WHAT IS DODECA?

When I started looking through modulargrid at options for converting midi into usable signals for Eurorack modular I saw that while there were a great number of fixed configurations to suit different needs none were a good match for my specific needs.

So Kris Northern of Magpie Modular and Jim Matheson of Neutron Sound set out to create a DIY Open Hardware, Open Source Teensy based extensible platform which you can build and custom code to fit your own needs. We named this platform 'Dodeca' (a greek word meaning 12 referring to the 12 outputs)

From one stereo midi jack you can custom configure 12 outputs to suit your needs This includes Gates, & Trig outouts, 1v/OCT Midi Voice, and Midi Continuous Controller to Unipolar 0/+5v CV signal . Additionally, you can daisy-chain the units and expand until you hit the bandwidth limit of the MIDI spec (which is a paltry 3,125 bytes per second).

If you don't want to mess around with programming We have already coded several default configurations which are available on our Github. At the time of writing we have configurations for a 6hp Default setup, A 12hp DBLDECCA configuration specifically for Elektrons Octatrack and a 12HP DBLMONO configuration for Elektrons Monomachine. All of which we will cover later.

We offer the PCBs, a variety of different panels but most likely won't be offering built modules. If you really want one of these and don't have the DIY chops to build your own I recommend you join our facebook group and find a builder who is making these to order.

ORDER ALL THE THINGS!

First, you should decide which implementation you are going to use and order accordingly ie; The DBLDECCA for the Octatrack has two dodeca boards and therefore requires 2x of everything. You can skip to the end of this manual and see explanations of some of the firmwares we have created.

PCBs
Panels
Midi to Stereo cable
Teensy 3.2

Mouser cart:
https://ca.mouser.com/ProjectManager/ProjectDetail.aspx?State=EDIT&ProjectGUID=0082110f-d597- 4c76-925b-8146d7b16813

Note: Trimmers are optional but recommended if you are planning to use 1/vOCT on some of the outputs. Trimmers are one of the more expensive parts, you can have 0-3 of them.

In the BOM you will see 9x 20k resistors, 3x 16.9k resistors, and 3x trimmers.
If you want to remove trimmers, then remove also the 16.9k resistors and add 1 20k for each trimmer you remove.

For example with 1 trimmer there would be 11x 20k, 1x 16.9k, and 1x trimmer.

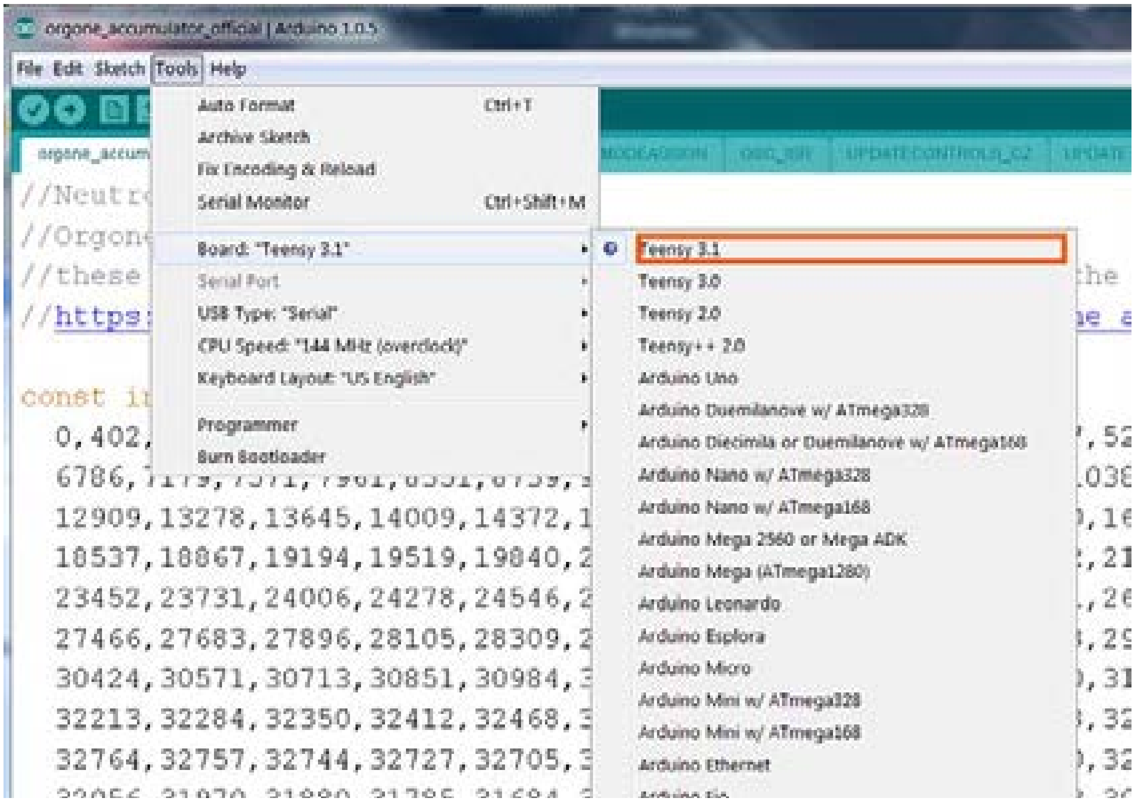
The Dodeca uses 0805 SMD components but they are not very small except one of the diodes. ICs are SOIC.
It is quite easy and actually faster than through hole because there is no “lead bending” and “lead cutting” steps.

LED brightness and resistors.
The BOM calls for 1k resistors which is fine for super efficient red ones, but some other colors have a higher voltage drop, and need a much lower resistance. You may want to experiment if you choose different LEDs

PROGRAMMING THE TEENSY

Before you start assembly, it is a good idea to program the Teensy to become familiar with the Arduino software and the Teensy loader.

The Teensy is a microcontroller which you can program via a USB cable
First go to this site and follow the instructions to install Arduino and the Teensyduino software.
https://www.pjrc.com/teensy/td_download.html
Note! Quite often, arduino is one version ahead of teensyduino, so get teensyduino first, and then get the latest version that is supported from the arduino “older versions” page.
Connect your teensy to the computer via USB A to Micro-B cable Open the arduino IDE program and set the board to Teensy 3.1/3.2

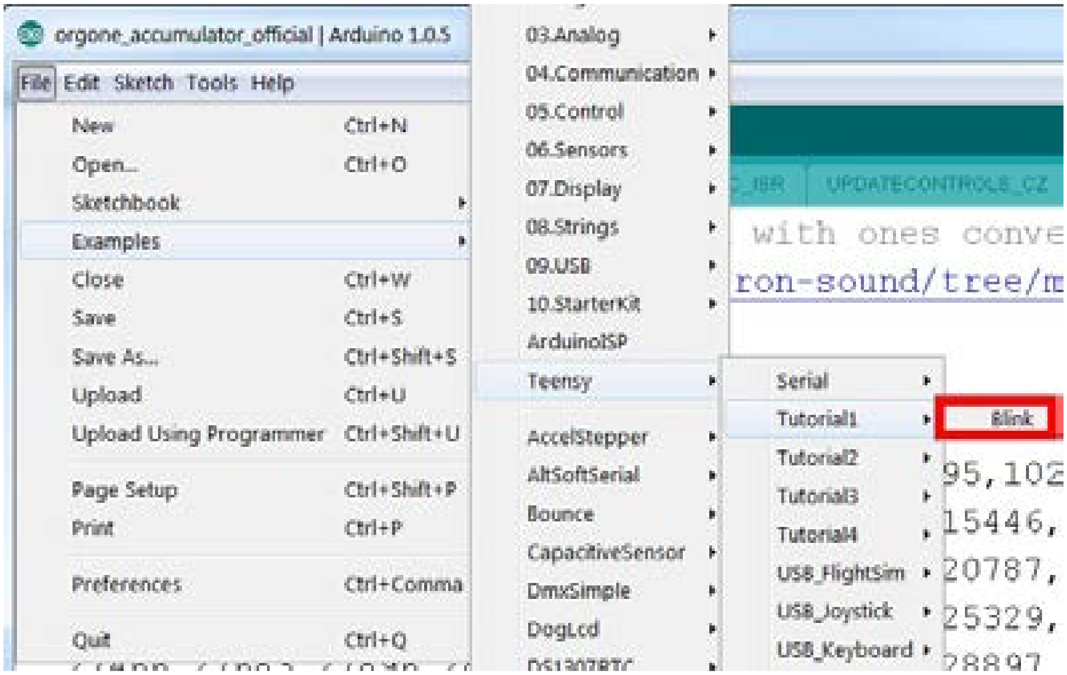

orgone_accumulator_official | Arduino 1.0.5
File Edit Sketch Tools Help
orgone_accum
MODULATION
OSC_ISR
UPDATE CONTROLS_C2
UPDATE
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Port
Programmer
Burn Bootloader
Teensy 3.1
Teensy 3.0
Teensy 2.0
Teensy++ 2.0
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet

Board: "Teensy 3.1"
USB Type: "Serial"
CPU Speed: "144 MHz (overclock)"
Keyboard Layout: "US English"

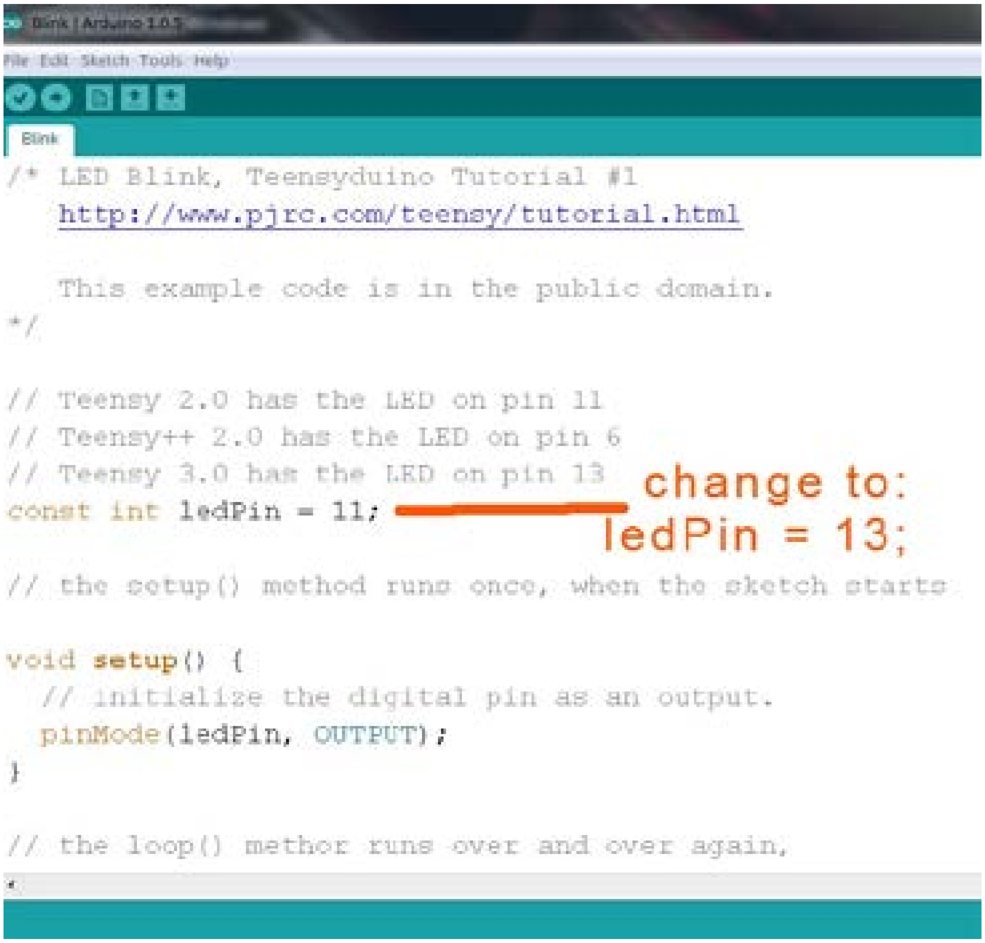
```
//Neutro
//Orgone
//these                                                                      the
//https:                                                                     he a

const in
  0,402,                                                                     ,52
  6786,7179,7571,7961,8351,8739,                                             .038
  12909,13278,13645,14009,14372,1                                            ,16
  18537,18867,19194,19519,19840,2                                            ,21
  23452,23731,24006,24278,24546,2                                            ,26
  27466,27683,27896,28105,28309,2                                            ,29
  30424,30571,30713,30851,30984,3                                            ,31
  32213,32284,32350,32412,32468,3                                            ,32
  32764,32757,32744,32727,32705,3                                            ,32
```

Open the “blink” sketch in the teensy examples



You will need to do a small edit for the blink program to work



```
/* LED Blink, Teensyduino Tutorial #1
   http://www.pjrc.com/teensy/tutorial.html

   This example code is in the public domain.
*/

// Teensy 2.0 has the LED on pin 11
// Teensy++ 2.0 has the LED on pin 6
// Teensy 3.0 has the LED on pin 13
const int ledPin = 11;

// the setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output.
  pinMode(ledPin, OUTPUT);
}

// the loop() methor runs over and over again,
```

change to:
ledPin = 13;

PROGRAMMING THE TEENSY

Now press the button that looks like a checkmark, you will see a bunch of text on the bottom of the window while the code is compiled. Then, if the teensy is new to your computer,The software may ask you to press the button on the Teensy, you should only have to do this the first time. Press it, then it should load. The LED on the teensy should blink. Your Teensy works!

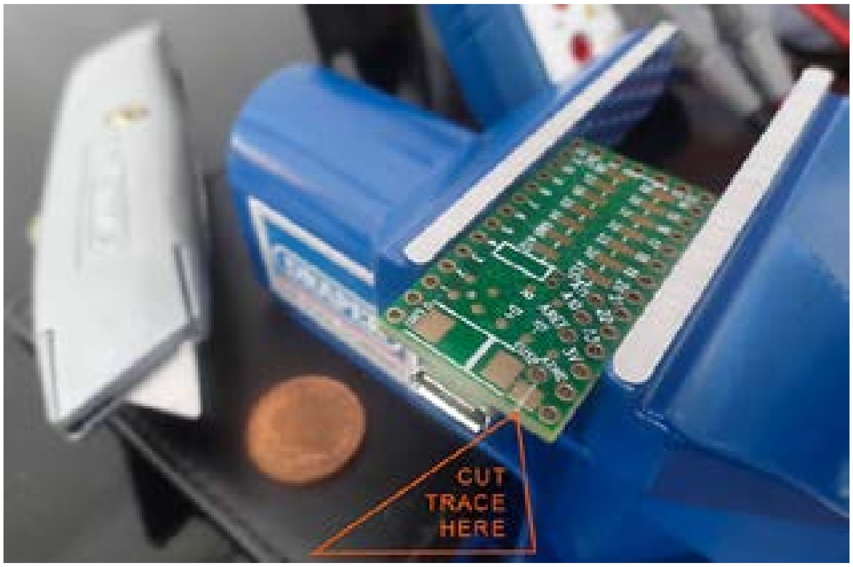
Note: on windows 7 arduyino IDE will sometimes display a message asking you to press the button on the teensy, but if you wait a little longer, it uploads anyway.

Now load the firmware of your choosing. (If you are building a 2x Teensy configuration be sure to find a way to mark which Teensy is Left and which is Right - I use a piece of tape)

LEAVE NO TRACE

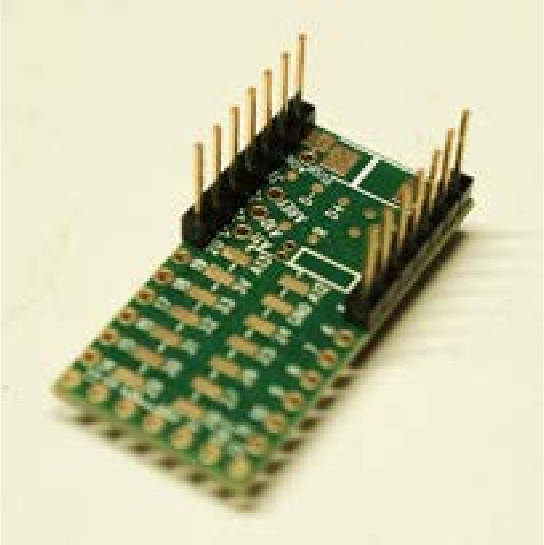
this joke won three honorable mentions for the 2017 the International Guild of Dad Joke writers

You should have programmed the Teensy by now with at least the blink program to make sure you know how to operate the software, and that the Teensy is working. Now you need to separate the 5V VIN from the power supplied by USB. Make the cut between the two rectangular pads. If you have a multimeter, check that the two pads no longer make an electrical connection.

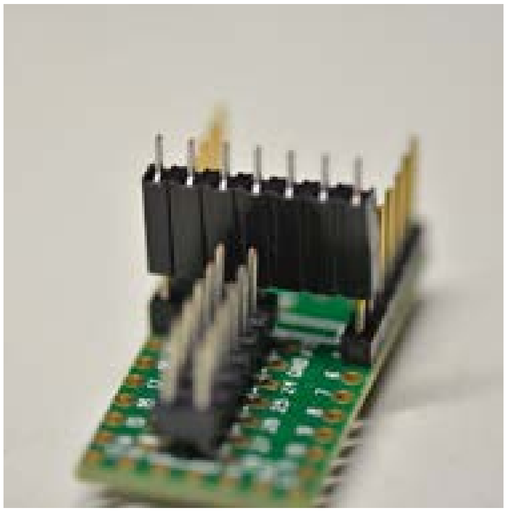


Cutting this trace still allows the Teensy to be programmed via USB while it is connected to the synthesizer power, however, the Teensy will no longer work on USB power alone.
If the trace is not cut, the 5v regulator and the computers USB power will be connected together, which could break something.

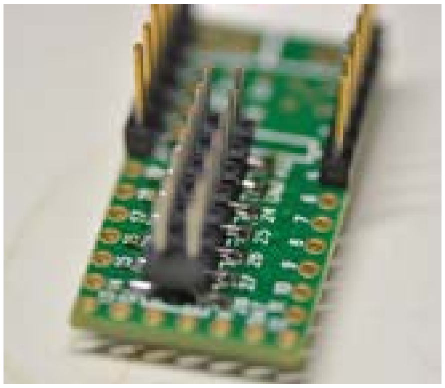
BUILDING THE TEENSY HEADERS



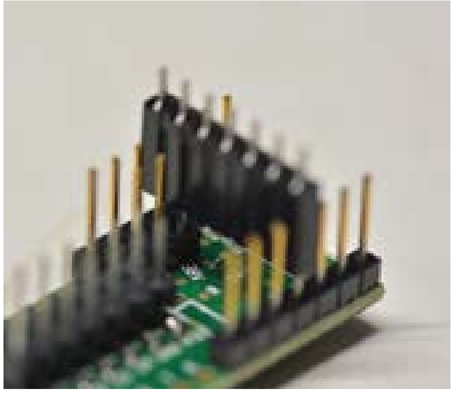
The Teensy needs to have pins along both long sides, along the short side, a group of three near one edge, and a special set of 14 pins in the middle which use a SMD connector. Lining these all up correctly can be tricky, so proceed as follows.
Warning! be careful not to get any solder on any of the pins long ends, even the tiniest blob of solder makes it impossible to insert the pins in to the headers, and it is hard to clean off without a vacuum desoldering tool.
Instead of the 14 rows you might expect for the two long sides, break off 2 rows of seven from the breakaway pin strip. These will cover half of each long side, nearest the USB connector. This makes getting at the SMD pins easier, while also giving you an alignment point.
Get the 7 pin female board header, we are going to use that as an alignment tool, we will call it “alignment tool” for now.
Solder in 2 of the 7 pin rows, at the USB end of the Teensy.
Just solder one pin for now. Make sure they are vertical and flush to the teensy (you can use “alignment tool”)
Now place the alignment tool over the rows you soldered, and also the SMD header,



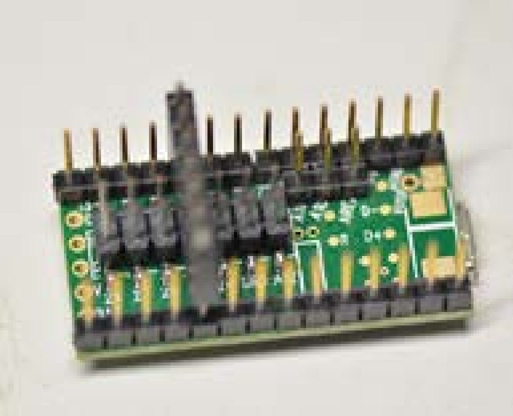
BUILDING THE TEENSY HEADERS CONT.



Solder a couple of the SMD pins, and then remove the "alignment tool"
Now you can solder the rest of the SMD pins fairly easily.

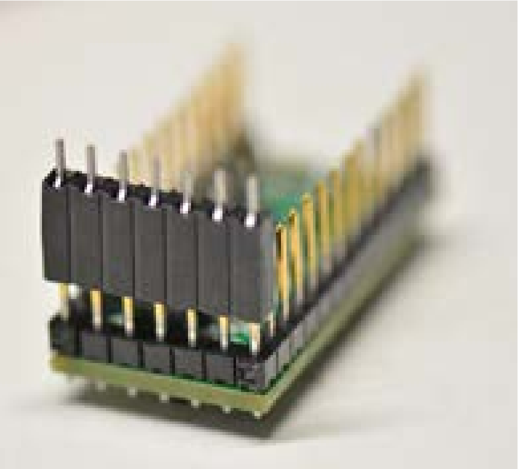


Now break off 3 pins from the breakaway header and use the alignment tool to hold them in the 3 holes marked AREF A10, and A11 on the Teensy.
And solder one pin, to hold them in place.

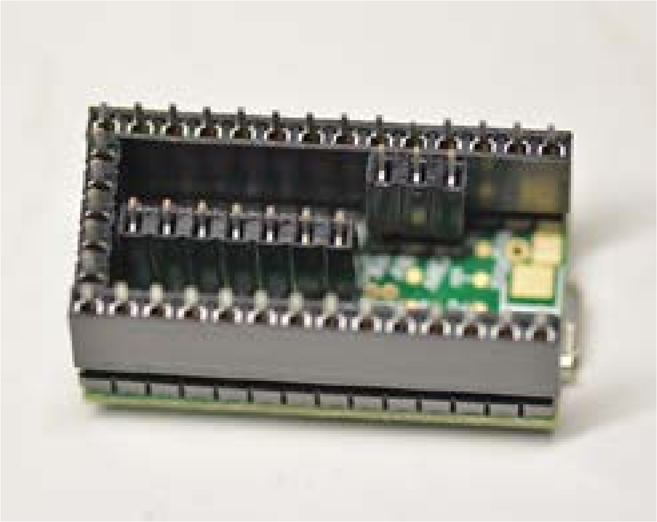


Now you can break off 2 more strips of 7 pins and place them along the remaining side holes. The alignment tool can be used again
You may need to file or sand the ends of the strips you cut, if they interfere with the strips already in place.
Just solder 1 pin on each for now.

BUILDING THE TEENSY HEADERS CONT.



Now utilize the alignment tool yet again for the remaining 5 pins at the short end of the Teensy. You may now retire the alignment tool and call it a board header again.
Again solder only 1 pin.

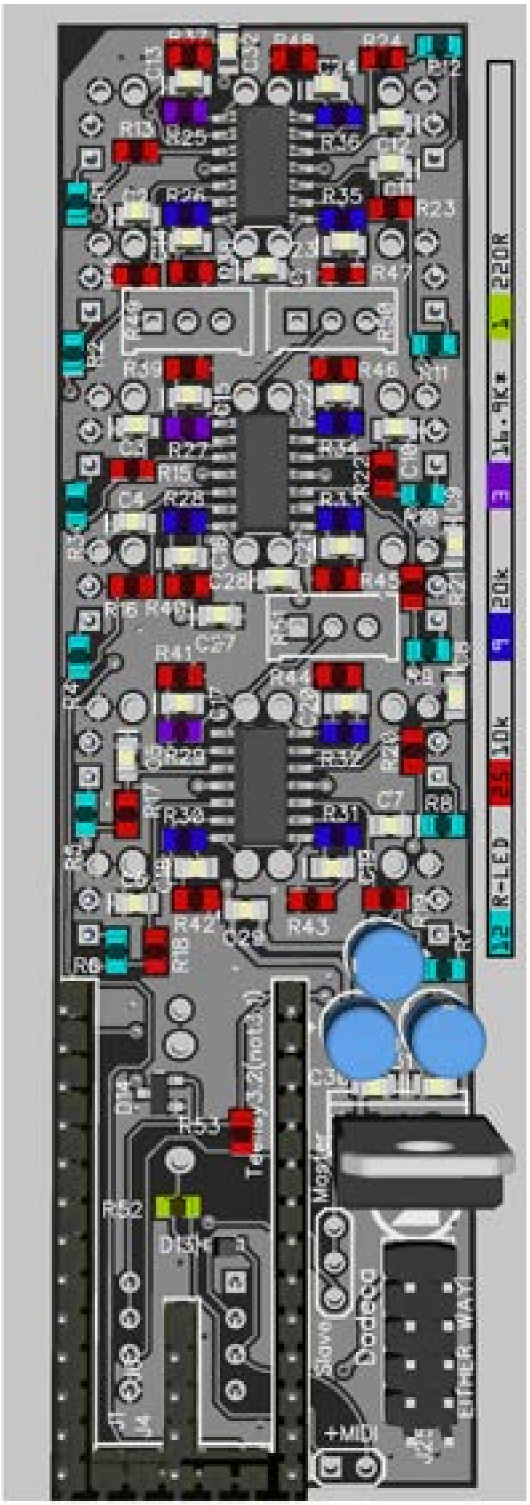


Note this picture is not of the dodeca board. Being lazy, I just copied this section of the orgone accumulator build manual

You can put all the headers on their respective pins, to align them all. You will not be able to push them all the way down though; this is fine.
Press the 4 pin one on the middle row of the SMD pins on the 4 pins farthest from the USB port first and use that as a guide for how far to press the others.
Note that only the “middle” row is used on the SMD header. It is the row that goes down the middle of the Teensy. The other row is just to add stability to the SMD contacts if you have to remove the Teensy.
It can be a bit tricky to get all the header pins in to the Orgone Accumulator board, and you may need to move the pins/header a bit to make if work nicely.
This is why you only soldered one pin on each (Idea thanks to hexinverter).
Solder all the board headers to the PCB, then all the Teensy pins to the Teensy. Be careful not to get any solder on any Teensy components.

At this point, remove the.
Teensy. You should be able to pry it out quite easily by going around it bit by bit with a screwdriver.
Don’t rush it and bend pins.`

POPULATE THE BOARD: RESISTORS



Start by soldering the 3 TL074 ICs taking note of the polarity. Then place and solder the SMD resistors noticing the repeating similarity in layout. I work in U shaped pattern from the left going down because the layout of the components is similar

🟥 25x 10k

🟦 9x 20k

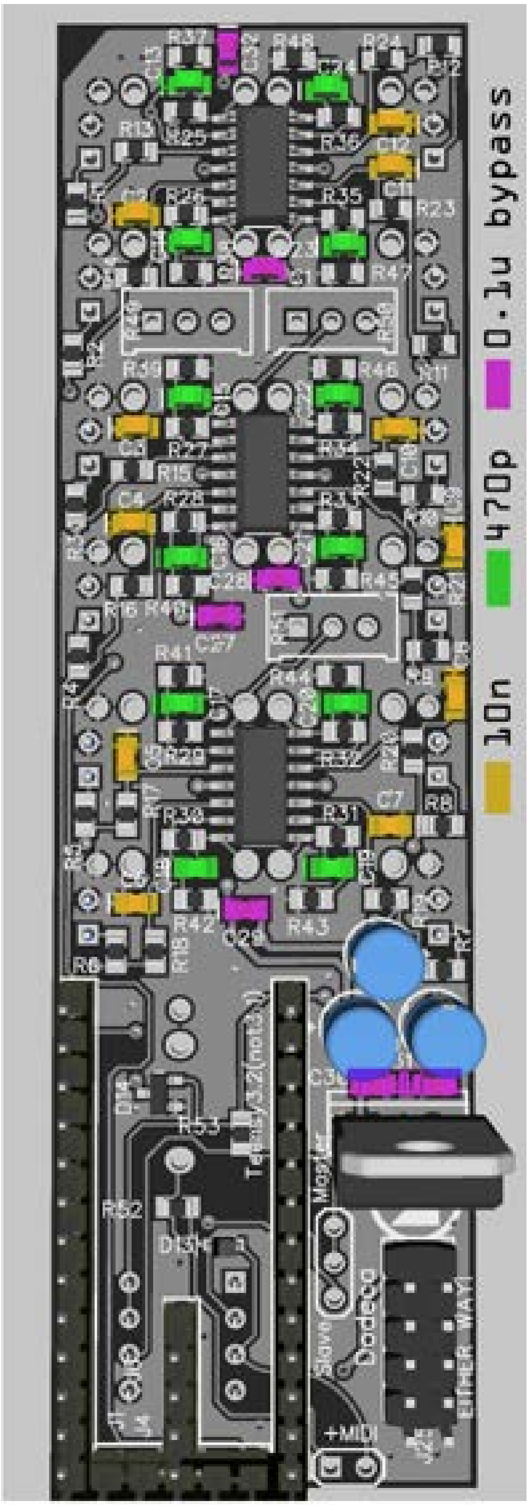
🟩 1x 220R

🟪 3x 16.9k

SUBSTITUTING OP-AMPS

If you decide to use different op amps for some reason, be aware that this circuit relies on the TL074 internal 128 ohm resistor as the output resistor. It means the output can drive more modules before the voltage starts to droop, but if you use a "better" op amp that is not short circuit protected, a short could kill it. There is probably no point anyways!

POPULATE THE BOARD: CAPACITORS



Now lets place all the SMD capacitors.

🟪 7x .01 uf Bypass

🟩 12x 470 pf

🟨 11x 10nf

POPULATE THE BOARD: ERRR'THANG ELSE

Add these components (Do not jump ahead and add the IC socket just yet)
3x electrolytic caps, Voltage regulators and the Diodes.

TRIMMERS

If you decided on trimmers, then solder them now, where there are no trimmers, then you need to solder a jumper (a bit of component lead wire you just cut off the capacitors is fine)
the trimmers are for these outputs:
R49 > out2 (DAC)
R50 > out6
R51> out 10
The jumpers go between the square pad and the middle one.

TEENSY BOARD SOCKETS *IMPORTANT*

If you don't follow these instructions you are going to spend a few minutes cursing the board and being angry at us, even though you are the one who didn't read the instructions.

In order to fit everything on a 6hp board we had to be creative with component placement. Because of this you need to build up the teensy headers in a specific order so that the IC socket doesnt prevent you from soldering the pins to the middle row of the teensy header.

1) Solder in these three small headers
2) Flip the board over and add the IC socket soldering all 8 pins.
3) Solder 1 pin of the remaining header sockets and place your built up teensy into the socket to acheive perfect alignments. solder the rest of the teensy headers using teensy as an alignment tool.

COMPONENTS & LEDS

Solder the thonkiconn jacks flush to the board using the panel to align them Dont solder the stereo jack until after the 12 thonkiconn ones.
After the thonkiconn jacks are soldered cut the tabs from the stereo jack, leaving as much of the pin as possible. Place the jack but do not solder yet. Attach the panel to a couple of the jacks with the nuts
now tighten the nut on the stereo jack which will pull it forward and align it on the panel with the taller thonkiconn jacks.
Now you can solder the stereo jack pins..
the LEDs can be a bit tricky. There might be a better way, but what I did is this: I put them in their places on the board, than attached the panel with a couple of jack nuts, then use a thin strip of tape over the holes to stop the LEDs sliding all the way out and turned it over so it was face down on the desk. Please note LED polarity. The longer pin on the LED goes through the square pad.

POWER BUS TESTS

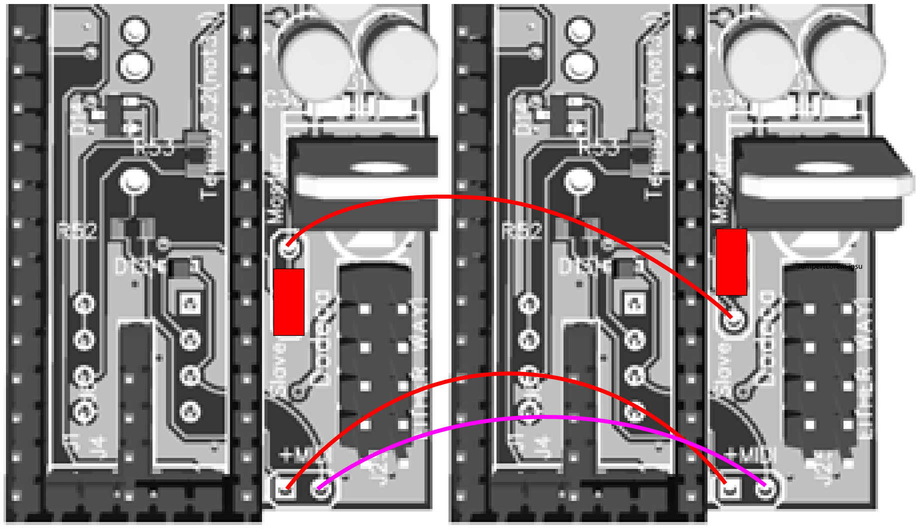
Before you put the teensy back in, its a good idea to do some basic power bus tests,
the tab of the 7805 regulator is at ground, and a handy place to measure from.
check the continuity of the "+" pin of the rectifier to pin 4 of each TL074
check the continuity of the "-" pin of the rectifier to pin 11 of each TL074
make sure there is no continuity between + and ground, - and ground. (if you use a meter with continuity beeper, it may beep briefly as the capacitors charge)
power up and check that there is + approximately +11.4v on the + side of the rectifier output. -11.4v on the negative side. (careful you dont short those 2 pins while measuring)
5v on the output of the regulator, (bottom pin when viewed as the resistor and capacitor diagrams above)

If everything is good, then install the teensy, and you should be ready to go!
Most of the dodeca programs on github have a LED test sequence at startup. Note, the LED on output 2 appears dimmer than the others at low level, because it uses the DC output of the teensy DAC instead of PWM. Its not a build fault.

MASTER / SLAVE PIN CONFIGURATIONS

DBLDECA & OTHER 2X BOARD CONFIGURATIONS:

One of the features I like most about the dodceca is the ability to daisy chain the board to expand the capabilities to fit your needs. Here is the configuration required for any 2 daisy chained boards. Notice our naming convention for the firmware (Left & Right) is based on the back view of the module, not the panel view.

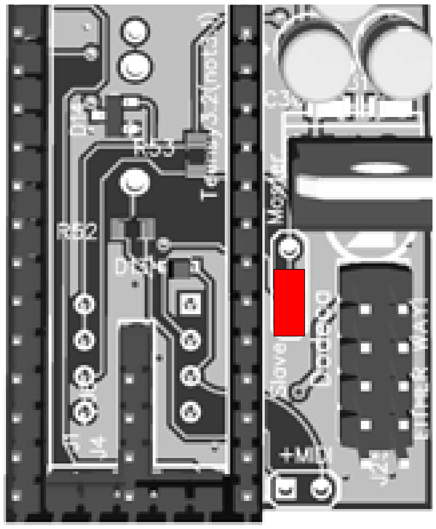


RED BLOCKS
ARE JUMPERS

DBLDECA LEFT
FIRMWARE
(MASTER)

DBLDECA RIGHT
FIRMWARE
(SLAVE)

DODECA & OTHER SINGLE BOARD CONFIGURATIONS:

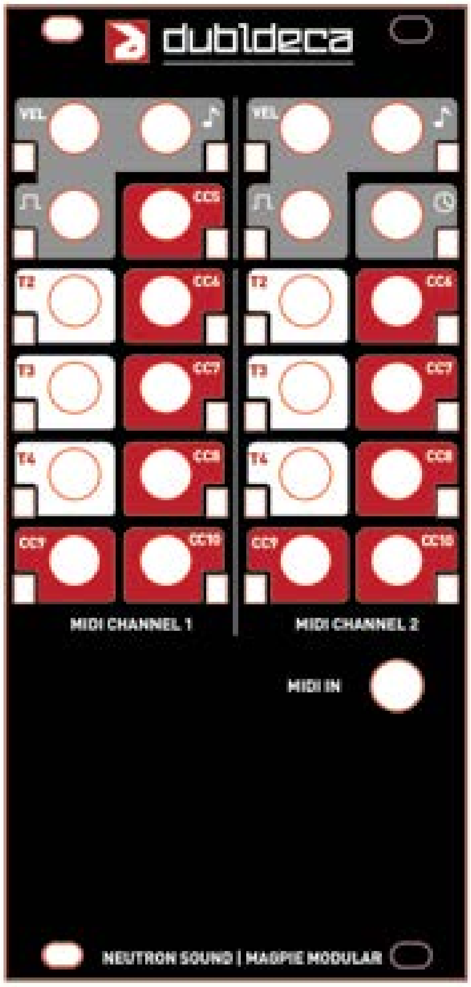


DBLDECA FIRMWARE: OVERVIEW

The DBLDECCA firmware requires two dodecas and is specifically created to use in conjunction with the Elektron Octatrack midi sequencer. The layout of the module mirrors the MIDI track buttons on the Octatrack.

The default configuration of the module features the following
2 full midi voices (Note, Gate velocity)
6 gate outs
1 Midi Clock out

TRACK 2 GATE OUT
CC6 - CV OUT
TRACK 3 GATE OUT
CC6 - CV OUT
TRACK 2 GATE OUT
CC6 - CV OUT



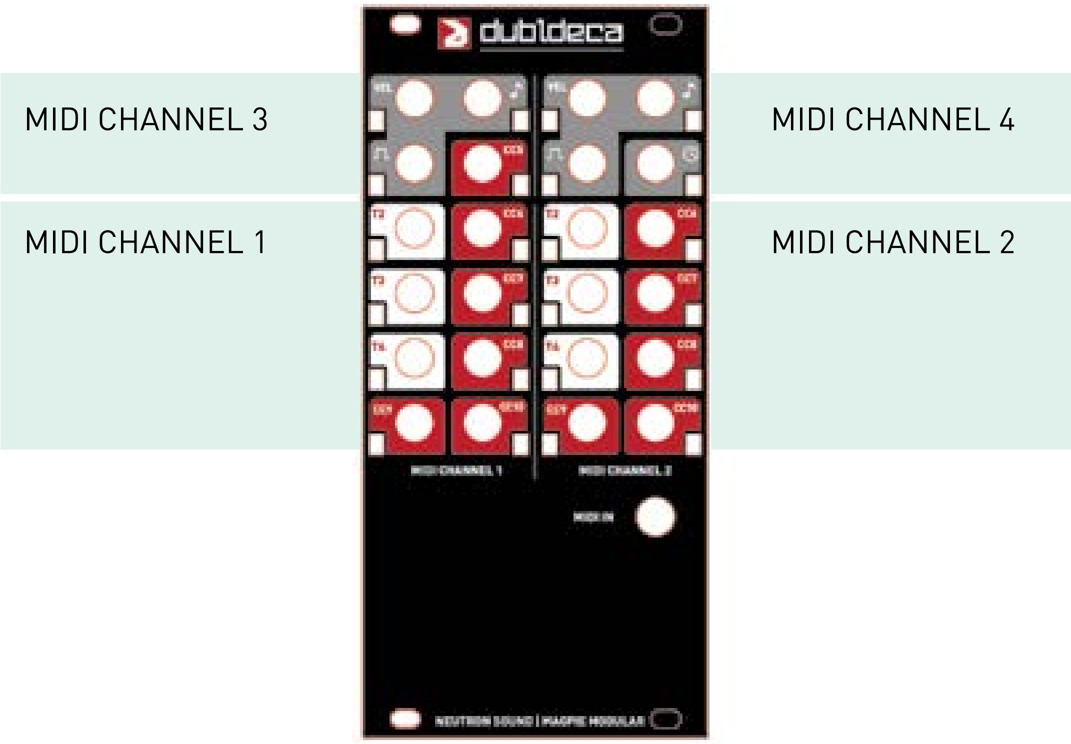
DBLDECA FIRMWARE : SETTING UP THE OCTATRACK

We have done our best to follow the Elektron naming convention in the below setup guide. Keep in mind that some of these settings will not take hold unless you press enter to save them/ or will not be editable until you press function and turn the knob. These will be mentioned specifically below.

Create a new project, name it and save it. You will need to adjiust your global midi settings
Press **[FUNCTION] + [PROJECT]**
Select **SYNC**
Set Transport to **SEND + RECEIVE**
Set Clock to **SEND + RECEIVE**

Notes: Transport SEND is what start and stops the DBLDECA midi clock when you press the start or stop button, Receive is only necessary if you have the Octatrack slaved to another unit. The same goes for the MIDI CLOCK SEND & RECEIVE

MIDI CHANNEL SETUP
Select MIDI Part to enter into the midi sequencer
Select Track 1, quickly double tap [Note] to set the midi channel to 3, press enter
Select Track 5 quickly double tap [Note] to set the midi channel to 4, press enter
Select Track 6,7,8 quickly double tap [Note] to set the midi channel to 2, press enter
Tracks 2,3,4 will be set to midi channel 1 by default.



NOTE TO GATE SETUP

For tracks T2 & T6 Select Note and set the note to C - it doesnt matter which note C0-C8 but these tracks will note output gates if its not set to C. Note2,note3,note4 should all be set to zero and stay there.

For tracks T3 & T7 Select Note and set the note to D - it doesnt matter which note C0-C8 but these tracks will note output gates if its not set to D. Note2,note3,note4 should all be set to zero and stay there.

For tracks T4 & T8 Select Note and set the note to E - it doesnt matter which note C0-C8 but these tracks will note output gates if its not set to E. Note2,note3,note4 should all be set to zero and stay there.

CC SETUP:

In order to enable a track to send out CC values you need to
select the track
Goto the [Control 2] page, press function and press the knob of the CC you want to make available, This will enable it. The value of this knob will now send a 0-5v unipolar scaled voltage out the output. Give it a wiggle and you should see the LED light up.

Due to how the octatrack works you need to make sure only one track addresses each CC. For instance if Track 3 has cc6 enabled and set to 0 and Track 4 also has cc6 enabled, only the first assigned track will send out values.

Additional setup: I like to allow for each track to be a different length - you can do this by making sure the record button isnt lit or flashing, and pressing [function] and [Bank] you can set this to advanced mode and you will then be able to quickly set the length and time signature of each individual track.

Save this set up and you are good to go.

ABOUT THE GATE OUTPUTS

About the Gate outs (Tracks 2,3,4 on the left and 6,7,8 on the right)
These respond to specific Notes i.e C1, C2, C3, C4, C5 - The note length you set in the octatrack will change the length of the output gate including syncing to note divisions of your tempo as well as infinite hold. This is useful for creating long held gates for modules such as Mutable Instruments Elements, or more advanced uses using logic modules.

From the Octatrack you can specify the velocity 0-127 of the gate and this will translate to the amplitude of the output gate from 0-5v. This can be useful for pinging Low pass gates. This of course can be recorded and automated using PLocks

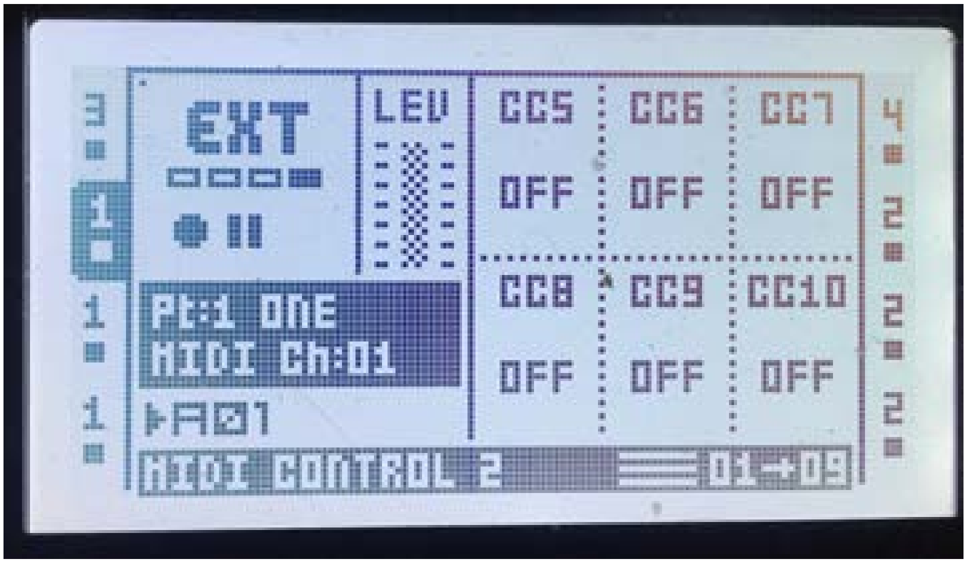
About the MIDI CC to CV output.
This is programmed to work out of the box with default MIDI CC assignments in the default Octatrack program, specifically page 2 however you will need to do a basic set up before the Dbldecca will work properly.

These outputs are programmed to send unipolar CV (from 0-5v) corresponding to specific MIDI CCs. You can set this value via plocks on sequencer steps, by LFO or by both.

Resolution: As anyone who is familiar with it MIDI is a slow slow slow protocol. the V1 implementation says it can send 3,125 bytes per second, or just over 3kb a second. Elektron has released a faster implementation called Turbo Midi which can get speeds around 10x faster but unfortunately the dodeca would need to be a USB host in order to implement that. For the most part i have not found the speed limitation affect the output but if you output 2-3 continuous data sources (ie LFOs assigned to CCs) you may notice it.

ABOUT THE MIDI CC TO CV OUTPUTS

This is programmed to work out of the box with default MIDI CC assignments in the default Octatrack program, however you will need to do a basic set up before the Dbldecca will work properly. If you make sure you are in MIDI mode and select **[TRACK 2]** and then you should see something similar to the image below. on the left and right You will notice the number of the MIDI Channel each track is listening on. These should match what is shown.



EXT

LEV

CC5 OFF | CC6 OFF | CC7 OFF

CC8 OFF | CC9 OFF | CC10 OFF

Pt:1 ONE
MIDI Ch:01

A01

MIDI CONTROL 2 01→09

These outputs are programmed to send unipolar CV (from 0-5v) corresponding to specific MIDI CCs. You can set this value via plocks on sequencer steps, by LFO or by both.

Resolution: As anyone who is familiar with it MIDI is a slow slow slow protocol. the V1 implementation says it can send 3,125 bytes per second, or just over 3kb a second. Elektron has released a faster implementation called Turbo Midi which can get speeds around 10x faster but unfortunately the dodeca would need to be a USB host in order to implement that. For the most part i have not found the speed limitation affect the output but if you output 2-3 continuous data sources (ie LFOs assigned to CCs) you may notice it.

DODECA FIRMWARE

The DBLDECCA firmware requires two dodecas and is specifically created to use in conjunction with the Elektron Octatrack midi sequencer. The layout of the module mirrors the audio/midi track buttons on the octatrack.

The default configuration of the module features the following
2 full midi voices (Note, Gate velocity)
6 gate outs
1 Midi Clock out

DBLDECA PATCH CHEAT SHEET