

RPI5 EEPROM Modification

„PSU_MAX_CURRENT=5000“

Anabling the use with a non-USB-PD Bench-Power Supply

Disclaimer / Warning

Do this at your own risk!

There is no warranty and support for the modifications described!

You are responsible if anything goes wrong with your RPi5!

This modification applies only and exclusively to the RaspberryPi 5 SBC (Single Board Computer).

The added line „PSU_MAX_CURRENT=5000“ is telling the RPi5 firmware that when you use a NON-PD (bench) power supply or Adapter-Charger with USB C cable and that it should not attempt USB-PD and assume that it is your's power supply 5A capable.

This will allow the RPi5 to boot also from other or older wall Adapters with usb-C cable

You really need to have a 3+ amp power supply for Rpi5 to work.

If you use the RPi5 as a headless streamer without a display and hard disk etc., the Ifi iPower X (5V, 3A) for example, will work. Tests showed that older type apple iPad chargers with a USB-C cable also worked after this line was added

When running without any peripherals, the RPi consumes less than 1A after boot, so a smaller supply will be fine as well, as long as it can deliver for a short time during boot more current.

If you want to use the RPi5 with a disk, display, etc., you should have a power supply with a minimum of 25Watt (5V, 5A) of power.

If you want to boot the RPi5 from the GPIO 5V pins, that will work regardless of the max current line, but it will demand stable 5V and enough current capabilities. Tests showed that it depends very much on the PSU used and you need to try till you find a working set-up.

[Detailed workflow how to edit the EEPROM, adding „PSU_MAX_CURRENT=5000“](#)

For starters you might need an operating system to be able to run your RPI5:
There is a detailed instruction page under this link with more information
<https://www.raspberrypi.com/documentation/computers/getting-started.html>

To set PSU_MAX_CURRENT=5000 you need an empty 32GB+ micro SD card.
Download Raspberry Pi OS from the link:
<https://www.raspberrypi.com/software/>

Raspberry Pi OS

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.

Open the download imager_x.x.x.exe

Click Install and wait

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

[Download for macOS](#)

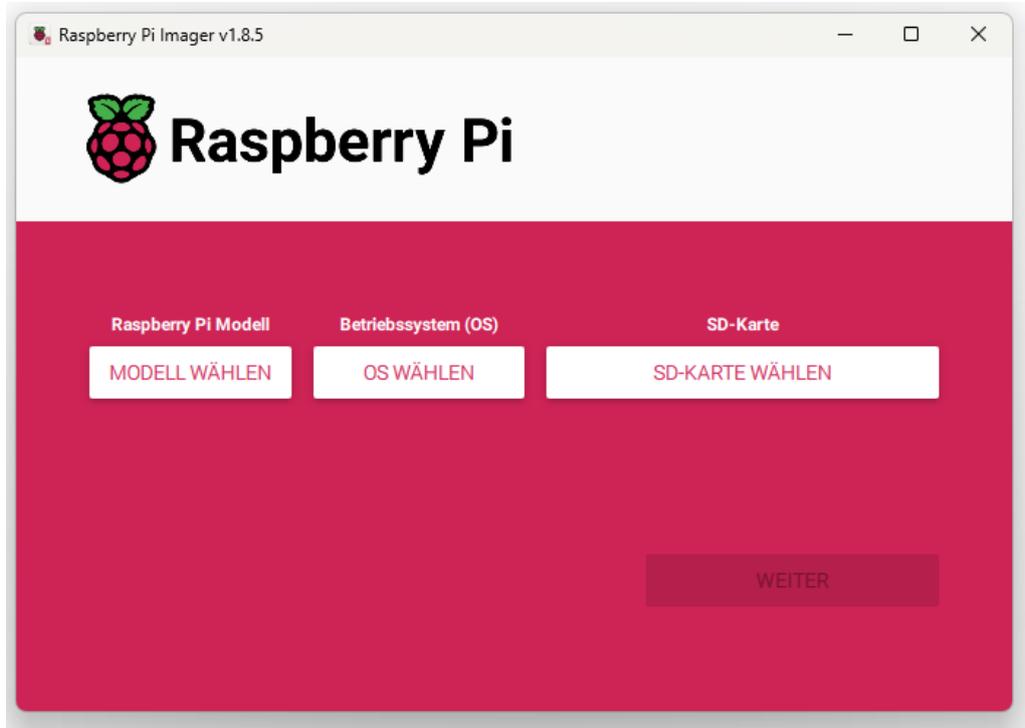
[Download for Ubuntu for x86](#)



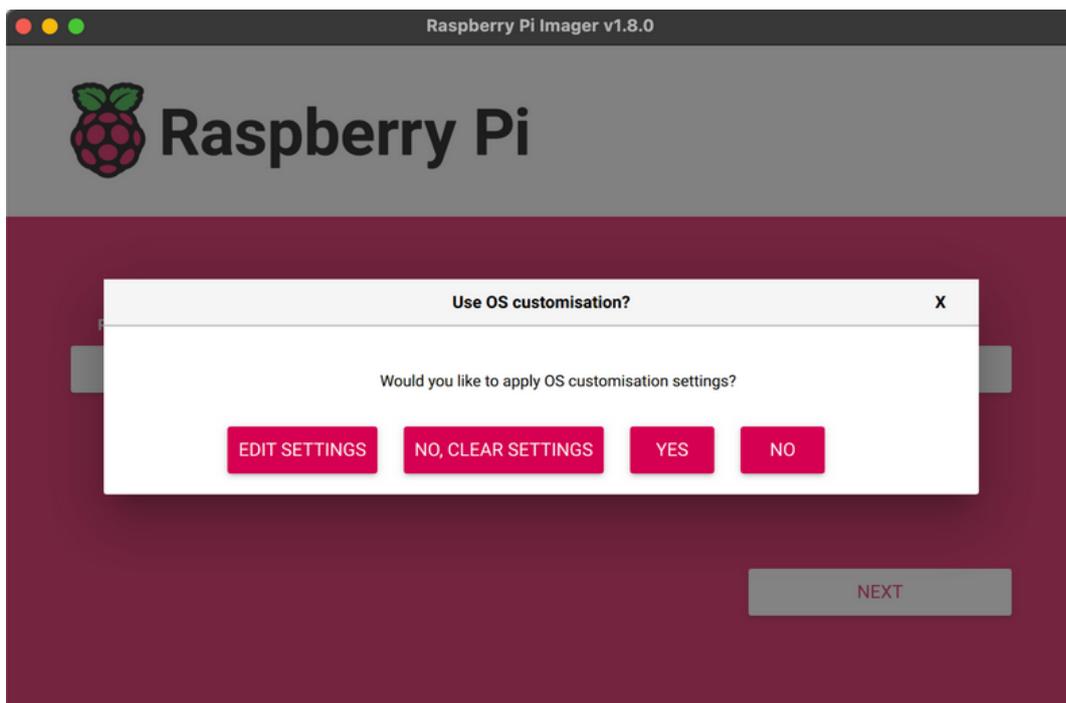
This pop up windows shows up now and use the 3 buttons to make your selections:

- Model: „RPI5“
- OS: „Raspberry Pi OS (64-BIT)“ is the recommended one
- SD Card: Select the Card to flash to – it must have formatted in FAT32

Click „Next“ („weiter“ in the image below).



Now the next pop-up screen („Use OS Customizations?“) comes up hit „Edit Settings“ (NOT „YES“ yet).



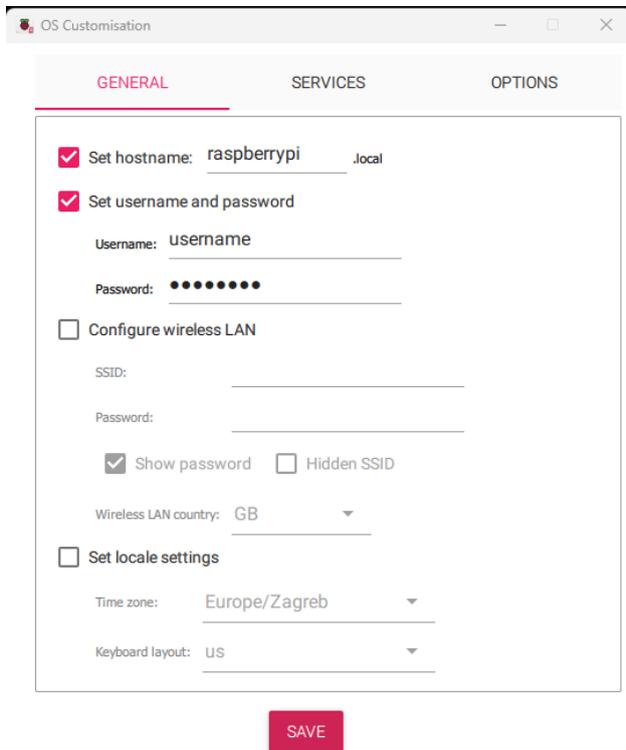
This pop up windows shows up now.

Go to the FIRST TAB „GENERAL“:

Check box „Set hostname“, you can leave „raspberrypi“.

Check next box and Fill in a „username“ and „password“ of your choice.

IF your Keyboard is NOT „US“ change it in your local language setting.

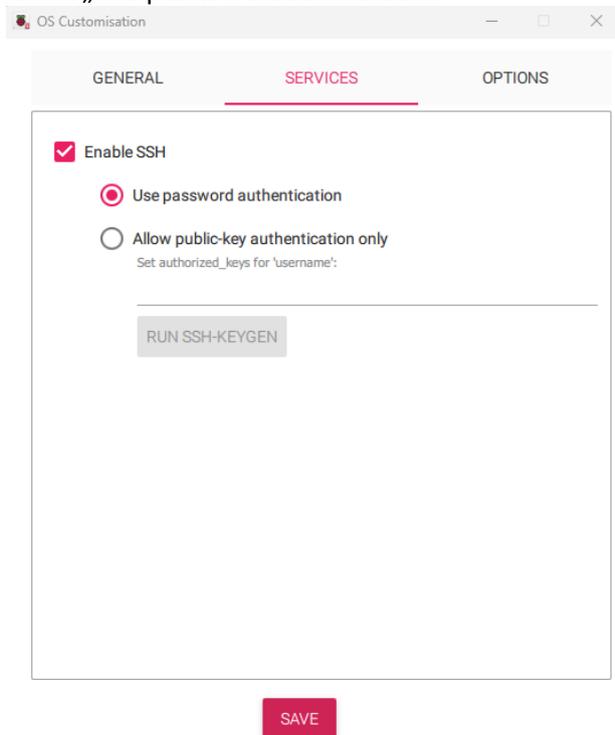


The screenshot shows the 'OS Customisation' window with the 'GENERAL' tab selected. The 'Set hostname' checkbox is checked, with 'raspberrypi' entered in the field. The 'Set username and password' checkbox is also checked. The 'Username' field contains 'username' and the 'Password' field is filled with dots. Below these are options for 'Configure wireless LAN' (unchecked), 'Show password' (checked), and 'Hidden SSID' (unchecked). The 'Wireless LAN country' is set to 'GB'. The 'Set locale settings' checkbox is unchecked. The 'Time zone' is set to 'Europe/Zagreb' and the 'Keyboard layout' is set to 'US'. A red 'SAVE' button is located below the form.

Go to the SECOND TAB „SERVICES“:

check „Enable SSH“

check „Use password authentication“

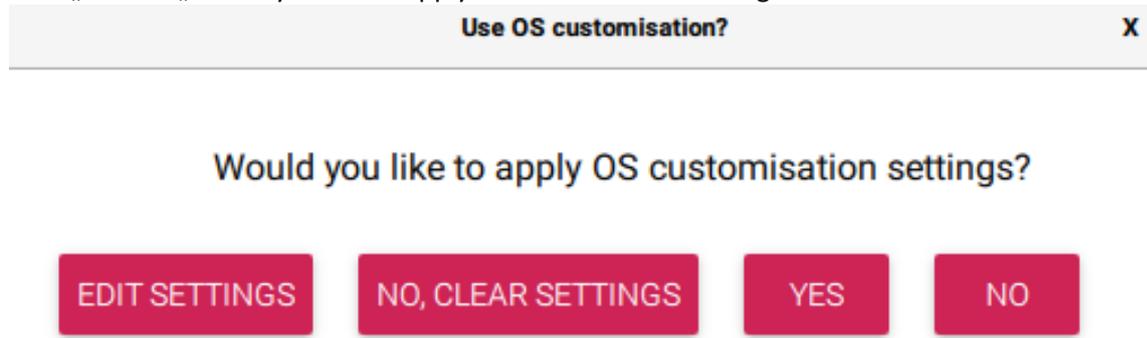


The screenshot shows the 'OS Customisation' window with the 'SERVICES' tab selected. The 'Enable SSH' checkbox is checked. The 'Use password authentication' radio button is selected, while 'Allow public-key authentication only' is unselected. Below the radio buttons is a text input field for 'Set authorized_keys for 'username':' and a 'RUN SSH-KEYGEN' button. A red 'SAVE' button is located below the form.

Click „SAVE“.

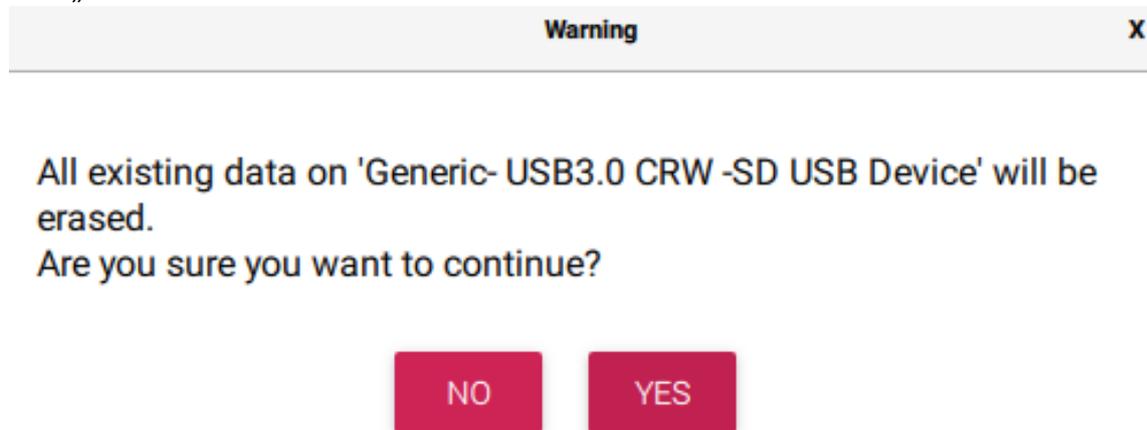
The next pop-up-screen comes up.

Click „YES“ on „Would you like to apply OS customization settings“.



There will be a message if you are sure you want to overwrite all Data on the Flash Card.

Click „YES“



Click „YES“ again.

The card will be flashed now – be patient, it might take several minutes... get coffee.

When the image is flashed, insert the sd card into the RPi5.

Make sure it is connected to your local LAN (if not you should have setup WIFI in the earlier „edit setting“).

Apply power and let it boot.

Now we need to connect your PC with the RPi5 with a SSH Client (Putty).

IF NEEDED: Download „Putty ssh client“

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Scroll down to this position and download „putty.exe“

Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel and pterm), but you can download standalone binaries one by one if you prefer. (Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

putty.exe (the SSH and Telnet client itself)

64-bit x86:	putty.exe	(signature)
64-bit Arm:	putty.exe	(signature)
32-bit x86:	putty.exe	(signature)

pscp.exe (an SCP client, i.e. command-line secure file copy)

64-bit x86:	pscp.exe	(signature)
64-bit Arm:	pscp.exe	(signature)
32-bit x86:	pscp.exe	(signature)

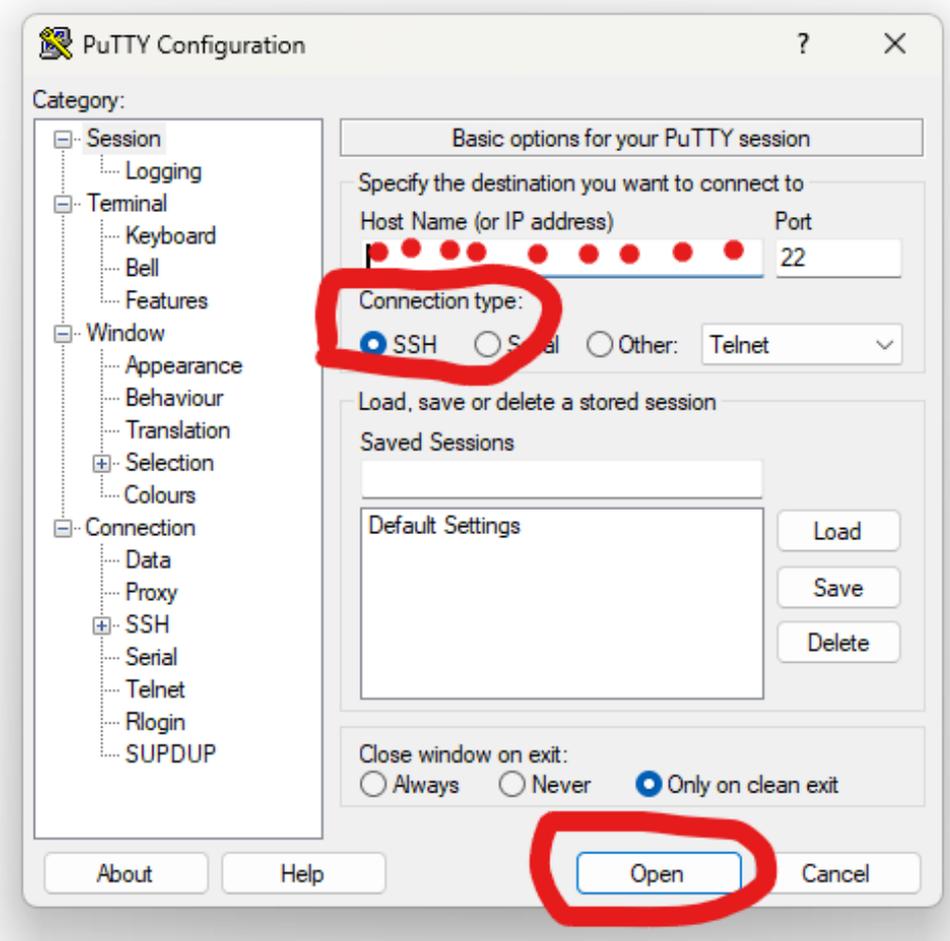
... for ... for SFTP client i.e. general file transfer sessions much like FTP)

If you already have Putty – continue here:

Run „putty.exe“.
 Enter the RPi IP in „Host Name“.
 (how to find IP? – [How To Find IP Address of Raspberry Pi: A Beginner's Guide \(robu.in\)](#))

OR.... Enter the hostname we gave to the RPi earlier „raspberrypi.local“
 (this may or may not work, I definitely recommend finding an IP).

check the box „SSH“
 ... under „connection type“
 click button „Open“



If you are asked something about the certificate, confirm it.
Enter the „username“ and „password“ when ask, you defined in the earlier RPI5 OS customization.
If everything is ok, a terminal window will open.

Optional

upgrade the firmware to the latest version (if you don't want you can skip these steps)

Type this command in the terminal:

„sudo apt update“ - Hit „enter“

When its done, enter next command:

„sudo apt full-upgrade“ - Hit „enter“.

Wait for it to do something and it will probably ask you to confirm with

Y (YES) – Hit „enter“.

This will take a few minutes, when it's done reboot with the command:

„sudo reboot“ - Hit „enter“.

Wait for the system to reboot – connection will be lost -

So, reconnect to the RPi via SSH through Putty as you did before.

Check if the firmware is on the latest version, enter:

„sudo rpi-eeprom-update“ - Hit „enter“.

The result should be that the firmware on the RPi is equal to the upgrade firmware. If so, everything is ok now.

Optional - END

In case you did NOT update the firmware you continue here.....

Now let's set the PSU_MAX_CURRENT=5000 line in the firmware boot

Type the following command in the terminal:

„sudo rpi-eeprom-config --edit“ – Hit „enter“ (Note the double - - !!!)

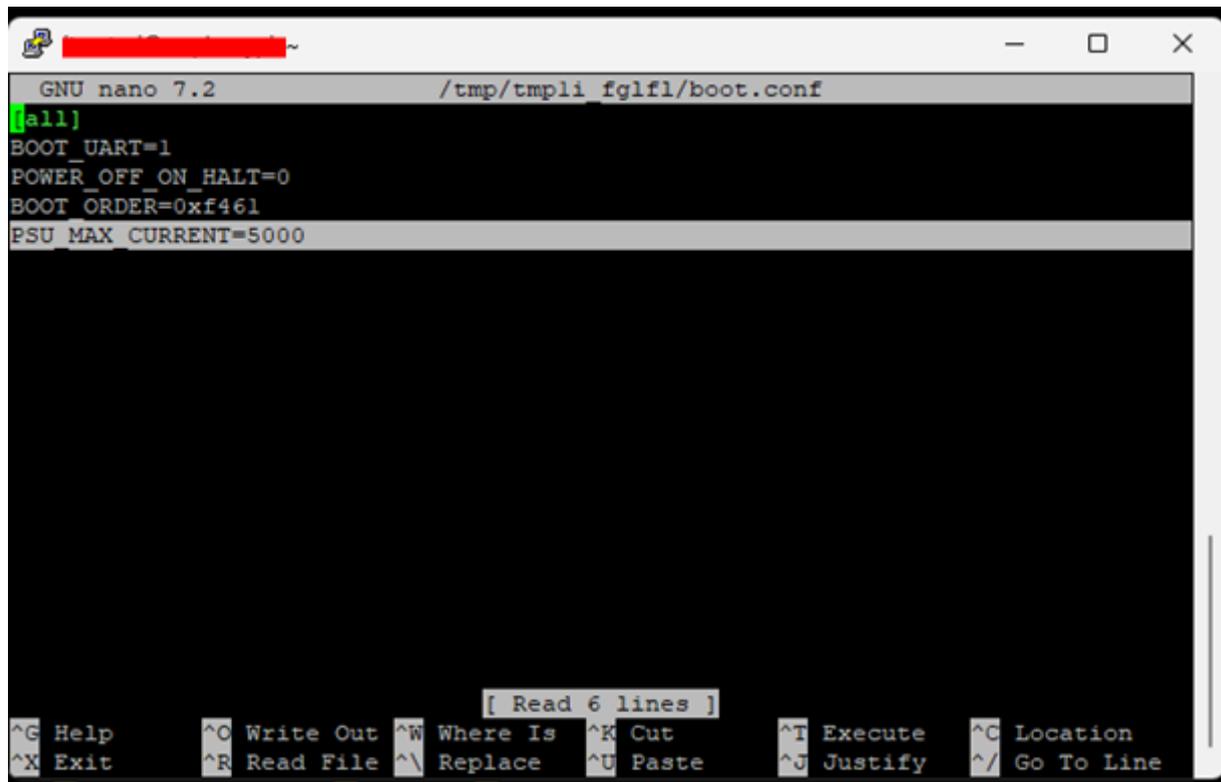
a small window like below in the image should open.

Use the arrows on the keyboard to move the cursor (here green block) to the end of the listing.

Type the following line:

„PSU_MAX_CURRENT=5000“

(as you can see in the picture)



```
GNU nano 7.2 /tmp/tmp1i fg1f1/boot.conf
[all]
BOOT_UART=1
POWER_OFF_ON_HALT=0
BOOT_ORDER=0xf461
PSU_MAX_CURRENT=5000

[ Read 6 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Then „Exit“ by hitting „**Ctrl+X**“ on your keyboard and confirm with „**Y**“ when ask to save the changes.

Hit „Enter“ one more time to confirm saving the config file

Reboot with the command „**sudo reboot**“ - Hit „enter“

After the Rpi has rebooted, reconnect with Putty on Rpi and check if `PSU_MAX_CURRENT=5000` is saved with the same procedure we now know: „**sudo rpi-eeprom-config --edit**“ – Hit „enter“

If the line remains as in my picture, then that's it – you're done !!!

You can reboot it again and test if PD is turned off and the RPI will always strat from a bench power supply..

If you want to undo what has been done, connect to the terminal as you have done so far and delete `PSU_MAX_CURRENT=5000`, save and restart. Then the USB-PD protocol will be enabled again.

Thanks for this tutorial goes to:

Mario Zablocki

Doede Douma – Audio Design blog.dddac.com