

In this article, we will help you create a custom DMD to use with Batocera. We are using ZeDMD as a basis as it's a cost-effective yet powerful solution, and now fully supported on Batocera 40+.

First, the final result



Step 1 : purchase the hardware

The shopping list:

- 2 dmd panels (don't forget to choose 2 of them) [DMD Panels](#)
- 1 ESP32 and its shield [ESP32+shield](#)
- 16 male/male 10cm jumpers [Jumpers](#)
- 1 power supply [Power supply](#)
- 1 plug [Plug](#)

You probably already have a plug at home.

Total cost including shipping: 39€ (on 2024 April 8).

P4 LED screen panel module 256*128mm 64*32 pixels 1/16 Scan Indoor 3in1 SMD RGB Fu...
7,79 €
Shipping: 11,91€
Estimated delivery between Apr 25 - 29
Free returns

✓Choice 3C Computer Peripheral Store
ESP32 Development Board Expansion Board Compatible with ESP32 WIFI Bluetooth modul...
ESP32-Board,CHINA
6,37 €
Shipping: Free shipping
Estimated delivery on Apr 18
3-day delivery

✓Choice Ci-Boom Store
20W AC 110V 220V To DC 5V 20W 25W 50W Switching Power Supply Module Transformer AC11...
20W 5V 4A
4,86 €
Shipping: Free shipping
Estimated delivery on Apr 18
3-day delivery

✓Choice ETERNALFAR Store
Dupont Jumper Wire Line 10CM 20CM 30CM Male to Male + Female to Male + Female to F...
Male VS Male,40pin,10cm
0,99 €
Shipping: Free shipping
Estimated delivery on Apr 18
6-day delivery

Step 2 : Put the hardware together

Step 2.0 - external links

- [ZEDMD project](#)
- [Original tutorial \(en\)](#)
- [Original tutorial \(fr\)](#)
- [Video tutorial \(fr\)](#)

The original tutorials are not related to batocera.linux, but may be very useful to get some reference information.

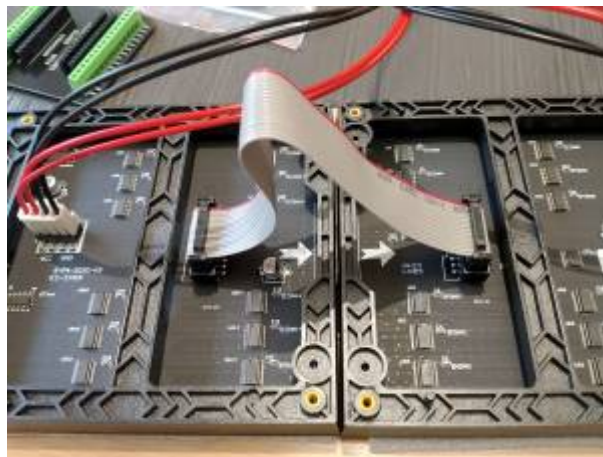
The hardware part is the same for Batocera. However, the course of action for software configuration at the end is a little bit different.

Step 2.1 - Cut and prepare the power wire



Step 2.2 - Plug LED matrices together with data and power wires

The data and power wires for the matrices are usually bundled with the matrices. Just plug them. Note that white arrows must go from left to right.



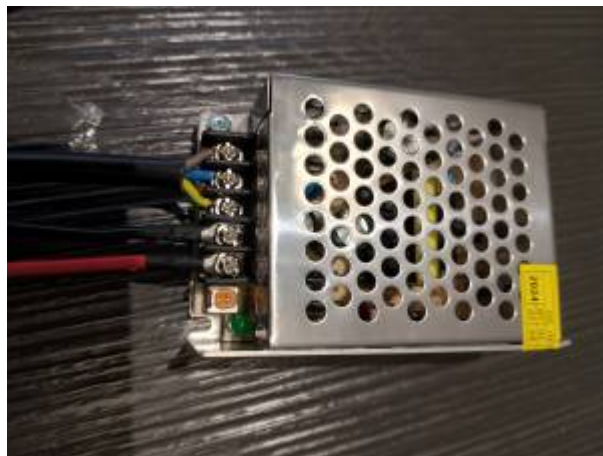


Step 2.2 - Plug the AC/DC power converter

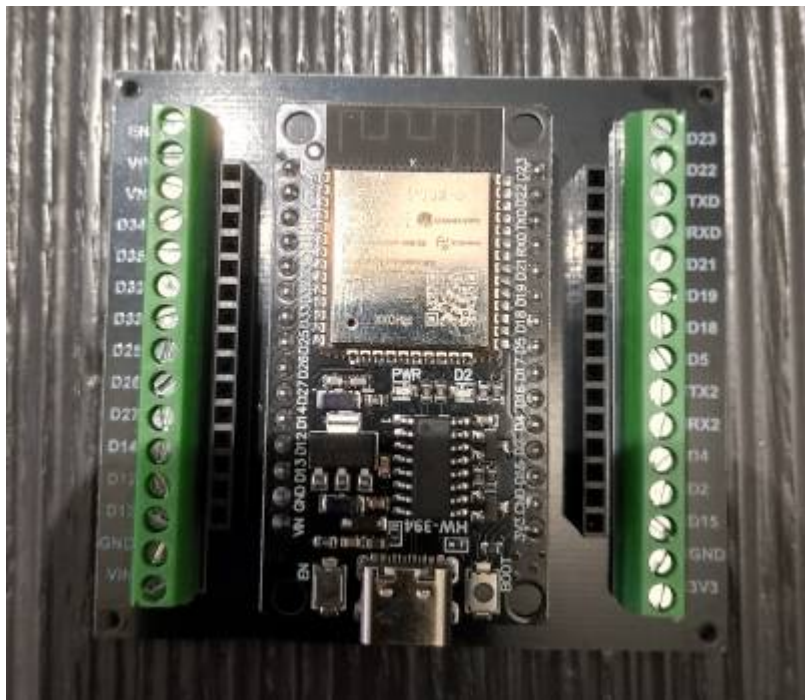
Plug the 220V power on the AC (if you live in a 110V country, make sure you have the right AC/DC converter, and obviously the right plug). **Be careful** with the color code of the wires: Phase (brown) / Neutral (blue) / Ground (yellow/green).

Plug the 5V connector, and make sure you connect the right colored wires.

The orange screw on right of the AC/DC converter can be used to adjust the 5V voltage. It is necessary in case the image is not perfect, and the LEDs are “bleeding” a little bit. See the pictures below, in general, adjusting the screw to power down a bit the the voltage makes the image better.



Step 2.3 - Microcontroller



Insert the ESP32 in its shield. This will be an easier way to plug the jumpers, as this will be a required part of the setup at a later stage.

Before plugging in the ESP32 board into the computer (ideally a computer running under batocera.linux), run the following command:

```
ls /dev/ttyUSB*
```

The result can be an error (no such file or directory) or a list of files (/dev/ttyUSB0 or /dev/ttyUSB0 /dev/ttyUSB1 or ...)

Now, plug the ESP32 USB-C connector to a USB port on that computer.

Run the `ls` command again the command, so that you can identify the device name the ESP32 microcontroller takes:

```
ls /dev/ttyUSB*
```

As a general rule, you would have gotten a `no such file or directory` error message before plugging in the ESP32, but now you should see a device name like `/dev/ttyUSB0` once plugged in. This means that the ESP32 can now be identified as a Unix pseudo-file `/dev/ttyUSB0`.

In case you had a `/dev/ttyUSB0` device before plugging in, you would now get `/dev/ttyUSB0` /`/dev/ttyUSB1` once the ESP32 is connected. This means that the ESP2 can now be identified as a Unix pseudo-file `/dev/ttyUSB1`.

Execute the following command, and don't forget to replace `ttyUSB0` by `ttyUSB1` if necessary, based on the previous explanation.

```
wget https://github.com/PPUC/ZedMD/releases/download/v3.6.0/ZedMD-128x32.zip
```

```
unzip ZeDMD-128x32.zip
wget
https://github.com/espressif/esptool/releases/download/v4.7.0/esptool-v4.7.0
-linux-amd64.zip
unzip esptool-v4.7.0-linux-amd64.zip
chmod a+x ./esptool-linux-amd64/esptool
./esptool-linux-amd64/esptool --port /dev/ttyUSB0 --chip esp32 write_flash
0x0 ./ZeDMD.bin
```

The first line downloads the ZeDMD firmware for a 128×32 matrix.

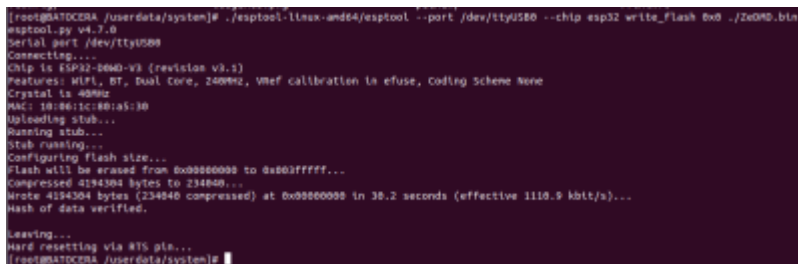
The second line unzips it.

The 3rd line downloads the tool we are going to use to flash the firmware on the ESP32 board.

The 4th line unzips that tool.

The 5th line makes the tool executable.

The last line flashes the firmware on the ESP32 device connected as /dev/ttyUSB0.

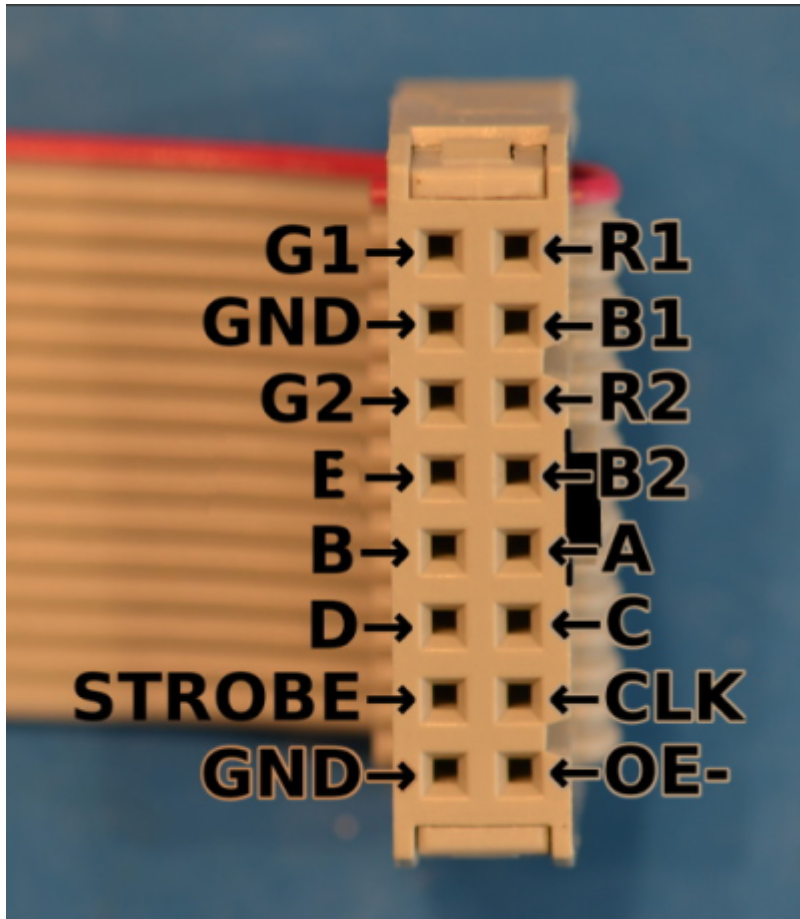


```
[root@BATOCERA /userdata/system]# ./esptool-linux-amd64/esptool --port /dev/ttyUSB0 --chip esp32 write_flash 0x0 ./ZeDMD.bin
esptool.py v4.7.0
Serial port /dev/ttyUSB0
Connecting...
Chip is ESP32-0640-V1 (revision v1.1)
Features: WiFi, BT, Dual Core, 240MHz, VREF calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 18:8e:1c:80:a5:30
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x0000ffff...
Compressed 4194304 bytes to 234840...
Wrote 4194304 bytes (234840 compressed) at 0x00000000 in 38.2 seconds (effective 1110.9 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
[root@BATOCERA /userdata/system]#
```

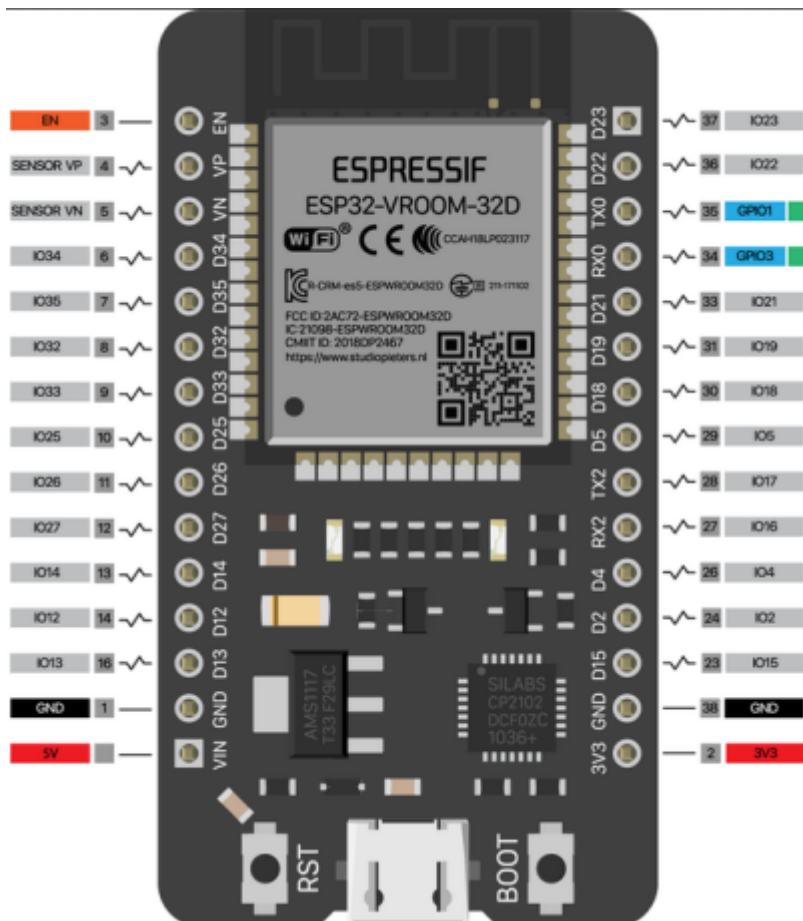
Step 2.4 - Plug the microcontroller

Check that your ESP32 is the 30-pin model (by counting the number of pins). There are several models, some with another number of pins, but this one seems to be the most common.

Now, you must plug the data wire to the ESP32 by respecting the pin codes as described in the following images.



```
#define R1_PIN 25
#define G1_PIN 26
#define B1_PIN 27
#define R2_PIN 14
#define G2_PIN 12
#define B2_PIN 13
#define A_PIN 23
#define B_PIN 19
#define C_PIN 5
#define D_PIN 17
#define E_PIN 22
#define LAT_PIN 4
#define OE_PIN 15
#define CLK_PIN 16
```



There are 15 pins to plug in total.

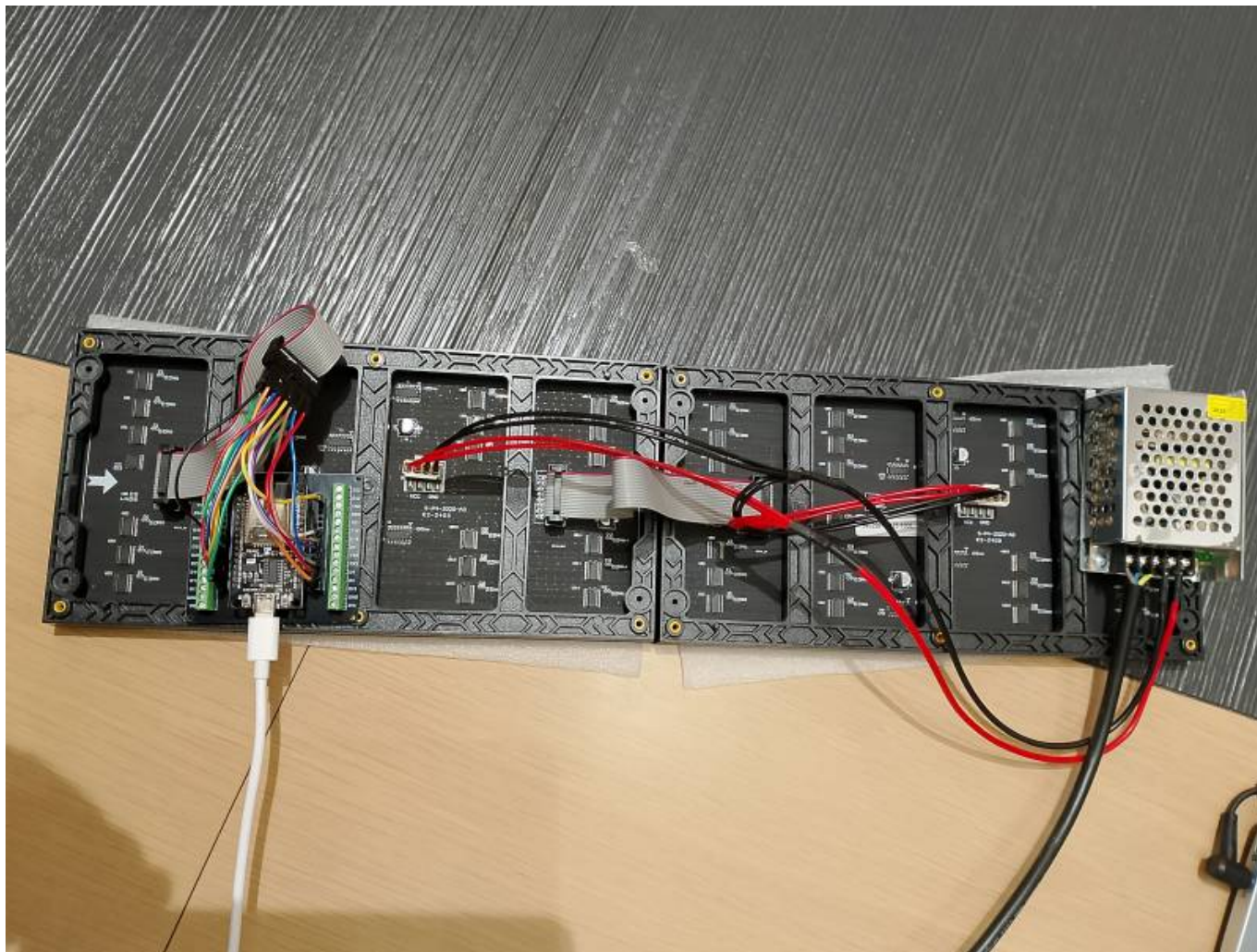
The pin E can be omitted.

The pin called LAT is the pin STROBE.

Example : plug a pin in the data cable on G1. G1 pin corresponds to pin IO26. Thus on the ESP32 side, plug it in the IO26.



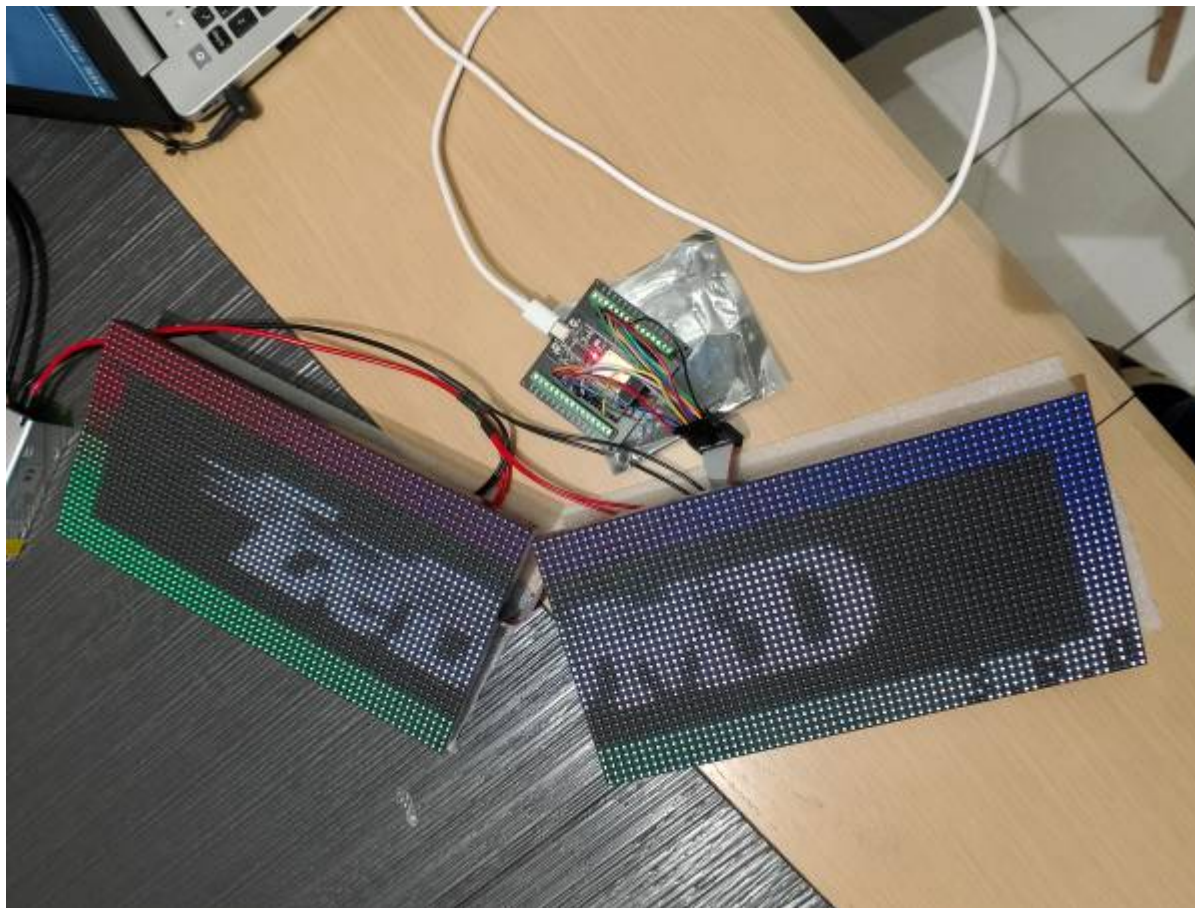
Step 2.5 - Final assembling



Step 2.5 - First boot

Plug your DMD the power wall socket. You should see the ZeDMD logo and the version.

The logo may be not perfect, as you can see on this picture, but no panic this is an easy fix: adjust the voltage to a slightly lower value to make it nicer (the orange screw on the power supply as explained at the beginning of this page).

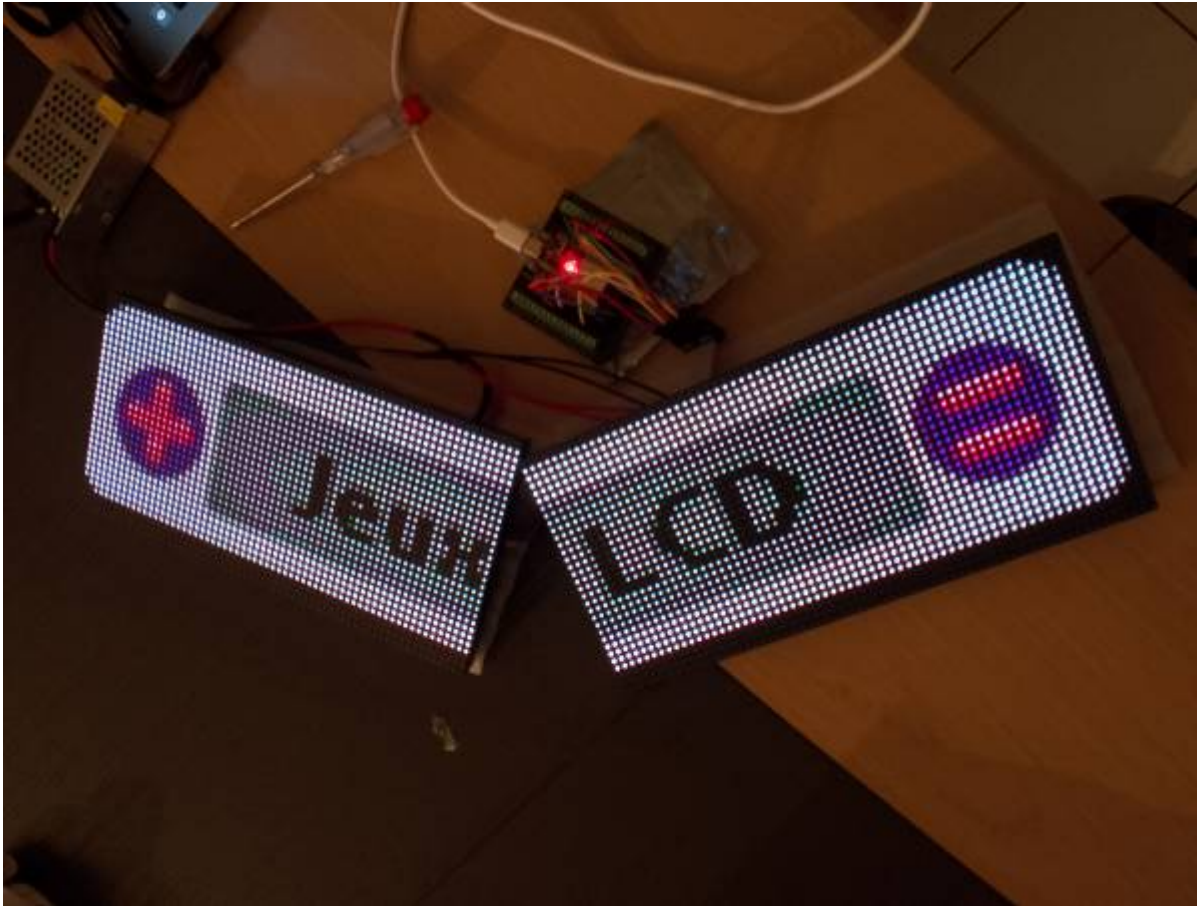


Step 3 - Batocera.linux software configuration

Plug your DMD with the USB-C cable to the Batocera box.

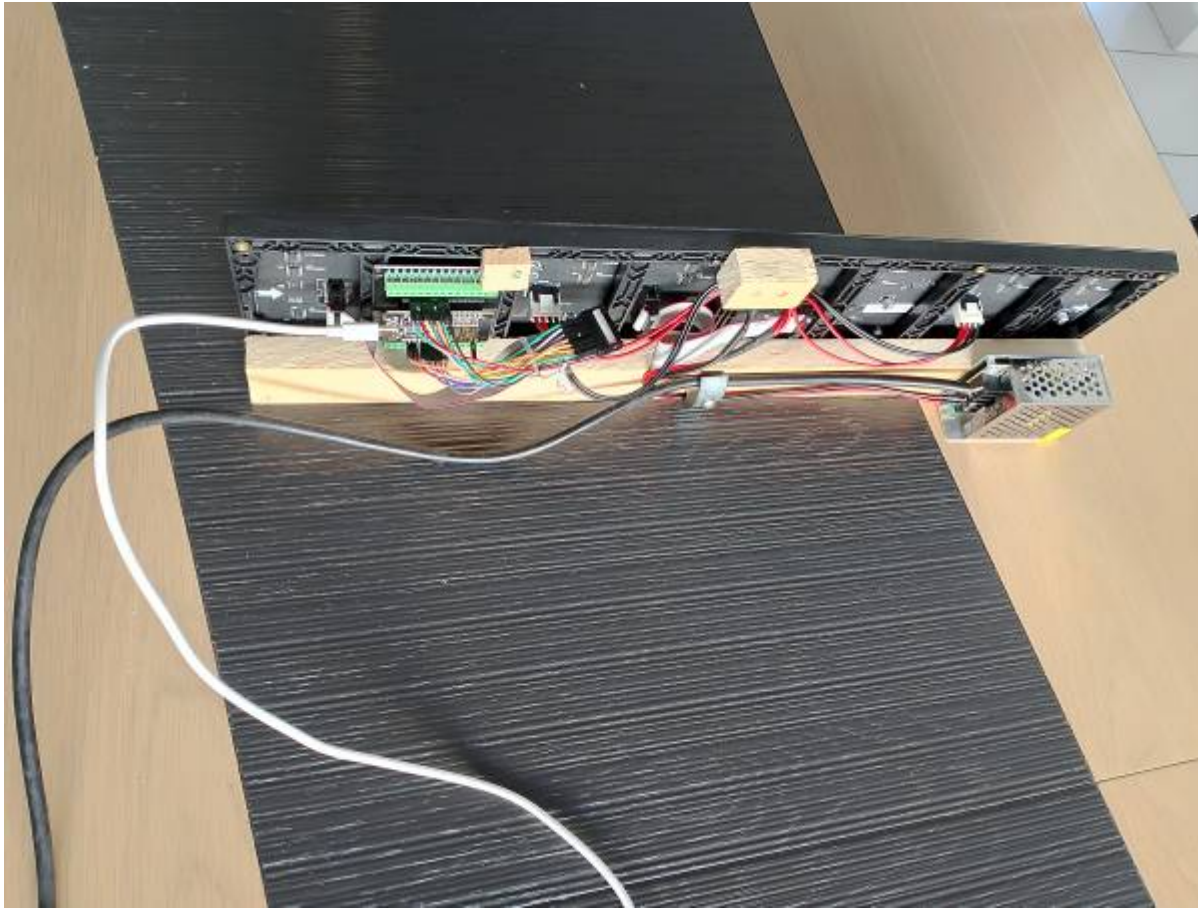
Go in **MAIN MENU** → **SYSTEM SETTINGS** → **SERVICES** and start (or stop/start) the `dmd_real` service.

Get back to the main menu, and switch to a different system, or select a game in EmulationStation. The images, or the names of the games (in case you don't have the right images) should appear on your DMD.



Step 4 : finalize the DMD hardware

Make sure you steadily lock the LED matrices, and if you're handy enough, make a nice case to wrap things up.



From:
<https://www.wiki.batocera.org/> - **Batocera.linux** - Wiki

Permanent link:
https://www.wiki.batocera.org/hardware:diy_zedmd?rev=1713995859

Last update: **2024/04/24 21:57**

