

FreeDV Digital Voice - ezDV Module Build process

<https://github.com/tmiw/ezDV/tree/HW-v0.7>

To build this standalone freedv digital voice module you need to download the assets from this web site in order to order the pcb from JLCPCB who will also assemble all the main required components on the board. This can be quite a challenging process at first glance but once it is done it is fairly straightforward.

The files you really need for the boards are:

ezDV.csv - this is the bill of materials or list of components required to be assembled on the board

ezDV.pdf - this is the circuit schematics for the board

output.zip - this is the Gerber files required by JCLPCB in order to create the pcb and position the components

Also included in this last zip file is the position files used to locate the components on the pcb as part of the assembly process by JCLPCB, they are:

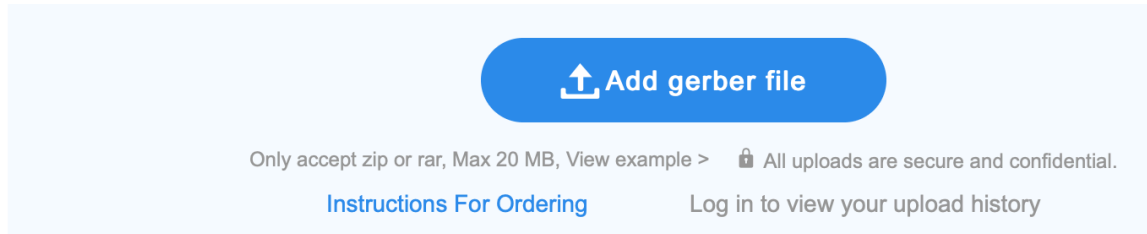
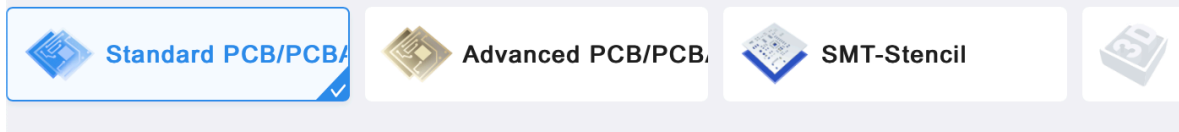
ezDV-bottom-pos.csv

ezDV-top-pos.csv

You will need to copy the ezDV-top-pos.csv file from the output.zip folder into a separate folder, leaving the original where it is, it needs to be within the Gerber file uploaded to JCLPCB and you need to upload the copied file separately as part of the process of ordering the assembled boards. The JCLPCB web site does provide guidance on how to place the order and I generally followed the steps shown accepting all the default options presented, noting that this particular board is a complex board using 6 layers so need to use the Standard option instead of Economy also as a result of a couple of the components that need to be placed.

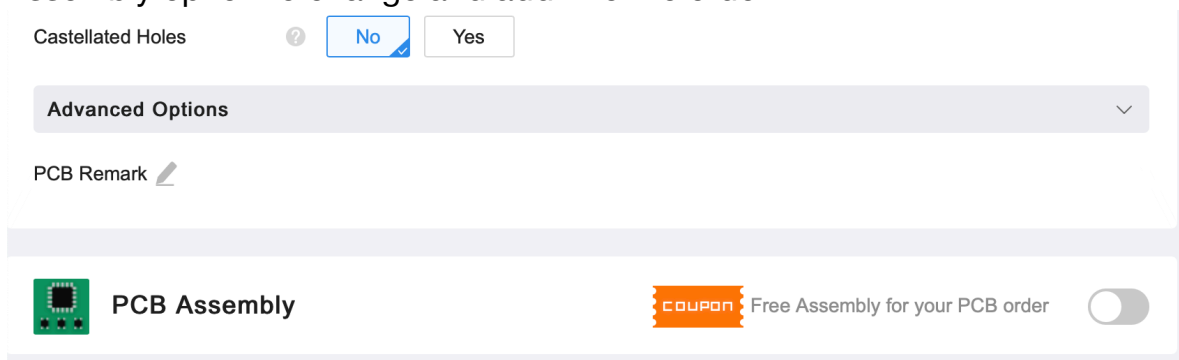
<https://cart.jlpcb.com/quote?orderType=1&stencilLayer=4&stencilWidth=50&stencilLength=50>

Click the "Add Gerber File" button to start the process, various options below this will change as a result of the file output.zip being uploaded



I left the options as they changed as default without changing them before moving to next step.

Because we want the components to be added to the board, click the PCB Assembly option to change and add it to the order:



You will now see further options that can be configured below this, again I left them as default. Note that there is an option to only assemble 2 out of the minimum order of 5 PCB's, I left this at 5 so I can ask some friends to purchase the assembled boards that I don't need to help lower the overall cost to a more respectable value per board rather than having to stand the full cost of the minimum order.

The boards will take approximately 10-14 days to arrive and then you will need to finish off the assembly required before testing and putting into use.

The overall cost of the module will increase as there are some other items needed that are not included in this process such as the battery and a pair of pcb mounted TRRS sockets and cables required to interface the board to the radio plus a suitable enclosure.

Battery - https://www.amazon.co.uk/s?k=3.7v+lithium+ion+battery&crd=WPCH3CWYHQUU&srefix=3.7v%2Caps%2C288&ref=nb_sb_ss_ts-doa-p_1_4

I ordered sockets to attach to the pcb so that I can ensure I can disconnect the battery but also ensure it gets connected with the correct polarity when reattached to the board or replaced in the future e.g. https://www.amazon.co.uk/YIXISI-Pairs-Micro-Connector-Female/dp/B08JV8ZH72/ref=sr_1_2_sspa?crid=A6LUJ5QB01YI&keywords=JST-PH%2B2.0%2Bbattery%2Bconnectors&qid=1691854680&prefix=jst-ph%2B2.0%2Bbattery%2Bconnectors%2Caps%2C91&sr=8-2-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&th=1

TRRS Audio Connectors - <https://uk.rs-online.com/web/p/jack-plugs-sockets/2596700?gb=b>

Suitable enclosure - TBC

Reference for building ESP32-S3 programming environment
For Linux and Mac OSX:

<https://docs.espressif.com/projects/esp-idf/en/release-v5.0/esp32s3/get-started/linux-macos-setup.html>

For Windows:

<https://docs.espressif.com/projects/esp-idf/en/release-v5.0/esp32s3/get-started/windows-setup.html>

These are the steps that I followed on my Apple M2 MacBook Air and tried out again on my old 2012 Apple MacBook Pro to prove they worked
All following commands are run within a Terminal session:

Install Homebrew from <https://brew.sh/>:

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Install git using Homebrew: `$ brew install git`

Install Python3 using: `$ brew install python3`

Install CMAKE and NINJA using: `$ brew install cmake ninja dfu-util`

It is strongly recommended to also install [ccache](#) for faster builds. If you have [HomeBrew](#), this can be done via: `$ brew install ccache`

This step worked fine on my M2 MacBook Air but after testing these instructions on a 2012 MacBook Pro this step failed and hung while installing rust as a dependency at the make step, I left it running for hours before terminating the session. The following steps continued to work ok without it.

Install ESP environment:

cd to target root directory - I used Users/paulhelliwell/

```
$ mkdir -p ~/esp
```

```
$ cd ~/esp
```

```
$ git clone -b release/v5.0 --recursive https://github.com/espressif/esp-idf.git
```

```
$ git clone https://github.com/tmiw/ezDV/
```

```
$ cd ezDV
```

```
$ git submodule update --init --recursive
```

```
$ cd firmware
```

```
$ idf.py build
```

To flash the image created in Users/paulhelliwell/ezDV/firmware onto the device:

Connect device to USB port

Check where the port is located using ls /dev

Insert port information in the command below where (PORT)

Run command in terminal from firmware directory:

```
$ run idf.py -p (PORT) flash monitor
```

or

Project build complete. To flash, run this command:

```
$ /Users/paulhelliwell/.espressif/python_env/idf5.0_py3.9_env/bin/  
python ../../esp/esp-idf/components/esptool_py/esptool/esptool.py -p  
(PORT) -b 460800 --before default_reset --after hard_reset --chip esp32s3  
write_flash --flash_mode dio --flash_size detect --flash_freq 80m 0x0 build/  
bootloader/bootloader.bin 0x8000 build/partition_table/partition-table.bin  
0xf000 build/ota_data_initial.bin 0x20000 build/ezdv.bin 0x7f8000 build/  
http_0.bin
```

NOTE - Depending on machine you run this on, you may notice a different python sub version used in the first line of this command set - on my old MacBook Pro after installing python3, the version shown for this command was py3.11

Now follow the instructions at <https://github.com/tmiw/ezDV> for final assembly, programming and testing the board and then configuring it for your

radio to get started on FreeDV Digital Voice.

Refer to <https://freedv.org> for more information about FreeDV and further links to more information about this digital mode along with spot frequency information to find operators using this mode around the world.