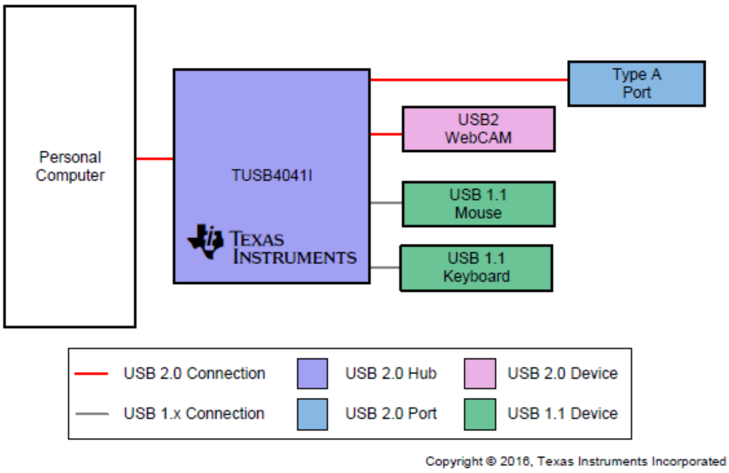
Balthazar IO board and some updates...

IO-board basically means connectivity with peripheral devices. These can be seen as on-board and out-board. On-board devices include: keyboard + trackpad, webcam and optional audio-card. USB is the usual solution for this. Here for the on-board periphery a simple usb 2.0 hub is enough. Main mini-computer board usually provides a much better hub for 4 downstream usb devices. At this point this is still mainly USB 2.0. Of course, some type of mass storage / system disk would be needed: microSD – and especially versions of SSD (sata or pcie) – directly or via usb adapter. This would benefit a faster usb port – USB3.x (part of the mini-computer system-on-chip) – or PCIE bus.



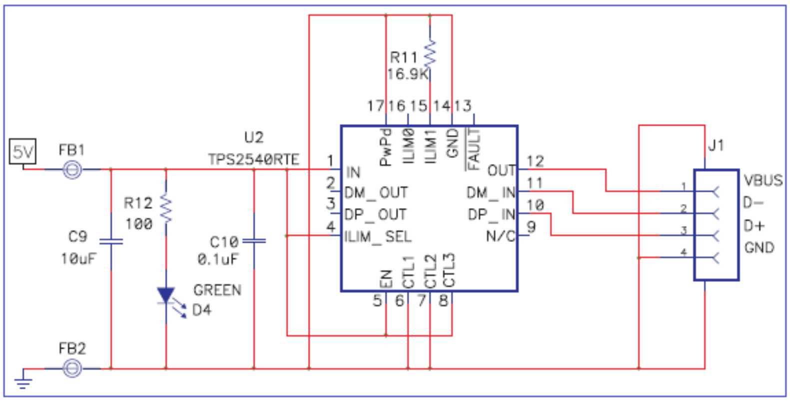
For internal periphery additional 4-port usb hub is ok - to stay on the low side (of price). USB hubs are abundant and quite cheap, but they miss some security option, which is claimed by the Balthazar project. System software can have control over the "passive" peripheral devices, but when a "smart" device is connected (a smartphone charging) or you plugged-in into a public charging station– a lot can be going on in the background. Rarely anything peripheral is not smart today.

For this an usb datalines switch would be a solution - something called usb "condom" (or data protector, or data blocker). The usb condom achieves this by blocking the data pins in the usb cable and allowing only power to flow through. However, even a charger and computer communicate during charging – defining the charging current. Below are some solutions.

Troopers 16 USB Condom

https://insinuator.net/2016/03/troopers-16-usb-condom/
https://www.electroschematics.com/diy-usb-condom-circuit/

TPS2540 from Texas Instruments can be used for condom task because TPS2540 can handle both divider mode and BC1.2 mode. Divider mode is used to charge Apple devices, BC1.2 mode is used to charge any BC1.2 mode device (Android phones, Blackberry phones and other compliant devices). Inside the iPhone the charging circuit looks for a specific voltage on the D+ and D– lines to ensure that it can pull the full amount of current. If it does not see these voltages it only pulls 100 mA. Observe below that usb datalines DM_OUT and DP_OUT are not connected.



Good USB – Protecting Your Ports With Two Microcontrollers

https://hackaday.com/tag/usb-condom/

BadUSB exploit: all USB devices rely on a microcontroller to handle the peripheral-side of USB communications. The computer doesn't care which microcontroller, nor does it have a way of knowing it. BadUSB is an attack that adds malicious functionality to this microcontroller - so it could set up an additional network interface, forward all your traffic to the attacker's server - and still keep serving up all those files and documents on the drive. Until now, there is no cure or fix for a device using an implementation of BadUSB. Robert Fisk came up with the first prophylactic USB device. The basic design of the system goes like this: take an ARM microcontroller with a USB host port, take another microcontroller with a USB device port, and have these devices talk to each other over SPI. The command protocol between these two microcontrollers is very simple and decreases the attack surface.

Conclusions:

Internal usb devices can be switched of manually. This helps with the privacy – the microphone and webcam. For external usb devices disabling the usb power line is the same as unplugging the usb device. The condom approach with disabling the data lines is a bit better, but solves things mostly with usb plugging into public charging stations. The TPS2540 solution above is usb charging current negotiator – with no datalines connected. A dedicated (micro-usb) charging port solves this problem. The already developed Balthazar PSU-charger has this option.

The job of preventing uncontrolled usb traffic from the external peripheral device which runs its own software system is problematic and should probably be done on the level of usb driver.

USB hubs

Rather than spend time and money - and develop our own IO board / USB hub – there is a host of available solutions. The TUSB4041I block schema below shows our internal needs: 1 x USB1.1 (mouse/ trackpad and keyboard), USB2.0 webcam and optional audiocard – and one more left free. The cheap ones available are based on these chips: GL850G (ssop28, lqfp-48, qfn-28), FE1.1.

Ozeki USB HUB / GL850G / less than 1$
http://www.ozeki.hu/index.php?owpn=1522

Opensource USB HUB / GL850G
https://hackaday.io/project/13240-opensource-usb-hub

Micro-USB-Hub : User's guide / GL850G
https://www.yoctopuce.com/EN/products/micro-usb-hub/doc/MHUB0000.usermanual.html

Easy DIY Tiny USB Hub For Raspberry Pi Projects / FE1.1 < 1$
https://www.retrocution.com/2020/01/15/easy-diy-tiny-usb-hub-for-raspberry-pi-projects/

Comment: The FE1.1 has just a single transaction translator. When using multiple usb 1.1 devices in parallel this is not a good idea as it slows the transfers down unnecessarily. I suggest the GL852G instead, is has multiple transaction translators.

USB2504 Four-Port USB 2.0 Hub with per-port power control (*Microchip USB2504-JT*) / $3
http://demandperipherals.com/cards/hub4.html

Cypress USB 2.0 Hub (HX2VL, CY7C65632) / about $3.50

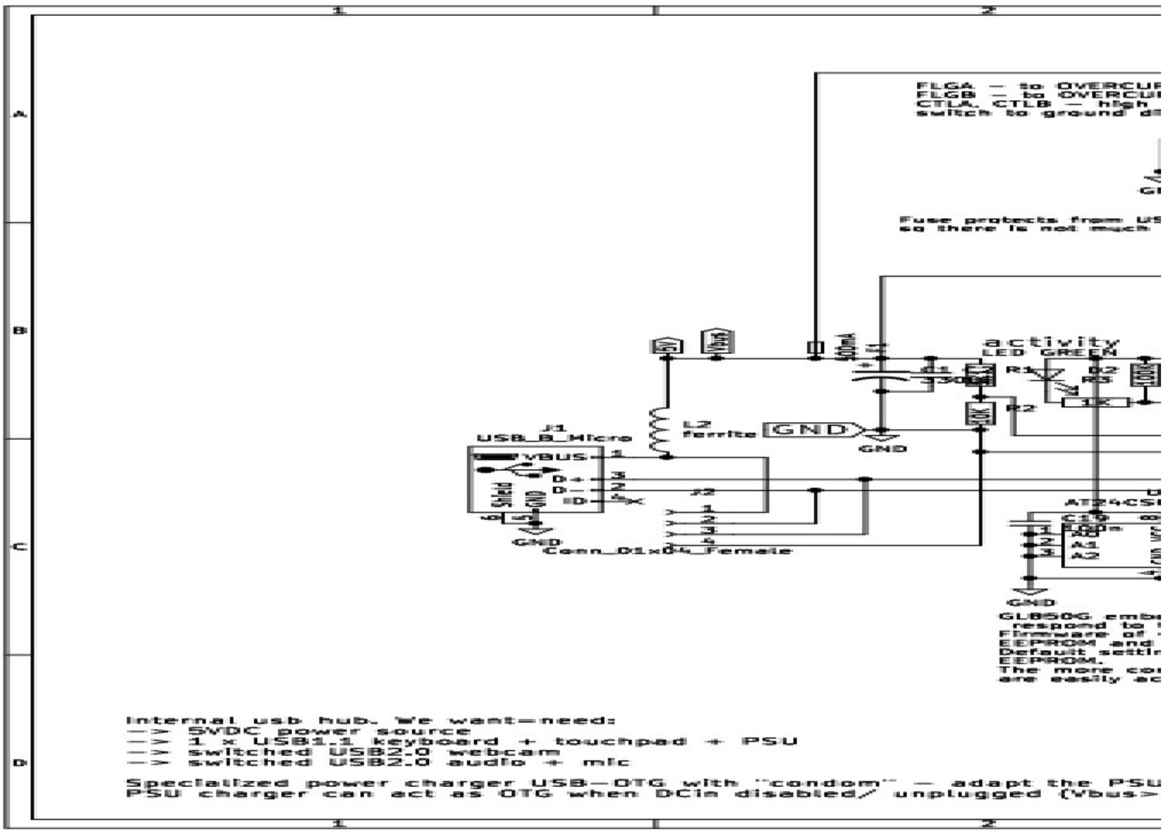
Texas Instruments TUSB4041I Four-Port USB 2.0 Hub (JULY 2015–REVISED SEPTEMBER 2017) / 64-HTQFP / about 5$

Texas Instruments TPS2044, TPS2054 – quad power-distribution switches – $2

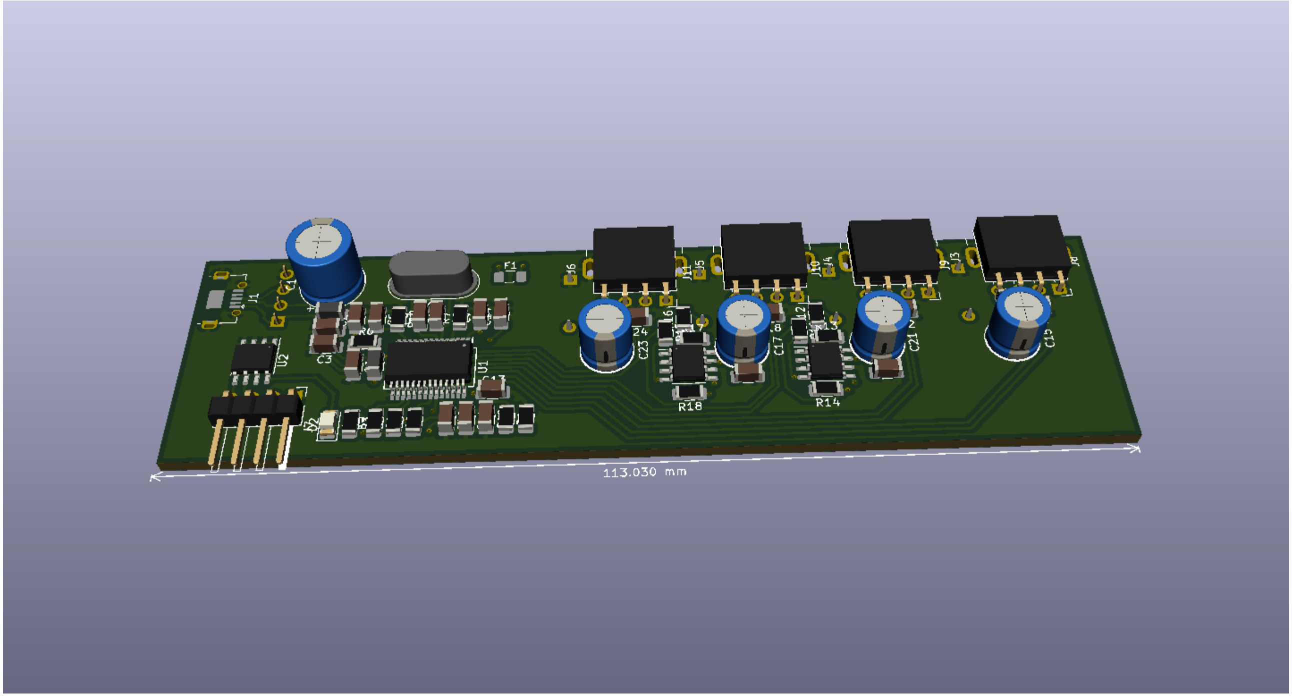
AIC AIC1526-0 – dual USB High-Side Power Switch – $0.6

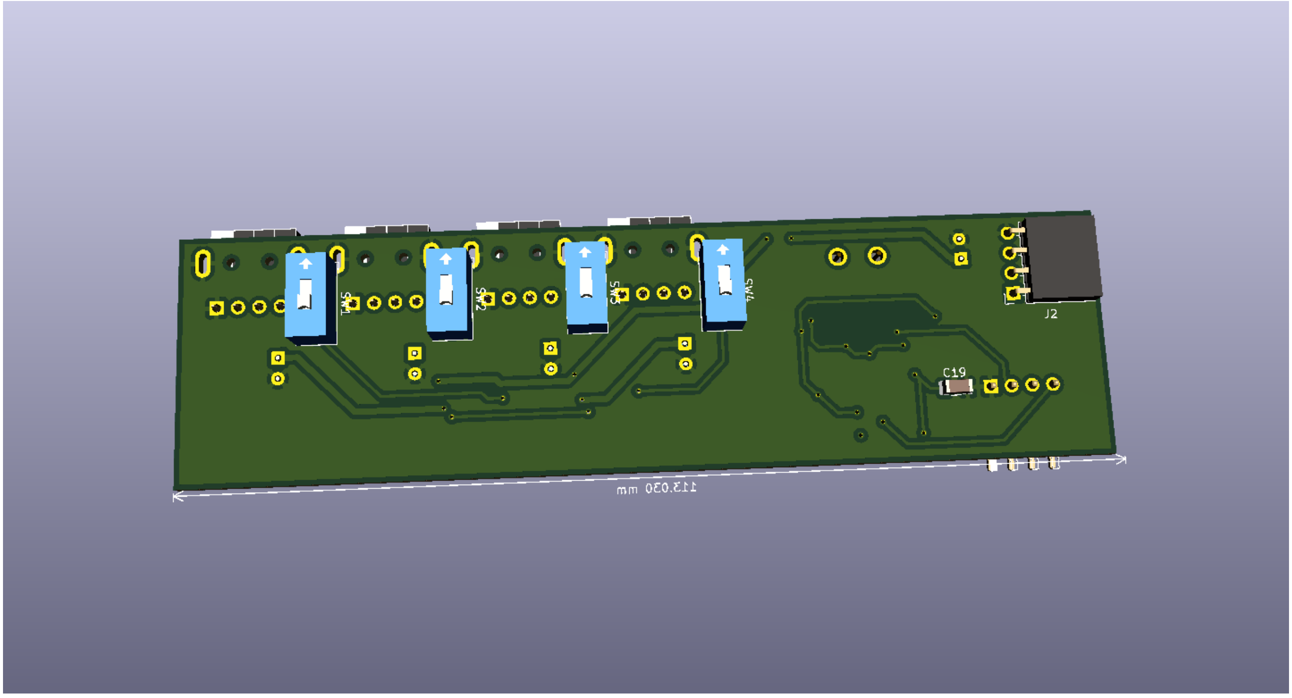
Our solution for internal hub:

We chose the GL850G (ssop28-pin) four-port USB 2.0 hub chip with additional two pieces of AIC1526-0 dual usb high-side switch – to switch-off/on manually. It is a well-proven design. Hub will be powered via the upstream port – connected to the mini-computer board. Internal usb periphery power demands are predictable = low, so the usual 500mA limit for the hub is ok. The keyboard + touchpad combination is USB1.1 protocol, webcam and optional audiocard are USB2.0.



First version of PCB looks like this: (top view, bottom view)

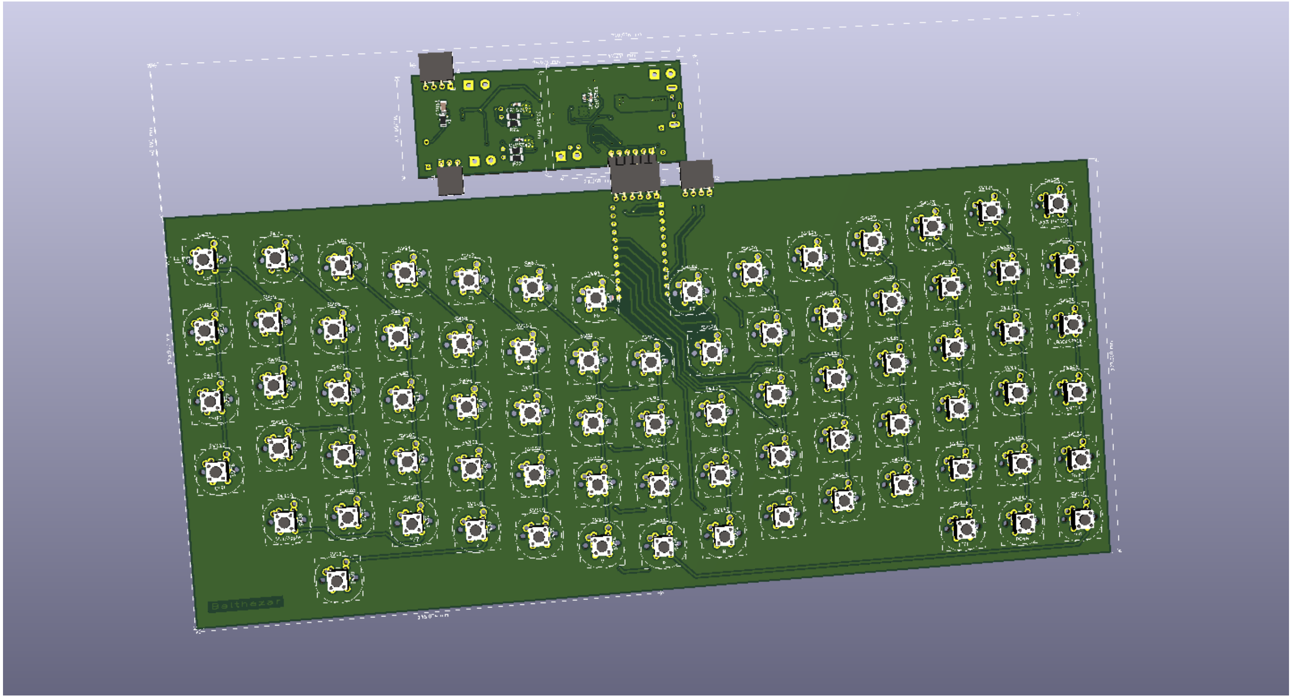




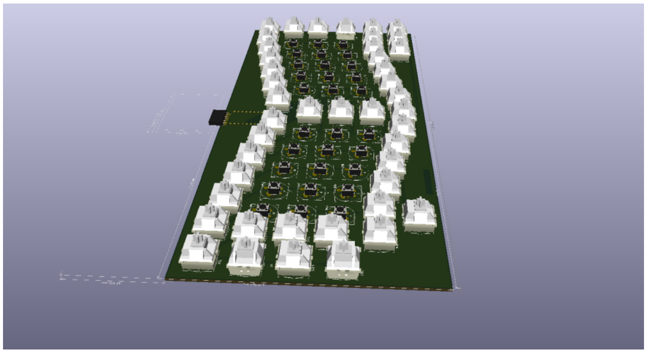
Blue switches on the bottom can of course be on the top – depends what needed. Connectors can be wires soldered – or can be regular USB type A connectors. Eeprom is an option – probably we'll get rid of it. The USB electronic switches might also be reduced to just one.

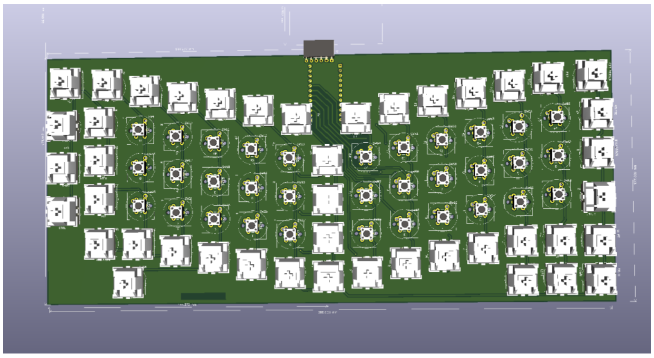
This is the third board in Balthazar series. The two previous are depicted in KiCAD 3D view below. The positions of interconnections are still provisional – and will change once some decisions are made. The small PSU charger board is about 31mm wide – almost the same as IO_USB board. Length is 84mm + the length of IO_USB is 113mm = 197mm. The Keyboard pcb is about 290mm.

The IO_USB and PSU_charger can be one board. Of course, the USB micro charging port has to be on one of the sides – not in the center. The keyboard PCB is now much more more consistently geometric.

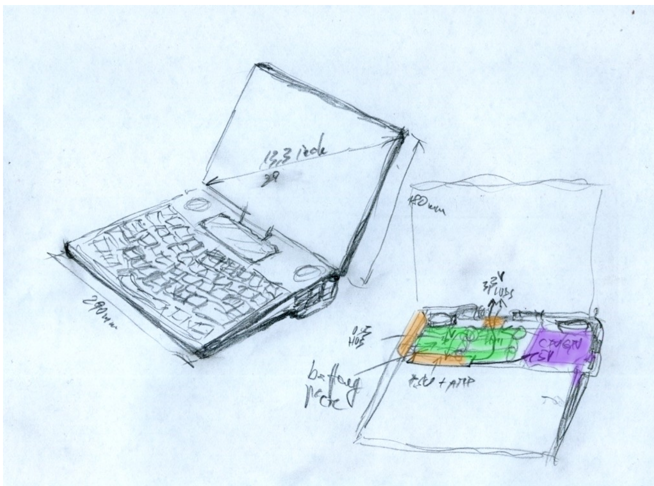


Maybe it could even accomodate some more serious keys! (not very likely...)





Next steps



Coming together with the three boards in some simple interconnected fashion. The touchpad module in the centre above the keyboard. Left, right buttons to the left and right of touchpad. Manual switches to disable / enable devices on left or right of the touchpad. Enough of place for the 12V/ 3A mains adapter and a LiIo or LiPo battery pack. Or an option for a double-sized battery pack and external mains adapter. On top also place for left & right loudspeaker, probably in the top center microphone and webcam. Somehere at the gimble the HDMI to LVDS adapter fo LCD.

A kind of high-heel shoe profile?