Advanced Info

The real work.

- Error-Checking
- Microcontroller Design
- Matrices and Duplex Matrix
- Routing Techniques
- Backlighting And RGB
- USB Type-C
- Good USB Routing Practices
- EMI Shielding and Reduction
- Breakaway Tabs

Error-Checking

Produce functional PCBs the first time around.

DRC Checks

Begin here.

- Run a DRC. Does it report anything?
- Are there any errors aside from overlapping stabilizer/switch pads?

Physical Checks

These errors are easy to miss, since some are not scanned by DRC.

- Is the PCB the correct size?
- Is USB connector placed at its proper location?
- Are there any components so large that they will interfere with the chassis?
- Are the keys placed in correct layout?
- Are there overlapping parts?

Trace Checks

Correct routing is important.

- Do any traces overlap each other?
- Do any traces go through holes and/or pads?
- Do any traces fall off the PCB edge?
- Are there any pointless disconnected traces?
- Do any traces go so close to pads that they enter their no-mask zone?
    - This can result in shorted circuits after soldering, even if they pass an unsoldered electrical check.

Component Checks

Will the parts function?

- Is the pinout of the components correct?
  - Specifically, is the crystal, MCU, and USB connector pinouts correct?
    - Often, on a 4-pad crystal, the crystal pads and gnd pads are placed diagonally from each other.
    - If you have two adjacent same nets or an unconnected pad on crystal, you may have incorrect pinout!
  - Even identical footprint components sometimes have differing pinout.
  - Check the documentation for the parts for their correct pinout.
- Are you using the correct parts for the task?
- Are you using the correct footprints?
- Will the placement of parts cause nearby parts to fail?
  - Heat, EMI, etc.
- Are decoupling capacitors placed at their correct positions?
  - Near VCC/GND MCU pairs, close to crystal, close to RGB LED, etc.

Cross-Checking

More eyes is better.

- Load the gerber in a gerber viewer. Does it render as expected?
- Let peers error-check your project. Do they find anything suspicious?
- Does the PCB avoid pitfalls described in this wiki?

Aesthetic Checks

For beauty-oriented PCBs.

- Are there any component label silkscreen components that overlap pads or holes?
- Do any labels overlap other labels?
- Do silkscreen decorations or branding cover up critical component labels?

Microcontroller Design

For successful and elegant controller circuitry.

Microcontroller Basics

Controller such as ATMEGA32U4 are often used as low-cost, low-power processing units for mechanical keyboard PCBs.
They often require various components of their own in order to function properly.

Choosing the Correct Controller

Space and function constraints will require different choices of controllers.

ATMEGA Series	Pros	Cons
atmega32u2	• Small form factor. • Powerful enough for most smaller keyboards.	• Not enough pins for large matrices. • Not much lower price than 32u4. • No I2C support.
atmega32u4	• 25 I/O pins, slightly larger than 32u2. • Enough pins for most keyboard matrices.	• Slightly increased size, and more required components. • Slightly more expensive than 32u2.
at90usb1286	• 40+ I/O pins. Handles the largest matrices just fine.	• Extremely large size. • Very expensive.
atxmega32a4u	• Same size as 32u4, but requires no crystal. • No crystal means less chance of failure. • Overall small footprint. • Many I/O pins. • Two separate I2C busses.	• Not QMK compatible. ◦ Runs fine on Keyplus. • Requires 3.3v voltage regulator.

When in doubt, pick 32u2 for sub-60%, 32u4 for 60-80% and splits, and 1286 for 80+% keyboards.
When using 32u2, it is strongly recommended to bind B7 to PWM backlight, since it is the only pin known to work with QMK flawlessly.

Controller Layout - The Basics

- Place the controller and its components in a position that allows for proper placement and breakout.
- All decoupling capacitors must be placed as close to VCC/GND pin pairs on the controllers as possible. (Detailed below)
- Break out other pins with even spacing and pattern in order to minimize wasted space.
- Try to keep "dirty ground current", such as from RGB LEDs or backlighting, away from the controller. For example, run a separate ground plane for the controller, and join it together with the rest of the ground close to the USB port.

Crystal Layout

General

- The crystal is one of the most sensitive parts of the controller circuit.
- Place the crystal in a well-shielded position away from other traces.
- Do not run any traces close to, through, or under the crystal area unless absolutely necessary.
- Crystal and its decoupling capacitors should be fairly close to the controller.
- Crystal layout should be relatively symmetrical.
- "Dirty ground current" should never run through the crystal.
- Do not use vias in crystal traces, since they mess with the fine capacitance requirements.
- Surround the zone with a ground plane.

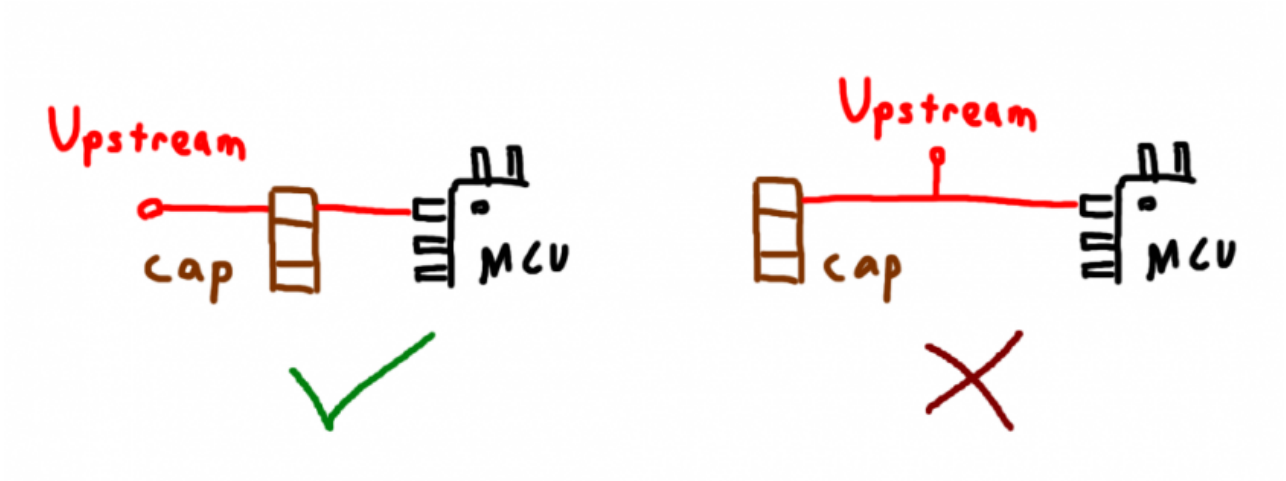
Crystal Load Capacitors

- Recommended load capacitor range is from 10pf to 33pf.
  - Atmel uses 22pf on their official documentation.
- For calculating exact capacitance for a crystal, use the following equation:
  Capacitance for each crystal = 2 * (Crystal Load Capacitance) - 2 * (Stray Capacitance)
- Crystal Load Capacitance can be found from the crystal datasheets.
- Stray capacitance comes from the traces of the board itself.
  - Usually it is around 2-5pf provided that the crystal traces are short, matched, and have no vias.

Controller Decoupling Capacitors

Placement is important

- The microcontroller decoupling capacitors exist to prevent the controller from unleashing plenty of noise into the VCC and GND channels.
  - Yes, a whole board can fail if these lines are affected enough.
- Place each decoupling capacitor as close to its respective VCC/GND pin pair as possible.
- Feed through the capacitor for them to have effect.



Choosing the right values

- Consult the official documentation and working prototypes for "correct" quantity and values.
- Official documentation will usually state 0.1uF per VCC pin and 10uF for UVCC pin.
- However, most common implementations will skip a 0.1uF to cut costs, and still run perfectly fine (i.e. Teensy 2.0, Teensy++ 2.0).

Regarding placement of the large capacitor

- Placing the largest decoupling capacitor far from the microcontroller makes it a power reservoir for the entire board.
- Placing the largest decoupling capacitor close to the microcontroller makes it a power reservoir for mainly the controller.
- Most sources will say that the largest decoupling capacitor should be placed close to the controller instead of near the USB port.
- It can be used as an "upstream filtering capacitor" by having all VCC current for the MCU travel across the capacitor pad before branching off to feed other VCC pins through their decoupling capacitors.

Sources

http://www.atmel.com/Images/Atmel-2521-AVR-Hardware-Design-Considerations_ApplicationNote_AVR042.pdf

https://blog.adafruit.com/2012/01/24/choosing-the-right-crystal-and-caps-for-your-design/

https://electronics.stackexchange.com/questions/20255/how-far-is-too-far-when-routing-traces-for-crystals-and-how-asymmetrical-is

http://www.atmel.com/Images/Atmel-8128-Best-Practices-for-the-PCB-Layout-of-Oscillators_ApplicationNote_AVR186.pdf

https://www.pjrc.com/teensy/schematic.html

https://www.silabs.com/documents/public/application-notes/AN0046.pdf

Matrices and Duplex Matrix

Improving upon their design.

Maximizing the matrix

- For non-macropad keyboards, routing a matrix with one pin per row and column will almost always lead to inefficient results.
- With the same number of keys, a more "square" matrix will require less cols and rows total than a more "rectangular" one.
  - I.E. For 30 keys, 5x6 is 11 pins, while 2x15 is 17 pins.

The Duplex Matrix

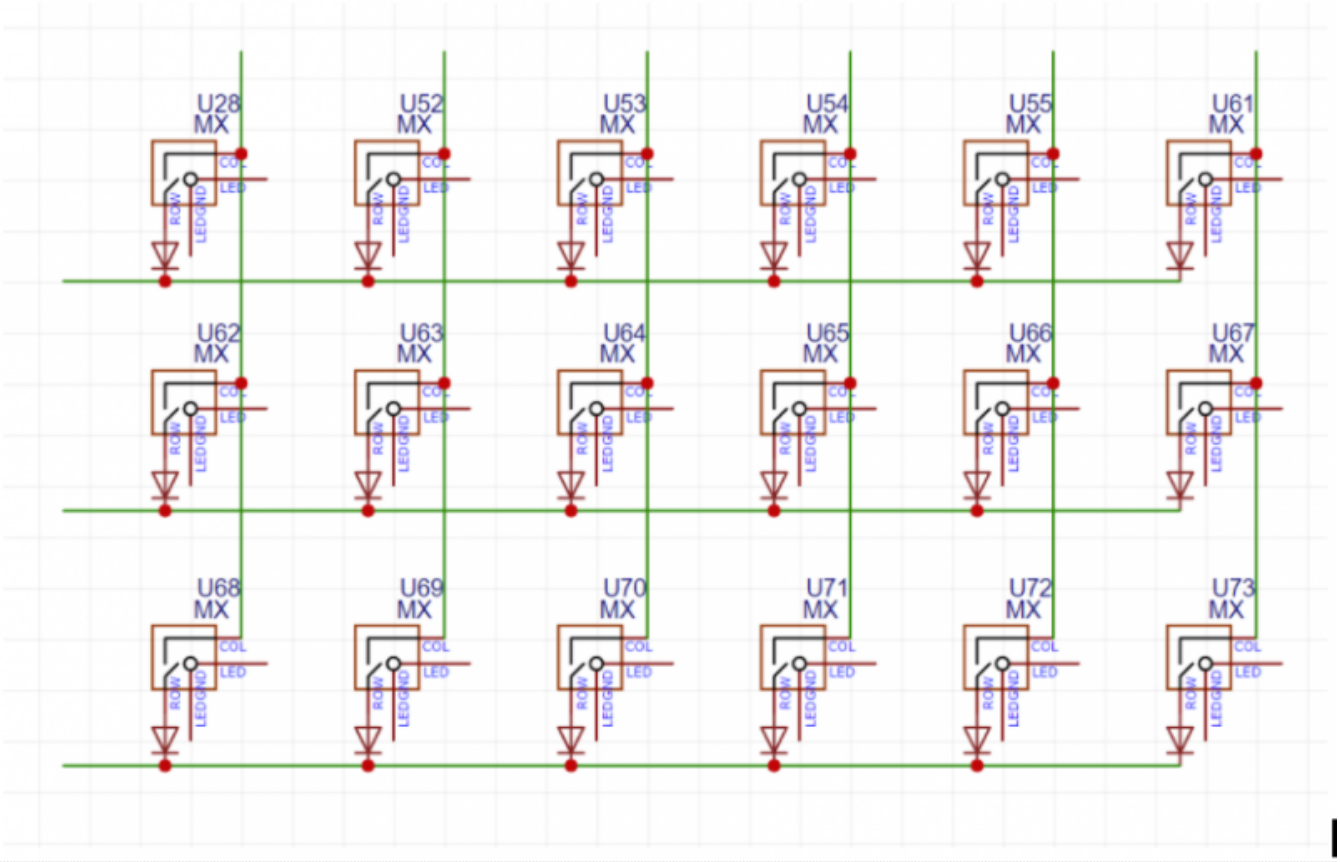
- Each "column" spans two physical columns.
- Two "rows" exist for each physical row.
- See schematic at bottom for a routing example.

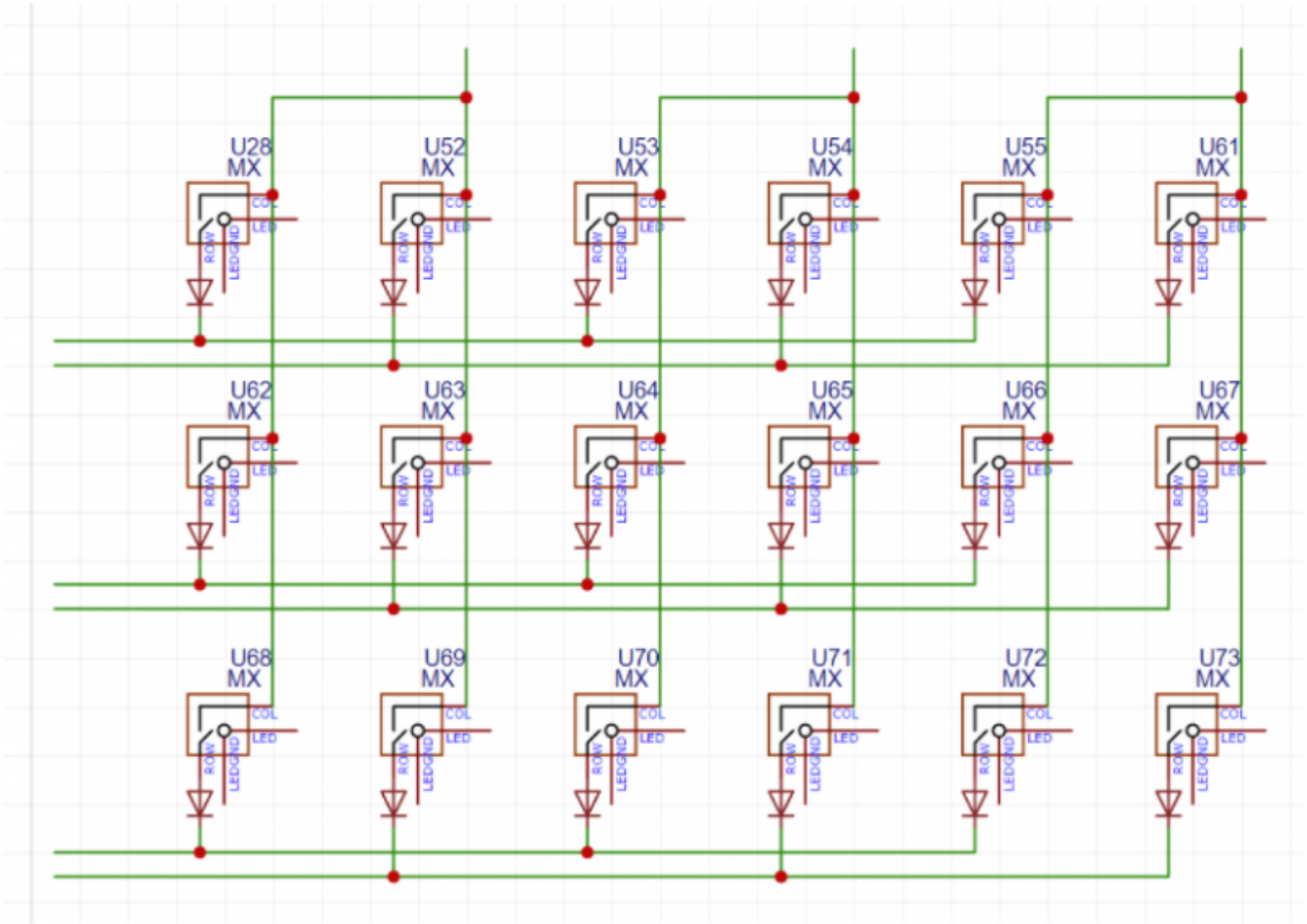
Pros

- Fit more keys for rectangular keyboards using less pins
- More pins to use for fancy features
- Use cheaper controllers to reduce the final product cost

Cons

- More complicated to wire
- Can be more difficult to add more keys to the matrix later on
- Harder to follow on a schematic

- Traditional Grid
- 

- Duplex Grid
- 

Routing Techniques

Tips for a proper, organized PCB

Good Routing Techniques

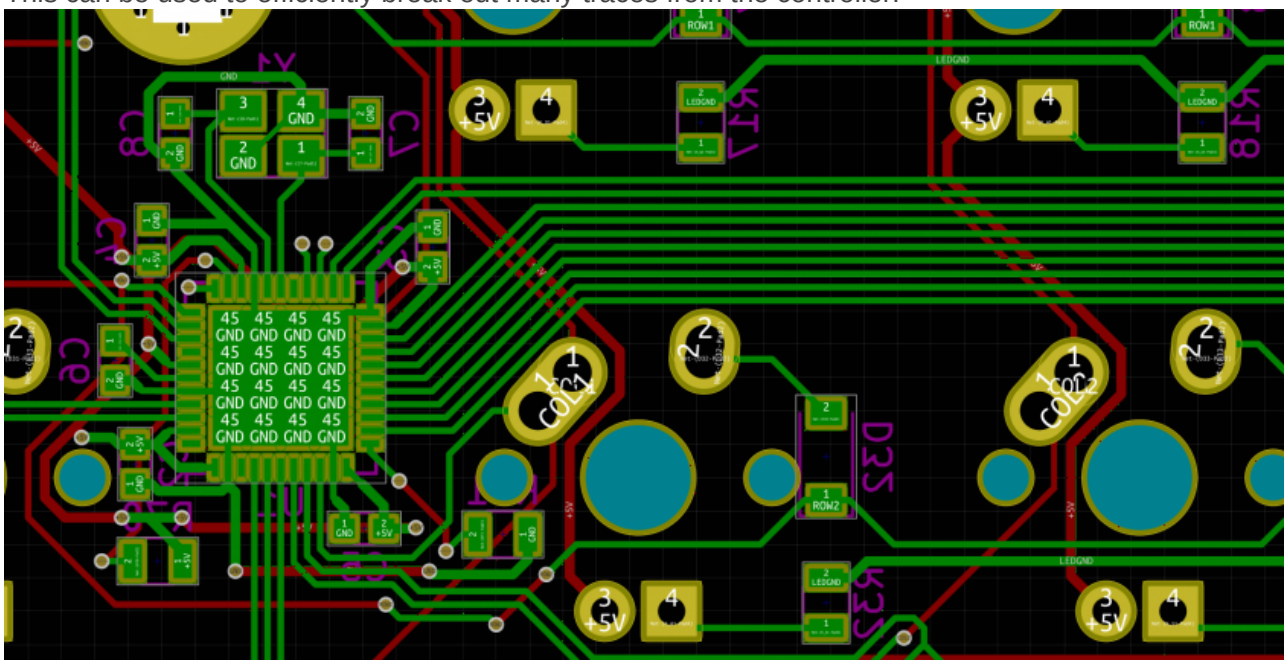
- Do not route traces close to the edge.
- Do not place EMI-emitting components, such as crystals, near a board edge.
- Surround pulsating traces with proper ground plane fill to reduce EMI.

Verticals and Horizontals

- Use front and back for horizontal and vertical traces respectively, or something similar to avoid clashing.
- Pick sides which do not force the abuse of vias for routing smd components.
- Let busses take precedence over singular traces (explained below).

Trace Bus

- Route many traces going in the same direction together for efficient use of space.
- This can be used to efficiently break out many traces from the controller.



Aesthetics

- Space things evenly.
- Bundle traces together if not susceptible to interference.

Backlighting And RGB

Bling bling.

QMK-Compatible LED backlighting

This is one LED per switch.

The structure is:

```
.     |- LED --- Resistor -|
      |                    |     N-Channel
5V ---|- LED --- Resistor -|----- MOSFET ----- GND
      |                    |       |
      |- LED --- Resistor -|     PWM PIN
```

- Use a N-channel MOSFET.
- A PWM pin can only deliver around 20mA absolute maximum. Drawing tons of LEDs from a single PWM pin will easily fry your microchip. Place a resistor between the PWM pin and MOSFET gate.
- All LED-Resistor pairs are placed in parallel.
- All LEDs can be run under a few or single resistor, but this leads to brightness inconsistencies among the LEDs.
- The current from the LED-Resistor pairs feed into the drain of the MOSFET.
- PWM feeds into the gate of the MOSFET with a 100~1k or so resistor, usually.
- Source of the MOSFET goes to ground.
- Each LED resistor should be planned to let around 1-5mA through the LEDs each. 20mA will hurt your eyes.

Choosing the correct MOSFET

Power Dissipation

- Make sure the current flowing through times the voltage drop doesn't lead to a power dissipation that doesn't fry the MOSFET.

Maximum voltage and current

- Check for Drain-Source voltage/current max ratings.

Size

- Do not pick a MOSFET which will not fit on your PCB.

RGB Underglow

WS2812B is the standard used for underglow.

- Draws 45-50mA per package maximum.
  - However, AliExpress GH60 PCBs have 16 WS2812B.
  - Current can be limited via QMK.
- Can be daisy-chained together. One LED passes data onto the next, and so on.
- Can be purchased in individual WS2812B or by strips. Use individual for assembled PCBs.
- Wired in the following way:
  - VDD to 5V
  - VSS to GND
  - DIN to previous LED DOUT, or AVR pin if first LED
  - DOUT to next LED DIN, or unconnected if last LED
  - Placing a 500 ohm resistor between AVR pin and first LED is highly recommended

Various other RGB choices

WS2812 (without B) - Unsimplified version of WS2812B with 6 pins
WS2811 - The controller within WS2812B. Can be used as standalone controller with other RGBs.
Controller-less RGBs - Use with separate PWM LED controllers.
WS2812C - Low current variant
SK6812 - An alternative that also works with QMK

Indicator LEDs

Simplest.

```
MCU PIN --- LED --- Resistor --- GND     (Source setup - Toggle pin high to turn on)


or


5V --- LED --- Resistor --- MCU PIN      (Sink setup - Toggle pin high to turn off)
```

LED and resistor can be swapped.

USB Type-C

Make your keyboard future-proof and reversible

Type-C USB

- Slightly wider than MicroUSB connectors. Keep in mind when designing PCBs to fit standard cases.
- Soldering is much more difficult.
  - Mini/Micro-USB has 5 pins.
  - Type-C has 24.
- Completely surface mount Type-C connectors are *impossible to solder by iron.*
  - Type-C connectors have two rows of pins, one of which exit from below the connector.
  - If hand-soldering via iron, choose a hybrid connector with through-hole pins for the under-connector row.
- Type-C connectors often do not extend as far as old Mini-USB connectors do from the edge of a PCB.
- Often, the legs of a Type-C through-hole connector are extremely short (<1mm). Thick PCBs may be difficult to assemble.

Design Considerations

- Type-C is physically larger.
- Type-C requires its own components to function.
- Some Type-C connectors have legs too short for 1.6mm PCBs.
- Assembly costs may increase.

Implementation

- CC pins must be connected to 5.1k pull-down resistors

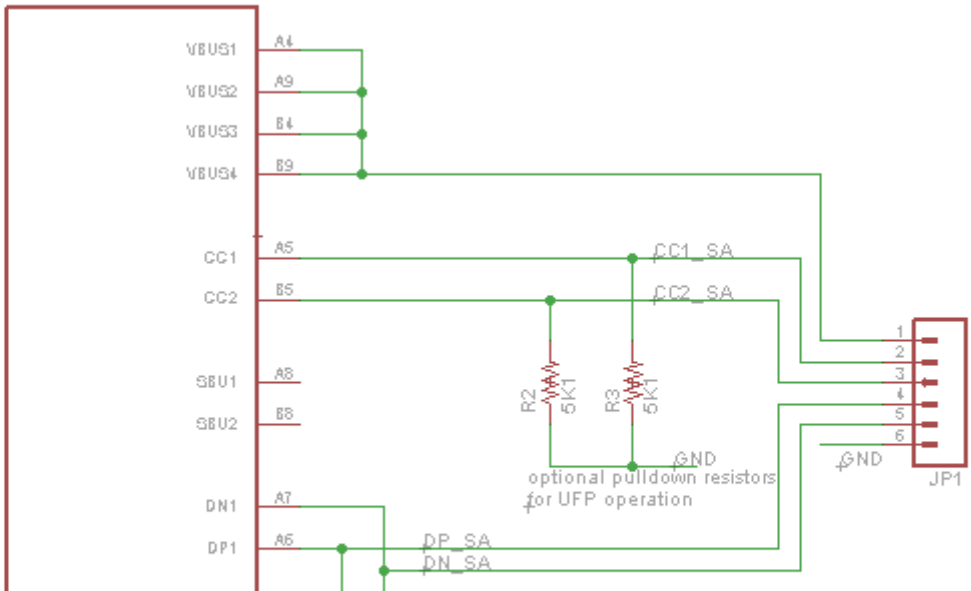


Diagram by Scorpia.co.uk

https://www.scorpia.co.uk/2016/03/17/using-usb-type-c-on-hobyist-projects/

This diagram breakouts to a header.

Keyboard use

- Pull-down the CC lines
- Use the DP/D+ and DM/D- lines just like any other USB connector
- Connect VBUS and GND like any other USB connector
- It is inferred that the pull-downed CC lines don’t have to be connected to anything else in particular.

Type C High-Current Legacy Devices

From ST’s PDF, it appears possible to draw more current through Type-C:

Converting STM32xx USB2.0 host to USB Type-C™ platform

This use case describes how to exchange a USB2.0 standard A receptacle with a USB Type-C™ receptacle.

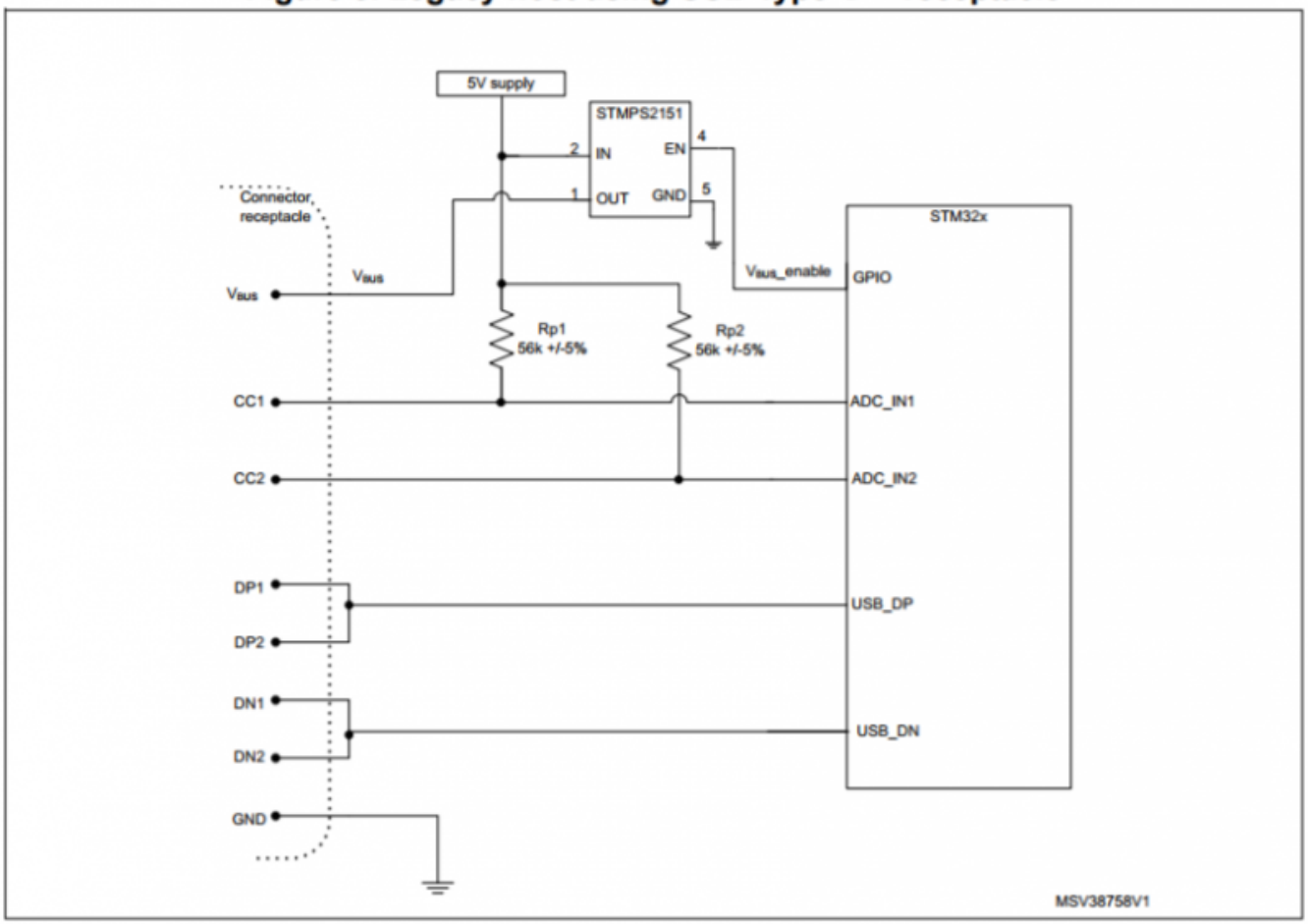
As the platform is designed for USB2.0, maximum current capacity is 500 mA. If higher supply current is available in application, Rp resistors could be adjusted to advertise 1.5 A or even 3 A.

A legacy host needs to be configured as a DFP by having Rp pull up between CC line and 5V supply.

As the plug is reversible, the two couples DP/DN need to be connected in pairs as close as possible to the receptacle, before being routed to the STM32xx device.

Monitoring CC lines thru ADC_IN inputs allow to detect device attachment and so to provide V_{BUS} on connector.

Figure 8. Legacy host using USB Type-C™ receptacle



Sources

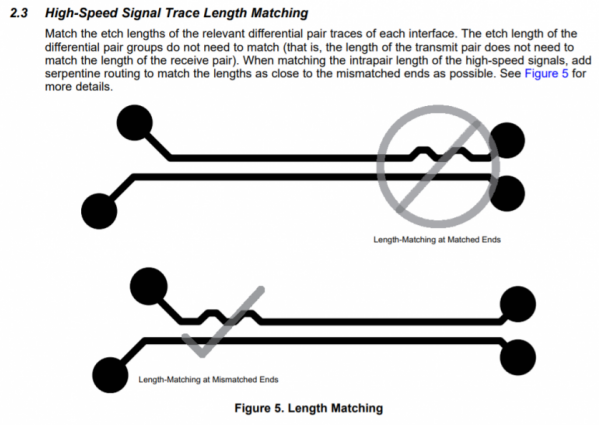
https://www.scorpia.co.uk/2016/03/17/using-usb-type-c-on-hobyist-projects/
https://www.mpe-connector.de/index.php?lang=en&menu=16&product_group[]=9&action=Search&id_product=7192
http://www.st.com/content/ccc/resource/technical/document/application_note/group0/a6/91/45/9e/12/a0/4d/42/DM00235987/files/DM00235987.pdf/jcr:content/translations/en.DM00235987.pdf

Good USB Routing Practices

A quick summary of Good Usb Practices

Trace length Matching

- Intel suggests that USB traces be matched to 150 mils within each other in length.
- Trace spacing is debated.
  - Intel suggests 20 mils spacing between the USB traces.
  - Texas Instruments suggests a minimum of 30 mils.
  - Other sources suggest 5x the trace widths as trace spacing.
- To forcefully match trace lengths, bend traces in a “zigzag” pattern at the mismatched end.
  - Diagram from Texas Instruments PDF



2.3 High-Speed Signal Trace Length Matching

Match the etch lengths of the relevant differential pair traces of each interface. The etch length of the differential pair groups do not need to match (that is, the length of the transmit pair does not need to match the length of the receive pair). When matching the intrapair length of the high-speed signals, add serpentine routing to match the lengths as close to the mismatched ends as possible. See Figure 5 for more details.

Figure 5. Length Matching

Signal Integrity

- Run USB lines in parallel at all times.
- Avoid jumping voids in ground planes.
- Lines may have to be routed non-parallel near connectors and controllers. If so, make sure the non-parallel area is within 0.25 inches of the connector and controller.
- Don’t bend USB signal lines at right angles.
- Try keeping the traces at least 100 mil from the edges of ground planes.
- Don’t run the USB signals in parallel to other pulsating traces such as the keyboard matrix or LED backlighting. If this is required, keep the traces at least 50 mils away.
- When the traces must go through parts (I.E. the 22-ohm resistors), keep the layout symmetric

- Diagram from Texas Instruments PDF

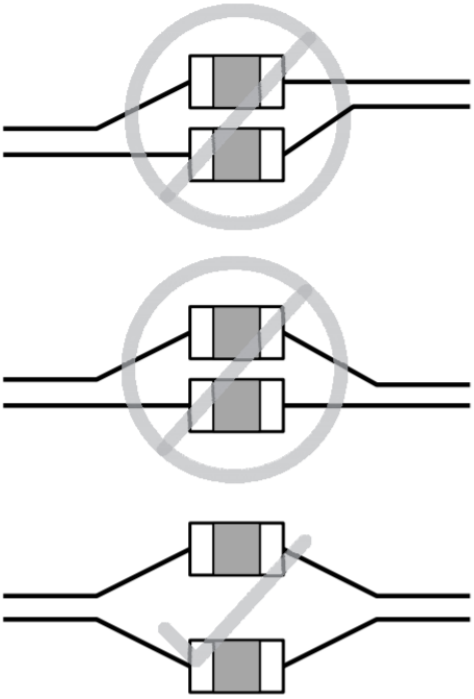


Figure 19. AC-Coupling Placement

Sources

http://www.usb.org/developers/docs/hs_usb_pdg_r1_0.pdf
http://www.ti.com/lit/an/spraar7g/spraar7g.pdf

EMI Shielding and Reduction

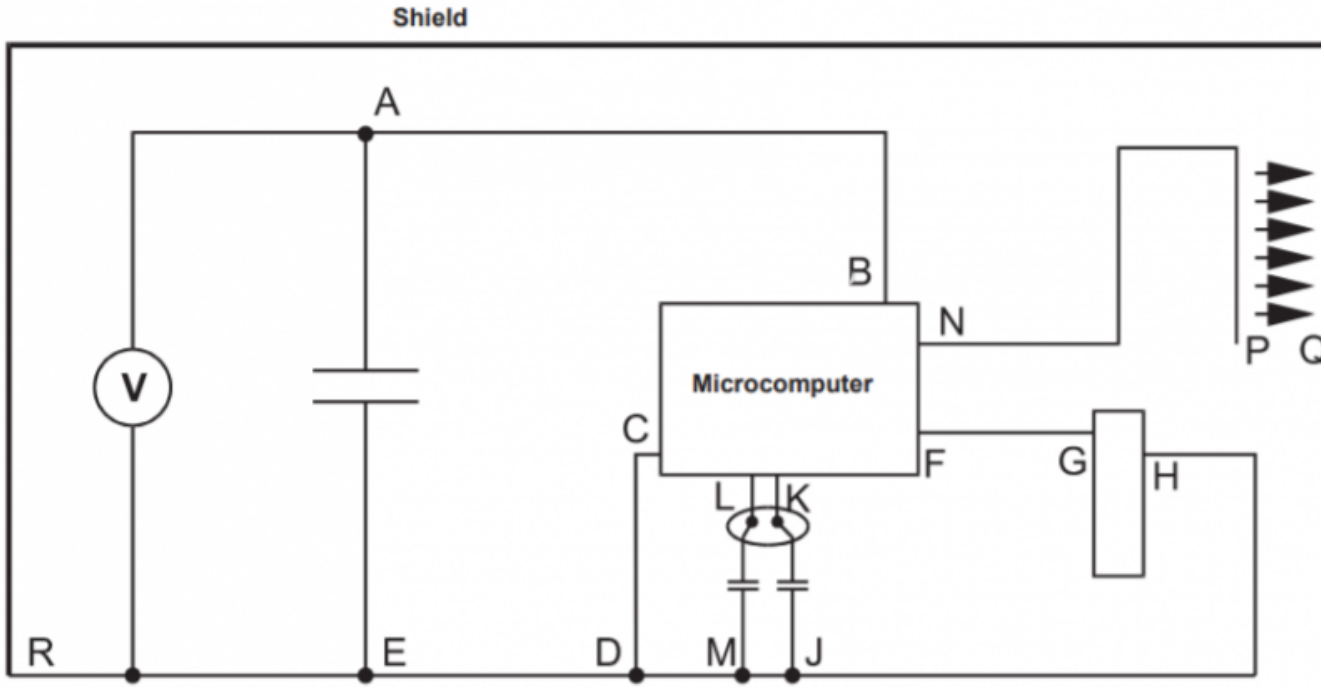
Healthy PCB designing.

Important to prevent a keyboard from malfunctioning from too much outside noise, and also to prevent a keyboard from wreaking havoc upon nearby devices such as pacemakers.
Texas Instruments states that any frequency above 80 MHz is a concern for radio frequency noise, but it’s not bad to be on the safe side, especially since these tips don’t change PCB costs at all.

General Design Considerations

Surface mount v.s. Through-hole

- Surface mount often has lower noise emissions because the smaller components can be crushed together into a smaller region.
- I cannot find the source, but I have read somewhere that round plated things such as through-hole pads can act as antennas much like a via under high frequencies.
  No loops
- Loops are antennas.



- This diagram from the Texas Instruments guide shows two loops.
  - A-B-C-D-E
  - F-G-H-D-C
- Avoid them whenever possible.

Vias can act as antennas under higher frequencies. Reduce when possible.

Ground Planes

For the microprocessor:

- Extend a ground pad on the other side of the controller, at least ¼ inches out beyond the area of the controller.
- Preferably, use the ground plane to shield crucial components for the controller as well.
- Don’t use the same ground plane for the controller as the rest of the board. Noisy ground current will flow into the controller area, and will destroy the point of having a ground plane in the first place.
  - Of course, this can be avoided with careful design of the ground plane.
  - Simply make sure that the ground planes do not act as a road for dirty ground signal to interfere with the clean controller ground area.
- For extended shielding, place a ground plane on the other side as well.
  - Again, this must follow the aforementioned rules: Do not connect the ground planes together in far distances at multiple points or it becomes a loop of its own.
  - Having just one via will be effective.
  - Having many vias distributed across the area at close range within each other is also effective. This will effectively turn the two planes into one huge “block” of copper.
- Preferably, separate the ground plane for the crystal from the ground plane of the controller area as well.
- Connect the controller plane to the rest of the board’s ground at only one location.
  - Since electricity travels in the shortest distance possible, proper connection of the two ground planes will make the noisy ground signal from other components such as LEDs travel straight to the USB connector, disregarding the area * near the controller.
- tl;dr: Shield the controller and don’t fling dirty ground current towards the controller.

General tips and info

- A ground plane not connected to the ground (USB ground in the case of a regular keyboard) is not technically grounded.
- For a wireless keyboard, connect the ground plane to the largest ground device possible, I.E. a metal plate or metal case.
  - If none of these are available, disregard this, since the ground plane itself serves as the largest ground.
- Making the ground plane go all the way to the rim of the PCB can cause some fabs to create the PCB with a bit of the ground plane visible at the edge of the PCB. This can cause short-circuiting if not careful.
- ALWAYS “rebuild” or “regenerate” the copper plane/ground plane area after changing any traces or component locations. Otherwise, the ground plane may be stuck to a bunch of other traces, causing huge short circuit issues.
- It is suggested to place the ground plane in the very end.
  - This makes for easier routing.
- Even if using ground planes, route ground traces.
  - This makes sure that all ground components are properly connected.
  - Using ground plane brute force may end up with improper connections for sensitive components near the controller. (See section 1)
- Larger ground planes are more effective. Thin ground planes basically do nothing.

Minimal Ground Plane

For when you want maximum aesthetics.

This bare minimum setup has been tested functional:

- Ground plane around crystal, same side as crystal
- Ground plane below MCU (for TQFP)
- Ground plane below USB connector
- Carefully isolating USB data lines, making sure other lines cross at 90 degree angles

Only attempt this if you are comfortable with careful component positioning and whatnot.

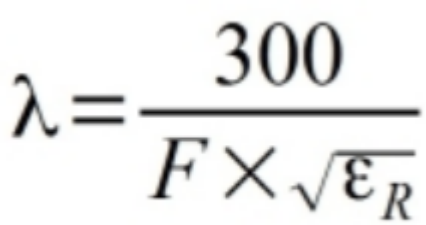
Via Fencing

It’s exactly how it sounds.

- Placing multiple ground vias in a line, connected on both sides to a non-noisy ground trace/area, acts as a “fence” or “wall” to vacuum the ugliest RF noise.
- Larger vias work better (Larger wall)

Calculating distance between vias

- At least 1/10 of wavelength
- At least 1/20 of wavelength recommended
- Based on this excerpt from EDN’s PDF:



$$\lambda = \frac{300}{F \times \sqrt{\varepsilon_R}}$$

- A common 16MHz signal used in keyboards has a wavelength of at least 23 inches, so the via spacing must be at most greater than 1 inches apart. Nothing to worry about here.
  Preferably, more vias = greater effect, but more vias = more drilling = higher PCB cost sometimes.
  Don’t overdo them to prevent turnaround time from skyrocketing or extra charges from piling up.

Uses

- Can be used to reinforce sensitive areas, such as the area around the controller.
  - This is an example of keeping RF interference out.
- Can be used when pulsing traces are too close to the edge of the PCB for the ground plane to effectively soak up the noise from exiting the board.

- Simply make an effective ground plane between the trace and the edge of the board.
- This is an example of keeping EMI in.

Sources

http://www.ti.com/lit/an/szza009/szza009.pdf

http://www.edn.com/Pdf/ViewPdf?contentItemId=4406491

https://electronics.stackexchange.com/questions/41871/via-fences-for-noise-reduction-of-a-chip-antenna

http://www.edn.com/Pdf/ViewPdf?contentItemId=4406491

Breakaway Tabs

Occasionally, you'll want to use breakaway tabs in your PCB to combine a few PCBs into a single physical board that breaks apart.

This guide explains how to use them to your advantage.

1. Obtaining the footprint

Grab the random-keyboard-parts.pretty library, either adding it to the project or adding it as a submodule (for git users).
Inside, there is a breakaway-mousebites footprint.

2. Placing the footprint

Simply place the footprint where you want the breakaway to be.
For the PCB holistic outline, simply draw on the edge.cuts layer.
For examples, see the Soyuz and Orbit PCBs.

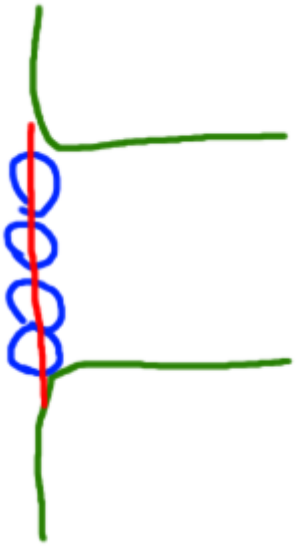
3. Considerations

There is no guarantee that every single PCB is manufactured the same.
Leave some space between the breakaway bites and traces/components, for an un-clean snap could tear into such nearby components.

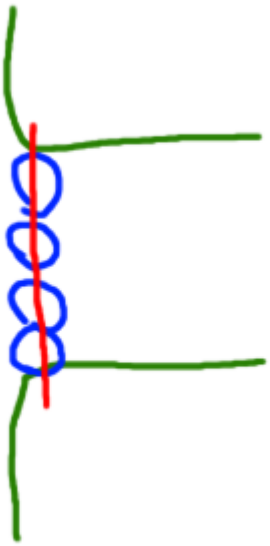
When having multiple breakaway tabs, take care to make it easy to break; for example, have a few "break lines", or lined up tabs so that things break smoothly instead of having to apply strange force and torque to the tabs.

The break usually occurs through the center of the mousebite drills. This means the placement of the mousebites determines how the board breaks apart.

Placing the footprint so the guide line matches up with the break line means that you will have small indents at the break tab, but protrusions will not extend beyond the expected outline. This is the ideal setup for internal things, such as regular PCBs. For example, see the Orbit repository.



On the other hand, placing the footprint so the drills are tangent to the break line means that you will have small protrusions at the cut, but no indents. This may be more ideal for exterior use parts such as plates, since the protrusions can be filed off whereas indents are much more difficult to deal with. For example, see the Soyuz repository.



4. Further reading

https://www.electronicdesign.com/boards/pcb-designers-need-know-these-panelization-guidelines