and functional mini synth, which is completed. Keyboard integration and body construction can be mentioned as future work. Lastly, we need to enter an instrument as a sound, and for this, a professional study is required, can be listed as futurework.

5.3.Social, Environmental and Economical Impact

We have mentioned in the first term’s report that the biggest constraint we may encounter socially will be to convince current musicians after developing our device. Of course, we don't set out with the wildest goals of dethroning Moog synthesizers, but if the synthesizer we developed inspires people and is produced by more engineers, it is a good goal to ask musicians to use it to reach more people. Being careful about environmental waste may limit us at some points.

Secondly we have mentioned that we will try to set up the circuits again and again, we will fail, we will use new materials. As we stated we have faced this issue plenty of times but we succeed at the end of the day. We are obliged to produce the least possible electronic waste here, which we were always careful about it. It was explained above that the difficulty of financial access to this instrument was one of the biggest goals this project aims to solve and we mentioned that, this situation may also constrain us, we must correctly calculate each material we use and ensure that our financial goal motivates us rather than restricts us. This situation was not a big issue to us, and we can say that we have created a synthesizer in a cheap way.

Materials

5.4. Cost Analysis

Name of material	Unit Price (TL)	Piece	Price (TL)
Switch	1.12	3	3.36
100K resistor	0.05	50	2.5
100Kb Linear Potentiometer	1.41	1	1.41
100N Polyester or Mylar Capacitor	0.4	10	4
100U Electrolytic Capacitor	0.5	10	5.4
10K resistor	0.075	50	3.80
10K Trimpot	1.40	2	2.80
24MHz XTAL	3.24	1	3.24
10KB Linear Potentiometer	1.41	2	2.82
10U Electrolytic	1.30	6	7.8
1N Polyester Capacitor	2.32	2	4.64
1N 4002 D	0.13	10	1.39
1N 4148 D	0.09	10	0.94
1N 5350B D	7.8	1	7.8
33P Ceramic	1.42	2	2.84
3N3 Polyester or Mylar, 25V	2.84	2	5.68
500 KB Linear Potentiometer	1.41	1	1.41
5KB Linear Potentiometer	1.4	2	2.82
74HC245	1.88	1	1.88
7805	1.69	1	1.69
79L05	0.85	1	0.85
820R	0.047	10	0.47
DIN1 80 Connector	1.88	1	1.88
LED	1.69	5	5.64
LM324	1.69	1	1.69
RCA Female Connector	1.88	1	1.88
Push Button Normally Open	0.47	2	0.94
TL084	1.88	1	1.88
AT89C2051-24PU 8-Bit 24MHz Microcontroller	16.90	1	16.90
TOTAL			**115.12 TL**

Table 1.1. Cost of Materials

PCB Printing Charges: 100 TL

Labor Cost: 8 months, 8 hour work for each week and 9.25 TL cost per hour for each engineer. Approximately 250 hour work. 250. 9.25 = 2312.5 TL for each engineer.

Workplace rent: 300 TL

It means total labour cost is approximately 5.025 TL.

APPENDIX

Main Code

```
//  implemented with AT89C2051 with 24 MHz and 2k(800h) flash memory
//  MIDI input is controlled by the serial interface associated to
//  timer 1 in mode 2 (8 bits auto reload).
//  The digital oscillator (DCO) frequency is controlled by the
//  timer 0 in mode 1 (16 bits without auto reload).
//  compiled with sdcc --no-pack-iram
//__code char versn[] = {"_Midi109_"};    // adapt to Tiny 4 ATOM - Leds
//__code char versn[] = {"_Midi110_"};    // sample buffer
//__code char versn[] = {"_Midi111_"};  // ASR envelope signal
__code char versn[] = {"__Midi112__"};  // AR envelope signal

#include <8051.h>                       // standard 8051 header
#include <string.h>
#define INT_DISABLE (EA = 0)            // disable interrupts
#define INT_ENABLE  (EA = 1)            // enable interrupts
#define UCHAR unsigned char

#define NOTE_ON     0x90                // midi command note on
#define NOTE_OFF    0x80                // midi command note off
                                        // pin 6 P3_2 INT0 portamento
                                        // pin 16 to 19 P1_4 to 7 vave out
#define BUTTON_WAVE P3_1                // pin 3 wave select button
#define LED_1       P3_3                // pin 7 INT1 indicator LED 1
#define LED_2       P3_4                // pin 8 T0 indicator LED 2
#define LED_3       P3_5                // pin 9 T1 indicator LED 3
```

```
#define LED_4       P3_7              // pin 11 indicator LED 4
#define AR_REF      P1_0              // pin 12 AR envelope reference(+)
#define AR_LEV      P1_1              // pin 13 AR envelope comparator(-)

static void dco_ini(void);
static void dco_int(void) interrupt 1 using 2;
static void midi_ini(void);
static void midi_int(void) interrupt 4 using 1;
static __data UCHAR input_byte;        // input byte MIDI interface
static __data UCHAR midi_state;        // parser automata state
static __data UCHAR note_curr;         // current note number
static __data UCHAR note_next;         // future note number
static __data UCHAR note_count;        // note on counter
static __data UCHAR frqdiv_lo;         // freq.divider timer 0 low
static __data UCHAR frqdiv_hi;         // freq.divider timer 0 high
static __data UCHAR wave_current;      // index current wave 0 to 3
static __data UCHAR wave_sample[16];   // waveform sample table
static __data UCHAR wave_buffer;       // wave sample and envelope buffer
                                       // bits 7 - 4 wave high to low
                                       // bit 3 ASR - bit 2 AR
                                       // bit 1 input comp - bit 0 refer.
static __data unsigned char toggle;    // DEBUG
#include "freq.c"                      // frequency divider table
#include "wave.c"                      // waveform table
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
void main()
{
   static __data int wave_debounce;    // debounce counter
   __data unsigned char i;             // table notes index
   midi_ini();
   dco_ini();
   IT0 = 1;                            // portamento
   wave_current = 0;                   // default waveform
   LED_1 = 0;                          // initialize default led
```

```
LED_2 = LED_3 = LED_4 = 1;          // initialize leds
memcpy(wave_sample, wave_table[0],  // load default sample wave
   sizeof(wave_sample));
wave_buffer = 0x03;                 // initialize AR ASR A0 A1 high z
while(1)
{
   if (IE0)                        // INT0 portamento
   {   IE0 = 0;
      if (note_next > note_curr)
      {   note_curr ++;
         i = (note_curr - 29)*2; // index of note table
         frqdiv_lo = note29[i+1];// freq.divider timer 0 low
         frqdiv_hi = note29[i];  // freq.divider timer 0 high
      }
      if (note_next < note_curr)
      {   note_curr --;
         i = (note_curr - 29)*2; // index of note table
         frqdiv_lo = note29[i+1];// freq.divider timer 0 low
         frqdiv_hi = note29[i];  // freq.divider timer 0 high
      }
   }
   if ((! BUTTON_WAVE)             // wave selection button on
   &&  (! wave_debounce))          // debounce complete
   {   wave_debounce = 500;        // load counter
      wave_current ++;             // change wave index
      wave_current &= 0x03;        // truncate
      memcpy(wave_sample,          // store waveform
         wave_table[wave_current],
         sizeof(wave_sample));
      LED_1 = LED_2 = LED_3 = LED_4 = 1; // initialize leds
      switch(wave_current)         // preparing
      {
      case 0: LED_1 = 0; break;    // wave leds indicators
      case 1: LED_2 = 0; break;
```

```
            case 2: LED_3 = 0; break;
            case 3: LED_4 = 0; break;
            }
        }
        if (BUTTON_WAVE                 // wave selection button off
        &&  wave_debounce)              // if counter
            wave_debounce --;           // debounce

        if (! P3_6)                     // analog comparator test
            wave_buffer &= ~0x04;       // AR envelope off
        for (i = 0; i < 10 ;i ++)       // delay
        {   i = i;
        }
    }
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
static void dco_ini(void)
{
    INT_DISABLE;                        // disable interrupts
    TMOD &= 0xF0;                       // clear timer 0 mode bits
    TMOD |= 0x01;                       // timer 0 in mode 1
    ET0 = 1;                            // enable timer 0 interrupt
    TR0 = 1;                            // enable timer 0 counter
    INT_ENABLE;                         // enable interrupts
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
static void dco_int(void) interrupt 1 using 2
{
static __data UCHAR w_i = 0;            // sample index
    INT_DISABLE;                        // disable interrupts
    TR0 = 0;                            // stop timer 0
    TH0 = frqdiv_hi;                    // load timer value high
    TL0 = frqdiv_lo;                    // load timer value low
    wave_buffer &= 0x0F;                // clear wave bits
```

```
    wave_buffer |= wave_sample[w_i ++]; // set wave sample
    P1 = wave_buffer;                   // output wave sample
    w_i &= 0x0F;                        // normalize interval 0 - 15
    TR0 = 1;                            // start timer 0
    INT_ENABLE;                         // enable interrupts
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
static void midi_ini(void)
{
    INT_DISABLE;
    TR1 = 0;                            // stop timer
    ET1 = 0;                            // disable timer 1 interrupt
    TI = 0;                             // clear transmit - not used
    RI = 0;                             // clear receiver interrupt
    SCON = 0x50;                        // mode 1 and enable receipt
    PCON = 0x80;                        // smod 1: baudrate doubler
    TMOD &= ~0xF0;                      // clear timer 1 mode bits
    TMOD |= 0x20;                       // timer 1 in mode 2
                                        // baud rate counter = 0xfc
    TH1 = (unsigned char)(256 - (24000000L / (16L * 12L * 31250L)));
    ES = 1;                             // enable serial interrupt
    TR1 = 1;                            // enable counter timer 1
    INT_ENABLE;
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
static void midi_int(void) interrupt 4 using 1
{
    static __data UCHAR note_temp;      // table notes index
    __data UCHAR aux_command;           // midi command


    RI = 0;                             // reset int. MUST BE 1rst!
    input_byte = SBUF;                  // store input byte
    aux_command = input_byte & 0xF0;    // clear channel
```

```
    if (aux_command == NOTE_ON)         // midi command parsing
    {   midi_state = 1;
        return;
    }
    if (aux_command == NOTE_OFF)        // midi command parsing
    {   midi_state = 4;
        return;
    }
    if ((midi_state == 1)               // midi note number parsing
    ||  (midi_state == 4))              // for both note on and off
    {   midi_state ++;                  // appoint to next state
        note_temp = input_byte;         // store future note number
        return;
    }
    if (midi_state == 2)                // velocity value parsing
    {   midi_state ++;                  // appoint to next state
        if (input_byte)                 // if greater 0 is note on
        {   wave_buffer |= 0x04;        // AR envelope on
            note_count ++;              // increment note on counter
            note_next = note_temp;      // store future note number
        }
        else                            // if velocity 0 is note off
            note_count --;              // decrease note on counter
    }
    if (midi_state == 5)                // velocity value parsing
    {   midi_state ++;                  // for note off command
        note_count --;                  // decrease note on counter
    }
    wave_buffer =                       // set ASR envelope
    (note_count) ? wave_buffer | 0x08   // ASR 1
                 : wave_buffer & ~0x08; // ASR 0 when notes off
}
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

REFERENCES

[1] Vail, M. (2013). *The Synthesizer*. Oxford University Press

[2] Wilson, R. (2013). *Make: Analog Synthesizers. A modern approach to old-school sound synthesis.* Maker Media

[3] Hewitt, A. (2018). *How to Program Any Synthesizer*. Stereo Output

[4] Shepard, B.K. (2013). *Refining Sound, A Practical Guide to Synthesis and Synthesizers.* Oxford University Press

[5] Williams, A. (2003) Build Your Own Printed Circuit Board. McGraw-Hill Education

[6] Liening, J., Scheible J. (2020), Fundamentals of Layout Design for Electronic Circuits. Springer

[7] Jenkins, M. (2019). *Analog Synthesizers: Understanding, Performing, Buying: From the Legacy of Moog*. Routledge

[8] Allen, J.A. (2018). *Music Theory for Electronic Music Producers: The producers guide to harmony, chord progressions, and song structure in the MIDI*. Slam Academy

[9] Zorilla, D.M. (2008). *Synthesizers: A Brief Introduction* .Universitat Oberta de Catalunya

[10] Staszewski, R.B. Leipold, D. Muhammad, K. Poras, B. (2003*). DCO Based Architecture for RF Frequency Synthesis*. IEEE Explore

[11] Holmes, T. (2015) *Bucla Analog Synthesizers*. Routlegde

[12] Platt, C. (2009), *Make: Electronics*. Maker Media

[13] Krekovic, G. , Poscic, A. (2013), *Controlling a Sound Synthesizer Using Timbral Attributes*. University of Zagreb

[14] Anand, V., Singh V., Ladwal,K. (2019) *Study on PCB Designing Problems and their Solutions.* IEEE Xplore

[15] Asparuho, K., Shehova, D., Lyubomirov, S. (2018) *Using Proteus to Support Engineering Student Learning: Microcontroller-Driven Sensors Case Study*, IEEE Xplore

[16] Rizvi, S.R.(2011), *Microcontroller Programming: An Introduction*, CRC Press

[17] Freke, O.(2021), *Synthesizer Evolution: From Analogue to Digital*,Velocity Press

[18] Schmidt, R. (2010), *Robin Schmidt's Music Engineering Tools*, RS-MET