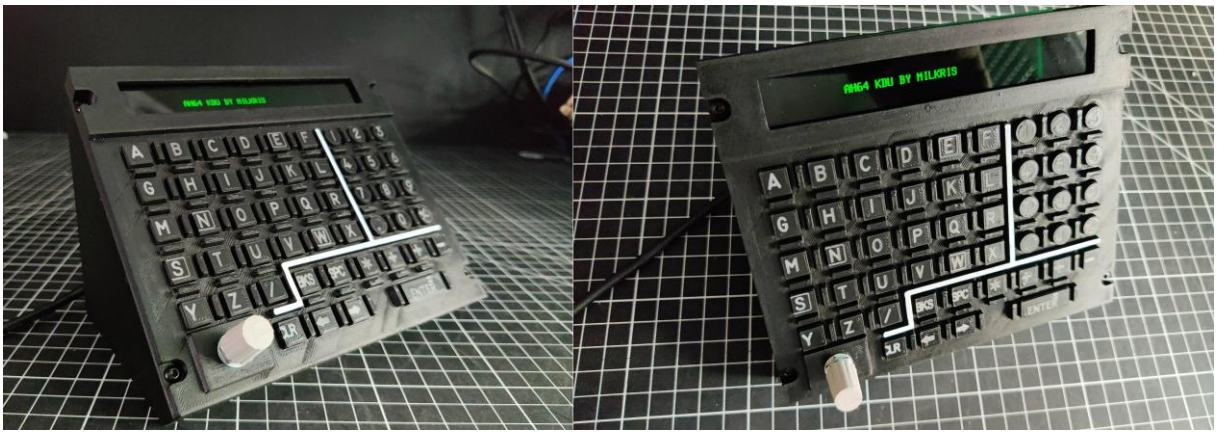
MilKris‘ Keyboard Display Unit (KDU) for the DCS AH64D Apache

DISCLAIMER: I’m not a professional maker! It’s not perfect, but it works and I’m fine with the result. Here and there, some filing, drilling, and gluing mightbe necessary...
DONATE: Feel free to say thanks... https://www.paypal.com/donate/?hosted_button_id=XG6RA9RWPM84Y



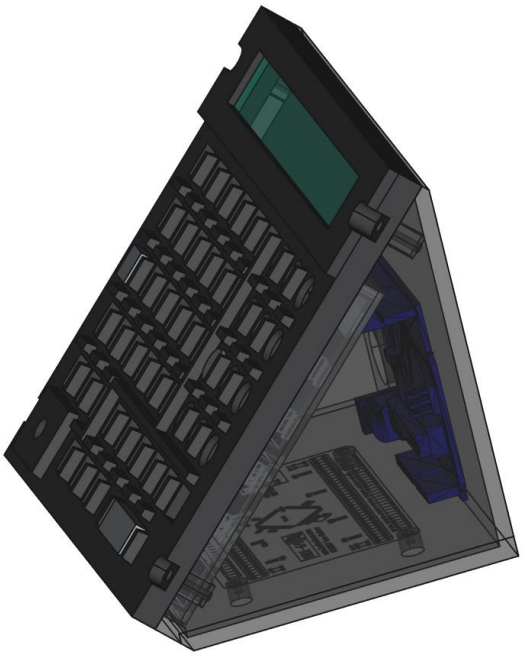
Here is my KDU (Keyboard Display Unit) for the DCS AH-64D Apache. The project includes files for 3D printing, laser engraving of the button labels, and, of course, the Arduino code.

You can switch the display source from Pilot to CP/Gunner by pressing a button. This button is wired to PIN 7. I integrated this function into the push-button of the rotary encoder for the KU Scratchpad Brightness Knob.

I engraved the labels with a laser, but I've added another version of the buttons where the labels are raised so they can be printed directly with a 3D printer.

In this project, you need to **flash my code** onto an **Arduino**. Here is my code: https://github.com/MilKris666/DCS-AH-64-KDU-Keyboard-Display-Unit/blob/main/AH64_KDU_Display_4.ino

If you have no idea **how to set up DCS BIOS** or **upload code to an Arduino**, make sure to check out this **tutorial by Hornetsnest**! https://www.youtube.com/watch?v=ZGoG54vNyyI



You need:

50x 6x6x4,3mm Tactile Switches: https://www.amazon.de/Youmile-100PACK-Schalter-Momentan-Tactile/dp/B07XWXV1TS?source=ps-sl-shoppingads-lpcontext&ref_=fplfs&smid=A3BN2T8LLIRB5S&th=1

1x Arduino UNO: https://www.amazon.de/Entwicklungskarten-Mikrocontrollern-Binghe-elektronische-Kompatibel/dp/B0DKX16PJT?dib=eyJ2IjoiMSJ9.-GyPB7HIcD3yPWLup1mEPiO0Wjg_9vk9AM3WGiiQ3ZXlRwz2utOf9JkfcthwLy0JsjpFQaMuiTxYaY7YiOCHgF603G5ZY4uaZ1h8bBS-o867a_1Q9Tb84MTR4D-trdAbmnVSXXPYqnZoSyKX0TeRXiqEHiOPOmVKBUXk8AOsp3lCFy-8L0Nz3ECF-cGFAMRimYRK52lNtGqIYe9KZfDdyTqi5ZIzOIDRMFcTg-bhJ3U.LEijxbTuxPC67ruhNkmPg6gHn0kl2ZxW8QCsaMkzSmM&dib_tag=se&keywords=arduino+uno&qid=1739455679&sr=8-17

1x SSD1322 SPI OLED: https://www.amazon.de/OLED-Display-SSD1322-Grafik-LCD-Modul-Parallel-L%C3%B6tstift-wei%C3%9F/dp/B0DQ52MRX1?source=ps-sl-shoppingads-lpcontext&ref_=fplfs&psc=1&smid=A9KVXEOZANNPE

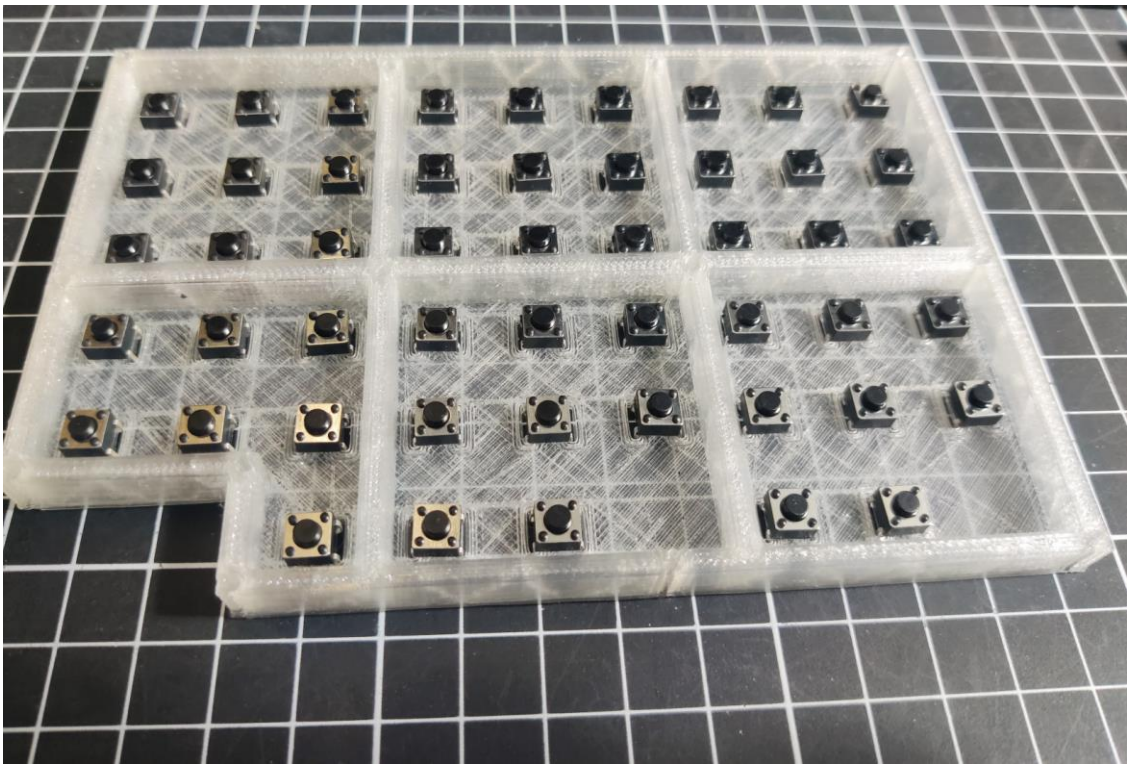
1x Rotary Encoder: https://www.amazon.de/Drehgeber-Drehwinkelgeber-Digital-Potentiometer-Encoder/dp/B0B63YN466?crid=BJCBX1ZRUE4Z&dib=eyJ2IjoiMSJ9.v4qzrG-nR_M2qWVfLskl3r3gAG7EopqaqFpVIKpxF4RteraOTkzoj6yaOKPL14TYGc2jWgus1q3SEiFL75Lyc28BYXHU2K8SeNbNwpCsJ6Vg5nMULRNwRJ6Uxj7ICXr25UlcgFOb-3piOo0v1sLLtAed3bi3OQrqieZ96s0bh6Isc40fqSFs0EnRit2HuflkEqfBKdsaTWKNQpdv2MtKj7ySbaM5aRN3hzxZqV0DbRGTKGv--0ag832N_qoXJ1pNoslYA7ILRrk8IlrCFuzK9kemtVBpTcHSolBqxi-kwKYLJnmGnGZsJKrRZmxj8P0noU8vwozXeCNM2uiJgmoJ31JzZGqgbn0GpdiFnrUbR8OdaQ8lCqI34VcyjPH3kQz2u68BhEH3EJD5XNeCAhPssJlhK9F5KphpeM_ZMFbVp-n8pHE8SaS2V0XvOyoZ1dMo.wv43IIVzUWdEnMpqotB_RK5SoSf9bwj2iXBWaIC8Qxs&dib_tag=se&keywords=rotary%2Bencoder&qid=1739520345&sprefix=rotary%2B%2Caps%2C103&sr=8-10&th=1

1x Leobodnar BBI64:
https://www.leobodnar.com/shop/index.php?main_page=product_info&products_id=300

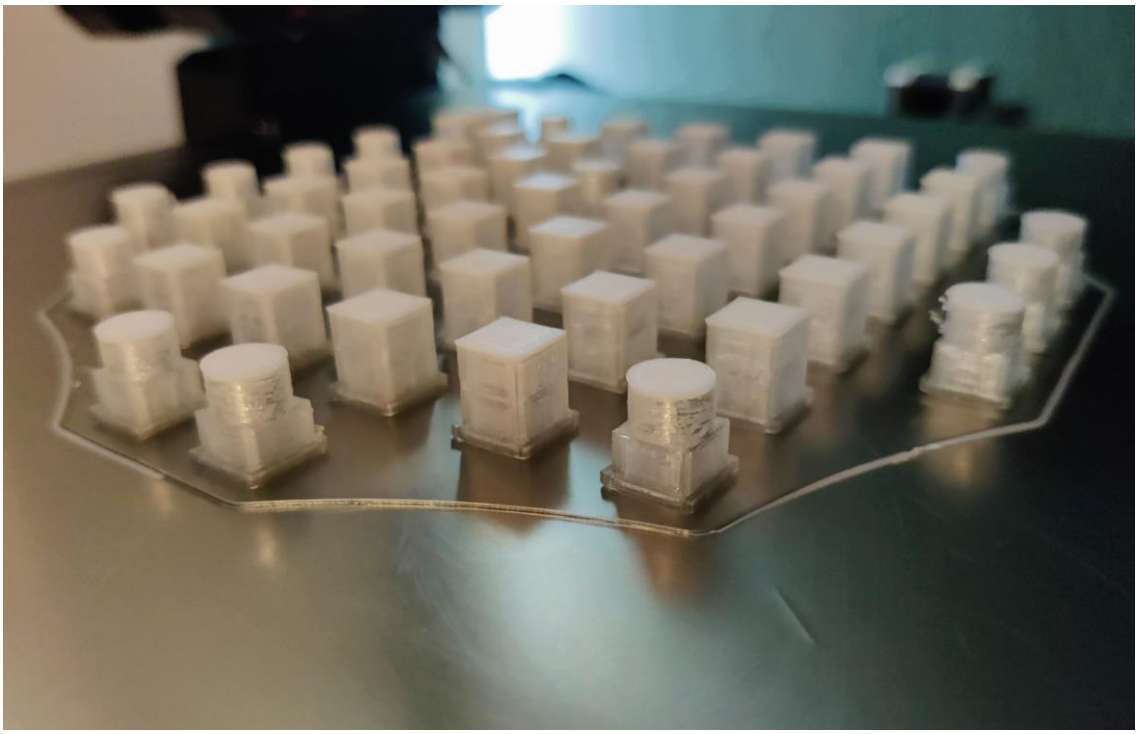
Instead of using the BBI64, you can create your own button matrix for the Arduino and extend my code. However, you might need an Arduino MEGA, as the memory could be too small.

2x green LED Filament:

https://www.amazon.de/LISOCLU-Flexibles-LED-Perlendiode-Lichtzubeh%C3%B6r-flexibles/dp/B0DGQ17CBC?__mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dib=eyJ2IjoiMSJ9.us4Y5oACoc7Vk8tlmFZ68xbk84dToLNiT2758C_uNMxvpO-_vwA9Y6i9jG-7PCsnNSwVx0Sxlv7WQq10fZOYFhu3w879jCbqXo-Y-ORUYPMZbgDdlUrU_7oPkDuGjv4DTNGXNnujj17r4uz9USDAbx6YfNPd4nAbZJZZYLFFGcn3uW8ep5ZecBCmtzVdW50ent--qQ-CyOkZNOvslpSpAtgoZY3nQMClW0Mzf0Zj4g5rqQw27ICXqFxBrfmv5sKjrEoVLaFTxNMOJOSqMytdpWu5uy43fifL_OVNZTsSb4VHM9HJDjPLyRUn3fuej3MXhCezRBJt5IEfB8FyzoDpritIx1rgfosUh3JGaqRmsI-P9MpDt8EH45LEnDPwRF5B0AaX2_3dn745xScEQbqDoa2JaSXo75Mm-cI5gKzkLawPMQL7N3aBiqfc3yb2Bxg3.3z_NiMQb6Ro_yU0_UNk8SAhppodXVby1FKetAXP1tPI&dib_tag=se&keywords=3V%2BLED-Filament%2B110mm%2BRot%2BGr%C3%BCn%2BBlau%2B2700K%2BFlexible%2BFilamentlampe&qid=1739521114&sr=8-3&th=1



This is the button matrix printed with transparent PLA. This way, it can later be backlit with green LED filament.



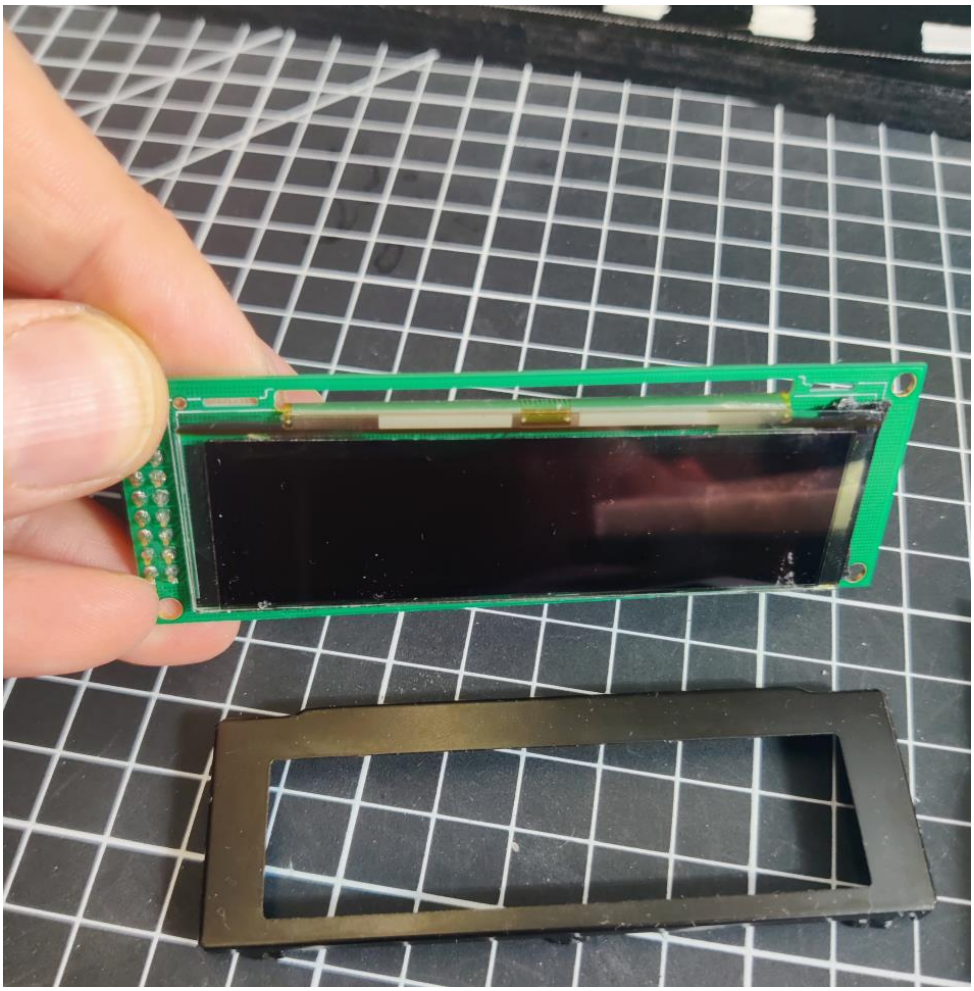
The buttons were also printed with transparent PLA. The top three layers were printed with white PLA, making them ideal for backlighting. The buttons are then **spray-painted with black acrylic lacquer** from a can and can be **engraved with a diode laser**.

If **laser engraving** is not an option for you, you can **3D print the buttons with raised lettering**. In this case, you should print the buttons with **black PLA** and use **white PLA only for the top layers**.

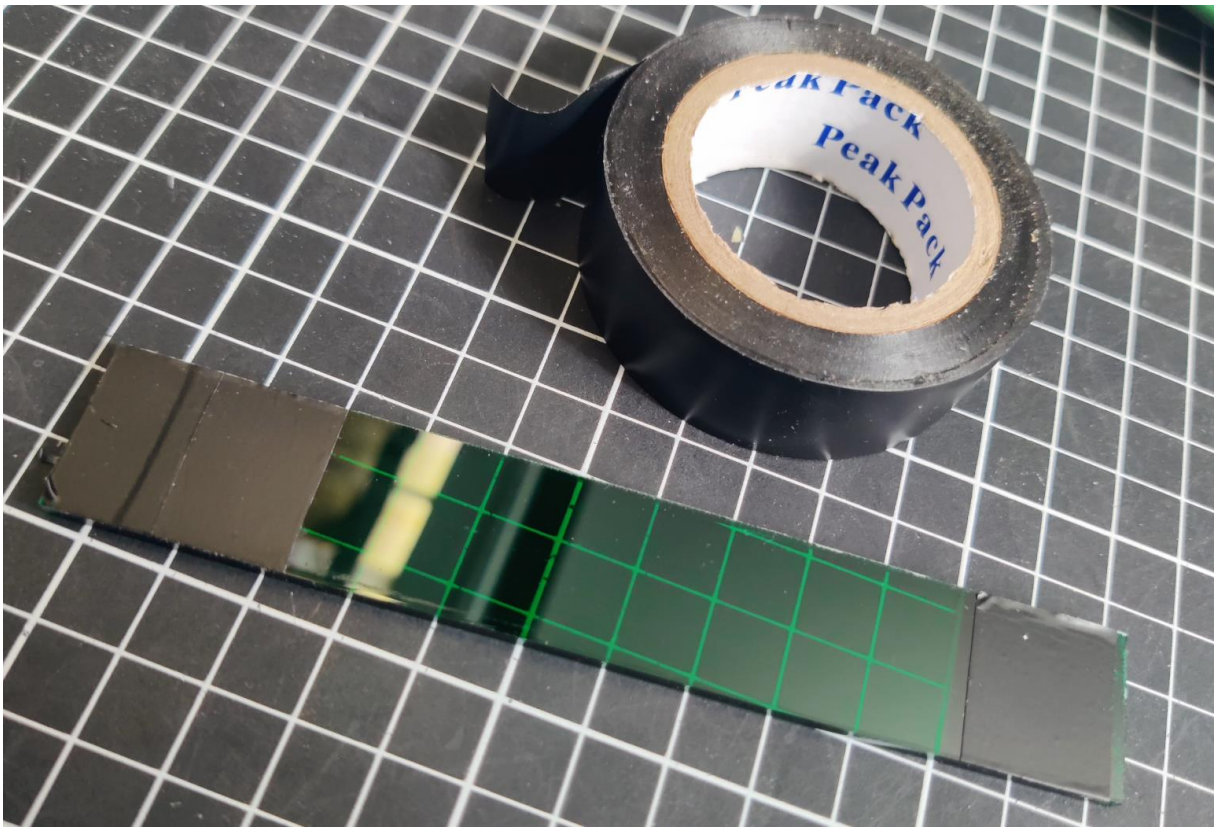
If you **don’t have a dual-extruder 3D printer**, you can set up a **Post-Process script** in **Cura** ("Pause at Layer Height") and manually switch the filament.

Alternatively, you can **print the buttons entirely in black** and carefully **paint the lettering white** using a **white marker (e.g., Edding)**. To make the finish **more durable**, you should **seal everything with a clear coat**.

You'll need to experiment a bit to find out **which method works best for you**.

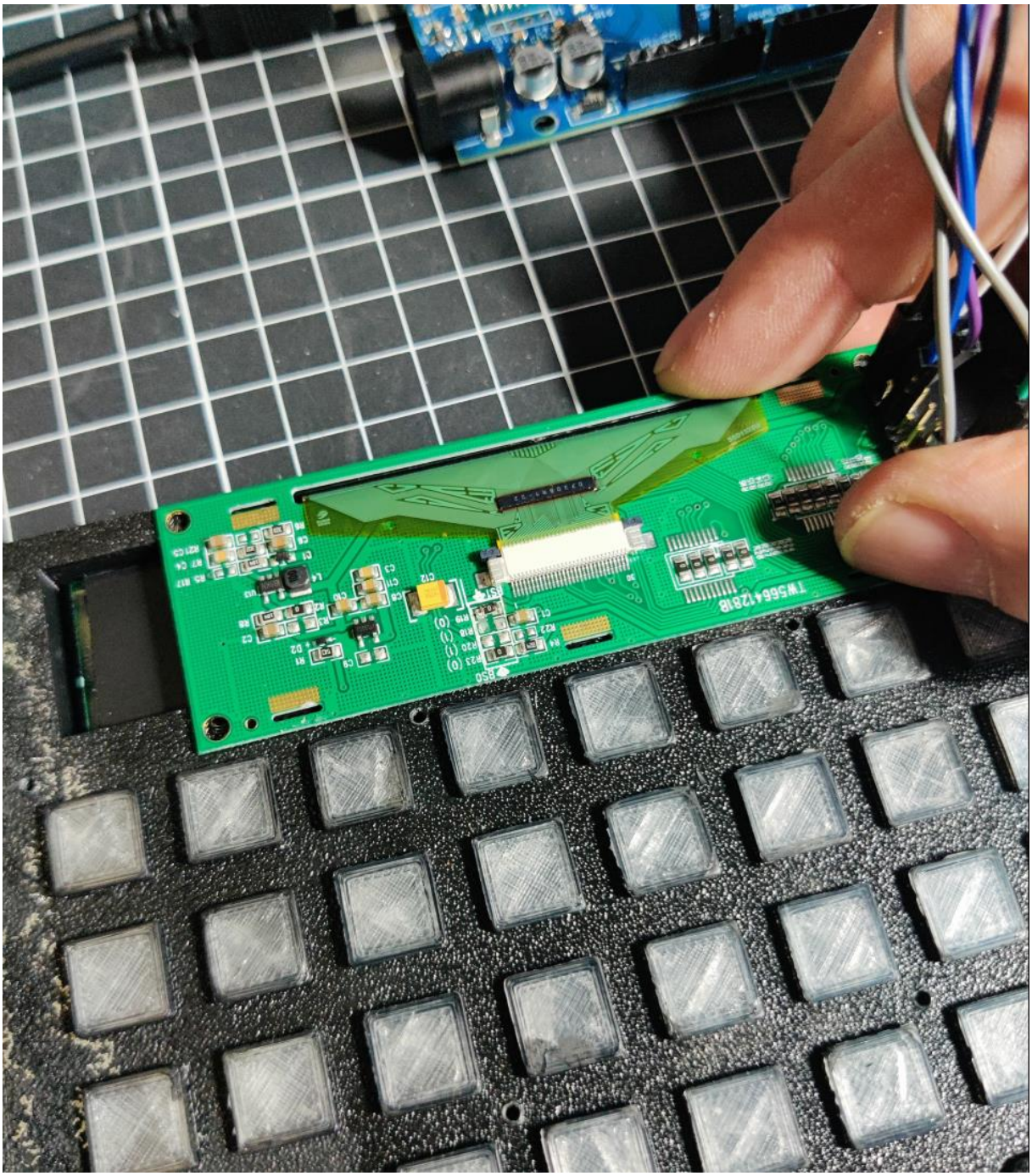


Remove the frame from the **OLED Display**.



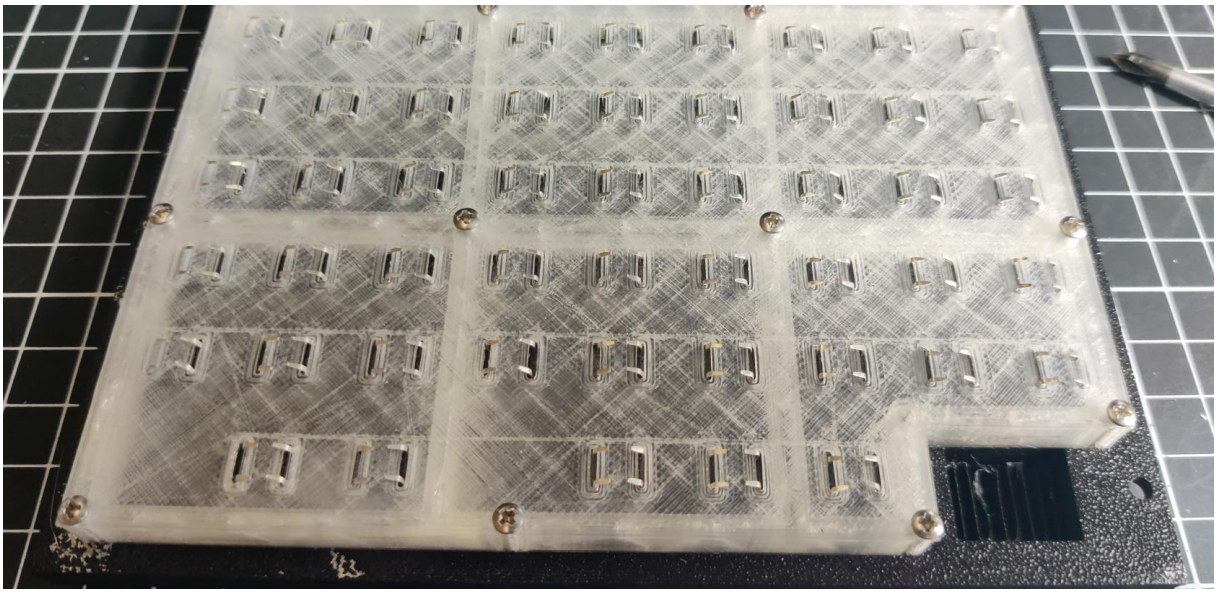
Instead of using the **KDU-OLED.MASK.STL**, I used a piece of 3mm thick **green transparent acrylic glass** and covered part of it with **black insulating tape** to prevent light from shining through later.

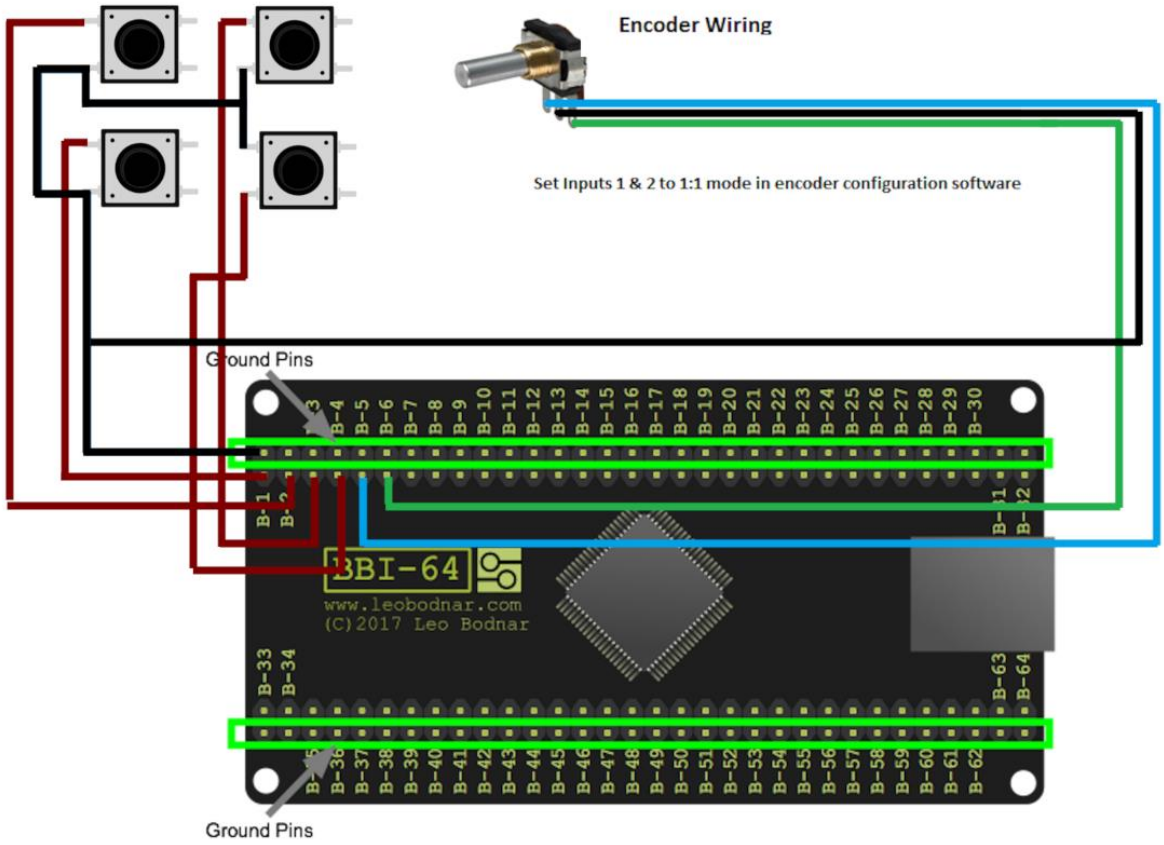
Use either the acrylic glass or the **KDU-OLED.MASK.STL**.



The OLED is simply inserted and secured by the button matrix. To ensure everything fits together later, the OLED pins need to be bent significantly. Even better, solder the wires instead of using Dupont connectors.

Silver screws for the Button Matrix, Black screws for the frontplate.





For the Leobodnar boards, you don't need much prior knowledge. Simply solder the wires to the switches and connect them to the board.

💡 **Tip:** It's sufficient for a single ground wire to run from the Leobodnar board, with the switches connected in series.

Rotary encoders are essentially digital rotary switches with **one ground pin and two input pins**. I've created a small diagram to illustrate this.

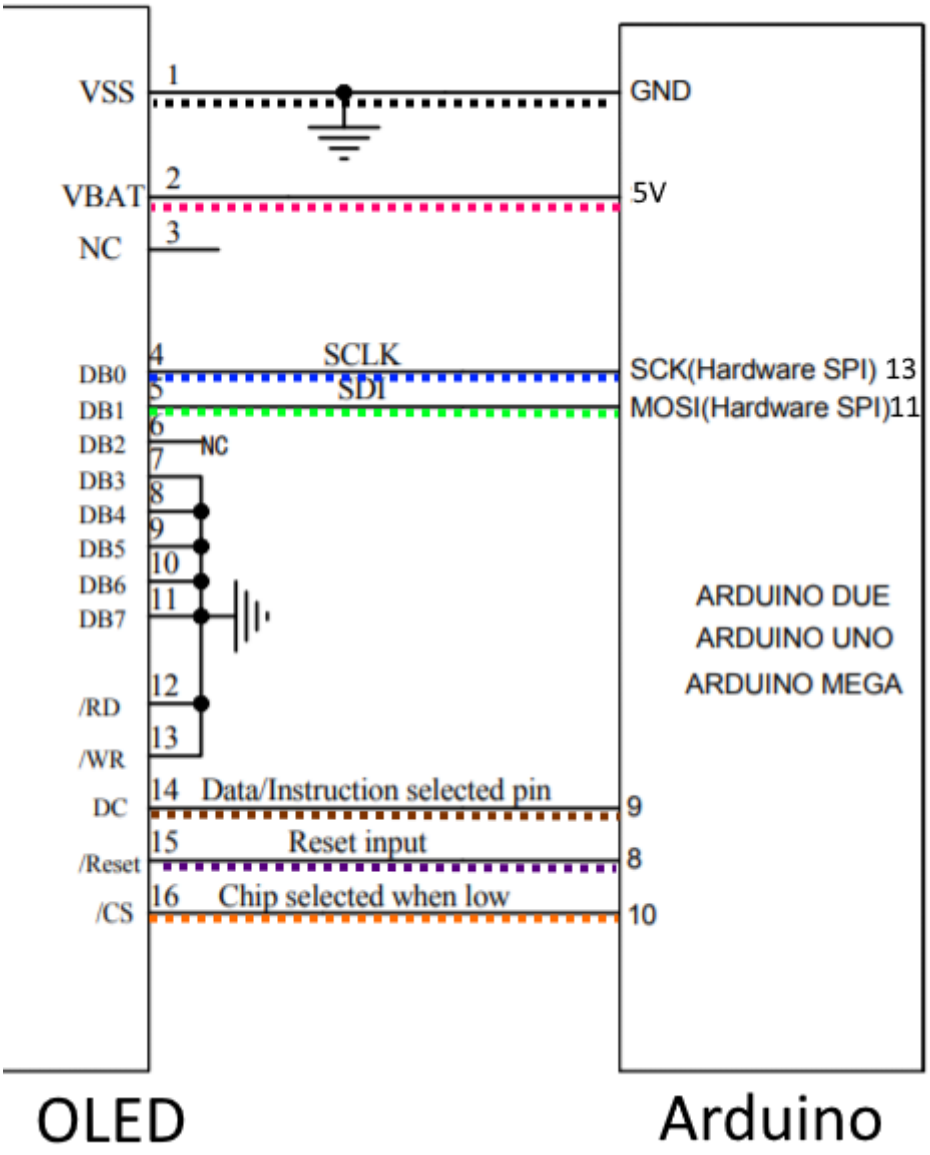
When using a **rotary encoder**, you need to define one or more **input pin pairs** in the **BBI-64 Config Tool** (available on the Leobodnar website) where the encoder is connected.

If you ever use **more than one BBI-64**, you must flash a separate firmware onto each board using the **HidFlasher tool** (also available on the website). Otherwise, Windows won't be able to distinguish between them. Leobodnar provides **10 different firmware versions** for this purpose.

🔧 **Tip:** You can rename the devices in the **Windows registry**. Instead of "BB64-3," for example, it could appear as **"AH-64D KDU."**

If you, like me, have **more than 10 BBI-64 boards**, it's not an issue. Since not all devices are always connected at the same time, **DCS has no problems handling input configurations**, even when plugging and unplugging the boards.

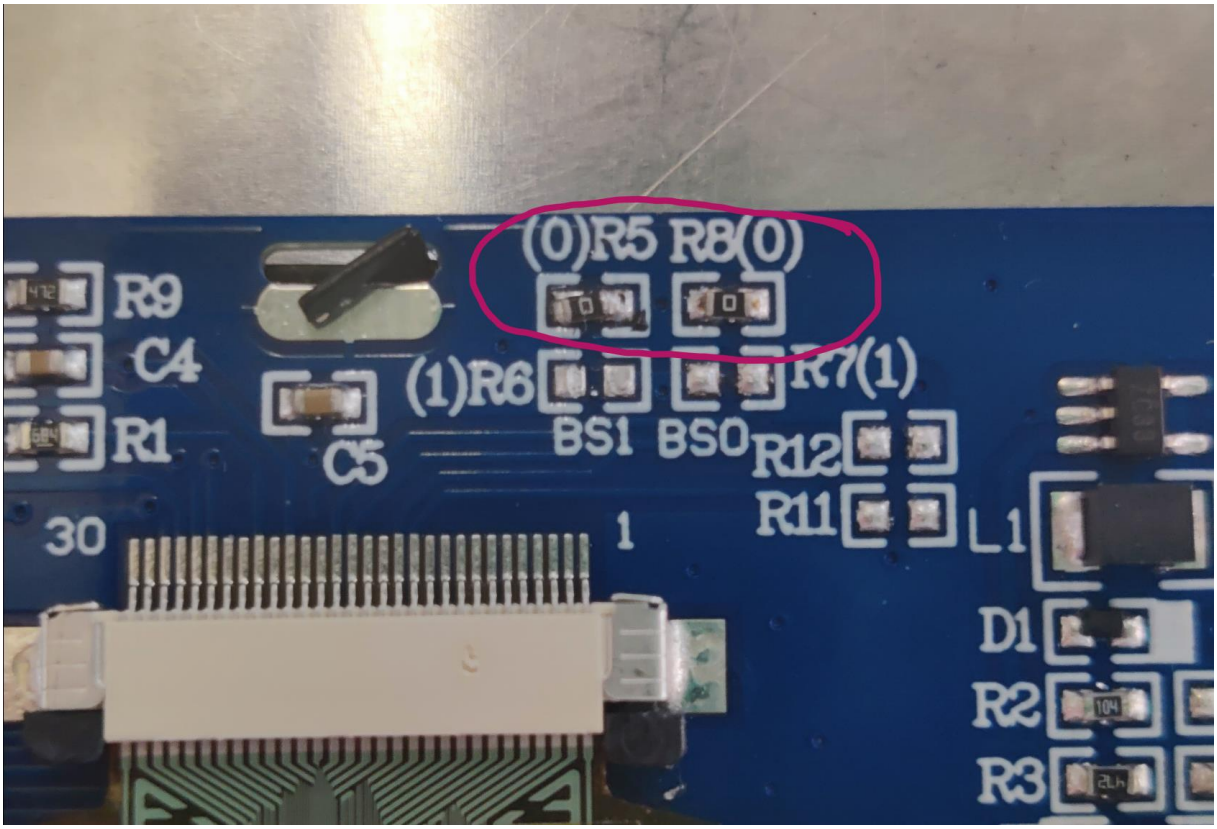
Simply connect the **OLED display** pin by pin to the **Arduino**, as shown in the diagram. The **seven required connections** are highlighted with **colored dashed lines**



You can switch the display source from Pilot to CP/Gunner by pressing a button. This button is wired to <u>PIN 7</u>. I integrated this function into the push-button of the rotary encoder for the KU Scratchpad Brightness Knob.

WARNING! Some SSD1322 OLEDs arrive in I²C mode instead of SPI. To enable SPI mode, the SMD resistors must be configured as shown in the image below:

- **R5 (0) → closed**
- **R6 (1) → open**
- **R7 (1) → open**
- **R8 (0) → closed**



For the lighting (which I haven't installed yet), I'll use green LED filament. I'll connect it directly to the 3V of the Arduino with a series resistor. I'll glue the LED filament onto the "button matrix" with hot glue. Not the prettiest solution, but it works fine.