

Otto's DIY Cookbook

Vol. 1: The basics of analog design

Table of Contents

Release note.....	3
Introduction.....	3
Workflow.....	4
Absolutely essential tools.....	4
Lesson 1: Op-Amp Basics.....	5
Prerequisites.....	5
Basic use-cases for an op amp.....	5
Understanding op-amps.....	6
Pinout.....	6
Software base model of an op amp.....	7
Op amp circuit examples.....	8
Voltage follower (buffer).....	8
Non-inverting voltage amplifier.....	9
Conclusion.....	10
Chapter 1: Simple VCO.....	11
Introduction.....	11
Base Schematic.....	11
Analyzing the circuit.....	12
Charging phase.....	12
Discharging phase.....	13
Design considerations.....	14
Optimizing for Eurorack.....	14
Characteristics of the VCO.....	15
The output waveforms.....	15
LFO mod.....	16
Multi-Core oscillator mod.....	17
Example Implementation.....	18
Parts list.....	19
User interface.....	19
Drilling the front panel.....	20
Soldering the panel mount components.....	21
Miscellaneous circuits.....	22
Building the oscillator.....	23
Testing the oscillator.....	24
Connecting the user interface.....	24
Mechanical assembly.....	25
Let's make some noise.....	25

Release note

This is the initial release of the Otto's DIY Cookbook containing one lesson on operational amplifiers and a VCO build. We included the Op-Amp Basics lesson because the VCO build heavily relies on that knowledge. As the document evolves in the future we plan to include more theory based lessons as well as more builds.

We are happy to incorporate user contributions into this document, both theory and builds. This document should be considered an example of a well documented build. We encourage users to document their builds accordingly, this allows external contributions to one day become part of this pool of knowledge.

Introduction

Most other DIY synth kits require you to solder everything in place, once your done the module either works or not. Every week we see dozens of failed attempts posted in forums threads or synth DIY groups. Even if you can find schematics for the kit, chances are you wont find what causes your module to malfunction. It's very sad to see people giving up on builds because it discourages them and others from building DIY kits.

Otto's DIY takes a different approach. We try to show you how to build simple functional circuits that can be used as building blocks for more complex projects. This way you will be able to understand a schematic drawing by identifying the functional blocks and analyzing their behavior independently. This is one of the crucial skills required if you wish to design your very own modules one day.

Good luck on your journey!

Workflow

As much as we would like to see people turning off their computers and focusing on building synthesizers, we cannot disregard the role of the powerful design and simulation tools available nowadays. We encourage everyone to familiarize themselves with one of the online or offline circuit simulators. This is a great way to quickly verify parts of the circuit or even the whole design before building it. There are a number of free circuit simulators, pick your flavor!

All of the circuits in the cookbook can be built using commonly available tools and through-hole (THT) electronic components that you will find at your local hobby electronics store. The circuits can be built on a breadboard for testing or directly on the Otto's DIY panels.

We recommend to build functional circuit blocks one by one, testing each one thoroughly before moving on to the next one. Fixing mistakes in a circuit block is much easier than trying to debug a finished build.

Absolutely essential tools

Here we compiled a list of the tools that you will need to get started with basic electronics and synth DIY. Please understand that these are NOT optional, you will need every single one of the listed items. You will find a good variety of all of these tools at your local hobby electronics outlet, but if you are just getting started you don't have to spend more than \$100 for the whole set. (Even cheaper if you order online)

You will need a soldering iron/station and solder. It's nice to work with a quality iron and good tips but if you are just starting out then we don't recommend getting a really expensive one. You will also need a desoldering pump. This will come in very handy when you have to remove a misplaced or damaged component.

Next you will need a digital multimeter. The DMM is your third eye when it comes to electronics. Its used to verify components, measure current and voltages at different parts of the circuit, an essential tool for debugging. I recommend getting an auto-ranging meter, possibly one that can not only measure resistors and diodes but capacitance as well.

You will also need a set of basic tools. A side-cutter to cut off component lead, small pliers to bend and break and tighten stuff, a set of screwdrivers and some solid core hookup wire or ribbon cables to make connections on the board.

An oscilloscope one of the tools that is not absolutely essential to get started with synth DIY, but a worthy investment late on. I recommend getting a simple 2 channel 30Mhz analog oscilloscope to get started, it is much better investment than getting one of those low-cost pocket digital scopes.

Lesson 1: Op-Amp Basics

Prerequisites

- Understanding the concept of voltages and currents.
- Understanding resistors and the usage of Ohm's Law.
- Practical understanding Kirchhoff's circuit laws.
- Understanding the definition of input and output impedance.

Basic use-cases for an op amp

- Amplifying and/or inverting signals
- Buffering signals
- Comparing two signals
- Adding or subtracting signals
- Creating oscillators
- Integrating and differentiating signals

These are just some of the most common use-cases. Understanding op amps is an inevitable skill for everyone aiming to effectively design, build and modify analog circuits. Analyzing op amp based circuits can be tricky, but following a couple of golden rules you will quickly understand the behavior of such circuits.

The aim of this lesson is to present the required theoretical knowledge to understand op amps. Through practical examples we will showcase some of the techniques of analyzing op amp based circuits.

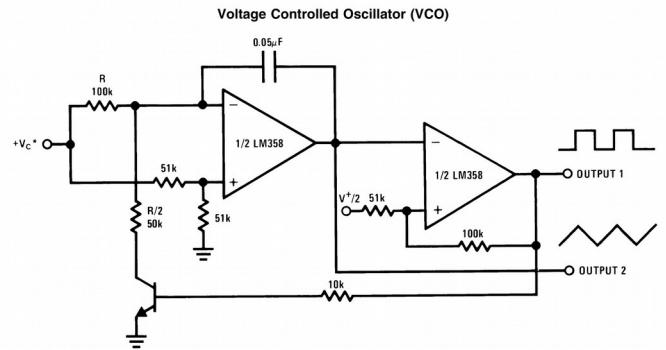


Illustration 1: Schematic drawing of an op amp based VCO circuit found in the LM358 datasheet. This is the base of the SimpleVCO chapter in the Cookbook.

Understanding op-amps

There are a number of more advanced properties that we could consider (ex. bandwidth, offset, phase-shift... etc) but for now the following three are the most important characteristics of an ideal op amp:

Rule 1: Infinite input impedance. This means that the current flowing in or out of the inputs is practically 0 at any given time. (The load on the input signal is insignificant, even if the source is fairly high impedance. This is usually a close enough approximation for most applications)

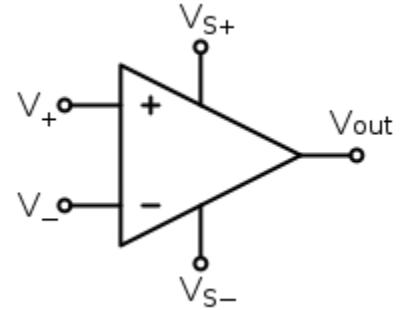


Illustration 2: Typical schematic symbol of the operational amplifier

Rule 2: Zero output impedance. This means that the output of an op amp is an ideal voltage source capable of driving any given load. (Depending on the op amp in question you will find that it can source or sink somewhere between 20-30 mA. This may not sound like much but it's more than capable of driving synth modules or even a pair of headphones.)

Rule 3: Infinite open-loop gain. This effectively means that if there is a small voltage difference between the two inputs ($V_+ > V_-$) then the voltage output of the op amp will very quickly rise to ∞ . (In practice this is obviously not possible, the output voltage will always be limited by the supply voltages).

Pinout

An op amp has typically two supply pins, two input pins and one output:

- Vs+ is the positive supply pin (ex. +12V)
- Vs- is the negative supply pin (ex. -12V)
- V+ is the non-inverting input pin
- V- is the inverting input pin
- Vout is the output pin

Sometimes there are two, four or even more op amps in a single IC¹. In these cases typically all of the op amps in the package share the same supply rails. For example the LM224 is a low-cost Quad op amp available in the DIP14 or SOIC14 packages.

¹ IC, Integrated Circuit or chip is typically a silicon based electronic circuit encapsulated in plastic, metal or ceramic packaging used as ready made functional blocks in circuit design.

Software base model of an op amp

A very descriptive way to understand op-amps is to look at this simplified software based model.

```
OPEN_LOOP_GAIN = 100000;
V_SUPPLY_POS    = 12;           //+12V
V_SUPPLY_NEG    = -12;          //-12V
```

```
while(1) {
    if (V_OUT < (V_IN_POS-V_IN_NEG)*OPEN_LOOP_GAIN) {
        if (V_OUT<V_SUPPLY_POS) {
            V_OUT++;
        }
    } else{
        if (V_OUT>V_SUPPLY_NEG) {
            V_OUT--;
        }
    }
}
```

As long as the non_inverting input (V_IN_POS) is greater then the inverting input (V_IN_NEG) the voltage on the output pin will increase rapidly. (Rule 3: Infinite open-loop gain)

Note that we can work with the input voltages without knowing the surrounding circuitry since the inputs are very high impedance (Rule 1: Infinite input impedance)

Op amp circuit examples

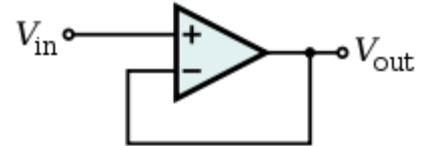
This are the most common op amp circuits, fundamental building of electronics. By analyzing the following circuits we can practice applying the above mentioned three golden rules to op amp based circuits.

Voltage follower (buffer)

If we have a high-impedance voltage source (or a signal that is very sensitive to load) we can use a voltage follower circuit to buffer it, effectively making it low-impedance (more robust). This way the output of the buffer can be loaded without effecting the input signal.

Buffering of an analog signal can be done using op amps configured as a unity gain voltage amplifier. This is often called a unity gain buffer or voltage follower.

The output is directly connected to the negative input of the op amp. When V_{in} is applied the opamp will drive its output to match the voltage of the negative input to the voltage of the positive input through the feedback loop.



Drawing 1: Schematic of the opamp buffer circuit.

$$V_{out} = V_{in}$$

```
while(1) {
    if (V_OUT < (V_IN_POS-V_IN_NEG) *OPEN_LOOP_GAIN) {
        if (V_OUT<V_SUPPLY_POS) {
            V_OUT++;
        }
    } else{
        if (V_OUT>V_SUPPLY_NEG) {
            V_OUT--;
        }
    }
    V_IN_NEG = V_OUT;
}
```

Note that $V_{IN_NEG} = V_{OUT}$; is the code equivalent connecting the output to the inverting input of the op amp.

Non-inverting voltage amplifier

The output will source current (increase its voltage) as long as the voltage on Inverting input is lower than the voltage on the non-inverting input. The trick is that in this circuit there is a voltage divider between the output and the non-inverting input.

Take a look at the following example with given initial parameters:

$$R_1=1k \quad R_2=1k \quad V_{in}=3V \quad V_{out}=0V$$

Note that in this example R_1 and R_2 have the same resistance, so the voltage on the inverting input is always half of the output voltage.

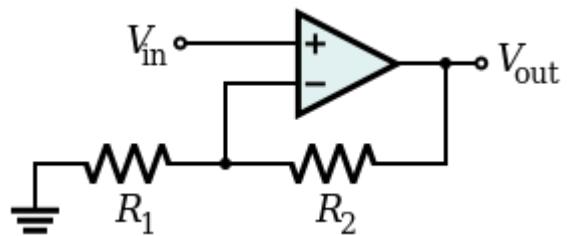
If V_{out} is 0V in the beginning then the inverting input is at 0V as well. 0V is less than 3V on the non-inverting input so the output will start sourcing current. The current is forced through R_2 and R_1 to Ground. The voltage increases on the output until the voltage difference between the inverting and the non-inverting input is 0V. At this point the Output voltage is 6V which is double of the input voltage. In this configuration the amplifier has a voltage gain of 2.

$$G=1+\frac{R_2}{R_1}=1+\frac{1k\Omega}{1k\Omega}=2$$

```

while(1) {
    if (V_OUT < (V_IN_POS-V_IN_NEG) *OPEN_LOOP_GAIN) {
        if (V_OUT<V_SUPPLY_POS) {
            V_OUT++;
        }
    } else{
        if (V_OUT>V_SUPPLY_NEG) {
            V_OUT--;
        }
    }
    V_IN_NEG = V_OUT * (R1 / (R1+R2));
}

```



Drawing 2: Schematic of the non-inverting amplifier circuit.

$$V_{out}=\left(1+\frac{R_2}{R_1}\right)*V_{in}$$

Conclusion

If you want to you can take a look at more examples on Wikipedia. Analyzing op amp circuits can only be done by carefully applying the rules explained above. We recommend taking a look at some of the interactive examples on Falstad.com. (Circuits/Op-Amps/)

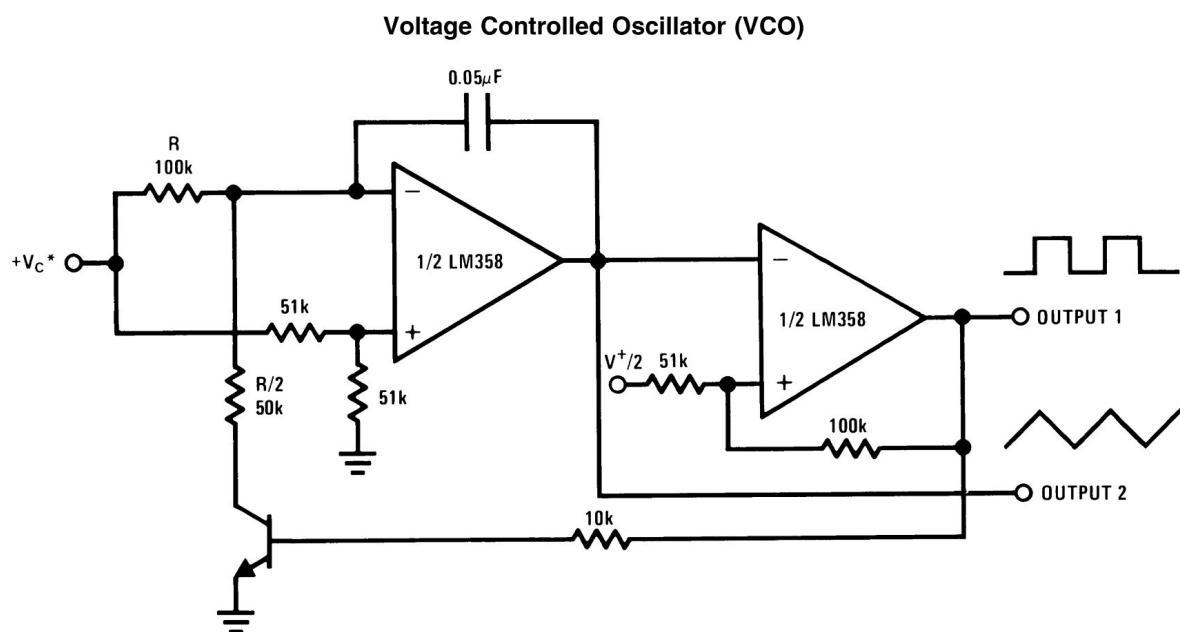
Chapter 1: Simple VCO

Introduction

In this project we build a simple VCO² using only commonly available electronic components. We only require two op amps and a couple of miscellaneous components to get started. Once the circuit is working, there are endless different modifications we can do to improve the design or change the behavior of the oscillator.

Base Schematic

This is one of the classic circuits found in many op amp datasheets and application notes. The input $+V_C^*$ is the control voltage (CV) that determines the oscillator's output frequency. The circuit has a square wave and a triangle wave output. This circuit is perfect for our purpose, it beautifully demonstrates the inner workings of oscillator circuits.



*WIDE CONTROL VOLTAGE RANGE: $0 \text{ V}_{\text{DC}} \leq V_C \leq 2 (V^+ - 1.5 \text{ V}_{\text{DC}})$

Illustration 3: Schematic drawing of the VCO circuit found in the "Typical application" section of the LM358 datasheet.

² VCO or Voltage Controlled Oscillator is a common module of Eurorack systems responsible for generating waveforms based on a control voltage.

Analyzing the circuit

This oscillator uses a single timing capacitor, this determines the frequency range. The output voltage range is determined by the supply rails. The square-wave output will have a rail-to-rail swing while the voltage of the triangle output is determined by the voltage divider generating the reference voltage for the comparator. The example in the op amp datasheet uses a single supply setup. For us it is easier to work with +12V and -12V supply voltages since the Eurorack standard specifies dual rail power distribution.

The behavior of the oscillator can be determined by analyzing the charging and discharging phase of the circuit independently.

Charging phase

OUTPUT 1 (Square) is HIGH

OUTPUT 2 (Triangle) is RISING

Q1 transistor is CLOSED (Conducting)

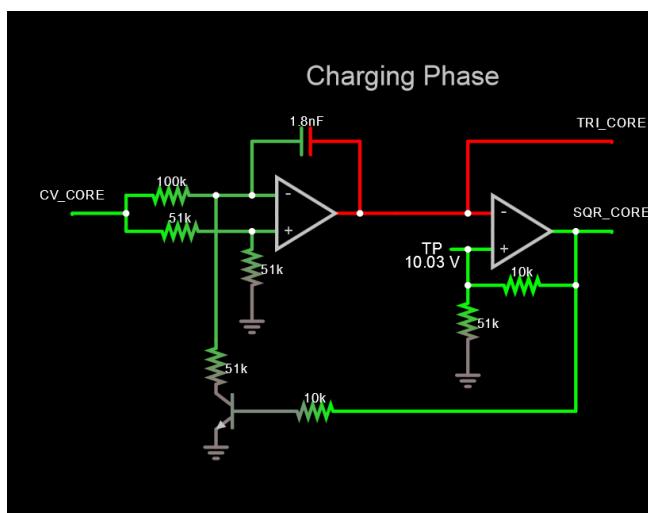


Illustration 4: Snapshot of the circuit in the simulator at the beginning of the charging phase with voltages represented by color.

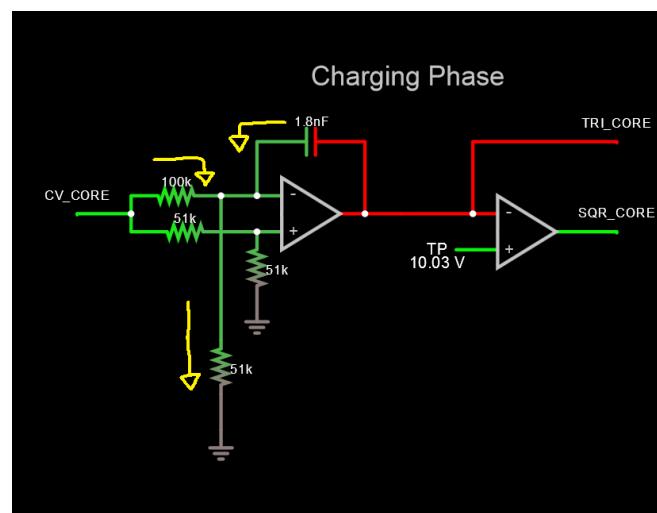


Illustration 5: Simplified equivalent circuit of the discharge phase with the yellow arrows representing the path of current through the timing capacitor.

Once this capacitor is charged up to the reference voltage of the comparator, OUTPUT1 transitions to LOW disabling the Q1 and starting the discharge phase.

Discharging phase

OUTPUT 1 (Square) is LOW

OUTPUT 2 (Triangle) is FALLING

Q1 transistor is OPEN (Not conducting)

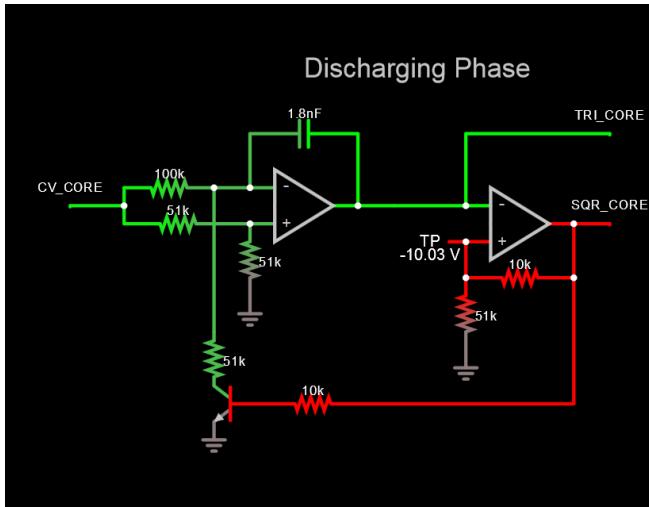


Illustration 6: Snapshot of the circuit in the simulator at the beginning of the discharging phase with voltages represented by color.

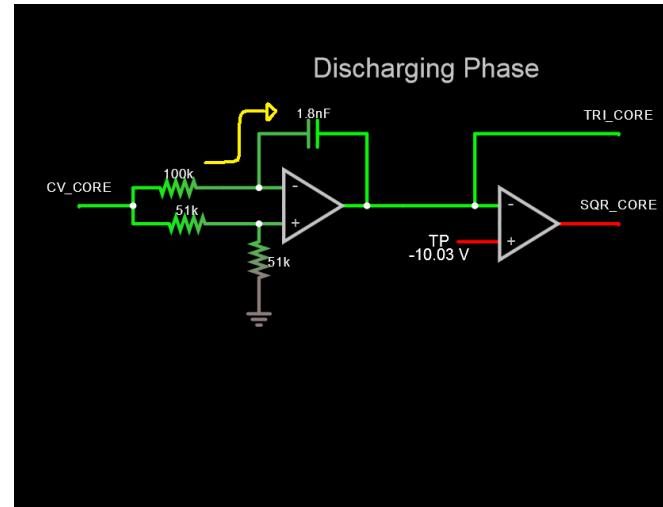


Illustration 7: Simplified equivalent circuit of the discharge phase with the yellow arrows representing the path of current through the timing capacitor.

The discharging continues until the comperator hits its new reference voltage. When this happens OUTPUT1 transitions to HIGH and the cycle starts again.

Note that the state of OUTPUT1 will have an effect both on the reference voltage of the comperator and when the transistor is conducting it will pull current from the negative input node of the first op amp causing the capacitor to discharge.

Visit falstad.com the check out this circuit using their free online simulator:

www.falstad.com/circuit/circuitjs.html → Op-Amps / Oscillators / Voltage-Controlled Oscillator

Design considerations

This circuit can be built on a breadboard, it works, it oscillates but there are a couple of things we have to consider in order to make a usable synthesizer oscillator from this building-block circuit.

Note that the output of the first op amp is directly connected to the main timing capacitor while the output of the second op amp is the voltage source driving the feedback loop. This means that these two nodes are responsible for the timing of the oscillator. Even though the output of an op amp is generally modeled as a low impedance voltage source, loading these outputs resistively and/or capacitively will slow down the oscillator (introducing unwanted detune) or cause it to fail completely. To avoid this we can add additional buffering to isolate the load impedance from the output of the oscillator.

The NPN transistor in the feedback loop is used as a basic switch, when U_{BE} (Base-Emitter voltage) is high current will flow through the transistor. In this case it is common to use an N-Channel MOSFET instead of the bipolar transistor to do the switching.

Optimizing for Eurorack

There are a couple of things we have to consider in order to successfully use this oscillator circuit in Eurorack. First its good to utilize the +12V and -12V power rails of the Eurorack, this way we can build an oscillator that generates signals without any DC offset. Next we have to decide the voltage levels of the outputs. In this circuit, the square-wave output is rail-to-rail, while the output voltage of the triangle wave can be modified by

changing the voltage reference of the comperator (tuning the values of the resistors in the positive feedback loop). Note that this will have an effect on the output frequency as well, since the timing capacitor will charge to a different voltage. Next we have to choose the capacitor value, this will determine the overall frequency range of the oscillator. Choosing a larger value will decrease the frequency range. Note that the circuit accepts positive control voltages (0V or any negative input voltage will cause the oscillator to stop). For synthesizer modules it is desirable to have very high input impedance. This way multiple modules can be connected to the same signal source without significantly loading it. Output impedance of modules should be somewhere between 470 to 10000 Ohms depending on the type of module.

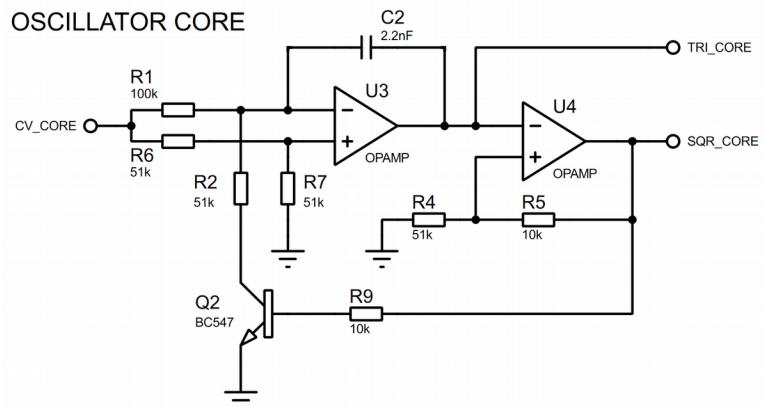


Illustration 8: The schematic drawing of the optimized oscillator core.

Characteristics of the VCO

Most synthesizer oscillators nowadays are designed to have exponential characteristics called Volt Per Octave control. This means that a 1V increase in the control voltage causes the oscillator to double its output frequency (+1 Octave).

Our oscillator however, has a linear voltage to frequency response often called Volt Per Hertz control. This means that a 1V increase in the control voltage causes the oscillator to change its output frequency by a set amount. In general most oscillator cores will have linear response, this is due to the fact that we use the charging and discharging of a capacitor for timing. When we charge a capacitor with constant current, the voltage across its terminals ($V(t)$) is going to be a linear function of time, the change in voltage divided by the change in time (dV/dt) is constant. Note that this is the same principle that makes the astable 555 timer circuits behave in a linear fashion too.

V [CV]	1	2	3	4	5	Volt
f [V/Oct]	55	110	220	440	880	Hz
f [Hz/V]	61	121	180	239	296	Hz

Table 1: Comparing the Voltage to Frequency characteristics of a linear and an exponential VCO.

A common way to make a linear VCO work with CV sources using exponential tuning is to use an Exponential V/I Converter circuit. This can be a separate eurorack module or built into the VCO itself.

In a later chapter we will take a look at creating such converter circuits. (Both analog and digital)

The output waveforms

This circuit has two outputs (A, B) (square and triangle wave respectively). These outputs have the same base frequency but the harmonic content varies.

[Illustration: Wavform A and B in the time domain]

[Illustration: Wavform A and B in the frequency domain]

Note that the square-wave contains much more hi-frequency content making it “harsh” compared to the mellow “humming” sound of the triangle-wave.

The signal level of the square-wave is -12V to +12V, while the triangle has a -10V to +10V swing.

LFO mod

An LFO or Low-Frequency Oscillator is a signal generator that is used to output periodic waveforms usually well below the audible frequency range. This can be used to modulate filters, amplifiers or oscillators. The frequency range of our oscillator can be configured by replacing the main timing capacitor in the circuit.

Note that a DPDT Switch can be used to switch between two different timing capacitors on the fly.

Input CV (V)	0	1	2	3	4	5
Output (Hz) (C=2.2nF)	0	61	121	180	239	296
Output (Hz) (C=4.7nF)	0	28	55	83	110	138
Output (Hz) (C=10nF)	0	13	26	39	52	65
Output (Hz) (C=22nF)	0	6	12	18	24	30
Output (Hz) (C=47nF)	0	2.9	5.9	8.9	11.8	14.8
Output (Hz) (C=100nF)	0	1.3	2.6	3.9	5.2	6.5

Table 2: Frequency range of the oscillator using different E3 series standard capacitor values for timing.

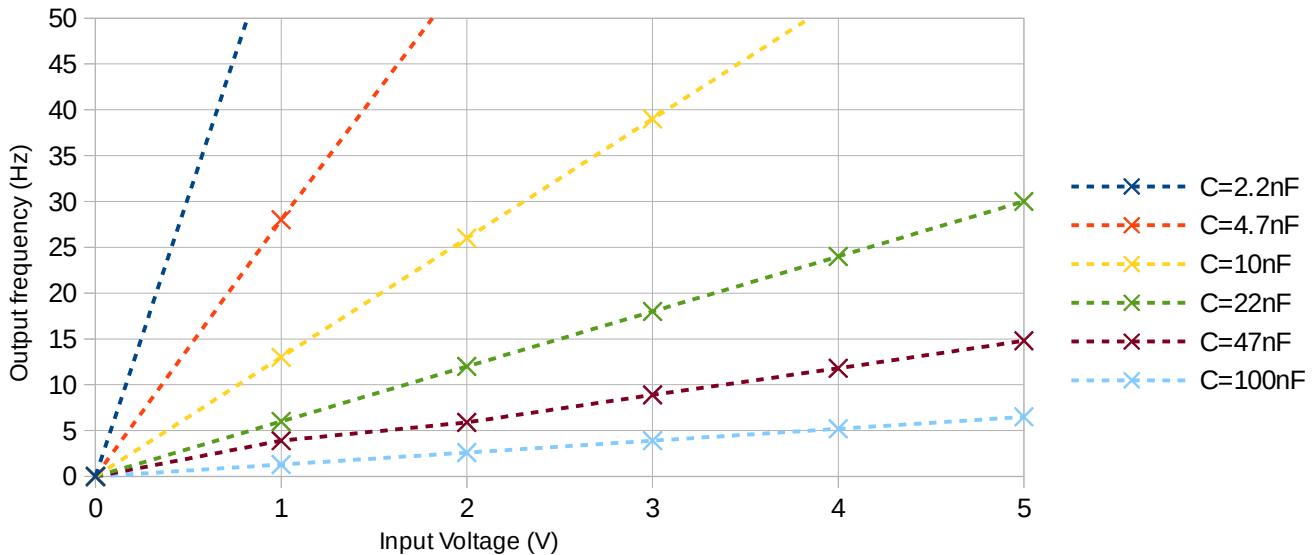
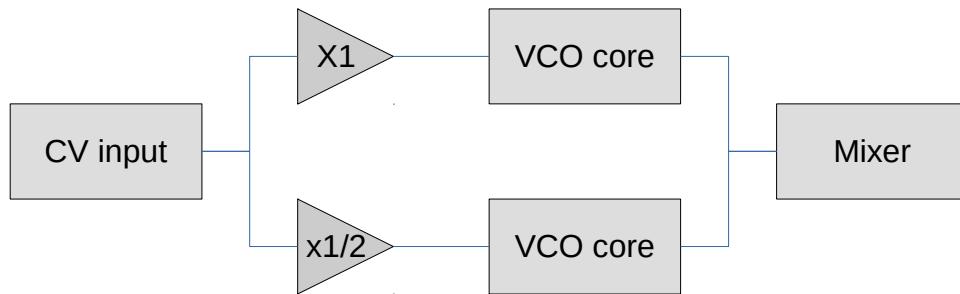


Illustration 9: The transfer characteristics of the voltage to frequency converter using different timing capacitors.

Multi-Core oscillator mod

One of the advantages of using linear oscillators is that we can very easily add multiple oscillator cores to create overtones. Feeding the identical oscillator cores with different control voltages will cause them to output different frequencies. For example if we feed 5V CV into one of the cores and feed 2.5V to an other, the second oscillator's output frequency will be exactly half of what the first core outputs. The pitch difference between the two core will be one octave. This can be done using two resistors as a voltage divider before each oscillator core. To eliminate any detune caused by this added impedance, we can buffer the CV directly before the oscillator core. Mixing oscillators with different base frequencies or introducing deliberate detune between two oscillators can result in very interesting sounds, the possibilities are endless.



Example Implementation

Summarizing all of the great thing we just learned, we can start building our very own synth module. In this example we create a basic two channel oscillator with independent CV and GATE inputs. Both channels have their own TUNE control allowing them to be tuned separately. The output of each channel has a BLEND control that determines the mix between the SQUARE and TRIANGLE outputs. There are two momentary switches per channel on the front panel, a TRIGGER for manually opening/closing the gate and an OCTAVE mode button which halves the output frequency of the given oscillator channel.

For this module we will use three Protoboard-480 modules and a Powerbrick. The first is going to serve as a front panel, the second layer accommodates the user interface while the third PCB contains the oscillator circuitry. The Powerbrick will safely power up the build while minimizing the possibility of permanently damaging components due to common prototyping mistakes.

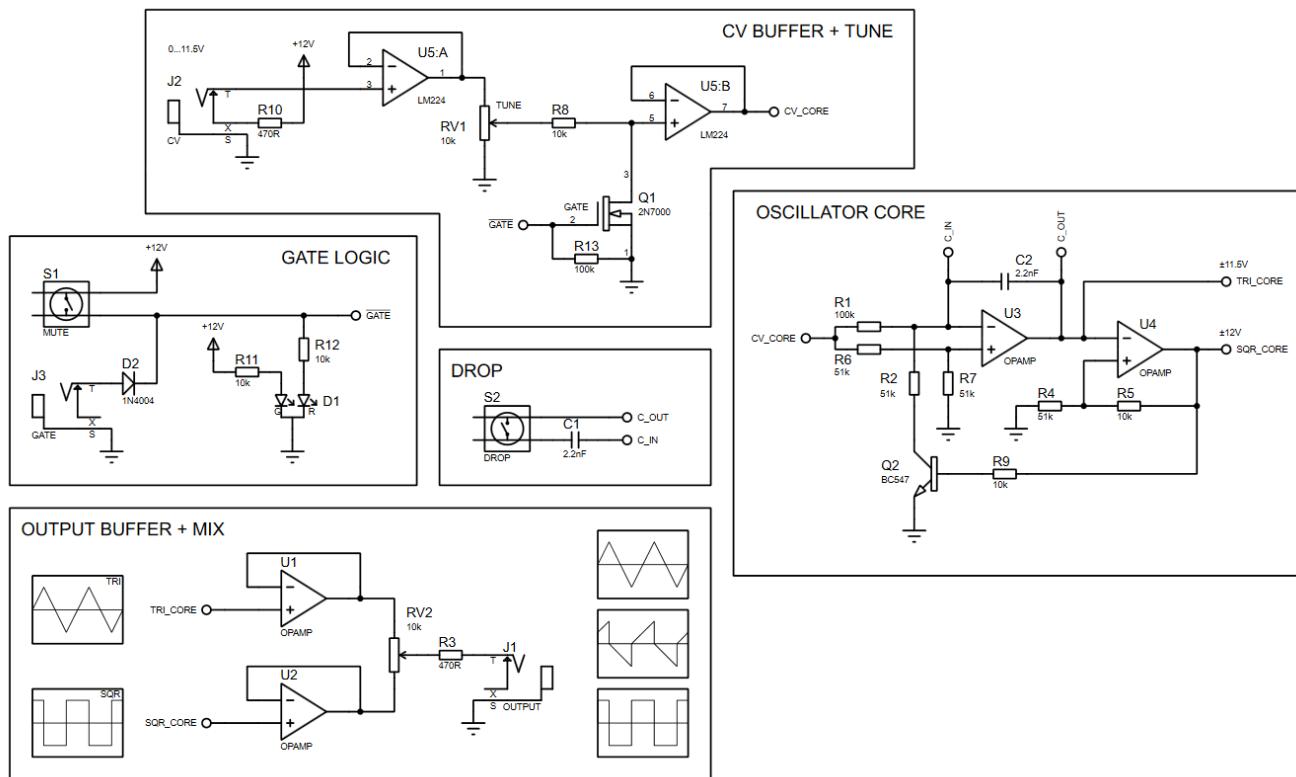


Illustration 10: Schematic drawing of a single oscillator channel. Check out this circuit in the Falstad simulator at <http://tinyurl.com/yb59wsxm>

Parts list

- 6x TS socket (Thonkicon style)
- 4x B10k potentiometer (TT-Electronics)
- 4x Momentary pushbutton (Ninigi)
- 2x Common cathode bi-color LED (Lucky Light)
- 5x LM224 Quad Opamp
- 2x 2N7000 N-Channel MOSFET
- 2x BC549 NPN Bipolar transistor
- 5x 470R 1/4W Carbon-film Resistor
- 10x 10k 1/4W Carbon-film Resistor
- 10x 51k 1/4W Carbon-film Resistor
- 10x 100k 1/4W Carbon-film Resistor

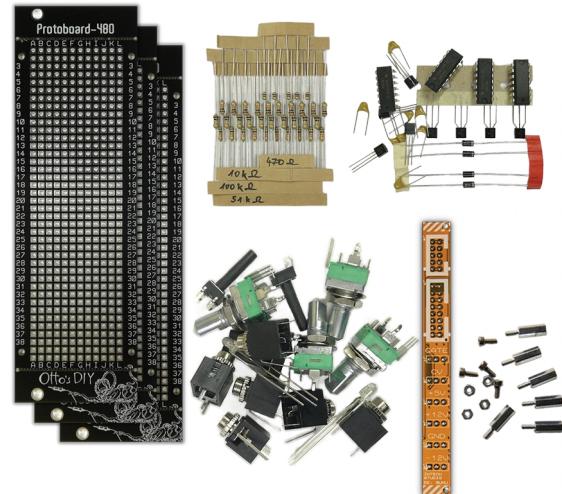


Illustration 11: Contents of the kit for this project

User interface

The *TUNE* control allows fine adjustment of the frequency range.

The *BLEND* control mixes the triangle and square outputs.

The *TRIGGER* button mutes the oscillator.

The *DROP* button decreases the oscillator frequency by one octave.

The external *GATE* input can be used to mute the oscillator. (Active-low logic → High level mutes)

The external *CV* input provides linear control of the pitch between 0 Hz and 296 Hz. (0-5V CV)

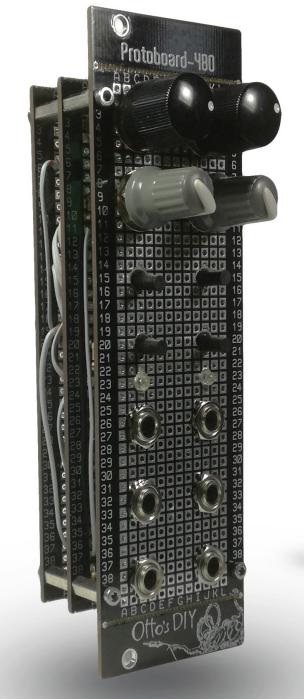


Illustration 12: Photo of the finished oscillator module.

Drilling the front panel

The first we have to mark and drill the required holes for the potentiometers, buttons, jacks and LEDs.

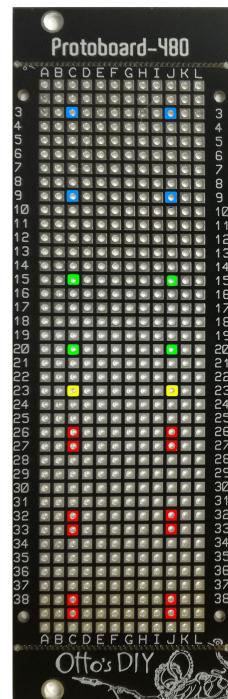
Component	Potentiometers	Buttons	LEDs	Jacks
Part number	P0915N-FC1BR10K	TACTM-619P-F	LL-309VUGM2E-V11A-2B	PJ398SM
Drill diameter	7mm	3.5mm	3mm	6mm
Drill position	C3 C9 J3 J9	C15 C20 J15 J20	C23 J23	C26.5 C32.5 C38.5 J26.5 J32.5 JC38.5

Note that for the jack sockets it is best to drill between two holes of the raster. It can be done by using drill bits designed for working with wood since these types of bits have a really sharp point.

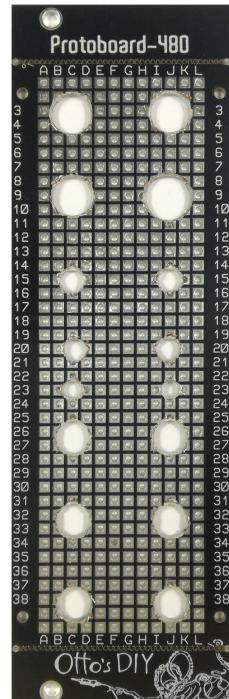
Alternatively, you can used drill bits for metal, drill both of neighboring holes on the panel couple of millimeters at a time. Try to create a 3*5mm oval cutout by carefully connecting the drilled holes.

In both cases, start by drilling a 3mm hole first, this way you can correct any misalignment later. Slowly increase the drill diameter until you reach the desired hole size.

Using a stepped drill bit can significantly speed up the drilling process.



*Illustration 14:
Drill holes marked
on the front panel*



*Illustration 13:
Finished front
panel*

Soldering the panel mount components

The next step is to mount the all of the user interface components to the second PCB.

Break off the top end bottom edges of the Protoboard, otherwise the module will not fit into the rack correctly. Identify the top and bottom sides of the board, make sure to use the correct orientation (The bottom side has ten easy-access ground points).

Install all of the components without soldering any of the leads. It's important to check if the layout of the user interface matches the layout of the front panel.

The potentiometers have a position alignment pin that we have to break off with a pair of needle-nose pliers or side-cutters. It is also important to align all of the potentiometers in the same orientation, so we only need one type of D-shaft knob later on.

The legs of the push buttons must be bent outwards to match the 5*3 raster holes. This can be a little bit tricky, take your time to double check the alignment with the front panel.

The LEDs provided in the kit are bi-color (red and green) common cathode type. The middle pin is the common cathode the other two are the anodes for each color. Check the pinout with a digital multimeter before soldering.

If everything aligns perfectly, we can start soldering the components.



Illustration 15: The potentiometer included in the kit.

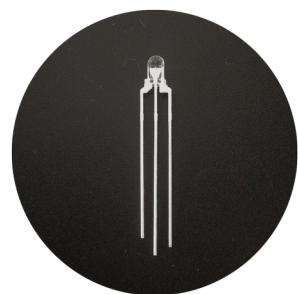


Illustration 16: Bi-color led from the kit. Pinout from left to right: Anode (green), Cathode (common), Anode (red)



Illustration 17: All of the panel mount user interface components are test fitted before soldering.

Miscellaneous circuits

The resistors for the input and output jacks, the current limiting resistors for the LEDs and the diode for the gate logic can all be soldered directly to the user interface board to save space on the oscillator board later. Any component that has a small footprint and is not essential for the operation of the VCO core should be on this board. This makes it easier to test the functional blocks of the circuit one by one.

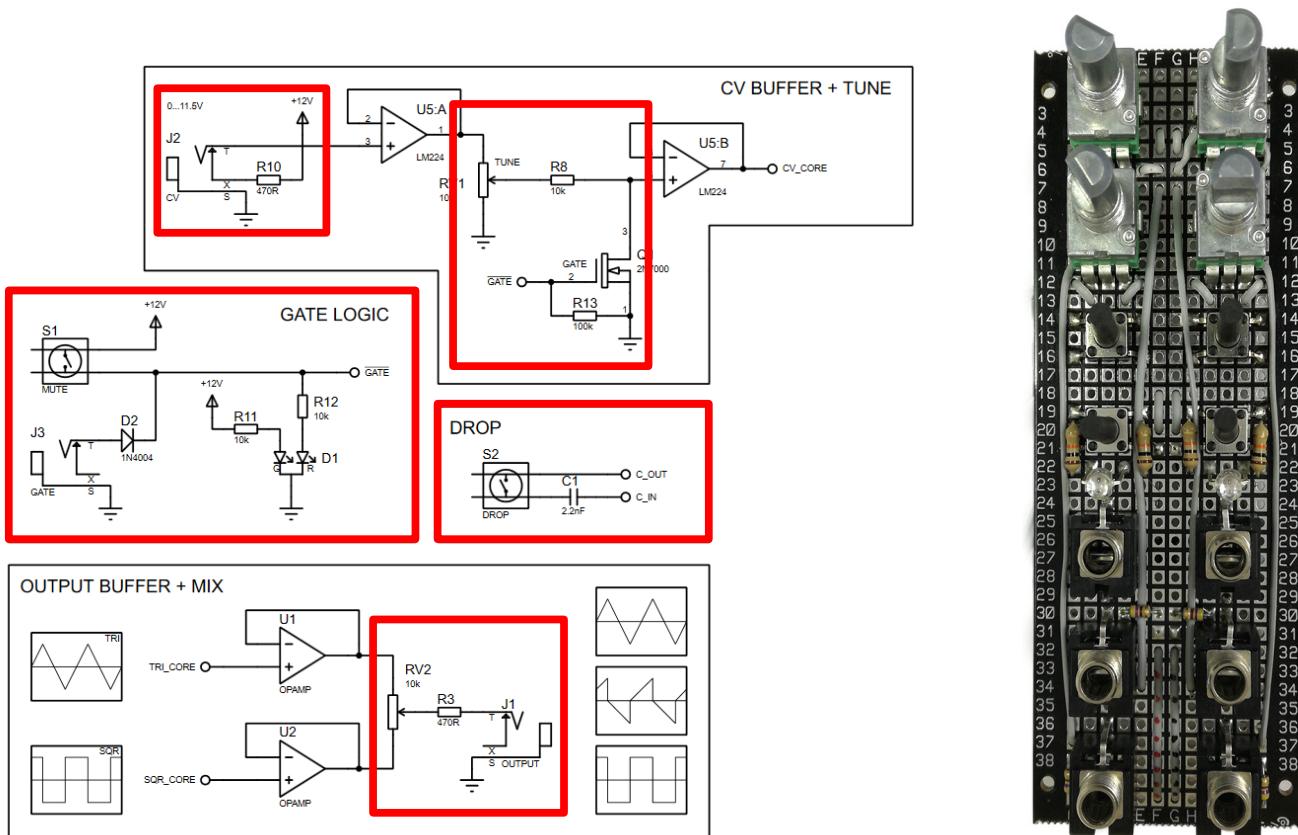


Illustration 18: Schematic drawing of a single oscillator channel. The red rectangles mark the components that are soldered onto the user interface board.

Building the oscillator

Now that we finished the user interface we can start assembling the oscillator circuitry. It is important to test the user interface and the oscillator separately to make sure both of them function correctly. It is much harder to trace signals once all of the jacks and potentiometers are connected to the core.

The circuit has two identical oscillator channels, they can be built and tested separately. It is easiest to build the “Oscillator Core” section of the schematic first. This can be tested with a positive control voltage applied to the CV_CORE input. You can check the outputs using an oscilloscope. The TRI_CORE signal should be a symmetrical triangle wave while the SQR_CORE is a square wave with the same frequency. Carefully check the duty cycle of the signals, both should be very close to 50%.

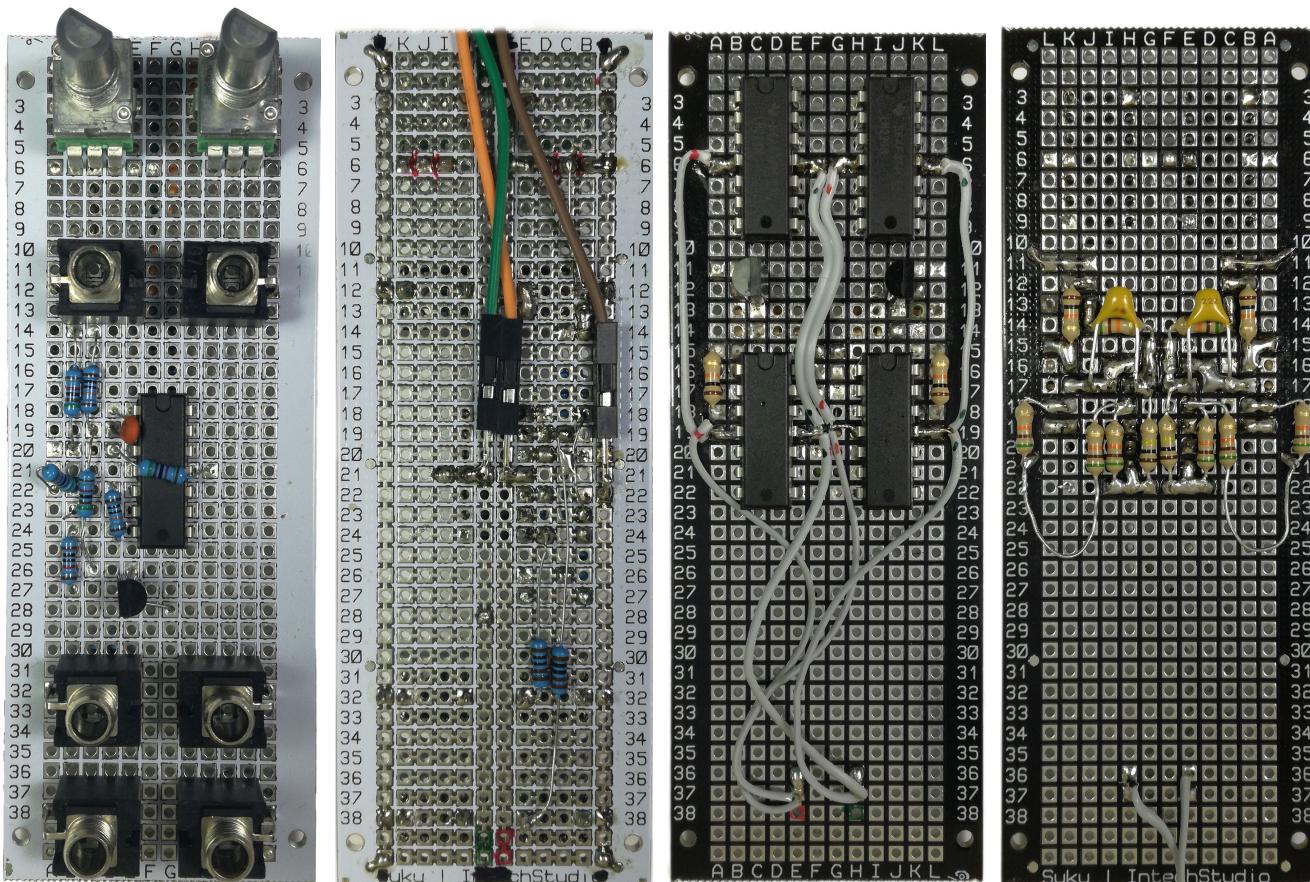


Illustration 19: Two different realization of the oscillator core. On the left is a simple single channel VCO prototype with the user interface and the circuit on the same panel. On the right is the final dual channel build of the SimpleVCO circuit. The component density is fairly high in this build, some previous veroboard prototyping experience is required.

Testing the oscillator

It is recommended to work on one channel at a time. Once one of the cores are built we can start verifying its behavior. Apply +12V and -12V to the power rails and +5V to the CV input using to Powerbrick module. The square and triangle outputs should produce symmetrical waveforms with a frequency of ~300Hz. If for some reason the oscillator does not work try the following debug procedure:

1. Check the schematic one more time. Count the components in your build.
2. Verify the voltage on the supply pins of the op amp using a digital multimeter.
3. Verify that the control voltage is applied to the correct node.
4. Verify that the voltage on the inverting and non-inverting inputs of the timing opamp. These should be exactly half of the control voltage.
5. Check the pinout of the transistor.
6. Check the circuit in the Falstad simulator, seeing the circuit in action with animated voltages and currents can help to find mistakes in your build.
7. If you are stuck try posting detailed photos and description of the problem to the Otto's DIY User Group.

Connecting the user interface

Once both of the channels are working as intended, we can start connecting the user interface board to the oscillator core. It is easiest to connect all of the wires to the user interface board first. Start connecting the user interface components to the oscillator core one by one, testing the circuit after each connection.

1. Connect the buffered outputs to the mix potentiometers.
2. Connect the CV input jack to the CV buffer input.
3. Connect the CV buffer output to the TUNE potentiometer.
4. Connect the TUNE potentiometer to the CV_CORE buffer.
5. Connect the gate logic and the DROP button with its capacitor.

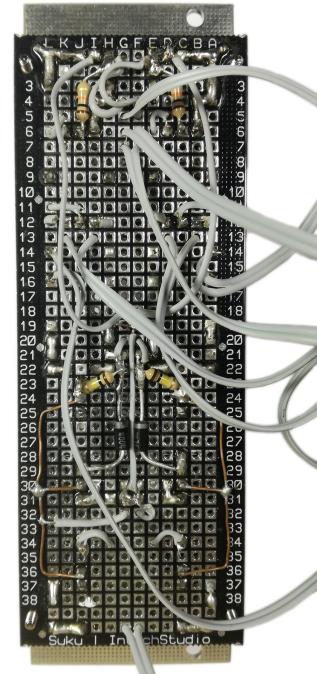


Illustration 20: Jumper wires connected to the back of the user interface board.

Mechanical assembly

Once everything is working it is time to assembly the module using the included hardware pack. Starting from the front panel, using the M2 bolts secure the four standoffs. Then the user interface board and the oscillator board with four standoffs each. Be careful not to damage any of the jumper wires during this procedure. The last step is to connect the Powerbrick using the two remaining standoffs and the four included M2 nuts.

Let's make some noise

Congratulations, you finished your SimpleVCO build. You can use this module as a dual channel sound source, interval generator or LFO, you can modulate one channel with the output of the other etc. This is a very versatile module and a great introduction into analog oscillator circuits.

We encourage you to remix this circuit, try modifying it and make sure to share your builds with the community.

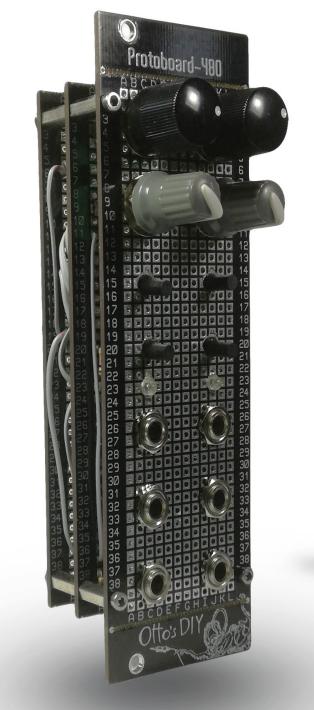


Illustration 21: Photo of the finished oscillator module.