INTRO | mki x es.edu

Hey there, thanks for buying this DIY kit! We – **Erica Synths** and **Moritz Klein** – have developed it with one specific goal in mind: teaching people with little to no prior experience how to design analog synthesizer circuits from scratch. So what you'll find in the box is not simply meant to be soldered together and then disappear in your rack.

Instead, we want to take you through the circuit design process step by step, explaining every choice we've made and how it impacts the finished module. For that, we strongly suggest you follow along on two **breadboards[1],** which are non-permanent circuit prototyping tools that allow you to experiment and play around with your components. To help you with this, we've included suggested breadboard layouts in select chapters.

In addition to this, you can also play around with most of the chapter's circuits in a **circuit simulator** called CircuitJS. CircuitJS runs in your browser. You'll find weblinks in the footnotes which will direct you to an instance that already has example circuits set up for you. We strongly encourage you to fiddle with the component values and general structure of those circuits to get a better understanding of the concepts we're laying out.

Generally, this manual is intended to be read and worked through front to back, but there were a few things we felt should go into a dedicated appendix. These are general vignettes on electronic components & concepts, tools, and the process of putting the module together once you're done experimenting. Don't hesitate to check in there whenever you think you're missing an important piece of information. Most importantly though: have fun!

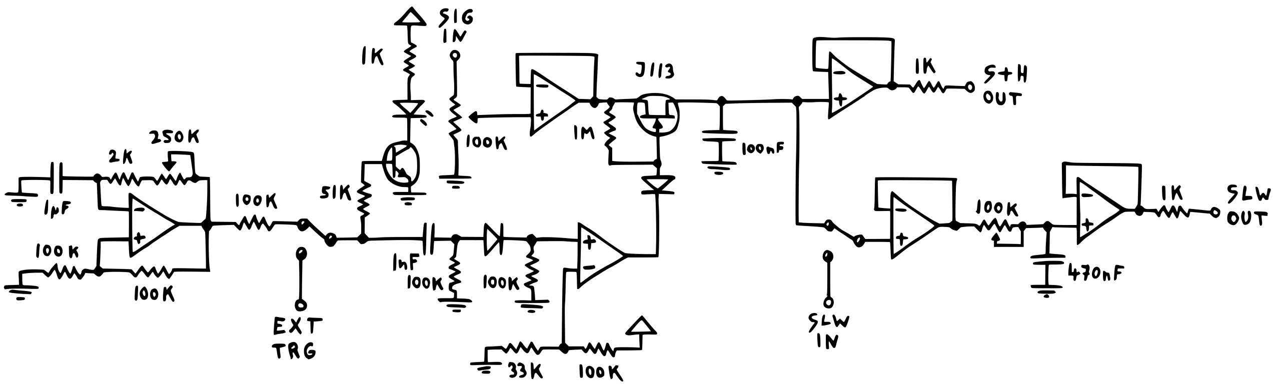
TABLE OF CONTENTS

CIRCUIT SCHEMATIC | S&H 2
CIRCUIT SCHEMATIC | NOISE 3
BILL OF MATERIALS 4
POWERING YOUR BREADBOARD 7
CIRCUIT DESIGN CLOSE-UP | S&H 8
CIRCUIT DESIGN CLOSE-UP | NOISE 29
COMPONENTS & CONCEPTS APPENDIX 42
TOOLS APPENDIX 55
MODULE ASSEMBLY APPENDIX 58
SOLDERING APPENDIX 69

[1] Note that there are no breadboards included in this kit! You will also need a pack of jumper wires and two 9 V batteries with clips. These things are cheap & easy to find in your local electronics shop.

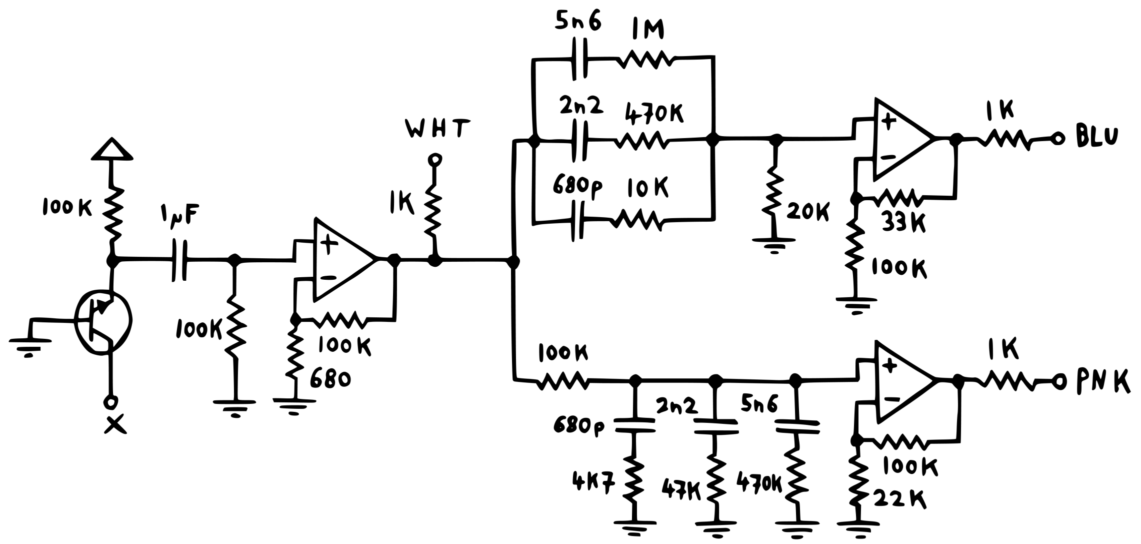
THE mki x es.edu SAMPLE & HOLD

Wether you use it for self-sequencing, for adding a dash of randomness to your filter movements or for rough analog bitcrushing, a sample & hold-circuit can often be the secret sauce that takes your patch from decent to distinctive. That’s why I came up with this super simple JFET-based implementation – and then upgraded it with a couple extra features like an internal clock generator or a dedicated slew output.



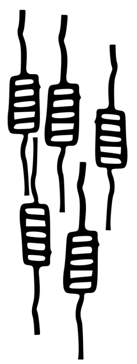
THE mki x es.edu NOISE GENERATOR

Of course we'll need an interesting input to sample – which is why you'll often find sample & hold modules with an integrated noise generator. So we'll do the same! But instead of just one noise color, we'll throw in two extra: pink and blue in addition to the standard white noise.



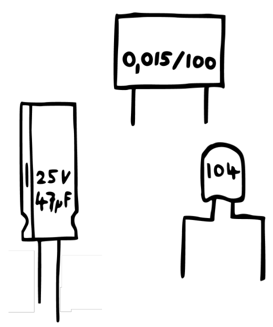
BILL OF MATERIALS

Before we start, please check if your kit contains all of the necessary components. In addition to a PCB, panel and power cable, your box should also contain:



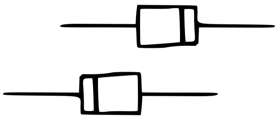
A ton of resistors. The specific values (in ohms, which you should check for with a multimeter) are

1M	x2
470k	x2
100k	x12
51k	x1
47k	x1
33k	x2
22k	x1
20k	x1
10k	x1
4k7	x2
2k	x1
1k	x6
680 Ω	x1
10 Ω	x2



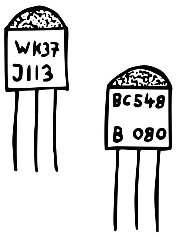
A bunch of capacitors. The specific values (which are printed onto their bodies) are

47 µF (electrolytic)	x3
1 µF (1J63/foil)	x2
470 nF (47k63/foil)	x1
100 nF (104/ceramic)	x8
100 nF (1J100/foil)	x1
5.6 nF (5n6J/foil)	x2
2.2 nF (2n2J100/foil)	x2
1 nF (102/ceramic)	x1
680 pF (680/foil)	x2



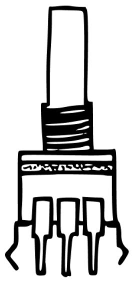
Some diodes. The specific model names (which are printed onto their bodies) are

SB140[2] (schottky) x2
1N4148 (signal) x2



A couple of transistors. The specific model names (which are printed onto their bodies) are

2N3904 (NPN) x1
BC548 (NPN) x1
J113 (JFET) x1

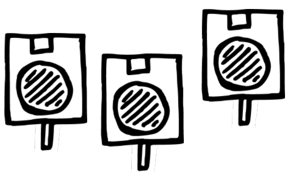


A couple regular potentiometers. The specific values (which may be encoded & printed onto their bodies) are

250k (B254) x1
100k (B104) x2

An LED (light emitting diode). The specific model (which you can identify by measuring their body's width) is

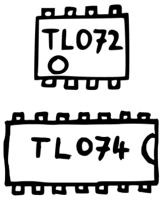
3mm (red) x1



A bunch of jack sockets. The specific models (which you can identify by their color) are

Switched mono (black) x8

2 Please note that these could also be a different model (e.g. 1N5818).



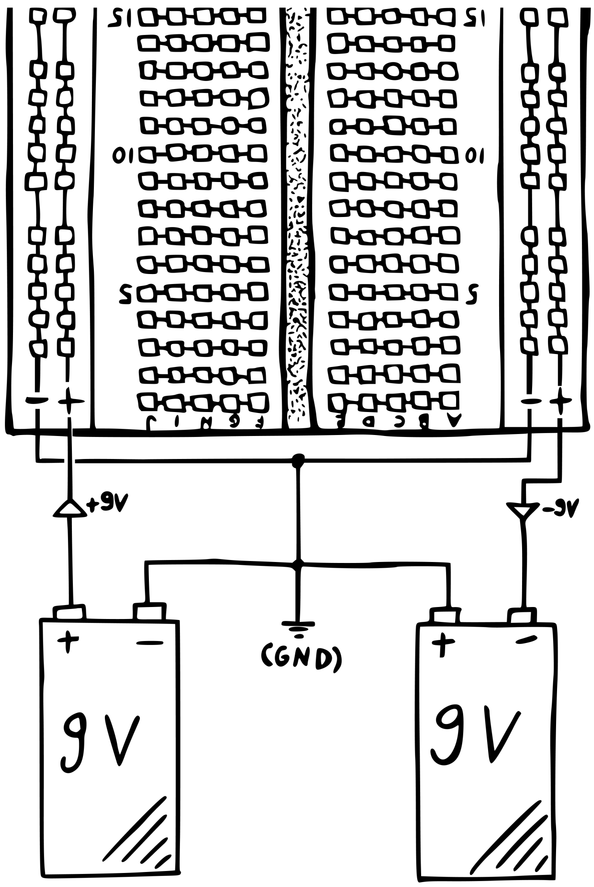
A couple of chips. The specific model (which is printed onto their bodies) is

TL074 (quad op amp) x2
TL072 (dual op amp) x1

You will also find a few sockets that are only relevant when assembling the module in the end.

POWERING YOUR BREADBOARD

Before we can start building, you'll need to find a way of providing your breadboard with power. Ideally, you'd use a dual 12 V power supply for this. Dual power supplies are great – and if you want to get serious about synth design, you should invest in one at some point. But what if you're just starting out, and you'd like to use batteries instead? Thankfully, that's totally doable. **You just need to connect two 9 V batteries to your breadboard like shown here.**[3] For this, you should use 9 V battery clips, which are cheap & widely available in every electronics shop.



By connecting the batteries like this, the row on the left side labeled **+** becomes your positive rail, the row on the right side labeled **+** becomes your negative rail, and both rows labeled **–** become your ground rails.[4]

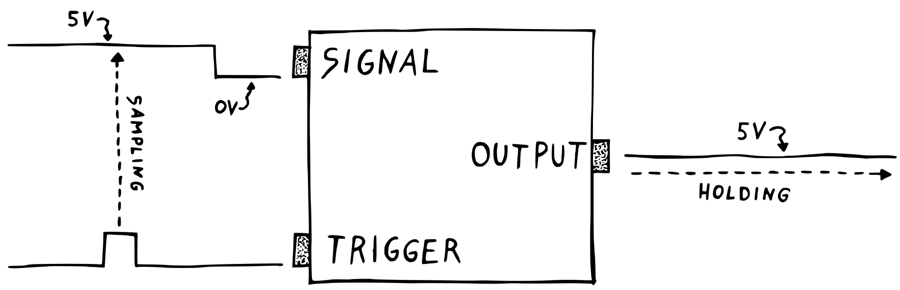
Please make sure you disconnect the batteries from your breadboard when you make changes to the circuit! Otherwise you run the risk of damaging components.

[3] Since all circuits in this manual were designed for a 12 V power supply, we assume that to be the default. Everything will still work roughly the same with 9 V, though.

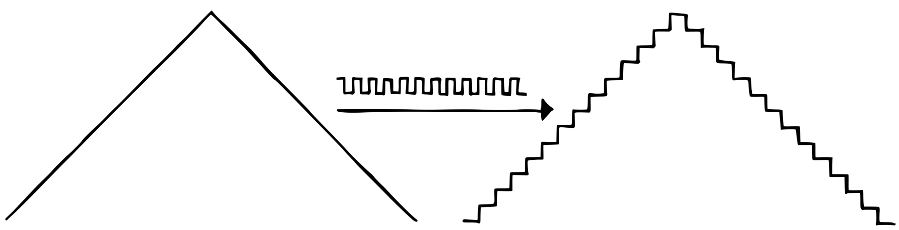
[4] This is a bit awkward because breadboards weren't really made with dual supply voltages in mind.

SAMPLE & HOLD BASICS

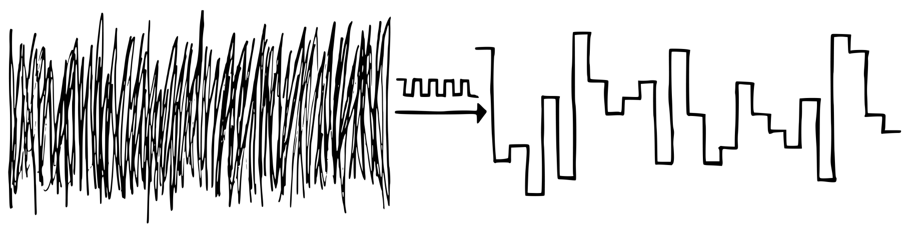
Even though the name might sound somewhat intimidating, designing a sample & hold-circuit is actually quite straightforward and simple. Before we can start with that though, we'll have to make sure we understand what it does, exactly. Thankfully, the name is actually quite helpful here. **At its core, a sample & hold-circuit does two things: sampling and then holding a voltage**.



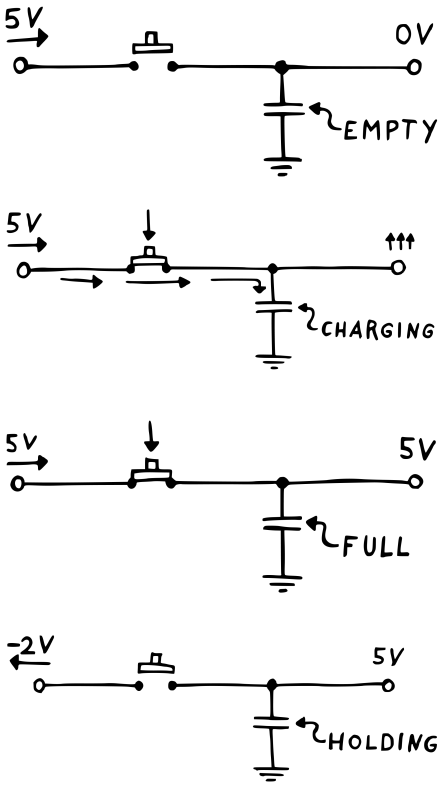
For that, it has two inputs – one that gets the to-be-sampled voltage and another that gets a trigger signal initiating the sampling process. At the output, the circuit should then provide a copy of the sampled voltage until the next trigger comes in (i.e. holding it).
If that description's too abstract for your taste, let's look at this with an example. We'll assume we have two input signals. The first is a slow triangle oscillation, which we'll feed into our circuit's sampling input. The second is a significantly faster square wave oscillation, which we'll plug into the circuit's trigger input.



The output then looks like a repeating set of stairs going up and down. **So instead of the continuous slope of the input triangle wave, we get a kind of digital approximation**. If you do this with white noise instead of an LFO, you'll get weird, random voltages at the output instead.



And while implementing this might seem like a tough job – at its core, the mechanics of a sample & hold-circuit are dead simple. We only need two elements: some sort of switch – and a capacitor.[5]



To start out simple, let's take a look at the most naive implementation: using an actual manual momentary switch.[6] Here's how it works. We apply our to-be-sampled voltage to the switch, while connecting the other end to the capacitor. Before we push down the button, the capacitor will be empty, so the voltage on the right is going to be 0 V.

As soon as we push the button, a current will flow through the switch and into the capacitor, charging it up. The same goes in reverse for a negative input voltage. Once the voltages on both sides are equal, that current flow will stop. We've sampled the input voltage.

If we now release the button, the voltage on the right will stay exactly the same, no matter what happens on the left – since we've cut the connection. Meaning that we're holding the sampled voltage.

Only trouble is that we're not interested in manual control here – instead, we want to trigger the sampling-phase electrically. So we'll have to exchange the push button for a different type of component.

---

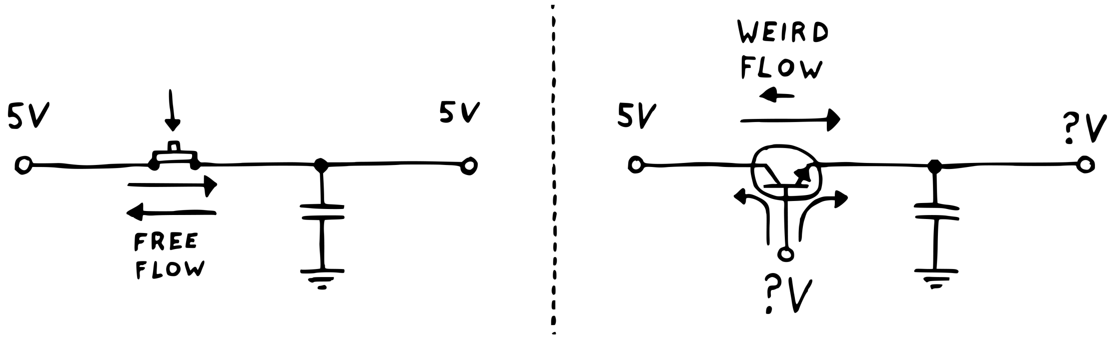
[5] Read more about capacitors in the components & concepts appendix (page 45).

[6] You can try this chapter's circuit in a simulator. I've already set it up for you right here: https://tinyurl.com/2axk2n34 – you can change all values by double clicking on components.

SWITCH VS. BJT

If you're now already thinking about transistors, then you're on the right track. Transistors, as we know, are often used as electrically controlled switches. In this series of DIY kits, we've exclusively focused on a specific type of transistor so far – the BJT or bipolar junction transistor.[7]

Unfortunately, that type of transistor isn't ideal for our purposes here – for a whole bunch of intertwined reasons. I don't want to go into too much detail, so we'll heavily simplify the issue. Let's assume we directly replace our momentary switch with a BJT.



As we know, the transistor will allow for current to flow between its collector and emitter if we apply a voltage to the base. You might then think that applying a voltage to the base is the same as pushing the button on our momentary switch. Only trouble with that is that it's totally not true. **A closed switch behaves exactly like a piece of conductive wire. An active BJT does not.**

The closed switch will allow for current to freely flow in both directions, depending only on the voltage applied across it. The BJT, on the other hand, is much more complicated in this regard. The amount of current allowed to flow between collector and emitter is fixed – and depends on a multitude of factors: the amount of current flowing into the base, the overall direction of flow, the transistor's specific gain level and so on.

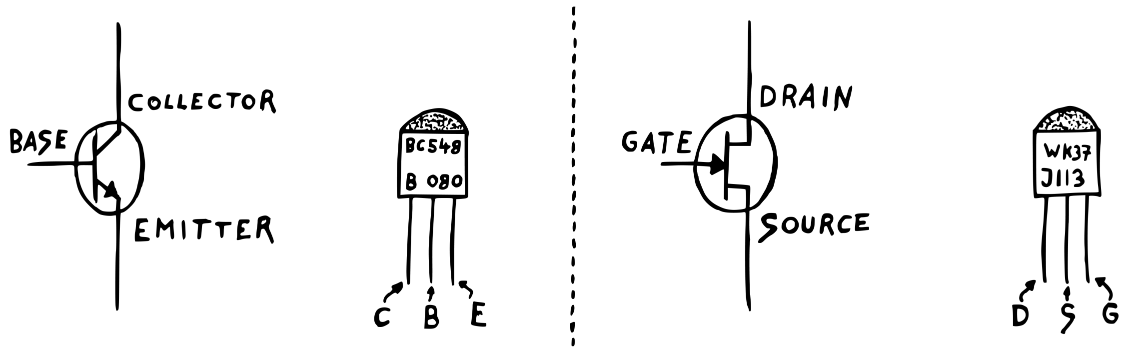
Worse yet: if there's not enough current flowing on the main path, the transistor will allow for large currents to flow into the base and fill the gap. **So where pushing the momentary switch reliably and instantaneously charges (or discharges) our capacitor to the input voltage level, the BJT will behave much more inconsistently.** As a result, the held output voltage will rarely be the same as the supposedly sampled input voltage. That's not to say that it doesn't work at all – it would just be a fairly sloppy implementation.

[7] Read more about BJTs in the components & concepts appendix (page 53).

JFET BASICS

Thankfully, there's a different type of transistor that is better suited to our use-case: the junction field effect transistor or JFET for short. There's two variants of these, called the N-channel and the P-channel JFET – which can be either symmetrical or asymmetrical. To keep things manageable, we'll focus on just one of these: the symmetrical N-channel version. So whenever I say JFET from now on, I'll be talking about this type.

The main difference between such a JFET and our trusty BJT is that the JFET, while active, behaves somewhat like a weak resistor.[8] Meaning that it allows for current to flow through it depending on the voltage applied across it – no matter the direction. There are a few caveats here though, which we'll talk about in a minute. Before that, let's first clear up the basics.

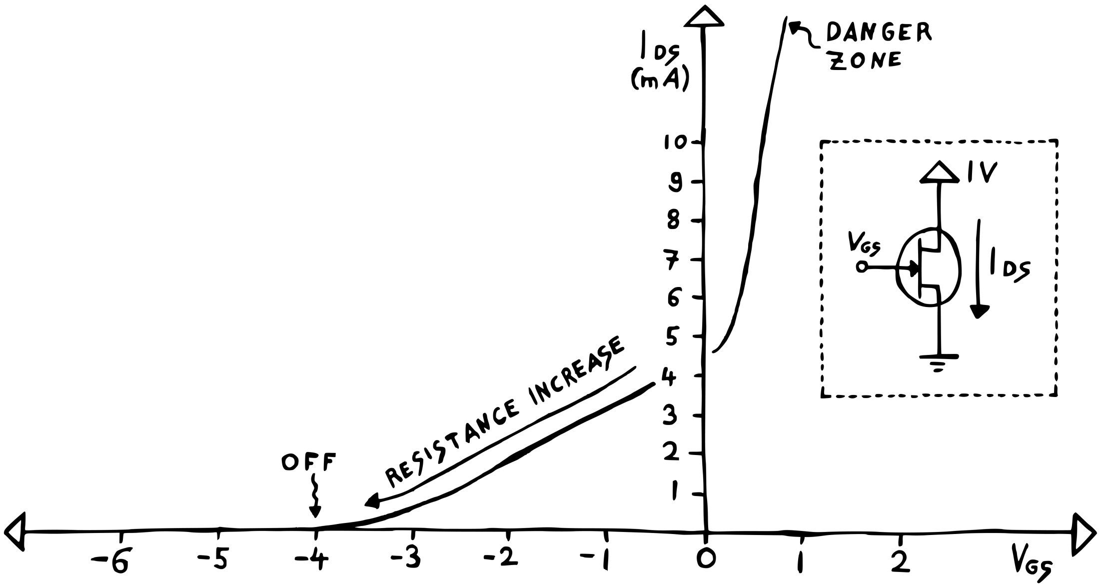


Like the BJT, a JFET (on the right) has three terminals. The naming convention is different, though. Instead of collector, base, and emitter, a JFET has a drain, a gate, and a source terminal. **The flow of current between drain and source is controlled by a voltage applied to the gate.** Since voltages are always just relative, though, we'll have to ask what the point of reference for that gate voltage actually is.

The short answer to this is simple: the point of reference is the source terminal. So when we're talking about the gate voltage, we're talking about the voltage between gate and source. The long answer is a tiny bit trickier, though. As I said earlier, the direction of flow between drain and source can be reversed freely. And if we think of the source as the terminal that current is flowing out of, then source and drain basically switch positions once we do reverse the current flow.

So effectively, the gate voltage is always measured in relation to the terminal that current is flowing out of. And that will always be the one where the applied voltage is lower.

[8] Read more about resistors in the components & concepts appendix (page 44).

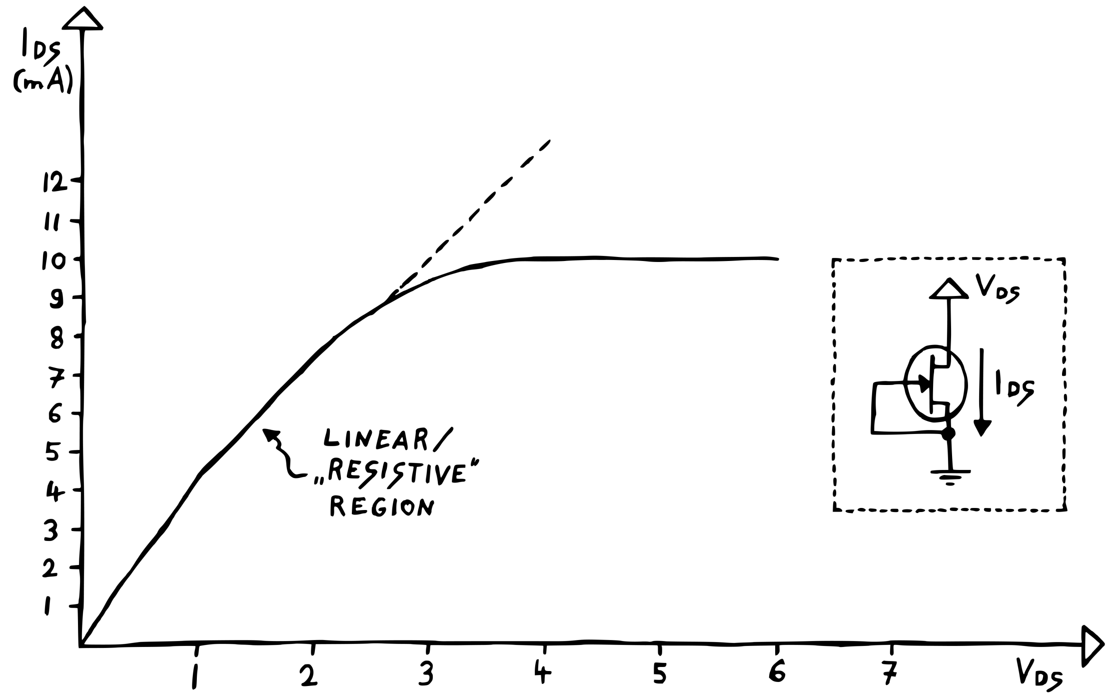


With that in mind, here's how the gate voltage relates to the drain-to-source current. For this, we're assuming that the drain and source voltages are fixed at 1 V and 0 V respectively. Now generally, the JFET is considered to be fully active when there's no voltage between gate and source – meaning that they'd both have to be sitting at ground level in this example. **Then, the transistor will act basically like a very weak resistor – and so ohm's law applies, which tells us that the amount of current flowing is determined by the voltage between drain and source fighting the JFET's resistance**.

If we push the gate voltage up from there, we can increase the drain-to-source current further. This happens because we are further decreasing the JFET's resistance. But beware – if you push it up too much, a large current will begin flowing into the gate, potentially destroying the component.

Now, if we pull the gate voltage below the source, we can reduce the amount of drain-to-source current. This happens because we're increasing the JFET's resistance. And as we know, a steady voltage faced with an increasing resistance results in a decreasing current. Once we exceed a voltage difference of more than 4 V, the JFET turns off completely and prevents any current from flowing. **So in short: via the gate-to-source voltage, we can vary the resistance between drain and source from very weak to almost infinite**.

Now, you might have noticed that the drain-to-source voltage in our example is suspiciously low. I did that because the resistor-comparison only really holds up for relatively small drain-to-source voltages. To show you what happens with bigger voltages, we'll look at a different graph.



Here, we can see the relation between drain-to-source voltage and drain-to-source-current, if the gate-to-source voltage stays at 0 V. So: source and gate are fixed at 0 V, while we push up the drain voltage. **Interestingly – and annoyingly – the JFET stops behaving like a resistor after we increase that drain voltage beyond approximately 3 V**. Because with a resistor, the current would simply keep increasing in a straight line. With our JFET, we run into a hard upper threshold.[9]

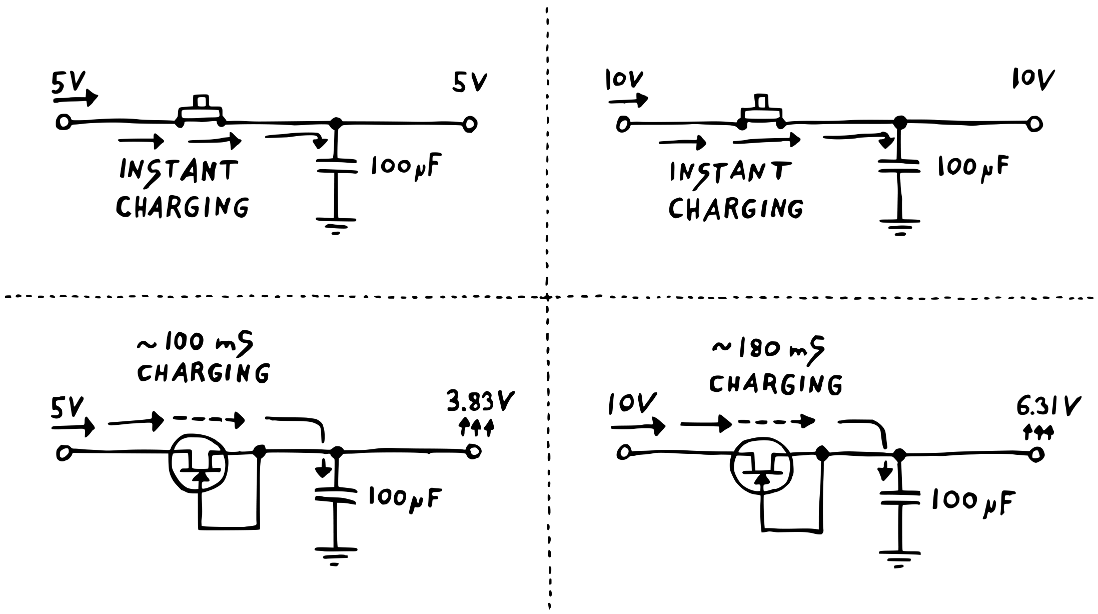
Note that this threshold – along with the general relation between gate voltage and drain current – can vary quite heavily depending on the JFET model you're looking at. The values I'm using here are just a kind of rough average.

[9] You can try this chapter's circuits in a simulator. I've already set them up for you right here: https://tinyurl.com/2aakdfus – you can change all values by double clicking on components.

SWITCH VS. JFET

Now, for our use-case, the fact that the drain current is capped is not ideal, but it's also not the end of the world. **All it means is that the bigger the drain-to-source voltage is, the slower our capacitor will be charged up**.

Think of it like this. To get the voltage above the capacitor up to 5 V, a specific amount of current has to flow into it. With our switch, this happens almost instantaneously, since there's barely any resistance, and certainly no speed limit. **With our JFET, it will take a short while, though – precisely because it has a speed limit**.

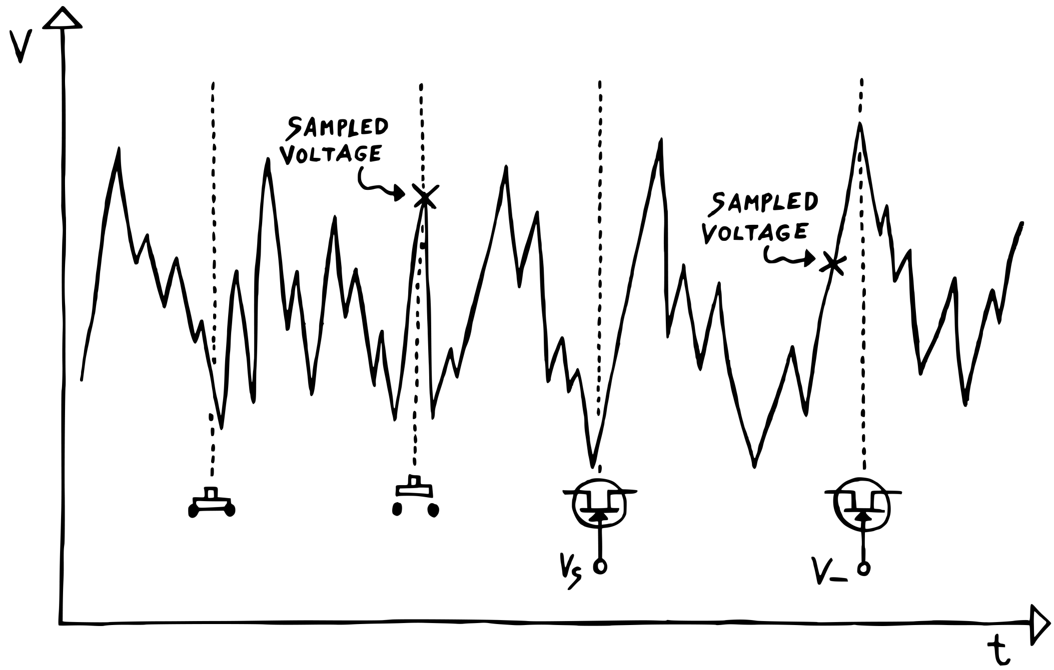


Now if we double the input voltage to 10 V, twice the amount of current has to flow into the capacitor. With our switch, again, that will happen almost instantaneously. **With our JFET though, it will take nearly twice as long – since the charging speed stays mostly constant.**

Why is this a problem? Well, it has to do with the accuracy of the sampling process. Imagine we're trying to sample a white noise signal. In such a signal, the voltage is jumping around wildly and randomly. **With our momentary switch, the voltage we sample will always be accurate and true to the input.** Because while we press down on the button, the capacitor voltage will precisely follow that input voltage – as the charging process is instantaneous. And once we let go, we hold it exactly at the then-current value.

With our JFET, things work a bit differently. First, instead of pushing and releasing a button, we'll have to bring the gate up to the source voltage, and then pull it down to a really negative value – to make sure it's always at least 4 V below the source. **Second,**

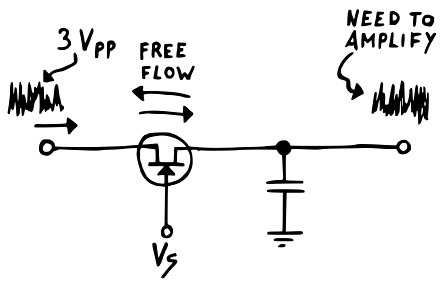
while we do that, the capacitor voltage will be lagging behind the input voltage during big jumps – because of the JFET's maximum current cap.



Here's how that would work out in detail. If we activate the JFET in the middle and then de-activate it on the right, the capacitor voltage won't be equal to the then-current input voltage. Simply because there wasn't enough time to get it charged up to that value. **Instead, we'll be getting an output that is somewhat lower.**

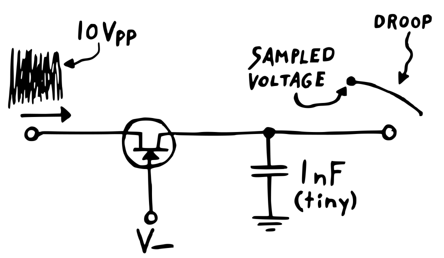
IMPROVING ACCURACY

What can we do about this? Well, there's at least two things we might consider. **First, we could reduce the volume of our input voltage.** Because as we said earlier, the JFET's speed limit only kicks in if the drain-to-source voltage is more than about 3 V.



So if we'd scale our input down to 3 V peak-to-peak, then we could charge and discharge our capacitor without that speed limit – which should increase our sampling accuracy quite a bit. **The only pain point is that we'd have to also amplify the output signal to get it back up to the original input signal's volume**.

The second option is a lot simpler – but it comes with a trade-off. We can use a very small capacitor. This helps because a smaller capacitor is charged (or discharged) more quickly. Meaning that we'll have to funnel a smaller amount of current through our JFET to get the capacitor voltage up to the input level.



This will make the speed limit less of a deciding factor here – which in turn increases our sampling accuracy. **The downside of using a very small capacitor stems from the fact that every capacitor leaks**. This means that even when there's seemingly no way for the charge to leave the cap, it will do so anyways.

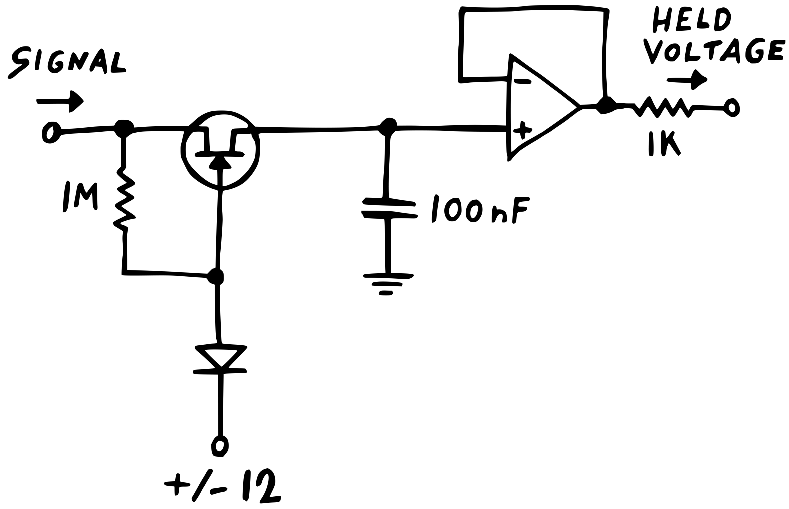
To try and avoid this, we can use capacitor types that are described as being „low-leakage", but those will still leak. We're dealing with real things here, after all, and real things are never ideal. Even our JFET will let a tiny amount of current through when it's supposedly closed. Now, why is this a problem? Simple: Because as our capacitor loses its charge, its voltage will change. **This is sometimes called „droop", and it means that our circuit's output is not going to be steady during the hold-phase**.

But as I said before, we're going to have this problem with every capacitor. So why is it worse with very small capacitors? Well, since small capacitors hold less charge, the same amount of leakage will affect their voltage level much more severely. Bigger capacitors leak the same way, but proportionally to their capacitance, they lose less charge per second and thereby hold their voltage steady for longer. As you can see, there's quite the dilemma here. **Because if we want to increase the sampling accuracy, we have to decrease the capacitor size. But if we decrease the capacitor size, we'll get more severe droop in the hold-phase**.

Now, after talking so much about accuracy, it might somewhat annoy you to hear that in a sample & hold circuit meant for use in a synthesizer, accuracy is really not all that important. At least not important enough to risk noticeable droop. Because if you're simply trying to sample some white noise to get a random sequence of voltages, for example, it really doesn't matter if the circuit is tens or even hundreds of mV off. It will still sound just as random.

This is not to say that high accuracy is never necessary. If you'd want to sample a tuned sequence for your VCO, for example, you'd absolutely need precision. Otherwise, it would sound way out of tune. **But that's a very specific use-case that I think we can neglect for our very basic implementation**.

THE S&H CORE



With all this in mind, here's how I decided to set the core circuit up. For now, the input signal is going straight into the JFET. On the other side, I've settled on a 100 nF capacitor. This is big enough to avoid obvious droop, but also small enough to have at least some degree of accuracy.

After the capacitor, I've placed a simple op amp-buffer.[10] **This is necessary because we need to isolate the capacitor as much as possible to keep the charge inside it steady**. With a buffer, we can provide a copy of the voltage level above the capacitor to be used by other modules – without pulling current from it. The 1k resistor between the buffer and the output socket protects the op amp from short circuits.

So far, so simple. But what's up with the resistor and diode at the JFET's gate?[11] Well, as we know, we need to bring the gate up to the source voltage when we want the JFET to conduct – and then pull it down when we want it to block. **The problem with that is that depending on which way the current is flowing, source and drain are switching positions**. We've discussed this before: by definition, the source terminal is the terminal that current is flowing out of.

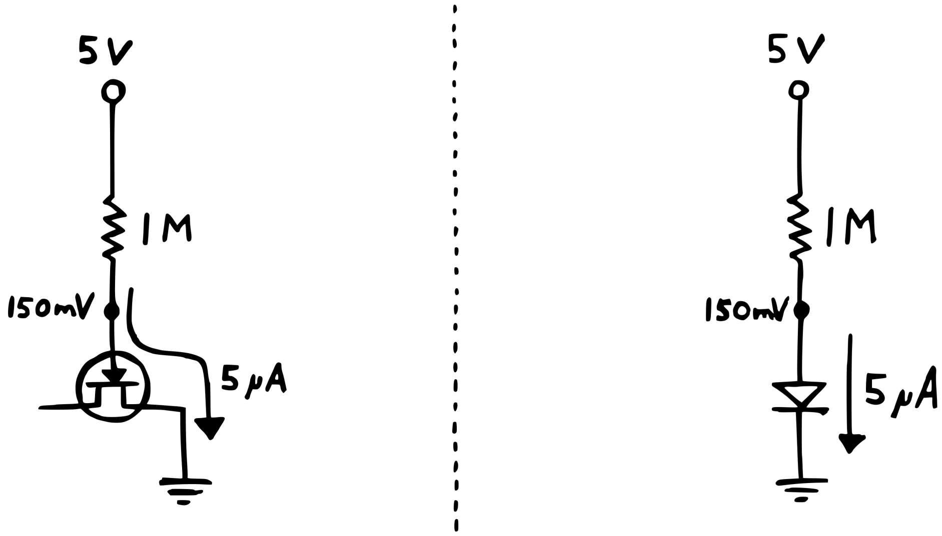
So whenever the input voltage is higher than the voltage at the capacitor, the right-hand terminal will be the source. But once the situation reverses, the left-hand terminal becomes the source. **So strictly speaking, we would need to always monitor the voltage levels at both terminals and then apply whichever is lower to the gate**. And although this is doable, there is a way simpler, although dirtier, solution: Connecting the input to the gate via a very strong resistor.

Here's how that works. Whenever the input is lower than the capacitor voltage, everything is fine anyways: gate and source voltage are identical. But if the input is higher than the

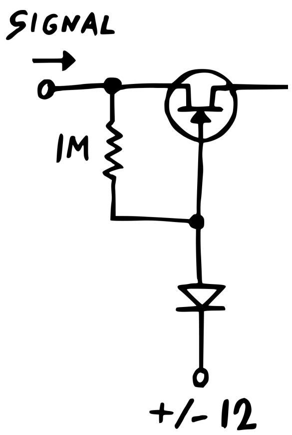
[10] Read more about op amps and buffers in the components & concepts appendix (page 50/51).

[11] Read more about diodes in the components & concepts appendix (page 46).

capacitor voltage, we'll run into trouble, right? Because now, the gate might sit significantly above the source. And in that situation, as we discussed earlier, we'll get a large current flowing into the gate and potentially destroying the transistor!
That would be true – if it weren't for the huge 1M resistor severely limiting that current. Because of this, we'll never see more than a few microamps going into the gate. Which it can handle easily. Better yet: **those few microamps flowing into the gate will make the gate voltage drop to a value just slightly above the source voltage**. Why is that? Because essentially, the JFET's gate-to-source path is a simple diode.



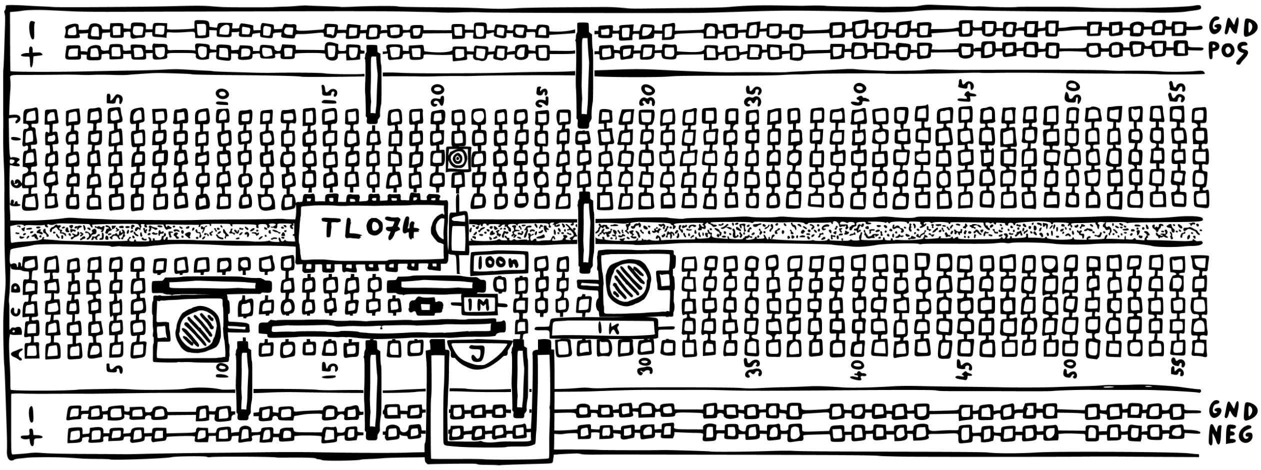
So as long as the voltage at the top is 0 V or below, that diode is blocking. But as soon as we go positive, it will start to conduct. And that conductivity ramps up exponentially as the voltage increases linearly. **So, simplified: the more we push from above, the quicker and wider the diode will open up, sinking most of the current we manage to squeeze through that ginormous resistor**.
It's as if there's a weird little trap door at the bottom that opens further the more you push against it. Because of this, no substantial voltage can build up above the one at the source. We'll get a couple hundred mV, yes, but that's it.



So with this setup, the gate voltage will always be at or close to the source voltage – as long as the diode below the gate doesn't conduct. Why is it here, then? **Simple: to activate and de-activate the JFET**.
This is how it works. Whenever we want to start the sampling process, we push the voltage at the bottom up to the highest level we have – that's 12 V in a eurorack system. Then, the diode will block – because we're pushing the trap door shut from below. **Leaving the gate voltage at (or near) the source and thereby activating the JFET**.

Once we want to enter the hold-phase, we simply pull the voltage to the lowest level we have – that's -12 V in a eurorack system. This way, we sink all of the current coming through the resistor and drag the gate voltage close to those -12 V. Shutting the JFET off in the process.[12]

Cool! So let's finally try this on the breadboard. For the op-amp, we'll be using a TL074 IC, which is four op-amps in a single chip. **Make sure that you set it up exactly as shown here – if you reverse the power connections, it will heat up and die!**



To see if everything works as expected, connect a slow oscillation (e.g. from a looped envelope with long attack & release times) as your to-be-sampled signal. Next, hook up a jumper to the diode at the marked spot. **This way, you can easily trigger the sample- and hold-phases by plugging the other end into the positive or negative rail.**

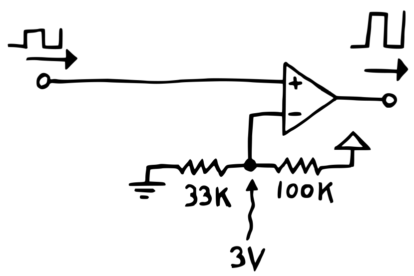
Next, connect the circuit's output to a multimeter and set it to measure DC voltage. Plug the jumper into the positive rail. This will cause the circuit to enter the sampling phase – meaning that the voltage at the capacitor starts following the input voltage. Then, plug the jumper into the negative rail. This will cause the JFET to shut off and the circuit to enter the hold-phase. Meaning that the output stays fixed to whatever voltage was present at the input when you made the switch. **Check your multimeter – you should get a different voltage every time you switch phases!**

[12] You can try this chapter's circuits in a simulator. I've already set them up for you right here: https://tinyurl.com/25bx7n3d – you can change all values by double clicking on components.

TRIGGER PROCESSING

Of course operating the circuit manually is not really what we want. So let's try and automate this! You might be tempted to just use a regular square wave LFO here that you connect to the control diode. **The first problem with this is that the signal will probably not be beefy enough.** Because remember: for the sampling phase, we need to push the diode shut with enough force. And for the hold phase, we need to pull the voltage below it into the negatives as far as possible.

So what we'll have to do is is take a square wave signal and blow it up as much as we can. For that, we'll use a simple op amp-based comparator.

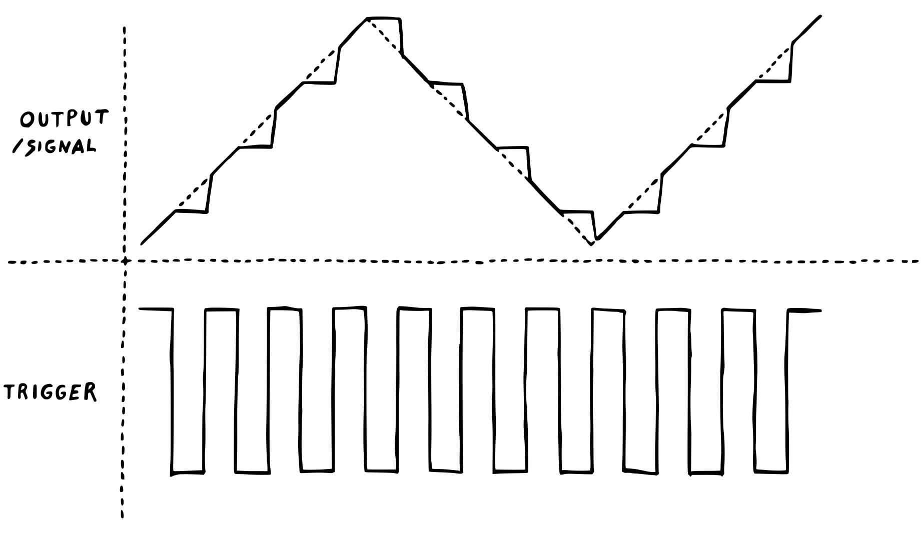


A comparator does just what its name implies: it compares an input voltage to a threshold voltage. Whenever the input is above the threshold, it'll raise its output to the positive supply voltage – and vice versa.

Okay, but what threshold voltage should we choose here? Why not simply go with 0 V – ground? **Easy: because we don't want our comparator to fire randomly if we leave the input unconnected – which can happen if electromagnetic interferences push the input voltage slightly above the 0 V-line.**

By using a voltage divider to set up a 3 V threshold, we can make our circuit interference-proof – with the only downside being that our input square wave has to be at least around 6 V peak-to-peak to trigger it.[13]

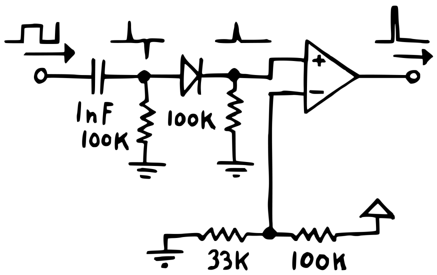
Now, if you'd set this up and tried it, you'd run into a second problem right away. Instead of nice, distinct steps, you'd get these strange, partly angled monstrosities.



[13] Read more about voltage dividers in the components & concepts appendix (page 47).

What’s up with that? Well, the problem is simple. **Since we’re using a square wave oscillation as our triggering signal, the sampling-phase is exactly as long as the hold-phase.** And as we know, during the sampling phase, the output voltage will follow the input voltage.

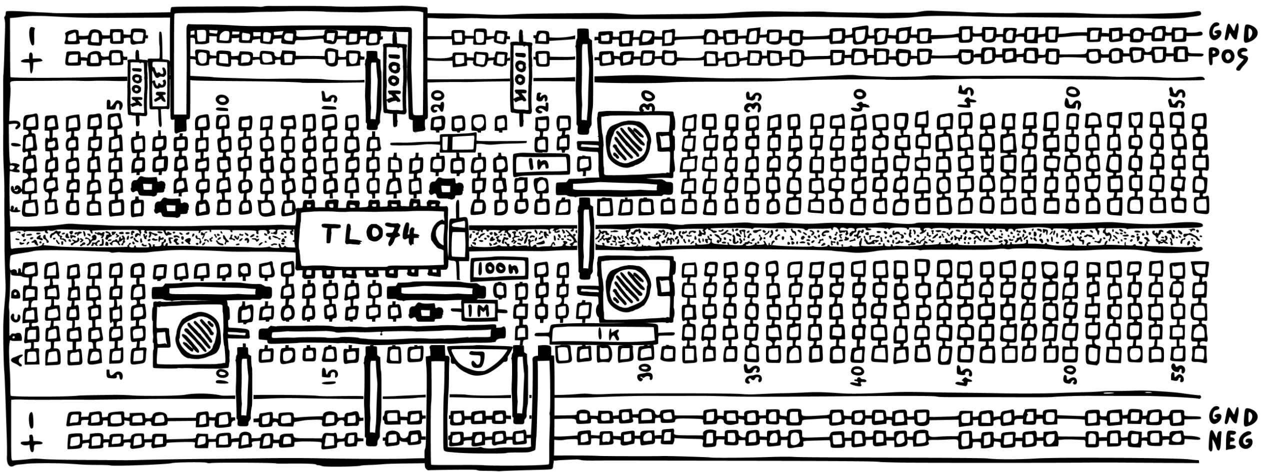
That’s why we get these angled parts in our output – it’s just the input signal pushing through during the sampling-phases. What can we do about this? **You guessed it: severely shorten the sampling phases.** Because the shorter they are, the less of our input signal will show up in the output.



To do that, we just need a simple high pass filter, which is really only a capacitor followed by a resistor to ground. This will convert our square wave into two voltage spikes – one positive, and one negative. To make those spikes really short and snappy, I’m using a super small 1 nF capacitor and a big 100k resistor.

Next, we need to get rid of the negative spike – simply because our op amp will glitch out if you feed it a voltage that is too close to the negative rail. For this, we’ll use another diode followed by a 100k resistor to ground.

This way, only the positive spike can pass through – and we’re holding the voltage at ground level while the diode is blocking. Our comparator should then spit out a super quick pulse every time our square wave input goes high.[14] Great – so let’s give this a try!

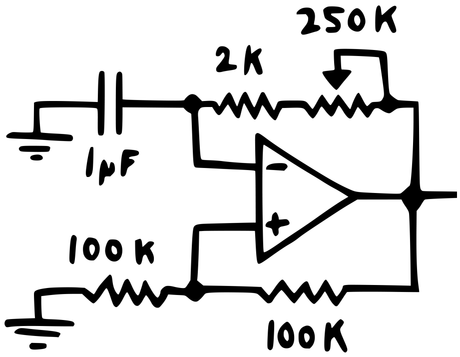


If you now send a square wave oscillation into the trigger input, while connecting the circuit’s output to a VCO’s CV input, you should hear its pitch go up and down in obvious, distinct steps. Nice – but what if you don’t have a square wave LFO at hand? No worries! Since our module should have an internal clock anyways, we’ll take slight detour and set one up right now.

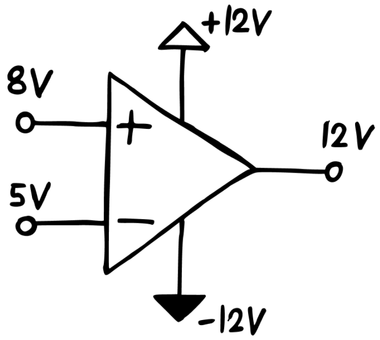
[14] You can try this chapter’s circuits in a simulator. I’ve already set them up for you right here: https://tinyurl.com/26feh3qd – you can change all values by double clicking on components.

THE CLOCK GENERATOR

To do that, we'll use just an op amp, a few resistors, a potentiometer and a capacitor. If we set these components up like this, we can pick up a square wave oscillation at the op amp's output.



Confused about how this works? If so, let's analyze this circuit step by step. For that, we'll first have to take a closer look at how an op amp works. The basic concept is this: every op amp has two inputs and one output. Think of those inputs like voltage sensors. You can attach them to any point in your circuit and they will detect the voltage there without interfering. **No current flows into the op amp's inputs – that's why we say their input impedance is very high**. Near infinite, actually. Okay, but why are there two of them?

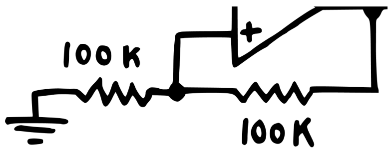


The key here is that op amps are essentially differential amplifiers. **This means that they only amplify the difference between their two inputs – not each of them individually**. If that sounds confusing, let's check out a quick example. So we'll imagine that one sensor – called the non-inverting input – is reading 8 V from somewhere. The other sensor – called the inverting input – reads 5 V.

Then as a first step, the op amp will subtract the inverting input's value from the non-inverting input's value. Leaving us with a result of 3. (Because 8 minus 5 is 3.) This result then gets multiplied by a very large number – called the op amp's gain. Finally, the op amp will try to push out a voltage that corresponds to that multiplication's result.

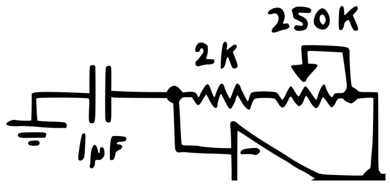
But of course, the op amp is limited here by the voltages that we supply it with. If we give it -12 V as a minimum, and +12 V as a maximum, the highest it can go will be +12 V. So in our example, even though the result of that multiplication would be huge, the op amp will simply push out 12 V here and call it a day.

So far, so simple. But in our clock generator, the relation between the two input voltages and the op amp's output voltage is quite complicated, because we feed the output voltage back to both inputs. To simplify things, we can isolate the two feedback paths for now.



Let's start with the lower one. Here, the two resistors between output, non-inverting input and ground form a 50% voltage divider. It works like this. **By taking two resistances, connecting them in series, tying one end to a voltage and the other to ground, we can pick up a fraction of that voltage where they meet**. The relation between them will determine what fraction we get.

Since both resistors are of the same value here, the op amp's output voltage will be slashed in half before it reaches the non-inverting input. So we know that the voltage at that input is always exactly half of what the output voltage is. With this in mind, let's move on to the upper feedback path.



Here, things are a bit more messy, because instead of connecting output and input with a voltage divider, we use a combination of a variable resistance and a capacitor. **That resistance is made up of a fixed 2k resistor and a 250k potentiometer set up as a variable resistor.**[15]

The added 2k resistor is then setting a minimum resistance – so that when the potentiometer is set to 0 Ω, there is still some resistance between the op amp's output and the capacitor. **This is important, because the whole idea is that we want to slowly fill and drain that capacitor by having a restricted amount of current flow from and to the op amp's output**.

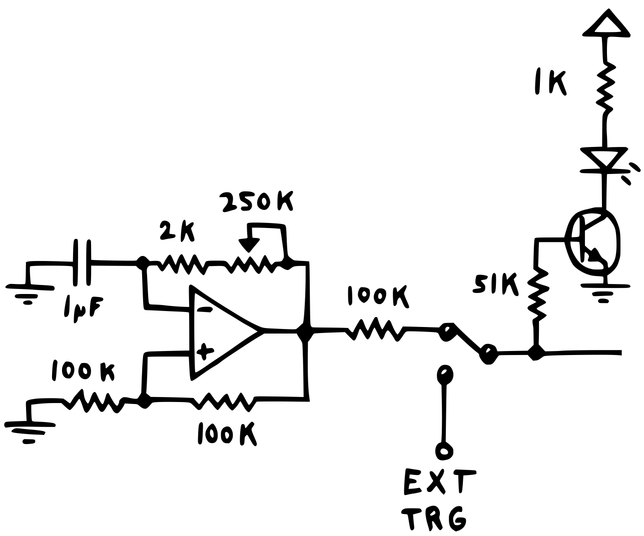
Why? Because this is what makes this circuit oscillate. Here's how. Imagine that we've just set this circuit up and connected our power supply. Then you might assume that the voltages at the inverting and non-inverting inputs and the op amp's output must be 0 V. In theory, this is true – which means that nothing should happen, really, because no current is flowing on either path.

In reality, there will be subtle voltage differences though, because we're dealing with analog components in a non-ideal world. And because the op amp's gain is so strong, we can expect its output voltage to either crash down to -12, or jump up to +12 V immediately. Let's assume it's the latter.

Then, the voltage divider on the lower feedback path will set the voltage at the non-inverting input to 6 V exactly. At the same time, we'll see a small current flow through the potentiometer and resistor on the upper feedback path and into the capacitor. As it fills

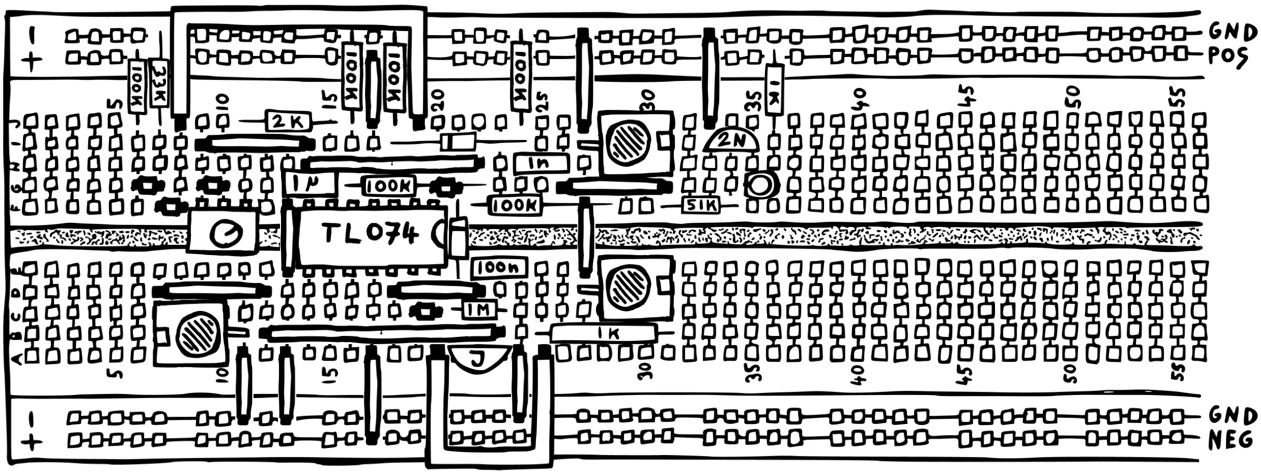
[15] You can also read more about potentiometers in the components & concepts appendix (page 48).

up, the voltage at the inverting input rises slowly. Until it crosses the 6 V-line. **Then, because the result of the op amp's input subtraction is now negative, the output will crash down to -12 V**. Pulling the voltage at the non-inverting input down to -6 V with it.
At the inverting input, the voltage will only slowly start dropping down from 6 V, as the charge inside the capacitor is pulled into the op amp's output. As soon as it crosses the – 6 V-line, everything resets: the output jumps to +12 V, pushing the non-inverting input's voltage up to 6 V, and the capacitor is slowly filling up again.
Since this process will repeat indefinitely, we get a constant square wave-oscillation at the op amp's output as a result. Better yet: we can adjust the frequency of that oscillation by turning the potentiometer's knob. **Because the lower the resistance on the top feedback path is, the quicker the capacitor is charged and discharged.** (Of course, if there's no resistance on that path, the mechanism falls apart – hence the 2k baseline resistor.)
Now, there are a few small additions we should make before sending our clock signal into the sample & hold's trigger input.



As we've just learned, the clock will swing between +12 V and -12 V. Connected straight to the high pass at the trigger input, this would result in massive ~23 V spikes. **To calm those down a bit, we'll simply put a 100k resistor between the clock generator's output and the trigger input**. This will limit the amount of current flowing into and out of the capacitor there, cutting the spikes down to just around 8 V.
Next, we'll route the clock through a switched jack socket. **This will set it as the default trigger source, while also allowing you to plug in an external signal, which would then override the internal clock**.
Finally, we add a status LED to get some visual feedback on the trigger input. Because we don't want to eat up a lot of current from the input, we drive the LED using a standard NPN transistor here. Now whenever the trigger goes high, a small current flows through

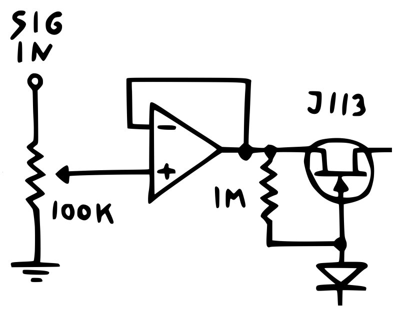
the 51k resistor and into the transistor's base, activating it. **As a result, a much bigger current will flow directly from the power supply through the LED, lighting it up.**[16]



Once you've set this up, try plying around with the clock generator's potentiometer. You should see the status LED flash slower or faster, depending on the amount of resistance you dial in. Also, the sample & hold-circuit should do its thing in sync with that status LED. If you want, you can also test plugging in an external trigger source.

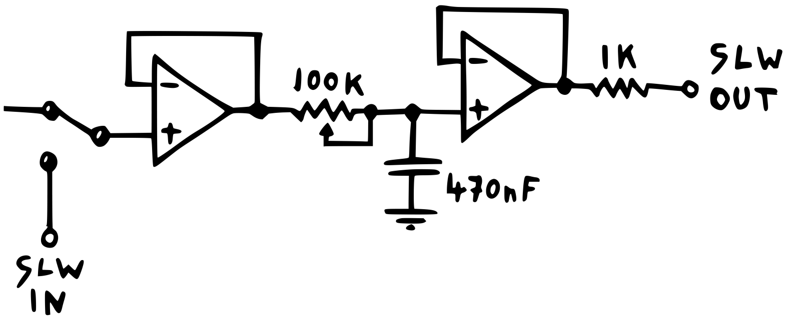
[16] You can try this chapter's circuit in a simulator. I've already set it up for you right here: https://tinyurl.com/27q4g54w – you can change all values by double clicking on components.

BUFFERED, ADJUSTABLE INPUT & SLEW OUTPUT



While our circuit is now perfectly usable as it is, we should round it out with a few minor additions. **First, we'll route our signal input through a potentiometer set up as a variable voltage divider.**[17] This way, we can change the input volume on the fly, which might be helpful if we want to reduce the voltage range at the output. **Then, we'll buffer the result with another op amp to avoid loading effects.**

Finally, we'll add in a separate output path that allows us to dial in different slew rates. A slew rate, in case you don't know, is pretty much the same as glide or portamento. **Giving our output a slew rate simply means having it glide to varying degrees between consecutive output voltages**. And we achieve that with a simple input- and output-buffered low pass filter whose cutoff frequency can be controlled with another potentiometer.[18] Now the more resistance we dial in, the longer it takes for the capacitor to be filled up and the output voltage to climb to the input voltage level.



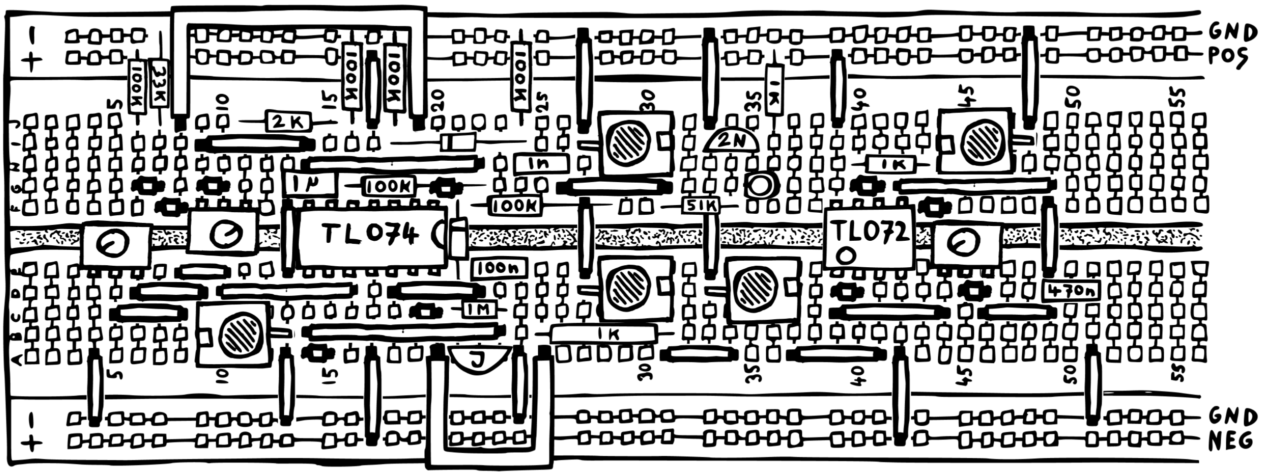
And since you might want to use the slew circuit on some other, external signal, we'll route the sample & hold's regular output through another switched jack socket before sending it into the input buffer.[19]

---

17 Read more about variable voltage dividers in the components & concepts appendix (page 48).

18 We'll look at low pass filters in more detail in a later chapter.

19 You can try the complete circuit in a circuit simulator. I've already set it up for you right here: https://tinyurl.com/288h556d – you can change all values by double clicking on components.



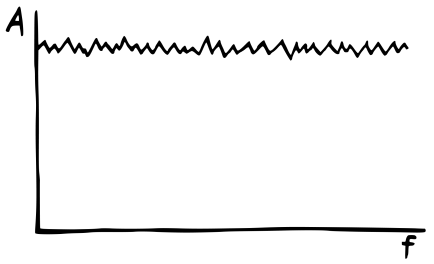
Once you've set this up, try playing with the input volume knob – you should be able to reduce the output voltage range. Next, dial in a bit of slew. The slew output should start gliding noticeably between voltages. Great!

And with that, our sample & hold-circuit is complete. **Next, let's tackle its classic companion: the noise generator**.

NOISE BASICS

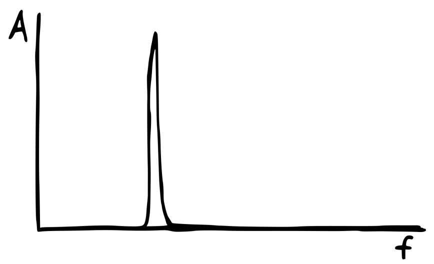
Before we can do that, though, we'll have to answer a seemingly simple question. What, exactly, is noise? **Well, from a physics standpoint, there is actually no distinction between noise and any other sound: both are by definition just vibrations in a medium.** So we have to be a bit more specific with our question.

When dealing with synthesizers, you often hear people talk about colored noise: white noise, red noise, pink noise, and so on. This is the type of noise we're interested in here. So what distinguishes colored noise from any other type of sound? To answer that, let's start off by looking at what's usually referred to as white noise.



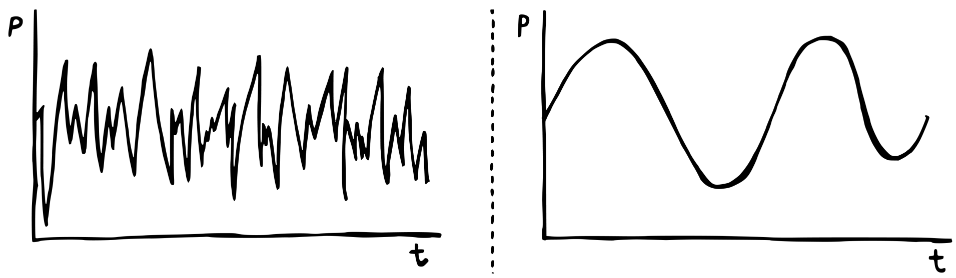
White noise is a noise that occupies the full range of audible frequencies. To illustrate this, we'll check out a simple spectrogram of a white noise signal – where the x-axis is showing frequency, while the y-axis is showing amplitude.

As you can see, every frequency is present here – in slightly fluctuating, but generally similar amplitudes. Now, let's contrast this with a spectrogram of an ideal sine wave oscillation.



Right away, there's a stark difference. **Where the white noise signal is spread out across the whole diagram, the sine wave signal occupies just a tiny patch**. That's because a sine wave is what we refer to as a pure tone. It consists of just a single frequency – unlike other waveforms, which contain a wide variety of additional frequencies, also called overtones.

Basically, sine waves and white noise couldn't be more different – you might even conceptualize them as polar opposites. This becomes even more apparent if we visualize them in a different way.

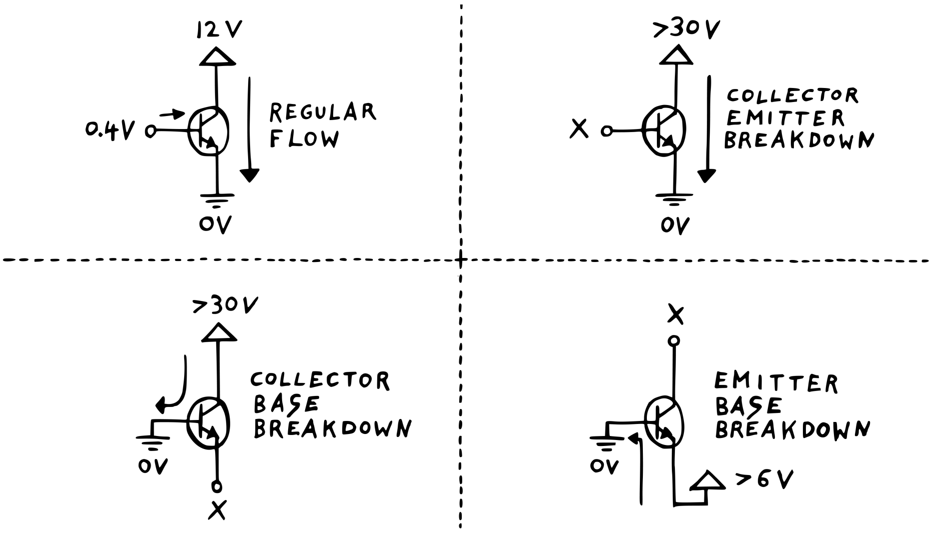


In these diagrams, we draw pressure in a medium over a given period of time. This way, we get what you'd usually refer to as the sound's waveform. For our sine wave (on the

right), this results in a smooth and repetitive shape. For our white noise signal, on the other hand, we get a chaotic, frantic and unpredictable mess.
This leads us to a more useful description of what our ears perceive as noisy: randomness. **The more random and unpredictable a sound is, the more noisy it's going to seem to us**. Now, in the world of electronics, sound is encoded as fluctuations in voltage over time. So if we want to build a circuit that generates white noise, we'll essentially have to build a circuit that generates a randomly fluctuating voltage.

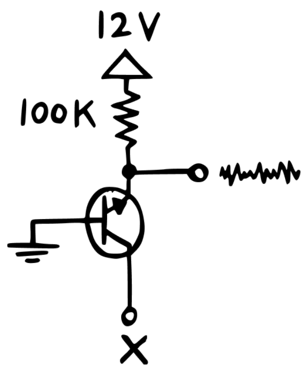
TRANSISTOR BREAKDOWN

How do we do that? Easy: by abusing a semiconductor device like an NPN transistor. **As you already know, the deal with semiconductors is that they only allow for current to flow under certain conditions**. For NPN transistors, it works like this: no current can flow into the collector as long as there is no current flowing into the base. At least in theory.
In practice, if we check the data-sheet for the BC548 transistor included in your kit, we'll find multiple mentions of something called a breakdown voltage. There's one specified each for the paths between collector and emitter, collector and base and emitter and base.



What does this mean? **Simple: it means that you can only push the transistor so far before it starts misbehaving**. If we apply more than 30 V between collector and emitter, for example, the transistor will start to conduct even when it normally shouldn't.
Same idea for the path between emitter and base: under normal conditions, current should not be able to flow into the emitter and out of the base. But if we apply more than 6 V to the emitter, while the base is sitting at ground level, the transistor will break down and allow for current to flow.
Okay, but how does this help us create a randomly fluctuating voltage? Well, as is often the case with analog electronics, things are not as clean cut as you might assume. A transistor „breaking down" doesn't mean that the floodgates simply open up and current can pass through unobstructed.
In reality, something far weirder happens: we'll see random small-ish bursts of current flow through the transistor – random in timing and random in intensity. It's as if the transistor is still fighting to not break down, but can't quite keep up. Though it might feel a bit cruel, this is exactly the behavior we are going to exploit for our noise-generating purposes.

TRANSISTOR NOISE



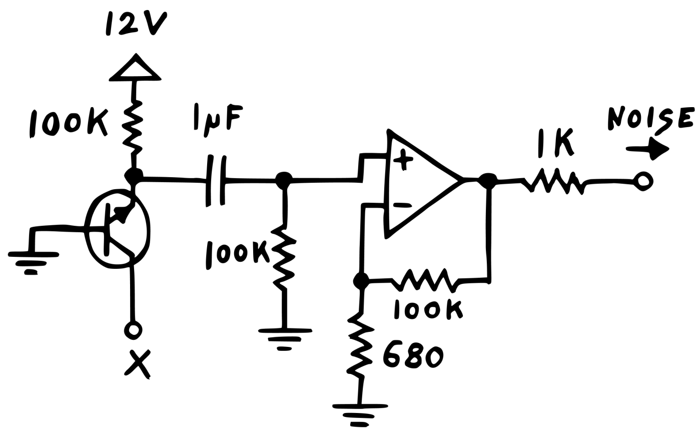
Now, to actually make noise happen, we only need two components: the transistor and a resistor. Here's how it works. We connect our transistor's emitter to the positive supply voltage through a resistance. Then, we ground the transistor's base while leaving the collector unconnected.

The resistor serves one main purpose here: we need to convert the fluctuations in current flow into a fluctuating voltage. Why does it help us with this?

To understand that, let's first assume that we've dialed our supply voltage down, below the 6 V-threshold. Now, no current flows into the transistor's emitter, because the transistor is not breaking down yet. **This means that the pushing force – which a voltage essentially is – does not actually push any current through the resistor**. So the voltage at the emitter is the same as it is above the resistor.

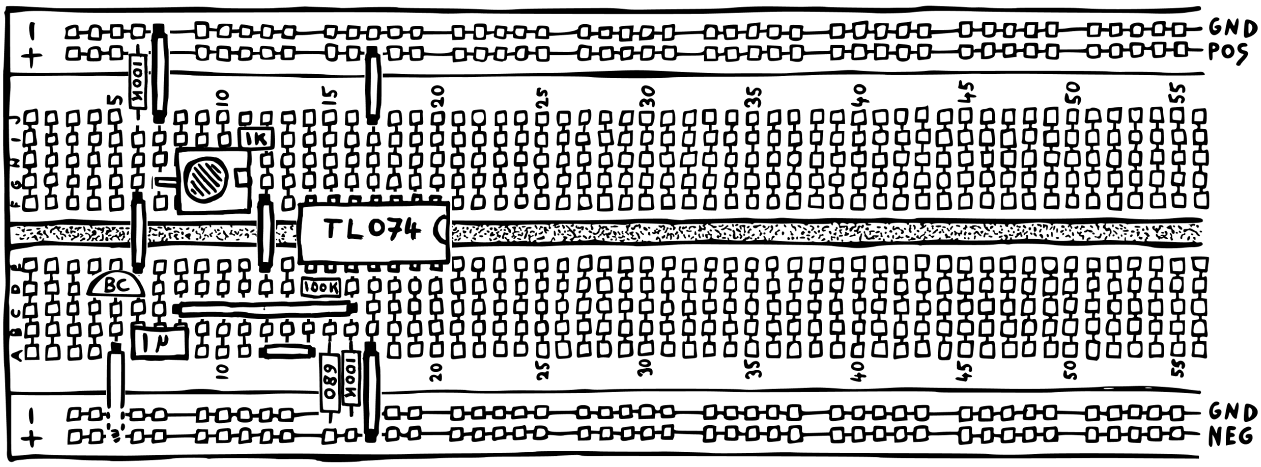
But as we increase the voltage significantly above the 6 V threshold, the transistor will begin to break down. **Resulting in random bursts of current that flow from the power supply, through the resistor, into the emitter and out of the base.** Now, the more current flows through the resistor, the more the voltage at the emitter will drop. That's because the pushing force there is allowed to transform into actual movement of electricity.[20]

This way, we convert the random fluctuations in current flow into a voltage that drops below the set supply voltage in both random amounts and random intervals. Great! Though if you'd set this up and tried it, you probably wouldn't hear much. That's because the output signal's volume is really low. **So we should probably try to amplify it if we want to listen to it.**



[20] Unfortunately, this does not work in the circuit simulator – that's why I won't include circuit simulator links for the noise generator.

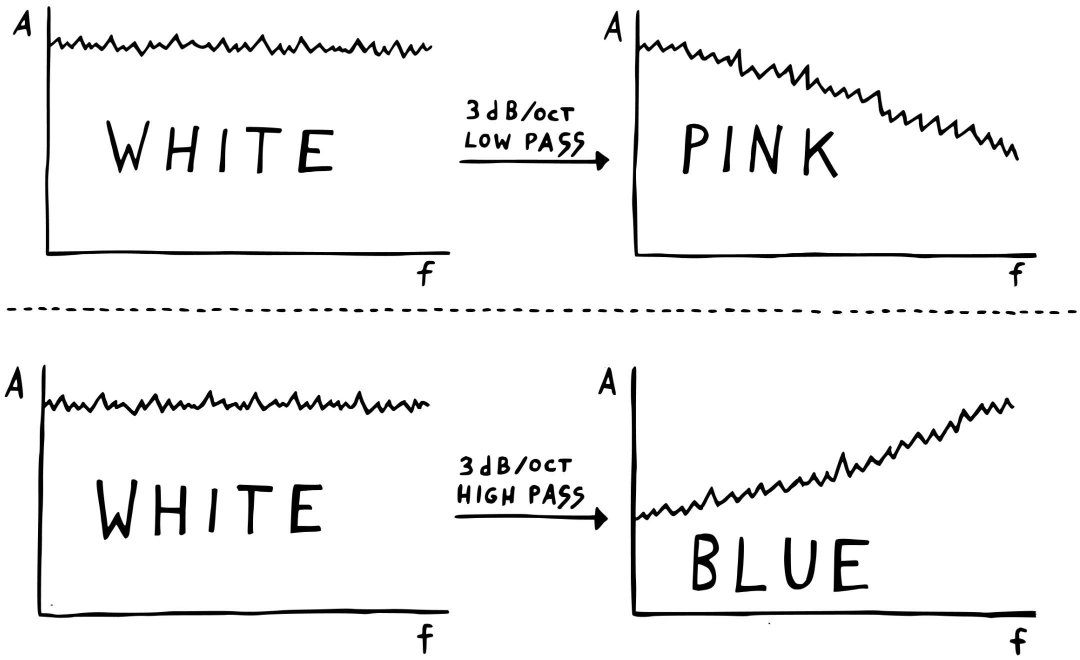
Before we can do that, we first have to center the signal. That's because an amplifier needs enough headroom to be able to work properly. We'll use a standard op amp here, which we'll supply with +/- 12 V. This allows it to operate in a voltage range that's slightly smaller than that. Since the voltage at our transistor's emitter peaks at 12 V, we're already outside the op amp's range. **And because amplifying a waveform basically means stretching it along the y-axis, it should be clear why we need more headroom here**.
So to center the signal, we'll route it through a high pass filter. Since a high pass only allows frequencies above its cutoff frequency to pass through, this will remove the constant positive offset from our signal. **Simply because a constant offset is effectively an oscillation with a frequency of 0 Hz**. Now, since we want to remove as little of our noise signal as possible, we choose a capacitor/resistor-combination that gives us a very low cutoff frequency of just 1.6 Hz.
Now we can send our centered signal into the aforementioned op amp, which we'll configure as a non-inverting amplifier. For that, we put a 100k/680 ohms voltage divider in the feedback path, which should give us a gain factor of around 150. This will blow our noise signal up to around 10 V peak-to-peak – the standard output level in eurorack systems.
In order to try this, I'd recommend you build it on a separate breadboard, so that we can eventually feed the noise signal into our sample & hold circuit. If you're using 9 V batteries, make sure you connect the transistor's base to the negative rail instead of ground – otherwise there won't be enough voltage to make the transistor break down noticeably!



If you now plug your headphones into the output socket, you should get a decently loud burst of static. Nice!

ADDITIONAL NOISE COLORS

Okay – one noise color down: white. The module advertises two additional colors, though: pink and blue. Before we can tackle those, we have to ask what differentiates them from a standard white noise signal. The answer is generally quite simple. **Pink noise is just white noise sent through a low pass filter**. **And conversely, blue noise is white noise sent through a high pass filter**.

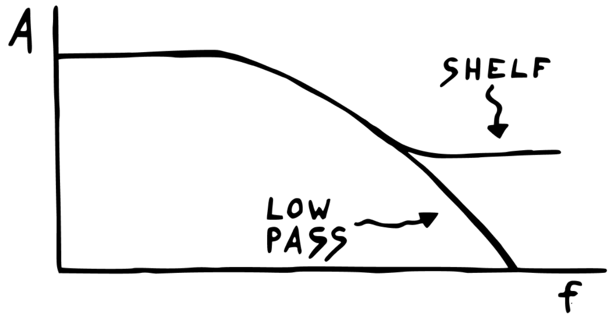


Once you look closer, the requirements get pretty tricky though. **Because both filters should have a slope of 3 dB/octave across the entire signal's spectrum**. So that the pink noise signal very gradually gets quieter towards the higher frequencies – and vice versa for the blue noise signal. Why is this tricky? Because a standard analog filter comprised of a capacitor and a resistor has a slope of 6 dB/octave. And that is actually too steep for our purposes.

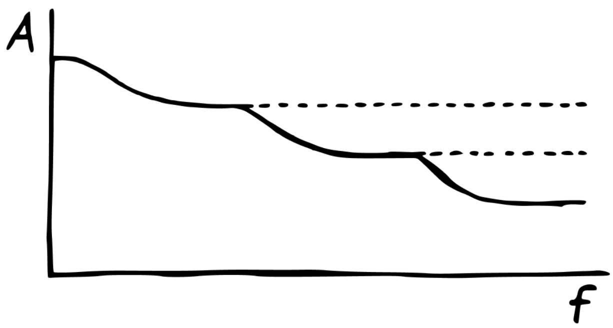
Now, in the VCF kit's manual, we learned that we can get a steeper slope by chaining multiple filter stages. But how do you get a slope that's less steep? Turns out that this is not exactly a trivial problem, because there is no inverse strategy to chaining stages. The 6 dB/octave slope is pretty much like the atom of analog filtering. **So we'll have to find a way to approximate 3 dB/octave slopes by using some dirty little tricks**. We'll start out with the white-to-pink conversion.

THE SHELF FILTER

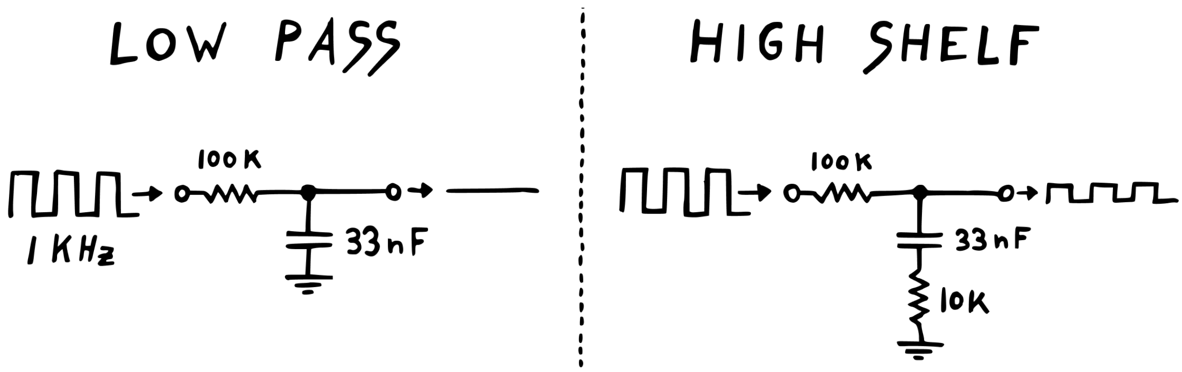
For this, we'll turn our regular low pass filter into into a shelf filter with separate slopes for the low, mid and high frequency spectrum. Now, in case you don't know what a shelf filter is or what it does, let's recap the basics here. For that, I've overlaid a standard low pass slope with a high shelf slope in this diagram.



As you can see, they are almost identical – save for how they handle the upper frequency range. Where the low pass cuts off the signal completely after a certain point, the high shelf allows a baseline amount of high frequencies to pass through.
Now, how does this help us with our 3 dB/octave slope-problem? As a single shelf filter, it doesn't really. **But what if we were able to apply multiple different shelf filter slopes to the same input signal**?



Because then, we could arrange those so that the result approximates a 3 dB/octave slope. Now, if you really care about precision here, you'd have to use a lot of shelf filter slopes to get something that resembles a smooth line as much as possible. **For me, a rough approximation is good enough – which is why I decided to go with just three slopes in total: one for the low-end, one for the mids, and one for the high-end**. Giving us this coarsely stepped pseudo 3dB/octave-slope.
But while this sounds great in theory, so far, we don't even know how to set up a shelf filter with a single slope, let alone three. So let's again cover the basics here first.

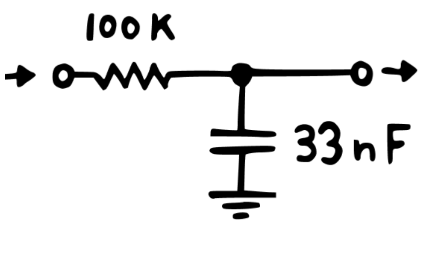


We'll start with a simple high shelf filter. **If you compare it to a basic low pass, you'll notice that they're almost identical – save for the additional resistor between the capacitor and ground.** To understand why this little change allows a baseline amount of high frequencies to pass through, we'll have to think about the mechanics of a basic low pass filter first.

So imagine that we apply a 1 kHz square wave oscillation to the low pass filter's input. We'll assume that the input resistor is very strong – say a 100k – and the capacitor is quite big – a 33 nF. Resulting in a cutoff frequency of just 48 Hz. Then at the output, we'll measure an almost unchanging 0 V signal – silence. **That's because our filter is cutting off the input signal completely, since it is way above its cutoff frequency.**

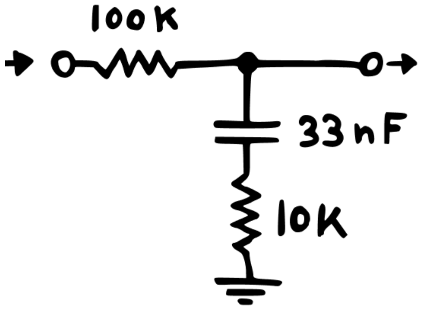
Here's how it works. While the input signal is in its high phase, it pushes a very small current through the resistor and into the capacitor. Now, because that current is so small – and the high phase is so short, due to the signal's high frequency – the capacitor will absorb it without filling up noticeably. And the voltage above it will basically stay unchanged.

Why is that? **Well, we could conceptualize the capacitor as a resistor whose resistance value depends on the amount of charge stored within it**. That's because the capacitor is displacing the same amount of current that it absorbs. It's as if that current is simply flowing through it. But the more the capacitor fills up in the process, the harder it gets to push more current through it. Hence the resistor comparison.



With this in mind, this structure essentially becomes a voltage divider. And in a voltage divider, the output voltage depends on the relation between its two resistors. **The weaker the resistance going to ground is compared to the other one, the lower the output voltage will be**. And in our scenario here, the capacitor's „resistance“ is extremely low, because it's almost empty. Which is why the voltage above it stays close to 0 V.

And it never goes beyond that, because during the input signal's low phase, the tiny amount of current we pushed into the cap gets pulled out again.



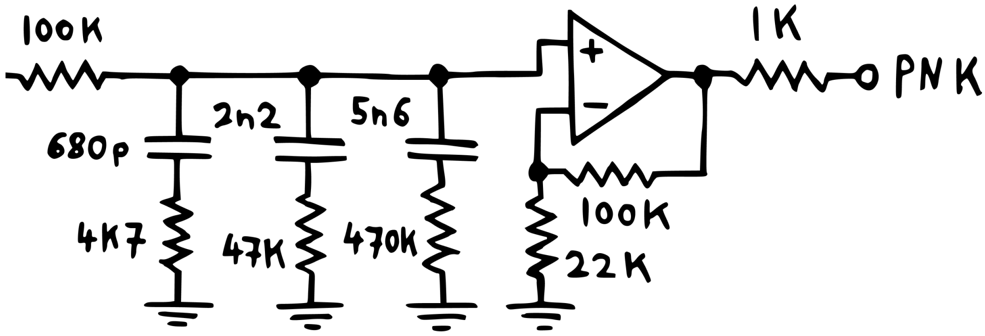
Next, let's imagine we apply the same 1 kHz square wave to the shelf filter with its additional 10k resistor between the capacitor and ground. Where before, the resistance to ground stayed close to 0, now, we've got a default value of 10k ohms. Which, by the law of voltage dividers, must result in a significant output voltage swing. Even with our super high-frequency input and an almost-empty capacitor.

That's because the resistance to ground will go above 10k as the cap fills up – but it can never fall below that value. **So as the input signal oscillates at 10 V peak-to-peak, we'd pick up a significantly quieter, but definitely audible 1 V peak-to-peak signal at the output**. And voila: we've set an amplitude shelf that the filter's output cannot fall below.[21]

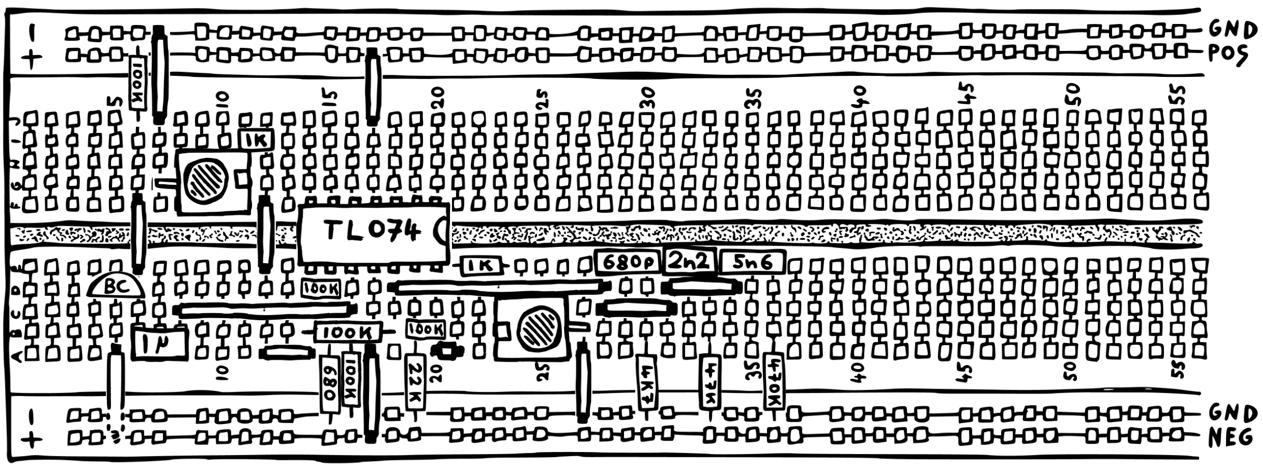
[21] You can try this chapter's circuits in a simulator. I've already set them up for you right here: https://tinyurl.com/2afpuvlq – you can change all values by double clicking on components.

THE WHITE-TO-PINK CONVERTER

Great, so we've got the single slope shelf filter down. Next, we'll need to figure out how to add more slopes to it. Thankfully, doing that is really straightforward. **All we have to do is add in more capacitor and resistor-pairs in parallel with the original one**. Now if we fine-tune their values, we should get an output the approximates a true pink noise signal.



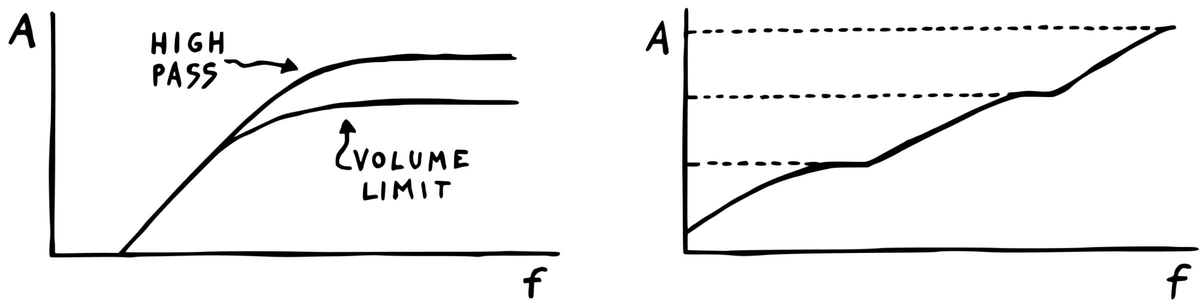
Here's how it works. As we've seen before, the amount of resistance between the capacitor and ground in relation to the input resistance will determine the amplitude at which the individual slope straightens out. **Similarly, the size of the capacitor will determine the point at which it starts dropping off – just like in a regular low pass**. Knowing this, we can select our component values so that all three slopes affect different parts of the frequency spectrum.
The values I've chosen here gave me a result closest to the pink noise examples I found online.[22] Because the filter lowers the overall volume of the noise signal, we'll have to amplify it by a factor of about 5.



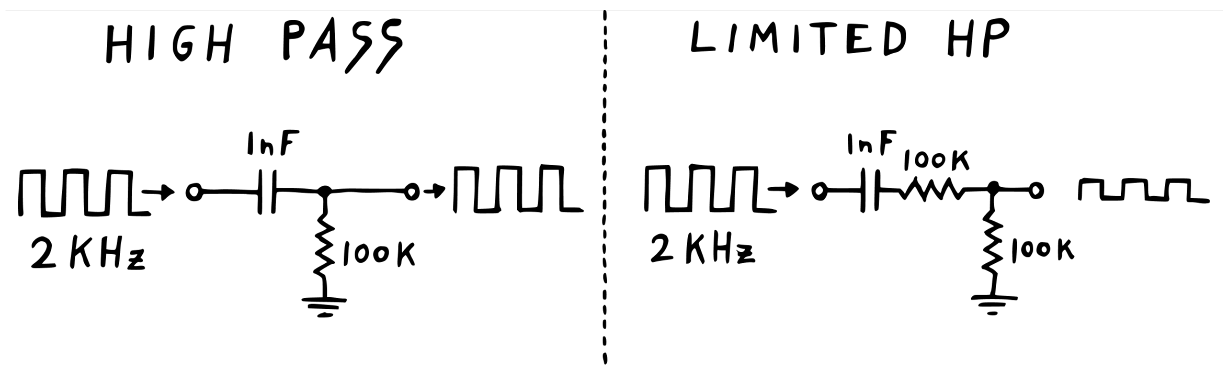
[22] Please note that when I designed this module, I've done the comparison purely by ear. When working on the corresponding video for my channel, I did a much more thorough comparison using a spectrum analyzer. If you care about your pink noise being spot-on, I suggest using the resistor- and capacitor values shown in my video.

THE LIMITED HIGH PASS FILTER

With the pink noise working, let's move on to the white-to-blue conversion. As I said before, blue noise is really just white noise sent through a 3 dB/octave high pass. Since a standard high pass filter also has a slope of 6 dB/octave, we'll have to cheat our way towards that goal with a trick very similar to the one we've just learned about.
Only this time, we're not using a shelf filter – but instead, for lack of a better term, a limited high pass. To illustrate what I mean by this, I've overlaid a standard high pass slope with a limited high pass slope in this diagram

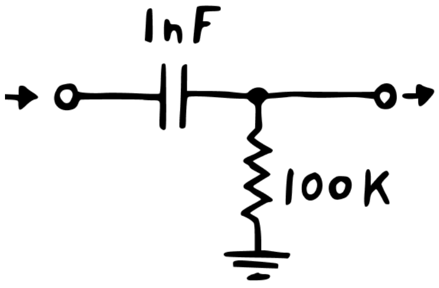


As you can see, they're almost identical – save for the unaffected parts of the signal, which differ in amplitude. Hence the „limited"-idea. Then, just like we've done with the shelf filter before, we can combine multiple of these slopes to create a coarsely stepped 3 dB/octave slope approximation. But first, we'll again have to figure out how a single slope version works. For that, we'll compare it to a basic high pass.



You'll notice that the only difference here is the additional resistor after the input capacitor. To understand the effect this has, let's recap the mechanics of a high pass filter. As the name already implies, a high pass is the exact inverse of a low pass – functionally, but also on the component level.
So where the low pass had an input resistor, the high pass has an input capacitor. And the capacitor to ground gets swapped for a resistor to ground. Easy! If you now again

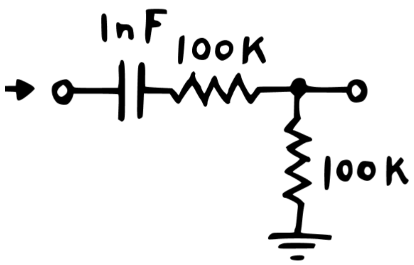
think of the capacitor as a charge-dependent resistor, it should become clear why high frequency oscillations are allowed to pass through – but low frequencies are blocked.



Let's imagine that we use a 2 kHz square wave as the input signal here. **Then the amount of current flowing through the capacitor in the wave's high- and low-phases is so small that it never builds up a significant amount of charge**.

Which is why its resistance stays very close to 0 the entire time. And in a voltage divider, if the input resistance is very weak in relation to the resistance going to ground, the output voltage will be almost identical to the input voltage.

So our high frequency square wave is allowed to pass through unchanged. **Conversely, if the input oscillation was very slow, the capacitor would get charged to the point of completely blocking**. Cutting out any low frequency content.



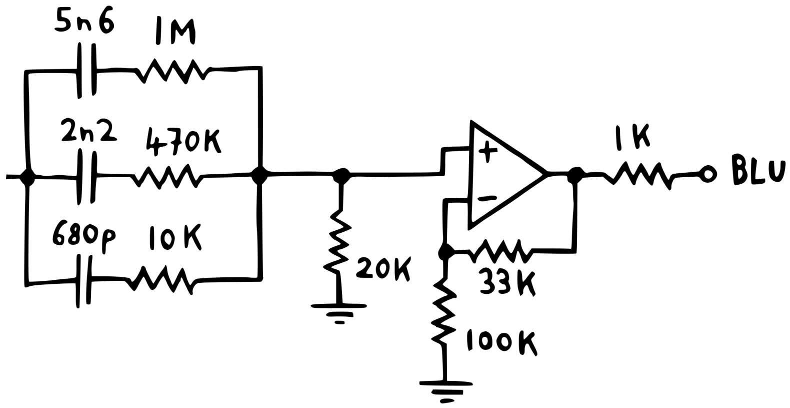
Now, by placing a resistor after the capacitor, we give that path a default resistance – just like we've done for the shelf filter. The effect is different, though. Instead of letting a baseline amount of high frequencies pass through, we set an upper volume limit for them. **That's because where before, high frequency oscillations could pass through the cap as if it wasn't even there, now, they have to fight the additional resistance.**[23]

Heads up, though: as a side-effect, doing this will lower the cutoff frequency. Simply because we also increase the total resistance between our capacitor and ground, causing it to fill up more slowly.

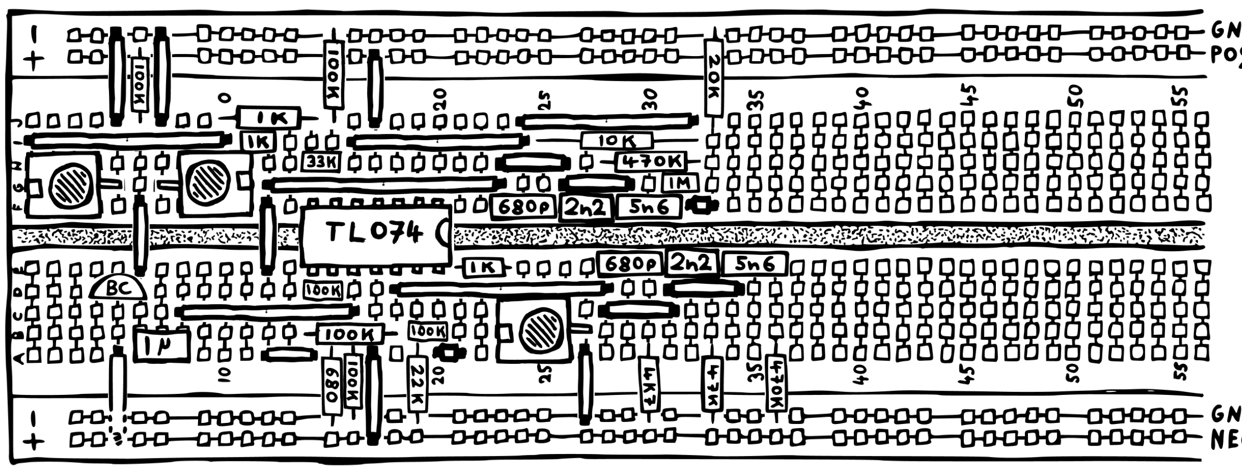
[23] You can try this chapter's circuits in a simulator. I've already set them up for you right here: https://tinyurl.com/2a2vvs2m – you can change all values by double clicking on components.

THE WHITE-TO-BLUE CONVERTER

Cool, so we've got the single slope limited high pass down. Next, let's go for multiple slopes! Just like before, we simply have to add in more capacitor/resistor-pairs in parallel with the original one and fine-tune their values until we get a decent enough result.



As we now know, the amount of resistance on the input path in relation to the resistance going to ground will determine the amplitude at which the individual slope maxes out – but also the point at which it starts dropping off. The values I've chosen here gave me a result closest to the blue noise examples I found online.[24] Because the filter lowers the overall volume of the noise signal, we'll have to amplify it by a factor of about 1.3.



And with this, our triple noisemaker is done. Next, try using one of the three noise colors as an input signal for the sample & hold circuit! Once you're done experimenting, dig out the panel and PCB from the kit, heat up your soldering iron and get to building. You can find more information on how to populate the board & how to solder in the enclosed appendix.

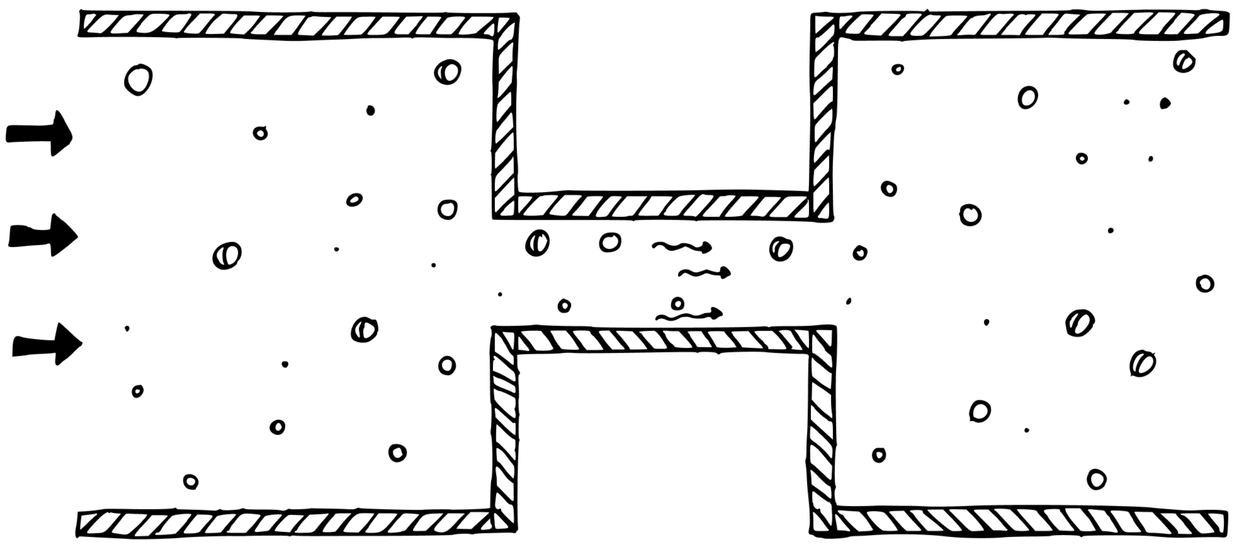
[24] Again, if you care about your blue noise being spot-on, I suggest using the resistor- and capacitor values shown in my video.

COMPONENTS & CONCEPTS APPENDIX

In this section, we'll take a closer look at the components and elemental circuit design concepts we're using to build our module. Check these whenever the main manual moves a bit too fast for you!

THE BASICS: RESISTANCE, VOLTAGE, CURRENT

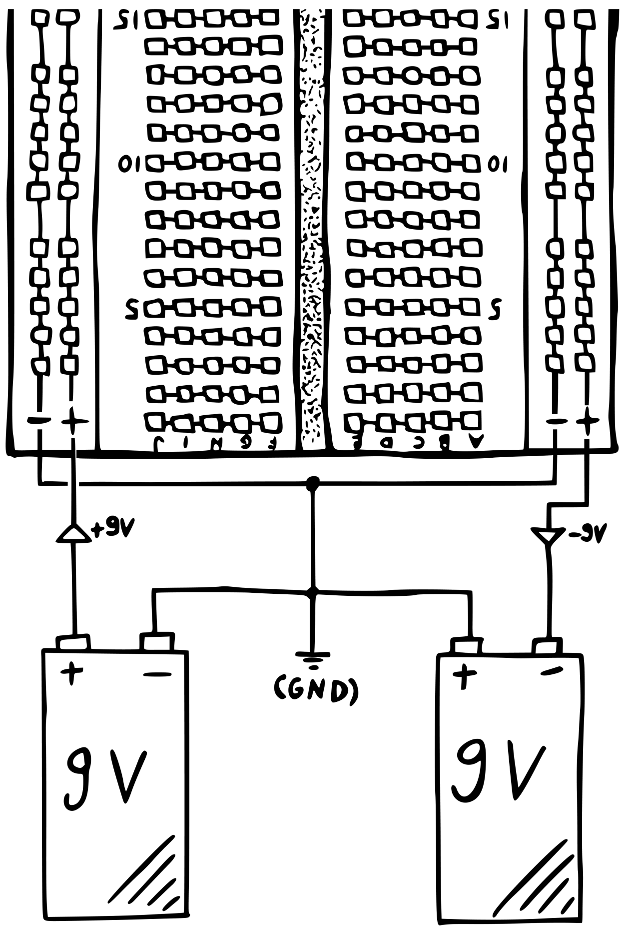
There are three main properties we're interested in when talking about electronic circuits: resistance, voltage and current. To make these less abstract, we can use a common beginner's metaphor and compare the flow of electrons to the flow of water through a pipe.



In that metaphor, resistance would be the width of a pipe. The wider it is, the more water can travel through it at once, and the easier it is to push a set amount from one end to the other. Current would then describe the flow, while voltage would describe the pressure pushing the water through the pipe. You can probably see how all three properties are interlinked: **more voltage increases the current, while more resistance to that voltage in turn decreases the current.**

USING TWO 9 V BATTERIES AS A DUAL POWER SUPPLY

Dual power supplies are great – and if you want to get serious about synth design, you should invest in one at some point. But what if you're just starting out, and you'd like to use batteries instead? Thankfully that's totally doable. **You just need to connect two 9 V batteries like shown here.** For this, you should use 9 V battery clips, which are cheap & widely available in every electronics shop.

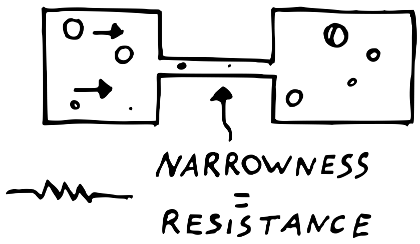


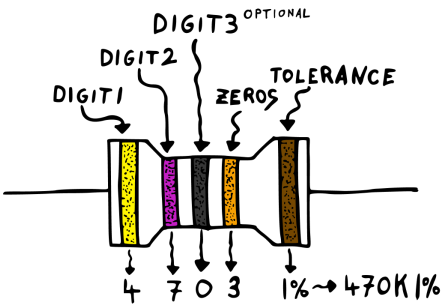
By connecting the batteries like this, the positive terminal of the left battery becomes your +9 V, while the negative terminal of the right is now your –9 V, and the other two combine to become your new ground.[25] **Please make sure you disconnect the batteries from your breadboard when you make changes to the circuit!** Otherwise you run the risk of damaging components.

[25] If you're struggling with setting this up, you can watch me do it here: https://youtu.be/XpMZoR3fgd0?t=742

RESISTORS

While a conductive wire is like a very big pipe where lots of water can pass through, **a resistor is like a narrow pipe that restricts the amount of water that can flow**. The narrowness of that pipe is equivalent to the resistance value, measured in ohms (Ω). The higher that value, the tighter the pipe.





Resistors have two distinctive properties: linearity and symmetry. Linearity, in this context, means that for a doubling in voltage, the current flowing will double as well. Symmetry means that the direction of flow doesn't matter – resistors work the same either way.

On a real-life resistor, you'll notice that its value is not printed on the outside – like it is with other components. Instead, it is indicated by colored stripes[26] – along with the resistor's tolerance rating. In addition to that, the resistor itself is also colored. Sometimes, depending on who made the resistor, this will be an additional tolerance indicator.

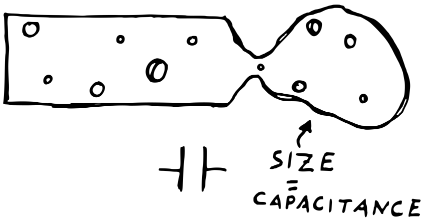
For the resistors in this kit, a yellow body tells you that the actual resistance value might be ±5 % off. A dark blue body indicates ±1 % tolerance. Some kits will also contain light blue ± 0.1% resistors to avoid the need for manual resistor matching.

While in the long run, learning all these color codes will be quite helpful, you can also simply use a multimeter to determine a resistor's value.

[26] For a detailed breakdown, look up resistor color coding. There are also calculation tools available.

CAPACITORS

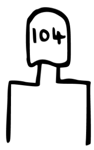
A capacitor is a bit like a balloon that you can attach to the open end of a pipe. If there's some pressure in the pipe, the balloon will fill up with water until the pressure equalizes. (Since the balloon needs some space to expand into, both of the capacitor's legs need to be connected to points in your circuit.)

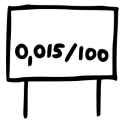


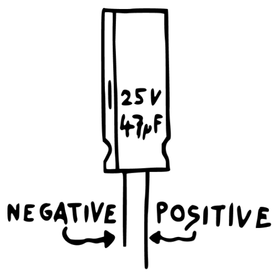
Then, should the pressure in the pipe drop, the balloon releases the water it stored into the pipe. The maximum size of the balloon is determined by the capacitor's capacitance, which we measure in farad (F). There are quite a few different types of capacitors: electrolytic, foil, ceramic, tantalum etc. They all have their unique properties and ideal usage scenarios – but the most important distinction is if they are polarized or not.

You shouldn't use polarized capacitors against their polarization (applying a negative voltage to their positive terminal and vice versa) – so they're out for most audio-related uses like AC coupling, high- & low-pass filters etc.

Unlike resistors, capacitors have their capacitance value printed onto their casing, sometimes together with a maximum operating voltage. **Be extra careful here!** That voltage rating is important. Your capacitors can actually explode if you exceed it! So they should be able to withstand the maximum voltage used in your circuit. If they're rated higher – even better, since it will increase their lifespan. No worries though: the capacitors in this kit are carefully chosen to work properly in this circuit.







Ceramic capacitors usually come in disk- or pillow-like cases, are non-polarized and typically encode their capacitance value.[27] Annoyingly, they rarely indicate their voltage rating – so you'll have to note it down when buying them.

Film capacitors come in rectangular, boxy cases, are non-polarized and sometimes, but not always, directly indicate their capacitance value and their voltage rating without any form of encoding.[28]

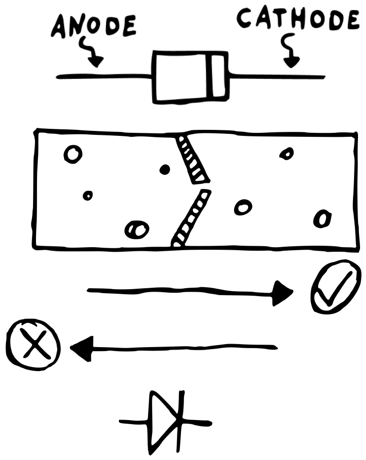
Electrolytic capacitors can be identified by their cylinder shape and silver top, and they usually directly indicate their capacitance value and their voltage rating. They are polarized – so make sure you put them into your circuit in the correct orientation.

---

27 For a detailed breakdown, look up ceramic capacitor value code. There are also calculation tools available.

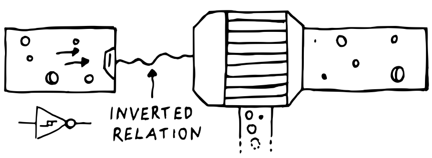
28 If yours do encode their values, same idea applies here – look up film capacitor value code.

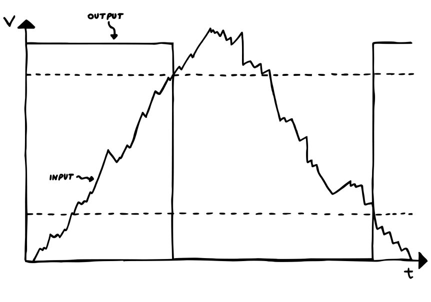
DIODES



Diodes are basically like one-way valves. Current can only pass through in one direction – from anode to cathode. That direction is indicated by the arrow in the diode symbol and by a black stripe on the diode's casing. So any current trying to move in the opposite direction is blocked from flowing.
There are a few quirks here, though. For one, the diode will only open up if the pushing force is strong enough. Generally, people say that's 0.7 V**,** but in reality, it's usually a bit lower. Also, diodes don't open up abruptly – they start conducting even at much lower voltages, although just slightly.
There are a lot of different diode types: Zener, Schottky, rectifier, small signal etc. They all have their unique properties and ideal usage scenarios – but usually, a generic 1N4148 small signal diode will get the job done.

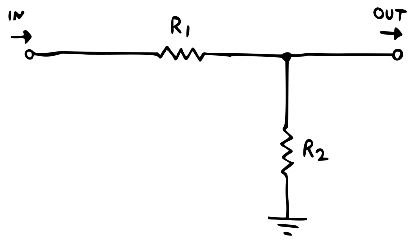
SCHMITT TRIGGER INVERTERS

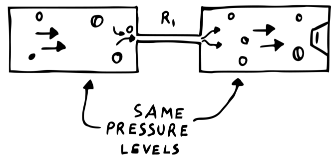


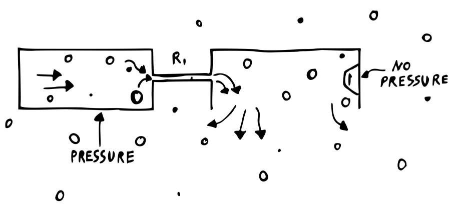


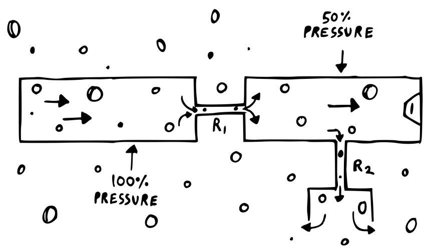
You can think of a Schmitt trigger inverter as two separate things. On the left, there's a sensor that measures the pressure inside an attached pipe. On the right, there is a water pump. This pump's operation is controlled by the sensor. Whenever the pressure probed by this sensor is below a certain threshold, the pump will be working. If the pressure is above a second threshold, the pump won't be working.
Here's a quick graph to visualize that. The squiggly line represents the voltage at the input, while the dotted line shows the voltage at the output. So every time we cross the upper threshold on our way up, and the lower one on our way down, the output changes its state. One thing that's very important to keep in mind: no current flows into the sensor! It's really just sensing the voltage without affecting it.

VOLTAGE DIVIDERS









A voltage divider is really just two resistors set up like this: input on the left, output on the right. If R1 and R2 are of the same value, the output voltage will be half of what the input voltage is. How does it work?

Let's use our analogy again: so we have a pipe on the left, where water is being pushed to the right with a specific amount of force. Attached to it is a narrow pipe, representing R1, followed by another wide pipe. Then at the bottom, there's another narrow pipe, representing R2, where water can exit the pipe system. Finally, imagine we've set up a sensor measuring the voltage in the right hand pipe.

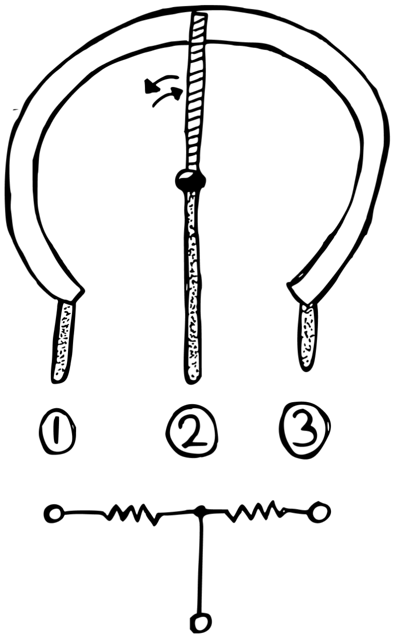
First, think about what would happen if R2 was completely sealed off. Our sensor would tell us that **the pressure on the right side is exactly the same as the pressure on the left**. Because the pushing force has nowhere else to go.

On the other hand, imagine R2 would just be a wide opening. Then **the pressure on the right would be 0**, because it'd all escape through that opening. But what happens if R2 is neither completely closed off nor wide open? Then the pressure would be retained to varying degrees, depending on the narrowness of the two resistor paths.

If pipe R1 is wide and pipe R2 is narrow, most of the pressure will be retained. But if it's the reverse, the pressure level will be only a tiny fraction. And if R1 and R2 are identical, **the pressure will be exactly half of what we send in**.

POTENTIOMETERS

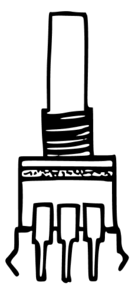
Potentiometers can be used as variable resistors that you control by turning a knob. But, and that's the handy part, they can also be set up as variable voltage dividers. To see how that works, let's imagine we open one up.

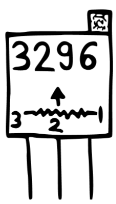


Inside, we would find two things: a round track of resistive material with connectors on both ends plus what's called a wiper. This wiper makes contact with the track and also has a connector. It can be moved to any position on the track. Now, the resistance value between the two track connectors is always going to stay exactly the same. That's why it's used to identify a potentiometer: as a 10k, 20k, 100k etc. But if you look at the resistance between either of those connectors and the wiper connector, you'll find that this is completely dependent on the wiper's position.

The logic here is really simple: **the closer the wiper is to a track connector, the lower the resistance is going to be between the two**. So if the wiper is dead in the middle, you'll have 50 % of the total resistance between each track connector and the wiper.

From here, you can move it in either direction and thereby shift the ratio between the two resistances to be whatever you want it to be. By now, you might be able to see how that relates to our voltage divider. If we send our input signal to connector 1 while grounding connector 3, we can pick up our output signal from the wiper. Then by turning the potentiometer's knob, we can adjust the voltage level from 0 to the input voltage – and anything in between.





In these kits, you will encounter different types of potentiometers. First, there's the regular, full-size variant with a long shaft on top. These are used to implement user-facing controls on the module's panel and they usually – but not always – indicate their value directly on their casing. Sometimes, they'll use a similar encoding strategy as capacitors, though.[29]

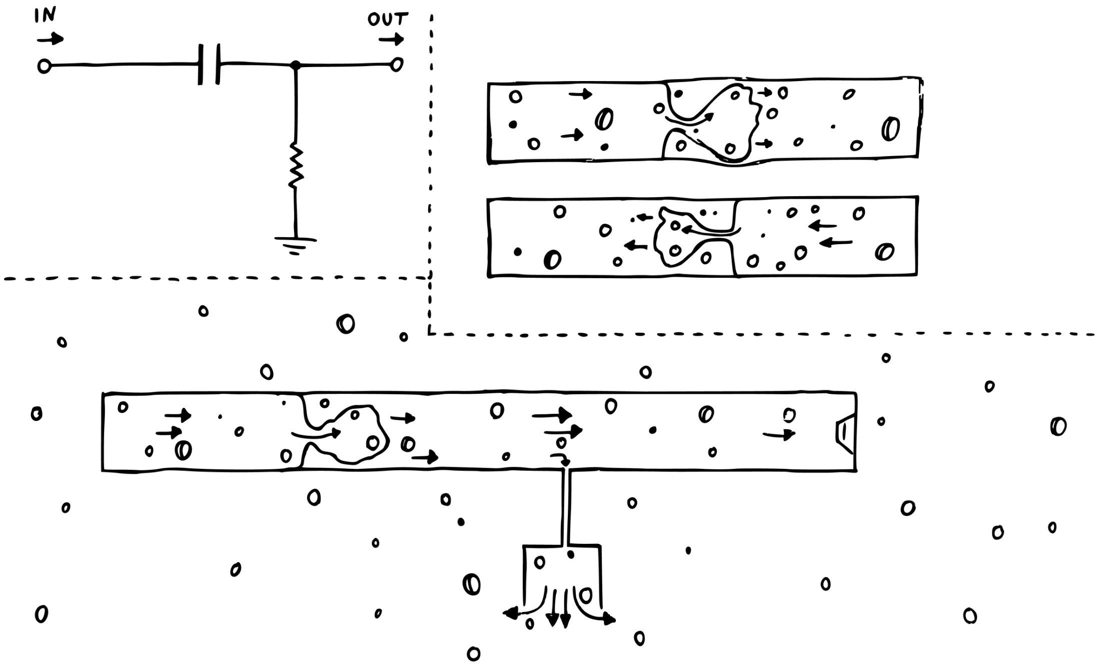
Second, we've got the trimmer potentiometer, which is usually much smaller and doesn't sport a shaft on top. Instead, these have a small screw head which is supposed to be used for one-time set-and-forget calibrations. Trimmers usually encode their value.

---

[29] Look up potentiometer value code for a detailed breakdown.

AC COUPLING

What is AC coupling – and how does it work? Imagine two adjacent pipes with a balloon between them. Now, no water can get from one pipe into the other, since it's blocked by the balloon. But, and that's the kicker, **water from one side can still push into the other by bending and stretching the balloon, causing a flow by displacement**.

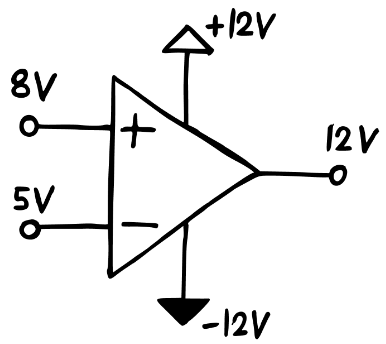


Next, we'll bring in a resistor after the coupling point, going straight to ground. **This acts like a kind of equalizing valve**. Now imagine we apply a steady 5 V from one side. Then on the other side, we'll read 0 V after a short amount of time. Why? Because we're pushing water into the balloon with a constant force, causing it to stretch into the other side, displacing some water. If we didn't have the equalizing valve there, we'd simply raise the pressure. But since we do have it, the excess water can drain out of the system. Until the pressure is neutralized, and no water is actively flowing anymore.

Okay, so now imagine that the voltage on the left hand side starts oscillating, let's say between 4 V and 6 V. When we start to go below 5 V, the balloon will begin contracting, basically pulling the water to the left. This will create a negative voltage level in the right hand pipe – like as if you're sucking on a straw, making the voltage there drop below 0 V. Then, once the pressure on the other side rises above 5 V, the balloon will inflate and stretch out again, pushing water to the right. And the pressure in the right hand pipe will go positive, making the voltage rise above 0 V. **We've re-centered our oscillation around the 0 V line.** Okay, but what about the resistor? If current can escape through it, doesn't that mess with our oscillation? Well, technically yes, but practically, we're choosing a narrow enough pipe to make the effect on quick pressure changes negligible!

OP AMPS

Op amps might seem intimidating at first, but they're actually quite easy to understand and use. The basic concept is this: every op amp has two inputs and one output. Think of those inputs like voltage sensors. You can attach them to any point in your circuit and they will detect the voltage there without interfering. **No current flows into the op amps inputs – that's why we say their input impedance is very high**. Near infinite, actually. Okay, but why are there two of them?



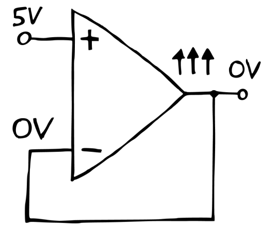
The key here is that op amps are essentially differential amplifiers. This means that they only amplify the difference between their two inputs – not each of them individually. If that sounds confusing, let's check out a quick example. So we'll imagine that one sensor – called the non-inverting input – is reading 8 V from somewhere. The other sensor – called the inverting input – reads 5 V. Then, as a first step, the op amp will subtract the inverting input's value from the non-inverting input's value. Leaving us with a result of 3. (Because 8 minus 5 is 3.) **This result then gets multiplied by a very large number – called the op amp's gain**. Finally, the op amp will try to push out a voltage that corresponds to that multiplication's result.

But of course, the op amp is limited here by the voltages that we supply it with. If we give it –12 V as a minimum and +12 V as a maximum, the highest it can go will be +12 V. So in our example, even though the result of that multiplication would be huge, the op amp will simply push out 12 V here and call it a day.

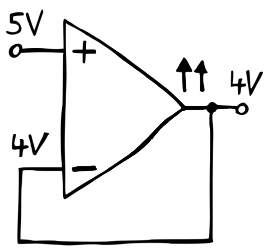
The handy thing though about op amps outputs is that they draw their power directly from the power source. This means that they can supply lots of current while keeping the voltage stable. **That's why we say an op amp has a very low output impedance**.

OP AMP BUFFERS/AMPLIFIERS

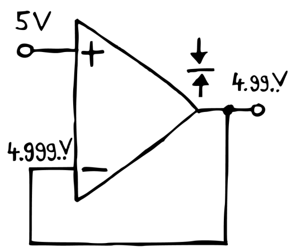
Buffering, in the world of electronics, means that we provide a perfect copy of a voltage without interfering with that voltage in the process. With an op amp-based buffer, the buffering process itself works like this. We use the non-inverting input to probe a voltage, while the inverting input connects straight to the op amp's output. **This creates what we call a negative feedback loop**. Think of it this way. We apply a specific voltage level to the non-inverting input – let's say 5 V.



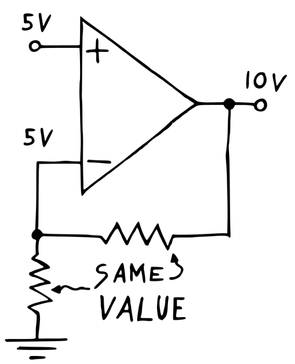
Before the op amp starts processing the voltages at its inputs, the output will be switched off. This means that **output and inverting input sit at 0 V at first**. So then, the op amp will subtract 0 from 5 and multiply the result by its gain. Finally, it will try and increase its output voltage to match the calculation's outcome.



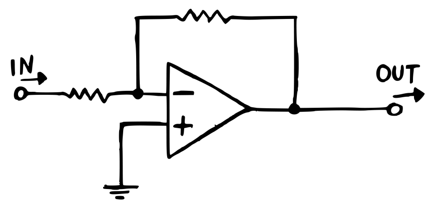
But as it's pushing up that output voltage, the **voltage at the inverting input will be raised simultaneously**. So the difference between the two inputs is shrinking down. Initially, this doesn't matter much because the gain is so large. As the voltage at the inverting input gets closer to 5 V though, the difference will shrink so much that in relation, the gain suddenly isn't so large anymore.



Then, the output will **stabilize at a voltage level that is a tiny bit below 5 V**, so that the difference between the two inputs multiplied by the huge gain gives us exactly that voltage slightly below 5 V. And this process simply loops forever, keeping everything stable through negative feedback. Now if the voltage at the non-inverting input changes, that feedback loop would ensure that the output voltage is always following. So that's why this configuration works as a buffer: the **output is simply following the input**.



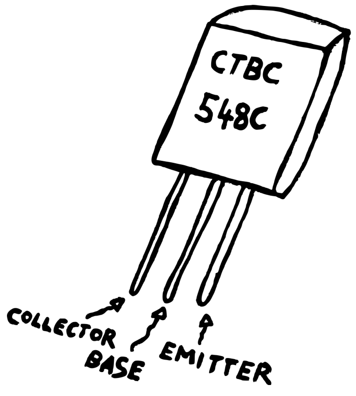
How about amplifying a signal though? To do that, we'll have to turn our buffer into a proper non-inverting amplifier. We can do that by replacing the straight connection between inverting input and output with a voltage divider, forcing the op amp to work harder. Here's how that works. Say we feed our non-inverting input a voltage of 5 V. Now, **the output needs to push out 10 V in order to get the voltage at the inverting input up to 5 V**. We call this setup a non-inverting



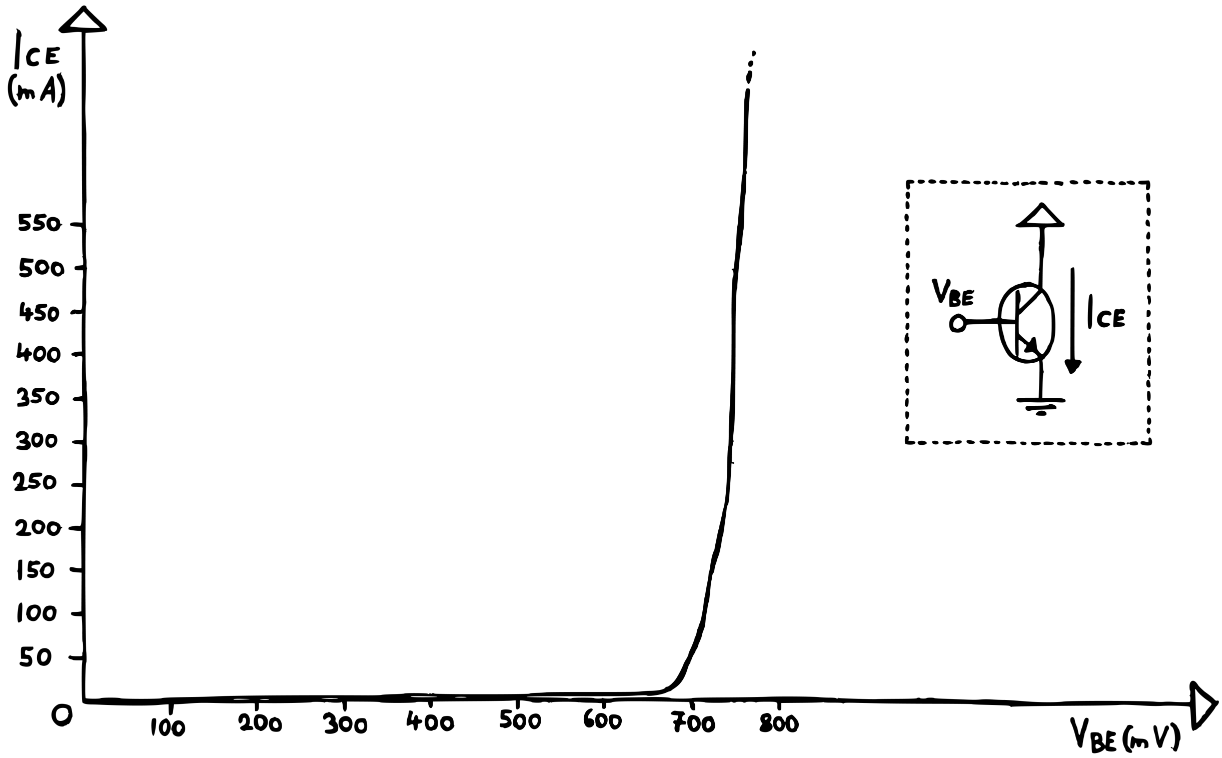
amplifier because the output signal is in phase with the input.
For an inverting buffer/amplifier, the input signal is no longer applied to the non-inverting input. Instead, that input is tied directly to ground. So it'll just sit at 0 V the entire time. The real action, then, is happening at the inverting input. Here, we first send in our waveform through a resistor. Then, the inverting input is connected to the op amp's output through another resistor of the same value.
How does this work? Well, let's assume that we're applying a steady voltage of 5 V on the left. Then, as we already know, the op amp will subtract the inverting input's voltage from the non-inverting input's voltage, leaving us with a result of –5 V. Multiply that by the huge internal gain, and the op amp will try to massively decrease the voltage at its output.
But as it's doing that, an increasingly larger current will flow through both resistors and into the output. Now, as long as the pushing voltage on the left is stronger than the pulling voltage on the right, some potential (e.g. a non-zero voltage) will remain at the inverting input. Once the output reaches about –5 V though, we'll enter a state of balance. Since both resistors are of the same value, the pushing force on the left is fighting the exact same resistance as the pulling force on the right. **So all of the current being pushed through one resistor is instantly being pulled through the other**.
And that means that the voltage at the inverting input will be lowered to about 0 V, allowing our op-amp to settle on the current output voltage level. So while we read 5 V on the left, we'll now read a stable –5 V at the op amp's output. Congrats – we've built an inverting buffer! **If we want to turn it into a proper amplifier, we'll simply have to change the relation between the two resistances**. By doing this, we can either increase (if you increase the right-hand resistor's value) or reduce (if you increase the left-hand resistor's value) the gain to our heart's content.

BIPOLAR JUNCTION TRANSISTORS

Bipolar junction transistors (or BJTs for short) come in two flavors: NPN and PNP. This refers to how the device is built internally and how it'll behave in a circuit. Apart from that, they look pretty much identical: a small black half-cylinder with three legs.



Let's take a look at the more commonly used NPN variant first. Here's how we distinguish between its three legs. **There's a collector, a base and an emitter.**[30] All three serve a specific purpose, and the basic idea is that you control the current flow between collector and emitter by applying a small voltage[31] to the base. The relation is simple: **more base voltage equals more collector current**. Drop it down to 0 V and the transistor will be completely closed off. Sounds simple – but there are four important quirks to this.

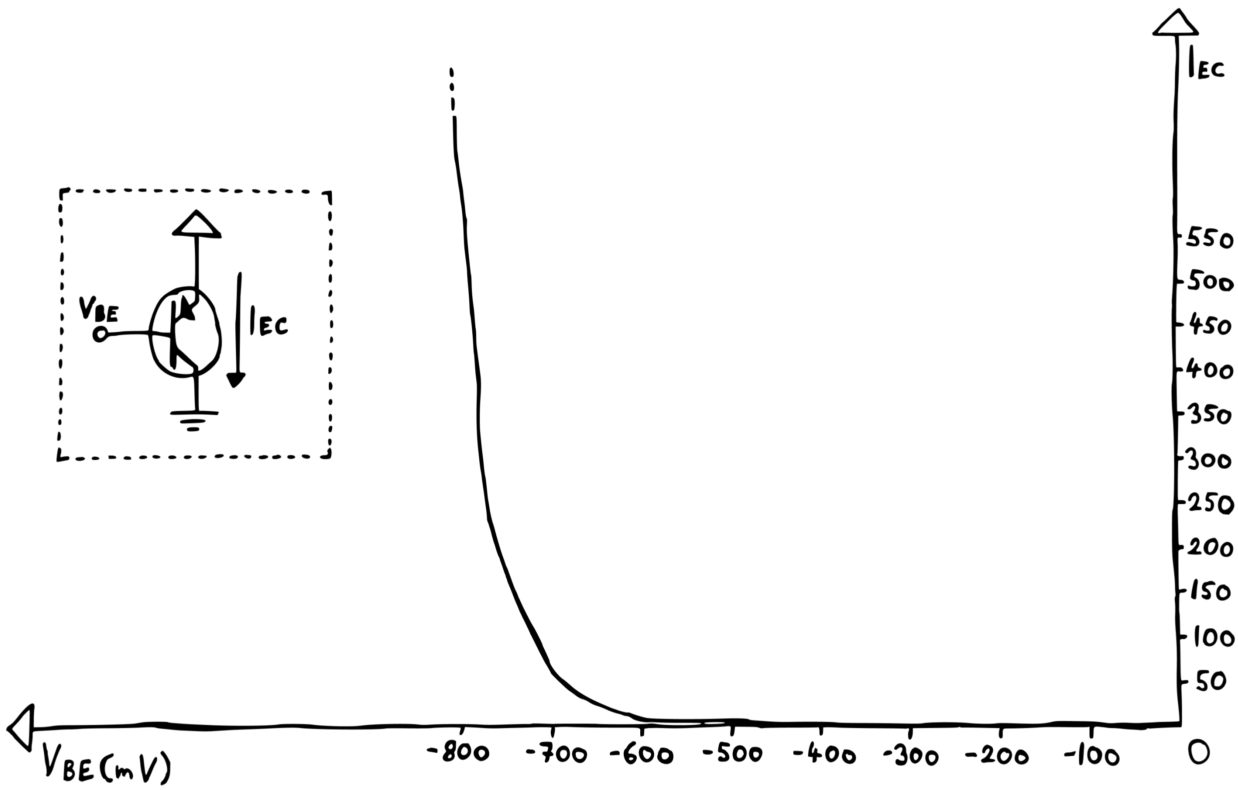


First, the relation between base voltage and collector current is exponential. Second, unlike a resistor, a BJT is not symmetrical – so we can't really reverse the direction of the

30 Please note that the pinout shown here only applies for the BC series of transistors. Others, like the 2N series, allocate their pins differently.

31 The voltage is measured between base and emitter. So „a small voltage" effectively means a small voltage **difference** between base and emitter!

collector current. (At least not without some unwanted side effects.) Third, also unlike a resistor, a BJT is not a linear device. Meaning that a change in collector voltage will not affect the collector current. And fourth, the collector current is affected by the transistor's temperature! The more it heats up, the more current will flow.
Now, for the PNP transistor, all of the above applies, too – except for two little details. Unlike with the NPN, **the PNP transistor decreases its collector current when the voltage at its base increases**[32]. So you have to bring the base voltage below the emitter to open the transistor up. Also, that collector current flows out of, not into the collector!

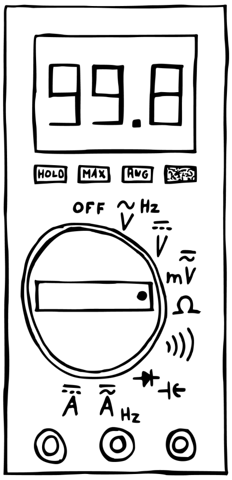


[32] Again, the voltage is measured between base and emitter.

TOOLS APPENDIX

There are two types of tools that will help you tremendously while designing a circuit: multimeters and oscilloscopes. In this appendix, we'll take a quick look at each of these and explore how to use them.

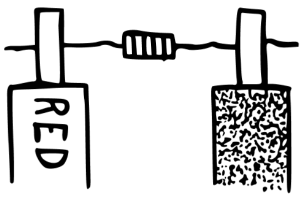
MULTIMETERS

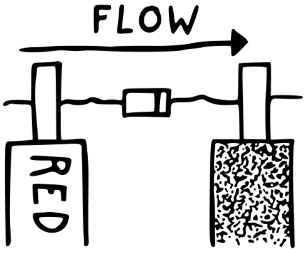


Multimeters come in different shapes and sizes, but the most common type is probably the hand-held, battery powered variant. It can measure a bunch of different things: voltage, current, resistance, continuity. Some have additional capabilities, allowing you to check capacitance, oscillation frequency or the forward voltage drop of a diode.

When shopping for one, you'll probably notice that there are really expensive models boasting about being TRUE RMS multimeters. For our purposes, this is really kind of irrelevant, so don't feel bad about going for a cheap model!

Using a multimeter is actually really straightforward. Simply attach two probes to your device – the one with a black cable traditionally plugs into the middle, while the red one goes into the right connector. Next, find whatever you want to measure and select the corresponding mode setting.



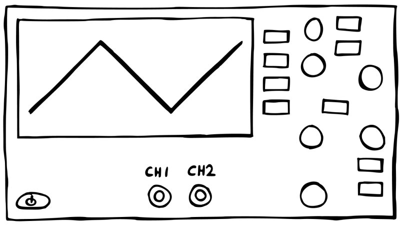


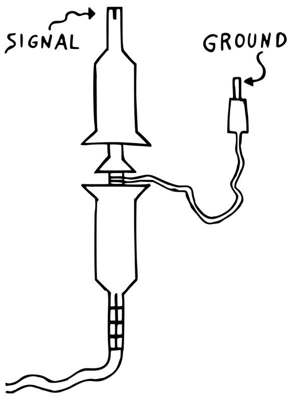
In some cases, it doesn't matter which probe you connect to which component leg or point in your circuit. This is true for testing resistors, non-polarized capacitors (foil/film, ceramic, teflon, glass etc.), continuity[33] or AC voltage.

In others, you'll have to be careful about which probe you connect where. For testing the forward voltage drop of a diode, for example, **the multimeter tries to push a current from the red to the black probe**. Here, you'll have to make sure the diode is oriented correctly, so that it doesn't block that current from flowing. For testing a DC voltage, you want to make sure the black probe is connected to ground, while you use the red one to actually take your measurement.

[33] Just a fancy word for saying that two points are electrically connected.

OSCILLOSCOPES



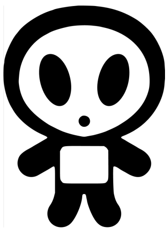


While multimeters are fairly cheap and compact, oscilloscopes are usually somewhat pricey and bulky. **If you're willing to make the investment, they are a huge help with the troubleshooting process, though**. Using one is, again, surprisingly straightforward – if you manage to work your way through the sometimes quite convoluted UI, especially on digital models.

To start using your scope, simply attach a probe to one of the channel inputs. These probes usually have two connectors on the other end: a big one that you operate by pulling the top part back – and a smaller one, which is usually a standard alligator clip. The latter needs to be connected to your circuit's ground rail, while you probe your oscillation with the former. Now what the oscilloscope will do is **monitor the voltage between the two connectors over time and draw it onto the screen as a graph**. Here, the x-axis is showing time, while the y-axis is showing voltage. You can use the device's scaling controls to zoom in on a specific part of your waveform.

Usually, digital oscilloscopes will also tell you a couple useful things about the signal you're currently viewing: minimum/maximum voltage level, oscillation frequency, signal offset. Some even offer a spectrum analyzer, which can be useful to check the frequencies contained in your signal.

BUILD GUIDE



MODULE ASSEMBLY APPENDIX

Before we start building, let's take a look at the complete **mki x es.edu S&H/NOISE** schematics (see next page) that were used for the final module design and PCB fabrication. Most components on the production schematics have denominations (a name – like R1, C1, VT1, VD1, etc.) and values next to them. Denominations help identify each component on the PCB, which is particularly useful during **calibration**, **modification** or **troubleshooting** of the module.

The module combines three parts that partly complement each other: the noise generator, the sample and hold unit and the slew (glide) processor. In order to save space on the front panel of the module, connections between parts are done via normalling lugs of jack sockets instead of switches as described in the breadboarding section of this manual.

XS1 is the **Sample and Hold (S&H) signal input** jack socket. As you can see in the schematics, it is normalled to the **white noise** output via the switching lug of the jack socket, which means if nothing is patched into XS1, a white noise is applied to the input and the S&H section will output stepped random voltages, but as soon you patch some other signal into XS1, the white noise lug is disconnected and the white noise is replaced with a signal from the connected patch cable. **XS2** is the S&H **clock input** jack socket. As you can see it is normalled to the internal clock generator, so if no external clock source is patched into XS2, the internal clock is prioritized. **XS3** is the **S&H output** jack socket.

XS4 is the **slew processor input** and it is normalled to the S&H output, so if nothing is connected to the slew input, you have a duplicate of the S&H output on **XS5**, the **slew output** jack with the slew rate potentiometer all way counterclockwise and smoothed-out random voltages as you turn the slew rate potentiometer clockwise. Obviously, the slew processor can work on its own and can be used to apply slew (or glide) to any control voltage, for example, you can feed the CV OUT from the DIY.EDY Sequencer to XS4 and feed the CV from XS5 to the 1V/Oct input of the DIY.EDU VCO. By adjusting the slew rate pote tiometer you can achieve a glide between notes produced by the Sequencer. **XS6** is the **Pink Noise output**, **XS7** is the **White Noise output** and **XS8** is the **Blue Noise output**.

All jack sockets are the very same we've already been using on the breadboard for interfacing with other devices. In our designs, we use eurorack standard 3,5 mm jack sockets (part number WQP-PJ301M-12).

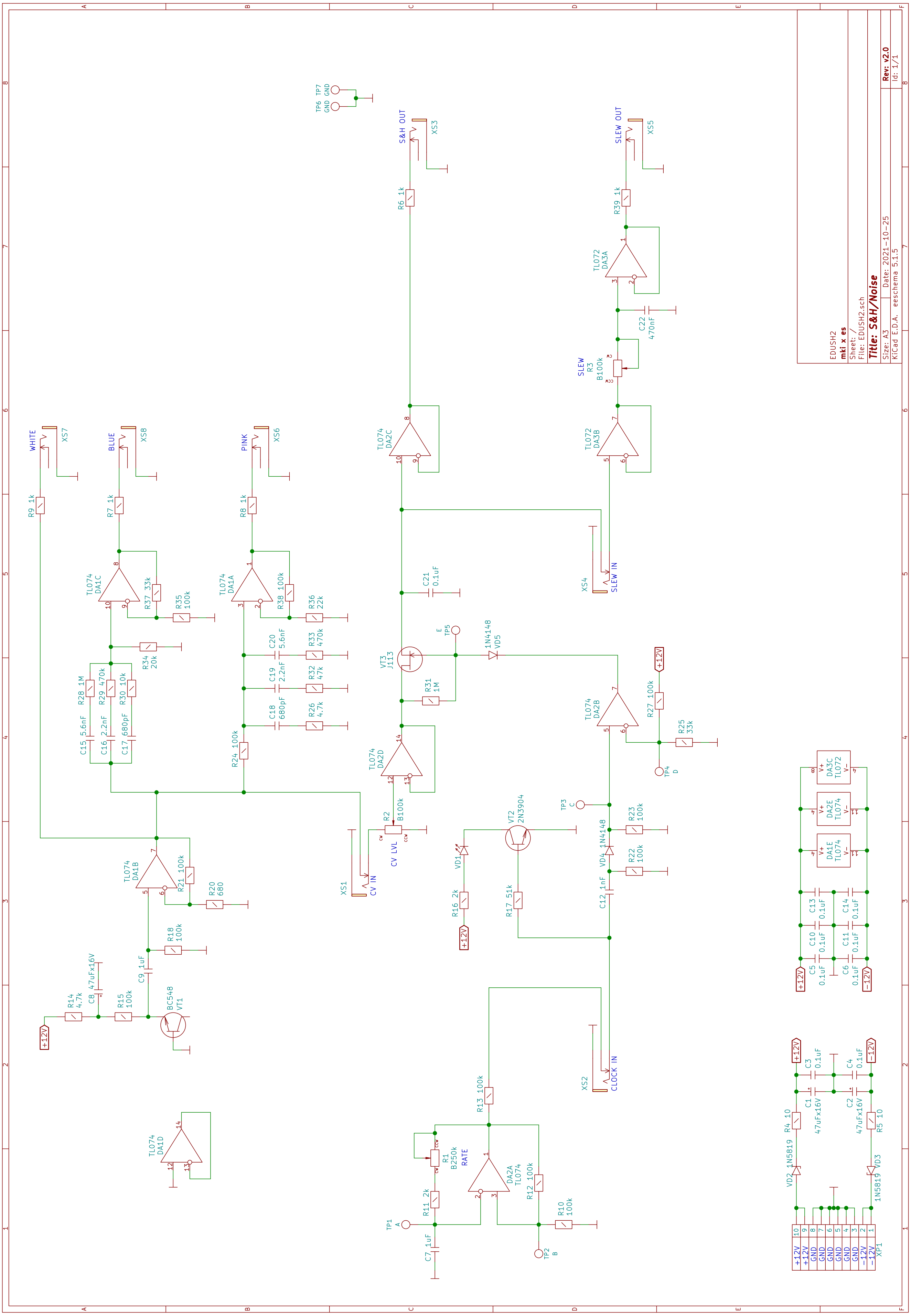
XP1 is a standard eurorack **power connector**. It is a 2x5 male pin header with a key (the black plastic shroud around the pins) to prevent accidental reverse polarity power supply connection. This is necessary because connecting the power incorrectly will permanently damage the module.

VD2 and **VD3** are **schottky diodes** that double-secure the reverse polarity power supply protection. Diodes pass current only in one direction. Because the anode of VD2 is connected to +12 V on our power header, it will only conduct if the connector is plugged in correctly. If a negative voltage is accidentally applied to the anode of VD2, it closes, and no current passes through. The same goes for VD3 which is connected to -12 V. Because schottky diodes have a low forward voltage drop, they are the most efficient choice for applications like this.

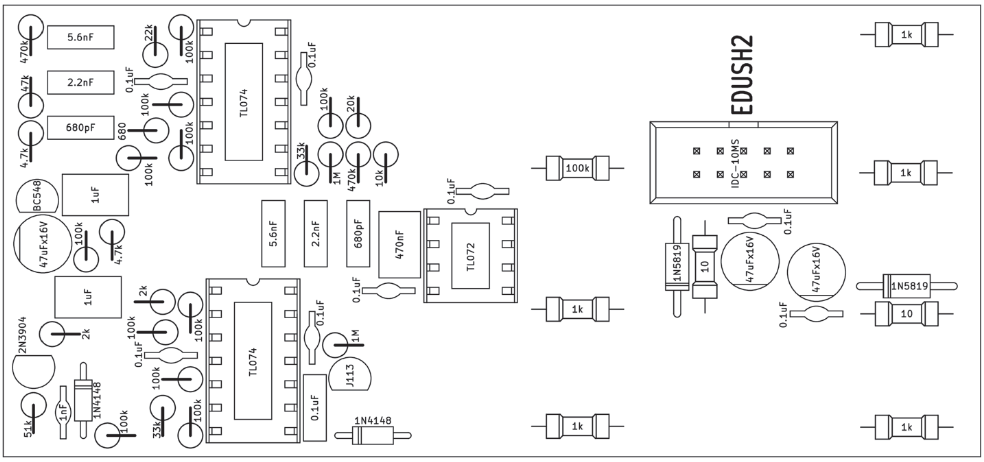
Next, we have two **10 Ohm resistors** (**R4** and **R5**) on the + and – 12V rails, with **decoupling** (or **bypass**-) capacitors **C1 – C4**. These capacitors serve as energy reservoirs that keep the module's internal supply voltages stable in case there are any fluctuations in the power supply of the entire modular system. In combination with R4 and R5, the large 47 microfarad pair (C1 and C2) compensates for low frequency fluctuations, while C3 and C4 filter out radio frequencies, high frequency spikes from switching power supplies and quick spikes created by other modules. Often another component – a **ferrite bead** – is used instead of a 10 Ohm resistor and there's no clear consensus among electronic designers which works best, but generally for analogue modules that work mostly in the audio frequency range (as opposed to digital ones that use microcontrollers running at 8 MHz frequencies and above), resistors are considered to be superior.

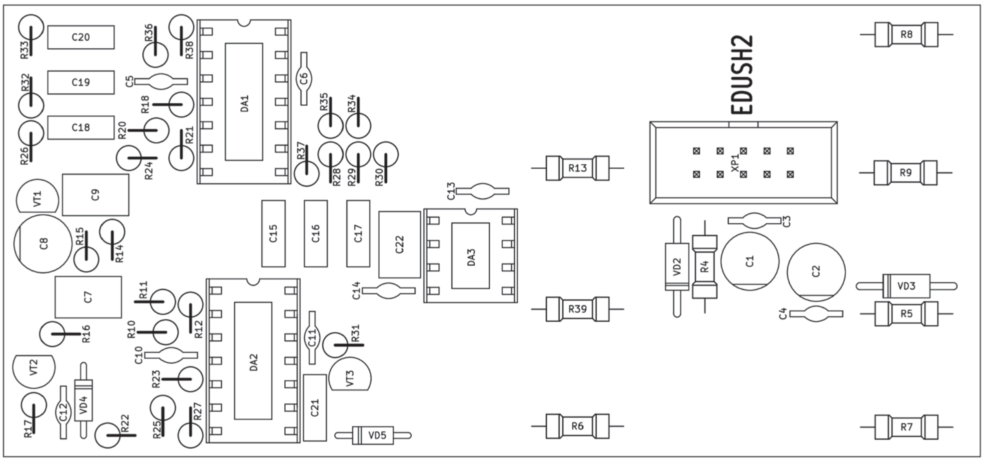
Another advantage of 10 Ohm resistors is that they will act like **slow "fuses"** in case there's an accidental short circuit somewhere on the PCB, or an integrated circuit (IC) is inserted backwards into a DIP socket. The resistor will get hot, begin smoking and finally break the connection. Even though they aren't really fuses, just having them there as fuse'substitutes is pretty useful - **you'd rather lose a cent on a destroyed resistor than a few euros on destroyed ICs**.

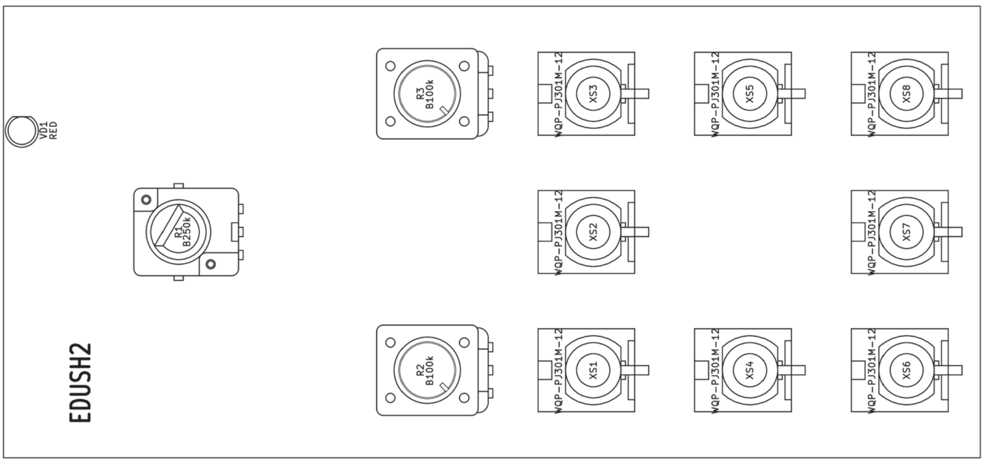
Capacitors **C5 – C14** are additional decoupling capacitors. If you inspect the PCB, you will see that these are placed as close to the power supply pins of the ICs as possible. For well-designed, larger PCBs you will find decoupling capacitors next to each IC. Like the others, their job is to simply compensate for any unwanted noise in the supply rails. If the input voltage drops, then these capacitors will be able to bridge the gap to keep the voltage at the IC stable. And vice-versa - if the voltage increases, then they will be able to absorb the excess energy trying to flow through to the IC, which again keeps the voltage stable. Typically, 0.1 uF capacitors are used for this purpose.

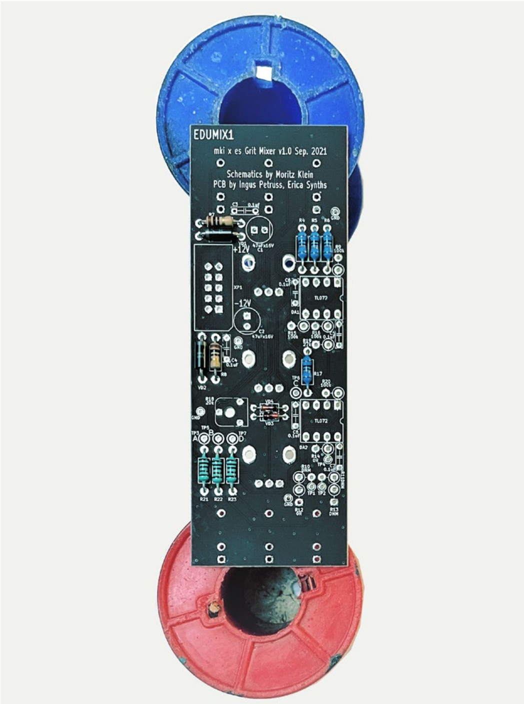

EDUSH2
mki x es
Sheet: /
File: EDUSH2.sch
Title: S&H/Noise
Size: A3
Date: 2021-10-25
KiCad E.D.A. eeschema 5.1.5
Rev: v2.0
Id: 1/1
S&H OUT
SLEW OUT
SLEW IN
CLOCK IN
CV IN
CV LVL
RATE
SLEW
WHITE
BLUE
PINK

Before you start soldering, we highly recommend printing out the following part placement diagrams with designators and values. Because some of our PCBs are rather densely populated, this will help you to avoid mistakes in the build process.

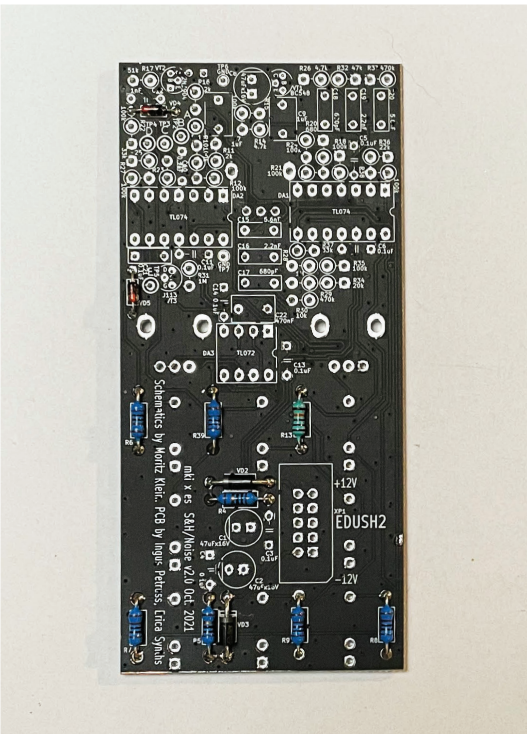




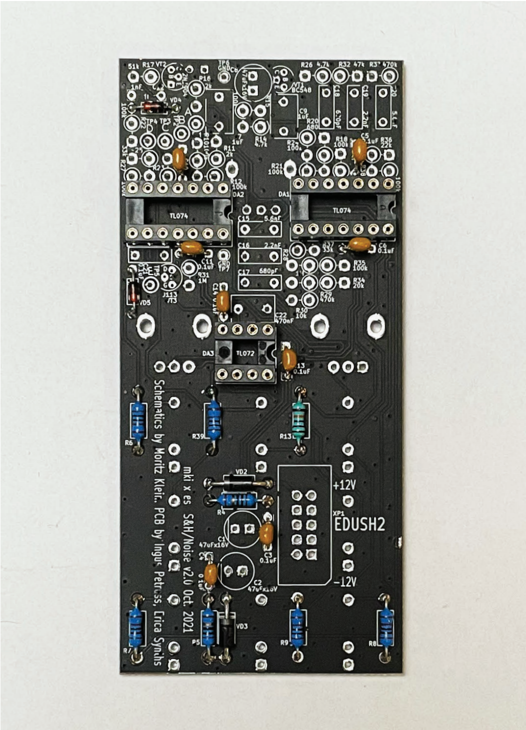




Place the S&H/Noise PCB in a PCB holder for soldering or simply on top of some spacers (I use two empty solder wire coils here).

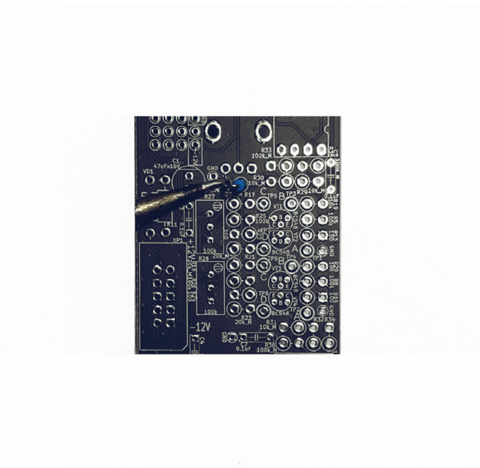


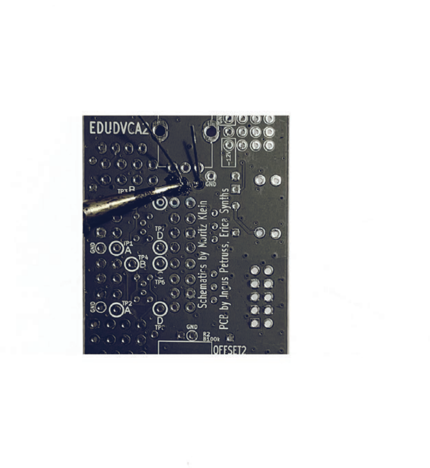
I usually start populating PCBs with lower, horizontally placed components. In this case, these are a **few resistors**, **switching diodes** and **power protection diodes.** Bend the resistor leads and insert them in the relevant places according to the part placement diagram above. All components on the PCB have both their value and denomination printed onto the silkscreen. If you are not sure about a resistor's value, use a multimeter to double-check. Next, insert the diodes. Remember – **when inserting the diodes**, **orientation is critical!** A thick white stripe on the PCB indicates the cathode of a diode – match it with the stripe on the component. Flip the PCB over and solder all components. Then, use pliers to cut off the excess leads.

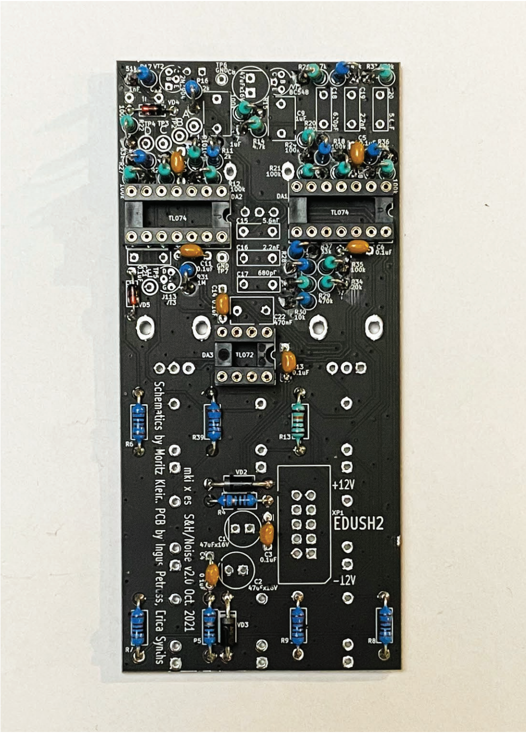


Next, **insert the first DIP socket**, hold it in place and solder one of the pins. Continue with the **next DIP socket.** Make sure the DIP sockets **are oriented correctly** – the notch on the socket should match the notch on the PCB silkscreen. Now, turn the PCB around and solder all remaining pins of the DIP sockets. Then proceed with the **ceramic capacitors.** Place the PCB in your PCB holder or on spacers, insert the capacitors and solder them like you did with the resistors diodes before. Now your PCB should look like this:

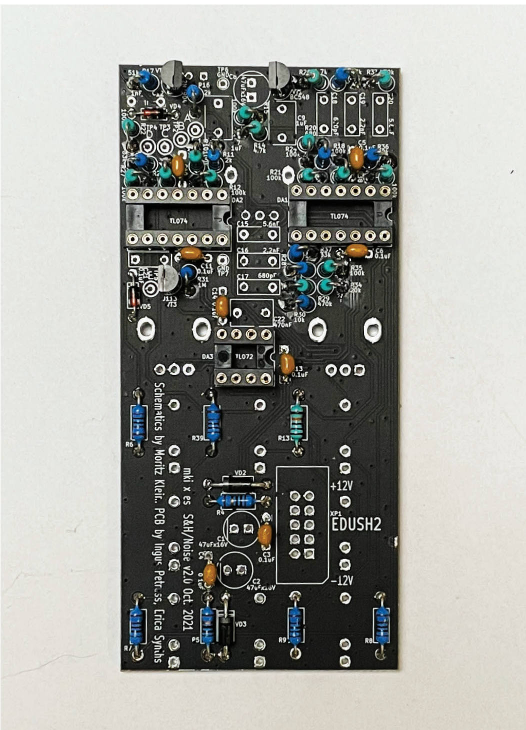
In order to save space on the PCB, some of our projects, including the S&H/Noise, have **vertically placed resistors.** The next step is to place & solder those. Bend a resistor's legs so that its body is aligned with both legs and insert it in its designated spot. Then solder the longer lead from the top side of the PCB to secure it in place, turn the PCB around and solder the other lead from the bottom. You can insert several resistors at once. Once done with soldering, use pliers to cut off excess leads.





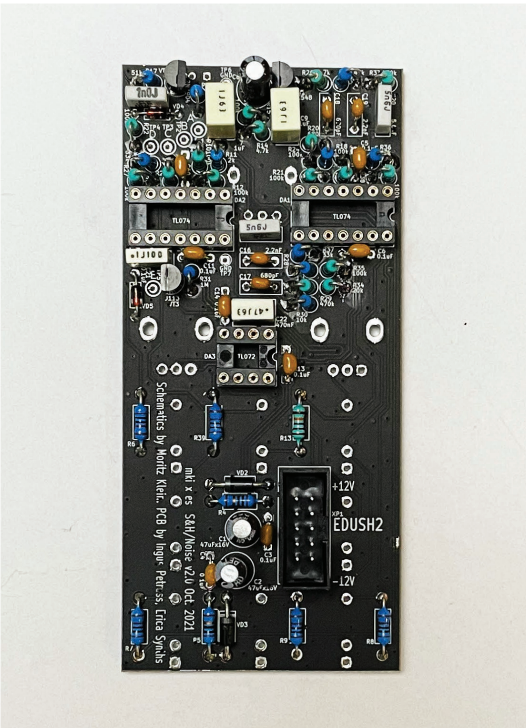


Once you are done with soldering all resistors, your PCB should look like this:

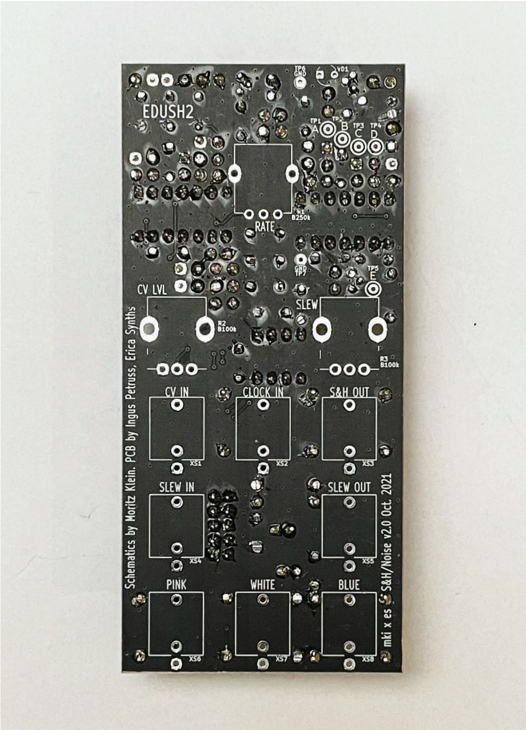


Next up: inserting & soldering the transistors. Make sure you align the transistors with the marked outline on the silkscreen – **orientation is critically important here.** Also, we have **three different** kinds of **transistors** on this module – a BC548 transistor in the noise generator, a 2N3904 transistor in the LED driver for S&H clock rate and a J113 transistor that acts as a switch in the S&H circuit. Pay close attention on the placement of the transistors!

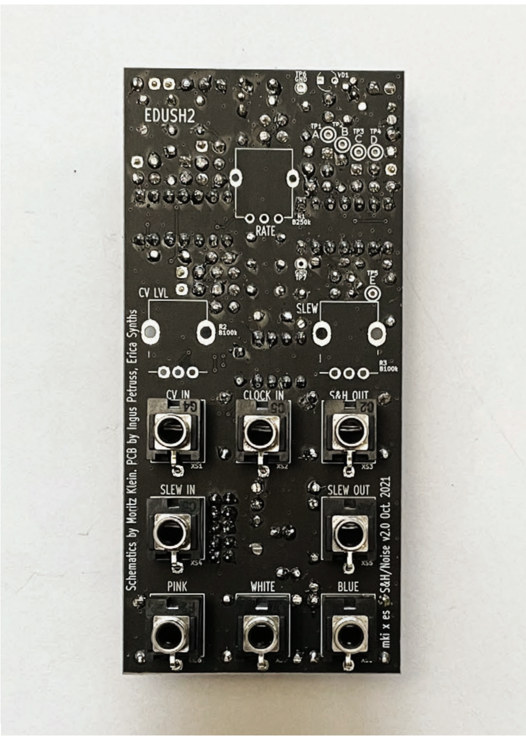
Next, insert **film capacitors**, remaining **ceramic capacitors** and solder them. Capacitors supplied with the kit may differ from ones on the photo to the left, therefore pay attention on capacitor values. Also, insert & solder the **electrolytic capacitors.** Electrolytic capacitors are bipolar, and you need to mind their orientation. The positive lead of each electrolytic capacitor is longer, and there is a minus stripe on the side of the capacitor's body to indicate the negative lead. On our PCBs, the positive pad for the capacitor has a square shape, and the negative lead should go into the pad next to the notch on the silkscreen.



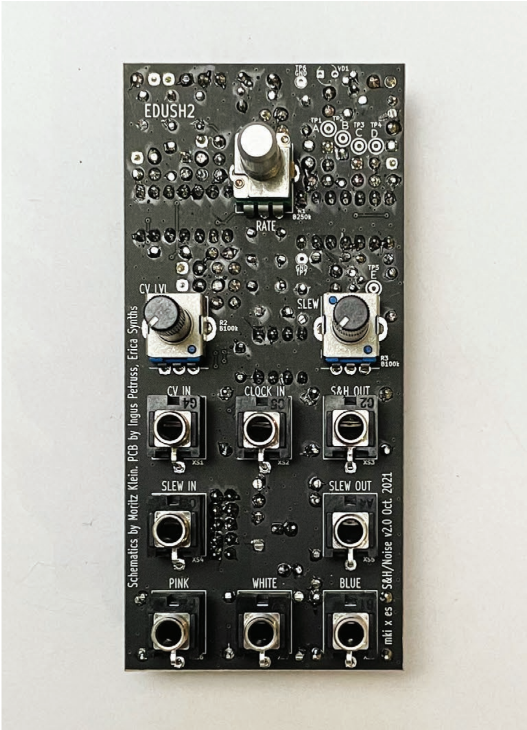
Then complete the component side of the Envelope PCB by soldering the PSU socket. Make sure the orientation of the socket is as shown in the picture below – the arrow pointing to the first pin is aligned with a notch on the silkscreen. The key on the socket will be facing right. Now your PCB should look like this:



Now, turn the PCB around and inspect your solder joints. **Make sure all components are soldered properly and there are no cold solder joints or accidental shorts**. Clean the PCB to remove extra flux, if necessary.

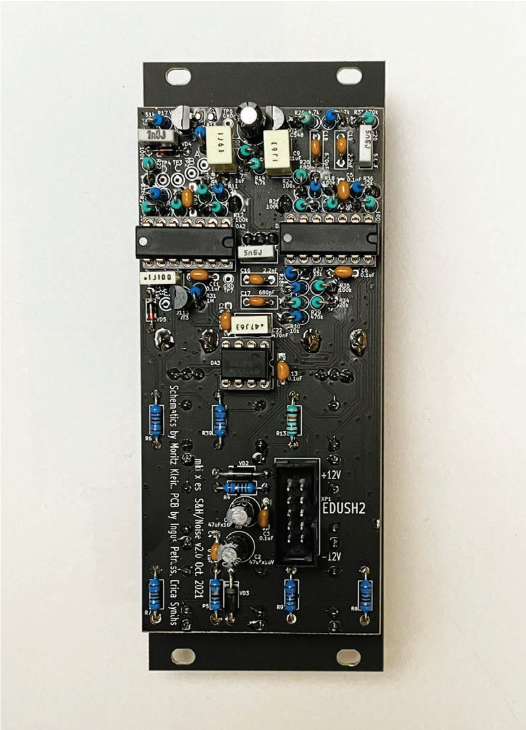


Insert the jack sockets and solder them.



Insert the potentiometers, but don't solder them yet! Fit the front panel and make sure that the potentiometer shafts are aligned with the holes in the panel – and that they're able to rotate freely. Now, go ahead and solder the potentiometers.

Now, remove the front panel and **insert the LED** in the relevant place on the PCB, but do not solder it, yet! Orientation of the LED is important – check the silkscreen! A notch on the silkscreen indicates the **cathode of the LED** (a shorter lead next to a notch on the LED) and the longer lead – the **anode of the LED** – has to go into a hole with square-like polygon on the PCB. **Fit the front panel** again and **fix it** with the nuts on jack sockets and S&H rate potentiometer. Now, solder the LED. Then install the knob on the S&H rate potentiometer. We are almost done!



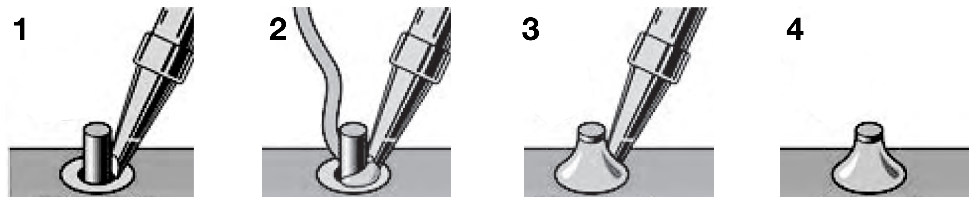
Now, **insert the ICs into their respective DIP sockets.** Mind the orientation of the ICs – match the notch on each IC with the one on its socket.

Congratulations! **You have completed the assembly of the mki x es.edu S&H/Noise module!** It does not need any calibration and, if assembly is correct, it should work straight away. Connect it to your eurorack power supply and switch it on. If there's no "magic smoke", it's a good sign that your build was successful. Rotate the S&H Rate knob and check if the LED is blinking at different rates. Time to bring some randomness into your modular setup! **Enjoy!**

SOLDERING APPENDIX

If you've never soldered before – or if your skills have become rusty – it's probably wise to check out some **THT** (through-hole technology) **soldering tutorials on YouTube**. The main thing you have to remember while soldering is that melted solder will flow towards higher temperature areas. So you need to make sure you apply equal heat to the component you are soldering and the solder pad on the PCB. The pad will typically absorb more heat (especially ground-connected pads which have more thermal mass), so keep your soldering iron closer to the pad on the PCB. It's critically important to dial in the right temperature on your soldering station. I found that about 320 °C is the optimal temperature for most of parts, while for larger elements like potentiometers and sockets, you may want to increase that temperature to **370 °C.**

Here's the recommended soldering sequence:

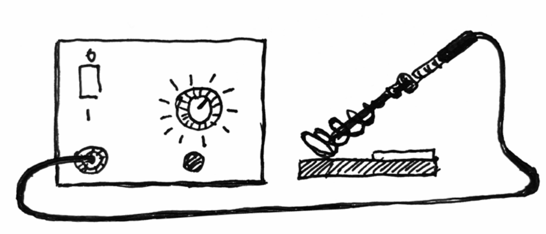


Heat part and pad 2 - 3 sec	Add solder	Continue heating 1 -2 sec.	Let cool

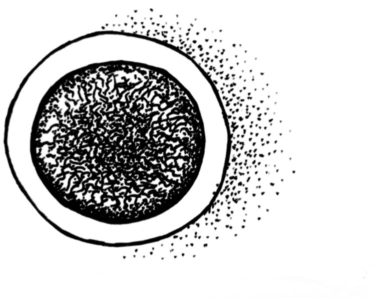
After you have completed soldering, inspect the solder joint:

Perfect	Too much solder	Not enough solder	Cold joint	Too much heat	Short

DIY electronics is a great (and quite addictive) hobby, therefore we highly recommend you invest in good tools. In order to really enjoy soldering, you'll need:

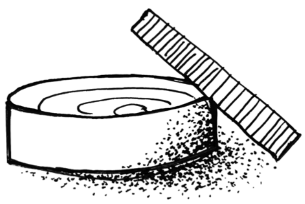


A decent soldering station. Top-of-the-line soldering stations (brands like Weller) will cost 200€ and above, but cheaper alternatives around 50€ are often good enough. Make sure your soldering station of choice comes with multiple differently-sized soldering iron tips. The most useful ones for DIY electronics are flat, 2mm wide tips.



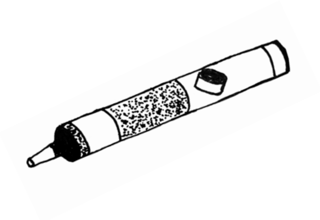
When heated up, the tips of soldering irons tend to oxidize. As a result, solder won't stick to them, so you'll need to clean your tip frequently. Most soldering stations come with a **damp sponge for cleaning the iron tips** – but there are also professional solder tip cleaners with **golden curls** (not really gold, so not as expensive as it sounds). These work much better because they do not cool down the iron.

Solder wire with flux. I find 0,7mm solder wire works best for DIY projects.

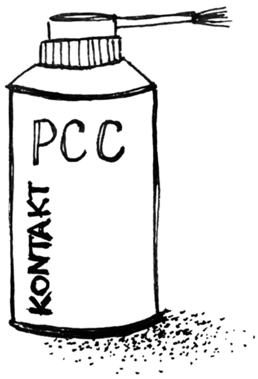


Some **soldering flux** paste or pen will be useful as well.

Cutting pliers. Use them to cut off excess component leads after soldering.



A solder suction pump. No matter how refined your soldering skills are, you will make mistakes. So when you'll inevitably need to de-solder components, you will also need to remove any remaining solder from the solder pads in order to insert new components.



Once you have finished soldering your PCB, it's recommended to remove excess flux from the solder joints. **A PCB cleaner** is the best way to go.

All of these tools can be found on major electronic components retailer websites, like Mouser, Farnell and at your local electronics shops. As you work your way towards more and more advanced projects, you'll need to expand your skillset and your tool belt – but the gratification will be much greater.

"When some systems are stuck in a dangerous impasse, randomness and only randomness can unlock them and set them free."

– Nassim Nicholas Taleb