In this article, we will create a custom DMD.

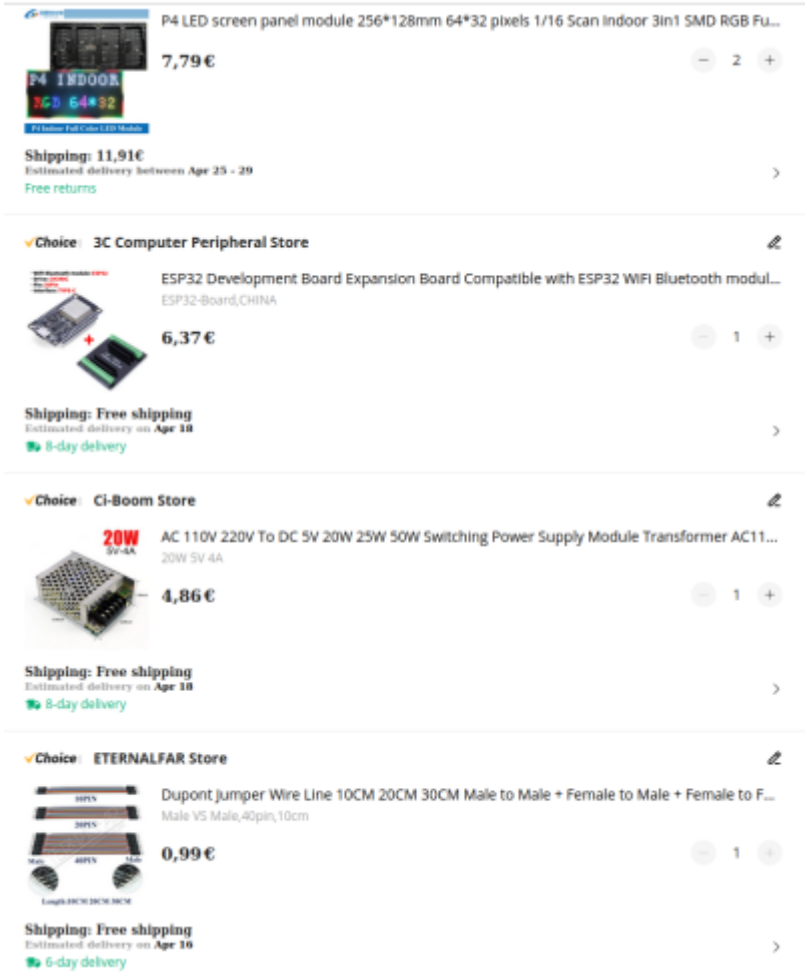
The result

Step 1 : buy the hardware

The list :

- 2 dmd pannels (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 price including shipping : 39€ (on 2024 april 8).



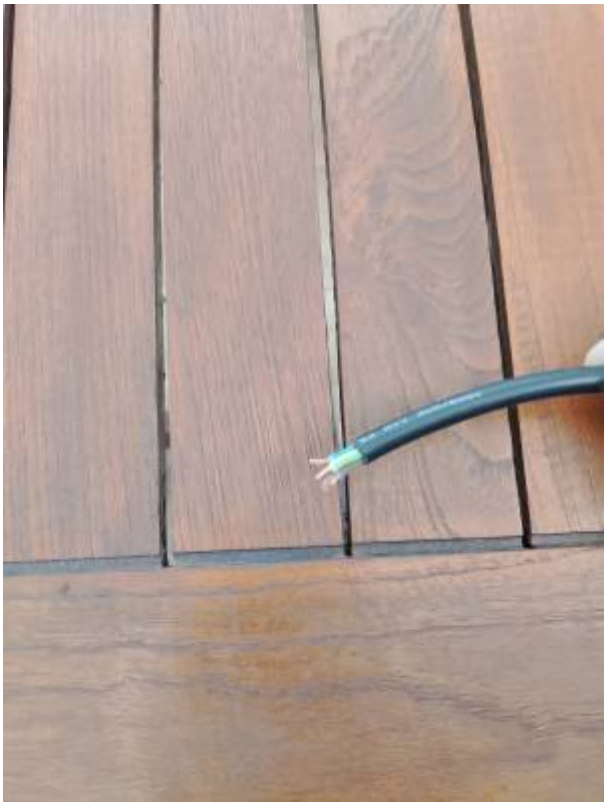
Step 2 : plug the hardware

Step 2.0 - external links

- ZEDMD project
- Original tutorial (en)
- Original tutorial (fr)

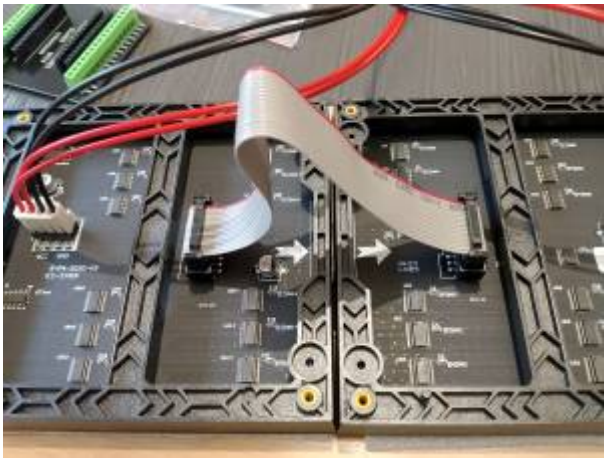
The original tutorials are not related to batocera.linux, but may be very usefull to get some informations. The hardware part is the same for batocera. Just configuration actions/software at the end changes.

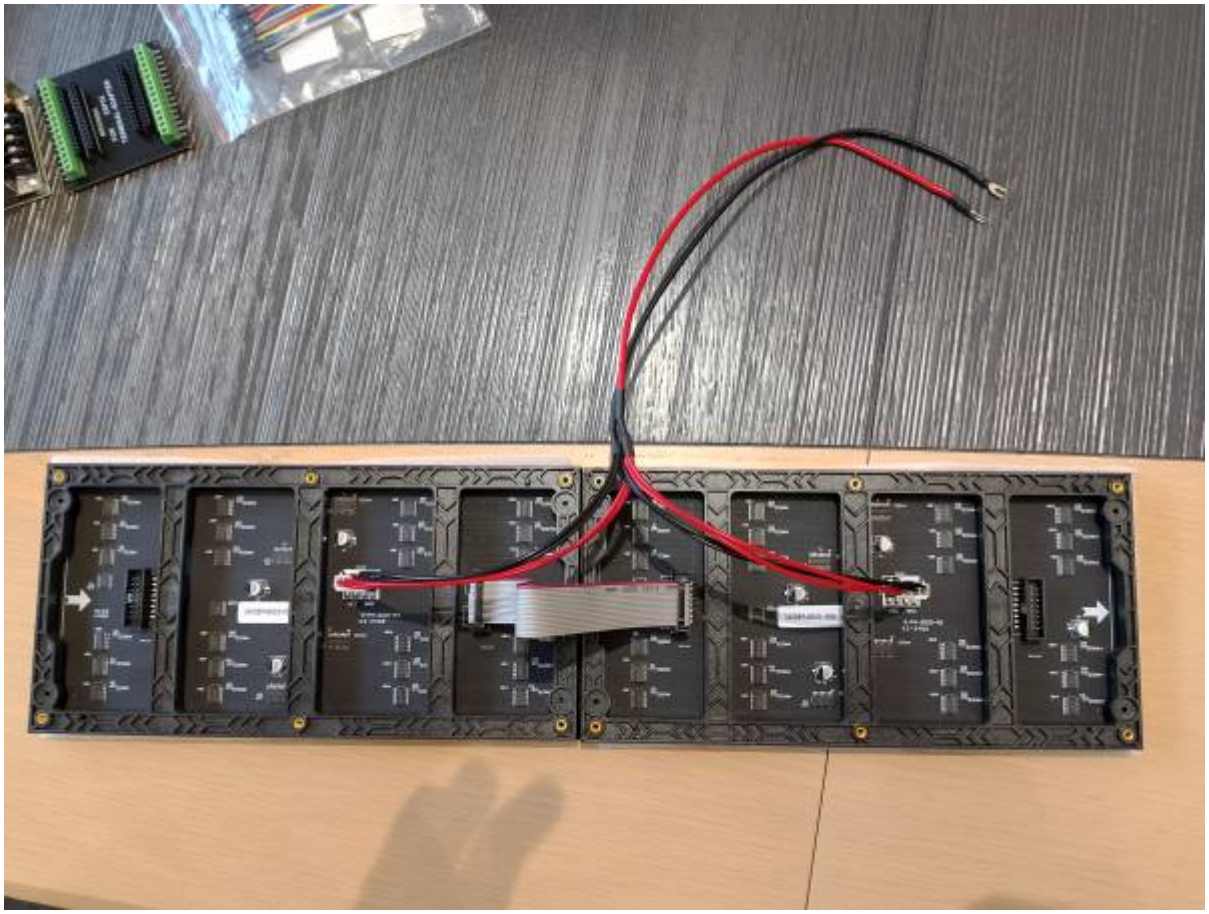
Step 2.1 - Cut and prepare the power wire



Step 2.2 - Plug matrices together with data and power wire

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



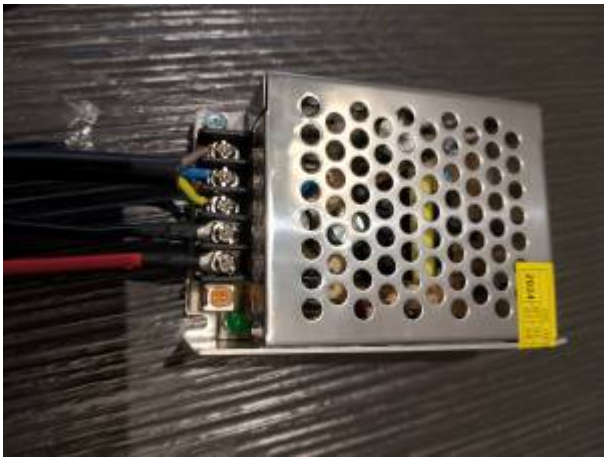


Step 2.2 - Plug the power

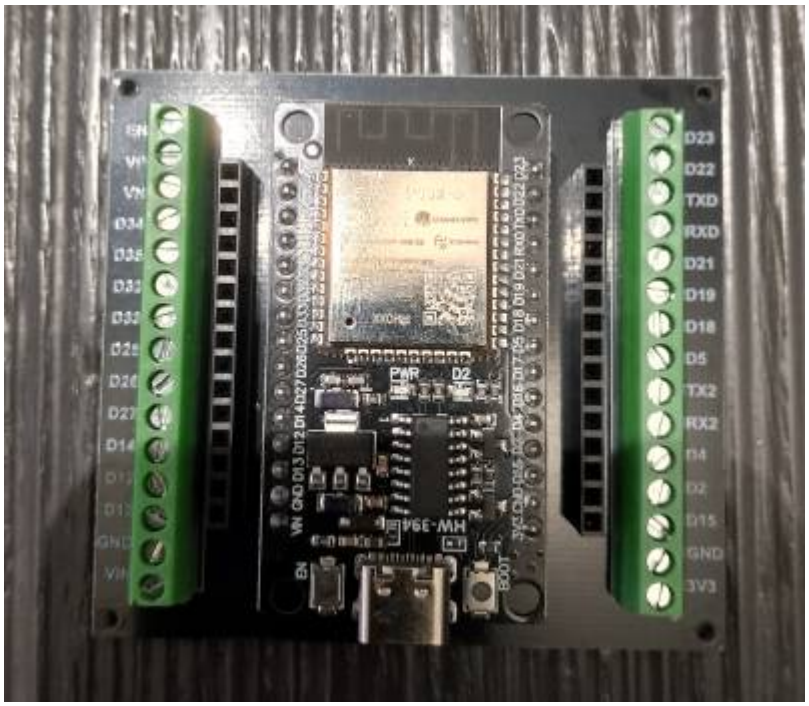
Plug the 220V power on the AC. Take care of colors. Phase (brown) / Neutral (blue) / Mass (yellow/green).

Plug the 5V. Take care of colors.

The orange screw on right can be used to adjust the 5V voltage. It is usefully in case the image is not perfect. Too much voltage make the image not perfect.



Step 2.3 - Microcontroller



Plug the esp32 in its socket.

Before plugin the esp32 on the computer (ideally 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 ...)

Plug it on a computer (ideally under batocera.linux) with a usb-c cable.

Rerun the command and identify the name the esp32 took.

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

in general, you'll get the error “no such file” before plugin, and “/dev/ttyUSB0” once plugged. This mean that the esp32 can now be identified as file /dev/ttyUSB0.

In case you had /dev/ttyUSB0 before plugin, you'll get /dev/ttyUSB0 /dev/ttyUSB1 after. This mean that esp32 can now be identified as file /dev/ttyUSB1.

Execute the following command by adapting ttyUSB0 to ttyUSB1 if necessary.

```
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 128×32 matrix.

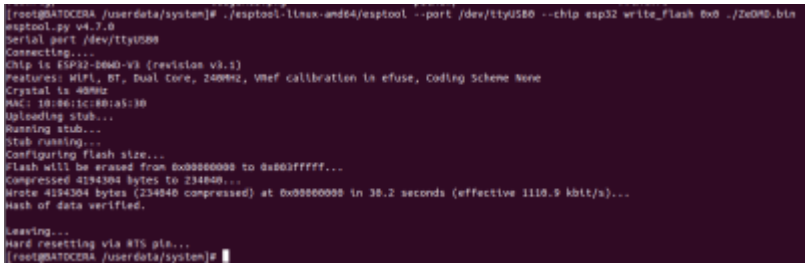
The second line unzip it.

The 3rd line downloads the tool used to flash the firmware.

The 4th line unzip the tool.

Th 5th line make the tool executable.

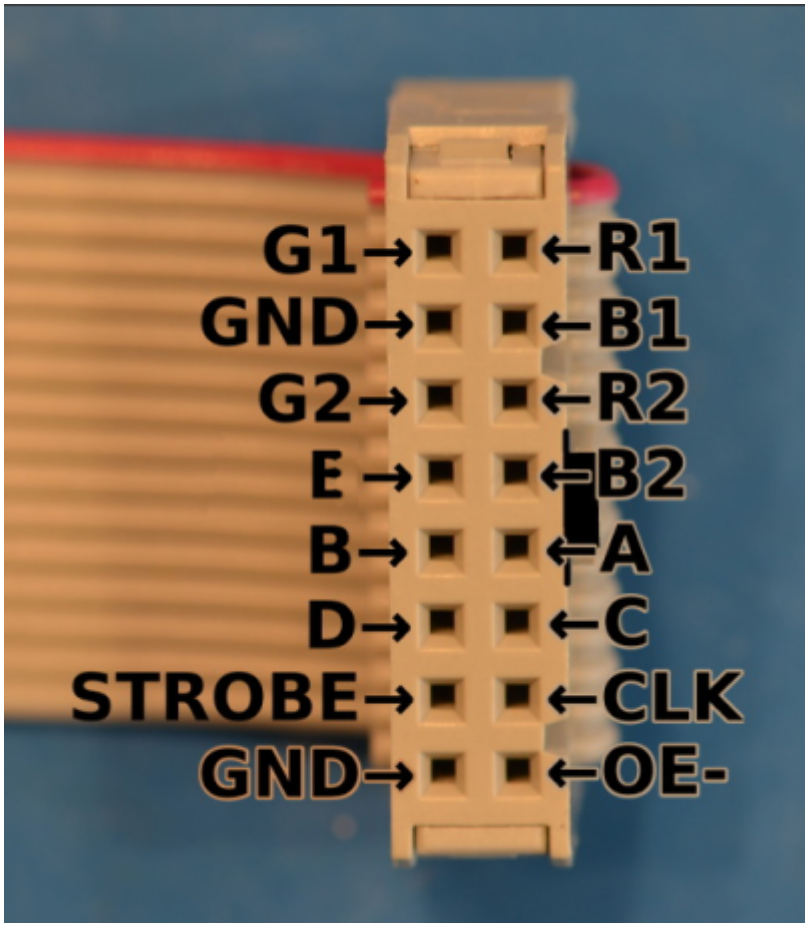
The last line flash the firmware on the device /dev/ttyUSB0.



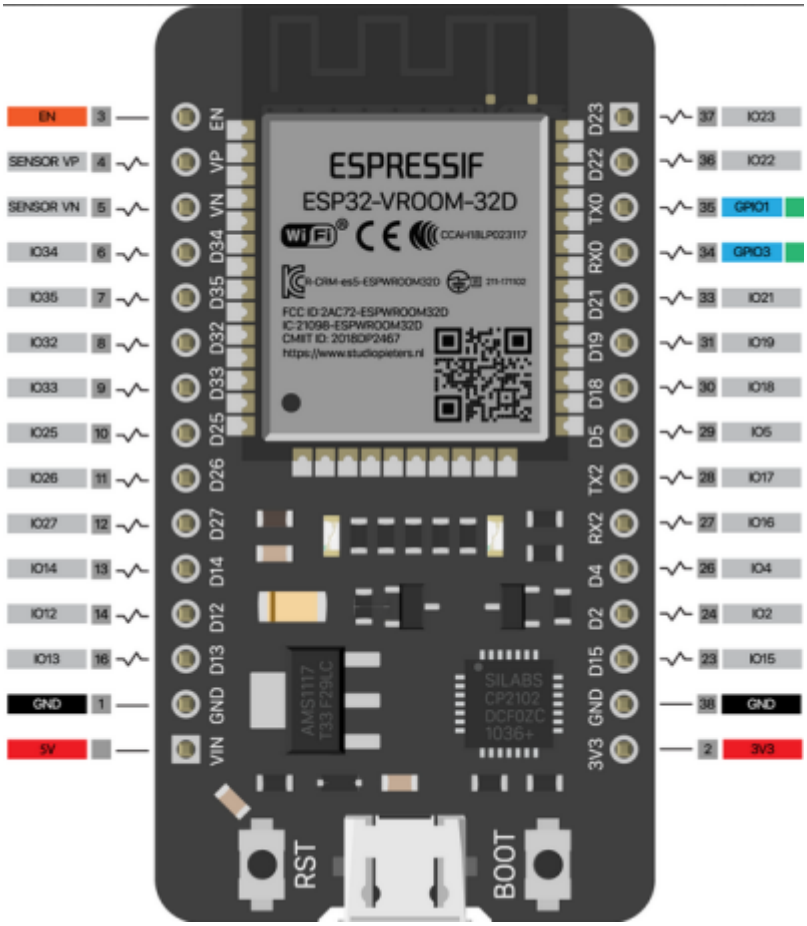
Step 2.4 - Plug the microcontroller

Check that your esp32 is the 30 pins model (by counting the number of pins).

Now, you must plug the data wire to the esp32 by respecting the pin codes as 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.

The pin E can be ommited.

The pin LAT is the pin STROBE.

Example : plug a pin in the data cable on G1. G1 pin corresponds to pin io 26. Thus on the esp32 side, plug it in the io 26.

to be continued