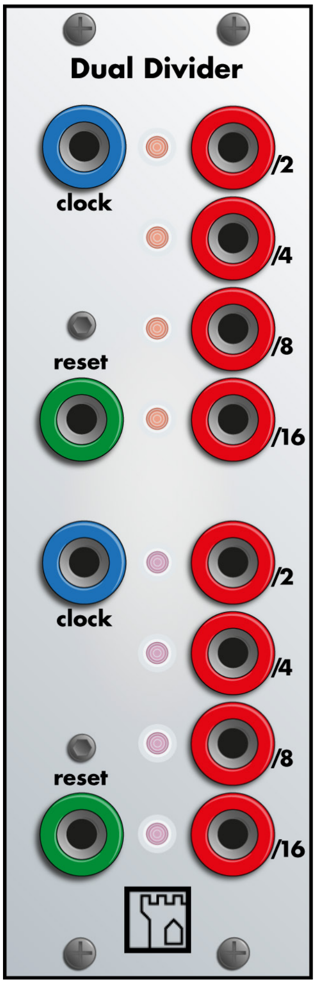
Castle Rocktronics

006 – Dual Divider

Two clock/frequency dividers with reset

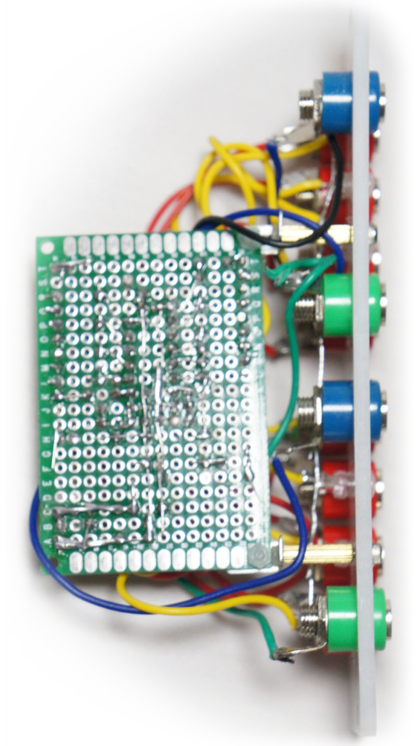


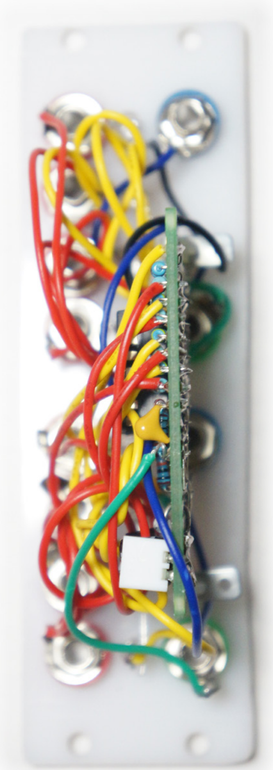
Comments, suggestions, questions and corrections are welcomed & encouraged:
contact@castlerocktronics.com

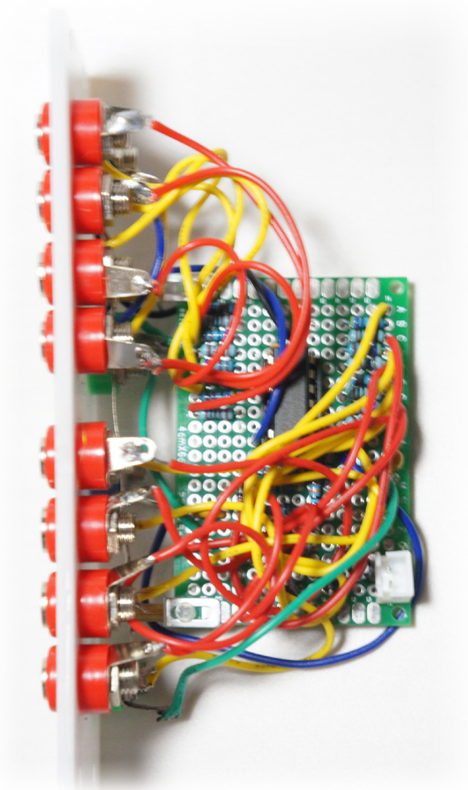
Contents

Version history:
07/April/2015 – First draft

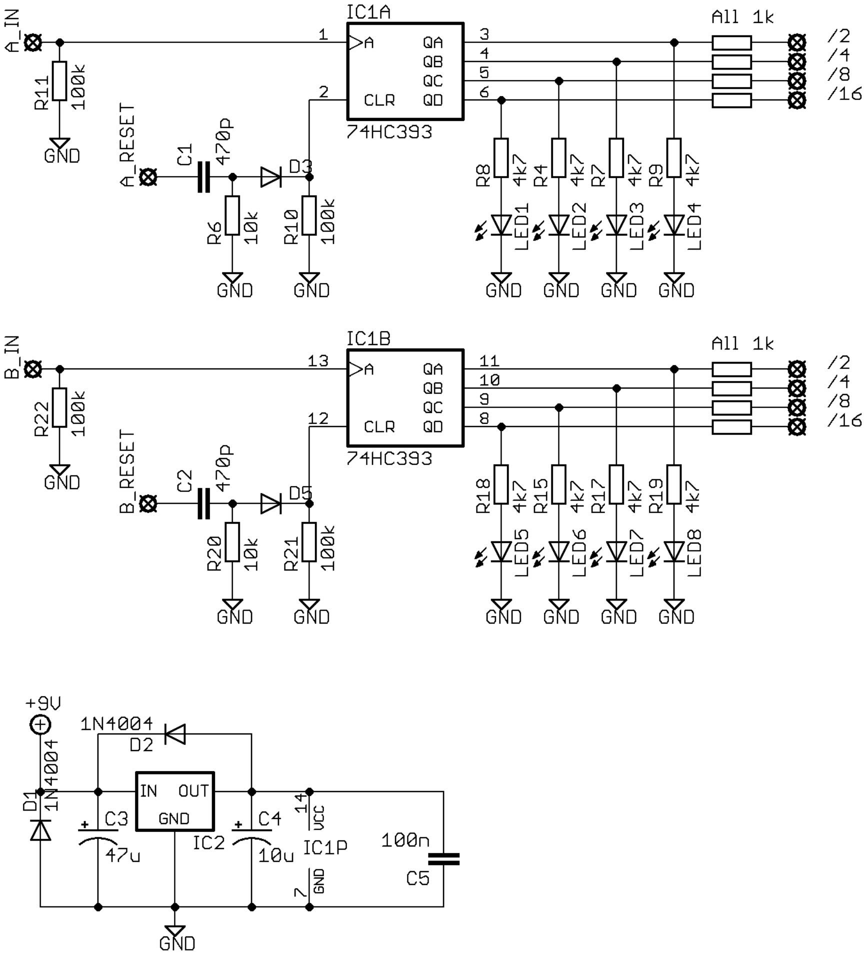
0. Build Documents
0.1 Photos







0.2 Schematic



0.2 Bill of Materials

ICs

78L05	x1
74HC393	x1

Resistors

1kΩ	x8
4.7kΩ	x8
10kΩ	x2
100kΩ	x4

Capacitors

470pF	x2
100nF	x1
10µF	x1
47µF	x1

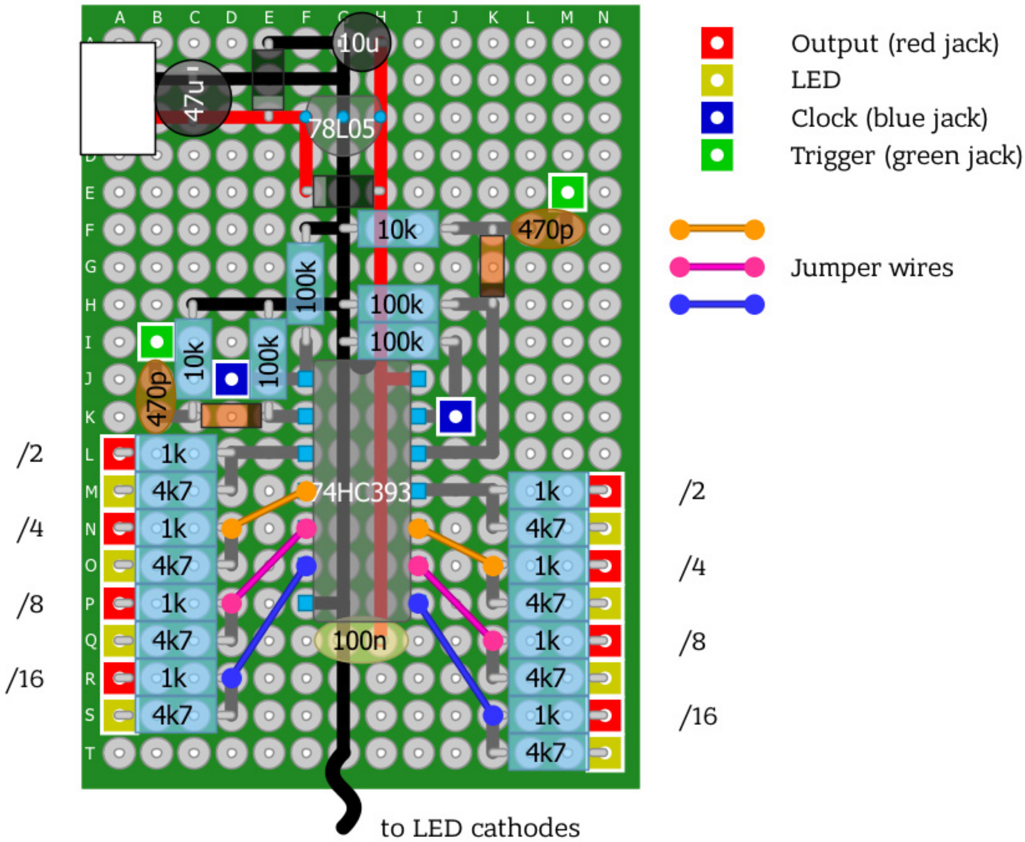
Diodes

1N4004	x1
1N4148	x1
LEDs	x8

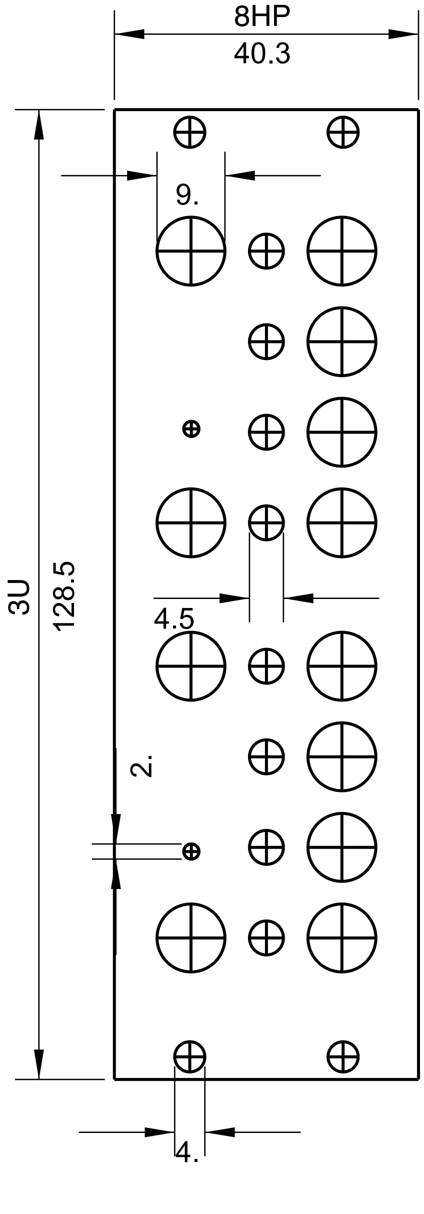
Connectors

Bannana jacks	
Blue	x2
Green	x2
Red	x4
XH-2.54	x1

0.3 Layout


A B C D E F G H I J K L M N
10u
47u
78L05
10k
470p
100k
100k
100k
10k
100k
470p
74HC393
100n
/2
/4
/8
/16
1k
4k7
Output (red jack)
LED
Clock (blue jack)
Trigger (green jack)
Jumper wires
to LED cathodes

0.4 Drill Template



1. Explanation & Analysis

1.1 A quick build

This is another chip-to-jack module, just like *#004 – Dual Shift Divider*. The great thing about modules like this is that they are really easy to build! Who doesn't love a module that can be done in an afternoon? Even better is that I don't have to write much in the way of an article so it's faster for me too. What is worth spending some time on though, is a little bit about different families of logic.

1.2 The 7400 series

The dual divider takes advantage of the lovely 74HC393, the first (but alMOSt certainly not the last) logic chip we are going to use that is not from the 4000 series. The 7400 series is just another collection of logic chips, though instead of CMOS most of them use TTL technology – a.k.a. transistor-transistor logic. This means that instead of the **C**omplimentary **M**etal-**O**xide **S**emiconductor (**CMOS**) transistors used in the 4000 series, the 7400 series is based on **B**ipolar **J**unction **T**ransistors (**BJT**). Without going into tons of detail about semiconductors, just know that BJTs came before MOS.

1.2.1 TTL vs. CMOS

I know what you are thinking now: newer means better, right? Well, yes and no. CMOS chips do have a number of advantages over TTL chips. In fact, the 4000 series was introduced to provide alternatives to the 7400 series. Notice I used the word "alternatives" and not replacements. The CMOS technology used in the 4000 series has a number of desirable qualities:

- Higher input impedance
- Lower output impedance
- Lower power consumption
- Wider range of usable supply voltages
- Simpler circuit design (cheaper!)

All of these things are obviously pretty great. Awesome even. So why did I say "alternative" over replacement? One reason: speed. TTL chips are still considerably faster than CMOS. CMOS usually can't handle much more than 1MHz, while TTL can blast along at speeds from 10MHz to around 200MHz! Luckily for us, human ears only hear up to 20kHz, and that's if you have amazing hearing. This means we can get by with CMOS and take advantage of all those really nice features without having to worry about them being too slow.

So why are we using a 7400 series chip in this module? Because there is no 4000 series equivalent. The 4000 series does have dividers – check out the 4020, 4024 & 4040 – but it does not have any that have two dividers on the same chip. I like having more than one of the same thing in a module, especially since you can chain things like shift registers and dividers together anyway.

There is another problem though – TTL and CMOS are not compatible. The outputs and inputs of CMOS chips behave a little too differently for us to be able to hook them up directly. There are a few solutions, though we are going for the simplest:

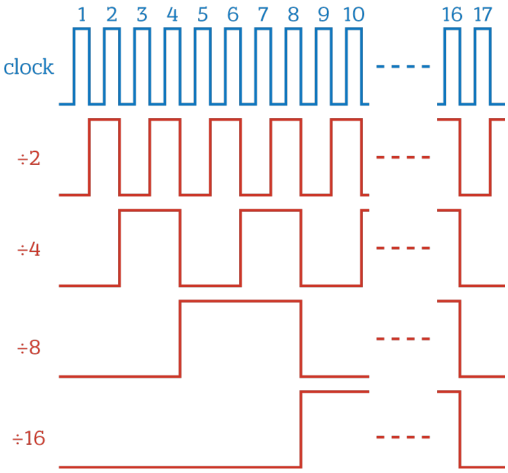
1.2.2 74HC00 chips

You may have noticed that the bill of materials demands that you pick up the 74HC393. Stress on the **HC**. Well, with the advantages of CMOS proving to be so great, manufacturers started making **H**igh-speed **C**MOS version of the 7400 series chips, as well as special HCT chips, which are compatible with TTL logic. These chips are still slower than the best TTL chips, but they are faster than the 4000 series. So why don't the make the 4000 series chips faster? The answer is that is not what the 4000 series is about. They sacrifice speed – though being more than fast enough for synth use – in favour of keeping all those other nice advantages.

1.3 What is a divider anyway?

Dividers, also called "binary ripple counters", are just chips that take a square wave input and run it through a bunch of cascaded flip-flops to half its speed, then half it again, and again, and again *(fig 1.3a - next page)*

Sometimes it will do this as many as 14 times, though the 74HC393 only does it four times. This can have a lot of useful applications, most of which have to do with timing and keeping things in sync even if they need to run at different



(fig 1.3a)

speeds. For us, this has two particularly musical uses: rhythm and pitch.

1.3.1 Using it for rhythm

In Western music, rhythm is usually "divisive". The idea is that a steady pulse is divided into equal sub-divisions – usually two, sometimes three, but you can use other prime numbers if you are pretentious – and then those may be divided again into further smaller parts. This can be seen in the American names for note durations: the whole note, half-note, quarter-note, 8th-note, 16th-note, etc. However, our divider goes in the opposite direction: since it creates an output at half the speed, this would be like using a note value twice as large. Two quarter-notes fit into a half-note.

As for using this in our modular, this means we can use one of our LFOs to create a steady pulse of changes that we can treat as the 16th-notes of our tempo, and then use the divider to generate the 8th-notes, quarter-notes, half-notes and whole-notes. This will let us clock different parts of our synth at different speeds while keeping it rhythmical.

1.3.2 Using it for pitch

Similar to rhythm, our ears treat pitch in a very mathematical way. An octave up from any pitch is double the frequency, the octave below is half the frequency. Since our divider halves frequencies, this means we can use it to get some nice fat sub-octave doubling, really fattening up the sounds we can get out of our synth. This is actually the technique used in lots of commercial analog synths and modules to get a nice square-wave sub-oscillator for cheap, though in this case they tend to use a single flip-flop rather than the cascaded flip-flops in a divider chip. One or two octaves down is often enough... nah, who am I kidding. We all want at least 4 down for speaker blowing bass action.

1.3.3 Using it to generate an average

There is one last use, and also something to keep in mind about the outputs of the divider. Each successive division will get closer and closer to a perfect square-wave, averaging out irregularities in the previous division. This can be used to smooth out faster streams of random bits, but it also means that intentional irregular rhythms at the output will tend to get lost at the outputs. This also means that if the input's pulse width gets modulated this ends up getting lost in the output. Just something to keep in mind.

1.4 One last note

Something that you should note about the output of the divider is revealed in *(fig 1.3a)*. Look carefully at where the output transitions happen in relation to the input waveform. You will notice that they happen on the falling edge. This isn't a big deal for making sub-octaves, but it is a big deal if you are using it for rhythm as MOST inputs act on the rising edge. This can be interesting to play with, as you can use it to make your main clock sound off-beat compared to everything else. However, if you don't want this you might want to run your clock through an inverter first, and don't worry, that's the next module!

2. Modifications

2.1 Faster reset

I decided on using much smaller caps in the leading edge detector than the ones I used in module *#004 – Dual Shift Register*. This is primarily because I knew I was going to use this module for fat bassline sounds and I wanted the option of being able to do audio rate sync on it for hyper-aggressive sync bass. BZZZZZZT.

2.2 Bigger current limiting resistors

I ran into a bit of an issue with some of the other modules when I did excessive cable stacking and I managed to track it down to the fact I was using 1kΩ resistors for the LEDs, which was causing things to get loaded down and stop behaving properly. What's more, 1kΩ was a bit wasteful in the first place. I've been wasting more current than I have to on making the LEDs excessively bright.