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...

My projects have been downloaded thousands of times in the past. Providing the files and creating a guide is very time-consuming. Since no one is willing to make even a small donation, I have decided to offer the 3D files only for a small fee from now on. https://cults3d.com/de/modell-3d/gadget/ah-64d-kdu-keyboard-display-unit

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.

`This is a completely new sketch for an Arduino-only KDU. No BBI64 board is required!`

The code has been completely redesigned. It constantly checks whether you are in the pilot or co-pilot/gunner position, and the controls are automatically assigned accordingly. No more manual switching !

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.

You need to **flash my code** onto an **Arduino MEGA**. 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:

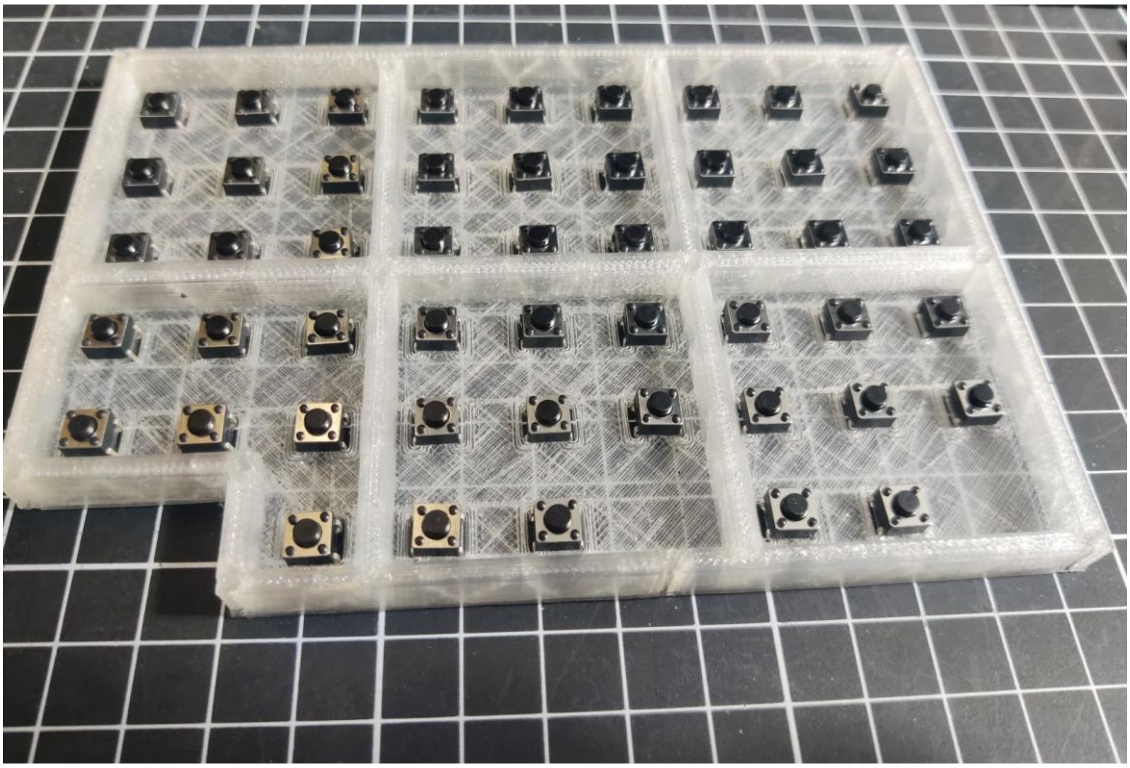
50x 6x6x4,3mm Tactile Switches: https://www.amazon.de/Youmile-100PACK-Schalter-Momentan-
Tactile/dp/B07XWXV1TS?source=ps-sl-shoppingads-
lpcontext&ref_=fplfs&smid=A3BN2T8LLIRB5S&th=1

1x Arduino MEGA 2560 R3 : https://www.amazon.de/Mega-2560-R3-Entwicklungsboard-Kompatibel-
Verbessert-
Stromversorgung/dp/B0CT8QC1FF?dib=eyJ2IjoiMSJ9.P62vr3QpLtUrpIu4TUr81qc44A8kadiltjswHyQPt
tQ72-
iIjSGfOJfFYUHD5PuFidfUraswgxqMx9MXz78wgLmiDwant_JbnZ_NY0oDgzMvK7E0oYrZU2uOg3ro92_N
WChiTh_zJs_pr4imTFr1d7__L1te9fTxnzzMvXxXB0SGgSXyXJv_Yy8J3jT-kYErX8UN9M0W-VX-
lGYDmsd6MggJdTmIvVJTZc8do-
13Y3U.Af5D_nItbnqtugG2HcTlGeUPVsNloIwUBOOvNPLbqow&dib_tag=se&keywords=arduino+mega
+2560&qid=1771854082&sr=8-1

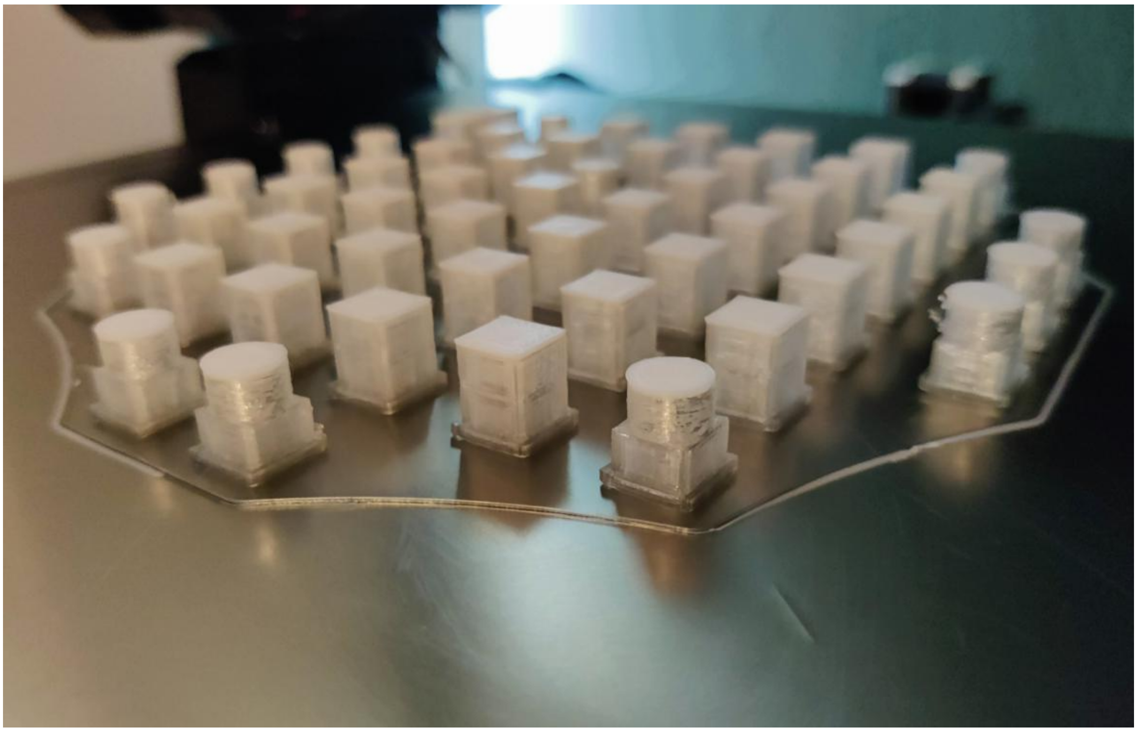
1x SSD1322 SPI OLED: https://www.amazon.de/OLED-Display-SSD1322-Grafik-LCD-Modul-Parallel-
L%C3%B6tstift-wei%C3%9F/dp/B0DQ52MRX1?source=ps-sl-shoppingads-
lpcontext&ref_=fplfs&psc=1&smid=A9KVXEOZANNPE

1x Potentiometer RV097NS 6mm
https://www.amazon.de/s?k=RV097NS+Potentiometer&__mk_de_DE=%C3%85M%C3%85%C5%BD%
C3%95%C3%91&crid=3YMY91GZ9X4D&sprefix=rv097ns+potentiometer%2Caps%2C117&ref=nb_sb_
noss

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.us4Y5oACoc7Vk
8tlmFZ68xbk84dToLNiT2758C_uNMxvpO-_vwA9Y6i9jG-7PCsnNSwVx0Sxlv7WQq10fZOYFhu3w879jCbqXo-Y-
ORUYPMZbgDdlUrU_7oPkDuGjv4DTNGXNnujj17r4uz9USDAbx6YfNPd4nAbZJZZYLFFGcn3uW8ep5ZecBCmtzVdW50ent--qQ-
CyOkZNOvslpSpAtgoZY3nQMClW0Mzf0Zj4g5rqQw27ICXqFxBrfmv5sKjrEoVLaFTxNMOJOSqMytdpWu5uy43fifL_OVNZTsSb4V
HM9HJDjPLyRUn3fuej3MXhCezRBJt5IEfB8FyzoDpritIx1rgfosUh3JGaqRmsI-
P9MpDt8EH45LEnDPwRF5B0AaX2_3dn745xScEQbqDoa2JaSXo75Mm-
cI5gKzkLawPMQL7N3aBiqfc3yb2Bxg3.3z_NiMQb6Ro_yU0_UNk8SAhppodXVby1FKetAXP1tPI&dib_tag=se&keywords=3V%2
BLED-
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.



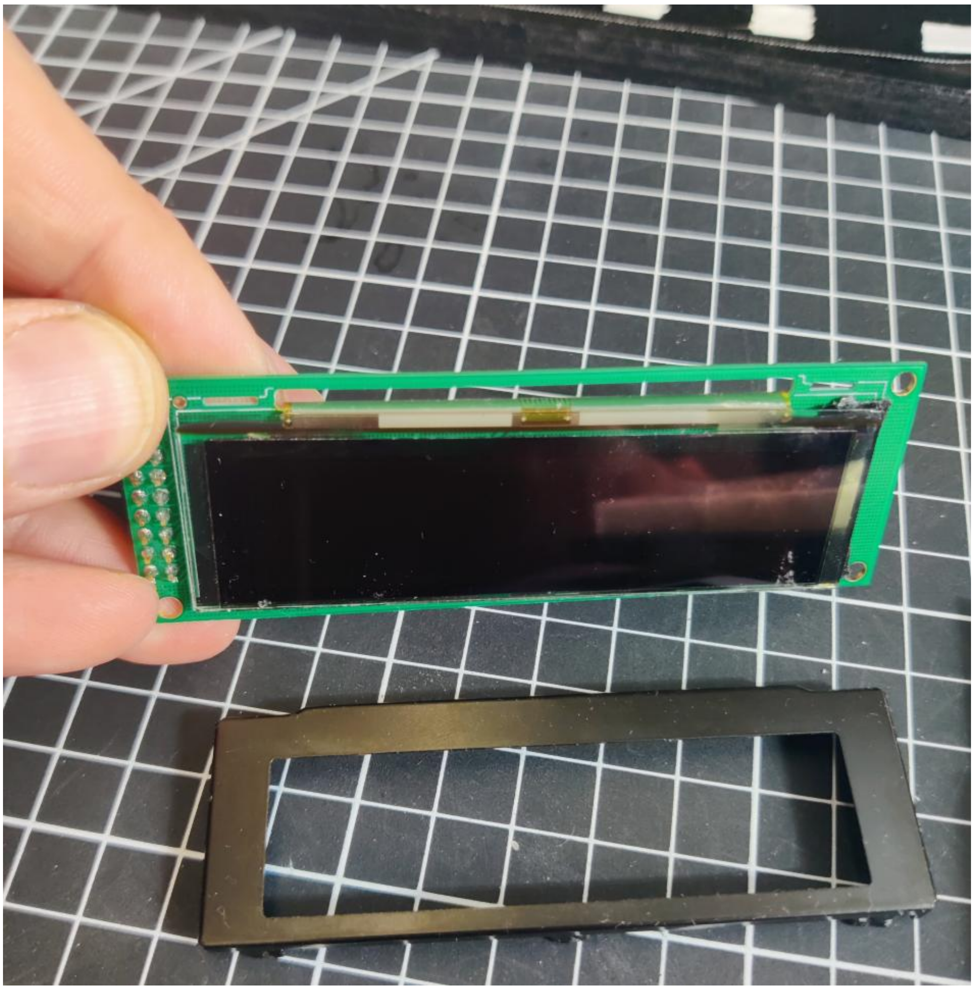
The buttons were also printed with transparent PLA. The top three layers were printed with white PLA, making them ideal for backlighting. The buttons are then **spray-painted with black acrylic lacquer** from a can and can be **engraved with a diode laser**.

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**.

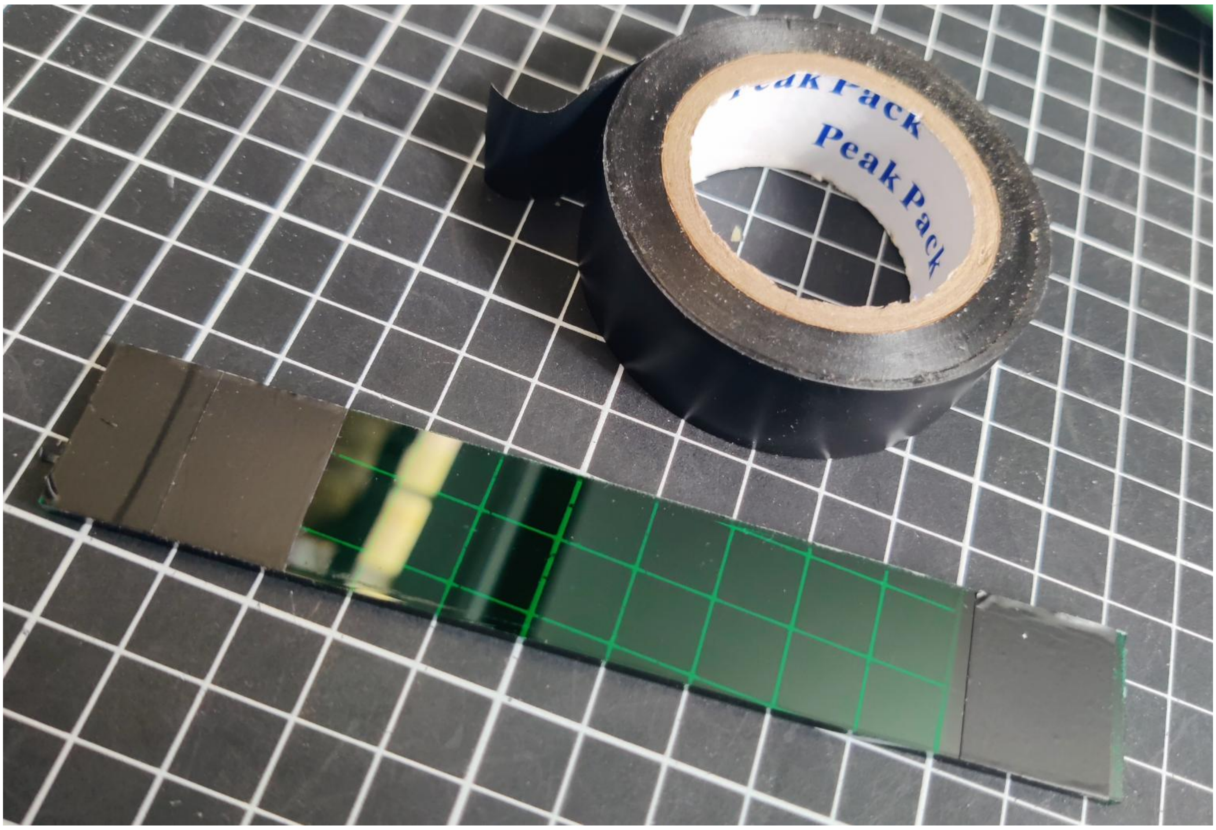
If you **don’t have a dual-extruder 3D printer**, you can set up a **Post-Process script** in **Cura** ("Pause at Layer Height") and manually switch the filament.

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**.

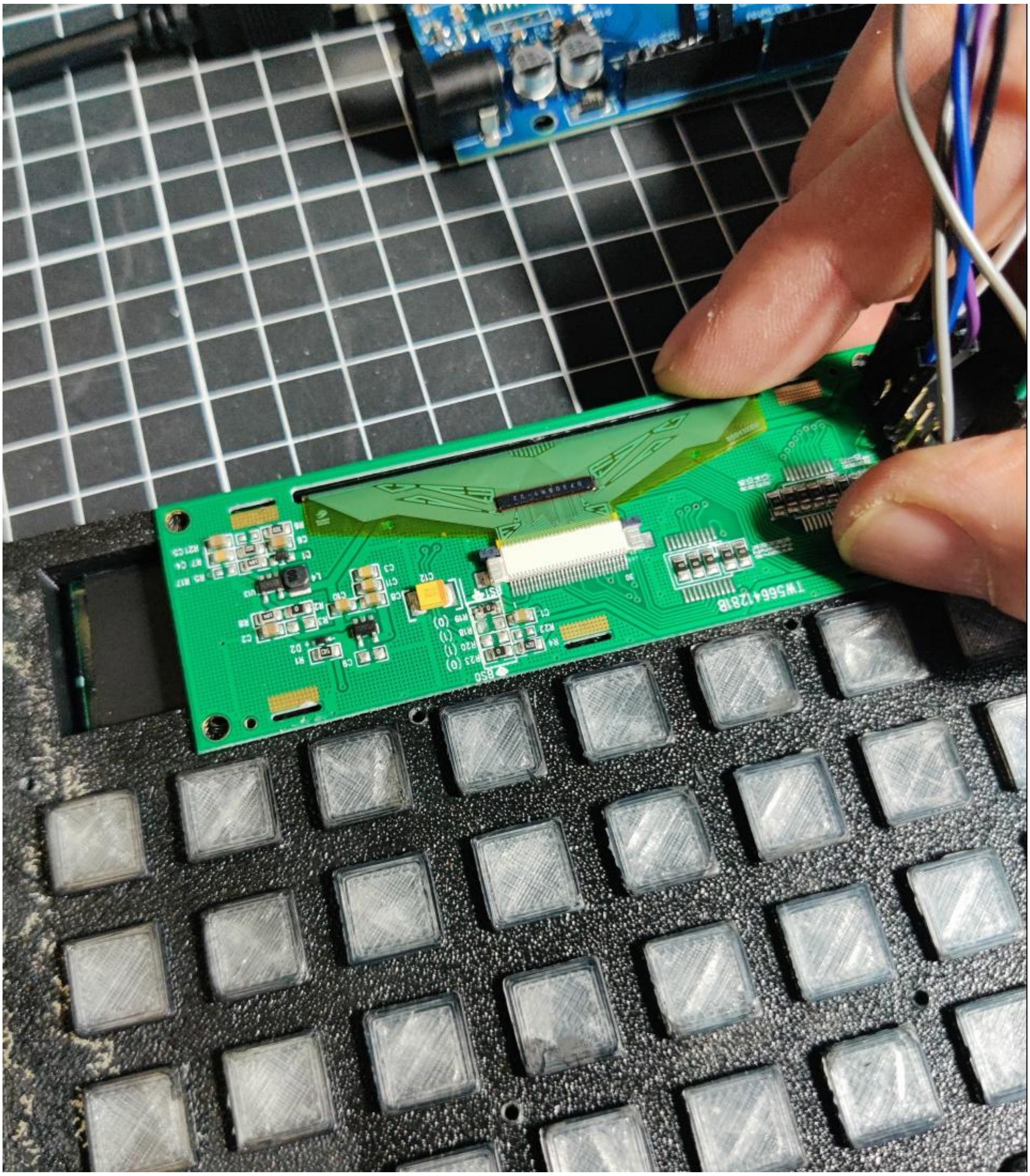


Remove the frame from the **OLED Display**.



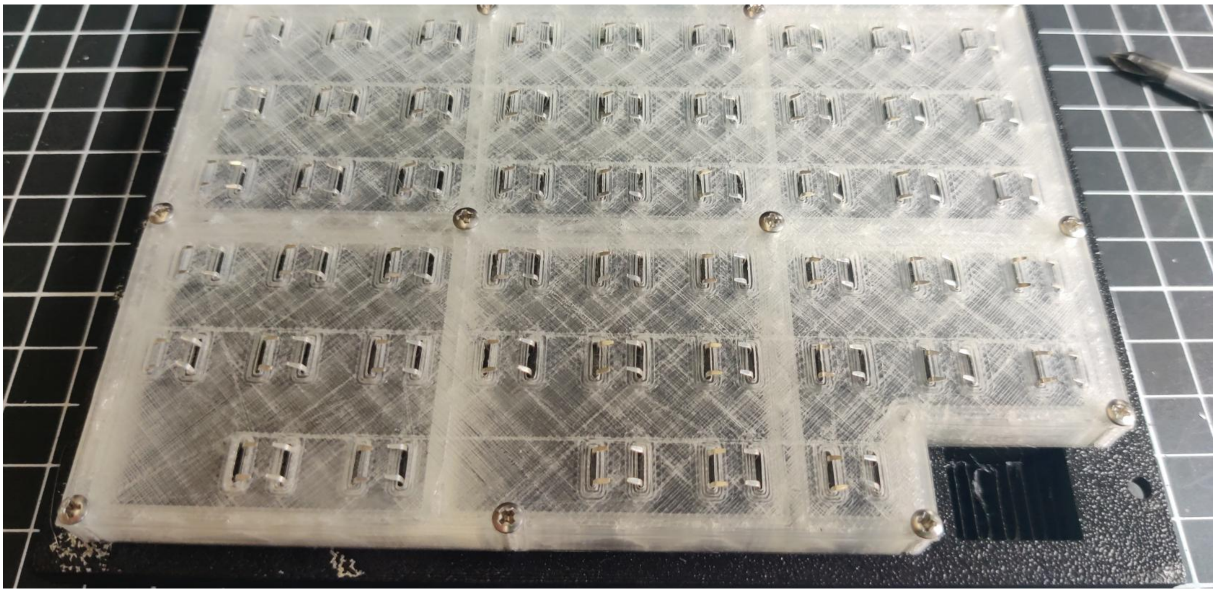
Instead of using the **KDU-OLED.MASK.STL**, I used a piece of 3mm thick **green transparent acrylic glass** and covered part of it with **black insulating tape** to prevent light from shining through later.

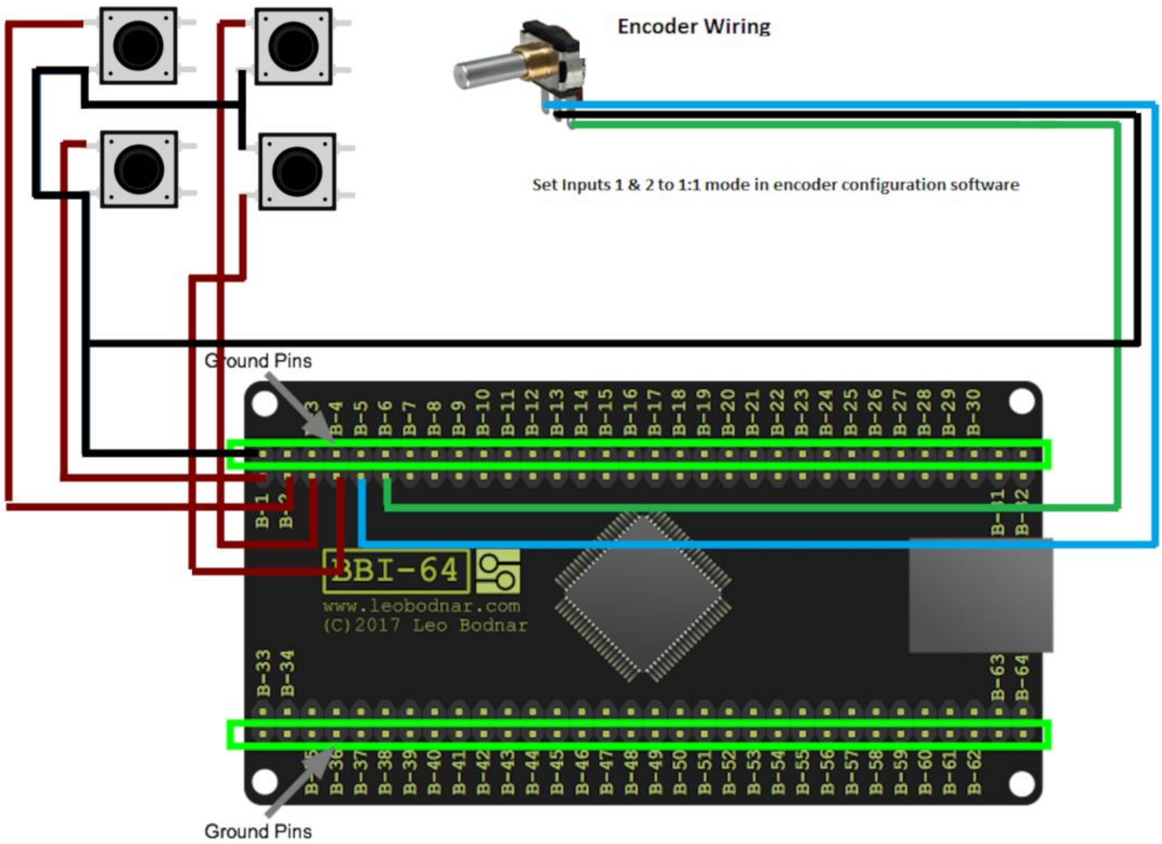
Use either the acrylic glass or the **KDU-OLED.MASK.STL**.



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.

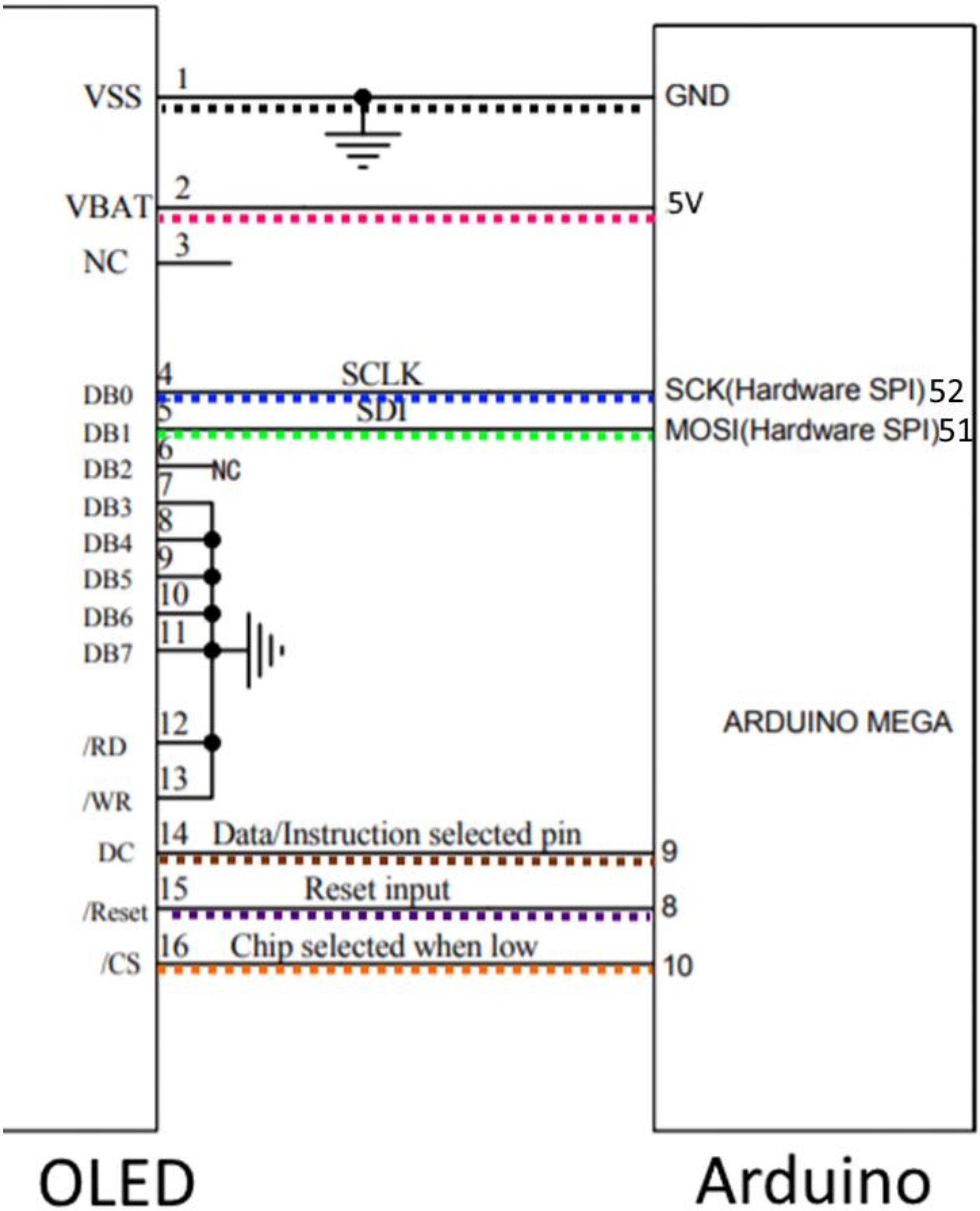
When using a **rotary encoder**, you need to define one or more **input pin pairs** in the **BBI-64 Config Tool** (available on the Leobodnar website) where the encoder is connected.

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.

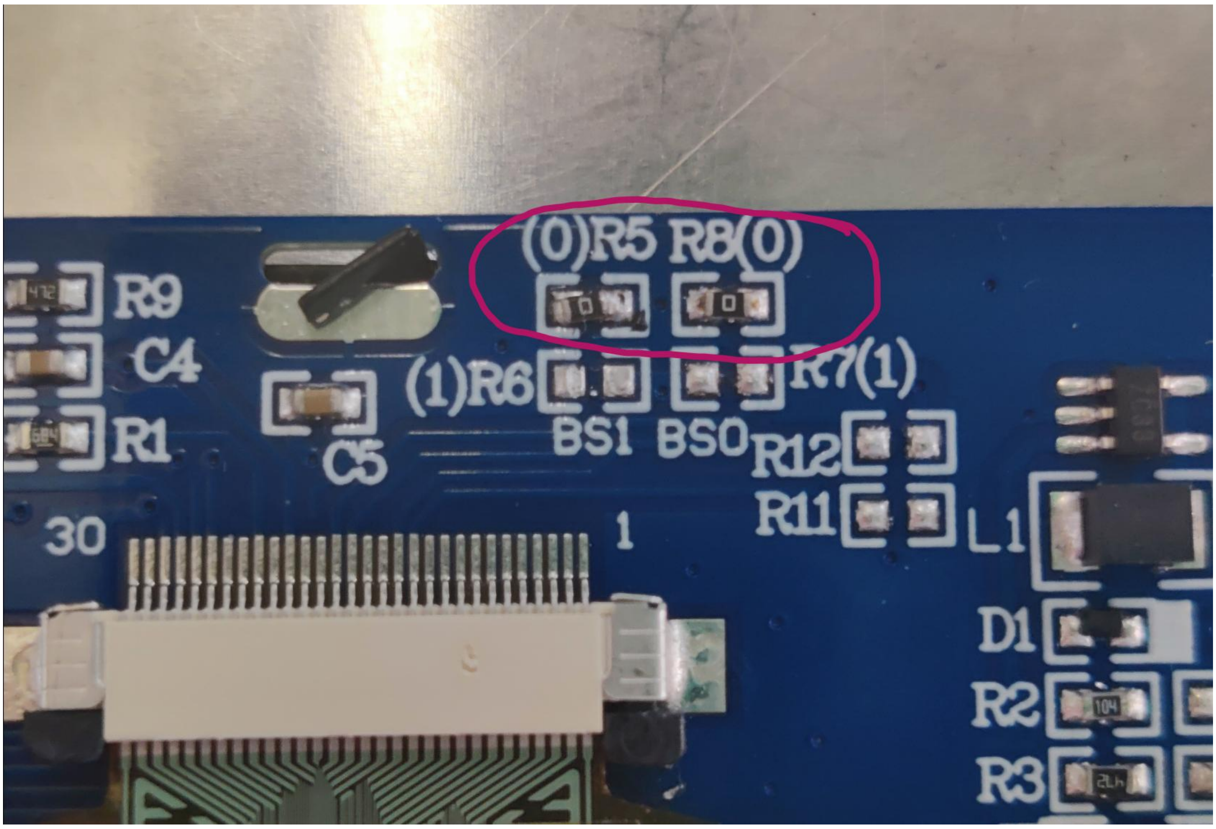
Simply connect the **OLED display** pin by pin to the **Arduino**, as shown in the diagram. The **seven required connections** are highlighted with **colored dashed lines**



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.

WIRING INFO

Display Pin	Arduino Mega Pin
VCC	5V *(oder 3.3V je nach Modul!)*
GND	GND
SCK / CLK	52
MOSI / DIN	51
CS	10
DC	9
RST	8

Poti	Pin	PLT	CPG
pot_01	A0	PLT_KU_BRT	CPG_KU_BRT

Switch	Pin	PLT Controlname	CPG Controlname
switch2_01	D24	PLT_KU_0	CPG_KU_0
switch2_02	D34	PLT_KU_1	CPG_KU_1
switch2_03	D35	PLT_KU_2	CPG_KU_2
switch2_04	D36	PLT_KU_3	CPG_KU_3
switch2_05	D43	PLT_KU_4	CPG_KU_4
switch2_06	D44	PLT_KU_5	CPG_KU_5
switch2_07	D45	PLT_KU_6	CPG_KU_6
switch2_08	A2	PLT_KU_7	CPG_KU_7
switch2_09	A5	PLT_KU_8	CPG_KU_8
switch2_10	D31	PLT_KU_9	CPG_KU_9
switch2_11	D14	PLT_KU_A	CPG_KU_A
switch2_12	A1	PLT_KU_B	CPG_KU_B
switch2_13	D12	PLT_KU_C	CPG_KU_C
switch2_14	D11	PLT_KU_D	CPG_KU_D
switch2_15	D32	PLT_KU_E	CPG_KU_E
switch2_16	D33	PLT_KU_F	CPG_KU_F
switch2_17	D37	PLT_KU_G	CPG_KU_G
switch2_18	D38	PLT_KU_H	CPG_KU_H
switch2_19	D39	PLT_KU_I	CPG_KU_I
switch2_20	D40	PLT_KU_J	CPG_KU_J
switch2_21	D41	PLT_KU_K	CPG_KU_K
switch2_22	D42	PLT_KU_L	CPG_KU_L
switch2_23	D46	PLT_KU_M	CPG_KU_M

Switch	Pin	PLT Controlname	CPG Controlname
switch2_24	D47	PLT_KU_N	CPG_KU_N
switch2_25	D48	PLT_KU_O	CPG_KU_O
switch2_26	D49	PLT_KU_P	CPG_KU_P
switch2_27	A3	PLT_KU_Q	CPG_KU_Q
switch2_28	A4	PLT_KU_R	CPG_KU_R
switch2_29	D30	PLT_KU_S	CPG_KU_S
switch2_30	D29	PLT_KU_T	CPG_KU_T
switch2_31	D28	PLT_KU_U	CPG_KU_U
switch2_32	D27	PLT_KU_V	CPG_KU_V
switch2_33	D26	PLT_KU_W	CPG_KU_W
switch2_34	D25	PLT_KU_X	CPG_KU_X
switch2_35	A7	PLT_KU_Y	CPG_KU_Y
switch2_36	A8	PLT_KU_Z	CPG_KU_Z
switch2_37	D18	PLT_KU_LEFT	CPG_KU_LEFT
switch2_38	D17	PLT_KU_RIGHT	CPG_KU_RIGHT
switch2_39	D7	PLT_KU_BKS	CPG_KU_BKS
switch2_40	D19	PLT_KU_CLR	CPG_KU_CLR
switch2_41	D4	PLT_KU_DIV	CPG_KU_DIV
switch2_42	D23	PLT_KU_DOT	CPG_KU_DOT
switch2_43	D16	PLT_KU_ENT	CPG_KU_ENT
switch2_44	D2	PLT_KU_MINUS	CPG_KU_MINUS
switch2_45	D5	PLT_KU_MULTI	CPG_KU_MULTI
switch2_46	D3	PLT_KU_PLUS	CPG_KU_PLUS
switch2_47	D22	PLT_KU_SIGN	CPG_KU_SIGN
switch2_48	A9	PLT_KU_SLASH	CPG_KU_SLASH
switch2_49	D6	PLT_KU_SPC	CPG_KU_SPC