

This source describes Open Hardware and is licensed under the CERN-OHL-S v2.

You may redistribute and modify this source and make products using it under the terms of the CERN-OHL-S v2 (<https://ohwr.org/cernohlsv2.txt>). This source is distributed WITHOUT ANY EXPRESS OR IMPLIED WARRANTY, INCLUDING OF MERCHANTABILITY, SATISFACTORY QUALITY AND FITNESS FOR A PARTICULAR PURPOSE.

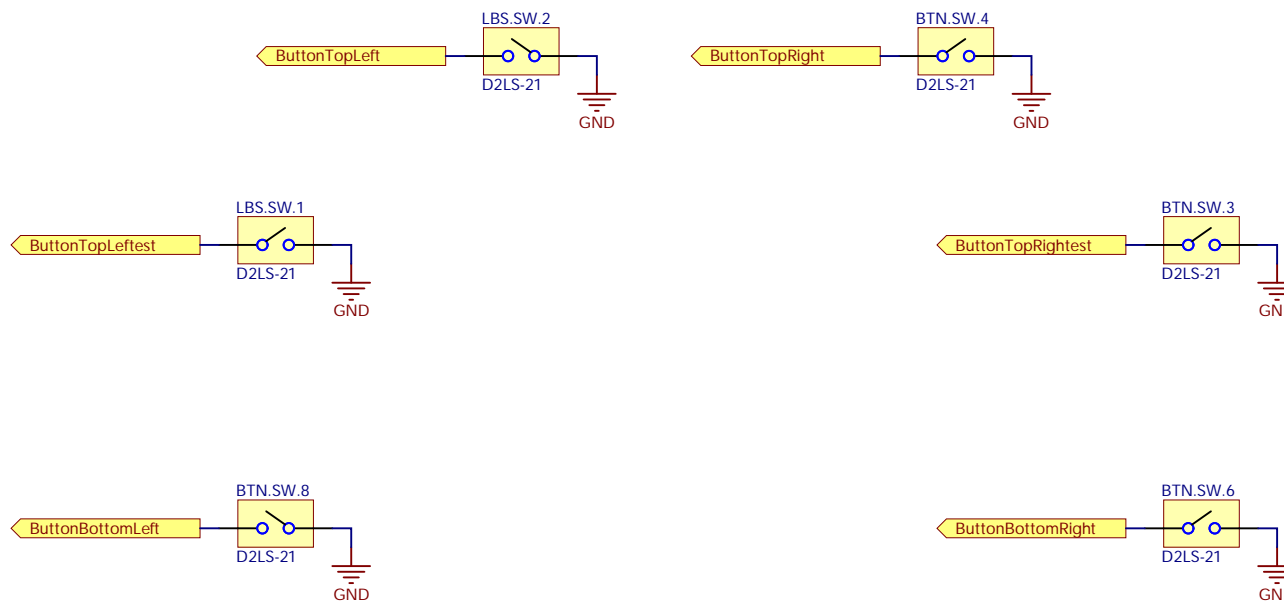
Please see the CERN-OHL-S v2 for applicable conditions.

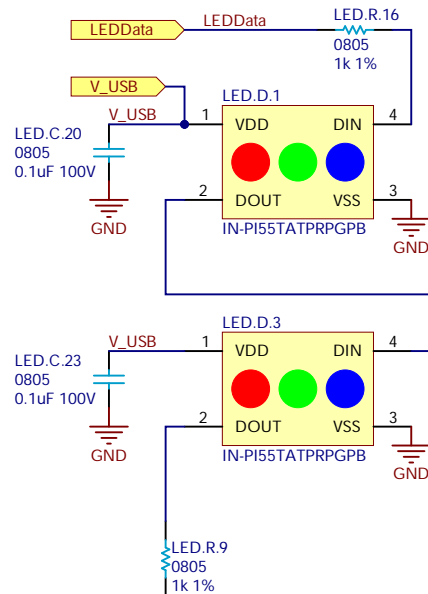
Source location: <https://github.com/ploopyco/>

As per CERN-OHL-S v2 section 4, should You produce hardware based on this source, You must where practicable maintain the Source Location visible on the external case of the products you make using this source.

These switches can be found under the various buttons on the mouse.

Activate the pull-up resistors on the GPIO pins attached to these switches to make them work.





Pick and Place Fiducials

MCH.FID.1



IPC-7351 Fiducial 1mm

MCH.FID.2



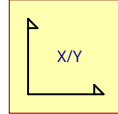
IPC-7351 Fiducial 1mm

MCH.LOGO.1



Ploopy logo, silkscreen

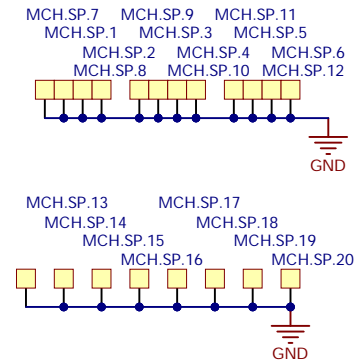
MCH.LOGO.2



Coordinate system marker for assembly

Spark Gaps -- Case

Since the case has gaps in it, we expect ESD to worm its way in via creepage and perhaps other ways. To protect the board from this eventuality, we place spark gaps along the edges.



The SCKI nets have to be connected to one of the GPOUT<n> pins on the RP2040. These are GPIO21,23,24,25.

Only one of the two outputs has to actually send a clock over to the PCM3060, which can operate from one clock pin.

I2C pins are connected to every GPIO pin! SCL is connected to odd GPIO pin numbers and SDA is connected to even ones. Make sure you connect both pins to the SAME I2C port though (there are 2!).

PIO pins are mappable to all GPIO pins, so it doesn't matter where the I2S lines are hooked up.

Pins adjacent to I2S lines are grounded for signal integrity, since these signals go quite quickly. Unused pins are also grounded and should be set to output LOW to improve grounding.

This header can be pinched with tweezers or otherwise shorted during startup to force the RP2040 into USB bootloader mode. To do this, short the header while the board is unpowered, and then plug a USB cable in. If successful, the computer the RP2040 is attached to should detect a USB mass storage device that you can then drag and drop a new firmware file onto. Very cool!

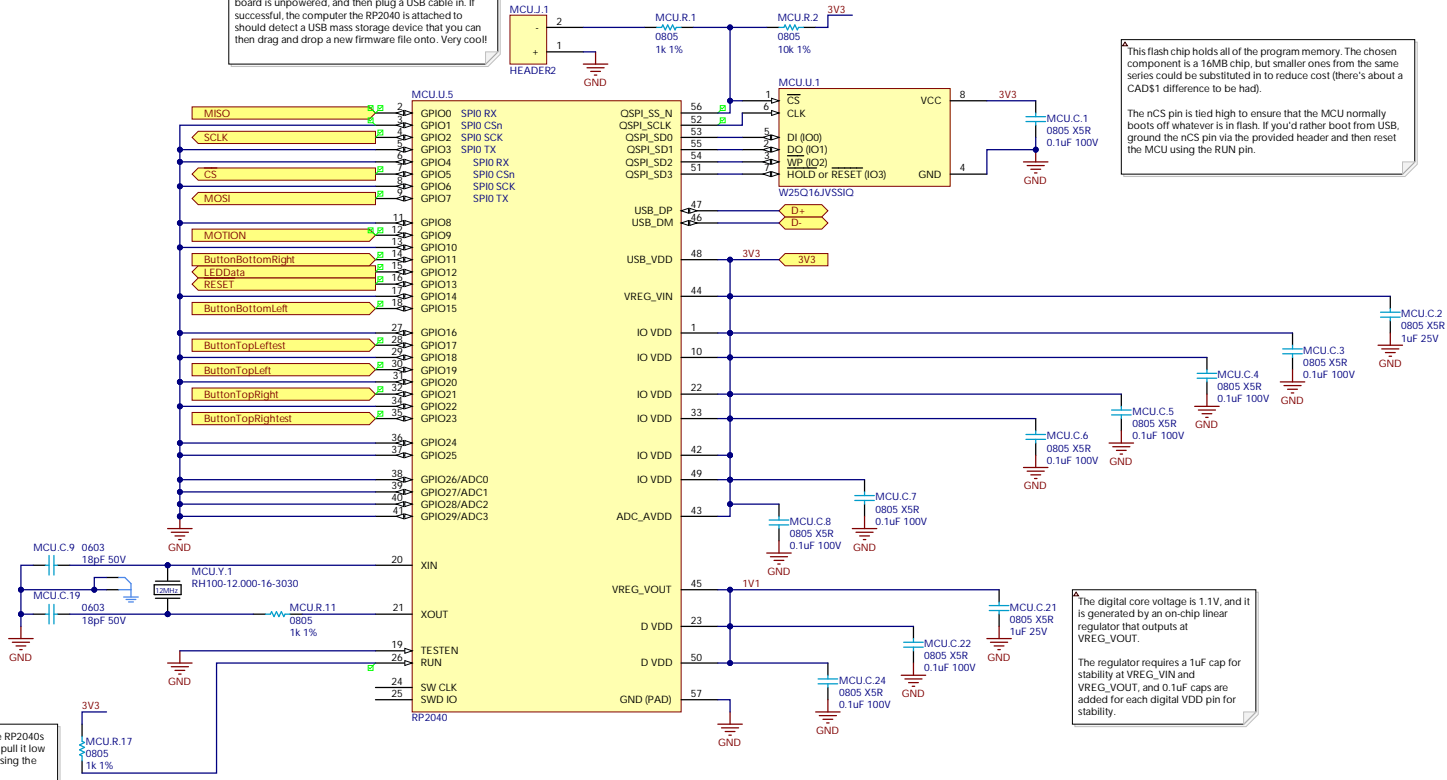
This flash chip holds all of the program memory. The chosen component is a 16MB chip, but smaller ones from the same series could be substituted in to reduce cost (there's about a CAD\$1 difference to be had).

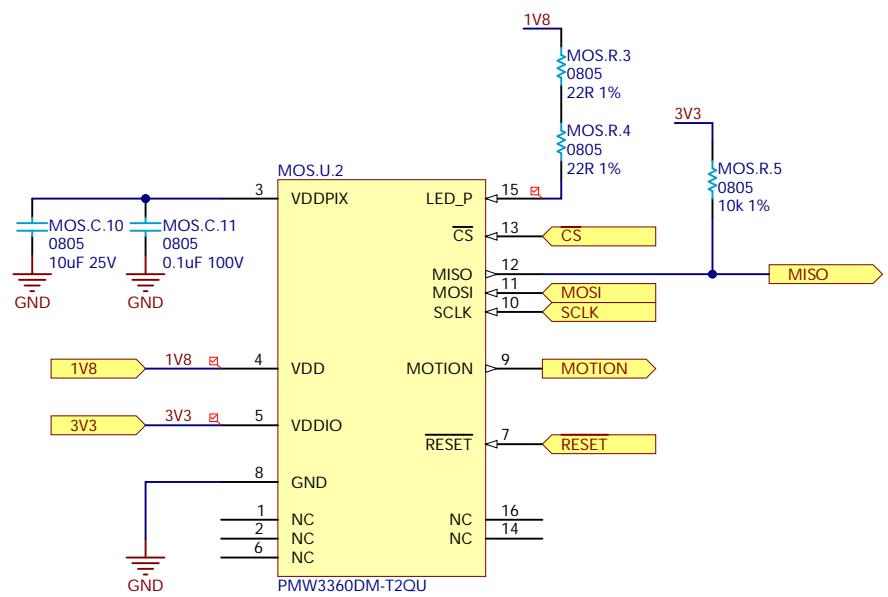
The nCS pin is tied high to ensure that the MCU normally boots off whatever is in flash. If you'd rather boot from USB, ground the nCS pin via the provided header and then reset the MCU using the RUN pin.

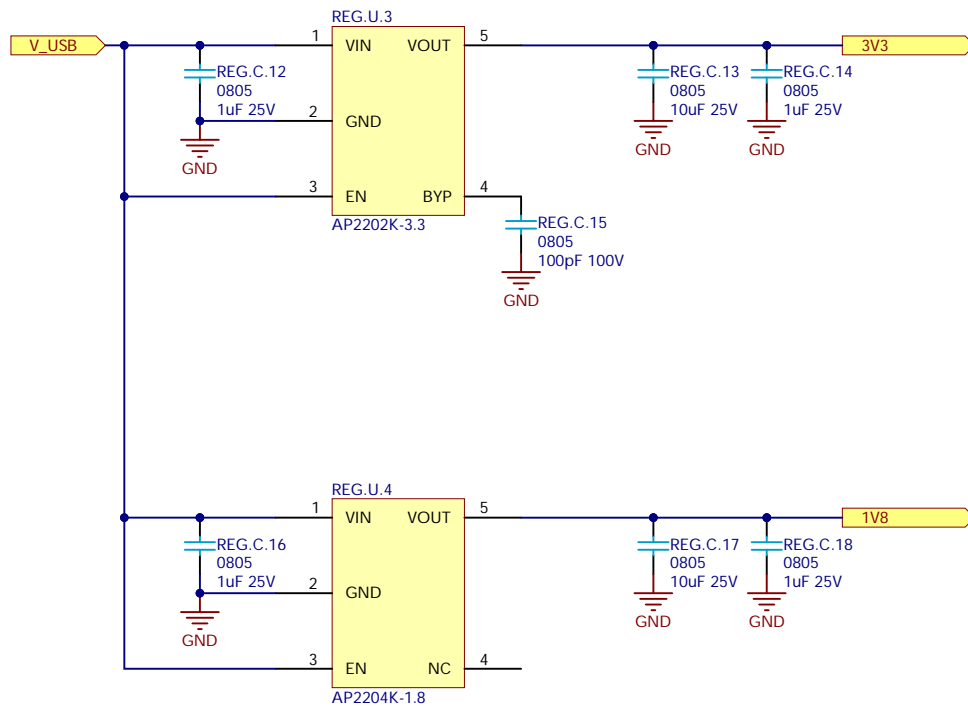
The digital core voltage is 1.1V, and it is generated by an on-chip linear regulator that outputs at VREG_VOUT.

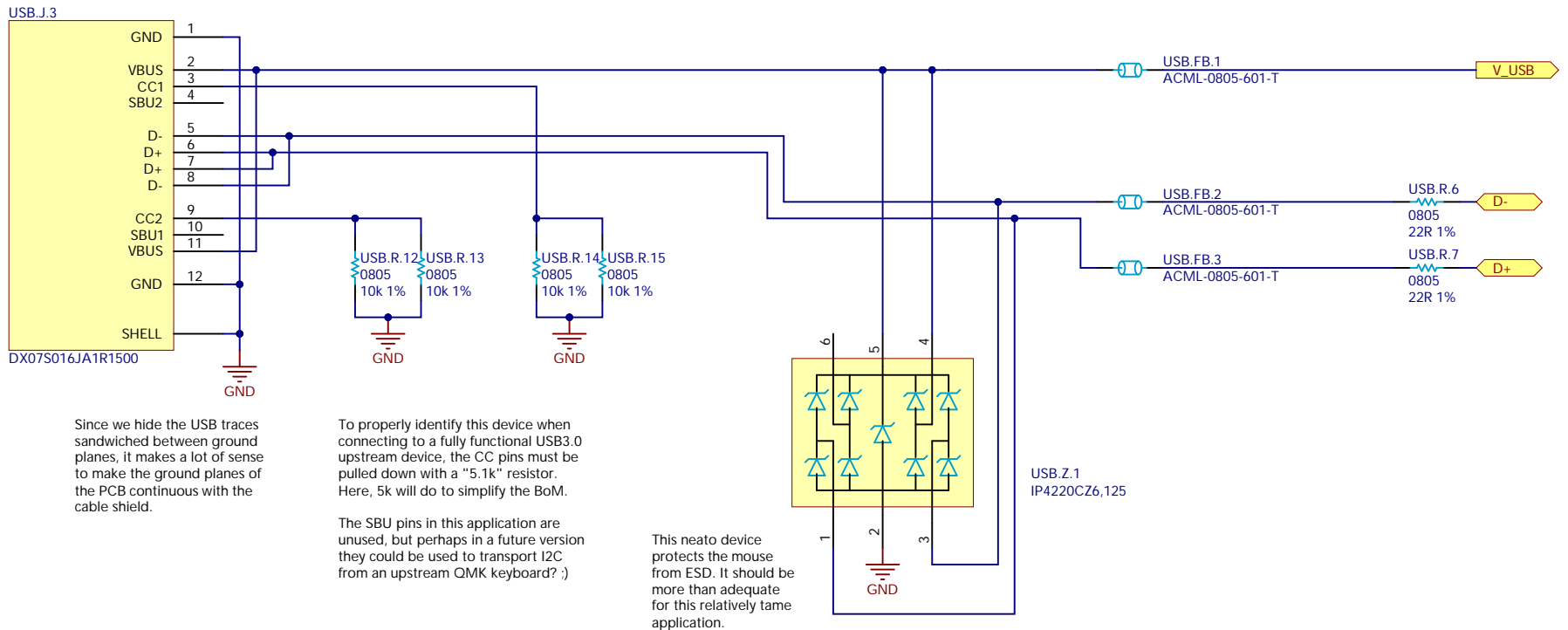
The regulator requires a 1uF cap for stability at VREG_VIN and VREG_VOUT, and 0.1uF caps are added for each digital VDD pin for stability.

The RUN pin is the RP2040s external reset pin; pull it low to reset the part, using the provided header.







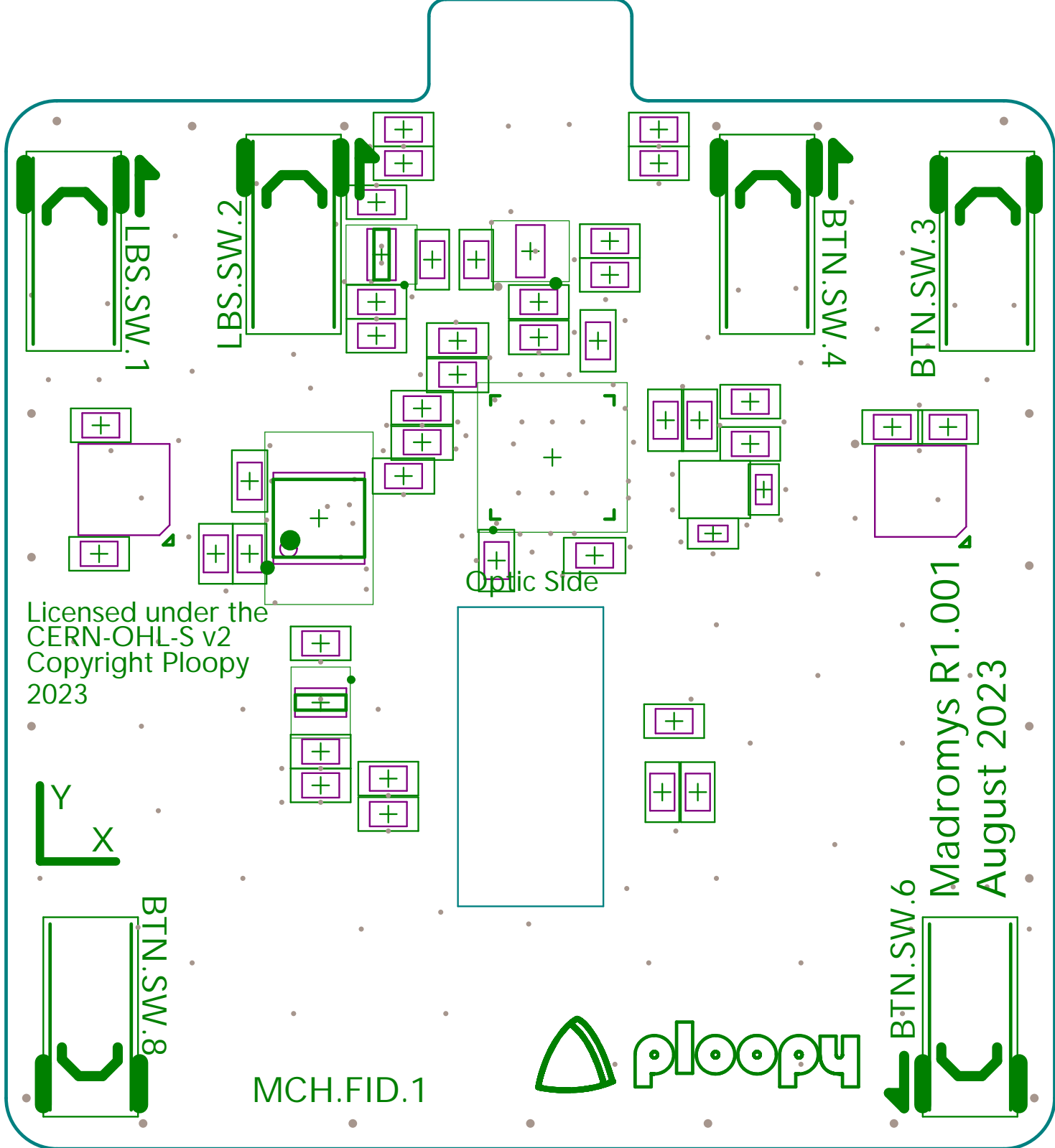


Since we hide the USB traces sandwiched between ground planes, it makes a lot of sense to make the ground planes of the PCB continuous with the cable shield.

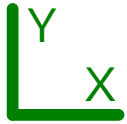
To properly identify this device when connecting to a fully functional USB3.0 upstream device, the CC pins must be pulled down with a "5.1k" resistor. Here, 5k will do to simplify the BoM.

The SBU pins in this application are unused, but perhaps in a future version they could be used to transport I2C from an upstream QMK keyboard? ;)

This neat device protects the mouse from ESD. It should be more than adequate for this relatively tame application.



Licensed under the
CERN-OHL-S v2
Copyright Ploopy
2023



MCH.FID.1



BTN.SW.6
Madromys R1.001
August 2023