Nikola Manolov

Board Outline

USB Power Injector

DETAILED DESCRIPTION



Table of Contents

[Document Changes 1](#_Toc110116973)

[Purpose and main design requirements 2](#_Toc110116974)

[Detailed design explanation 3](#_Toc110116975)

[Design overview of the data path. 3](#_Toc110116976)

[Design overview of the power path. 3](#_Toc110116977)

# Document Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Document Version | Date | Author | Changes |
| V1.0 | 2022.07.30 | Nikola Manolov | Document Creation |

# Purpose and main design requirements

The USB Power Injector is an open source PCB designed by Board Outline.

Its intent is simple – Allow a convenient way to power up USB 2.0 devices from a lab power supply. When powering devices from a standard USB 2.0 port, either from a PC or from a standard USB hub, you never have the niceties that an adjustable lab style power supply has. During development that means you can’t set precise voltage and you can’t set current limits. In the best case scenario, that means you can’t test under/over voltage protections. In the worst case, you run excessive current trough your device and you don’t even know that’s happened. USB hubs tend to have overcurrent protection, but most don’t send the overcurrent event to the operating system. That can cause confusion and loss of time for a hardware engineer during development or to testing personnel during production.

When you get to that point its common to sacrifice a standard USB cable so you can connect an amp meter or to modify your board in a way that allows a bodged external supply to be used. Neither is pretty, quick or convenient.

The USB Power Injector aims to remedy that by powering your device from two standard 4mm banana plugs, while still passing data between the host and device USB ports.  
The device needs to be as transparent as possible for both the USB Data signals and for the power signals.

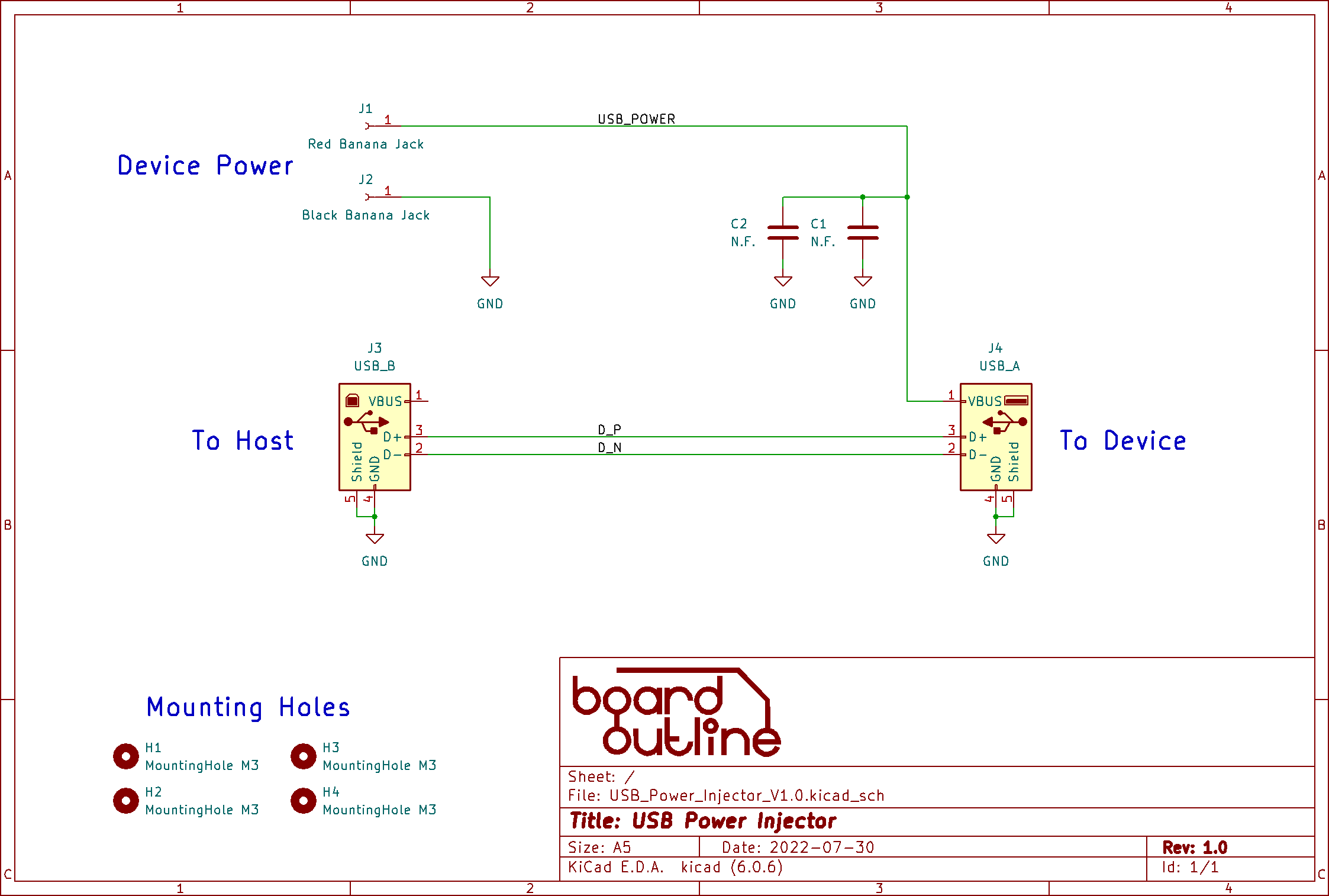


Figure 1. Schematic

# Detailed design explanation

The schematic is shown on Figure 1. The board was designed using KiCad 6. It’s a free and open source software, so anyone can read or edit the original design files.

## Design overview of the data path.

Connectors used for the data part are a horizontal USB B for the connection to the host and a horizontal USB A connector for the connection to the device under test. In order to get the required 90Ω impedance for USB lines, a 4 layer board was used. Stack used was JLC’s JLC7628. A conservative Space/Width of 5/8.1mils (0.127mm/0.2mm) was used, in order to get higher yield during manufacturing. As per the transparency requirement, no series or parallel components were used.

USB A and B connectors were used instead of two USB C connectors for the following reasons:

1. USB A and B connectors are physically stronger than USB C.
2. If two USB C connectors were used, the user could accidently connect his lab power supply to the power pins of the host USB port. That is potentially dangerous.

## Design overview of the power path.

Horizontal 4mm banana plugs were used in order to match the cable orientation of the USB ports/cables. Thus making the board less lightly to be moved by the cables by accident. The plugs are color coded – red for positive and black for negative. On the PCB they are positioned in the standard ¾ of an inch (19.04mm), so the user can use any standard banana plug accessories he may require.

As per the transparency requirement, no series elements, like diodes, were added. This allows the user to test any reverse polarity protection that may be present on the device under test. Parallel to the power line, unpopulated footprints for two 1210 capacitors were added close to the USB A connector. Their intent is to give the used the opportunity to add capacitors that would compensate the inductance of the power cables to the power supply. They are not populated in order to make sure the power injector does not improve the power delivery system of the device under test and making it seem like its working as intended, when in fact it is not.