Keyboard Firmware

All the code.

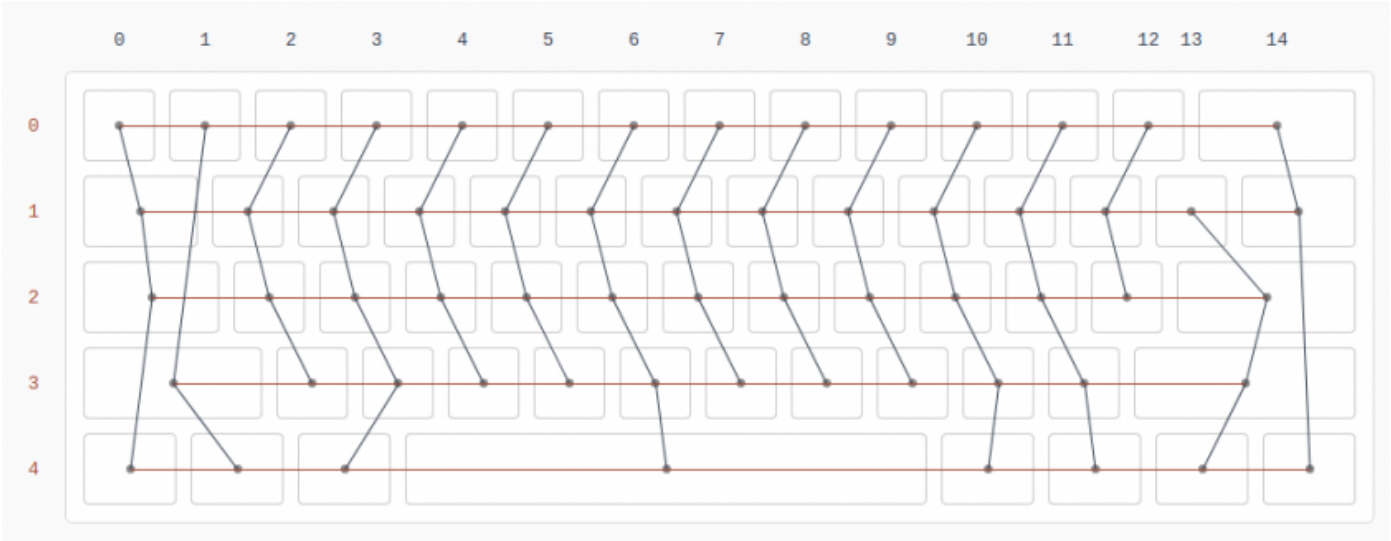
- Using Kbfirmware.com

Using Kbfirmware.com

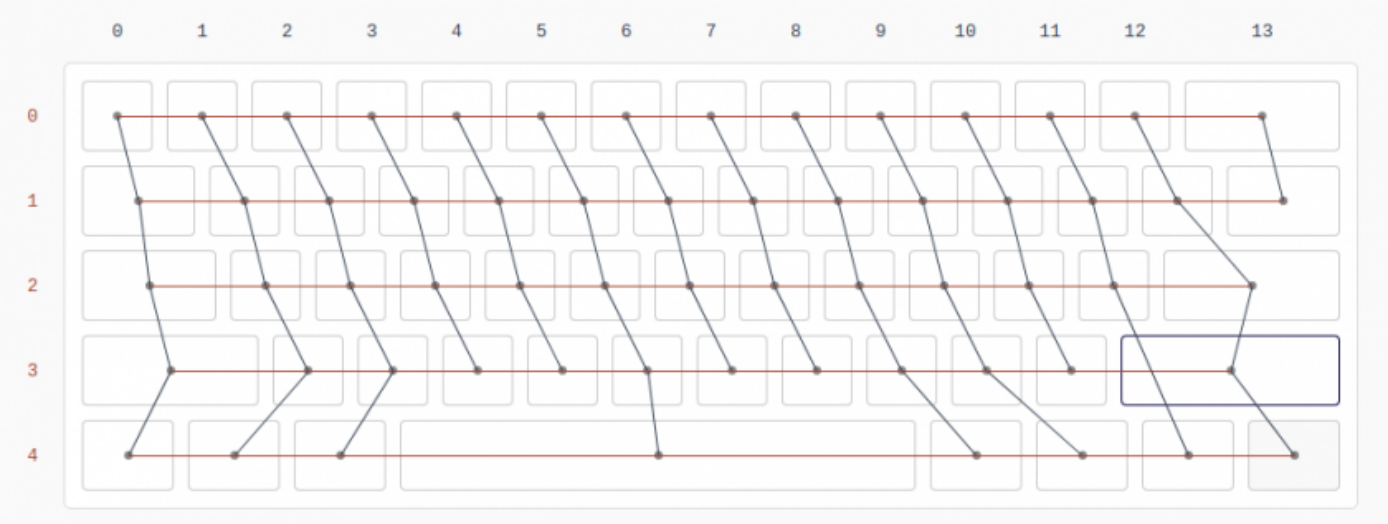
If you are using a premade pcb with a .json file provided such as a dz60 etc. Please load that .json file into kbfirmware.com and skip to the Keymap and Macro tips and tricks.

Basics of creating your matrix:

One of the easiest ways to do this is to simply import your KLE raw data into kbfirmware.com and have the matrix generated for you. This generally works well enough, but can sometimes be improved upon as it does not always use the smallest number of pins. My preferred method of creating my matrix is to take a screenshot of my KLE layout and simply draw the rows and columns in over the screenshot. I like doing this because I generally save a column or two compared to kbfirmware.com, and can also make some adjustments to make hand wiring easier.



Default wiring diagram for 60% layout from Keyboard-layout-editor.com via the Kbfirmware.com algorithm



Example of wiring diagram made for “easier” hand wiring and saves 1 pin

What pins are and why they are important:

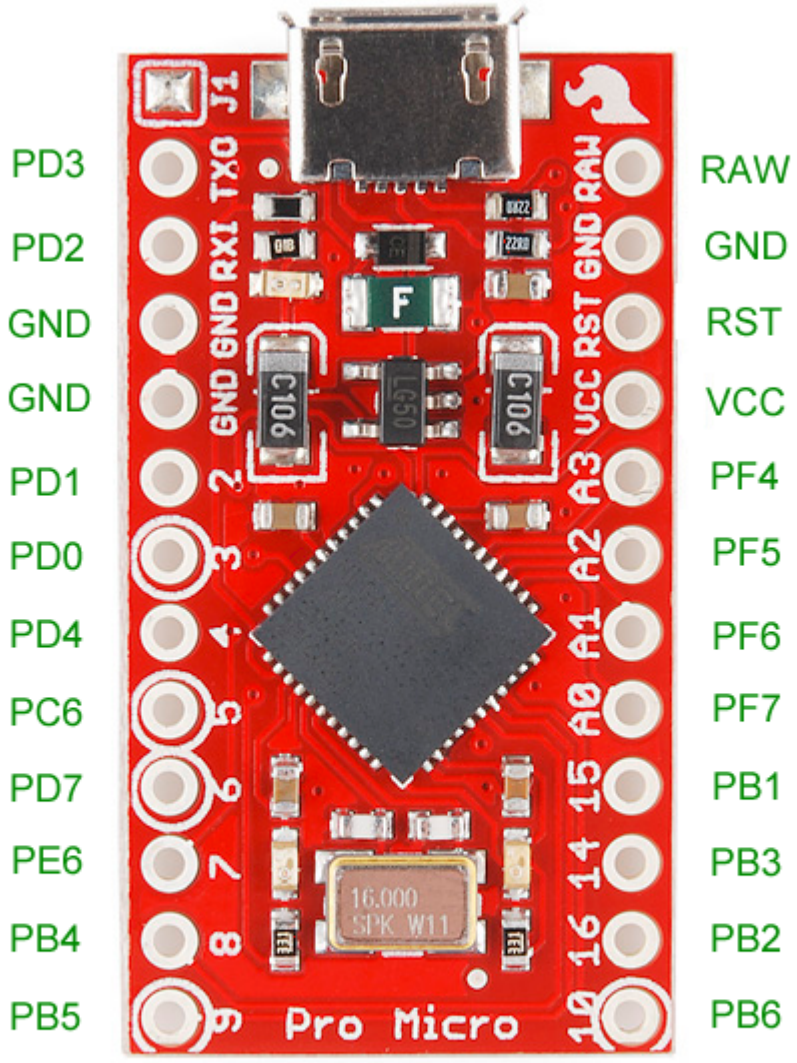
Pins in this context are the number of usable digital I/O outputs on the controllers. Pins are important because they dictate the number of rows and columns you can have in a layout. For every row and column one pin needs to be used per each. So in the examples above 19 (5 rows, 14 columns) and 20 (5 rows, 15 columns) pins are used respectively. The two most popular controller options for handwire and basic custom pcbs are the pro micro and the teensy 2.0 which have 18, and 25 usable pins respectively. What this all boils down to is that having 18 or less pins saves you money, and having 26 or more pins means that you will need to either user some rather involved wiring shenanigans to reduce your pin use, or jump up in price again to the teensy ++ which has 46 usable pins, but is more expensive and not used very much.

Editing your wiring diagrams:

This is pretty straight forward on kbfirmware.com. You simply have to select the key and use the + and - buttons to change the rows or columns. The diode direction should always be “Column to Row” if you are using most pcbs or following all hand-wiring guides I have read (I use a mix of techniques in the “A Modern Handwiring Guide” and “BrownFox step by step”).

Pinout Adjustments:

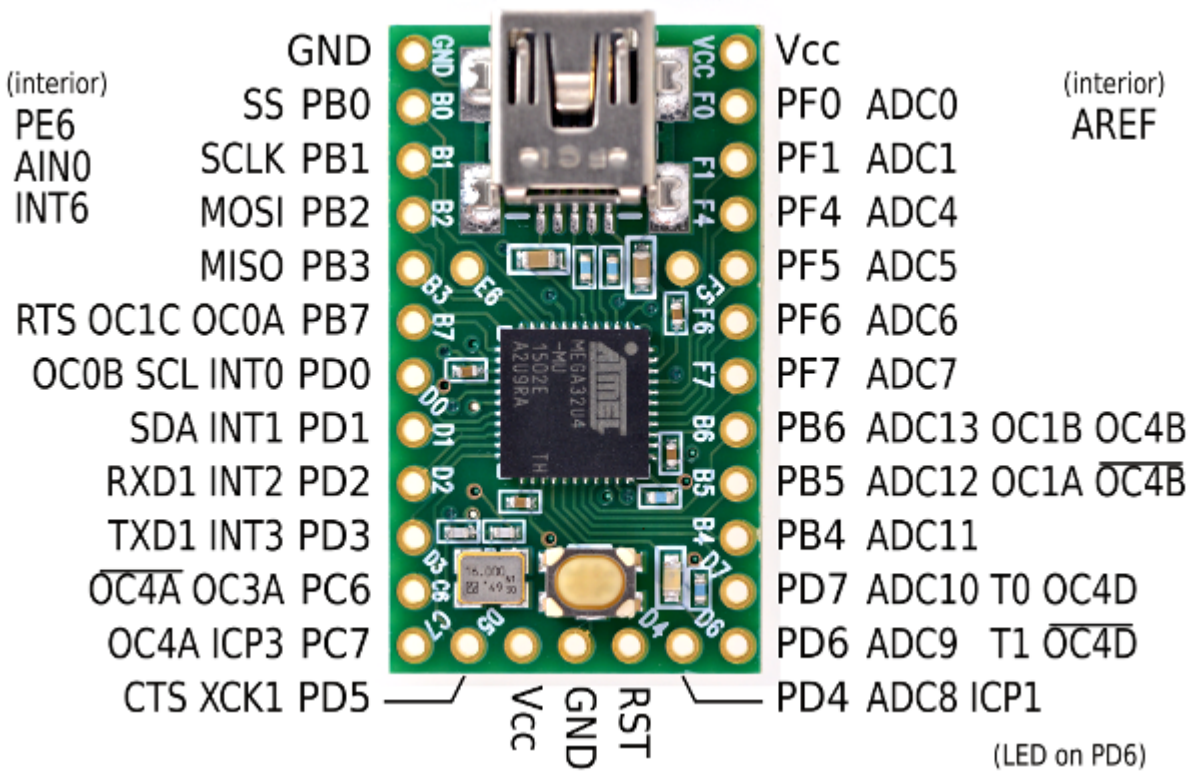
If you are using a promicro or teensy controller the ATmega32U4 will be correct setting, if you are using a different controller this might change, but most common ones used by keyboards use the ATmega32U4. For handwire builds I prefer to connect my rows and columns to the controller in a way that makes the length of wire the shortest/most direct. This part of the process is very important to document. After I connect the wires to the controller and document it, you just have to go through and use the drop down menu to change the rows and columns to their proper pin.



LEDs on PB0 (yellow) and PD5 (green), active low

RAW: power supply when not connected to USB, up to 16V DC

Pro Micro pinout



Teensy pinout

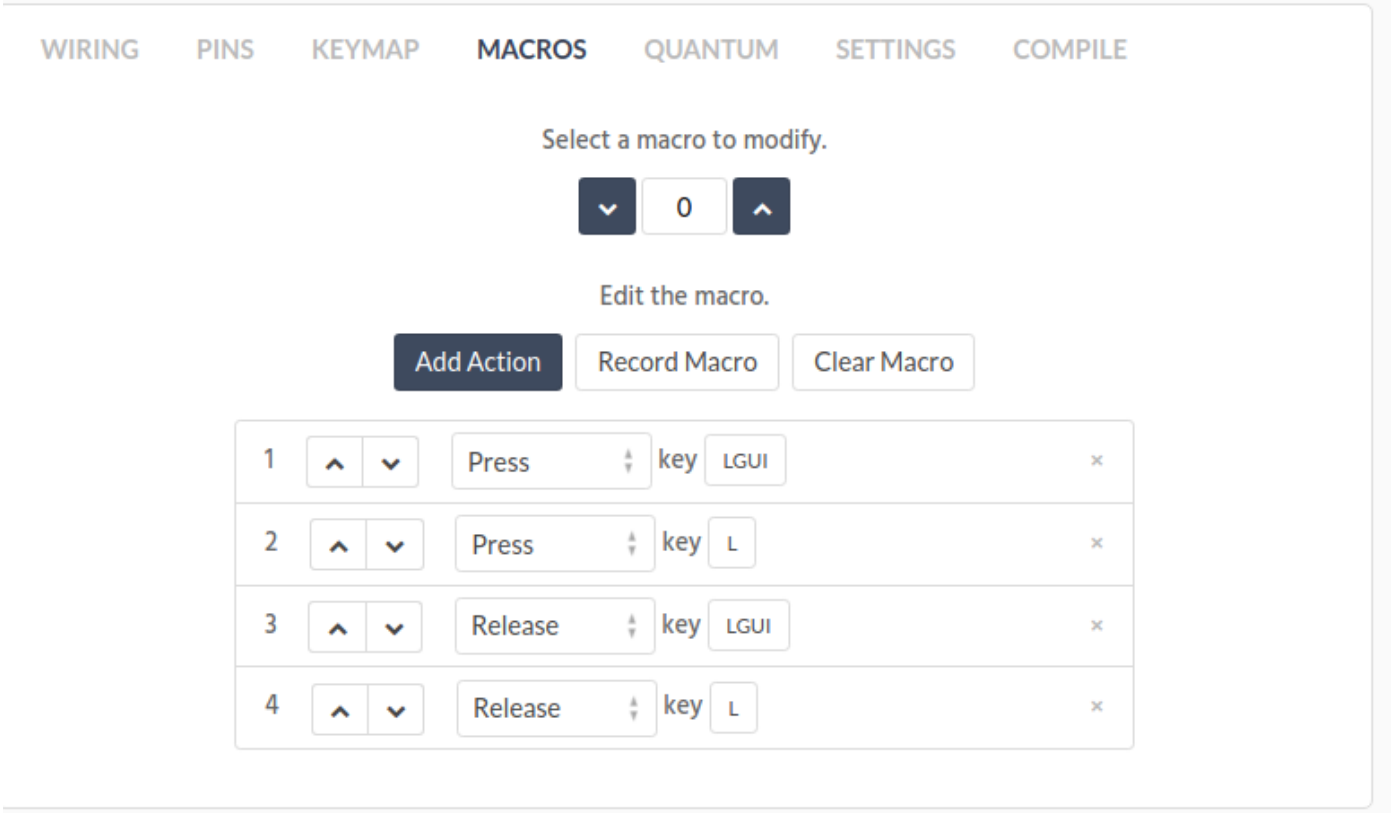
Keymap tips and tricks:

Creating a basic keymap is pretty straight forward when using kbfirmware.com, just select the layer (0 is the default/base layer) and then select the key you want to change and then you can click through the various options to find what you are looking for. If you are wondering what some of these weird looking codes are here: https://docs.qmk.fm/keycodes.html is pretty much all of them available on kbfirmware.com and some more that can be used to do some pretty advanced things.

Layers for QMK can be somewhat tricky to understand, but the basics are fairly simple and I find that the resource here: https://docs.qmk.fm/feature_advanced_keycodes.html#switching-and-toggling-layers helps most people be able to do what they are looking to do.

Macro tips and tricks:

The website makes it pretty dang easy to make your own macros and have them put on your board. You can add steps yourself or use the record function which is nice for making macros that type things such as “Omae wa mou shindeiru”. If your macro is a bit more complex or runs into issues with the record function you can use the add action button to step by step make your macro. A simple macro to lock your computer (GUI key + L) is shown below.



And after you have made your first macro, make sure to change the macro you are modifying with the arrows above.

When you are adding macros to your keymap, you go to the “FN” tab and select the M() option and it will have you select which macro you want to be bound to that button. BAM you made some simple macros.

Guide written by /u/friglesnart or friglesnart#1931

Feel free to contact me with any questions, comments, or suggestions. Thank you!