The GRUB Switch
Custom Hardware Rev. B

Ruediger Willenberg
Version 1.1
June 16, 2021

Table of Contents

1 GRUB Switch sources (incl. hardware)

The executable or compilable GRUB Switch files and sources are designed to be installed and used under Linux, however the files for our custom PCB (EAGLE files, Gerber files and documentation images, PDFs and Open Document files) are relative platform-independent.

There are two ways to obtain the files:

- Via **git clone**:
  - Go to your preferred working directory
  - Clone the repository with

    ```
    git clone  https://github.com/rw-hsma-fpga/grub-switch.git
    ```

  - All GRUB Switch files are now available in the path

    ```
    ./grub-switch
    ```

- Via **download**:
  - Download to your preferred directory: https://github.com/rw-hsma-fpga/grub-switch/archive/refs/heads/master.zip
  - Unzip with GUI-tool or on the command line with

    ```
    unzip grub-switch-master.zip
    ```

  - All GRUB Switch files are now available in the path

    ```
    ./grub-switch-master
    ```

All further examples will assume you are starting in the parent directory above **`grub-switch`**

All data directly related to the PCB design and components are in

`grub-switch/3_custom_hardware/`

while this document itself is in

`grub-switch/documentation/`

2 Introduction

2.1 Motivation

The GRUB Switch Custom HW (Revision B) was designed with an eye on the following unique features:

- An independent, open source, PCB design that can be self-assembled
- Firmware programmable via USB with the factory-state ATmega32u4 MCU, without having to install a custom bootlader.
- Featuring soldering pads for the two most useful electromechanical switch options, common 1x12 rotary switches and DPDT toggle switches. Both commonly offer mounting threads, so the entire PCB can be mounted directly into a front panel or PC case. No cable or wire soldering required.
- Regular USB socket, but also the option to directly connect internally to a PC motherboard, pin header-to-pin header via standard ribbon cable.
- Write-protect switch or jumper to block tampering with the boot configuration

On top of this, the board can be used for any other embedded design purposes separate from GRUB Switch, as 13 I/O pins are exposed as solder pads. In principle, any valid *ATmega32u4* firmware should be able to run (see last section).

2.2 File structure

There are 3 subdirectories in **`grub-switch/3_custom_hardware`**:

`./bill_of_materials`: Complete component list in ODS (Open Document Spreadsheet) and PDF formats, with multiple alternatives for some components. Order codes for two international distributors are included for convenience, but *do not* constitute an endorsement.

`./eagle_files`: Autodesk EAGLE **`*.brd`** and **`*.sch`** files, saved with version 9.6.2, but likely downwards compatible to at least version 8.x. Size and features conform to be viewable and editable with a free version of EAGLE. All components included without library files.

Further, PDF versions of the schematic as well as top and bottom layer views with dimension, pads, vias, components and silkscreen.

`./gerber_files`: Gerber RS274-X and NC drill files that should work with most low-cost PCB manufacturers.

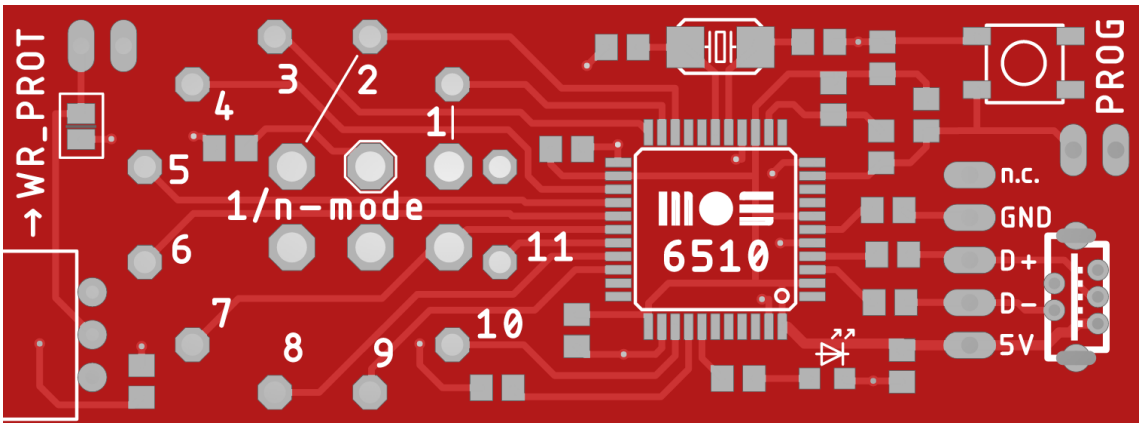
2.3 Description

The PCB board is a 2-layer (top/bottom) design with external dimensions of 68x25mm. It it designed for standard 1.6mm board thickness; variations should be unproblematic, but in that case keep an eye on the mounting depth of the rotary or toggle switches as well as the *Molex* USB-Mini-B receptacle.

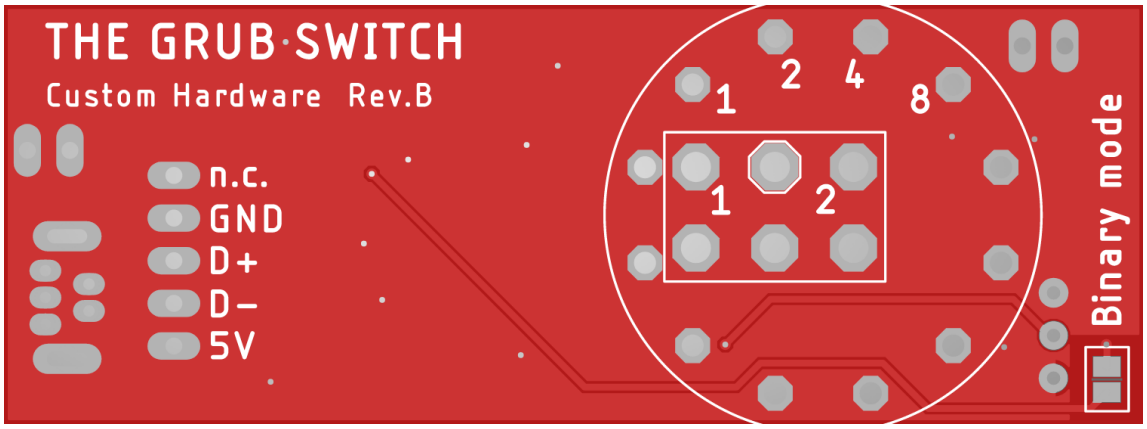
The board has no wire with less than 8mil width and only fully drilled, round solder pads and vias, with the exception of two oval mounting pads for the Molex USB Mini-B receptacle.

All components except for the rotary or toggle switches are mounted on the top layer side.

Top layer view:



Bottom layer view:



3 Components info

Also see the *bill of materials* for details and order options.

3.1 Single-manufacturer/proprietary components

3.1.1 Atmel/Microchip ATmega32U4-AU (essential)

The MCU being used in this design is the version in a **TQFP-44** package **without** an internal RC oscillator.

Do **NOT** use the ATmega32U4**RC**-AU version because it has different fuse settings and the factory-state firmware programming will not work with our board.

Do **NOT** use the ATmega32U4**-MU** version because it has a different package size and won't fit the footprint. It also does not have the pre-installed bootloader.

3.1.2 Molex USB Mini-B receptacle, through-hole, vertical, Part No. 502237-0517 (optional)

This is another entirely proprietary component, with no other vendor offering a compatible mounting and solder pad pattern. However it is not an essential component as you can use the 1x5 pinheader next to it to connect to USB, especially on internally mounted designs (also see the section on saving part costs).

3.1.3 Würth WS-SLTU THT Mini Slide Switch for Write-Protect (optional)

While the board is designed for this right-angled mini slide-switch to fit snugly on the PCB side, its through-hole pads have a common 2.54mm (1/10") pitch and might be replacable with other small switches (be aware of the 0,8mm padhole diameter).
Alternatively, the board has a parallel 1x2 pinheader that can be used for a Write-Protect jumper instead (also see the section on saving part costs).

3.2 Non-proprietary components

3.2.1 Crystal 16Mhz

Our bill of materials already lists 5 commonly available models, but there are likely others that fit the required specs:

- Frequency: 16Mhz, load capacitance: 18pF
- Size 5.0x3.2mm with 2 SMD pads (careful, there are incompatible 4-pad versions!)
- Both tolerance and ESR variations for other available models should not be critical.

3.2.2 PTS647 push button / tactile switch 4.5x4.5mm

There are very likely other switches with a compatible footprint. This switch is also replacable by another solution to short the 1x2 pin header contacts labeled **PROG**.

3.2.3 Osram LG R971-KN-1 ChipLED (SMD-0805)

The LED can be replace by any other 0805 format chip LED, but take care to choose the right series resistor value for R8 (currently 1k) to get the proper forward current for the LED. The pin voltage level for ON is 5V. While the I/O pins can source double-digit mA currents, we suggest staying in the single-digit mA range.

3.2.4 Resistors and Capacitors (SMD-0805)

There are, of course, plenty of other 0805 format resistors and capacitors you can use; respect the parameters for value, tolerance, dielectric material and maximum voltage in the *bill of materials*, though.

3.2.5 Boot choice switches

See the *bill of materials* and section on switch wiring below for details

3.3 Saving money on component cost

The most expensive components that are essential are the PCB (see comments below), the MCU, the crystal and the rotary or toggle switches you use. Nothing to be done there.

PCB manufacturing in East Asia is actually really cheap, but the per-board cost is only small if you need multiple boards and/or can minimize the shipping costs (which means waiting longer). Maybe you can pool an order with other people.

Expensive components that are non-essential are:

- The Molex USB Mini-B receptacle. You can use the 1x5 USB pinheader and a ribbon cable, especially if you're wiring internally to a PC motherboard.
- The Würth slide switch – you can use the 1x2 pinheader and jumper it for Write-Protect, or even short the corresponding solder bridge if you don't want to ever change your configuration again (sure??)
- You can use the **PROG** 1x2 pin header pads (w/ or w/o pinheader) to enter bootloader mode instead of the push button. Make sure to disconnect cleanly to get the proper **/RST** and **/HWB** timing.
- Who needs an LED indicator if everything works as advertised?

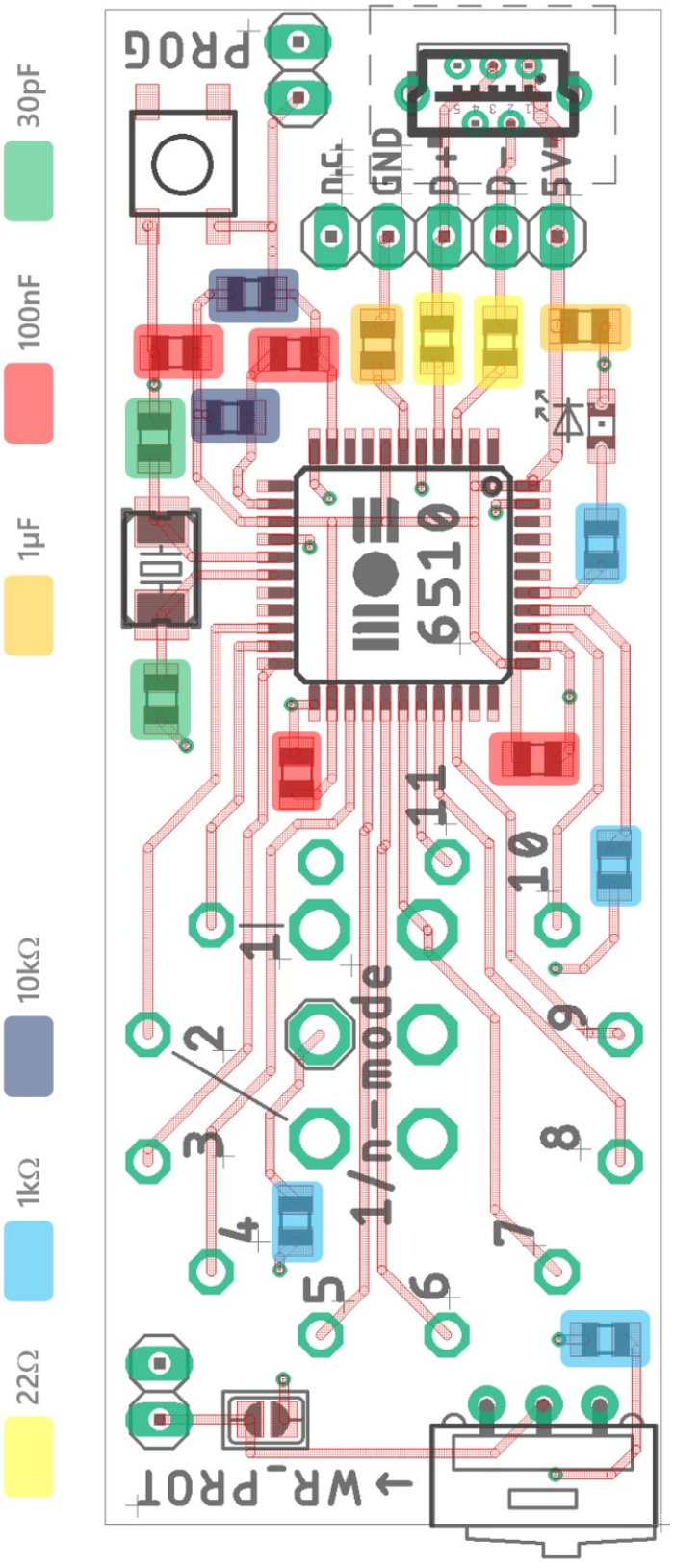
4 Placement and soldering info

While we tried to make components as big as possible without compromising too much on PCB size, at least a magnifier and possibly a stereo soldering microscope is useful.

We recommend to place and solder the components in the following order:

1. The ATmega MCU: Make sure that the pin 1 marker is in the right spot (see next page) and that no two pins are shorted with solder. Copper soldering mesh is great to draw excess solder out again under heat.
2. The crystal. This is a bit hard because part some of the solder is supposed to be on the side of the case and some is supposed to flow under the case. Put only a small dollop of solder on one pad, then place the component symmetrically with pincers while heating the solder again. When the solder cooled and the crystal is fixed, solder the second pad.
3. The resistors (see color-coded placement pattern on the following page)
4. The capacitors (see image again)
5. The LED. Check twice for anode/cathode direction.
6. The WR_PROT slide switch (if required). Nothing ist standing far out yet, so you can solder the pins on the bottom layer side while the switch is pressed flat between the table and the board. Center pins well inside the pad hole so the slide is snug on the PCB side.
7. The Mini-USB Mini-B receptacle (if required). Make sure all 5 USB contacts go through the holes. Solder the high-mass (lots of heat) mechanical contacts first, then the USB contacts.
8. The PROG push button
9. Any pin headers (USB 1x5, WR_PROT 1x2, PROG 1x2) you require.
10. See the section on boot switch options at the end for

It is fairly simple to adapt the firmware to another board with the ATmega32u4 and build a firmware image. See the detailed documentation for instructions.


30pF
100nF
1µF
10kΩ
1kΩ
22Ω
PROG
n.c.
GND
D+
D−
5V
6510
1/n-mode
WR_PROT

5 Programming and configuring the board

5.1 Flashing the firmware onto the chip

On the Linux command-line, go to the **grub-switch/2_usb_device/prebuilt_images** directory, where you can find the suitable image and programming scripts.

Our custom USB device design relies on the *DFU bootloader* which is factory-built into each ATmega32u4-AU chip. There is an open source command-line tool called **dfu-programmer** which needs to be installed for firmware flashing. Under Ubuntu Linux, it can be installed with

```
sudo apt-get install dfu-programmer
```

To simplify the use of **dfu-programmer**, we have provided a script in **prebuilt_images**. After plugging in your board, it can be started with

```
sudo ./program_grubswitch_dfu.sh GRUB_SWITCH_CUSTOMHW_REVB.hex
```

(The use of *sudo* might not be required on your installation)

Before or after starting the script, the chip needs to be brought into *bootloader mode* by pushing or shorting the **PROG** button. Pressing a key on the keyboard will start programming.

After successful firmware programming, the board identifies as a FAT12 storage device with the drive name **BOOTTHIS**. Depending on your Linux GUI, you might get a notification.

Read the detailed USB firmware documentation in

grub-switch/documentation/2_usb_device.pdf

to learn about how to build the firmware yourself or use **dfu-programmer** directly to program the board.

5.2 Writing a GRUB Switch boot configuration onto your storage device

If you have previously used the configuration tool
grub-switch/1_shell_scripts/CONFIGURE_GRUBswitch.sh

to generate a GRUB Switch configuration (**grub-switch/bootfiles/.entries.txt**), you can copy this file onto the drive yourself, or you can use option "**4**" in the **CONFIGURE_GRUBswitch** menu to auto-mount, write the file and unmount the drive again.

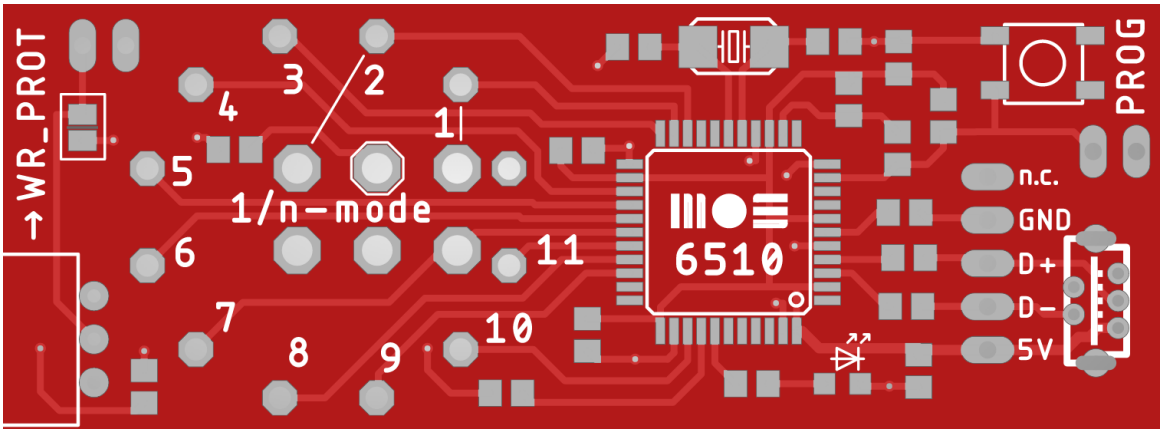
6 Boot switch variations

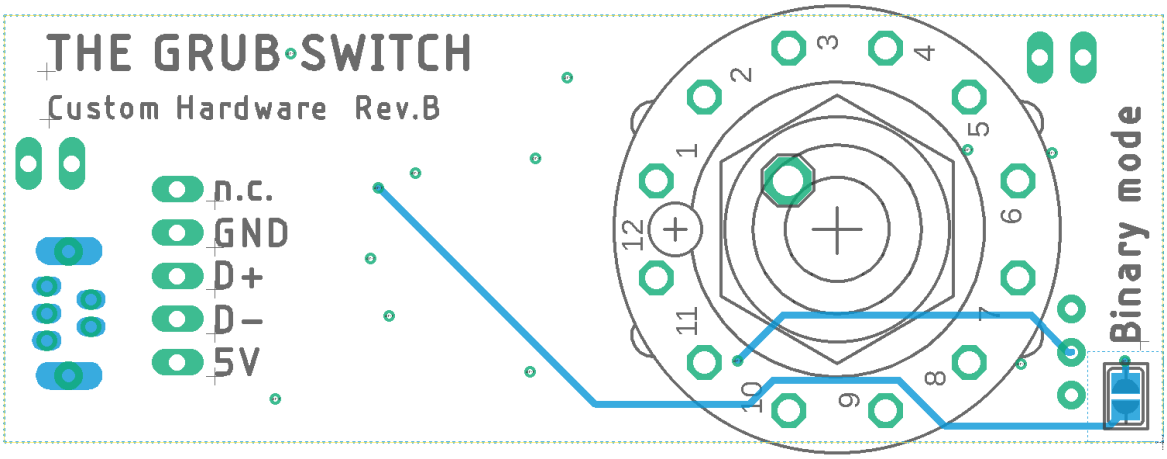
Our custom PCB has been designed with an eye to directly mounting and soldering two common types of switches for **1-of-n mode** (shorting one single choice pin to **0V/GND** at a time). Because both types of switches are available with a screw thread for front-panel mounting, the PCB and switch can be affixed mechanically to a case.

6.1 *1-on-n mode* with 1x12 rotary switch

The images below show that a common **1-of-12 rotary switch** with through-hole PCB pins is mounted on the bottom side and then soldered into the circular pattern on the top layer side.

The silkscreen shows both the switch's central input pin (surrounded by an octagon) and the 11 choice outputs of increasing order, one of which can be connected to the input at a time. Because the switch is mounted on the other PCB side, choices increase by turning the switch clockwise. There is also a 12th, unlabeled output pin, which we might consider position 0. If the switch is turned into this position, no choice pin on the MCU is connected and GRUB Switch yields to the Menu.





Compatible rotary switches with various actuator shapes and lengths, as well as metric and imperial mounting holes and threads, are available from multiple vendors:

C&K A112..*.NC.** E-Switch KC52.*.***.**.NP.***

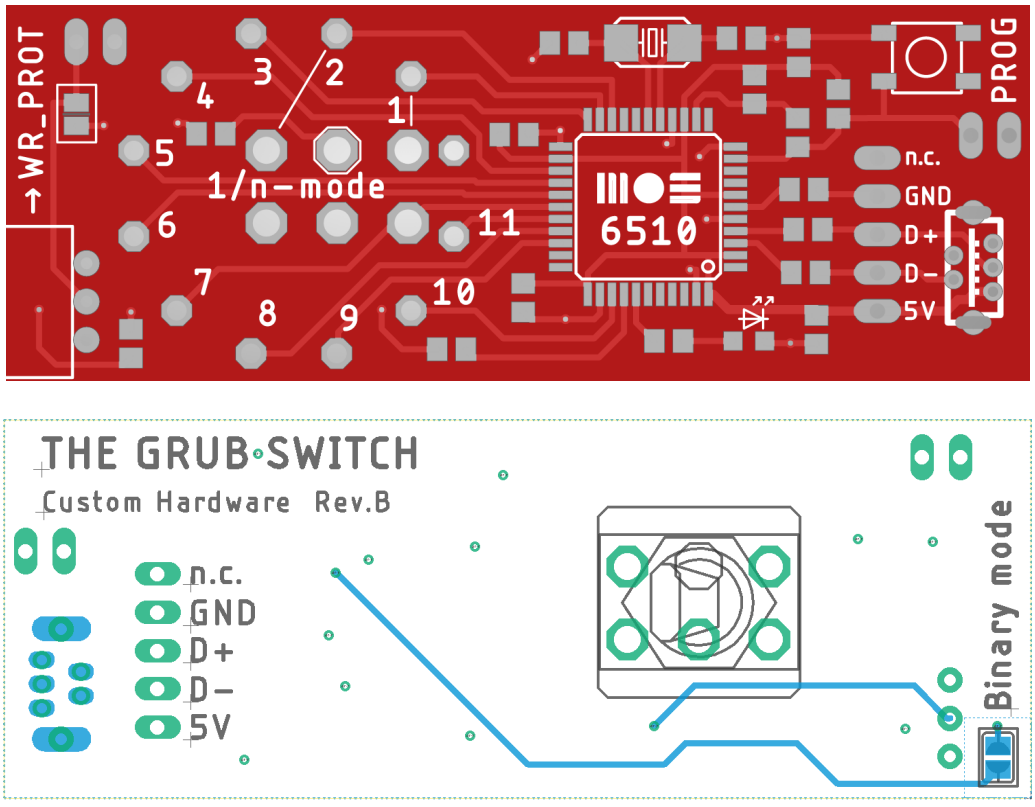
LORLIN CK1049/CK1059 (metric), CK1044/CK1054 (imperial)

(these suggestions should not be considered a commercial endorsement)

Make sure to order the 1x12 variation, as there are also 2x6, 3x4 and 4x3 models. Even if you don't need as many choices, these other models cannot be used because they have additional input pins on the inner ring, while our PCB does not. To limit the 1x12 switch to fewer reachable positions, most switches come with a metal keyring that can be inserted before tightening the screw and that will limit the range of movement; so if you only need, say, 4 boot choices (including the unconnected GRUB menu option), the range of movement can be limited to 4 positions.

6.2 *1-on-n mode* with on/off/on toggle switch

The bottom layer picture below shows the mounting position for a **two-channel on/off/on toggle switch** (DPDT - "double-pole/double-throw"). Only the upper channel is actually used; the two channel model was chosen to decrease mechanical stress on solder pads and increase stability.



The pad numbers 1 and 2 on the top level silkscreen indicate the boot choices made through electrical connection to the center pad (silkscreen octagon). Note that the toggle switch lever moves across a fixed pivot, so moving the lever to the left (as seen from the switch side) actually connects choice 2, while moving the lever to the right chooses boot option 1. The neutral center position makes no connection and therefore leads to the GRUB menu.

The following toggle switch series include DPDT on/off/on models with the required 4.7-4.75mm distance between PCB pins:

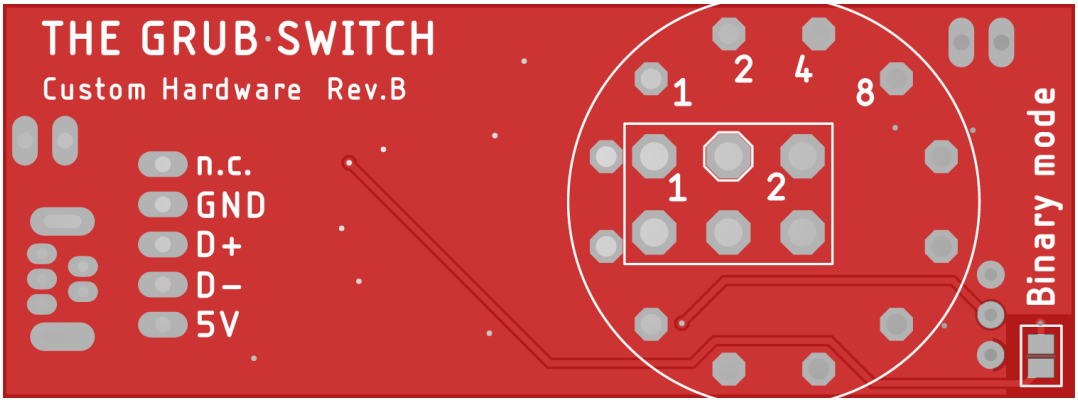
APEM Series 5000 **E-Switch Series 100**

NKK Series M **TE-Connectivity Gemini A**

(these suggestions should not be considered a commercial endorsement)

6.3 Binary mode

The Custom PCB isn't designed to mechanically hold a binary-encoding switch, but binary mode can be selected anyway by shorting the solder bridge marked with **Binary mode** on the bottom side:



The silkscreen numbers 1, 2, 4, 8 on top mark the pads to connect to 0V/GND to choose the corresponding power of 2. The 0V/GND level is available at the pad marked with an octagon silhouette. Connecting all 4 pins to GND, results in a choice of 1+2+4+8 = 15. If none of the pins is connected, the choice value is 0, and therefore the GRUB menu is shown.

7 Using the Custom HW board for other purposes

Obviously, the board can be programmed with other *ATmega32u4* firmwares that have nothing to do with GRUB Switch. 14 MCU pins are exposed through solder pads and can be easily used to provide digital I/O or alternative functions. Digital I/O is at 5V logic levels.

For details, check the ATmega32u4 datasheet as well as the schematic in **`3_custom_hardware/eagle_files/GRUBswitch_CustomHW_RevB_schematic.pdf`**

7.1 Boot choice pins

These MCU port pins are directly connected to the 11 boot choice pads:

1 - **PD3** / TXD1 / INT3
2 - **PD2** / RXD1 / AIN1 / INT2
3 - **PD5** / XCK1 / #CTS
4 - **PD4** / ICP1 / ADC8
5 - **PD6** / T1 / #OCD4 / ADC9
6 - **PD7** / T0 / OC4D / ADC10
7 - **PB4** / ADC11 / PCINT4
8 - **PB6** / PCINT6 / OC1B / OC4B / ADC13
9 - **PC6** / OC3A / #OC4A /
10 - **PF7** / ADC7 / TDI
11 - **PB5** / PCINT5 / OC1A / #OC4B / ADC12

7.2 LED Pin

The **left** SMD pad of LED series resistor **R8** is connected to port pin PF5:

LED - **PF5** / ADC5 / TMS

7.3 /WR_PROT Pin

The **right** SMD pad of resistor **R7** is connected to port pin PF6:

/WR_PROT – **PF6** / ADC6 / TDO

7.4 /BINARY_MODE Pin

The **lower** pad of the **Binary mode** solder bridge (PCB bottom layer side) is connected to port pin PB3:

/BINARY_MODE – **PB3** / PDO / MISO / PCINT3