A Primer Into USB Type-C for Legacy USB2 Applications and Simple Power Delivery

Hugo Hu
v1.0.1 released 11/03/2023
hugo@hugohu.me
hugohu.me

Contents

1 Introduction

1.1 "I've been living under a rock since 2014, what's USB-C?"

USB-C is the most modern USB connector from the last decade. It supersedes all previous USB connector versions. It is backwards-compatible with previous USB-generations and all new ones are designed around it (including USB3 and USB4).

Also, it's reversible! The USB-C connector has a total of 12 pins: two GND, two VBUS, a USB2.0 pair, a SuperSpeed TX pair, a SuperSpeed RX pair, a Configuration Channel (CC) pin and a Sideband Use(SBU) pin. Since the connector is reversible, there are a total of 24 pins on the connector itself (12 on the top, 12 on the bottom).

USB-C allows for extraordinary data and power transfer better than anything in previous generations. USB4 can support 40Gbps (theoretically 80Gbps with USB4 Gen 4) and power delivery up to 100W (Standard Power Range, 20V @ 5A) or 240W (Extended Power Range, 48V @ 5A).

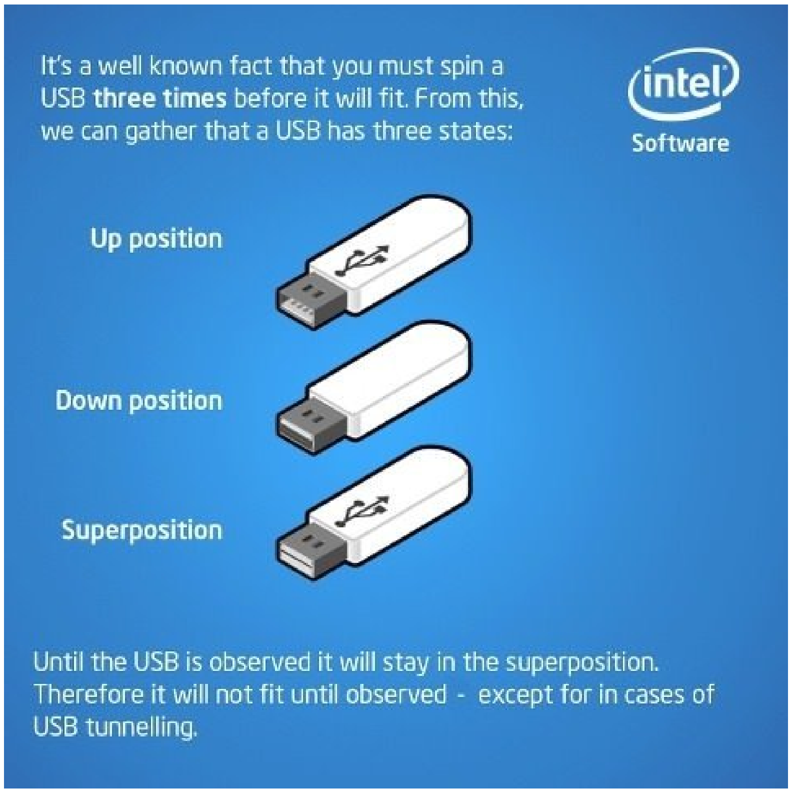

It's a well known fact that you must spin a USB three times before it will fit. From this, we can gather that a USB has three states:
intel
Software
Up position
Down position
Superposition
Until the USB is observed it will stay in the superposition. Therefore it will not fit until observed - except for in cases of USB tunnelling.

Figure 1: Rumors say that Intel made this ad themselves.

1.2 OK... and?

Since USB-C is backwards compatible with USB2 and even USB1.1, it should be pretty darn simple to just put in a new connector and rework the circuit, right?

Well, no. In 2019, the Raspberry Pi Foundation released the Raspberry Pi 4B- a powerful upgrade over the 3B+, and added USB-C to replace the micro-B power input.

The TL;DR is, they messed up. Instead of connecting separate resistors for the CC pins, they shorted the pins, which didn't allow power input for e-marked cables. Oops.

2 Converting USB2 to USB-C

Here's how a USB2 micro-B receptacle would be wired up (assuming it's a device):

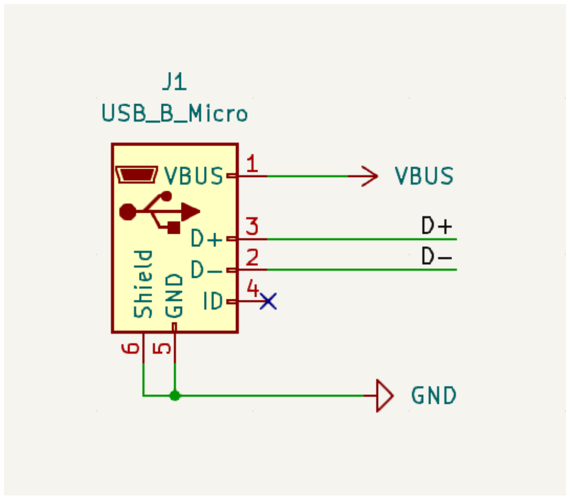


Figure 2: USB2 micro-B female port wiring.

A **incorrect** wiring to implement the same USB 2 signal through USB-C:

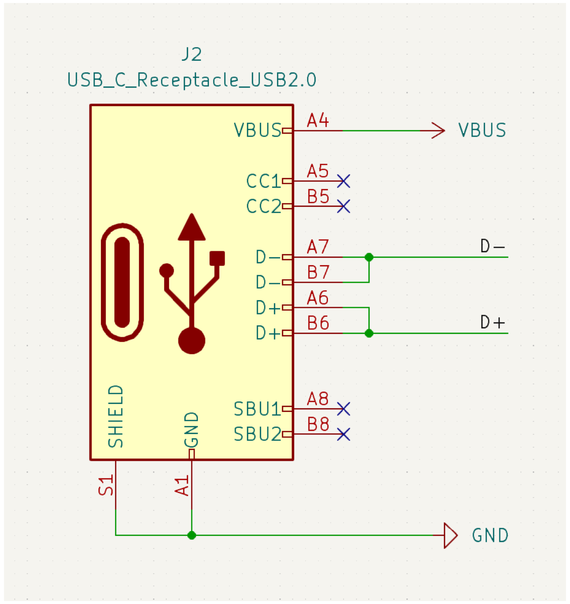


Figure 3: Faulty USB2 Type-C wiring.

This breaks out the D- and D+ pins, the VBUS pin, and GND. This should work!

Unfortunately, it doesn't. This would make your board perfectly OK with legacy non e-marked cables, but would not work with a "smart" e-marked cable. The reason for this is because USB-C requires you to negotiate for power. For legacy applications, your power source doesn't have this ability, so it just defaults to 5V @ 500mA(USB2.0) or 900mA(USB3.0).

With Power Delivery, you can essentially request a certain voltage and current from a PD-compatible source. One example is the $19 Apple 20W Power Brick which supports PD, capable of 5V@3A or 9V@2.22A.

2.1 Analog Power Negotiation

In order to request voltage, digital negotiation on the CC pin is required to negotiate for any voltage option above 5V. This clearly doesn't make sense to implement a chip like the CYPD3177-24LQXQ that adds cost and complexity just to get 5V for a legacy USB2 application. That's stupid.

Fear not. There's a cheap way to negotiate for additional current flow for USB Type-C charging at 1.5A@5V or 3.0A@5V. You might ask, "if I don't need more than 500mA, can't I leave the CC pins unpopulated?" No: CC pulldown resistors are required for a power source to enable VBUS at all. You need a 5.1K$\Omega \pm 10\%$ pulldown resistor on each CC pin (no, you cannot short CC1 and CC2) for a total of two resistors added. Not terrible, and pretty cheap. You'll get 5V as expected on VBUS making switching an older connector to USB-C as easy as adding in the connector and two pulldown resistors. Everything else in your circuit can stay the same.

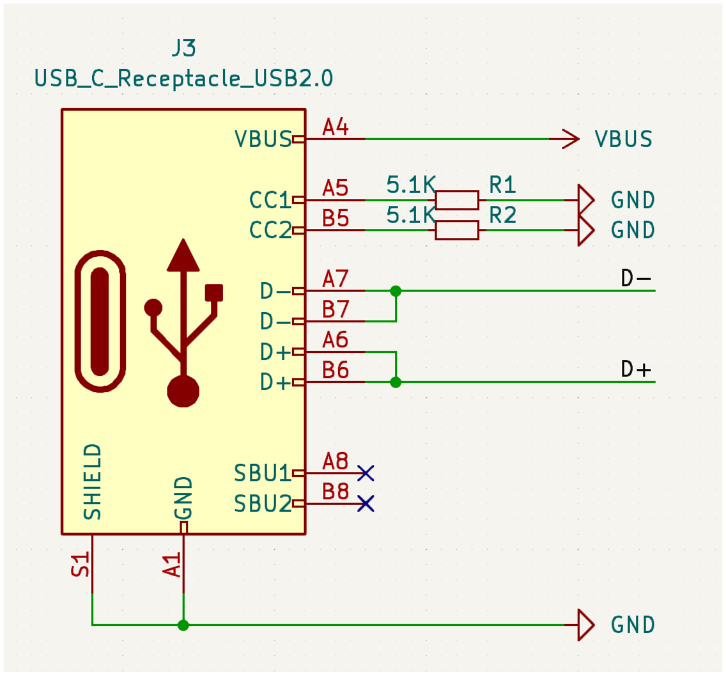


Figure 4: USB2 Type-C with 5V

3 Picking a Connector

There are a variety of connectors you can pick for USB-C: Power-Only, USB2, and USB3.

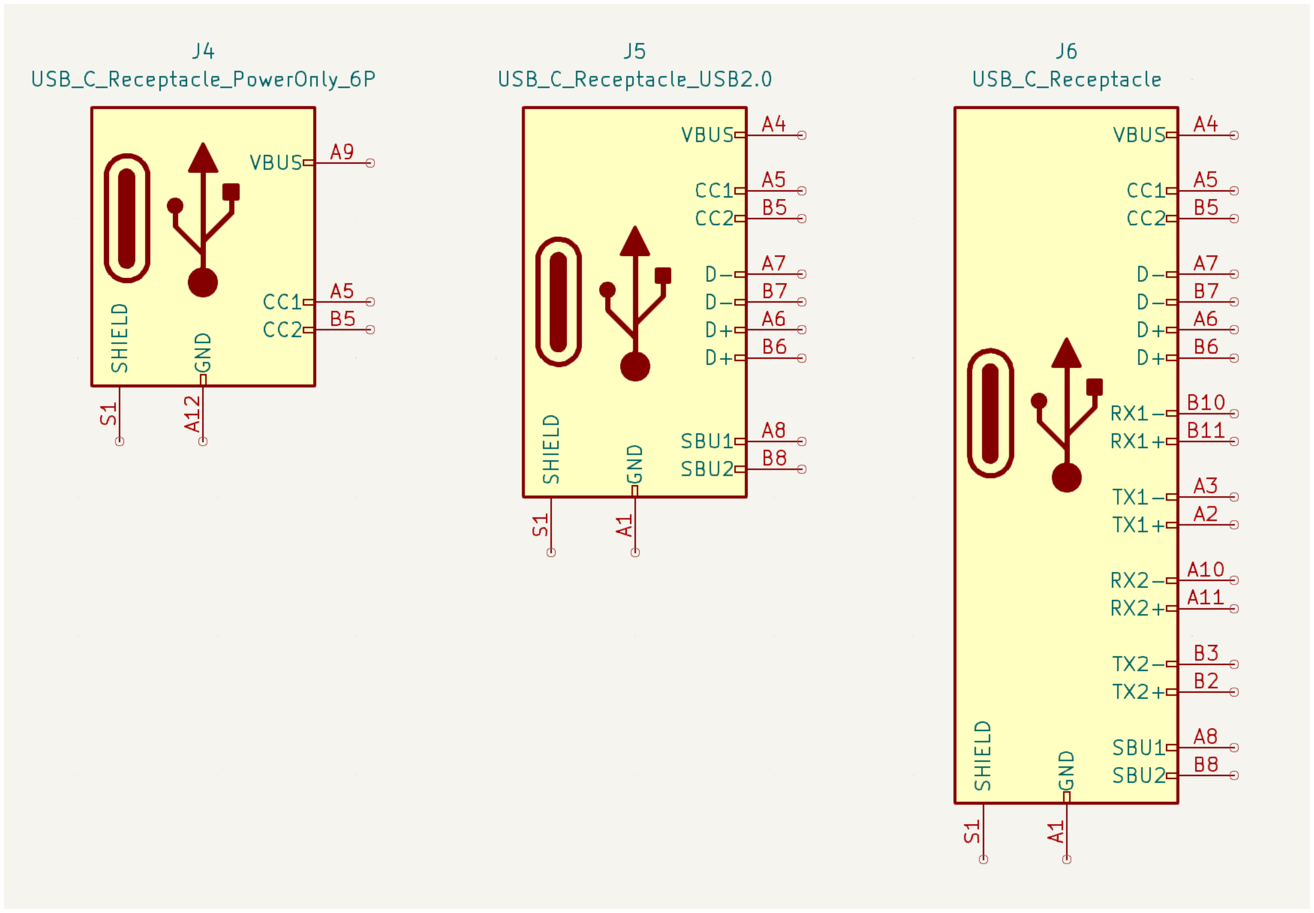
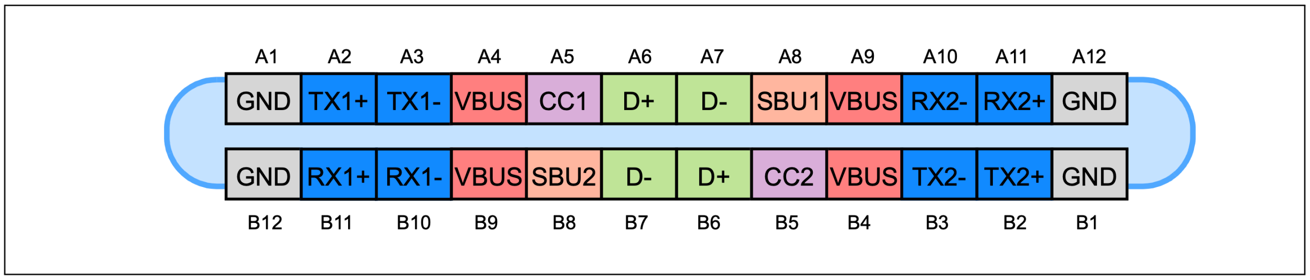


Figure 5: USB Type-C Selections

Additionally, for reference, the pinout of a USB-C connector.[1]



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
GND	TX1+	TX1-	VBUS	CC1	D+	D-	SBU1	VBUS	RX2-	RX2+	GND
GND	RX1+	RX1-	VBUS	SBU2	D-	D+	CC2	VBUS	TX2-	TX2+	GND
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1

Figure 6: USB-C Receptacle Pinout

[1]Figure 2, Microchip AN1953

3.1 Power Only

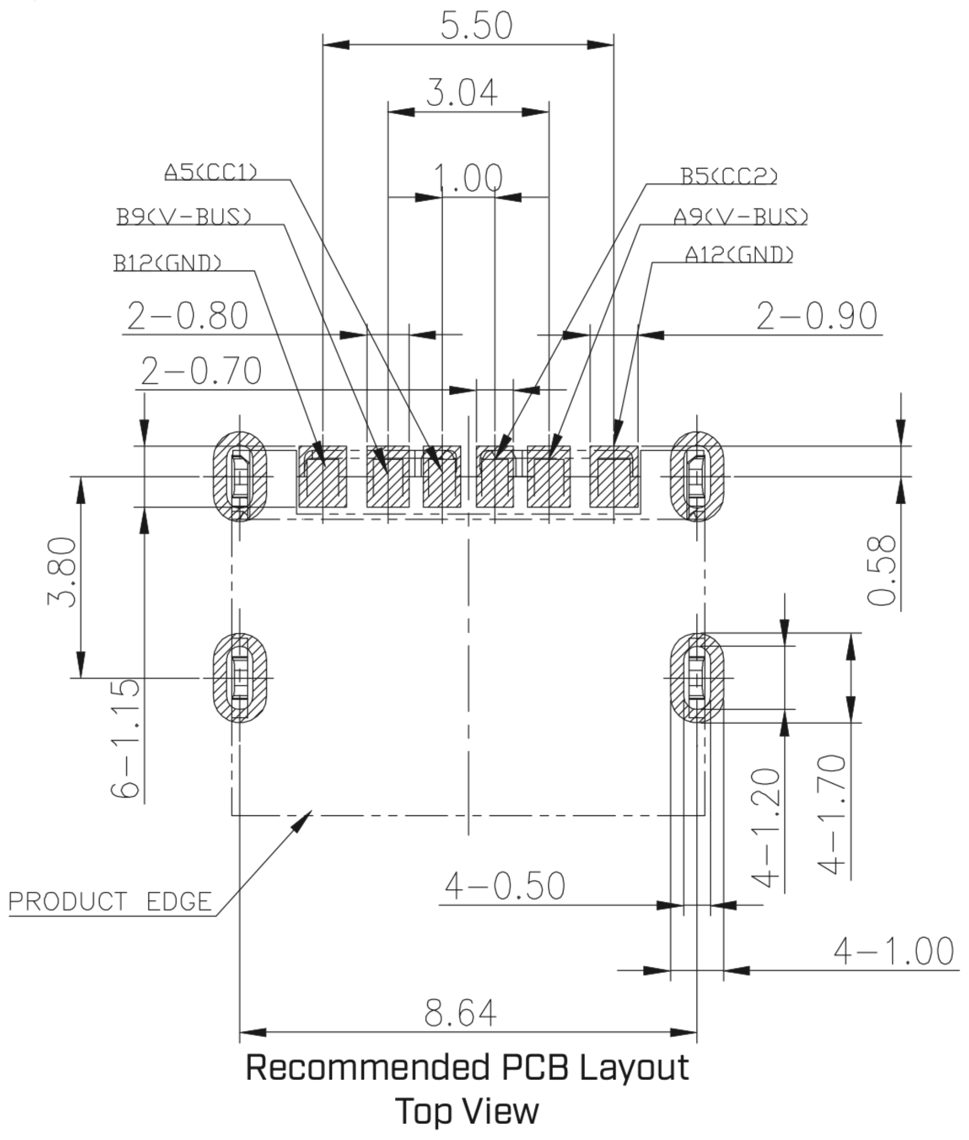


Figure 7: USB-C Power Only Pinout[2]

These connectors are very simple, having six pins each: VBUS, GND and CC on each side of the connector for a total of $3 \cdot 2 = 6$ pins. This does not support any data transfer and is much easier to route given the application that only requires power input. It has CC pins to allow for Power Delivery- though it's important to note not all connectors can support the full capability of Power Delivery- the UJC-HP-3-SMT-TR in the drawing shown only goes up to 20VDC and 3A on power pins together (1.5A individually).

[2]CUI Devices UJC-HP-3-SMT-TR

3.2 USB2 Type-C

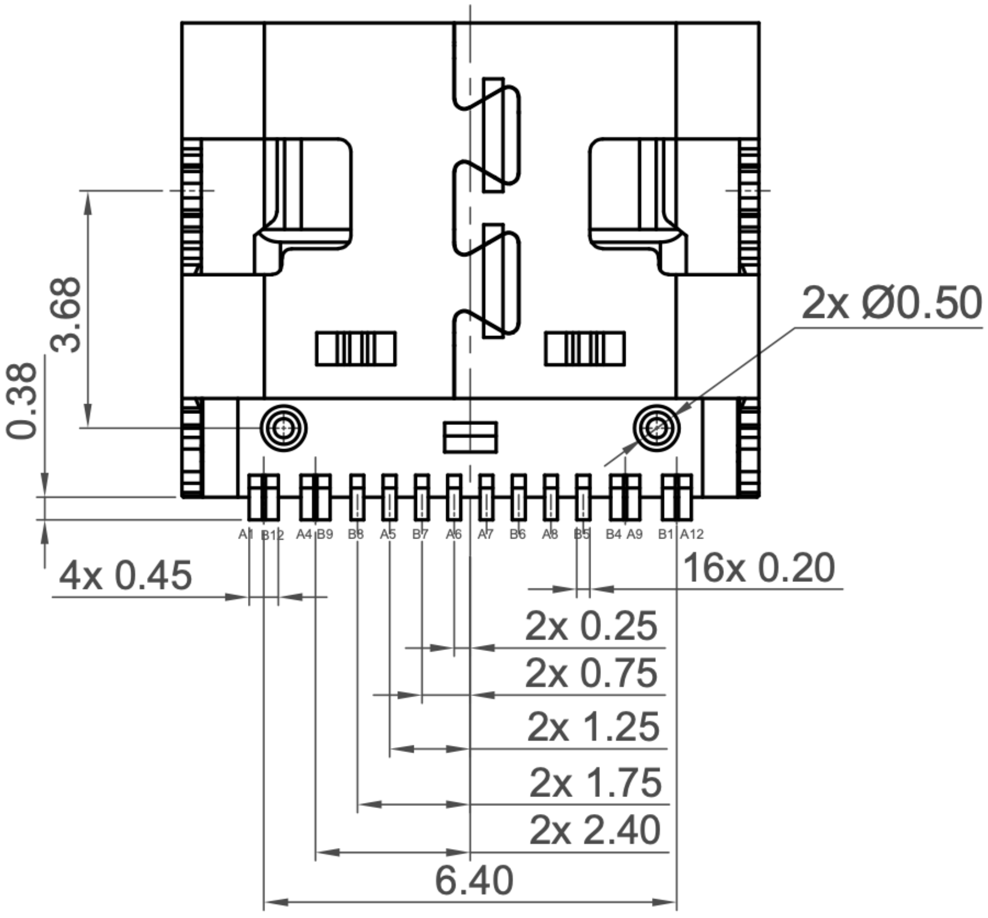


Figure 8: USB2 Type-C[3]

The USB2 connector has "12" pins as opposed to the standard expected 16 for USB2. This is due to some ingenious construction where they place the bottom and top GND / VBUS pins right next to each other, giving eight data pins and four "pairs" of power pins.

It's quite simple to use for USB2 applications. You must connect VBUS and GND on either side of the connector, the CC circuitry, as well as the D+/- data pins.

[3] GCT Co USB4105

3.3 USB3.2 Type-C

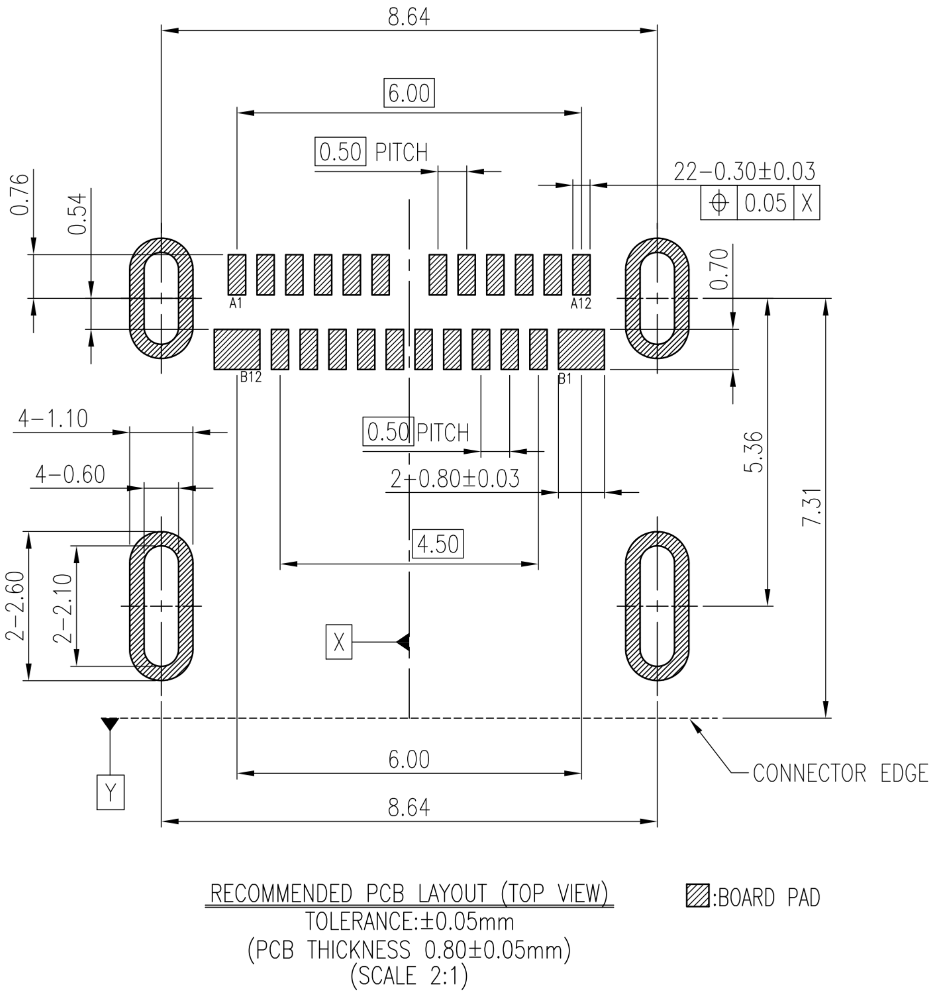


Figure 9: USB3.2 Type-C[4]

In order to implement the full USB3 (or USB4!) functionality, an additional two pairs of data pins are needed: TX+/- and RX+/-. This is an extra eight pins (four on each side) for a total of 24 pins. Unless you need USB3, it's generally a bad idea to use these connectors: they're more expensive, harder to route, and generally no benefit over a USB2 connector if your circuit is USB2.

[4]Amphenol Corp GSB3C31X3DSYHR

4 Power Delivery

As mentioned earlier in 2.1, you can get 5V at higher current levels than USB2-capability with the addition of 5.1KΩ $\pm$ 10% pulldown resistors on CC1 and CC2. However, it is impossible to negotiate higher voltage levels with resistors, and you'll need a dedicated USB Type-C port controller. Most of these aren't incredibly difficult to implement, but vary in capability and usage.

There are a few ways various chips expect you to configure voltage and current setting. Typically, it's one of the three:

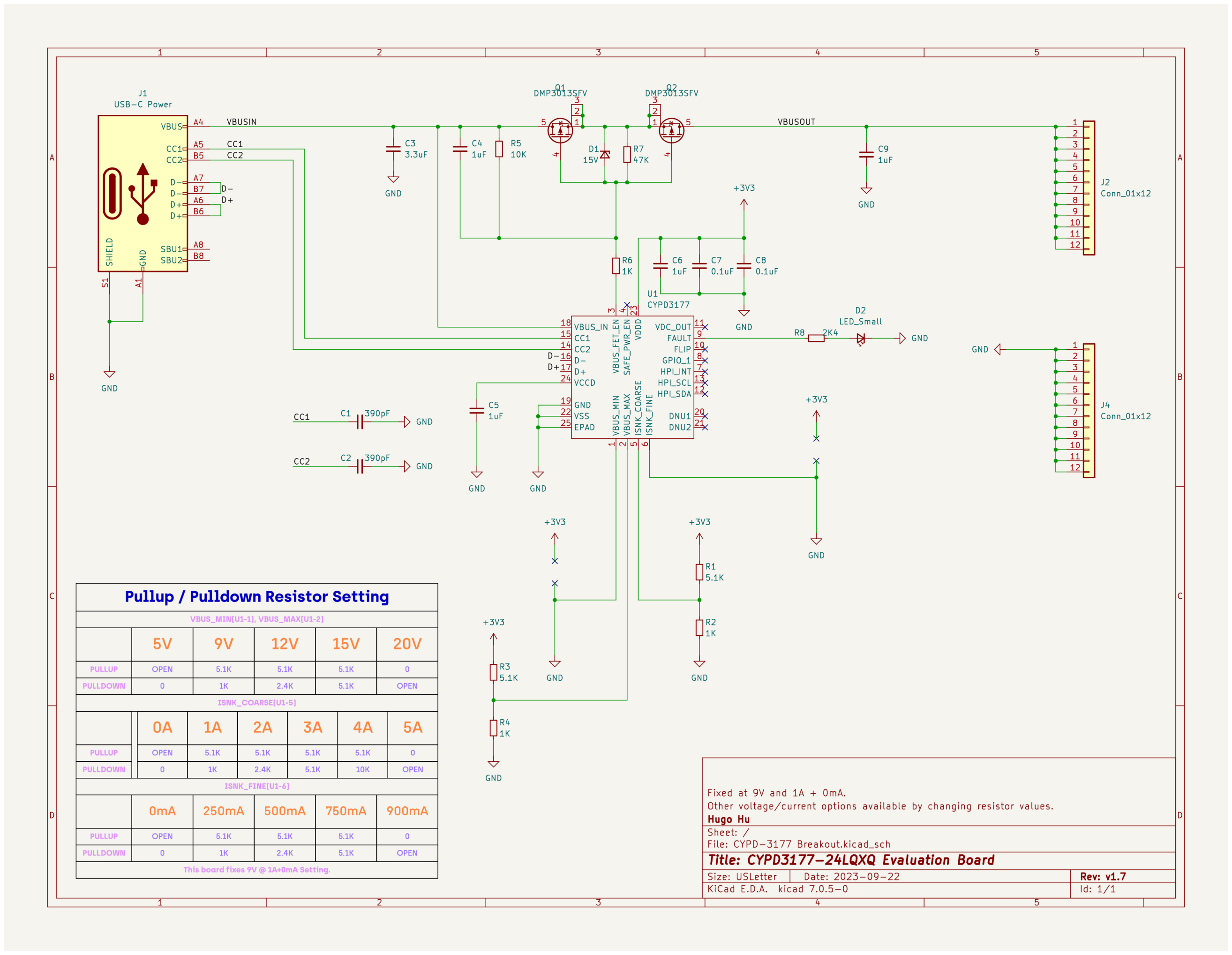
- Flashing setting to controller's onboard non-volatile storage
- Controlled through MCU (eg, STUSB4500)
- Controlled through resistor values (eg, CYPD3177)

While not all PD controllers support all these function, the more powerful ones do:

- Voltage settings (5V, 9V, 12V, 15V, 19V, 20V)
- Fine Current Adjustment
- Coarse Current Adjustment (up to 5A)
- Some ability to control via I^2C

A tested board based on the CYPD3177-24LQXQ, built for Blot, is open source and available here. An export of the schematic capture is available on the next page.

It's important to note that PD applications should be considered carefully. Power Delivery can provide above 5V of power, but not all applications can withstand high voltages and currents. One example would be having the output of a PD board be a standard USB-A connector. While this may seem like an OK idea in practice, it should be important to remember that the expected power from a USB-A port is 5V, and mistakingly configuring your PD output as 9V or above will likely destroy whatever's connected to the port.



Pullup / Pulldown Resistor Setting

VBUS_MIN[U1-1], VBUS_MAX[U1-2]

	5V	9V	12V	15V	20V
PULLUP	OPEN	5.1K	5.1K	5.1K	0
PULLDOWN	0	1K	2.4K	5.1K	OPEN

ISNK_COARSE[U1-5]

	0A	1A	2A	3A	4A	5A
PULLUP	OPEN	5.1K	5.1K	5.1K	5.1K	0
PULLDOWN	0	1K	2.4K	5.1K	10K	OPEN

ISNK_FINE[U1-6]

	0mA	250mA	500mA	750mA	900mA
PULLUP	OPEN	5.1K	5.1K	5.1K	0
PULLDOWN	0	1K	2.4K	5.1K	OPEN

This board fixes 9V @ 1A+0mA Setting.

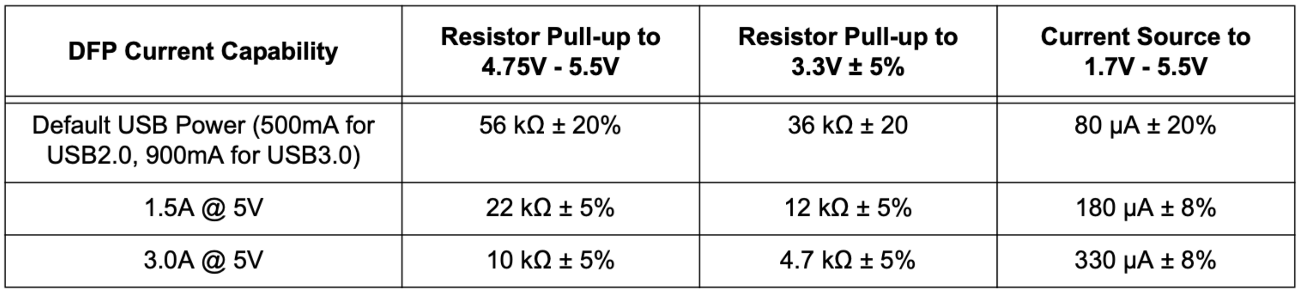
Fixed at 9V and 1A + 0mA.
Other voltage/current options available by changing resistor values.
Hugo Hu
Sheet: /
File: CYPD-3177 Breakout.kicad_sch
Title: CYPD3177-24LQXQ Evaluation Board
Size: USLetter Date: 2023-09-22 **Rev: v1.7**
KiCad E.D.A. kicad 7.0.5-0 Id: 1/1

Figure 10: CYPD3177 Schematic Capture[5]

[5]Image Source

5 Downstream USB-C Ports

We can now do a lot with a USB-C input, but what about converting a downstream facing USB-A port to USB-C? For example, if we had a USB hub with USB2.0 Type-A outputs, and wanted to make them USB2.0 Type-C. It's not necessarily as easy as simply adding the port, but close enough. We just need two CC resistors, one on each pin.



DFP Current Capability	Resistor Pull-up to 4.75V - 5.5V	Resistor Pull-up to 3.3V ± 5%	Current Source to 1.7V - 5.5V
Default USB Power (500mA for USB2.0, 900mA for USB3.0)	56 kΩ ± 20%	36 kΩ ± 20	80 μA ± 20%
1.5A @ 5V	22 kΩ ± 5%	12 kΩ ± 5%	180 μA ± 8%
3.0A @ 5V	10 kΩ ± 5%	4.7 kΩ ± 5%	330 μA ± 8%

Figure 11: DFP Resistor Chart[6]

To convert a traditionally USB2.0 design, it's most likely that the design only supports 500mA (Default Power), so in most cases, two 56KΩ ± 20% pull-up resistors should be sufficient. If your design receives external power, then it's a good idea to consider increasing maximum current to 1.5A or 3.0A with the resistor pull-up values specified.

For example, this USB2 hub design takes existing USB-A outputs and converts them to USB-C out. Keep in mind, changing the connector doesn't actually give you any increase in speed- a USB2 design will continue to operate at USB2 speeds even with USB-C.

[6]Microchip AN1953 DFP Resistor Chart

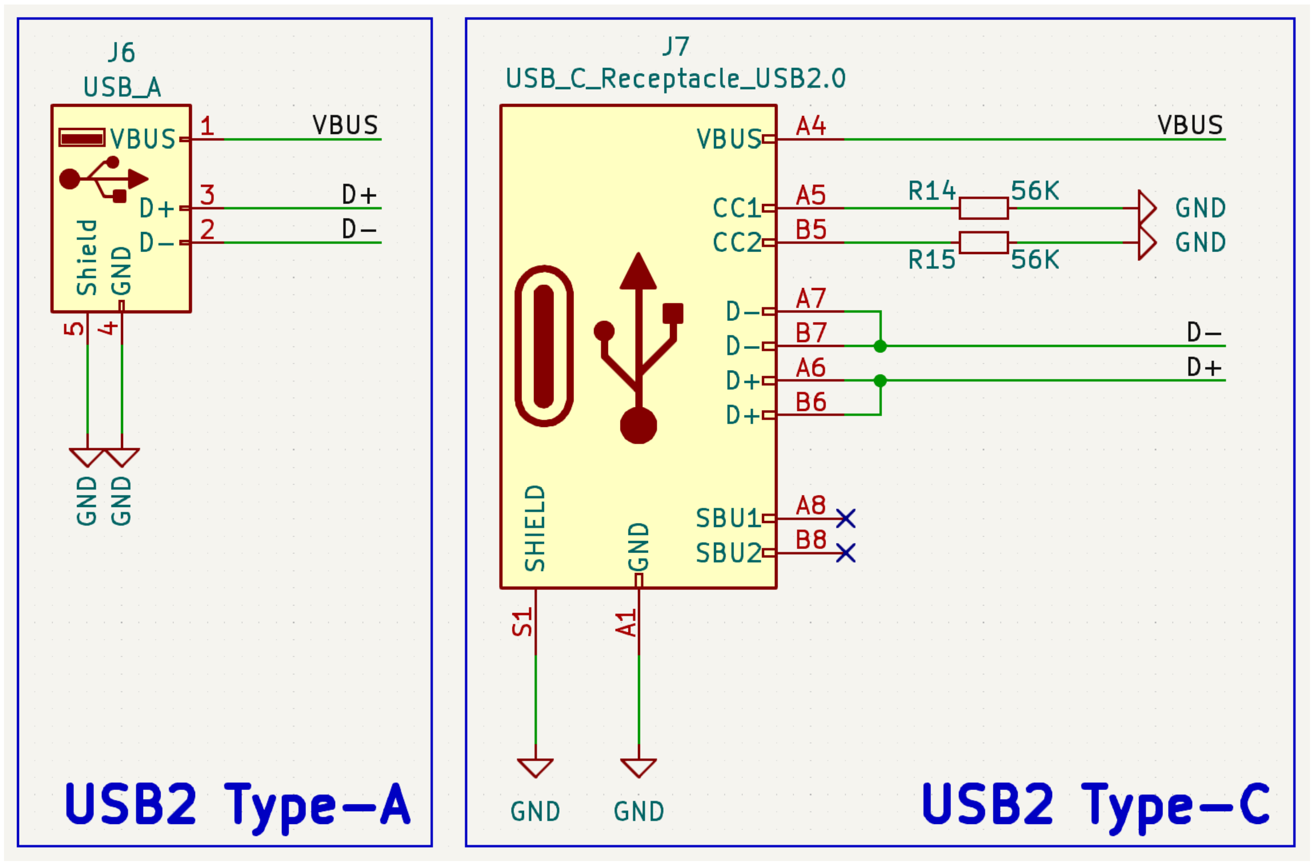
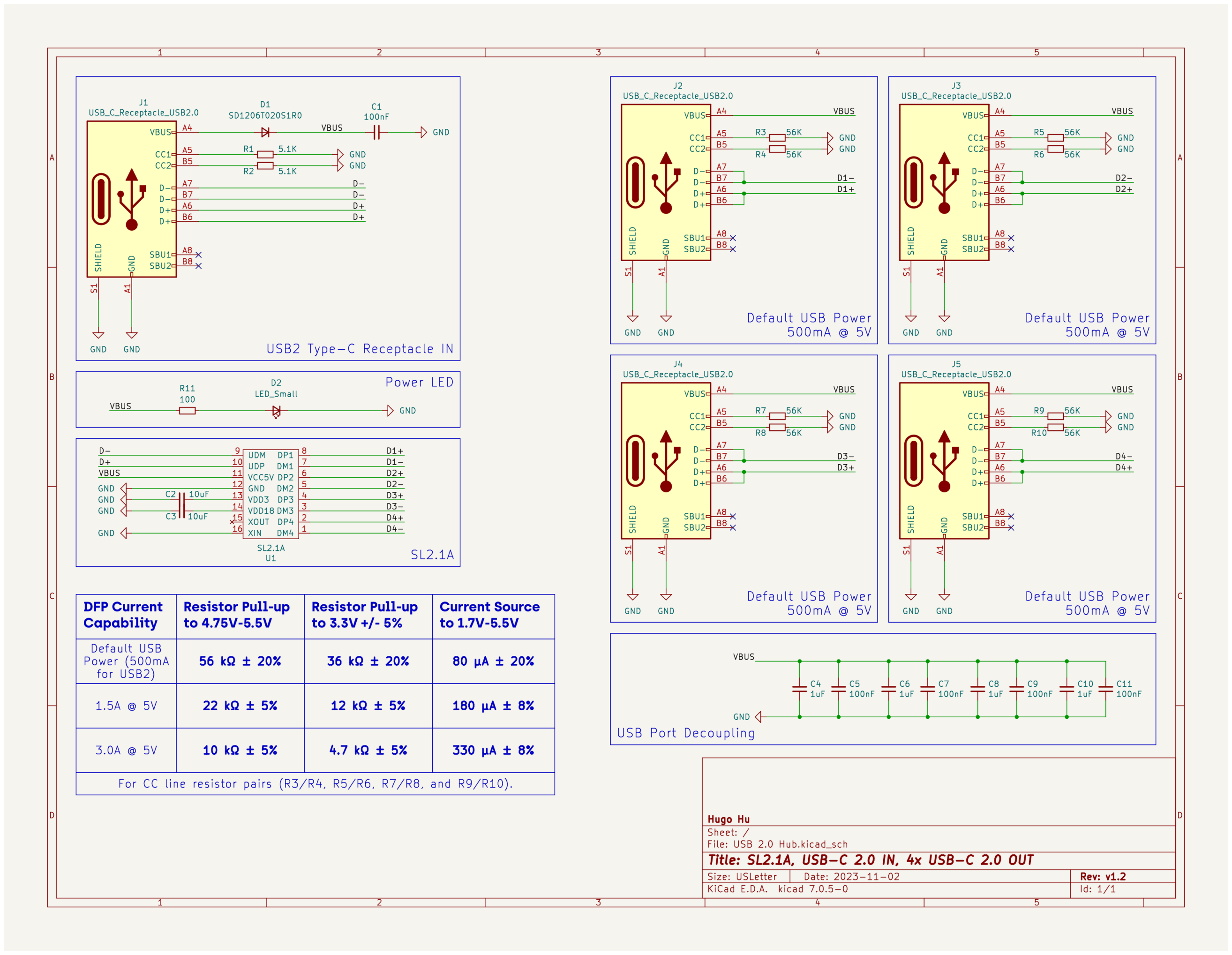


Figure 12: USB Type-A vs USB Type-C 500mA Power

[7]Can be found here: https://github.com/Hugoyhu/SL2.1A-Type-C-Hub



DFP Current Capability	Resistor Pull-up to 4.75V-5.5V	Resistor Pull-up to 3.3V +/- 5%	Current Source to 1.7V-5.5V
Default USB Power (500mA for USB2)	56 kΩ ± 20%	36 kΩ ± 20%	80 μA ± 20%
1.5A @ 5V	22 kΩ ± 5%	12 kΩ ± 5%	180 μA ± 8%
3.0A @ 5V	10 kΩ ± 5%	4.7 kΩ ± 5%	330 μA ± 8%

For CC line resistor pairs (R3/R4, R5/R6, R7/R8, and R9/R10).

Figure 13: SL2.1A based USB2 Type-C hub[7]

6 Conclusion

This document is hardly a comprehensive source for USB-C. This is more-so an introduction to new features offered by the USB-C standard. The information here is mostly accurate (for most applications) for a USB2 application, but lacks much critical information for a USB3+ based design. It's also important to note that not all information here is complete, and may at times be a simplification. For more information about the USB-C spec and implementation, you should take a look at the further reading below.

7 Further Reading

Microchip Application Note 1953: Introduction to USB Type-C

Texas Instruments: A Primer on USB Type-C and USB Power Delivery Applications and Requirements

ST Microelectronics TA0357: Overview of USB Type-C and Power Delivery Technologies