

Hand-Wiring Guide

Parts list

You will need: (where x is the number of keys on your planned keyboard)

- QMK compatible microcontroller board (Teensy, Pro-Micro, QMK Proton C etc.)
- x keyswitches (MX, Matias, Gateron, etc)
- x through hole diodes
- Keyboard plate and plate mount stabilisers
- Wire
- Soldering iron
- Rosin-cored solder
- Adequate ventilation/a fan
- Wire cutters/snippers

Optional but useful:

- Wire strippers/a sharp knife
- Tweezers and/or small needle nose pliers
- Soldering station/Helping hands

Starting the build

There are many ways to hand wire a PCB matrix, this guide will describe the fundamentals as well as some recommended ways to go about it.

As we are dealing with hand wiring, it is assumed that you already have a plate. If you are planning a completely custom layout, tools such as [ai03 Plate Generator](#) ↗ and [Swillkb Plate & Case Builder](#) ↗ can help when designing one.

Start by installing the switches and stabilisers in the plate. Depending on the thickness and material this may also involve hot gluing it in place.

Planning the matrix

If you are following a pre-existing handwire guide (e.g. for the keyboards in the [handwire firmware section](#) ↗ you can skip this step, just ensure you wire the matrix as described.

What you want to achieve is one leg from each switch being attached to the corresponding switches next to it (rows) and the other leg being attached to the switches above and below it (columns) and a diode to one of the legs, mosy commonly this will be the leg attached to the rows, and the diode will face away from it (Column to Row) i.e. with the wire furthest from the black line on the diode connected to the switch (as current will only travel in one direction through a diode).

It is fairly simple to plan for an ortholinear keyboard (like a Planck).

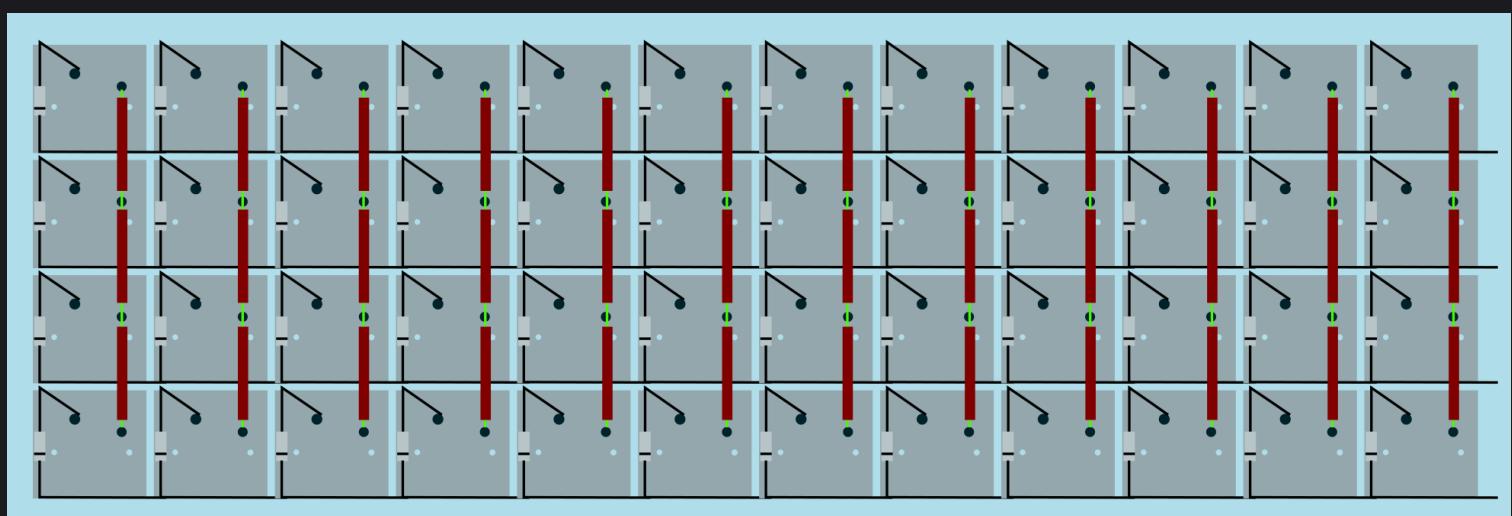
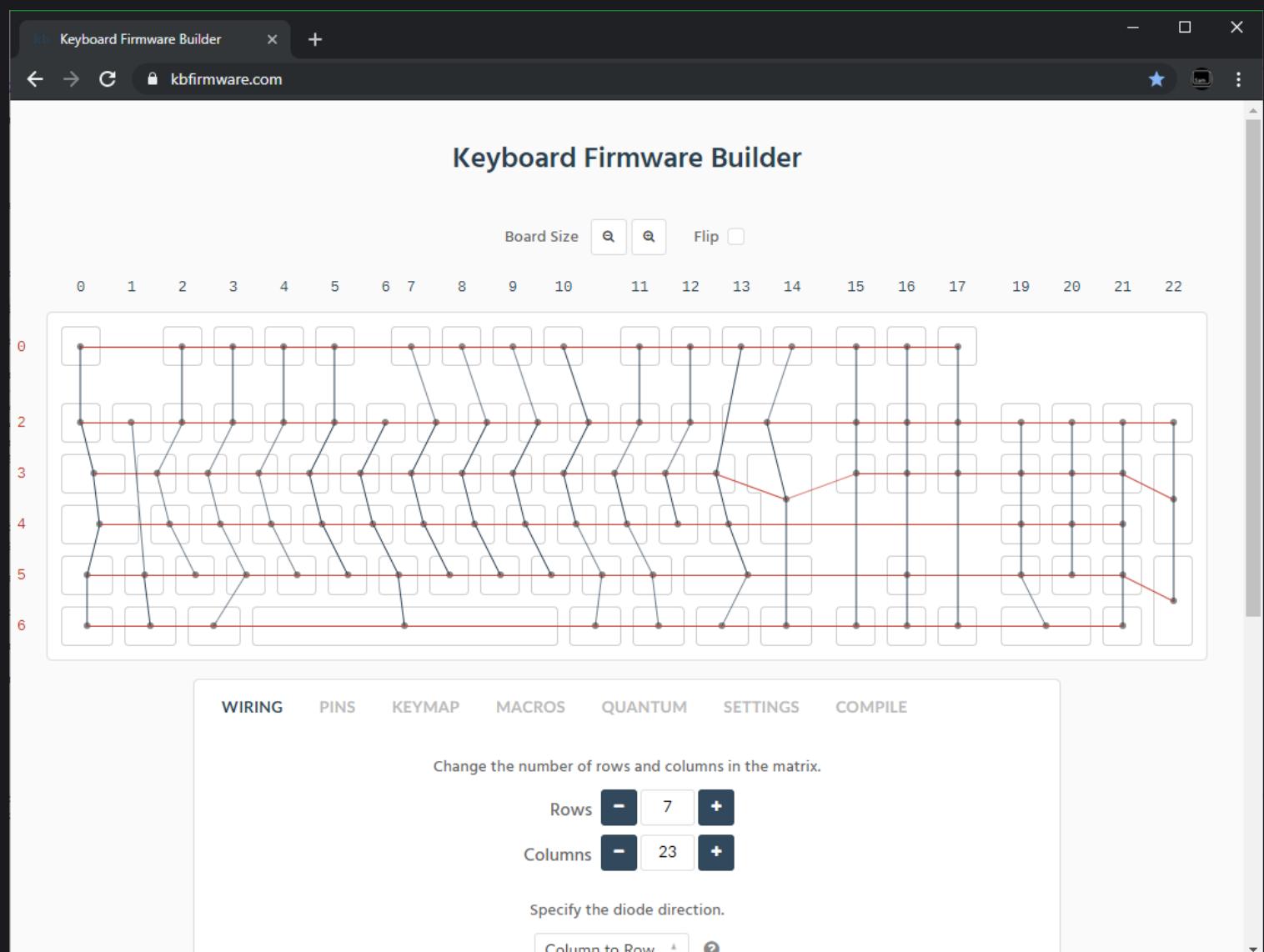


Image from [RoastPotatoes' "How to hand wire a Planck"](#)

But the larger and more complicated your keyboard, the more complex the matrix. [Keyboard Firmware Builder](#) can help you plan your matrix layout (shown here with a basic fullsize ISO keyboard imported from [Keyboard Layout Editor](#)).



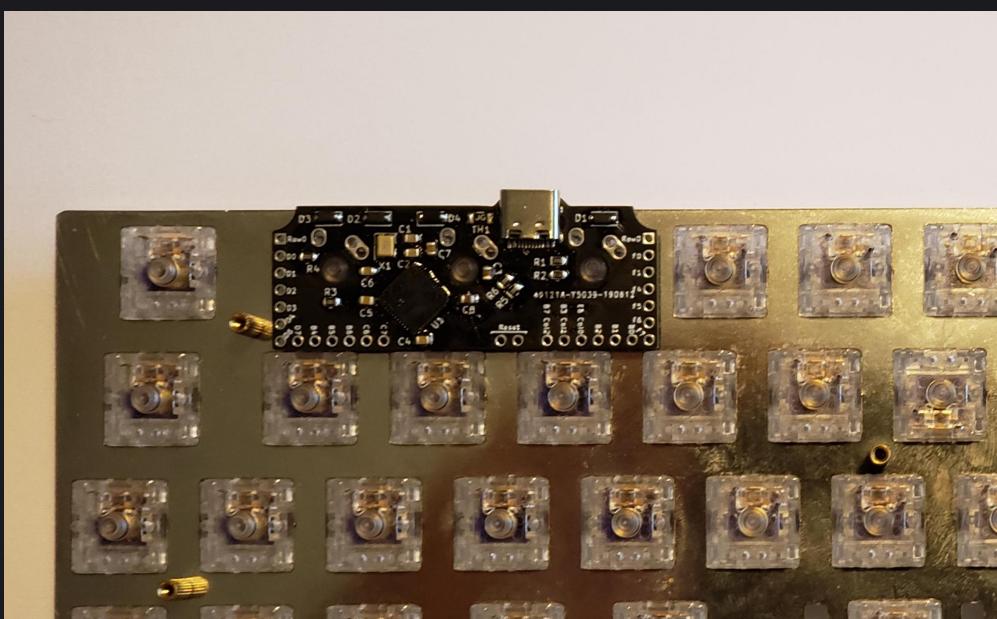
Bear in mind that the number of rows plus the number of columns can not exceed the number of I/O pins on your controller. So the fullsize matrix shown above would be possible on a Proton C or Teensy++, but not on a regular Teensy or Pro Micro.

Common Microcontroller Boards

Board	Controller	# I/O	Pinout
Pro Micro*	ATmega32u4	20	link ↗
Teensy 2.0	ATmega32u4	25	link ↗
QMK Proton C ↗	STM32F303xC	36	link 1 ↗, 2 ↗
Teensy++ 2.0	AT90USB1286	46	link ↗

*Elite C is essentially the same as a Pro Micro with a USB-C instead of Micro-USB

There are also a number of boards designed specifically for handwiring that mount directly to a small number of switches and offer pinouts for the rest. Though these are generally more expensive and may be more difficult to get hold of.



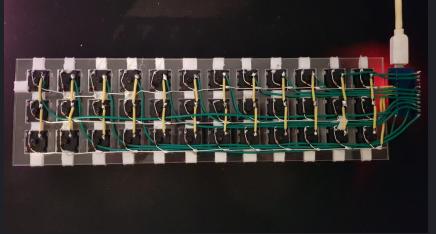
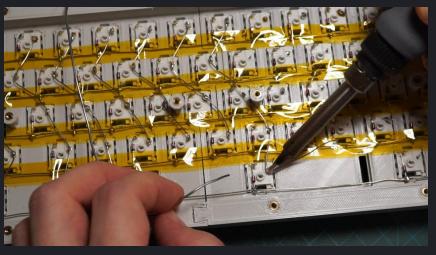
Board	Controller	# I/O
Swiss helper ↗	ATmega32u4	20
Postage board ↗	ATmega32u4	25
Postage board mini ↗	ATmega32u4	25

Wiring the matrix

There is no one right way to do this. What you want to achieve is good connection at all of the joints planned and no unintentional shorts.

Established materials and techniques include:

Technique	Examples	Pros	Cons	Image
Lengths of wire with stripped segments	Sasha Solomon's Dactyl ↗ and Cribbit's modern hand wire ↗	Neat and tidy	Some effort in stripping the wire	

Technique	Examples	Pros	Cons	Image
Short lengths of wire	u/xicolinguada's ortho build ↗	Easier to strip the wire	More difficult to place	
Magnet/Enamelled wire	fknraiden's custom board ↗	Can be directly soldered onto (insulation burns off with heat)	Appearance?	
Bending the legs of the diodes for the rows	Matt3o's Brownfox ↗	Fewer solder joints required	Uninsulated	
Using rigid wiring (e.g. brass tube)	u/d_stilgar's invisible hardline ↗ and u/jonasfasler's first attempt ↗	Very pretty	More difficult. No physical insulation	
Bare wire with insulation added after (e.g. kapton tape)	Matt3o's 65% on his website ↗	Easier (no wire stripping required)	Not as attractive	
Copper tape	ManuForm Dactyl ↗	Very easy	Only really works when your plate/case aligns with the bottom of your switches	

Note that these methods can be combined. Prepare your lengths of wire before moving on to soldering.

A note on split keyboards

If you are planning a split keyboard (e.g. Dactyl) each half will require a controller and a means of communicating between them (like a TRRS or hardwired cable). Further information can be found in the [QMK split keyboard documentation](#).

Soldering

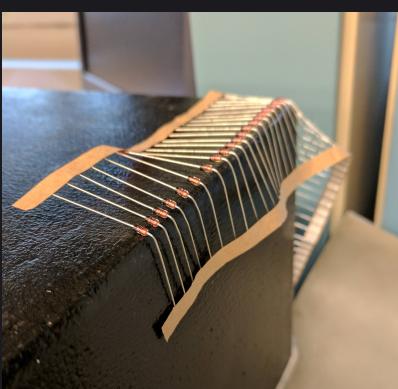
There are a lot of soldering guides and tips available elsewhere but here are some of the most

useful and relevant for hand wiring:

To ensure a strong solder joint you want a good amount of contact between the solder and the two pieces of metal you are connecting. A good way of doing this (though not required) is looping around pins or twisting wires together before applying solder.



If your diodes are on a packaging strip and need a bend in them (either the start of a loop or for connecting to its neighbour) this can easily done by bending it over something straight like the edge of a box, table, or ruler. This also helps keep track of the direction of the diode as all the bends will be on the same side.



If your iron has temperature control, set it to 315°C (600°F).

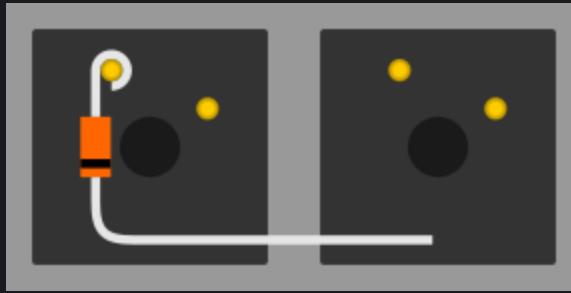
Once heated, tin your soldering iron - this means melting a small amount of solder on the end of the iron and then quickly wiping it off on a wet sponge or wire cleaning pad, leaving a shiny silvery coating on the end which helps keep oxidisation at bay and helps solder to flow.

When you come to apply the solder, hold the soldering iron against the two surfaces for a second to heat it, then apply a small amount of solder to join the two pieces together. Heating the surfaces ensures that the solder adheres to it and that it does not cool too quickly.

Don't hold the iron on the solder/joint longer than necessary. Heat will be conducted through the surfaces and can damage components (melt switch housings etc.). Also, solder contains flux, which aids in wetting. The longer heat is applied to the solder the more flux will evaporate meaning you may end up with a bad solder joint with peaks which, apart from looking bad, may also increase the risk of electrical shorts.

Soldering the Diodes

Starting at the top-left switch, place the diode (with tweezers if you have them) on the switch so that the diode itself is vertically aligned, and the black line is facing toward you. Make sure the diodes are soldered in parallel (diode outputs shouldn't connect to diode inputs). The input lead of the diode should be touching the left contact on the switch, and the bent, output end should be facing to the right and resting on the switch there, like this:

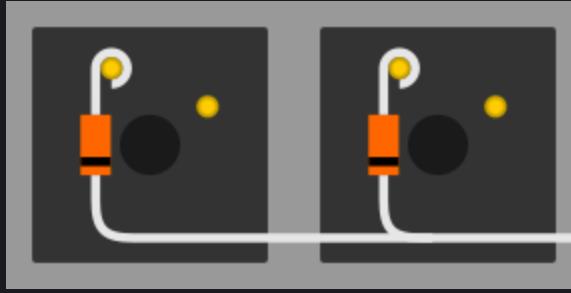


Letting the diode rest, grab your solder, and touch both it and the soldering iron to the left contact at the same time - the rosin in the solder should make it easy for the solder to flow over both the diode and the keyswitch contact. The diode may move a little, and if it does, carefully position it back in place by grabbing the bent end of the diode - the other end will become hot very quickly. If you find that it's moving too much, using needle-nose pliers of some sort may help to keep the diode still when soldering.

The smoke that the rosin releases is harmful, so be careful not to breath it or get it in your eyes/face.

After soldering things in place, it may be helpful to blow on the joint to push the smoke away from your face, and cool the solder quicker. You should see the solder develop a matte (not shiny) surface as it solidifies. Keep in mind that it will still be very hot afterwards, and will take a couple minutes to be cool to touch. Blowing on it will accelerate this process.

When the first diode is complete, the next one will need to be soldered to both the keyswitch, and the previous diode at the new elbow. That will look something like this:



After completing a row, use the wire cutters to trim the excess wire from the tops of the diodes, and from the right side on the final switch. This process will need to be completed for each row you have.

When all of the diodes are completely soldered, it's a good idea to quickly inspect each one to ensure that your solder joints are solid and sturdy - repairing things after this is possible, but more difficult.

Soldering the Columns

You'll have some options in the next process - it's a good idea to insulate the column wires (since the diodes aren't), but if you're careful enough, you can use exposed wires for the columns - it's not recommended, though. If you're using single-cored wire, stripping the plastic off of the whole wire and feeding it back on is probably the best option, but can be difficult depending on the size and materials. You'll want to leave parts of the wire exposed where you're going to be solder it onto the keyswitch.

If you're using stranded wire, it's probably easiest to just use a lot of small wires to connect each keyswitch along the column. It's possible to use one and melt through the insulation, but this isn't recommended, will produce even more harmful fumes, and can ruin your soldering iron.

Before beginning to solder, it helps to have your wire pre-bent (if using single-cored), or at least have an idea of how you're going to route the column (especially if you're making a staggered board). Where you go in particular doesn't matter too much, as we'll be basing our keymap definitions on how it was wired - just make sure every key in a particular row is in a unique column, and that they're in order from left to right.

If you're not using any insulation, you can try to keep the column wires elevated, and solder them near the tips of the keyswitch contacts - if the wires are sturdy enough, they won't short out to the row wiring an diodes.

Wiring up the controller

Now that the matrix itself is complete, it's time to connect what you've done to the microcontroller board.

Place the microcontroller where you want it to be located, give thought to mounting and case alignment. Bear in mind that the location of the USB socket can be different from the controller by using a short male to female cable if required.,

Find the pinout/documentation for your microcontroller board ([links here](#)) and make a note of all the digital I/O pins on it (note that on some controllers, like the teensy, analogue I/O can double as digital) as these are the pins you want to connect your wires to.

Specific instructions for the Teensy 2.0

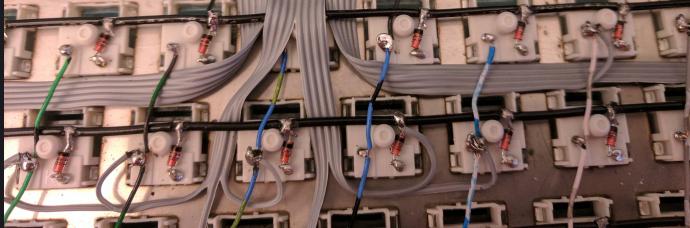
There are some pins on the Teensy that are special, like D6 (the LED on the chip), or some of the UART, SPI, I2C, or PWM channels, but only avoid those if you're planning something in addition to a keyboard. If you're unsure about wanting to add something later, you should have enough pins in total to avoid a couple.

The pins you'll absolutely have to avoid, as with any controller, are: GND, VCC, AREF, and RST - all the others are usable and accessible in the firmware.

Cut wires to the length of the distance from the a point on each column/row to the controller. You can solder anywhere along the row, as long as it's after the diode - soldering before the diode (on the keyswitch side) will cause that row not to work.

Ribbon cable can be used to keep this extra tidy. You may also want to consider routing the wires beneath the exisiting columns/rows.





As you solder the wires to the controller make a note of which row/column is going to which pin on the controller as we'll use this data to setup the matrix when we create the firmware.

As you move along, be sure that the controller is staying in place - recutting and soldering the wires is a pain!

Getting Some Basic Firmware Set Up

From here, you should have a working keyboard once you program a firmware.

Simple firmware can be created easily using the [Keyboard Firmware Builder](#) website. Recreate your layout using [Keyboard Layout Editor](#), import it and recreate the matrix (if not already done as part of [planning the matrix](#)).

Go through the rest of the tabs, assigning keys until you get to the last one where you can compile and download your firmware. The .hex file can be flashed straight onto your keyboard, or for advanced functionality, compiled locally after [Setting up Your Environment](#).

The source given by Keyboard Firmware Builder is QMK, but is based on a version of QMK from early 2017. To compile the firmware in a modern version of QMK Firmware, you'll need to export via the `Save Configuration` button, then run:

```
qmk import-kbfirmware /path/to/export.json
```

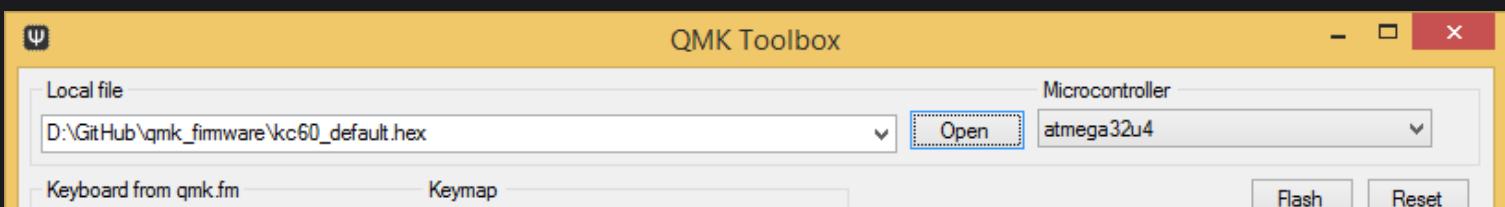
For example:

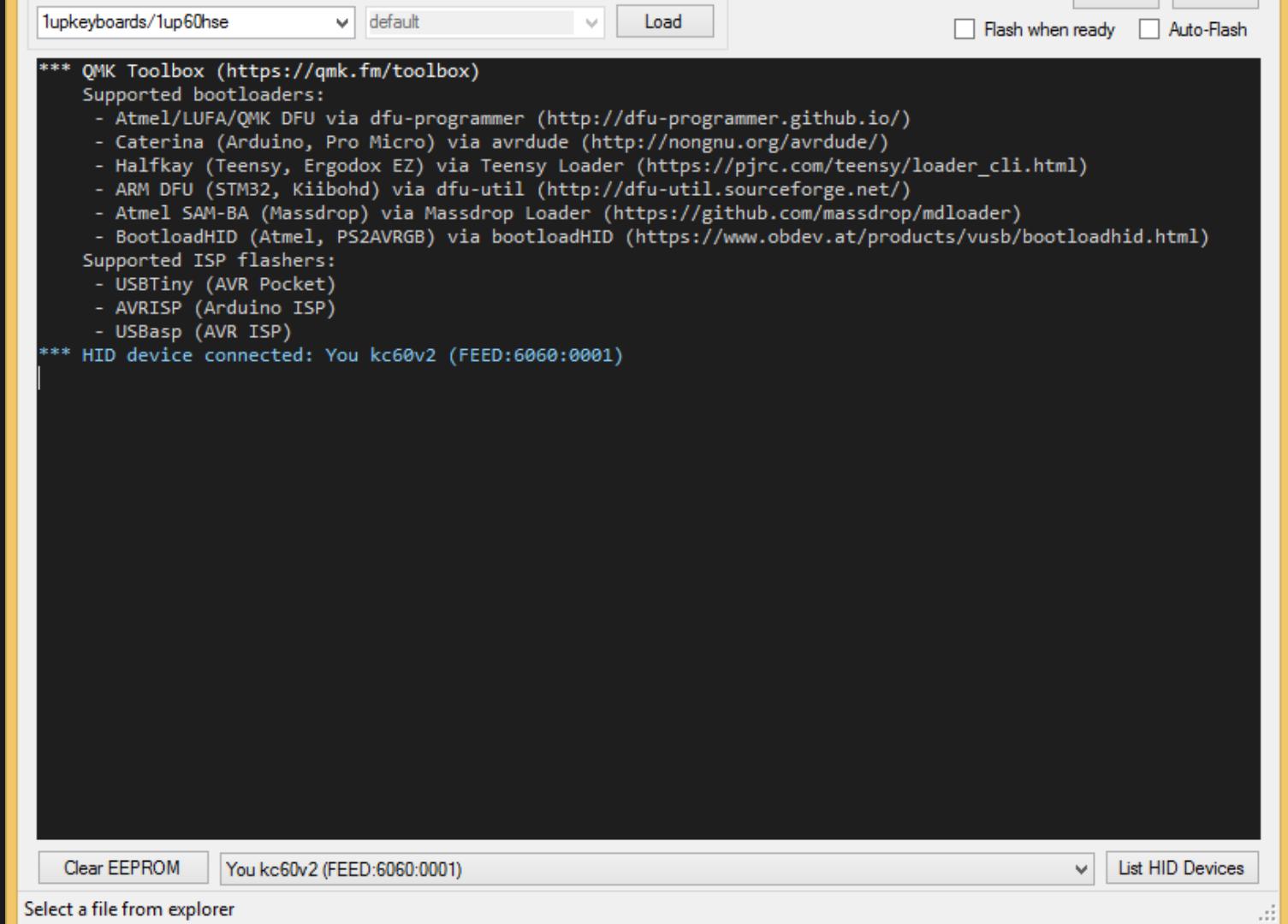
```
$ qmk import-kbfirmware ~/Downloads/gh62.json
Ψ Importing gh62.json.

⚠ Support here is basic - Consider using 'qmk new-keyboard' instead
Ψ Imported a new keyboard named gh62.
Ψ To start working on things, `cd` into keyboards/gh62,
Ψ or open the directory in your preferred text editor.
Ψ And build with qmk compile -kb gh62 -km default.
```

Flashing the Firmware

Install [QMK Toolbox](#).





Under "Local File" navigate to your newly created .hex file. Under "Microcontroller", select the corresponding one for your controller board (common ones available [here](#)).

Plug in your keyboard and press the reset button (or short the Reset and Ground pins if there is no button) and click the "Flash" button in QMK toolbox.

Testing Your Firmware

Use a website such as [QMK Configurator's Keyboard Tester](#), [Keyboard Tester](#), or [Keyboard Checker](#) or just open a text editor and try typing - you should get the characters that you put into your keymap. Test each key, and make a note of the ones that aren't working. Here's a quick trouble-shooting guide for non-working keys:

1. Flip the keyboard back over and short the keyswitch's contacts with a piece wire - this will eliminate the possibility of the keyswitch being bad and needing to be replaced.
2. Check the solder points on the keyswitch - these need to be plump and whole. If you touch it with a moderate amount of force and it comes apart, it's not strong enough.
3. Check the solder joints on the diode - if the diode is loose, part of your row may register, while the other may not.
4. Check the solder joints on the columns - if your column wiring is loose, part or all of the column may not work.
5. Check the solder joints on both sides of the wires going to/from the Teensy - the wires need to be fully soldered and connect to both sides.
6. Check the `<project_name>.h` file for errors and incorrectly placed `KC_NOS`s - if you're unsure where they should be, instead duplicate a kxy variable.
7. Check to make sure you actually compiled the firmware and flashed the Teensy correctly. Unless you got error messages in the terminal, or a pop-up during flashing, you probably did everything

correctly.

8. Use a multimeter to check that the switch is actually closing when actuated (completing the circuit when pressed down).

If you've done all of these things, keep in mind that sometimes you might have had multiple things affecting the keyswitch, so it doesn't hurt to test the keyswitch by shorting it out at the end.

Finishing up

Once you have confirmed that the keyboard is working, if you have used a separate (non handwire specific) controller you will want to secure it in place. This can be done in many different ways e.g. hot glue, double sided sticky tape, 3D printed caddy, electrical tape.

If you found this fulfilling you could experiment by adding additional features such as [in switch LEDs](#), [in switch RGB](#), [RGB underglow](#) or even an [OLED display!](#)

There are a lot of possibilities inside the firmware - explore [docs.qmk.fm](#) for a full feature list, and dive into the different keyboards to see how people use all of them. You can always stop by [the OLKB subreddit](#) or [QMK Discord](#) for help!

Links to Other Guides

- [matt3o's step by step guide \(BrownFox build\)](#) also his [website](#) and [video guide](#)
- [Cribbit's "Modern hand wiring guide - stronger, cleaner, easier"](#)
- [Sasha Solomon's "Building my first Keyboard"](#)
- [RoastPotatoes' "How to hand wire a Planck"](#)
- [Masterzen's "Handwired keyboard build log"](#)

Legacy Content

This page used to include more content. We have moved a section that used to be part of this page its own page. Everything below this point is simply a redirect so that people following old links on the web find what they're looking for.

Preamble: How a Keyboard Matrix Works (and why we need diodes)

- [How a Keyboard Matrix Works](#)

 [Edit this page](#)

