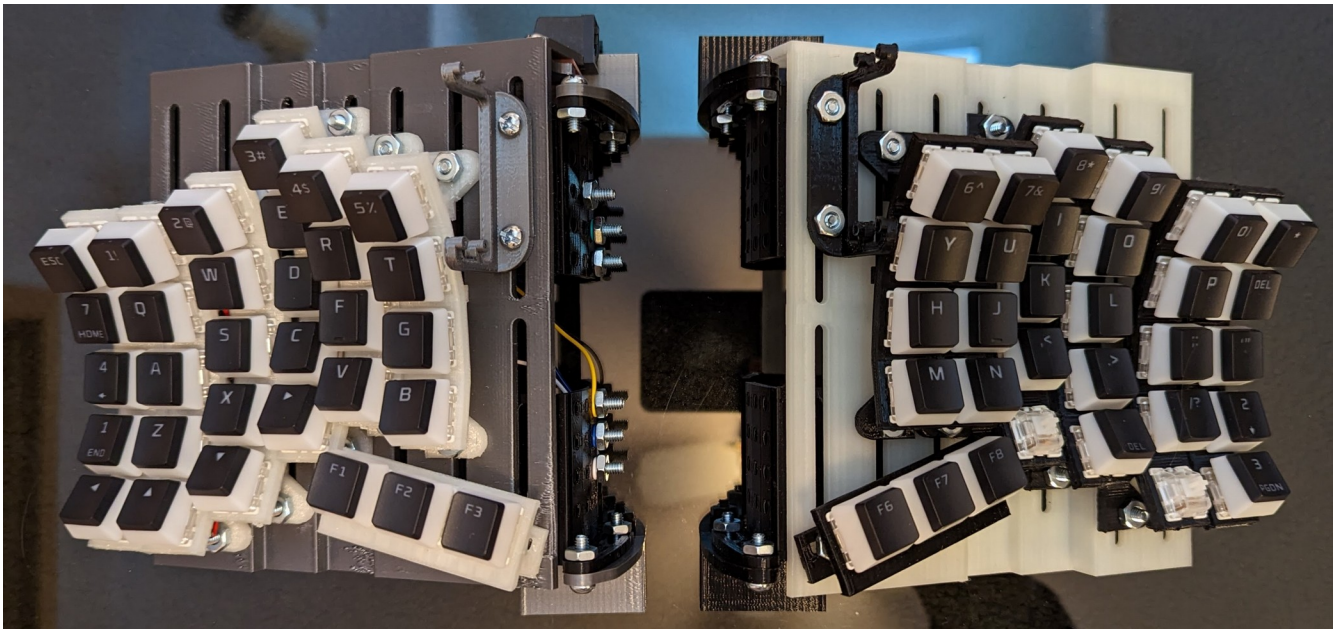
Table of Contents

Dactyl Chimera Handbook

Version 1, 2023-03-25



Two

Chapter 1: Is Dactyl Chimera right for you?

Dactyl Chimera is Split, not Ergo.

You've probably noticed that the Dactyl Chimera keyboard is split down the middle. Even weirder, it arranges its keys into columns, it removes the arrow keys, and it replaces the spacebar with "thumb clusters". To use this keyboard, you NEED to reinvent how you type. There's a theory behind the madness, as this video shows. https://youtu.be/AKGXZ1ReU54

Software tricks can be applied to any keyboard, so why this shape? Split keyboards try to reflect the shape of the hand, and often they are called "ergonomic" or "ergo", however **the Dactyl Chimera is NOT a medical tool and is NOT a treatment for RSI.** I have never studied ergonomics or anatomy. I started this project because I dislike the sound of large spacebars and because 3D printing is cool.

One more thing: this keyboard is designed for touch typing. If you have to look down at your keyboard to type, this keyboard will be terrible for you. If you want to learn to type faster, check out the section “The Qwerty arrangement of letters” in Chapter 3.

Dactyl Chimera is a DIY project.

Following this guide requires online purchasing, 3D printing, soldering, programming, and even screwing things together! Fortunately, the purpose of this handbook is to teach you these skills as cheaply and easily as possible. If you like DIY but want something flat and traditional, consider building a SiCK-68: https://www.thingiverse.com/thing:3478494

Dactyl Chimera is a Test Bench, not a retail product.

When building this keyboard, you will reshape, re-print, and re-assemble the columns until they perfectly match your fingers. You can then accessorize your keyboard by attaching scroll wheels, joysticks, and OLED displays. If that sounds like a hassle to you, there are professionally assembled Dactyls you can buy from many online retailers. Search the web for “Dactyl keyboard”, not “Dactyl Chimera”.

Dactyl Chimera is not a portable keyboard.

Most Dactyl keyboards feature a hard outer shell to protect the electronics. Messing with the internals is the whole point of the Dactyl Chimera, so its "case" is a skeletal frame with just enough plastic to get the keyswitches into position. Unfortunately, throwing this keyboard in a backpack would probably break it. For a travel-friendly, Bluetooth-ready, large yet portable keyboard, I recommend the Lily58: https://github.com/kata0510/Lily58

Chapter 2: Join the Cult

I'm ~~still new to~~ ***bad at*** GitHub, so I need some help fleshing out this part of the build guide. The easiest place to download the keyboard files is from the Releases link, https://github.com/WolfIcefang/dactyl-chimera-keyboard/releases but I can't give advice on the best way to help development of the project.

Get help!

Need help or want advice? You should join the Dactyl Chimera Center in [Matrix]. https://matrix.to/#/!mArixoOlqsCQNWsaFc:matrix.org?via=matrix.org This Space is currently hosted on the matrix.org Homeserver. As of 2022-01-06, "You must be at least 16 years old to use this Service." (according to terms of service https://element.io/user-terms-of-service.)

You can also reach out through the Issues and Pull Request features in GitHub: https://github.com/WolfIcefang/dactyl-chimera-keyboard

GitHub Fork Management

GitHub's user interface for forks is terrible. If you're wondering just how bad it can get, look at the original Dactyl and try to find a fork for the Arduino Pro Micro. (hint: it's impossible.) https://github.com/adereth/dactyl-keyboard/network/members

Since the Dactyl Chimera is all about making personal variations and accessories, our fork page could get even worse. With that in mind, here are some suggestions:

1. Please do not fork unless you intend to make changes. The change can be as small as a single adjustment to one column; that's fine. However, if you won't change *anything*, it's better to Star or Watch the main page.
2. Please give your fork a unique, descriptive name. If you are making a joystick accessory, add "joystick [part id number] mod" to your fork's name. If you are personalizing the columns for your fingers, add "personal columns" to your repo name.

If there are too many forks, https://github.com/techgaun/active-forks can help you find the best ones.

Licensing and Free use:

The Dactyl Chimera is not currently covered by a license, but that's just because I haven't gotten around to it yet. **Informally:** you are free to use the Dactyl Chimera for any purpose. If you share the Dactyl Chimera, please do so with a link to the original GitHub project. If you make changes to the "core parts" such as the arches, base plate, or conductor, and you want

to make them public, you must share your FreeCAD files, not just the .3mf or .amf files. If you create your own original attachments, such as alternative microcontroller holders, joystick mounts, etc., then you can license them however you'd like.

Chapter 3: Becoming a QMK sorcerer

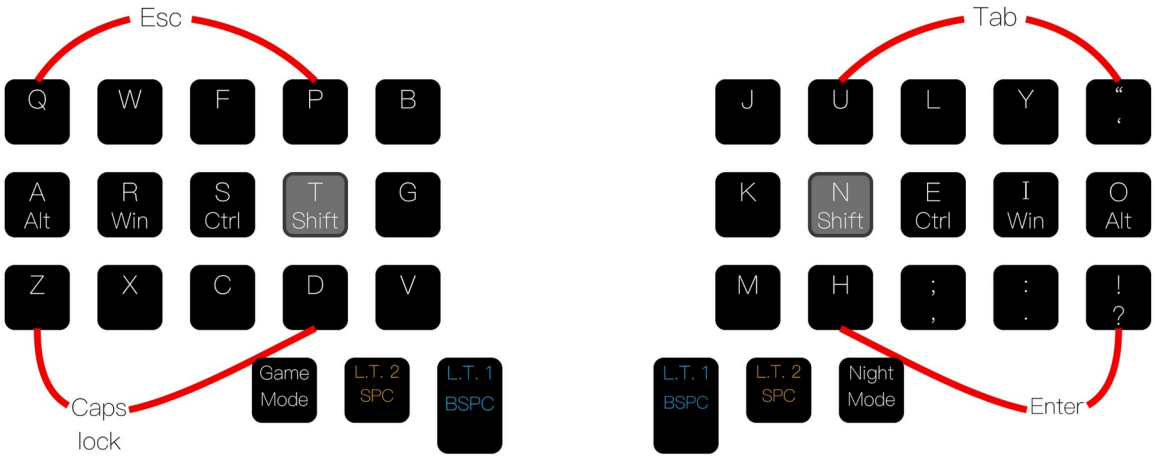
If you're looking for the firmware flashing instructions, check Chapter 7: The Brains.

You were warned in the introduction that you would need to reinvent the way you type. Perhaps you already know where the letters and numbers are going, but what about the arrow keys? F1-F12? The enter key? The answer is: wherever it fits you best. If the Dactyl Chimera matches your personal hand shape, the best layout of keys should likewise be based on your particular muscle memory.

Start thinking about your layout today. It will give you something to do while you wait for your online orders to arrive.

Layers are the keymap's basic building blocks.

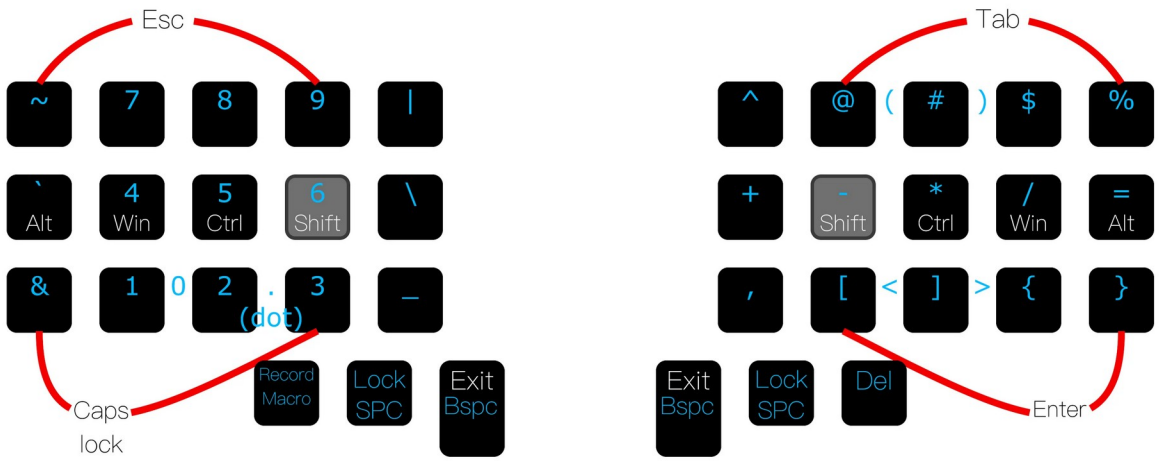
When I type on my keyboard, I don't think of having just one board in front of me. I imagine three keyboards stacked atop one another. I have a few dedicated keys, aptly called layer change keys, that let me instantly jump from one layer to another.



Layer 0

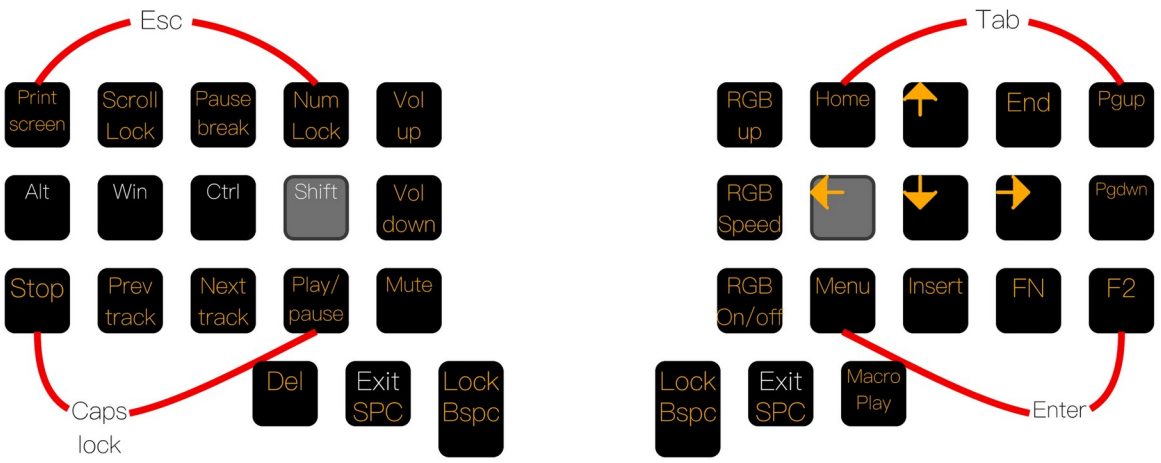
Layer 0 is my ground floor. This is where I keep the most common and important keyboard keys: letters, enter/backspace, shift, and anything else I use all the time or need occasional fast access to. This layer uses combos, where I hit two keys simultaneously to trigger a different action. I also use mod-tap, where a long press on a home row key can trigger Shift, Ctrl, Gui, or Alt.

Oh, right! You might not remember the phrase “home row”. The home row is the exact center of the keyboard, and your fingers are expected to return there whenever they aren’t pressing a key. On a Qwerty keyboard, the home row keys are ASDF and JKL;.



Layer 1

Layer 1 has my number pad, punctuation symbols, and math operators. QMK supports many ways to switch between layers, including holding down a key (similar to using shift), tapping a toggle key (which works like caps lock), or even writing custom C code for advanced functionality.

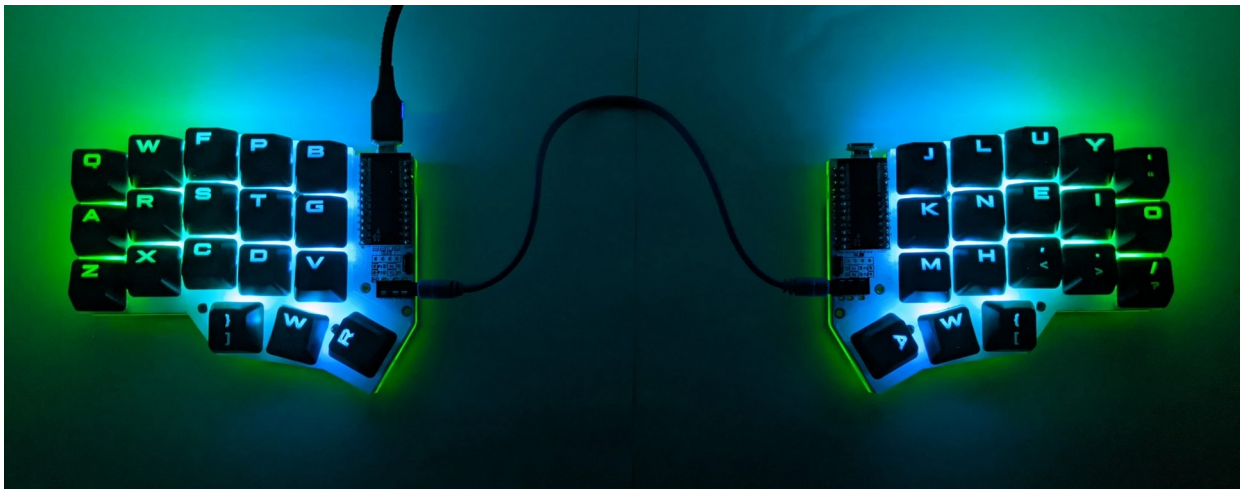


Layer 2

Layer 2 has the arrow keys, volume controls, and other navigation commands. I can go directly between layer 0 and layer 2.

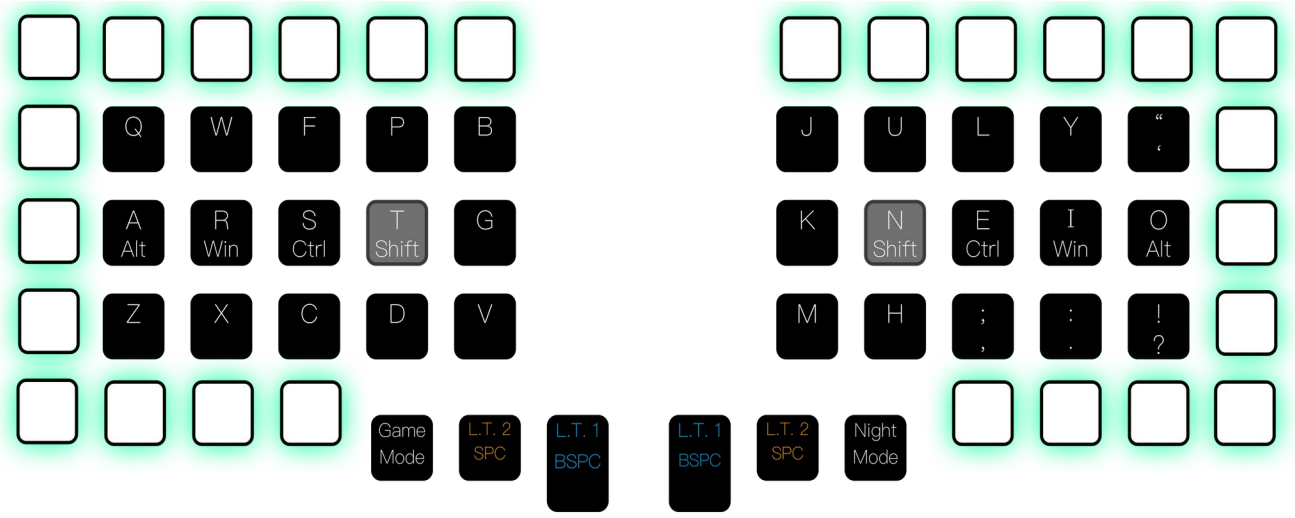
I have additional layers for video games and the F1-F12 keys, but those don't work very well or they're just plain boring.

The layout I've just described isn't for my Dactyl Chimera. It's my travel keyboard, the 36 key Atomic Green Corne:



I believe my Corne is an excellent example of "QMK magic": combos, coding, tapping, and holding. Seeing this super-small keyboard should also give some context for the next section:

The Dactyl Chimera is (relatively) massive.



Unlike the Corne, the **62 key** Dactyl Chimera doesn't need too much QMK magic. Shift, Ctrl, the number row, backspace, escape, and tab can reclaim somewhat traditional positions here. Only the enter key, some symbols, and the arrow keys would fall outside the perimeter.

ESC	1	2	3	4	5		6	7	8	9	0	DEL
TAB	Q	W	E	R	T		Y	U	I	O	P	BKSPC
ENTER	**A**	**S**	**D**	**F**	G		H	**J**	**K**	**L**	**;**	CTRL '
SHIFT	Z	X	C	V	B		M	N	,	.	**up**	SHIFT /
CTRL	GUI	ALT	Macro 1	Layer 1	**Space**	Layer 2	Layer 2	**Space**	Layer 1	Macro 2	**left**	**down**	**right**

It is possible to fit "everything" on layer 0, but in my opinion, playing with QMK layers is half the fun of building a custom keyboard like this.

If you liked my Corne layout or were enamored by the tiny keyboards explanation video, **62 keys** might seem intimidating. Worry not: as a prototyping tool, Dactyl Chimera emulates split keyboards of any size. To gain the benefits of a 40%, simply disable the outer perimeter of keys! You don't even need to put switches in all the holes. However, you may find that the Dactyl's arched columns bring keys closer to your fingertips, and that larger keyboards have real, everyday benefits for you to take advantage of. Extra keys can:

- Quickly turn a strange keyboard shortcut into a layer 0 macro. ("Why does Microsoft Excel force me to use F2 to edit cells?")
- Simplify one-handed keymap design. (You can keep one hand on your keyboard, and the other on your mouse/drawing tablet/trackball/trackpad.)
- Reduce the number of layer changes needed for complex sequences like passwords. (Remember: every layer change requires an extra key press.)
- Perimeter keys are easier to hit than home row keys when your hands aren't settled into a typing rhythm. (for actions like muting your computer while answering a phone call.)
- Reduce cognitive load. Cognitive load is the thought you waste trying to keep track of which layers are on and off. Dactyl Chimera is not meant to eliminate cognitive load.

If you'd still prefer a 3D keyboard with fewer keys, you can check out the squeezebox keyboard. It's possibly the closest to the Dactyl Chimera in design and features.
https://www.reddit.com/r/ErgoMechKeyboards/comments/rvuvk2/squeezebox_keyboard_v2112/

Guides to learn QMK's basics:

Pick one of these to read and learn:

- Thomas Baart's Introduction to QMK Features https://blog.splitkb.com/introduction-to-qmk-features
- Currently I only know about one guide...

Note: QMK's official documentation is a great resource https://docs.qmk.fm/ but I wouldn't call it "the basics". You'll need to read it in Chapter 7: The brains.

Articles about specific key placement:

- DreymaR's "Extend" for where to put the arrow keys and other navigation tools: https://dreymar.colemak.org/layers-extend.html
- Looking for where to put ctrl, shift, gui, and alt? Try one of these:
    - Precondition's Guide to Home Row Mods https://precondition.github.io/home-row-mods
    - Why you should BAN Key Chords http://xahlee.info/kbd/banish_key_chords.html
- You can join the Keyboard Layouts subreddit for some really mind-bending ideas. https://www.reddit.com/r/KeyboardLayouts/

Tips for designing your layout:

- Use **LibreOffice Calc** or **Inkscape** to brainstorm your layout. Calc spreadsheets keep your work arranged nicely, but drawing in Inkscape lets you break free of the grid to easily represent combos or tap dance sequences.
- Inside your Calc or Inkscape file, make a list of **all** 104 standard keyboard keys, plus volume keys, media keys, and any macros you want to include. Use Drag-and-drop or copy/paste to edit your brainstorming document; this way you don't create multiple copies of keys and you don't delete anything. Be sure to find a spot for *every single key*, even the ones you don't think you'll ever use.
- QMK Configurator is a good place to get your list of 104 keys. https://config.qmk.fm/ I don't use QMK Configurator for brainstorming because it doesn't autosave my work.
- Design each layer by placing the most common keys in the center, and important but less common keys near the edges.
- Beware of **Enter** and **Escape**! Pressing these keys can instantly erase text boxes or send incomplete messages. On a traditional keyboard, these keys are far from the home row and it's rare to hit them by accident. Don't make them *too* convenient on your own keyboard. **Tab** and **Alt** are also mildly dangerous.
- Never bridge your combos across both halves of the keyboard. There will come a time when you must move your mouse with one hand and type with the other. My Corne's first layout required a combo of left and right spacebar to hit "enter"; it was a disaster.
- Pay attention to "bigrams" and "trigrams", keys you often press in sequence. You never want to use the same finger/thumb multiple times in a row. For example, every sentence ends with the sequence [period] -> [space] -> [shift], so you shouldn't put spacebar and shift in the same thumb cluster.
- A spreadsheet of "starter layouts" is included in the GitHub's releases page. These layouts are provided to inspire you, not restrict you.
- You can also explore keymaps for many keyboards on https://keymapdb.com
- Finally, if you find a defect in your layout, fix it as soon as possible. Finding the perfect layout is a mixture of experimentation, logical patterns of symbols, muscle memory, and the PC software you use. It can take weeks or months to get everything into position.

The Qwerty arrangement of letters.

Qwerty is an objectively terrible way to arrange letters. There is a semicolon in the home row, 5/6 of the vowels are *not* in the home row, and J is one of the least-used letters in

the alphabet. With that being said, I don't recommend breaking the mold unless necessary. If you want to rearrange Qwerty, here are some tips:

If you're looking for raw speed, Stenography is the way to go. Steno has an extreme learning curve, but the ability to transcribe conversations in real time or even "write at the speed of thought" might just be worth it. The Dactyl Chimera is not designed for Stenography. https://www.openstenoproject.org/

[[The paragraph you are reading was supposed to be about improving your typing speed on Qwerty, but I don't actually know anything about improving typing speed. Please help! Alright moving on to typing comfort now:]]

If you were just looking to rearrange some letters, Colemak is my top suggestion. Colemak probably won't increase your typing speed, but it is comfortable to use. There are a couple of reasons I suggest Colemak above and beyond other layouts:

1. Colemak's step-by-step "Tarmak" learning system helps you transition from Qwerty without slowing your typing speed too much. It works by moving just a few keys in each "step"; you get as much time as you need to learn each small change.
2. Colemak has a few well-tested variants, or "mods". If you find some part of Colemak uncomfortable, there's probably already a mod to fix it. You won't have to learn another entirely new layout or invent your own mods. I use Colemak ***mod*** DH, for example.

Moving away from Qwerty probably won't impact the rest of your layout design, with one key exception: combos. On Qwerty, the keys "jk" make a perfect combo. They're adjacent, they're on the home row, and you'll never type a word containing those two letters next to each other. If you switch to Colemak, "jk" becomes "ne". NEar, NExt, NEw... you'll be triggering the combo far more oftEN than you'd expect. To ENsure your combos work on your layout, you can use Wolfram|Alpha to list words containing any two letters: https://www.wolframalpha.com/input?i=every+word+containing+ne Remember that the order of the letters matters. For example, you'd NEed to check both "en" and "ne".

Finally, I challenge someone to combine Asetniop with 52 macro keys. There is no reward.

Chapter 4: Anatomy of the Keyboard (parts list)

This is a brief overview of the parts and tools you'll need to make, buy, or assemble to create this keyboard.

Tools

Before you spend hundreds on bespoke machinery, search your local libraries, colleges, universities, and makerspaces for these tools:

- ☐ a 3D printer
- ☐ a soldering iron
- ☐ a pair of wire strippers
- ☐ a pair of needle-nose pliers
- ☐ a pair of flush cutters

To my surprise, every public 3d printer I've seen in my town is professionally maintained and either free or dirt cheap to operate. Even if you aren't so lucky, I kept the parts of this keyboard as bad-printer-proof as possible: short print times, no curving overhangs, and large flat bases. If you really must buy a 3D printer, the r/3Dprinting subreddit maintains a monthly purchase advice page. https://www.reddit.com/r/3Dprinting/collection/a02c1c57-579a-4ab4-858d-04df5276ff92 You'll need an "FDM printer", the type that takes a spool of plastic wire (filament) and squeezes it out of a nozzle (extruder).

Any soldering iron from the past century should get the job done for this keyboard. If you need new equipment, check Keebio's guide. https://docs.keeb.io/soldering-tools

For wire strippers, Project Farm made a review of many brands and styles. https://youtu.be/98DUiirnkIk You can technically use regular scissors or flush cutters to strip wire, but doing that is frustrating.

Needle nose pliers have also been tested on Project Farm. You'll be gripping and pulling, not cutting. https://www.youtube.com/watch?v=vO3UX4oEnZI

Flush cutter recommendations can be found on Keebio's website (same site as the soldering iron): https://docs.keeb.io/soldering-tools

Materials

If you're lucky you'll find bins of raw materials at your local makerspace, but you should expect to buy this stuff yourself. Like flour, sugar, and eggs, these materials are the staple ingredients of most DIY projects. You shouldn't worry about buying in bulk or having extra.

3D Printer filament:

- In 2023, PLA is king, and it'll be more than adequate for the Dactyl Chimera. It's cheap, it has low temperature requirements, and you'll be hard pressed to find a printer that can't print it.
- PETG is great for mechanical parts like gears and support beams. It's stronger and tougher that PLA, it's resistant to UV light and water, and just like PLA it'll come out exactly the size you want. PETG has a higher melting temperature, so some printers can't print it.
- ABS is not recommended for this keyboard. ABS is great for artistic projects: a quick acetone bath will melt layer lines into a glossy smooth outer finish. Unfortunately, ABS shrinks, or "warps", as it cools. This makes it difficult to print properly sized mechanical parts, and often difficult to print anything at all. Some older guides recommend ABS as stronger than PLA, especially for "harsh outdoor environments", but nowadays PETG thrashes ABS in every mechanical metric.

To explore the wider world of filaments and find one that's right for you, check out PrusaSlicer's Material Knowledge Base. https://help.prusa3d.com/materials Personally I've had good experiences with Hatchbox and Overture brands.

Filament normally comes in 1kg spools. One spool should be enough for a Dactyl Chimera, but I'm not 100% sure because I used multiple spools to have fun with colors.

Wires and solder:

- **Wire:** You will NEED solid core wire, not stranded wire, to assemble this keyboard. Make sure your wire strippers and wire gauge are compatible.
- **Solder:** You'll only be using through-hole components for this keyboard so any type of solder should work. Even the optional RGB LEDs, despite being surface mount devices, are relatively large and do not require flux.
- *Solder and wire recommendations can be found at https://docs.keeb.io/soldering-tools.*

Diodes:

- You'll need throughhole diodes for this keyboard. A throughhole diode looks like a cylinder with a metal wire, or "leg", sticking out each end. Here's an example: https://www.sparkfun.com/products/14884
- You'll need at least 62 diodes: one for every key.
- Diodes are extremely cheap and reliable, so you can buy them from just about anywhere. I recommend bundling them in with your purchase of microcontrollers, wires, or other electronic parts.
- If you can't find diodes, (and somehow modern society isn't crumbling as a result) try a company from this list: https://en.wikipedia.org/wiki/Category:Electronic_component_distributors

Screws and nuts:

- Just like diodes, screws are the same everywhere, right? Well, if you buy screws from Amazon like I did, you might get black paint coating your fingers and nuts that do not thread onto bolts! Don't make my mistake; please buy from a dedicated hardware store.
    - Physical hardware stores: https://en.wikipedia.org/wiki/Category:Hardware_stores
    - Online retailers: https://en.wikipedia.org/wiki/Category:Industrial_supply_companies some stores only ship to businesses, others cater to individual hobbyists. I recommend McMaster-Carr. Pay attention to the pack quantity. You can buy a pack of 100 regular screws for the same price as one specialty screw.
- ~~External Hex is ideal for this application. You can 3D print a wrench instead of trying to fit a screwdriver into awkward places.~~ Correction: Turns out my design was so good that you can use regular Philips head screws and fit a full size screwdriver in every space with a screw hole!
- Buy "machine screws", not tapping screws. Machine screws thread into nuts; tapping screws cut into material.
- "Steel" or "steel alloy" is the cheapest material. We'll be using some of the screws to conduct electricity; don't pick nylon or other plastic screws.
- Metric machine screws are sized like this: "M4 x 0.7". M4 is the diameter of the shaft in millimeters. 0.7 is the "thread spacing," or "thread pitch", and it is usually the distance the screw will travel after one full rotation. If you buy M4 x 0.7 screws, make sure to buy M4 x 0.7 nuts.
- The keyboard's default screw holes are sized for M4, but this is extremely easy to change. At the absolute minimum, 78 screws at least 10 mm long and 16 screws at least 14 mm long are necessary, but aim above that. To keep things simple, buy a 100 pack

of 10 mm long screws https://www.mcmaster.com/91280A043/ and a 50 pack of 20 mm long screws: https://www.mcmaster.com/91280A045/

- Don't forget nuts. Technically you could get away with 98 nuts, but the build process will be far easier with 152 nuts. Buy two packs (aka 200) of these: https://www.mcmaster.com/90370A204/
- To mount the microcontrollers, you'll need 8 screws of size M2, at least 12 mm long. Plastic is fine here. As an example: https://www.mcmaster.com/92005a035/ And again, don't forget nuts with your bolts: https://www.mcmaster.com/90591A111/

Rubber sticker mat:

- The Dactyl Chimera V3 is designed with rubber sticker sheets in mind. (this stuff: https://www.mcmaster.com/rubber/width~12/backing-type~adhesive/length~12/ These sheets can be cut to any size and should keep the keyboard from sliding around. Unfortunately, I have no idea where to buy good rubber sheets since prices and quality vary wildly.

Parts

Switches:

The Dactyl Chimera is compatible with MX-Style and Choc V1 switches. You need at least 62 switches. Here is a list of online switch sellers. https://www.reddit.com/r/MechanicalKeyboards/wiki/switch_suppliers/#wiki_cherry_mx_switches

If you don't know what "MX-style and Choc V1" means, you'll probably want to learn about keyboard switches here: https://www.reddit.com/r/MechanicalKeyboards/wiki/switch_guides/#wiki_general_mechanical_switch_guides

Keycaps:

You'll need keycaps for your keyswitches. Buy your keycaps from one of these vendors: https://www.reddit.com/r/MechanicalKeyboards/wiki/keycapsellers/

There are many types of keycaps. My suggestion is to just pick PBT caps that look good, and to avoid RGB backlit keycaps unless you plan to install per-key backlighting. You can also go down this rabbit hole: https://www.reddit.com/r/MechanicalKeyboards/wiki/keycap_guides/

Hotswap sockets:

You'll need hotswap sockets for your switches. This is not optional. Without hotswap, you'll need to redo the soldering every time you want to replace any part. Learn about (and buy) hotswap from one of these links:

- https://www.reddit.com/r/MechanicalKeyboards/wiki/switch_guides/#wiki_some_hot_swap_resources
- https://www.reddit.com/r/MechanicalKeyboards/wiki/key_removal_guides/#wiki_hotswap_keyboard_information

The Dactyl Chimera has space for 20mm by 20mm per-key PCBs, a feature typically only found on flat keyboards. If you know what you're doing, these PCBs can add per-key RGB and better cable management to your board, but keep in mind that hotswap is more important than RGB for this keyboard.

Microcontrollers:

You will need two microcontrollers, one for each side. I recommend using the same microcontrollers for both sides.

The Dactyl Chimera is designed to work with:

- The Raspberry Pi Pico: https://www.raspberrypi.com/products/raspberry-pi-pico/ (not the Pico W)
- ... That's it so far.

RJ11 jacks and cable:

The Dactyl Chimera is large enough to support RJ11 jacks to connect the left and right sides of the board. Make sure to stick to the brand listed here.

- Buy 2 of these jacks: https://www.sparkfun.com/products/132
- And one of these breakout PCB sets: https://www.sparkfun.com/products/14021
- For a cable... just look on Amazon. RJ11 was at one point popular for home telephones and Lego Mindstorms, but that time has long passed. I can't find a good reliable seller for RJ11 cables.

Magnetic USB cables:

Magnetic USB cables work similarly to Apple's MagSafe charger. While optional, these cables have many uses, including: quick-disconnect when troubleshooting, swapping which side is plugged in while flashing, reducing stress applied to the USB port, and generally being cool.

There are no reputable sellers of magnetic micro USB cables. Most are sold on AliExpress and resold on Amazon. You'll need a "Data + Charge" cable. Some cables are only for charging batteries and will not send keypress or firmware data. Finally, these cables are only good for simple peripherals like keyboards. For Lightning (Apple) and USB-C these cables do not meet the specifications for high power or reliable data transfer. Do NOT use these cables to connect a smartphone, external SSD, or other similar device.

Chapter 5: Adjusting your arches in FreeCAD.

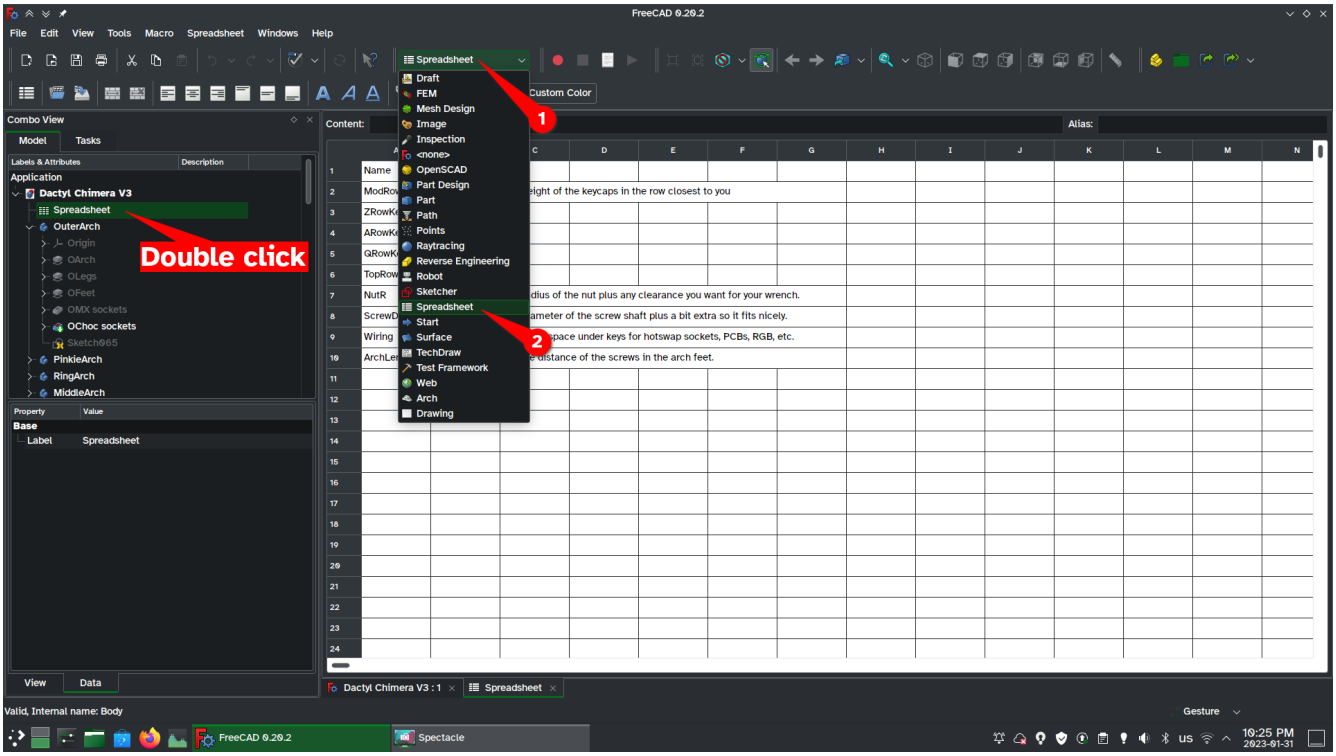
Here's how to install FreeCAD. https://wiki.freecadweb.org/Manual:Installing That URL is one page in the FreeCAD manual. The manual can help you make make your own 3D stuff, but it covers a lot more than you'll need for this keyboard project. **Make sure you actually read the "Installing" page I linked to!** It explains important settings like automatic file backups.

Navigating around in 3D CAD is different from moving in a video game, so you may want to read this: https://wiki.freecadweb.org/Manual:Navigating_in_the_3D_view

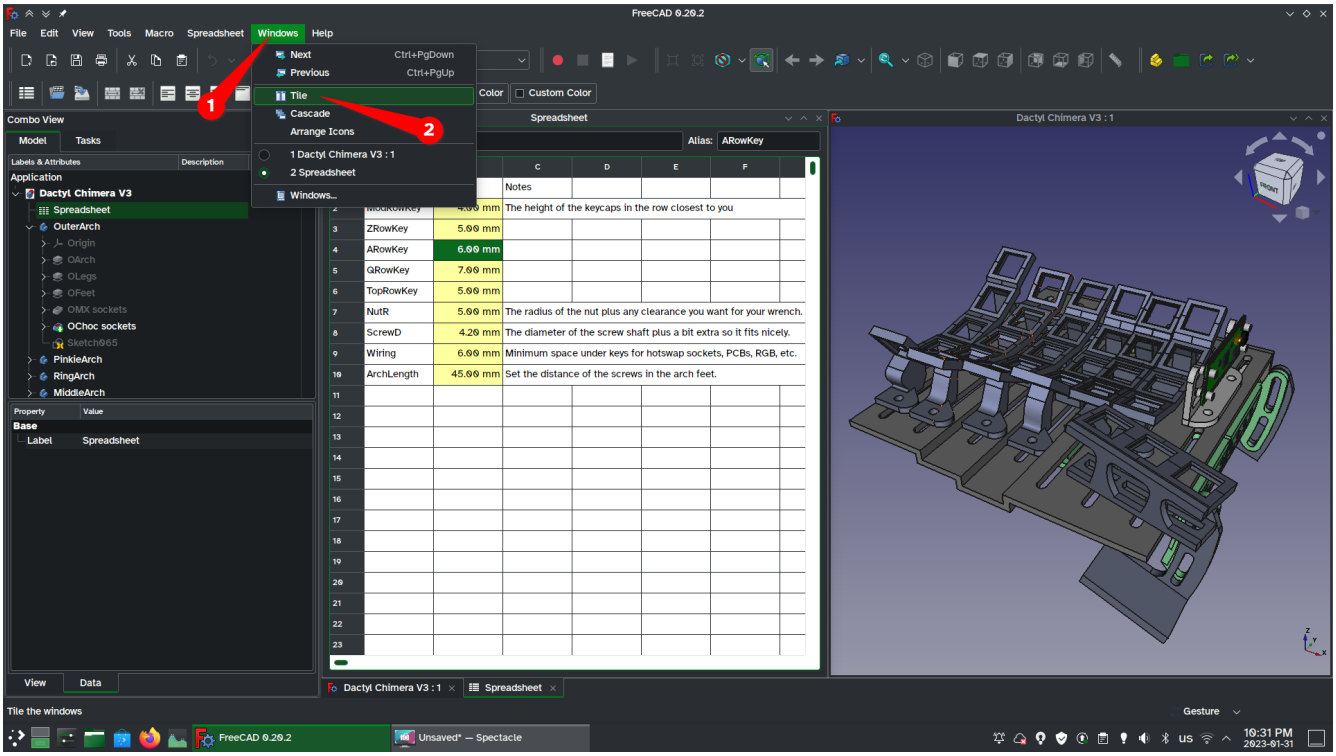
The name of the file with the keyboard in it is "Dactyl Chimera V3.FCStd"

Spreadsheet variables

The first thing you'll want to do is set your variables, such as screw size and wiring leeway. Double click on the Spreadsheet item in the Model list to open the spreadsheet. You should also change your workbench to "spreadsheet" in the workbench dropdown list.



If you want to see your model while you edit the spreadsheet, you can do that too. Click on Windows in the menu bar, then choose Tile.



Every time you make a change in the spreadsheet (even to edit text), the entire model will “regenerate”. Regeneration can be slow and frustrating, so think about your edits before you make them.

While I’ve taken care to reduce geometry errors (like things turning inside out or vanishing), large leaps in variable size can still cause issues. Save often! I usually save a copy called “Dactyl Chimera Sacrifice” before making major changes.

Explaining the Spreadsheet variables:

The first five variables are **ModRowKey, ZRowKey, ARowKey, QRowKey**, and **TopRowKey**. These variables define the height of each row of your keycaps. In other words, it defines how much taller your key becomes when you stick a keycap onto it. Some keycap sets create a slightly curved feel by making each row of keycaps a different height. Since we’re making our own curves, we need to counteract that effect. These variables offer a rudimentary way to flatten the uneven keycap height.

NutR: This stands for Nut Radius. Imagine a circle of protection around every screw: a guarantee that there won’t be any plastic in the way when you try to tighten the nuts down. NutR is the radius of this circle. NutR also sets the size of features like Arch Feet. 5 mm is a good minimum value, and you’ll want to bump this number up if you’re increasing your ScrewD.

ScrewD is the most important variable in the spreadsheet. It corresponds to the diameter of your screws. The default value is 4.2 mm (that's M4 + 0.1 mm extra on each side.) **Make sure you change this variable to match the diameter of screws you bought.** BTW, the Microcontroller screw holes will always be sized for M2 screws because the Pi Pico has M2 holes on it... so don't worry about that.

Wiring sets the minimum distance between the arch and the base plate. You want this number to be small; the slimmer your keyboard is on your tabletop, the more comfortable you will be. If you're having trouble getting your hotswap sockets, rgb lights, diodes, wires, etc to fit, you can bump up this number. I don't think you should go lower than 6 mm.

ArchLength: This variable was mostly used in development and it determines how far each arch's screw hole is from the arch's home row key. You shouldn't need to change this number; the default value is 45 mm. You can increase it to 55 mm if you want to see what it does. Hint: ArchLength*2 + NutR*2 = total length of every arch.

If you're having trouble using the spreadsheet, read this documentation: https://wiki.freecadweb.org/Spreadsheet_Workbench

Adjusting the arches:

This is the moment you've been waiting for: It's time to edit the document! Don't worry, it's "easy".

We need to open the "sketch" that defines our arch. In FreeCAD, you create a 3d model by "sketching" its outline in 2d. By changing the size of the sketch, you'll be able to change the shape of the 3d object!

Part 1: getting to the sketch

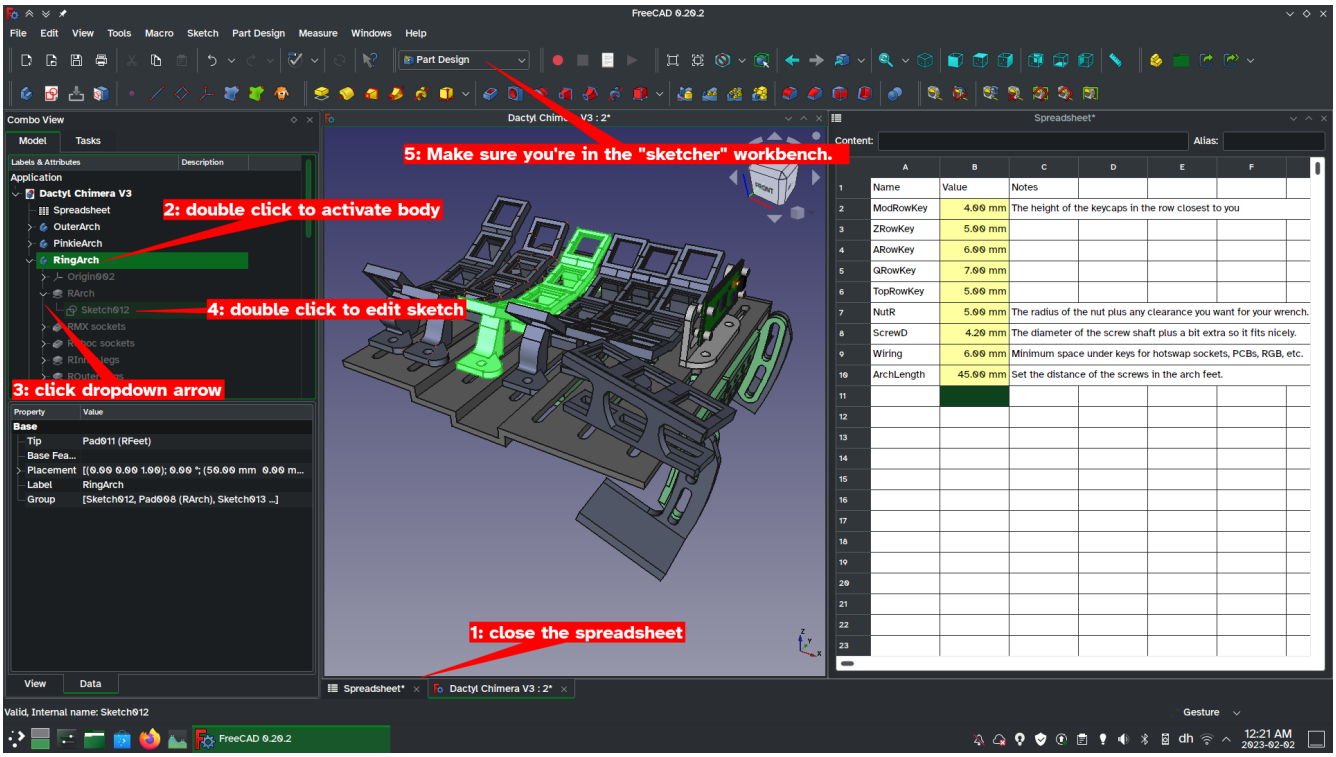
1. Close the spreadsheet tab, but keep the 3d model tab open.

2. Double click on the body you want to edit. Each "body" is a part that will be 3D printed separately. When a body is active, it becomes bold. In this example I'll be editing "RingArch".

3. Click the dropdown for the part you want to edit. In this case it's RArch. Usually the stuff you'll want to edit will have Arch in the name.

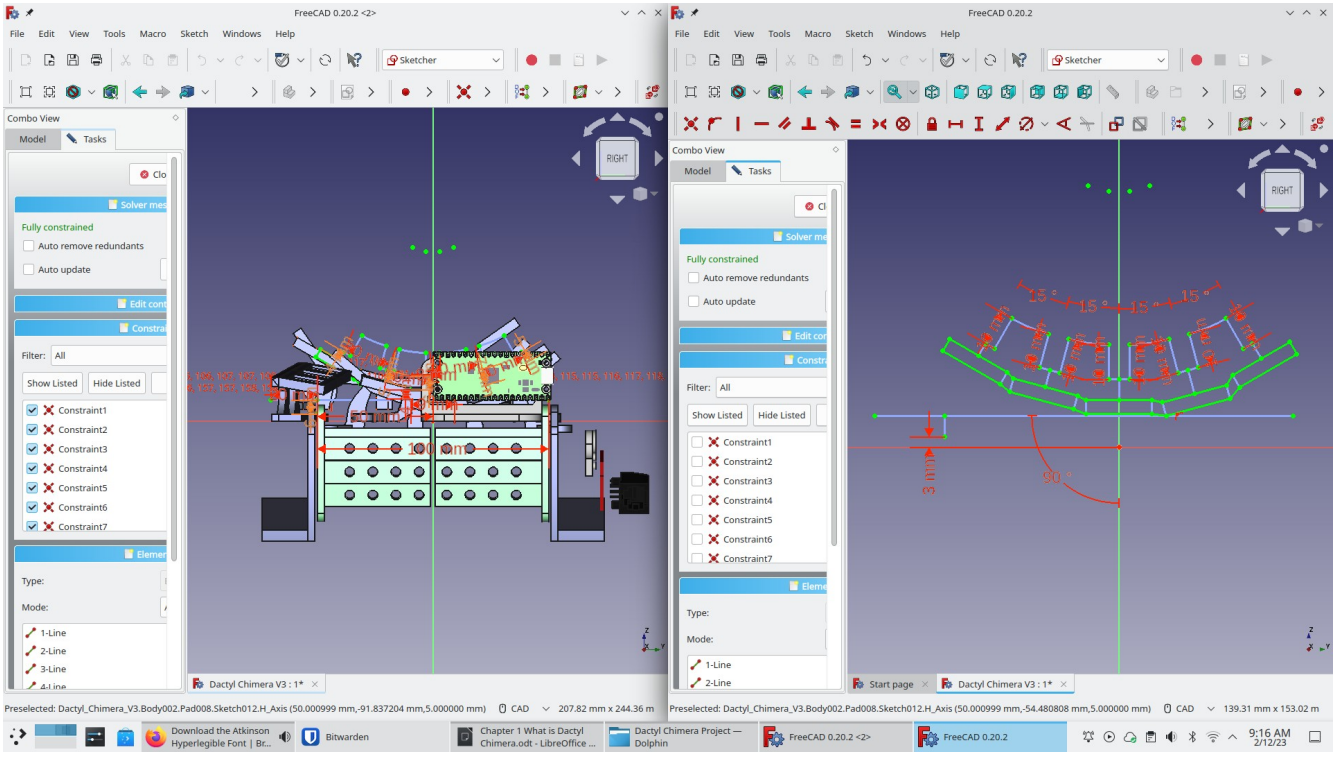
4. Within RArch you'll find a sketch. Double click on the sketch to edit it.

5. Now that you're editing the sketch, you should automatically be in the sketcher workbench, but you should check to make sure.

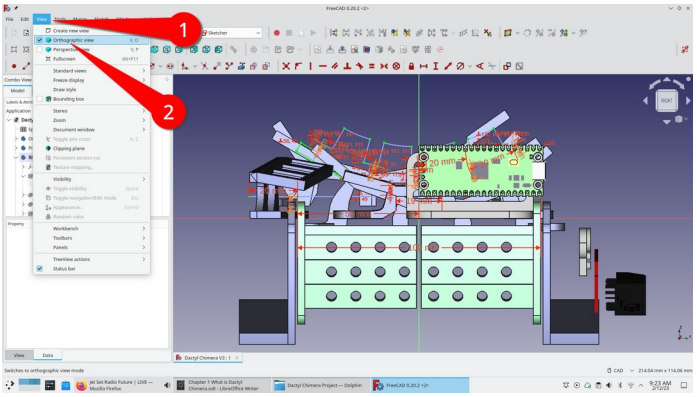


Part 2: cleaning out the rat's nest

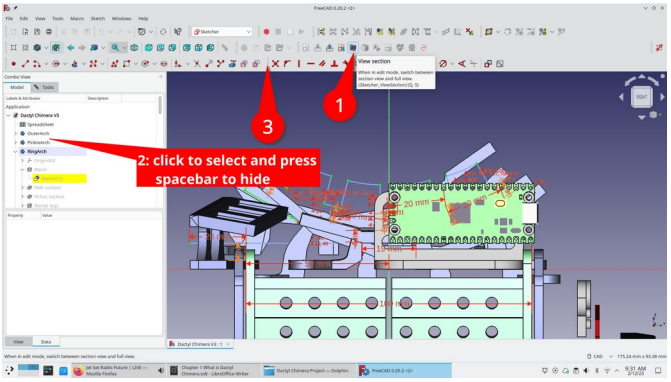
When you open the sketch, you'll probably be confronted by a mess that looks like the image below on the left. We want to change it into the image on the right.



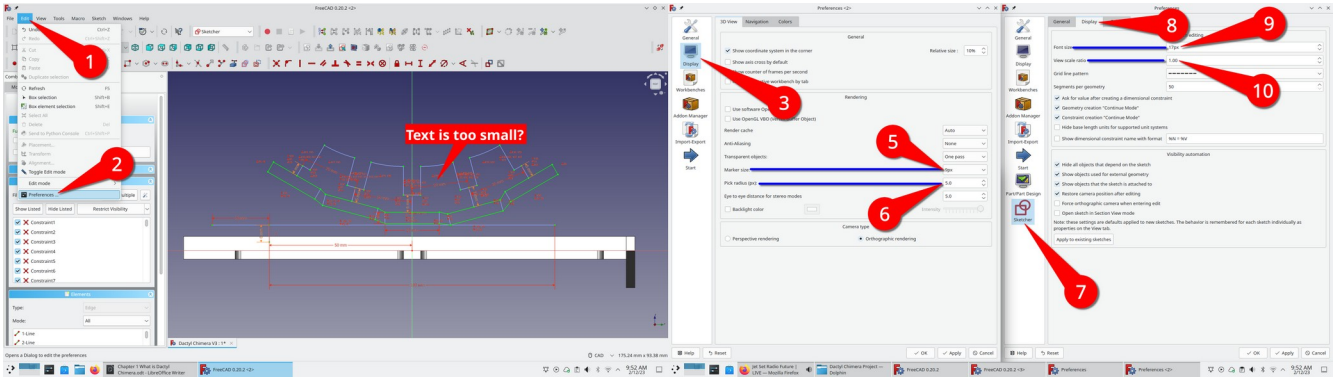
The first thing you'll want to do is make sure the view is orthographic, not perspective. Go to View in the menu bar and choose Orthographic. FreeCAD is supposed to switch to orthographic automatically when entering a sketch, but I was messing around with 3D glasses while making the keyboard and now it defaults to perspective mode.



The next thing to do is hide the items in front of and behind the sketch. To hide everything in front of the sketch, click the (1) "View section" button in the [sketcher] toolbar. To hide an object behind the sketch, click once to select it in the (2) Model view then press the spacebar to hide it. You may also want to rearrange your toolbars at this time to make all buttons visible (click and drag on the double vertical line marked in the screenshot as point 3)

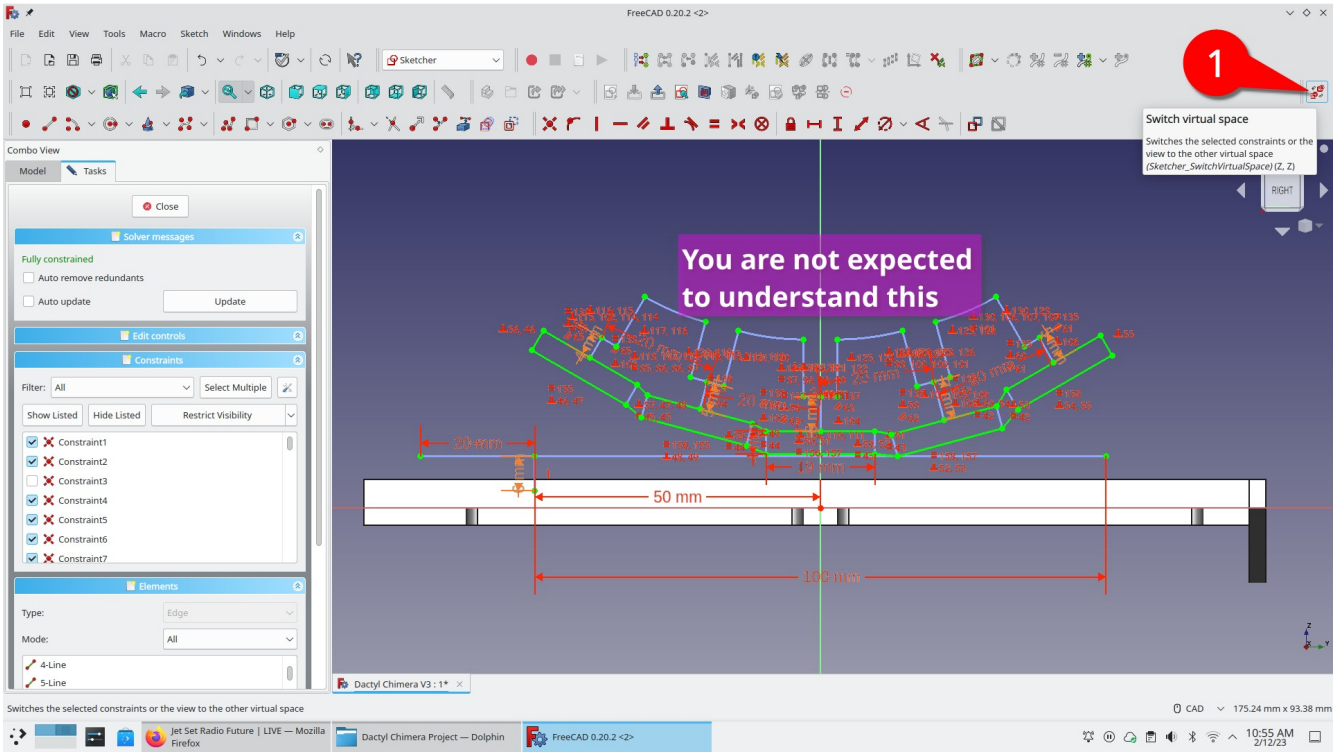


If you have a 4k monitor or other non-standard resolution, you may need to adjust a few settings to make text legible and lines visible. Under the (1) Edit menu, open your (2) Preferences window. In the (3) Display section, you can change your (5) marker size to make points bigger and your (6) "pick radius" size to make points and lines easier to click on. In the (7) Sketcher section, you'll want to go to (8) Display to bump up your (9) Font size and (10) scale to increase everything in the sketcher view. 2.00 is a good scaling factor for 4k monitors.



Note: Spectacle accidentally skipped the number 4 while I was annotating.

One last step: There are two sets of measurements in each sketch. If you see a garbled mess on screen, click the “switch virtual space” button.

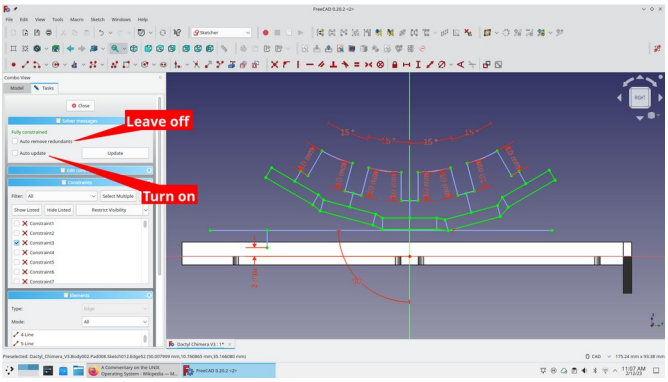


Part 3: modifying the sketch

You are now ready to edit the sketch!

Unfortunately, as stated way back in the beginning, I am not an ergonomics expert and cannot give you advice on what shape you should aim for. (hint: if you are an expert on dactyl shapes, please add that info!)

I can, however, give you some pointers on how to edit sketches in FreeCAD.

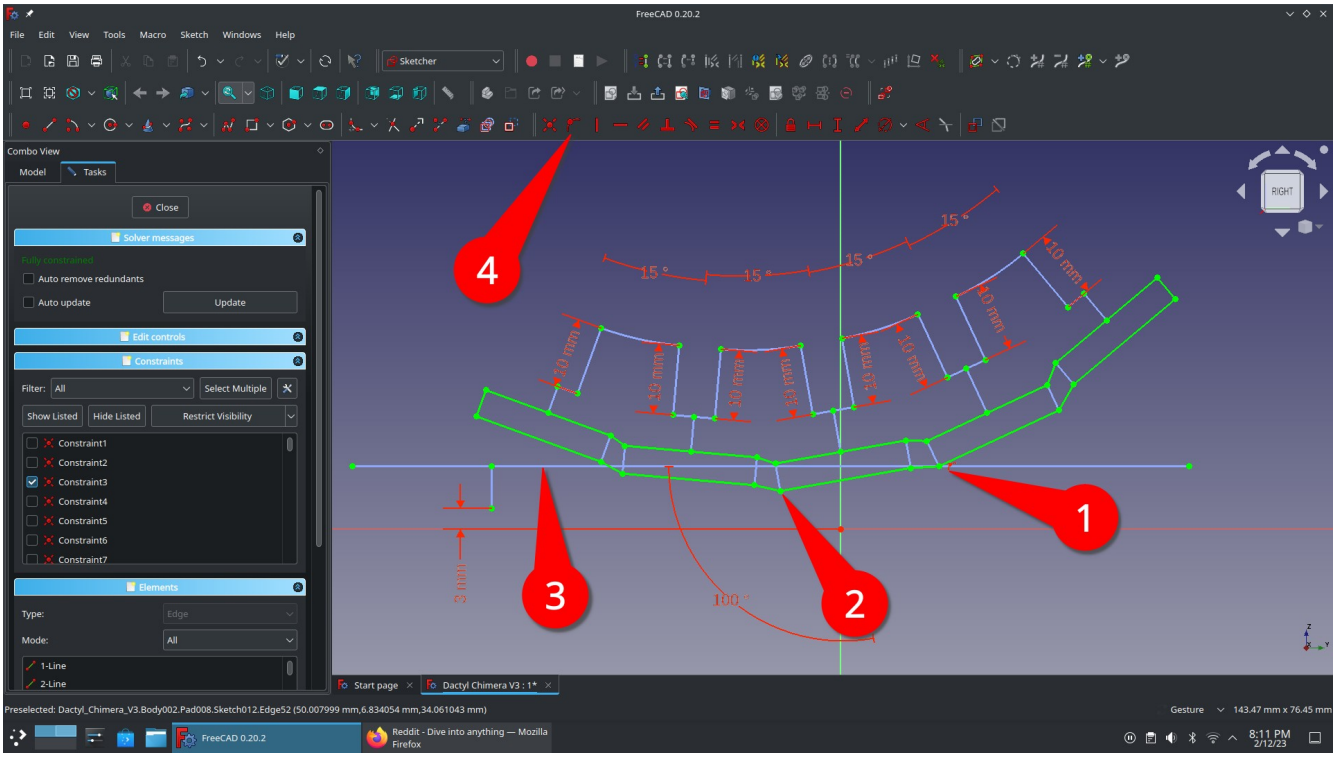


- Leave “Auto remove constraints” turned OFF. There is a glitch where FreeCAD sometimes identifies all constraints as redundant. There is no limit on the number of constraints FreeCAD can auto-remove at one time, so activating auto-remove can occasionally disassemble the entire sketch automatically.
- Turn on Auto Update or get in the habit of clicking “Update” often. If you don’t “update” often enough, FreeCAD can lose track of how the 2D sketch and 3D model are related to each other.

- Do not delete or add any lines to the sketch. Doing this will break the model; specifically it will make the holes for the keyswitches disappear. There is a way to fix this but it takes too long to explain.
- Oh yeah, you need to double click on constraints (distances and angle dimensions) to edit them. Have fun and play with all the numbers!

Part 3.5:

While editing, your arch may drop below the wiring baseline (blue line). If this happens, you will need to detach the arch and attach its new lowest point to the wiring line. Don't worry, this is a simple 4 step process.



1: Select the "point on line" constraint object and press your keyboard's delete key. (not the backspace key.) The arch should turn white, indicating that it is free to move around. (It is no longer "fully constrained".)

2: Click on the lowest point in the arch. That point will turn dark green.

3: Click on the horizontal blue line. The line should turn dark green. The point you selected in step 2 should also still be dark green.

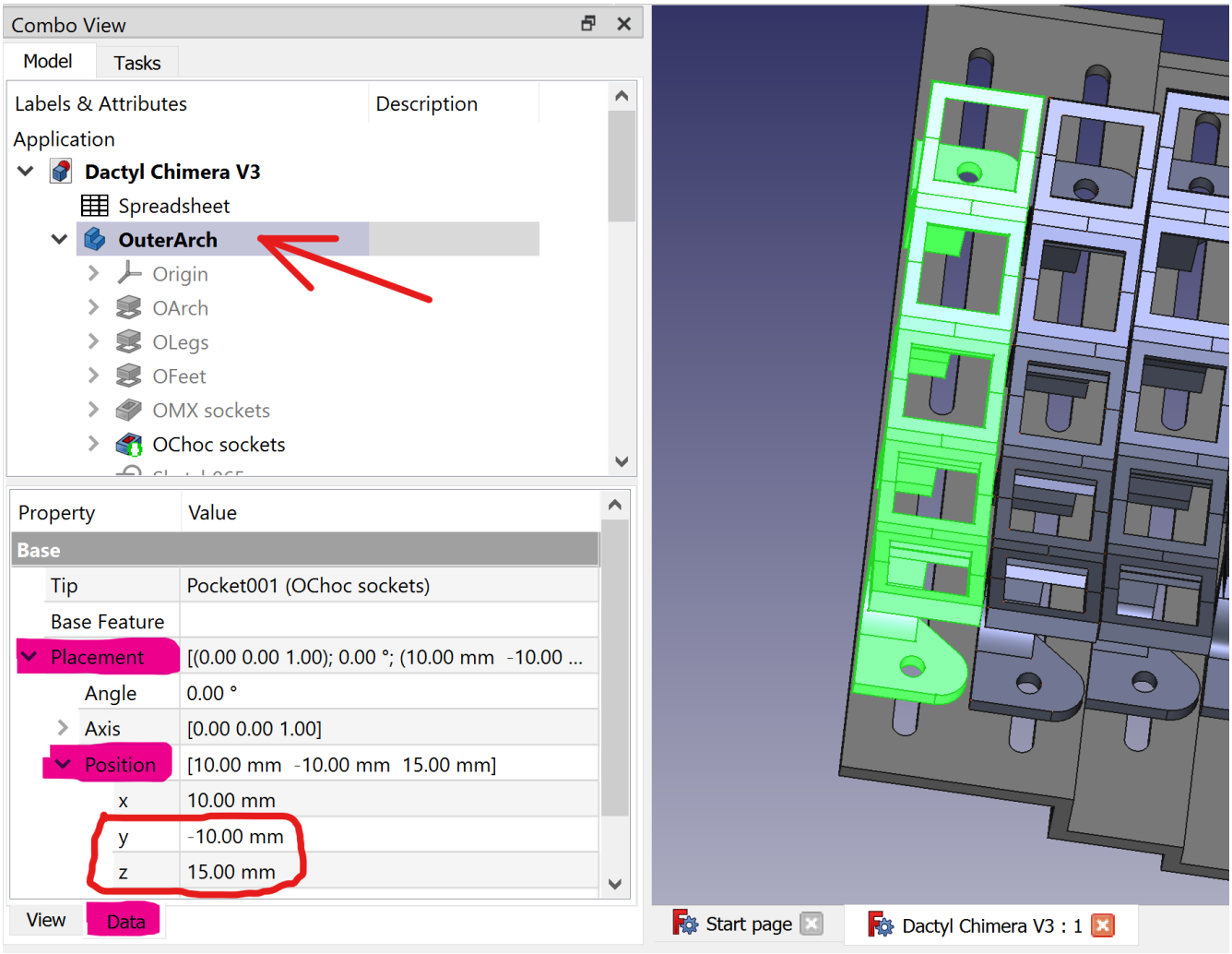
4: Click the "Constrain point onto object" tool in the [Sketcher constraints] toolbar. The point you selected should snap onto the line and the arch should turn green to indicate that it is once again fully defined.

Congratulations! That's all you need to know about editing an arch in the Dactyl Chimera!

Part 4: Raising and lowering the columns

If you want to raise or lower the columns, please **SAVE A COPY OF YOUR FILE FIRST.** These edits suffer from the Topological Naming Problem, a very nasty glitch that you **CANNOT FIX BY PRESSING THE UNDO BUTTON.** (If you're interested in learning more about TNP, it's documented here: https://wiki.freecad.org/Topological_naming_problem)

Now, to actually change a column's position, here's what you need to do:



1. Double click the column body you want to move
2. In the Data tab, open the Placement dropdown, then the Position dropdown

3. Edit the Y parameter to slide the column along its slot. Edit the Z parameter to raise or lower it above the table.

Notes:

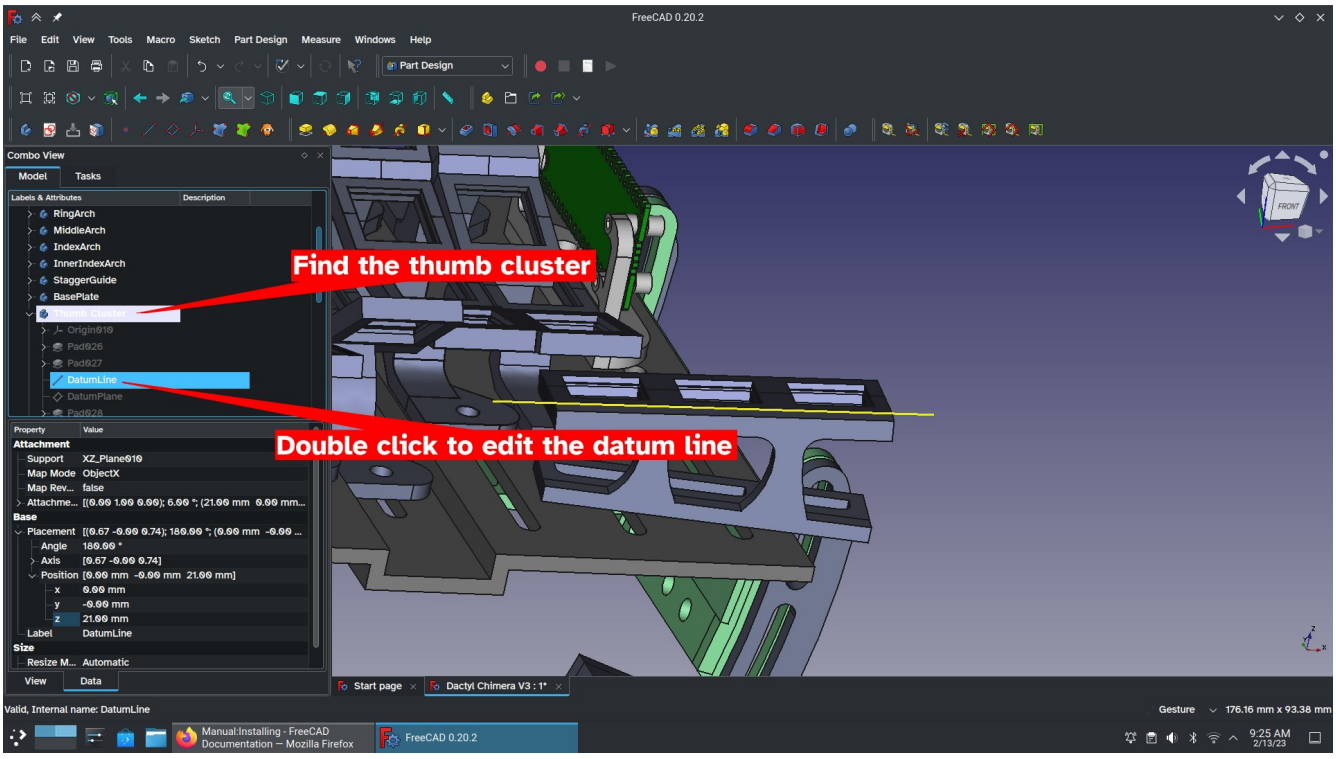
The Base Plate will automatically adjust to the new position of your columns, but you must still follow the basic Dactyl shape: i.e. the middle column must have a lower Z position than the other columns.

After adjusting the Y values, inspect the base plate to make sure the slot screw holes are being created properly. The holes under the Thumb Cluster are especially likely to cut through the outer perimeter.

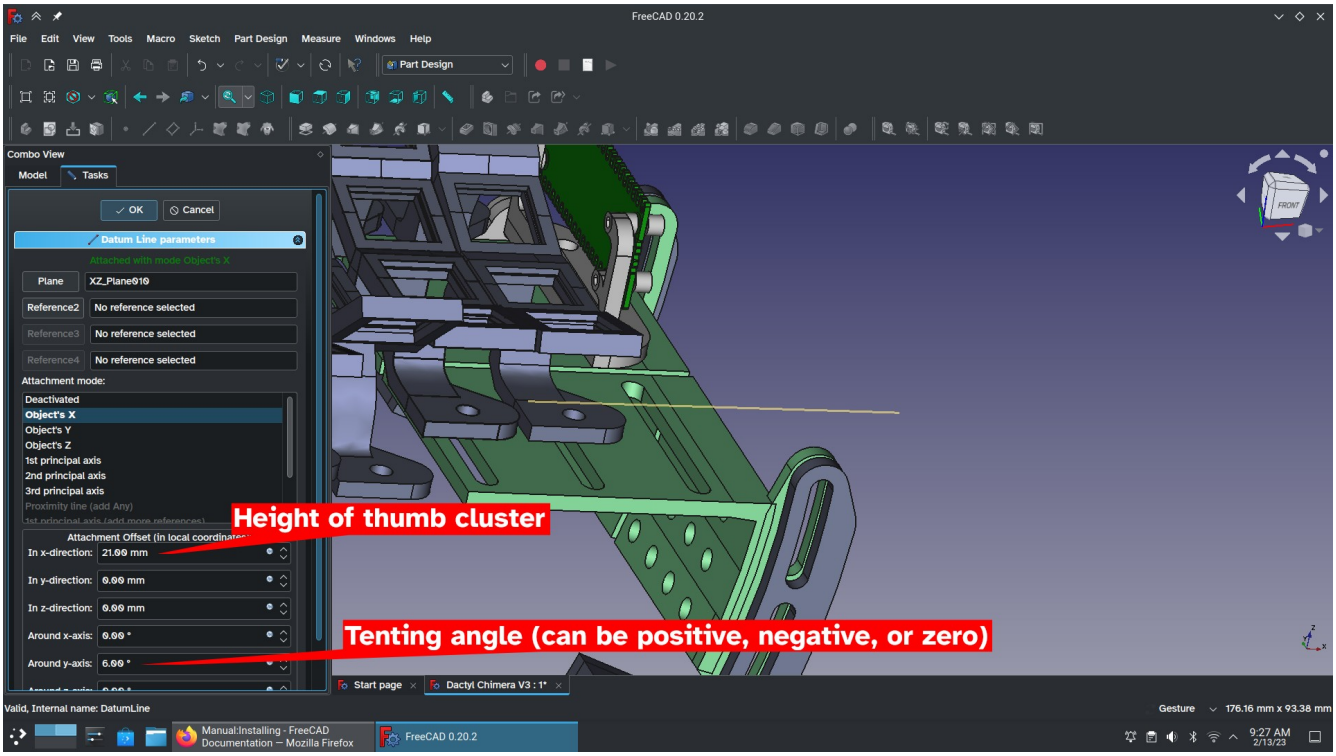
After adjusting the Z values, check the holes on The Conductor (the tenting legs.) If the holes have disappeared, you have experienced the topological naming problem. There are ways to fix your model and achieve extreme Z height values. Contact me and I'll help you out.

Part 5: Editing the thumb cluster

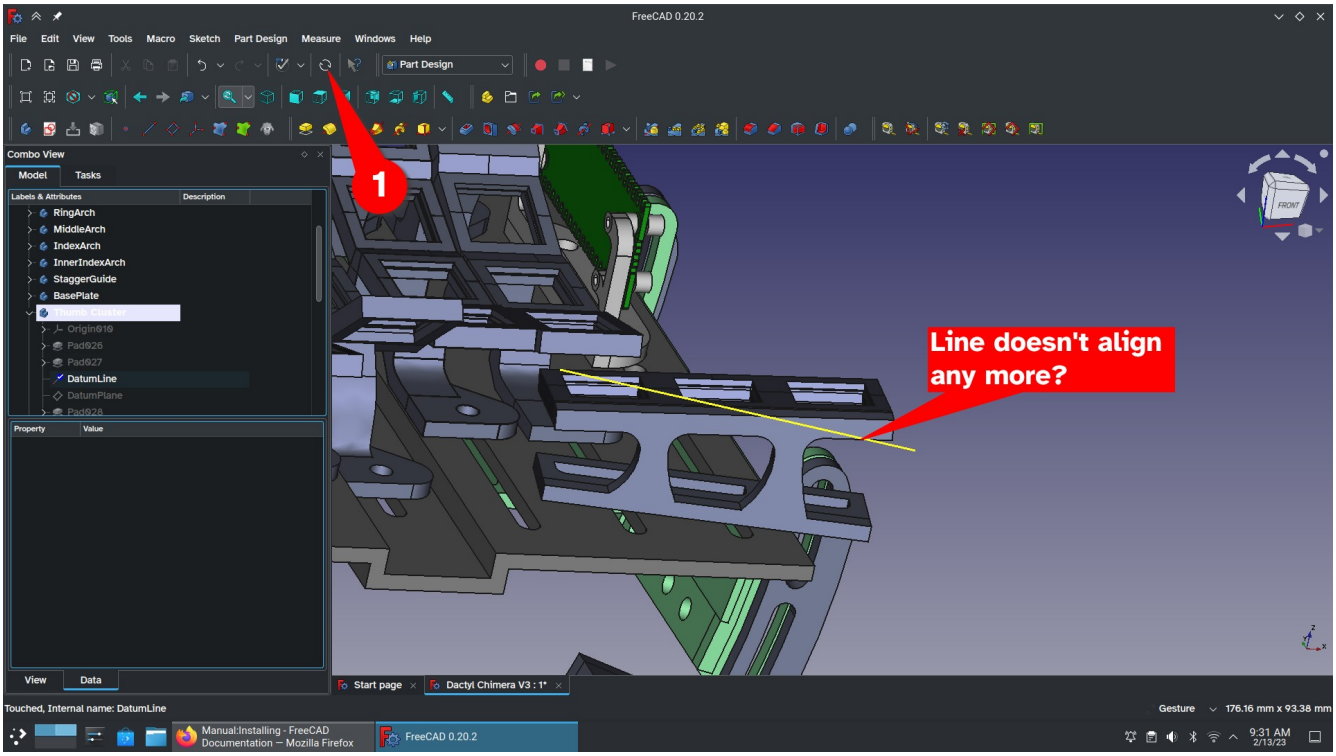
To edit the height and tenting angle of the thumb cluster, you'll need to edit the datum line as shown here:



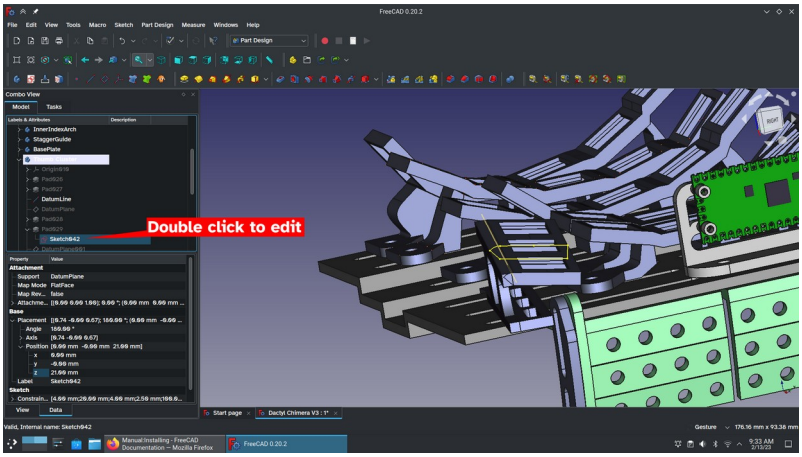
You must ONLY edit the “In x-direction” and “Around y-axis” values! If you edit any other values the model may break.



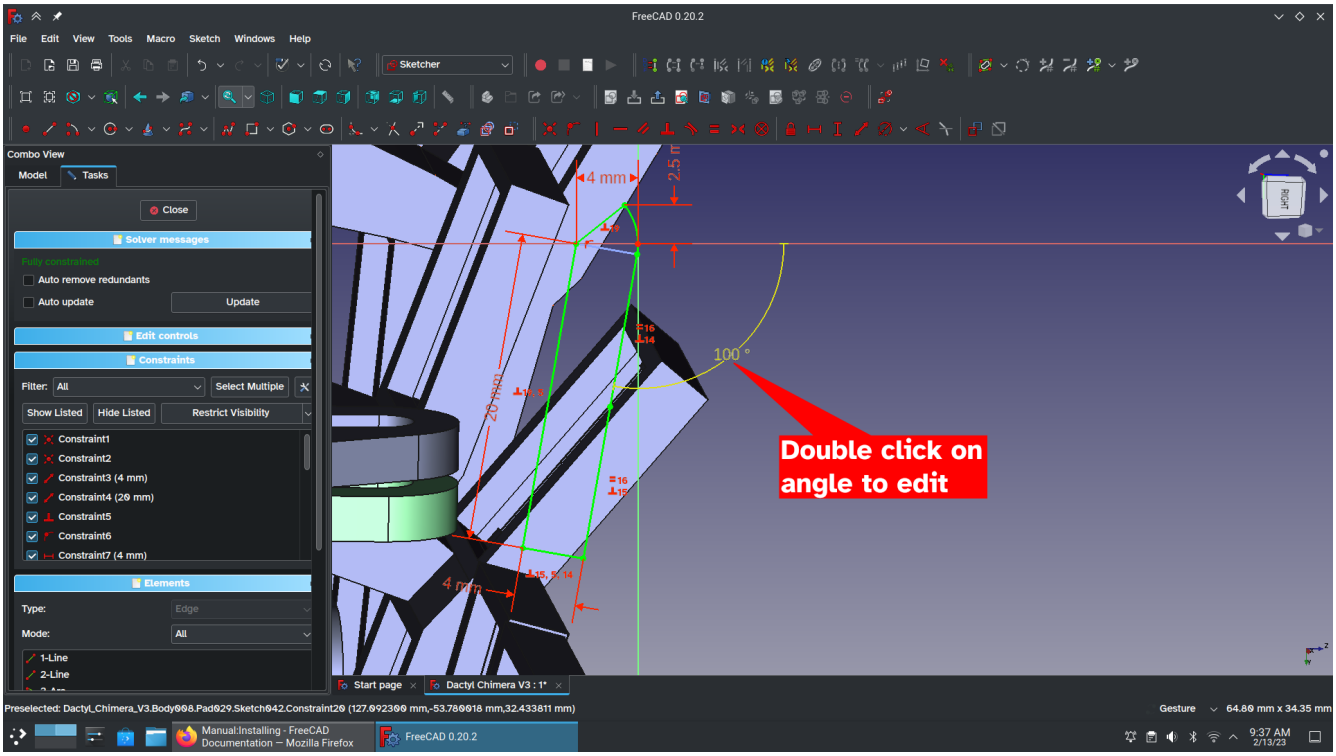
Finally, you may notice after changing some values that the 3D model doesn’t line up with your datum line any more. This is normal; it’s just one of the few times FreeCAD doesn’t auto-regenerate your model. Click the Regenerate/Refresh button to fix it. (this button may look different depending on your operating system.)



To edit the tilt angle of the thumb cluster, you need to edit Sketch042 inside of Pad029. Yes, I got lazy with my naming, sorry.



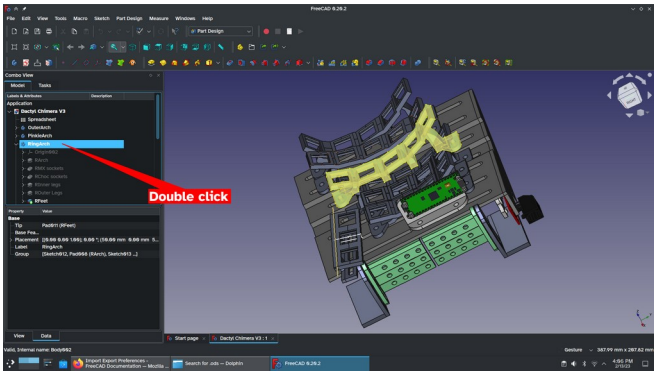
Once inside of the sketch you'll change the angle dimension to change the tilt. You should be able to set this value to anything between 70 degrees and 110 degrees.

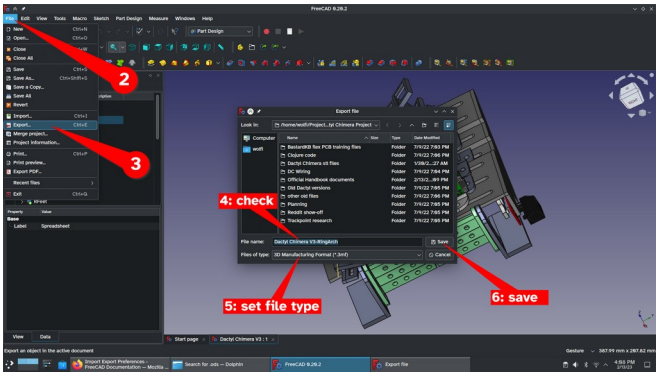


And that's it! You now know how to edit a Dactyl Chimera. Congratulations!!!

Part 5: Exporting your handiwork

You can only export one body at a time. First, double click on the body you want to export. Click on (2) File in the menu bar, then choose (3) "export" to open the export window. Check the name of your file (4). It should match the name of the **body** you want to export by default. Now set your (5) file type to .3mf or .amf. (Yes, STL is the "standard" for 3D printing, but it's a very old file format... we're talking 1980s old. .3mf and .amf are better.) With the file type selected, simply hit (6) save. Repeat this process for all the bodies you need exported.





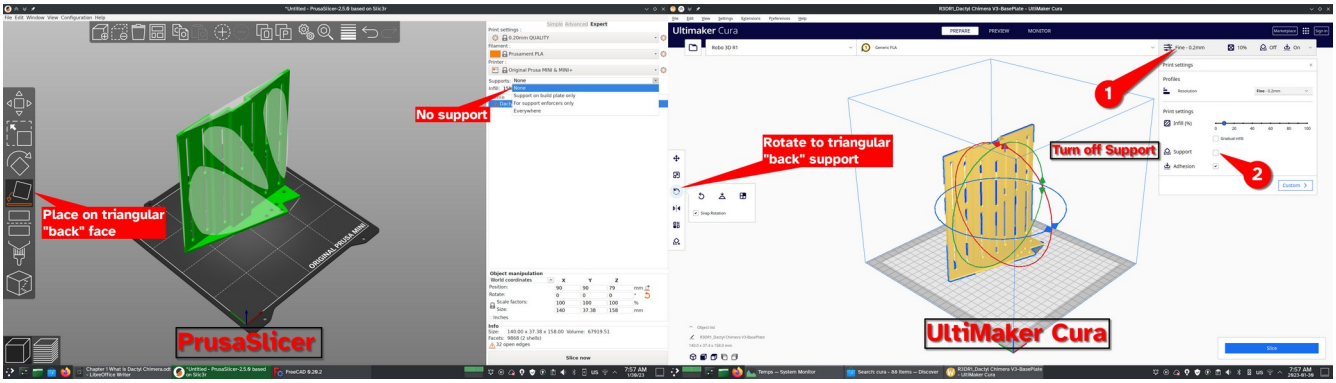
If you’re having issues with exporting, check this page:
https://wiki.freecad.org/FreeCAD_Howto_Import_Export

Chapter 6: Growing a Skeleton (3D printing)

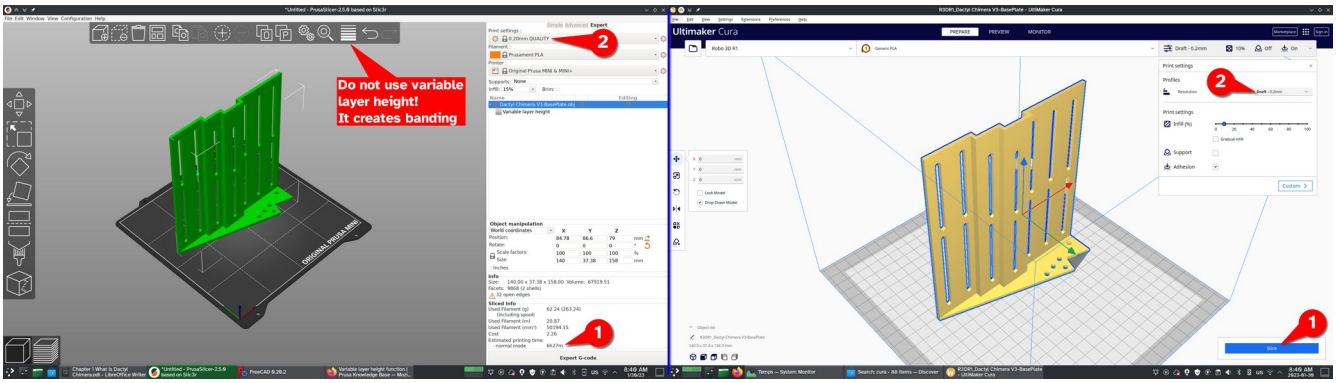
This chapter focuses on how to handle the .3mf and .amf files. It will not tell you how to set up your printer. For instructions on filament loading, bed leveling, and whatever else, check the printer's instruction manual. For a public printer, ask the owner for help. If you're having issues with print quality, the Fix My Print subreddit can help: https://www.reddit.com/r/FixMyPrint/

The base plate

The base plate is by far the largest part and it will determine if your printer is big enough. Fortunately, setting this one up is simple: all you need to do is rotate the model so that the triangular backside touches the build plate and then turn off "supports" for the model.



Since the base plate is the largest part, it also takes the longest time. I recommend checking the print time and then adjusting layer height ("quality" or "resolution") to match the time you want. I recommend against using PrusaSlicer's "variable layer height" tool here because it causes noticeable "bands" in the print at layers where the screw slots begin and end.

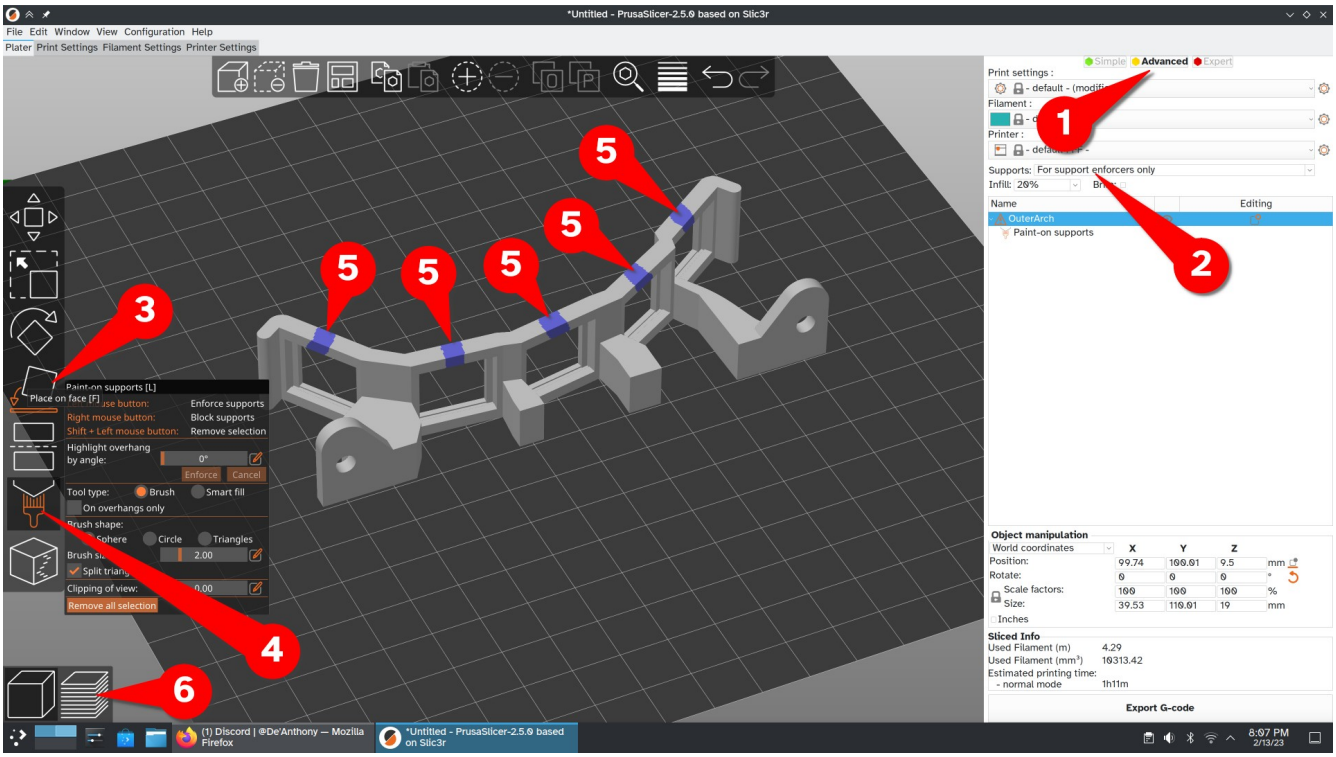


Remember: you'll need two base plates: one for the left hand and the other for the right hand. You must use the slicer's mirror button (found next to the Rotate button you found earlier) to get the other hand.

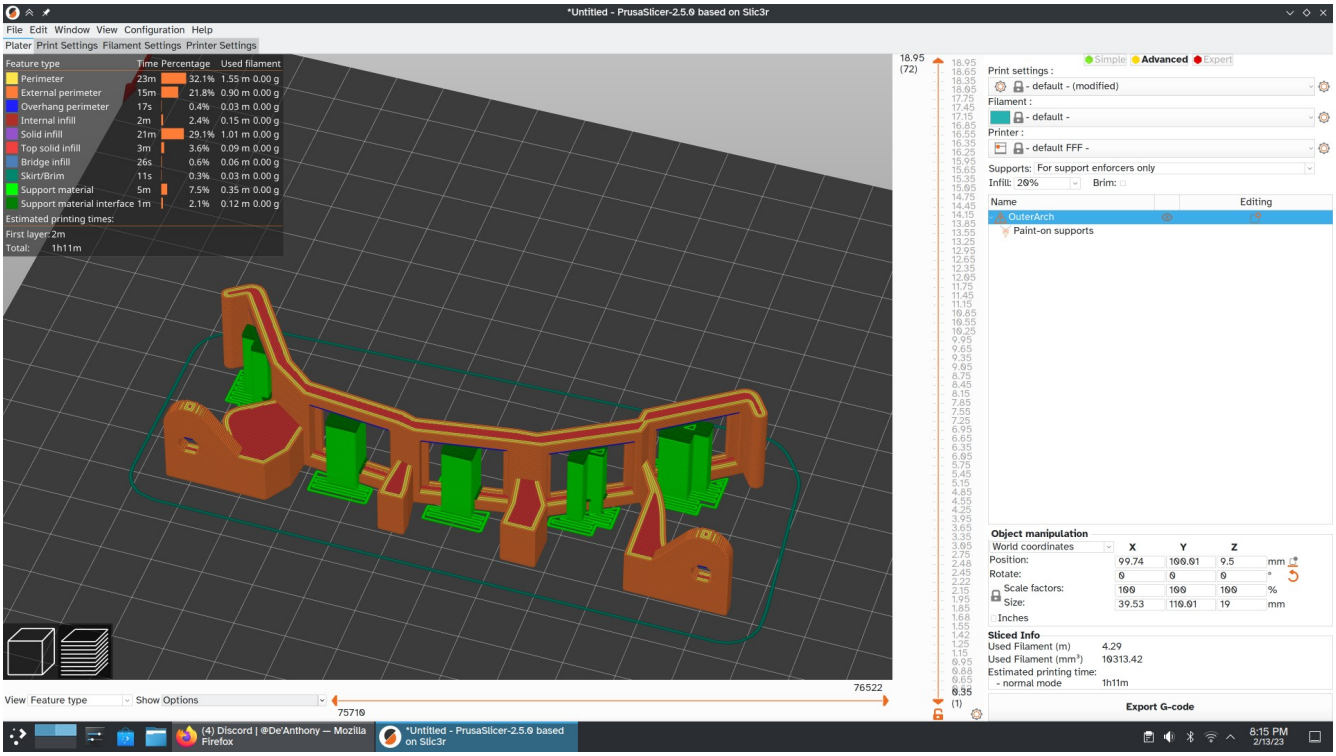
The Arches

(Cura stopped working on my desktop so for now you're only getting PrusaSlicer instructions.)

For the arches, you'll need to switch PrusaSlicer into Advanced (1) mode (this makes the paint-on supports button visible) and turn on support enforcers (2) only. After using Place on Face (3) to lay the part flat, turn on Paint-on supports (4). You need to add five blobs of blue support "paint" by painting over the top of the keyswitch holes. (5)



Check that the model is generating properly (6). The result should look like this:



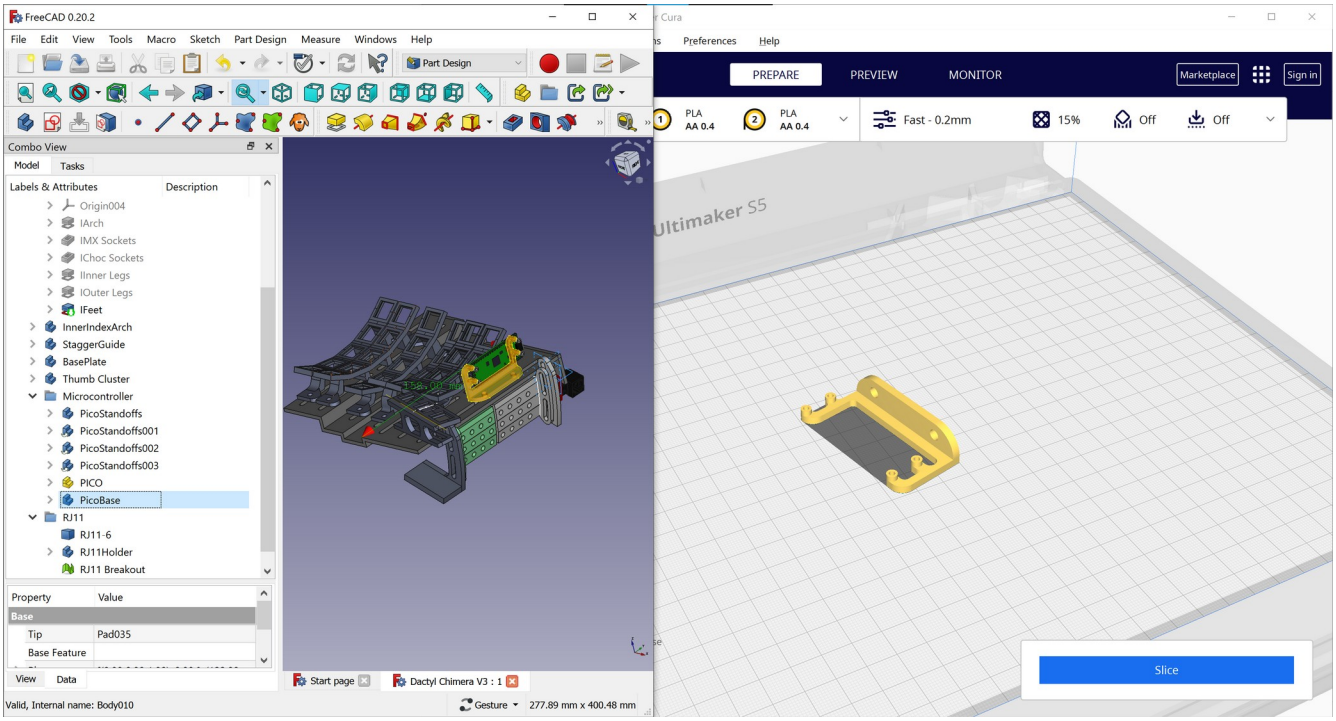
Now print the arch. Once your arch has been printed, you can easily remove the support material by yanking it away with your needle-nose pliers:

The thumb clusters

The default thumb cluster is printed almost identically to the arches: lay it on its side, paint support enforcers into the center of each hole, and print!

Microcontroller Holster

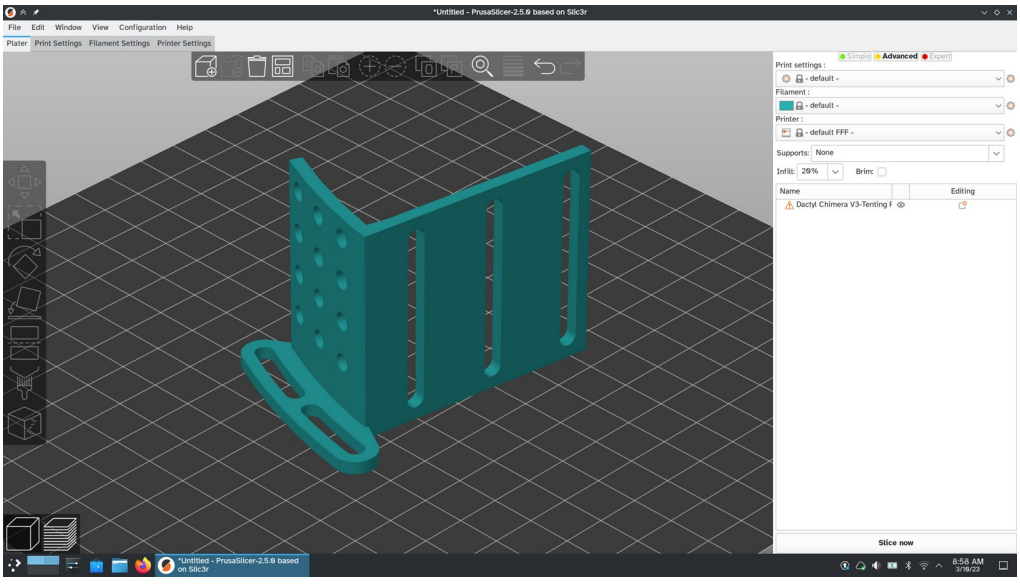
The microcontroller holster is called PicoBase and is in a folder titled “microcontroller”. It does not require supports and should be printed with the large complex side on the build plate.



Hey, look at you, Cura! You’re playing nice again!

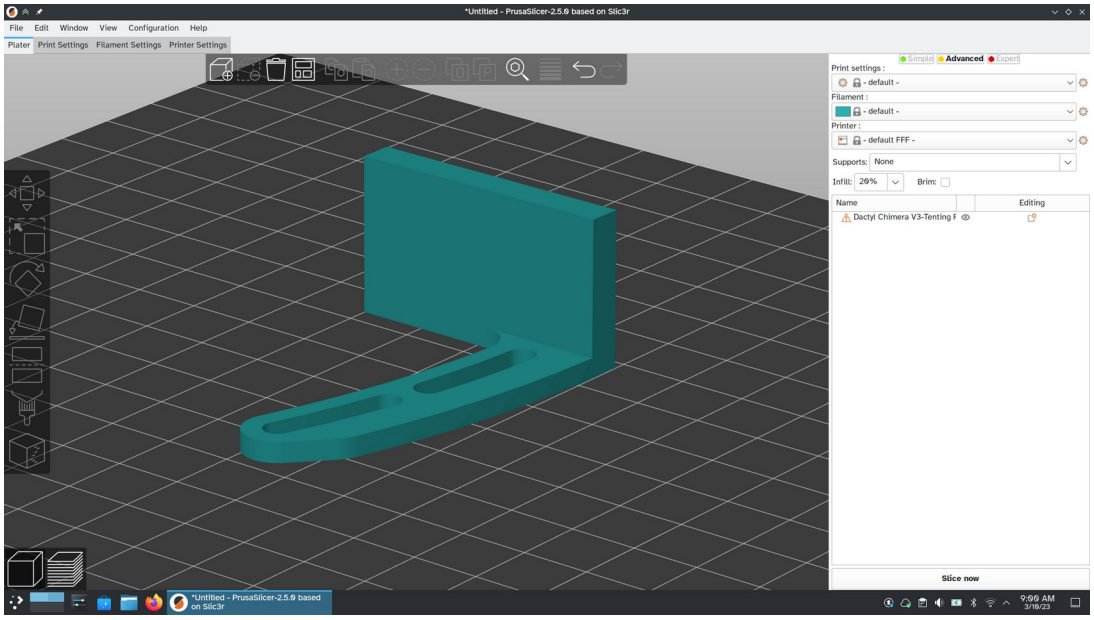
The Conductor

The conductor requires no support and should be printed like this:



The Feet

The feet do not require support and should be printed like this:



Chapter 7: The Brains (flashing firmware)

Dactyl Chimera uses a pair of microcontrollers to register keystrokes and relay them to the computer. But how do the microcontrollers know what to do when a key is pressed? Microcontrollers run a type of software called “firmware”, in this case the Quantum Mechanical Keyboard firmware (QMK). The process of installing firmware onto a microcontroller is called “flashing”. This chapter will explain how to modify QMK to match your layout design and how to flash that firmware onto your keyboard.

You’ll be following the official QMK tutorial: https://docs.qmk.fm/#/newbs This tutorial can be very confusing, so don’t feel bad if you get stuck. Check out https://docs.qmk.fm/#/support for help.

For compiling the Dactyl Chimera firmware, the name of the keyboard is: **It doesn’t exist yet.**

To put the Raspberry Pi Pico into Bootloader mode, unplug the RJ11 cable and then **I don’t know how to enter the bootloader.**

Beyond setting up your config.c file (which is explained in the tutorial linked above) there are two files you should check out and modify if necessary:
config.h, explained here: https://docs.qmk.fm/#/config_options
rules.mk, explained here: https://docs.qmk.fm/#/hardware_keyboard_guidelines?id=rulesmk

If you want to edit your keymap using a GUI, check out https://www.caniusevia.com/

Chapter 8: The Nervous System (wiring and soldering)

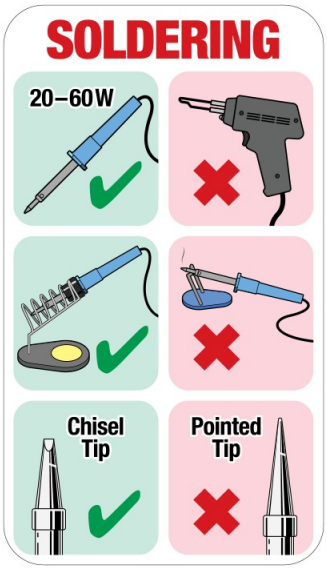
I am not qualified to explain soldering; fortunately there are other established guides that explain the process properly.

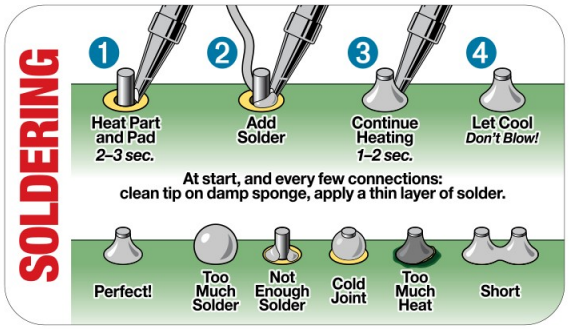
Due to the fabulous design of The Conductor, you don't have to worry about the pinout on your microcontrollers while soldering. You do want to make sure your diodes are facing the correct direction (black tip pointing towards the Conductor, orange half close to the keyswitches.) Finally, do not forget to use the hotswap sockets!

Here are the soldering guides:

- https://docs.qmk.fm/#/hand_wire?id=hand-wiring-guide
- https://www.reddit.com/r/MechanicalKeyboards/wiki/customkeyboards/#wiki_step_by_step_guides

I also found this infographic to be immensely helpful:





Once the QMK firmware pinout has been determined, photos of the complete wired up Dactyl Chimera will be included here. (there are no photos yet because my soldering skills are truly awful and pictures would probably confuse you more than help.)

Chapter 9: The Chimera (attachments and enhancements)

Now that you've built your Dactyl Chimera, it's time to create accessories and add-ons. There is a lot of 3d modeling software you can use to make what you want.

- Blender https://www.blender.org/ optionally with the CAD Sketcher addon: https://makertales.gumroad.com/l/CADsketcher *Probably the "best" at everything 3D!*
- SolveSpace: https://solvespace.com/ *Only 6.93 MB!*
- OpenSCAD: https://openscad.org/ *Popular among other Dactyl variants!*
- You can also learn how to use FreeCAD's "part" workbench. https://wiki.freecad.org/Part_Module *Place objects like you're building with LEGO!*
- If you like FreeCAD, I encourage you to try my custom keyboard shortcuts: https://www.reddit.com/r/FreeCAD/comments/o267hv/dialasketch_20_less_is_more/ They make a world of difference.

General Tips:

- The spacing between columns is 20 mm, not the 19.05 of traditional keyboards.

As add-ons are developed, instructions for them will be added here:

OLEDs

RGB lighting

Rotary encoders (scroll wheels)

Trackballs and Trackpoints

Analog Joysticks

Speakers

I don't know why, but someone decided it would be cool for QMK to play chiptune sound effects like NES music.

Improving this handbook

Links to external databases are strongly encouraged. For example, instead of listing every online store that sells keyswitches, I linked to the r/mechanicalkeyboard wiki page on the same topic. This online wiki is kept up to date by a larger pool of users and is viewed by a wider audience. We can edit that external page to help not just ourselves, but *everyone*. If an online wiki ever goes down, readers of this handbook can always retrieve an older version through the Internet Archive Wayback Machine or Google Page Cache.

Make sure to explain every step, and assume the reader knows nothing. This guidebook isn't just written for hackers who know nothing about 3D modeling or machinists who can't install Linux. This guide is written for office workers who want a better number pad, gamers who want to give their mouse more space, and kids looking for a cool STEM project.

Explain everything in chronological order. Don't tell people that a topic will be "explained in a later section" unless you really need to. https://youtube.com/watch?v=S_EkEACM5qc Writing in chronological order lets people keep track track of where they are and it helps people who might not read the entire manual before starting the project.

Explain your thought process behind the design of the keyboard, not just what to do to make a keyboard. This is THE guide to the Dactyl Chimera, not just a "build" guide but a guide to future upgrades as well.

Do not be afraid to make changes.

Epilogue: The Dark Arts

Reader beware: the enhancements listed below are too close to the cutting edge to have a written tutorial anywhere. If you attempt any of these suggestions, you will be entirely on your own with no support. With that being said, I have listed these possibilities here for a reason: If you have the time, the skills, or the dedication to do one of these tasks and describe the process, the result would be revolutionary.

Void Switches

These 3D printed switches use magnetism instead of springs. They can detect exactly how far they've been pressed, allowing analog movement in video games. The switches can be personalized to your exact preferences of tactility and clickiness. Unfortunately the creator's documentation ranges from "highly technical" at best to "possibly unwritten" at worst. https://github.com/riskable/void_switch

QMK Desktop app

QMK has a feature called raw HID: https://docs.qmk.fm/#/feature_rawhid which allows your PC to talk directly with the QMK firmware. ***Theoretically***, this would allow your computer to:

- Know the current QMK layer,
- Change the layer to better suit the current application (Media controls for VLC, gaming mode when gaming, etc.)
- Control keyboard's RGB lighting (brightness and tint based on time of day, gaming etc.)
- Sync the keyboard layout with the computer's layout (QWERTY vs Colemak, etc.)
- Edit and extend onboard Macros
- Type special characters like é, θ, or ° with a single keystroke

Unfortunately, while small tech demos have been written to prove some of these capabilities, a proper companion app for QMK has never been developed.

Custom Keycap Legends

With the correct combination of blank PBT keycaps, sharpie ink, and laser cutter settings, it is possible to create your own custom keycap legends. You'll need to experiment with different laser cutter settings, so make sure to buy extra keycaps. https://www.reddit.com/kon407/