

# MilKris' Multi-Purpose-Display (MPD) for the DCS AH-64D 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 might be 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-mpd-multi-purpose-display>



This is my MPD for the DCS AH-64D Apache. The project includes 3D-printable files, optional parts with raised labels or laser engraving support, and of course my code for the Arduino. The total cost to build one MPD is approximately €70 or about \$80. If Arduino feels like a complete mystery to you, I also describe an alternative build using a Leobodnar BBI-64 USB game controller board. This option is basically plug & play, but it does increase the cost.

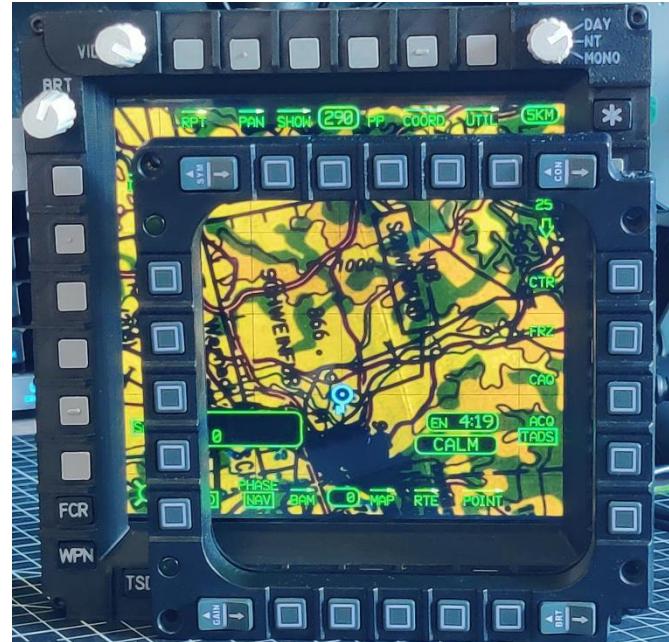
## A few things to know before you start building:

The MPDs are not true to the original scale! I intentionally resized them to perfectly fit an 8-inch display. Still, with dimensions of 174x174mm, they are significantly larger than, for example, the TM Cougar MFDs.

For VR users, an actual LCD screen doesn't make much sense — that's why I've included a "ScreenDummy.stl" for 3D printing. Additionally, I've designed a few special buttons with raised tactile markers to make identification easier by touch.

I chose the Arduino Mega 2560 for this build because it's affordable and offers a shitload of digital input/output pins.

**Both MPDs are built identically.** The enclosure features four cable openings, allowing you to route the wires according to your setup. This gives you flexibility to adapt the installation to your needs.



Since I'm using [DCS-BIOS \(Skunkworks\)](#), these MPDs are designed to work exclusively with the DCS AH-64D Apache! There is a separate Arduino sketch (code) for a left and right MPD.

Alternatively, I recommend using a [Leobodnar BBI-64](#). It acts as a simple USB game controller and requires **no additional configuration**. Plus, you won't need to follow a specific pin layout during wiring, which makes installation much easier. If you're going the BBI-64 route, check the instructions on the last page.

## **Important notes if you're using the Arduino Mega 2560 and DCS-BIOS:**

When the Arduino starts, each pin is assigned a fixed address by DCS-BIOS and cannot be changed while running. Since the Pilot (PLT) and Co-Pilot/Gunner (CP/G) require different addresses, you must select the desired mode *before* powering the Arduino via USB.

This is done using a switch that selects between PLT and CP/G mode (Pin 13 LOW = PLT, Pin 13 HIGH = CP/G).

I personally fly and operate the Apache exclusively from the front seat, so this limitation isn't an issue for me. However, if you plan to switch between PLT and CP/G during flight, here are your options:

- a) Unplug and replug the MPD's USB connection and restart the DCS-BIOS connection (sokat)
- b) Set up one MPD in PLT mode and the other in CP/G mode
- c) Skip DCS-BIOS entirely and use a Leobodnar BBI-64 or an Arduino Leonardo configured as a USB controller

If you want to build this project using the Arduino Mega 2560 you need to **flash my code** onto an **Arduino**. Here are my sketches:

**LEFT MPD:** [https://github.com/MilKris666/AH-64D-Apache-MPD-Multi-Purpose-Display/blob/main/AH64\\_MPD-LEFT.ino](https://github.com/MilKris666/AH-64D-Apache-MPD-Multi-Purpose-Display/blob/main/AH64_MPD-LEFT.ino)

**RIGHT MPD:** [https://github.com/MilKris666/AH-64D-Apache-MPD-Multi-Purpose-Display/blob/main/AH64\\_MPD-RIGHT.ino](https://github.com/MilKris666/AH-64D-Apache-MPD-Multi-Purpose-Display/blob/main/AH64_MPD-RIGHT.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>

## **3D printing:**

Depending on whether you're using the Arduino Mega 2560 or the Leobodnar BBI-64, there are separate versions of the **ScreenHousing** included.

The MPD frame's buttons and labels are 3D-printed with white PLA, spray-painted with black acrylic lacquer, and then laser-engraved using 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.

## **Display Setup:**

The displays are, of course, powered via HDMI and operate independently of the Arduino or BBI-64. The AH-64D uses the DCS' standard viewport export method for rendering the left and right MPDs. On the final pages of this build guide, I'll show you the correct pixel-precise configuration for your MonitorSetup.lua.

**I'm looking forward to your [feedback](#) — and of course, feel free to share pictures of your own MPDs!**

**Yours,**  
**MilKris**

**Follow me**  
[instagram](#)  
[Reddit](#)  
[YouTube](#)  
[Thingiverse](#)

## Part list:

1x Wisecoco 8 Zoll LCD 1024 x 768 IPS Display:

<https://aliexpress.com/item/4000155436637.html>

1x 40 Pin FFC/FPC Cable Extention | 40 Pin, spacing 0.5 mm socket, 200mm long:

<https://de.aliexpress.com/item/4000898466075.html>

1x 3-Way-Switch | 2 Pole 3:

<https://aliexpress.com/item/1005004551869432.html>

31x Push Button | 6x6x4.3mm

<https://aliexpress.com/item/4001166999847.html>

2x 360 Rotary Encoder | EC11 20mm Plum Handle

<https://aliexpress.com/item/1005002840845750.html>

1x ON-OFF Switch | SS12D00G4

<https://de.aliexpress.com/item/32721440534.html>

Dupont connector shell:

<https://aliexpress.com/item/32665232173.html>

If you don't already have one: a Dupont crimping tool:

<https://de.aliexpress.com/item/1005007130436011.html>

Arduino MEGA2560 R3 ATmega2560-16AU

<https://aliexpress.com/item/1005006177033344.html>

Dupont connector | **male** 2,54mm : (**if you build the Arduino version!**)

<https://aliexpress.com/item/1005006626744976.html>

Loads of 0.14mm<sup>2</sup> copper wire...

---

### Alternativ to the Arduino MEGA2560:

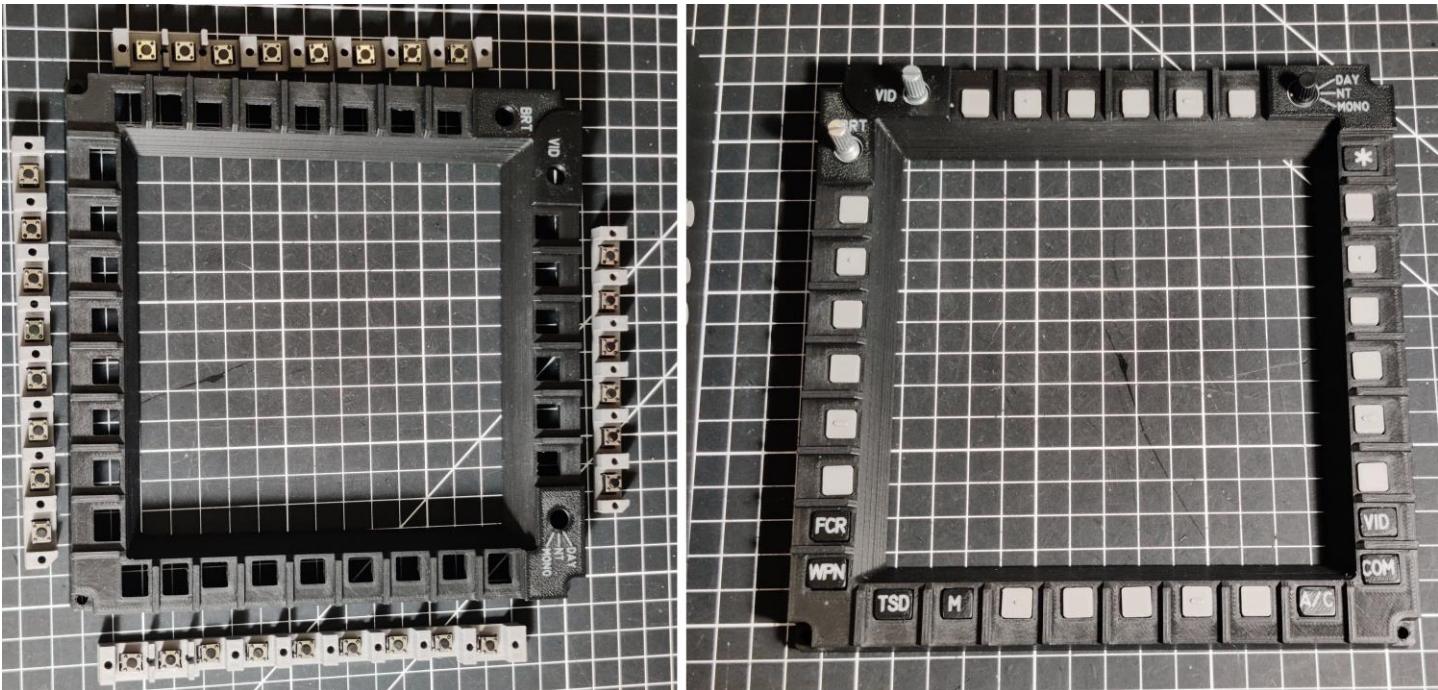
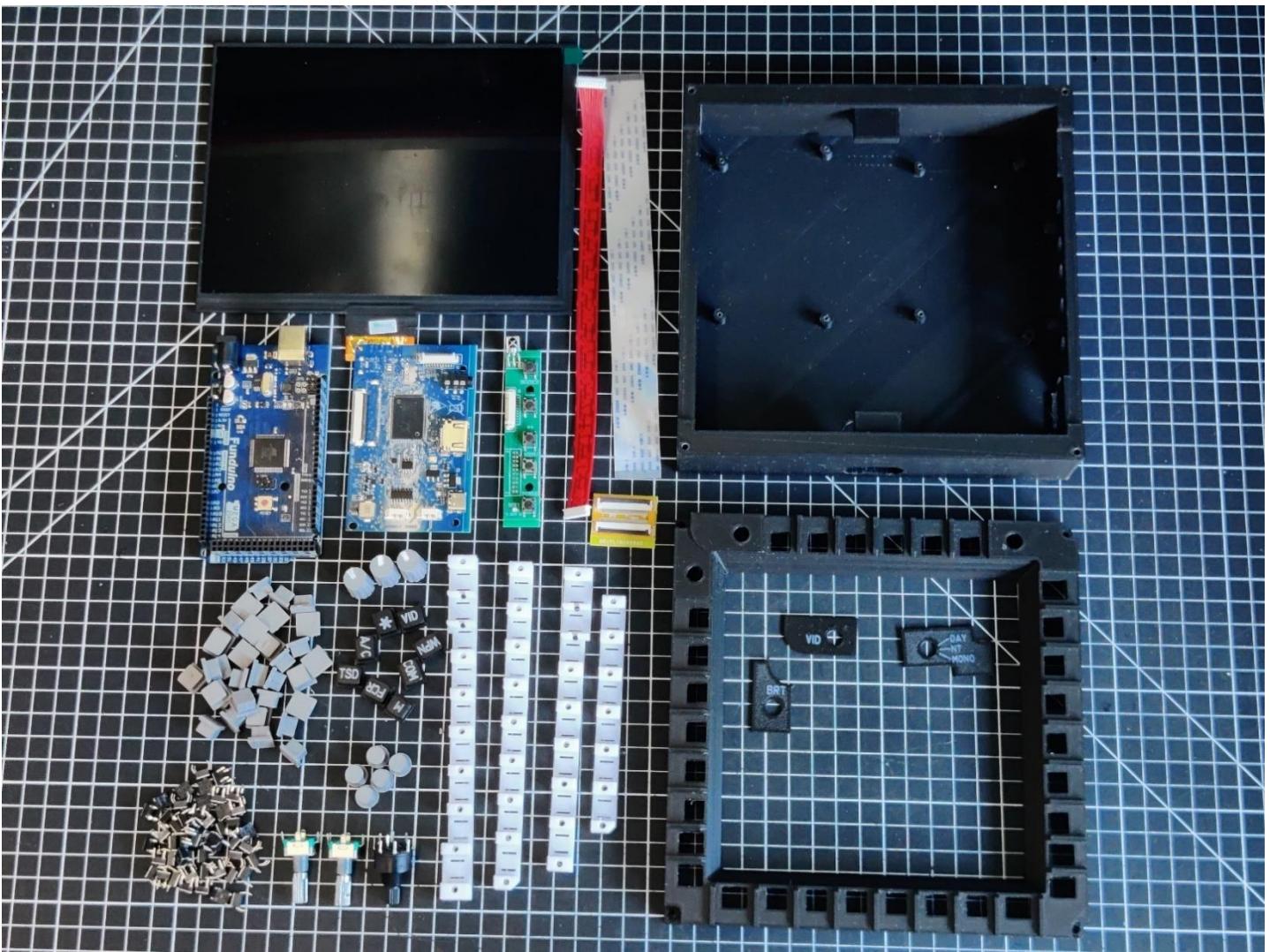
Leobodnar BBI-64 Gamecontroller Board

[https://www.leobodnar.com/shop/index.php?main\\_page=product\\_info&cPath=94&products\\_id=300&zenid=3ee50b521ef3f7be331c536754bb01a0](https://www.leobodnar.com/shop/index.php?main_page=product_info&cPath=94&products_id=300&zenid=3ee50b521ef3f7be331c536754bb01a0)

Dupont connector | **female** 2,54mm : (**if you build the BBI-64 version!**)

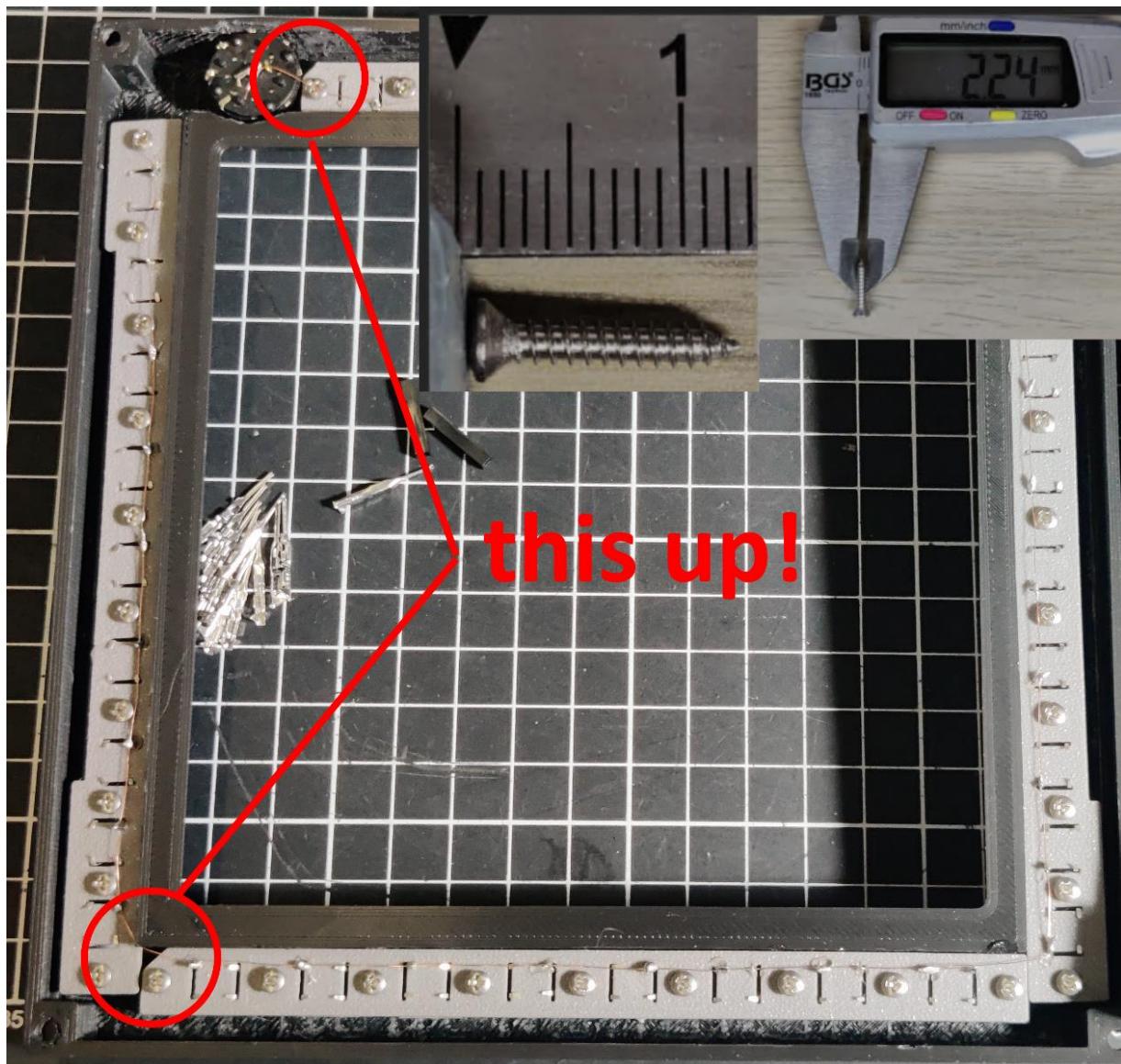
<https://aliexpress.com/item/1005006626744976.html>

# Now have fun building!



Insert the push buttons into the matrix boards and mount them into the MPD frame. I use regular screws with a diameter of 2.3 mm. The shaft length shouldn't exceed 10 mm. Simply glue the rotary encoder and the 3-way switch in place using superglue, and secure them with a bit of hot glue for extra stability.

Make sure the matrix boards are correctly oriented — see the reference image.

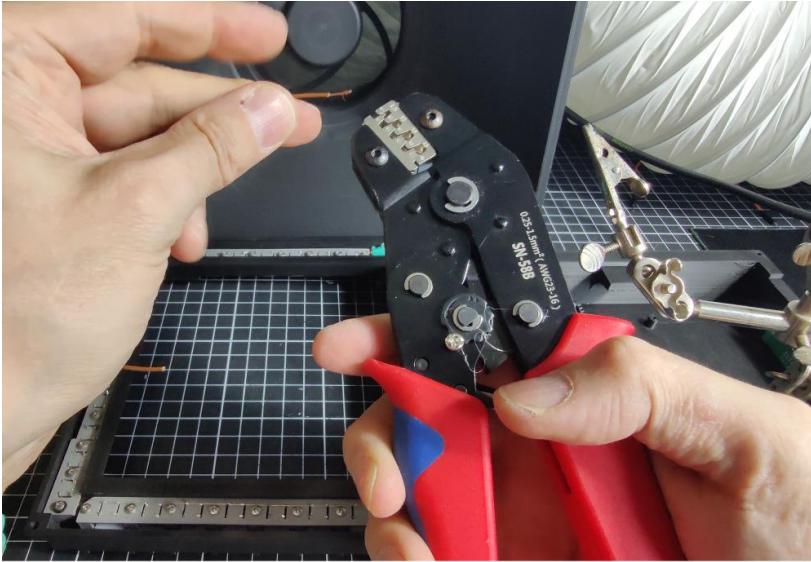


Screen buttons and screen controller Board.

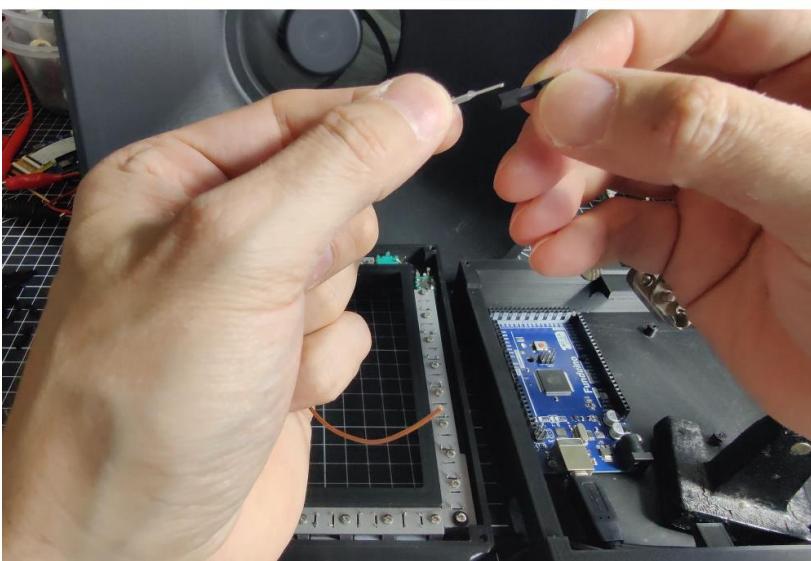




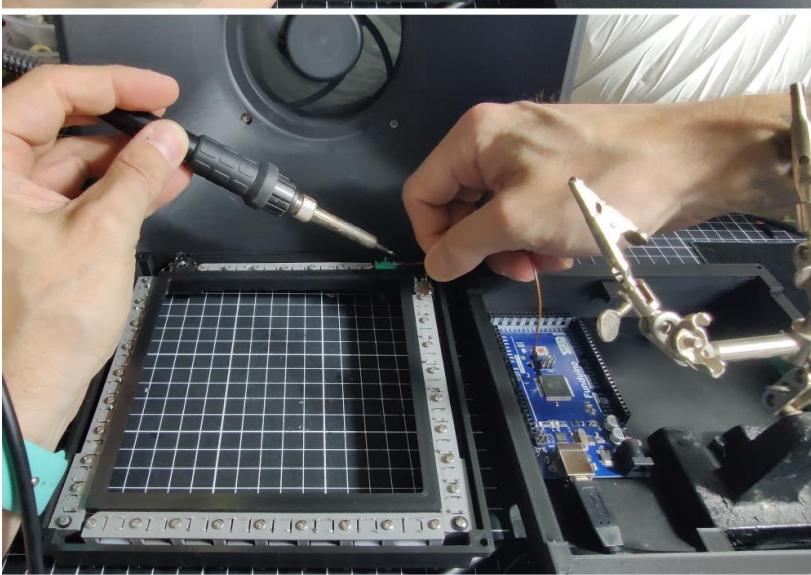
As shown in the image, solder a ground wire (you can use a stripped wire from a regular cable) in a loop to all inner pins of the push buttons, as well as to the GND pin of the 3-way switch and the two rotary encoders. Then connect the whole setup to a GND pin of the Arduino using a single wire.



💡 **Tip:** Use Dupont connectors for easy wiring!

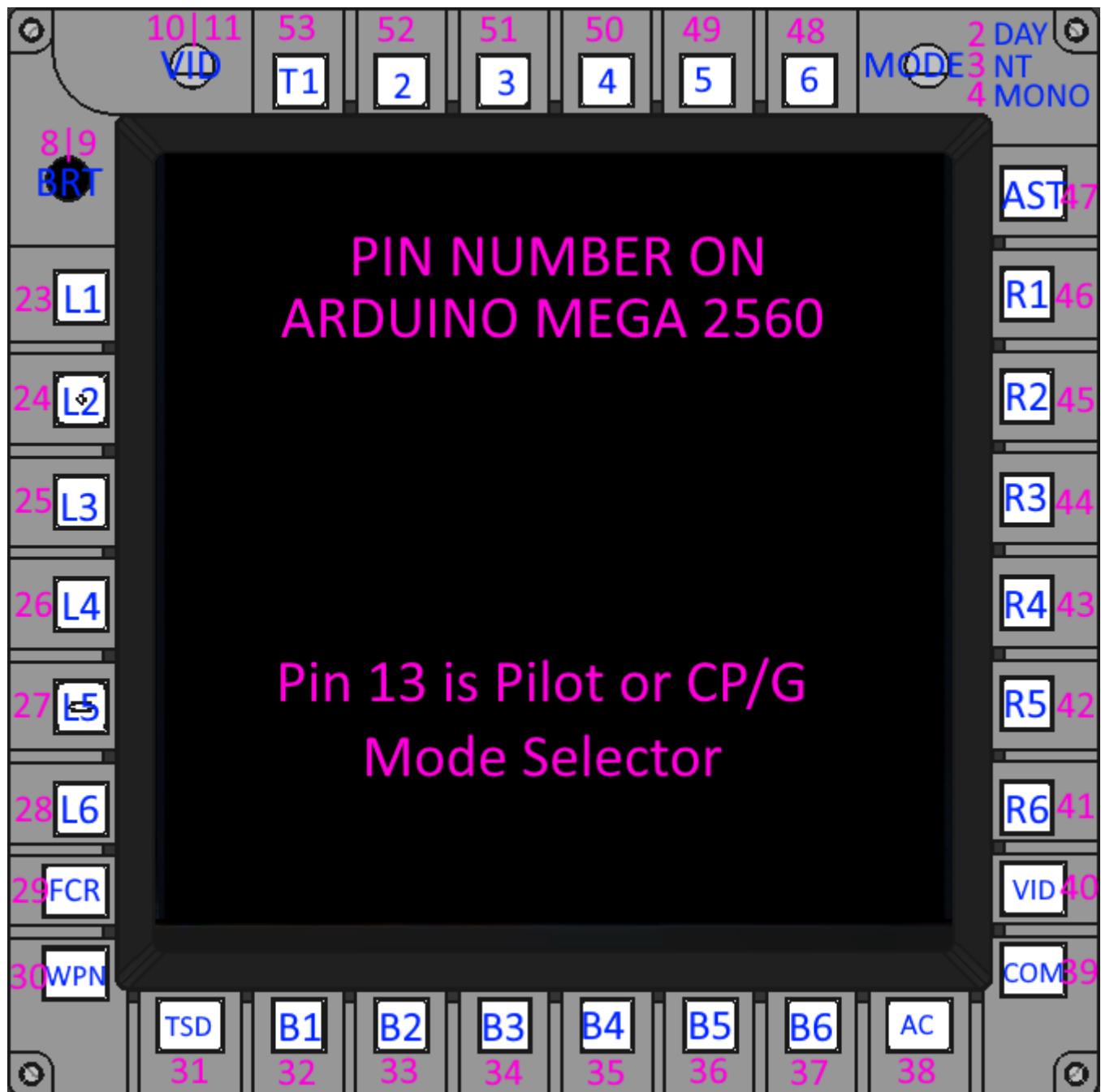


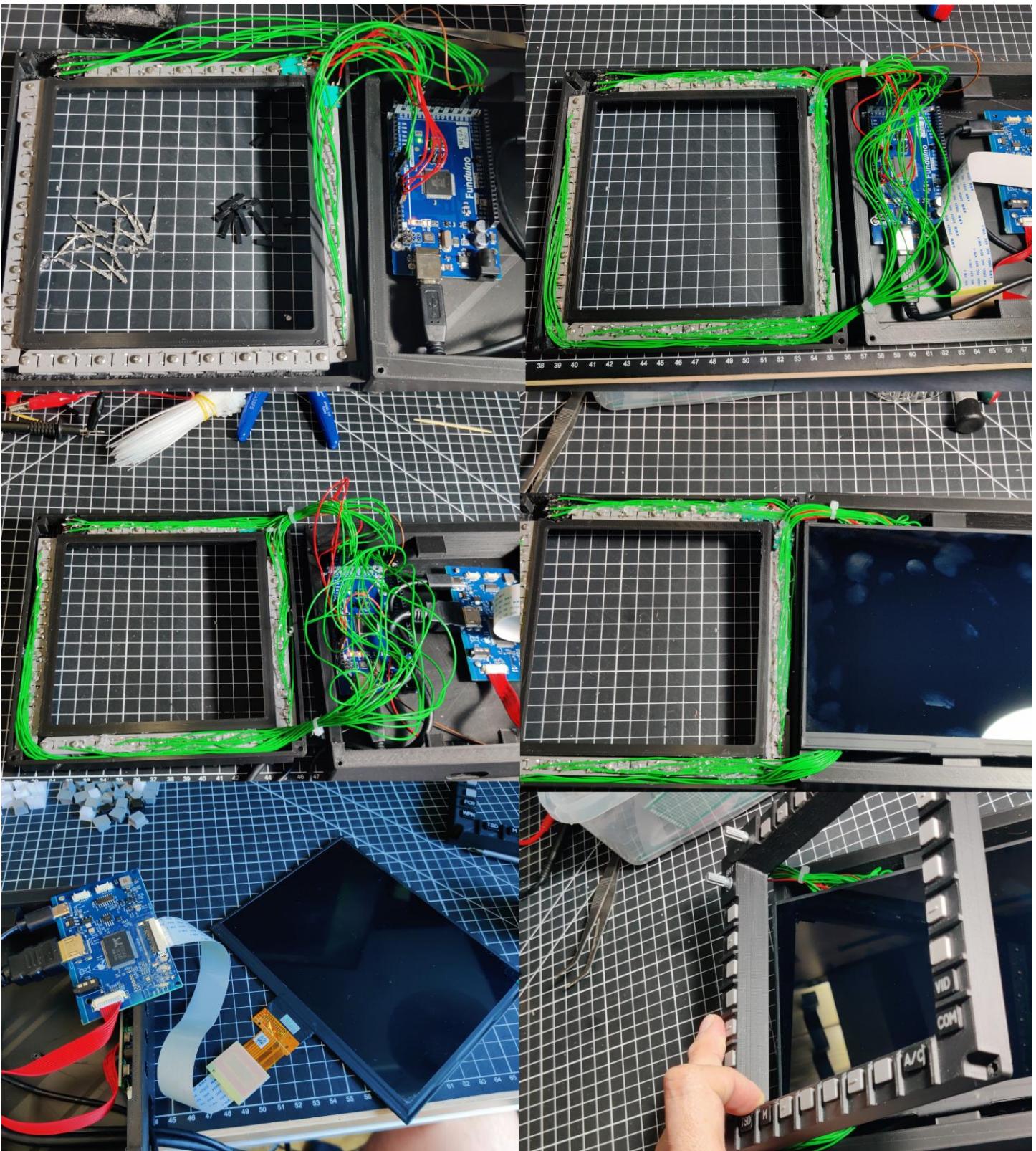
⚠️ **Important:** Keep the wires as short as possible — it will make final assembly much easier. To help with this, position the MPD frame to the left of the enclosure (screen housing) during soldering.



Now simply follow the image sequence below — it provides a clear illustration of the assembly process.

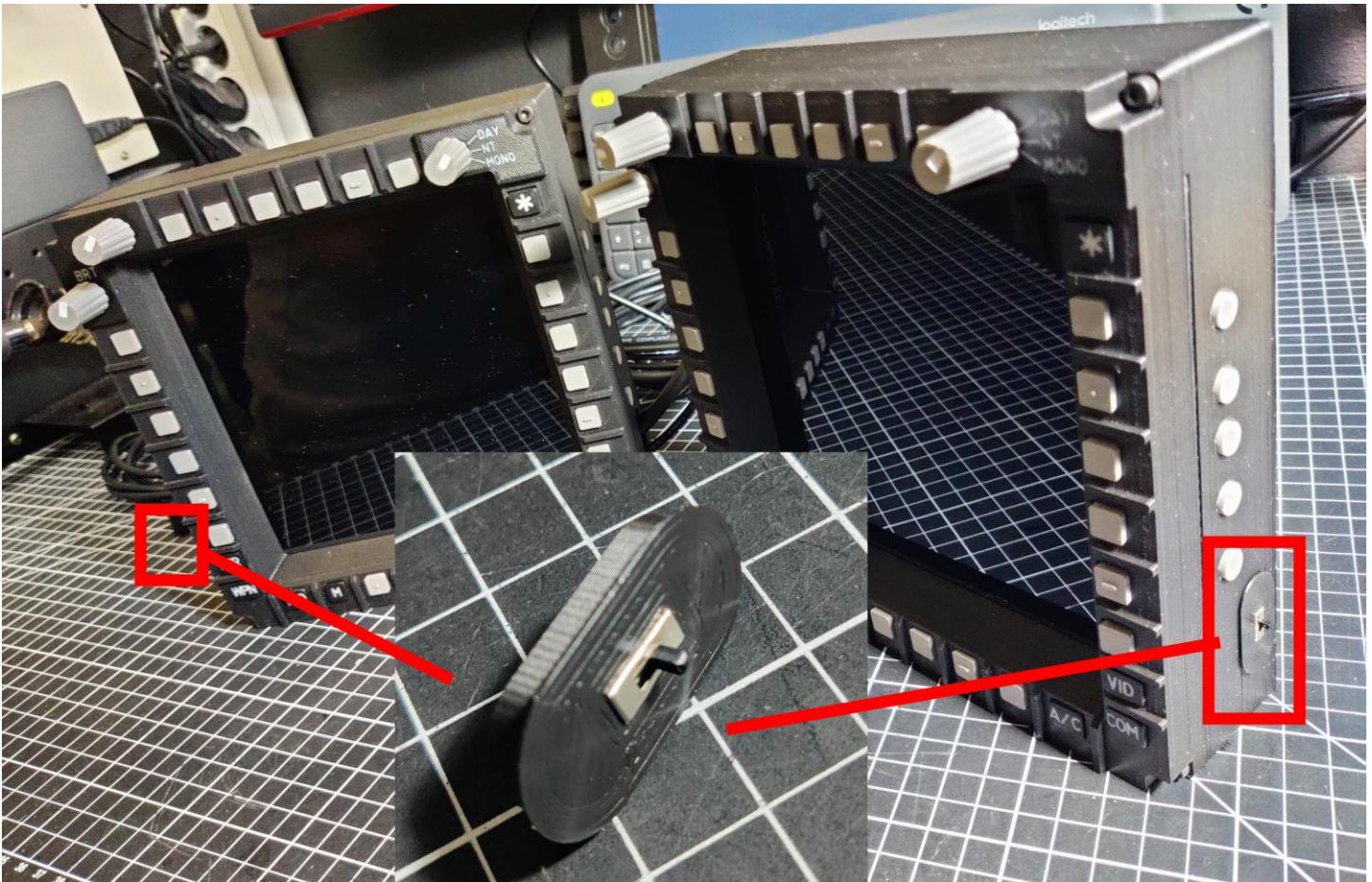
⚠️ **Note:** Pay attention to the pin number layout!





At the end, make sure to secure the wires with a bit of hot glue.  
The LCD screen is only placed into position — it is not permanently fixed.

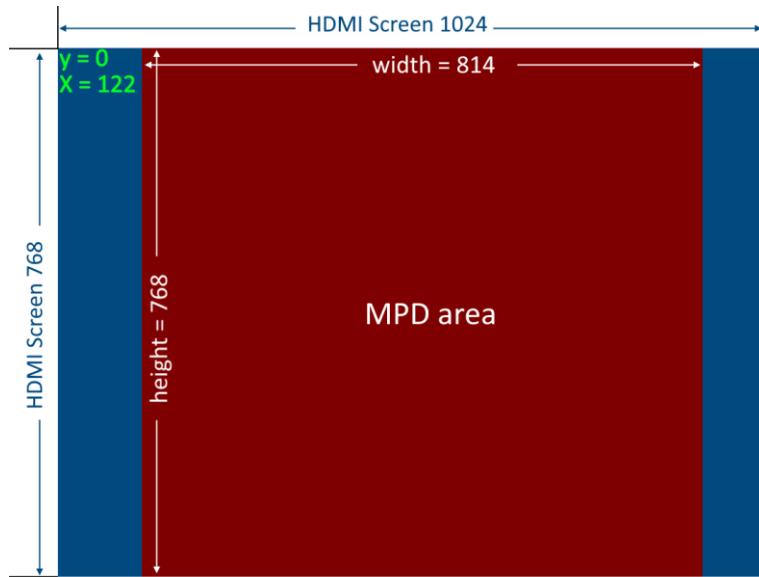
Simply use M3 screws between 12 mm and 20 mm in length to attach the MPD frame to the enclosure.  
You can screw them directly into the PLA — the tolerances are designed to allow this.



Here is the selector switch for choosing between Pilot and Co-Pilot mode. It can be mounted in different positions, depending on your setup. The switch is connected to **Pin 13** and a **GND** pin. **SELECTOR SWITCH IS NOT NEEDED IF YOU GO THE BBI-64 ROUTE!**

# Setting up the MPD Displays

As mentioned earlier, we use the viewport export method to display the MPDs on the HDMI screens. To do this, simply adjust your MonitorSetup.lua accordingly. You'll find all the details on how to configure external displays in DCS here: [https://wiki.hoggitworld.com/view/Exporting\\_MFCD\\_Displays](https://wiki.hoggitworld.com/view/Exporting_MFCD_Displays)



In this example, we assume a primary monitor resolution of 2560x1440.

The exported MPD viewports will be positioned relative to this main display.

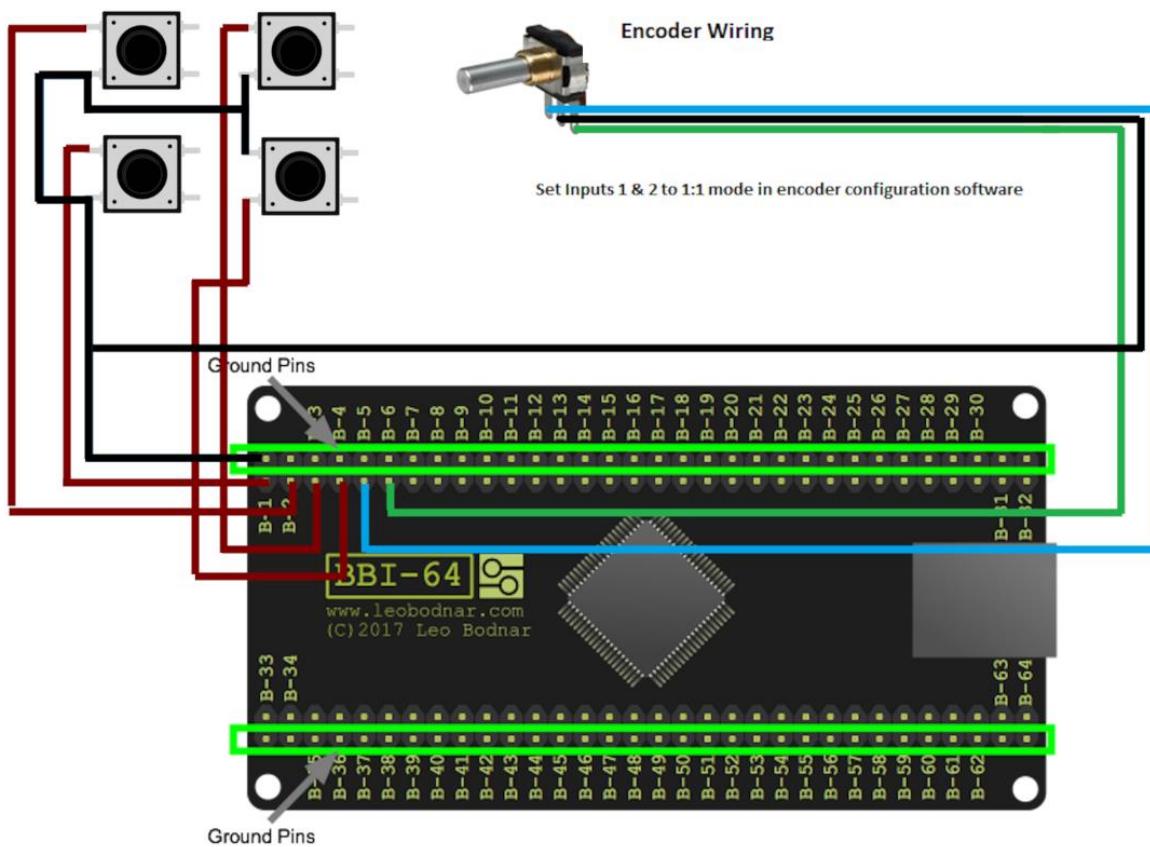
```
1 MPD = function(p) return p; end;
2 name = _("APACHE");
3 Description = 'Left MFCD on the left monitor, Right MFCD on the right and camera on the center'
4 Viewports =
5 {
6   Center =
7   {
8     x = 0;
9     y = 0;
10    width = 2560;
11    height = 1440;
12    viewDx = 0;
13    viewDy = 0;
14    aspect = 1.77777777;
15  }
16 }
17 LEFT_MFCD =
18 {
19   x = 2682;
20   y = 0;
21   width = 814;
22   height = 768;
23 }
24
25 RIGHT_MFCD =
26 {
27   x = 3706;
28   y = 0;
29   width = 814;
30   height = 768;
31 }
32
33 TEDAC =
34 {
35   x = 4726;
36   y = 0;
37   width = 812;
38   height = 768;
39 }
40
41 EUFD_plt =
```

Annotations on the code highlight specific values: +122 is shown near the y=0 line for the Center and Left\_MFCD viewports, and +1024 (LEFT MPD SCREEN) is shown near the x=2682 line for the Left\_MFCD viewport. The Right\_MFCD and TEDAC annotations are also present.

## Tip to save display outputs:

I'm installing the MPDs together with my new TEDAC (which also uses the Wisecoco 8-inch screen) and the EUFD (WIP with a 800x480 screen) into a [complete dashboard \(WIP\)](#) that can be mounted into my modular cockpit. Using the [StarTech.com USB 3.0 to Quad HDMI Adapter – USB to 4x HDMI Monitor Converter](#), I can connect all displays to the PC using just a single USB port.

# Setup with a Leobodnar BBI-64



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

**WARNING!** Windows only registers 32 buttons! But don't worry — DCS recognizes all 64 buttons of the BBI-64!

**💡 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're using more than one BBI-64 (which you probably will if you're building two MPDs), you'll need to flash a different firmware onto each board using the **HidFlasher** tool (available on the Leobodnar website). Without this step, Windows won't be able to tell the devices apart. Leobodnar provides 10 unique firmware versions specifically for this purpose.

**🔧 Tip:** You can rename the devices in the Windows Registry.

For example, instead of showing up as "BBI64-1" and "BBI64-2", they could appear as "**AH-64D MPD - Left**" and "**AH-64D MPD - Right**" — much easier to manage in DCS controller setup!

# Stay tuned....

