FabScanPi

Dec 31, 2020

Getting Started

0.1 About the Project 1
0.1.1 Contents 1

FabScan

Open Source 3D Laser Scanner

0.1 About the Project

FabScan is an open source 3D laser scanner. The project started in 2010 at Germany's first FabLab in Aachen. The FabScan PI is the next generation of the FabScan 3D Laser Scanner and since 2015 Mario Lukas took over as the lead developer of the project. The project is a community and spare time driven project.

Even if it looks like a team of many people is working in it, mostly only two or three people are working on the current branch (please thing about it while you are waiting for support).

The core of the project is based on a Raspberry Pi and a Raspberry Pi camera module. With those components the scanner can be driven standalone. It can be controlled over a web-enabled frontend.

0.1.1 Contents

First Steps

Build your FabScan

You can purchase the FabScanPi as a kit or in parts at Watterott Electronics

- FabScanPi Kit (contains all needed parts)
- FabScanPi HAT (a Raspberry Pi HAT for FabScan)
- FabScanPi Camera Mount (Raspberr‘y Pi cam mount with LED’s)
- FaBScanPi Case (laser cut wooden case parts)

But you can also build one by yourself. All needed files are located at GitHub.

Get the latest Software

The fastest way to use the FabScanPi is to use the FabScanPi Raspbian ISO image. Connect all needed hardware parts, download and flash the latest ISO image:

Open the User Interface

You have to unzip the file to get the .img file. After flashing the image to an SD card point your browser to

http://[ip-address-of-your-raspberry-pi]

It is also possible to join the web-enabled FabScanPi user interface by pointing your browser to

http://fabscanpi.local

Note: Calling fabscanpi.local requires zeroconf. This is povided by default on Apple devices. For Windows you will needto install the Bonjour printer driver to access your FabScanPi over thelocal address.

Bill of Materials

The following list shows what you need to build a FabScan.

General components

- Stepper Motor, NEMA 17 type, 1.8° step angle (200 steps/revolution)
- Pololu Universal Aluminum Mounting Hub for 5mm Shaft
- Motor driver (Silent Step Stick recommended)
- 5V red line laser module (second laser optional)
- FabScanPi HAT for Raspberry Pi
- 12V to 5V switching regulator
- Raspberry Pi 3 B+
- Raspberry Pi camera V2
- Raspberry Pi Camera Connection Cable - 50cm
- FabScanPi Camera Mount with LED ring
- FaBScanPi Case
- Screw Set or see the *Screws and Washers* table
- 1x 12V / 3A power supply

Even if you use the screw set from Watterott, as given in the *BOM*, here is the the usage reference:

Screws

Type	Length	Qty	Usage
M2	16	2	Security Switch
M2.5	15	4	RaspberryPi
M3	10	4	Stepper motor
		4	RPi-RingLight
		4	Turn-table Mounting Hub

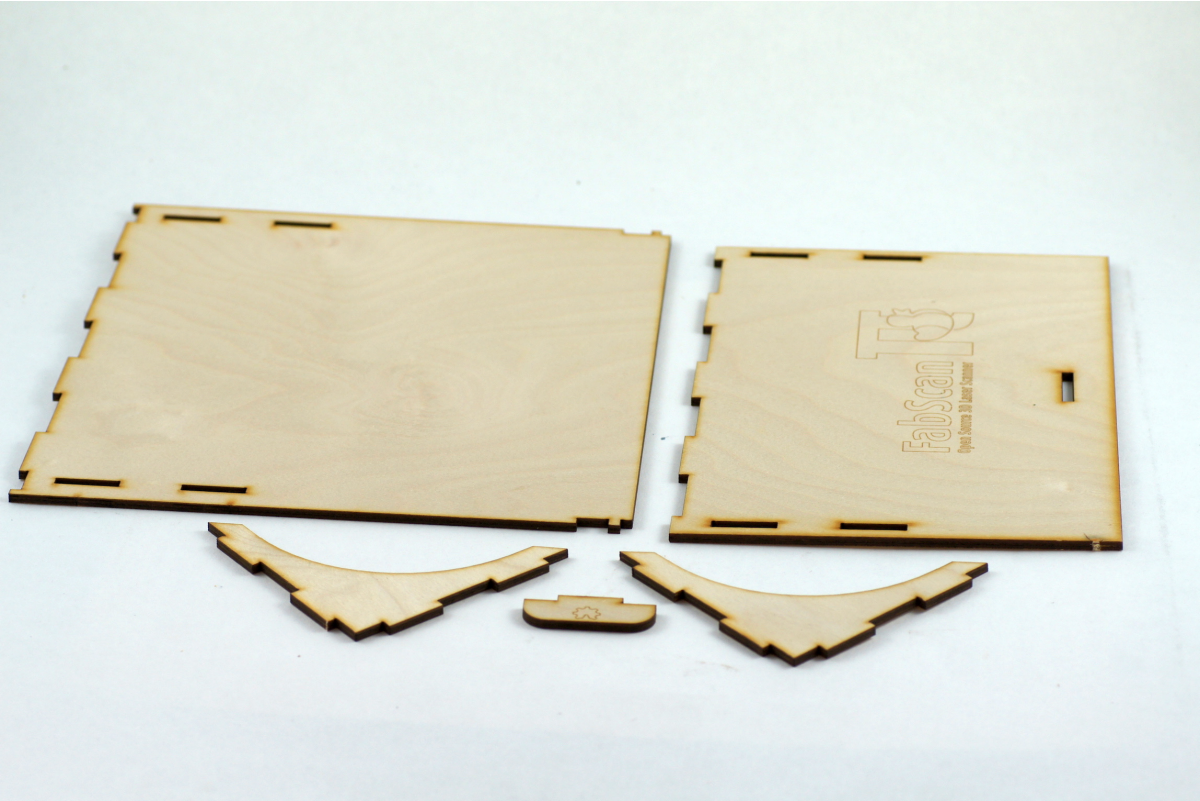
Washers

Type	Heigth	Qty	Usage
3.4/4.5	1	4	Stepper motor
2.5/4.5	3	4	RaspberryPi

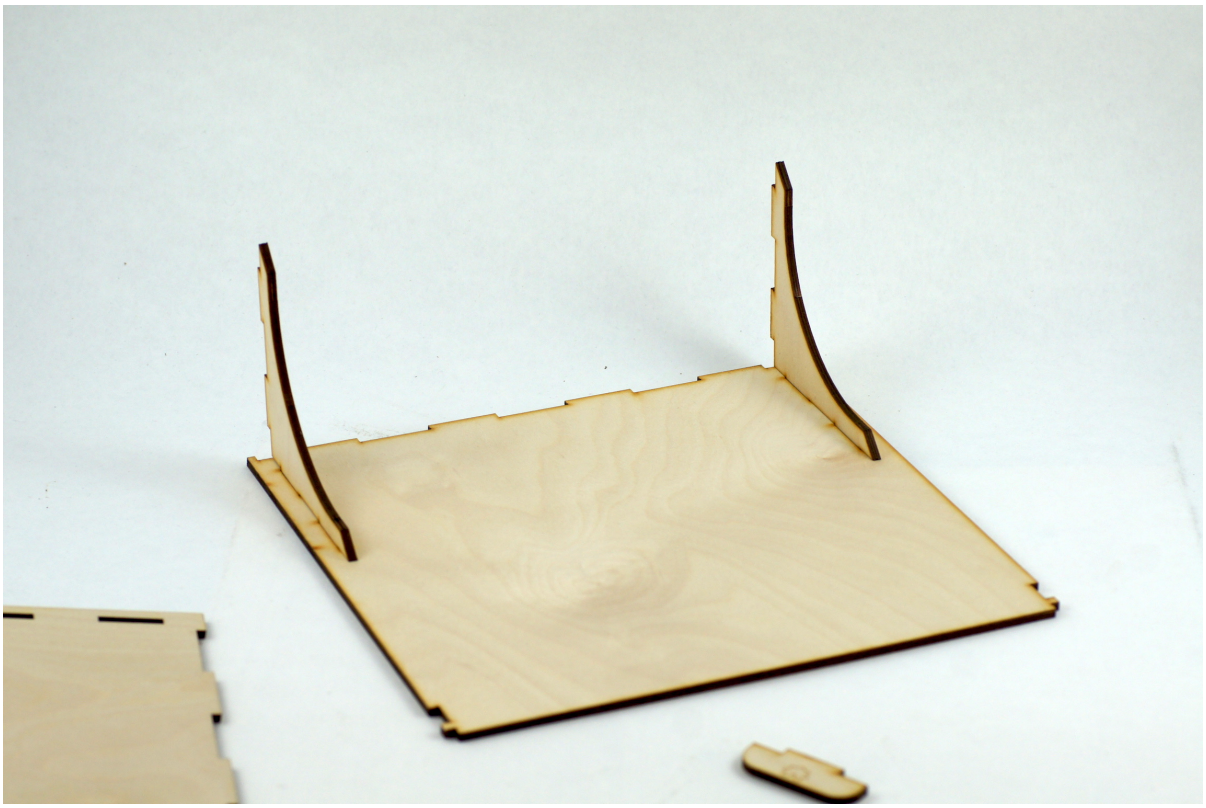
The Case

Lid Assembly

Step 1:



Step 2:



Step 3:

Step 4:

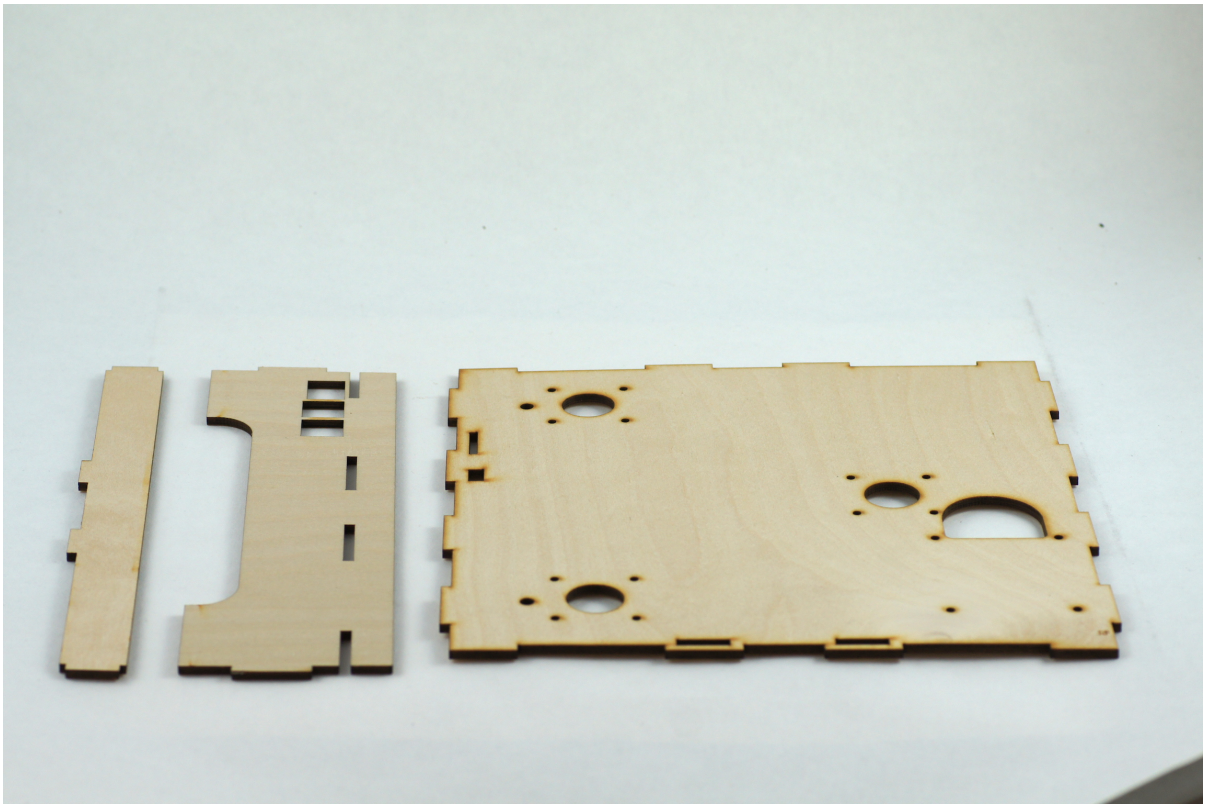
FabScan
Open Source 3D Laser Scanner

Step 5:

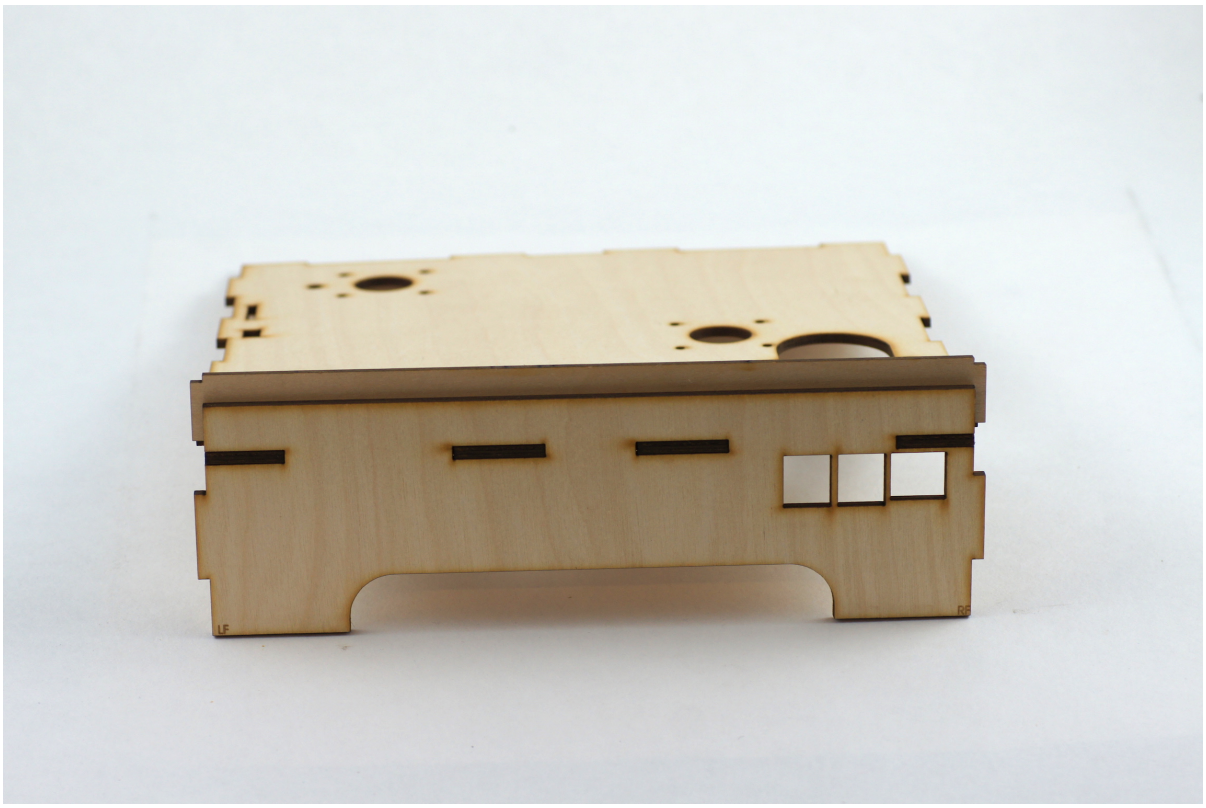
FabScan
Open Source 3D Laser Scanner

Cabinet Assembly

Step 6:



Step 7:



Step 8:

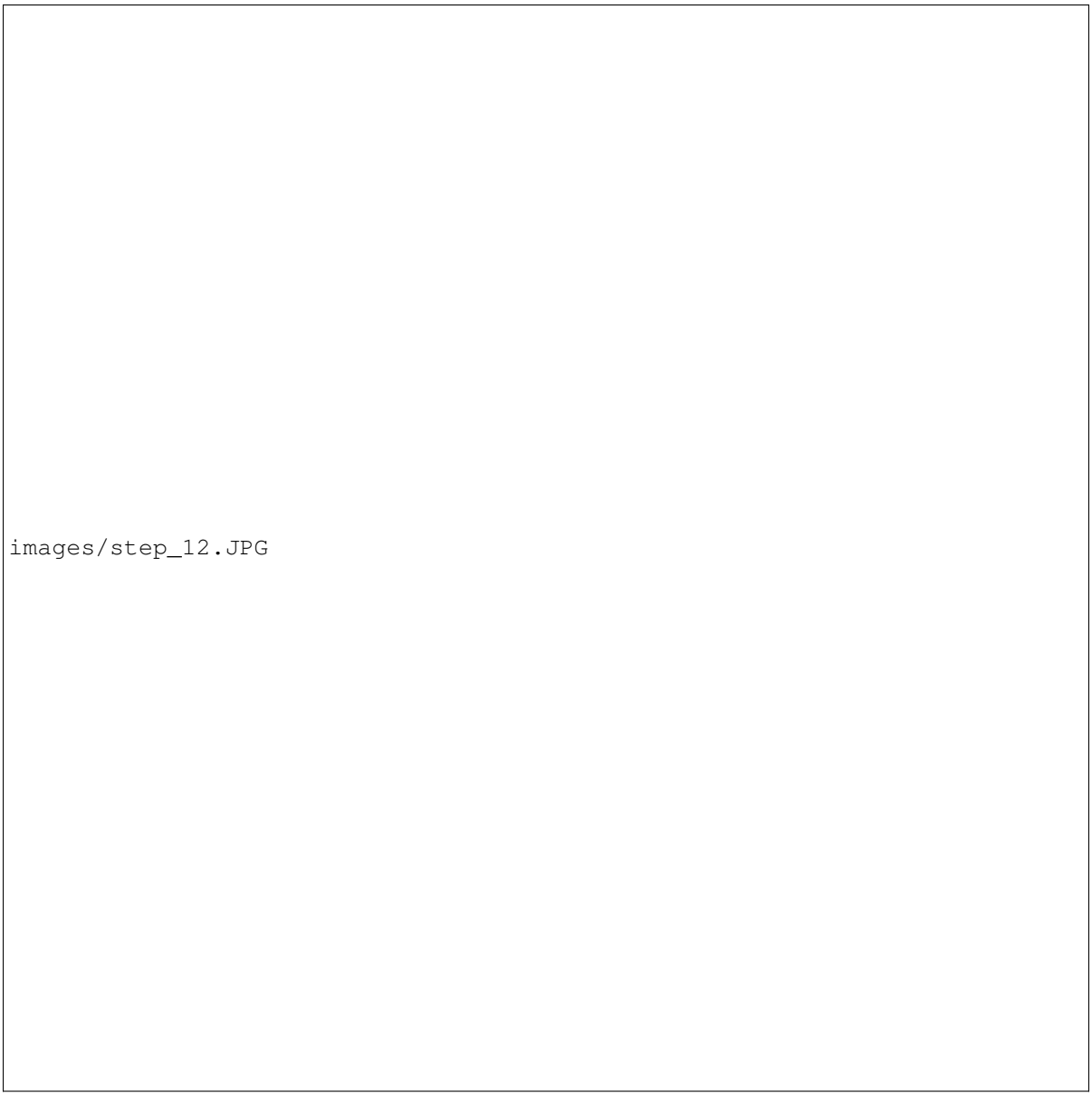
Step 9:

Step 10:

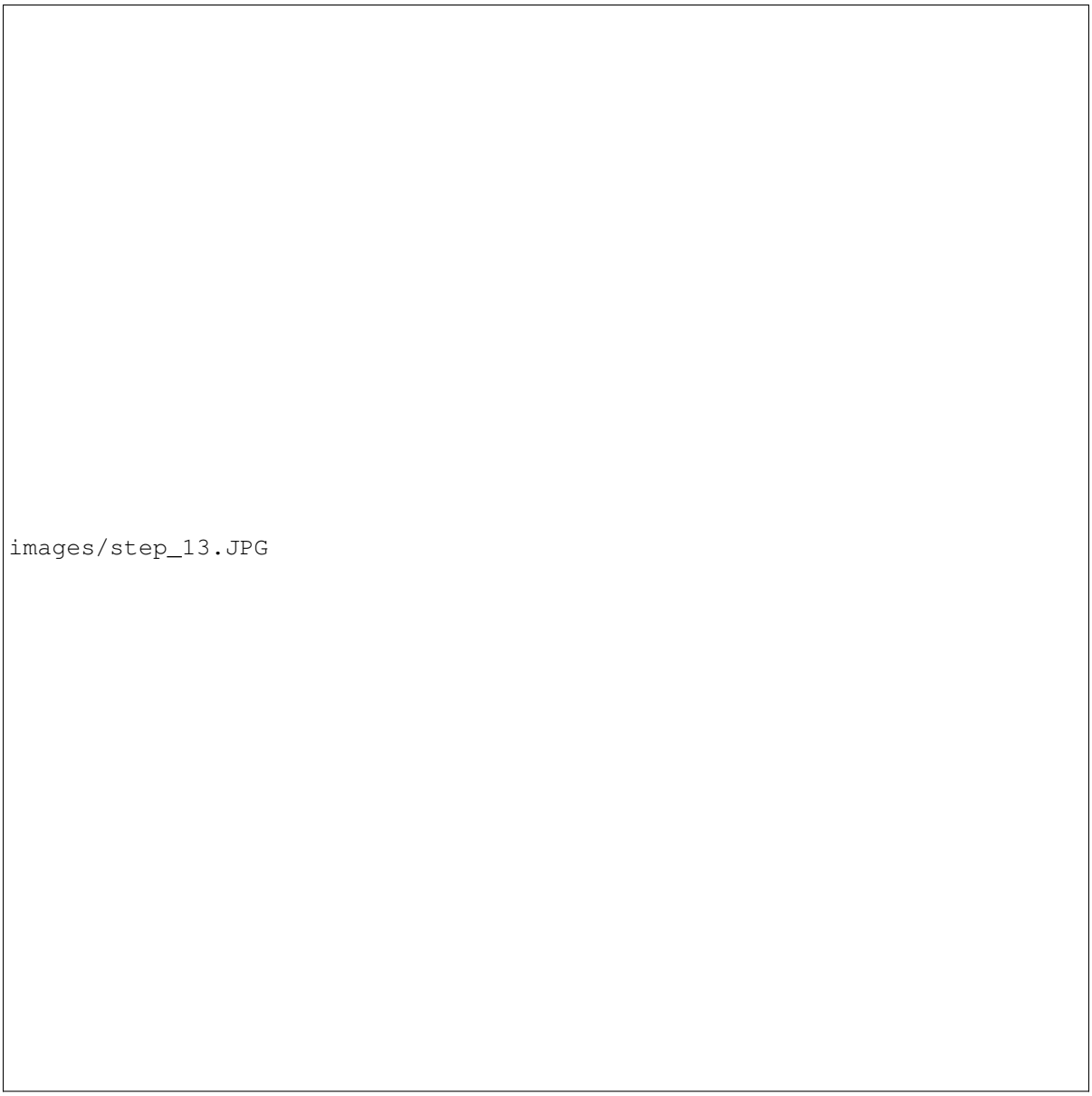
Step 11:

Laser Mount Assembly

Step 12:



Step 13:



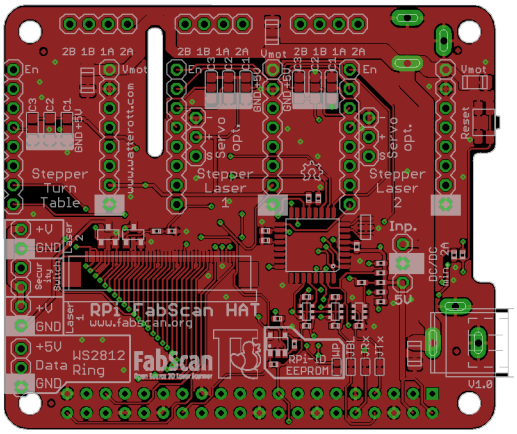
Step 14:

images/step_14.JPG

The Electronics

About the FabScanPi HAT

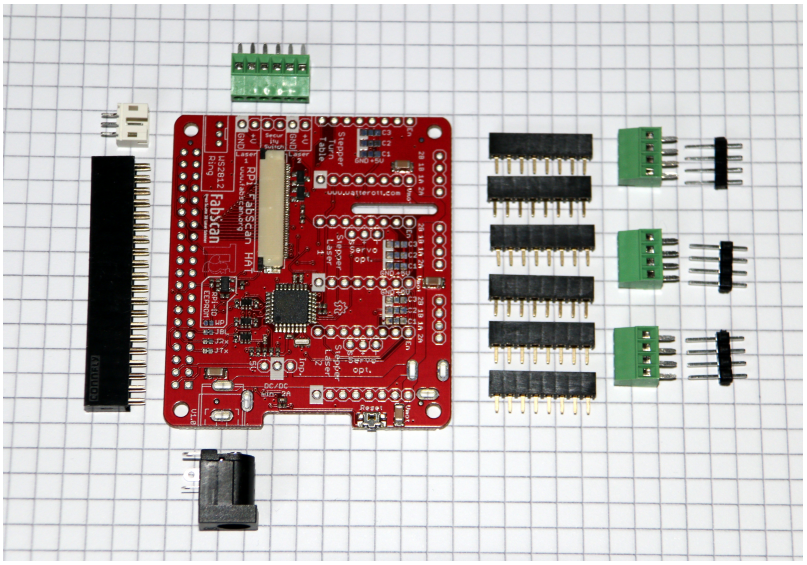
The FabScan HAT is basically a combination of an [Arduino](https://www.arduino.cc/) and the old [FabScan Shield](https://github.com/watterott/FabScan-Shield) for [Arduino](https://www.arduino.cc/). It provides all connectors for the hardware parts (like motors, servos, lasers, LED's etc.) Instead of an USB connection to the Raspberry Pi, the HAT is attached on the Pi's pinheaders. The HAT communicates over a serial connection with the Rasperry Pi. (GPIO14 and GPIO15 of the Raspberry Pi). The firmware and also updates are flashed automatically by the FabscanPi-Server application.



Assemble the FabScanPi HAT

Note: About soldering: If you are soldering for the very fist time, we suggest to read the comic [“Soldering is Easy”](https://mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf) by Mitch Altman (soldering wisdom), Andie Nordgren (comic adaption) and Jeff Keyzer (layout and editing)._

The FabScanPi HAT will be delivered with the main components already assembled. You only need to install the headers and connectors which fit your demand.



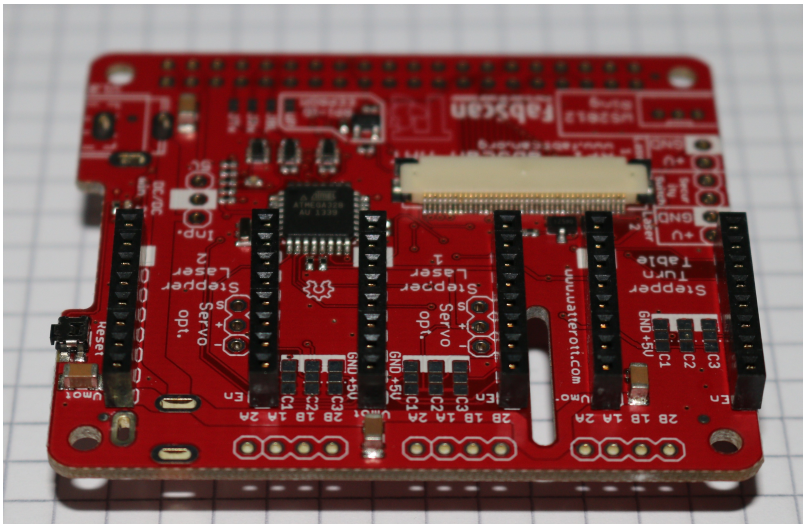
Let’s start with the female 8-pin headers for the stepper motor controllers. First the two headers which are designated with “Stepper Turn Table”. Put one in and flip the pcb to solder the pins on the rear side. Solder one pin first and make sure the header is still in the correct position. A “third hand” tool or a pcb-holder are very helpful for this job.

Now you can select between three options

- **Option A:** Two stepper motor ports for the laser adjustment or
- **Option B:** Two servo connector ports for the laser adjustment or
- **Option C:** Two stepper and two servo connector ports for the lasers (additional headers required)

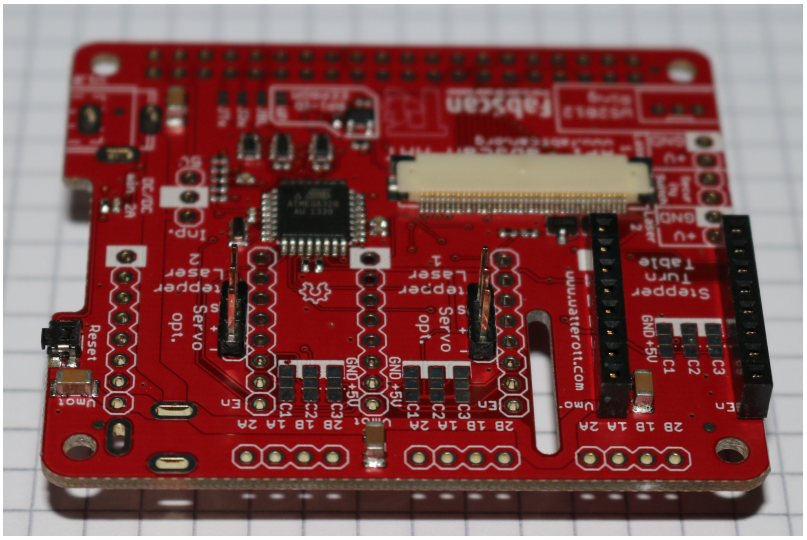
Option A:

Use the remaining four 8-pin femal pin headers and install them on the pcb. You can now use the HAT with three stepper motors (1x turn table, 2x laser control)



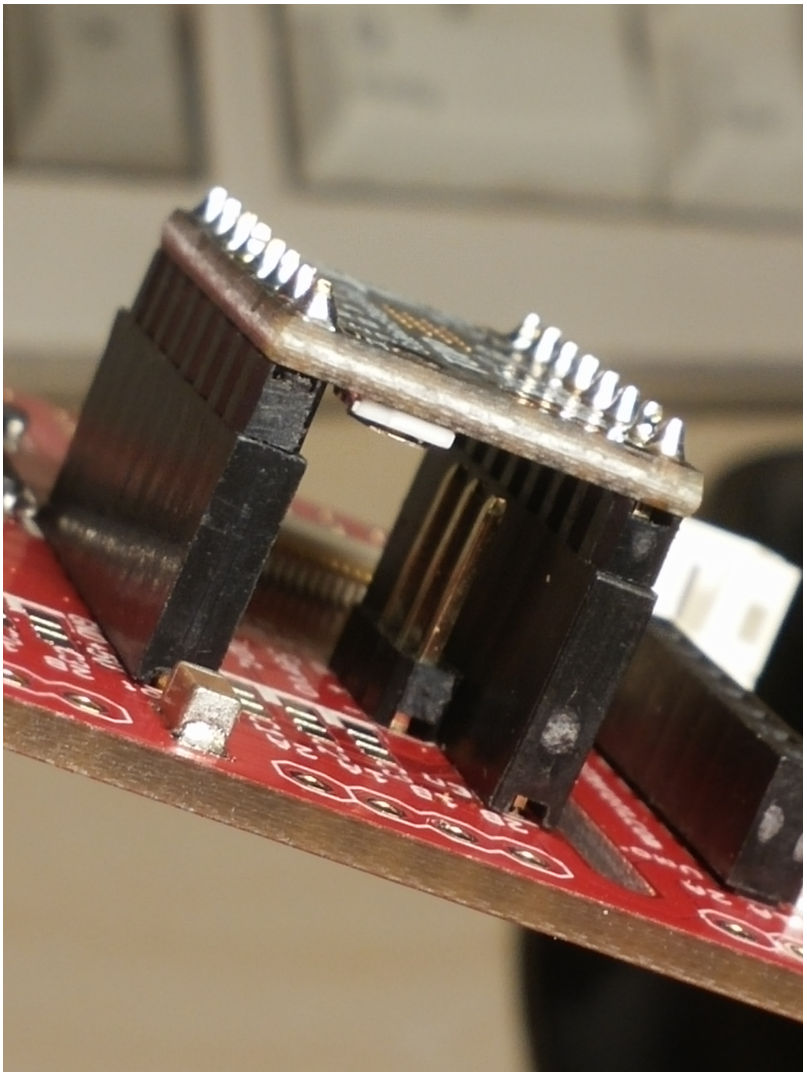
Option B:

Use an electronics wire cutter to cut off two 3-pin male headers from the long male-header. Put them into the pinholes labeled “Servo opt.” You can now use the HAT with one stepper motors (1x turn table) and two servos (2x servo opt.).



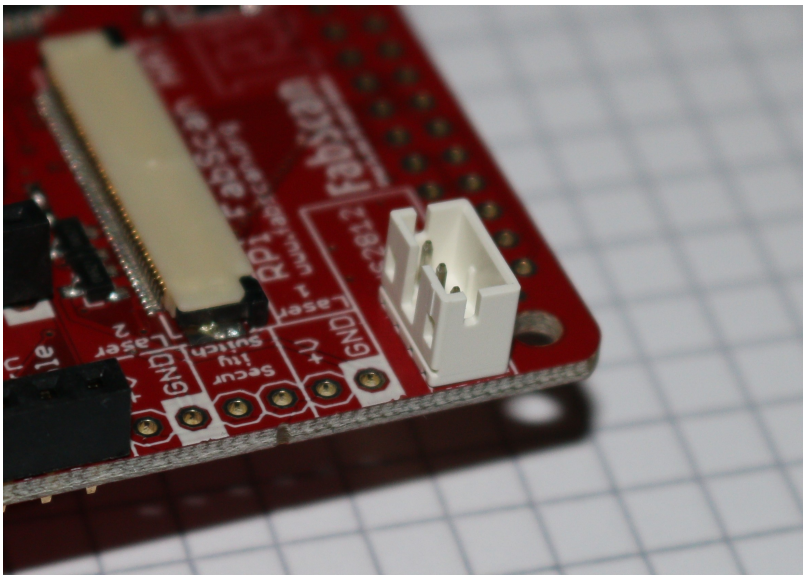
Option C:

If you are unsure about using servos or stepper motors you can install both. Therefore you’ll need to buy four 8-pin female headers which are at least 8 mm of height. Install them at the pinholes for the “Stepper Laser 1 and 2”. Also cut off two 3-pin male headers from the long male-header. Put them into the pinholes labeled “Servo opt.” Now you can use either servo controllers nor the connectors for the servos.

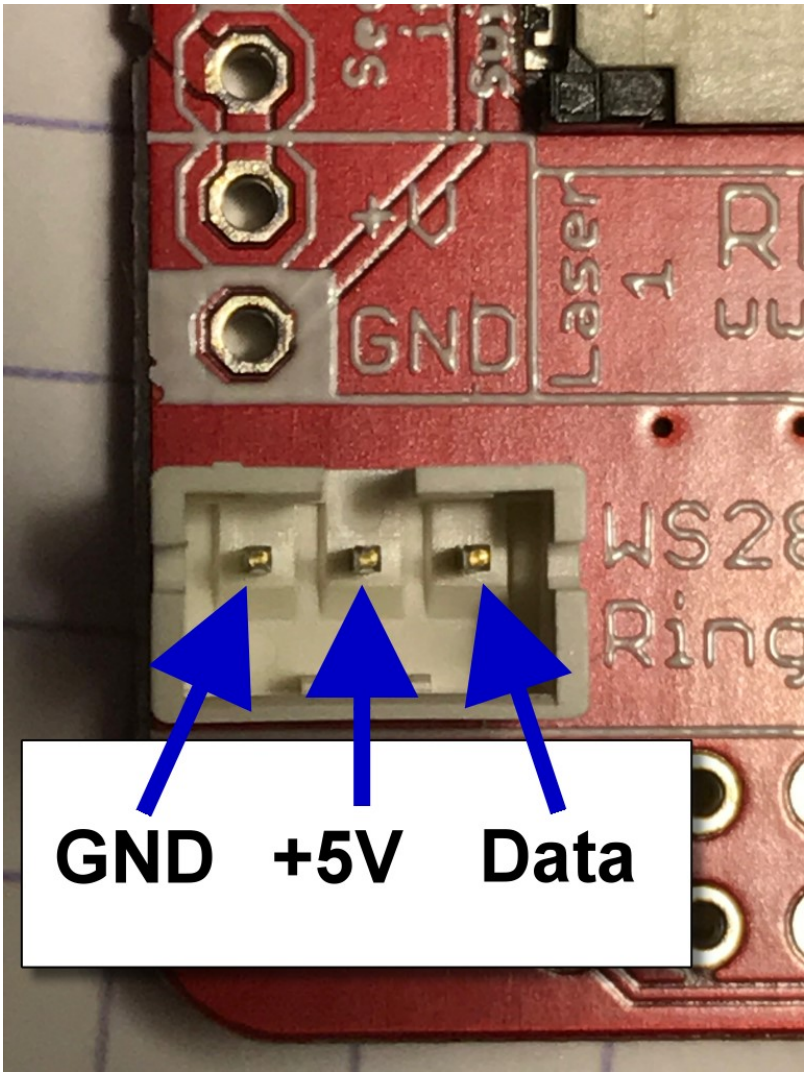


Connector for the LED-Ring

The next connector is the JST connector for the RPi-RingLight. Make sure the direction is correct. Fix it and solder the three pins on the rear side of the PCB.

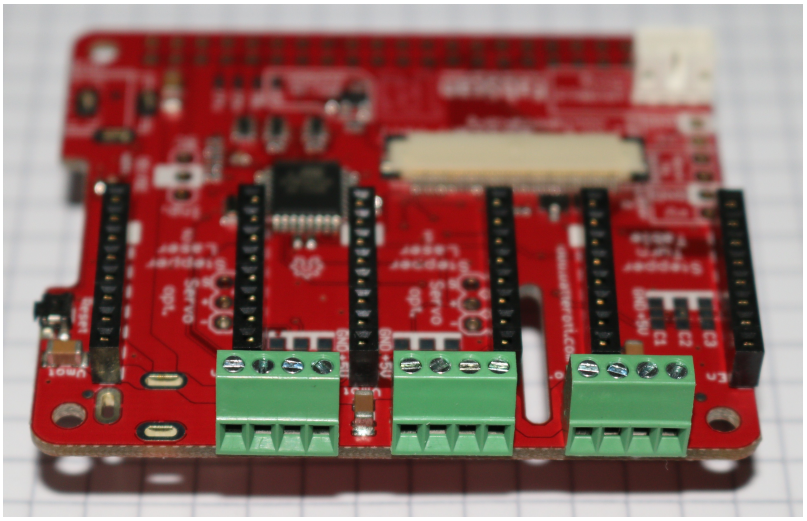


The pins of the JST connector on the HAT have the following configuration:

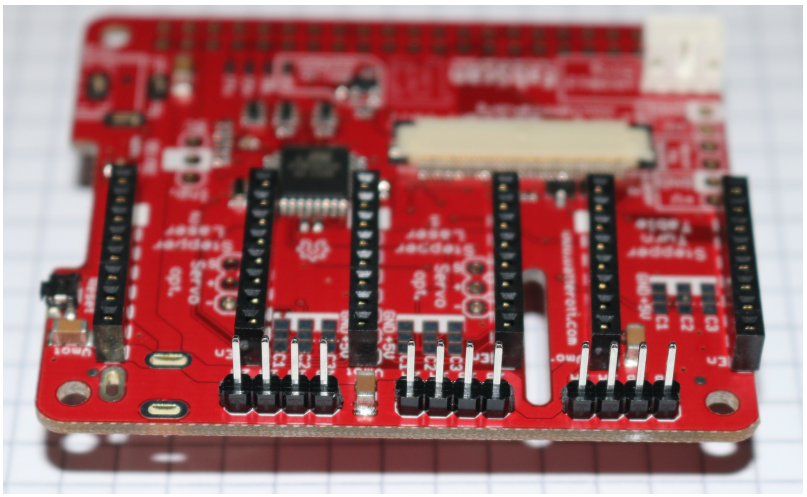


Connectors for the stepper motors

Here you can use the three 4-port screw terminals if your stepper cables have no connector plug.

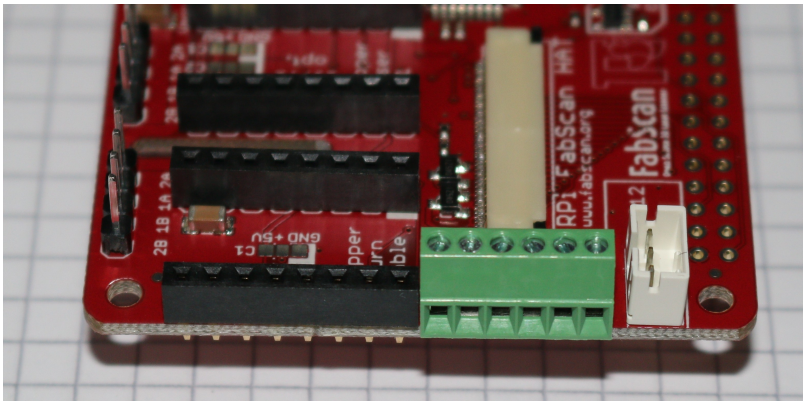


If the stepper motor cables have an 4-pin female connector you can use three 4-pin male headers (to be cutted off from the long header).



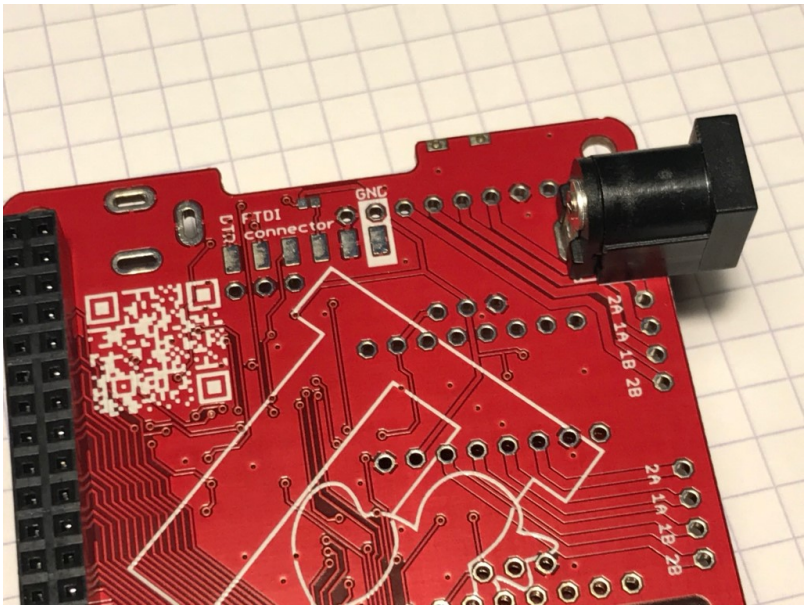
Connector for Laser(s) and Laser safety switch

The next connector to be installed is the 6-port screw terminal. It must be soldered into the corresponding pin holes between the JST connector and the 8-pin female header for the table stepper. Later this 6-port terminal will be used to connect the Laser(s) and the Laser safety switch.



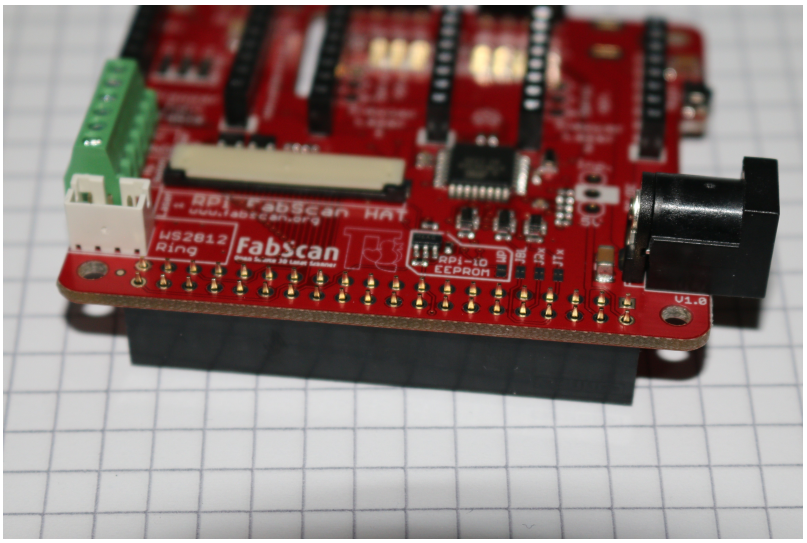
Power Connector (DC- jack)

Now the DC-jack power connector will be installed into three oval pin holes across from the 2x20-pin socket header. Soldering is done from the top side of the PCB.



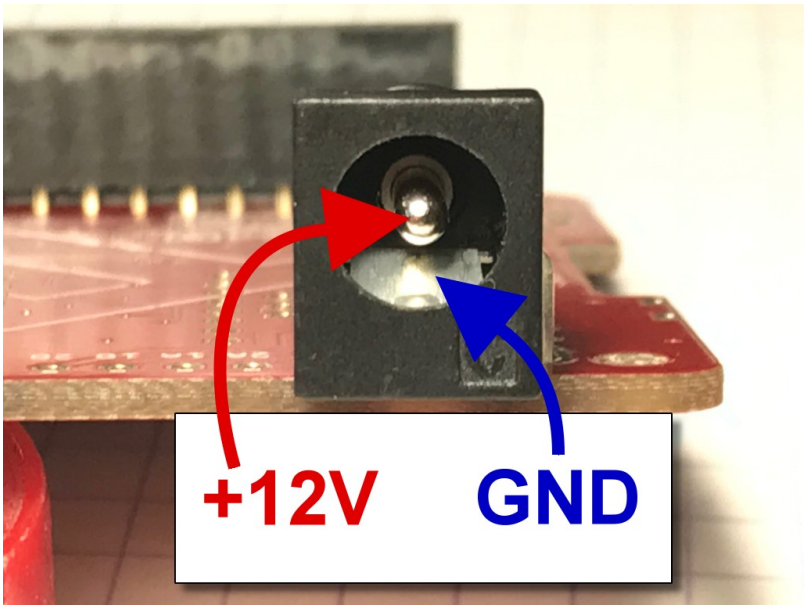
Assembling the Raspberry-Pi Connector

Now the 2x20-pin socket connector must be installed from the rear side of the PCB (where we only did soldering till now). Soldering of the 2x20-pin connector is done on the front side (where all our parts are placed). This connector is used as interface for the Raspberry-Pi.



Attention: The power connector in the image above image is soldered in an alternative way. The recommended way is shown in the second last image.

The center pin is for +12V DC, the outer connector tongue is for GND.



Mount the 5V DC/DC converter or 5V external Power Source

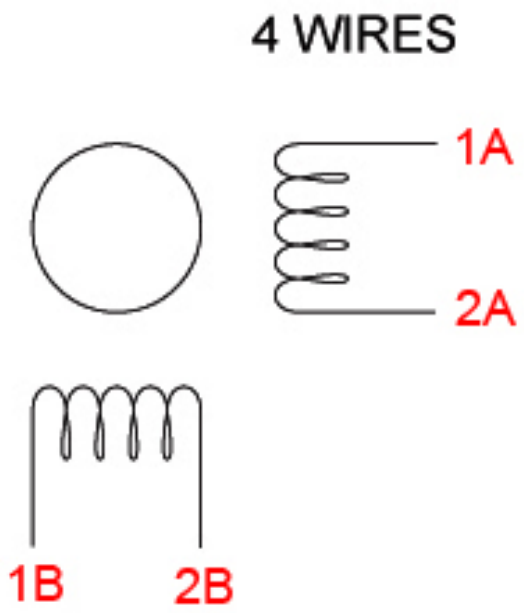
Additionally to the 12V DC coming from the power supply the FabScanPi needs 5V DC. Now you have to decide if you want to use a DC/DC converter IC which generates the 5V out of the 12V from the external 12 power supply (common solution).

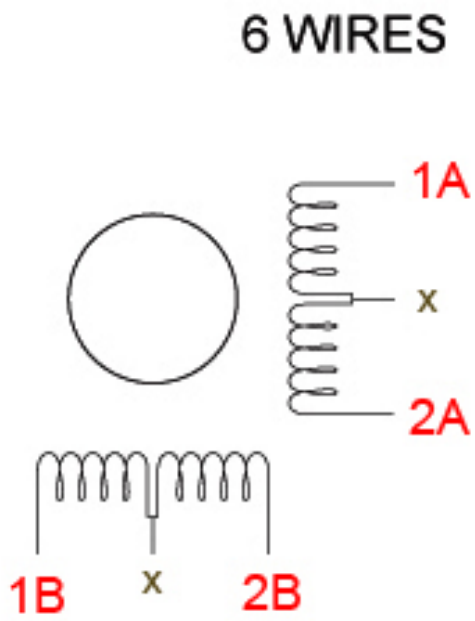
For the DC/DC converter option you can find further instructions and pictures on the [Watterott electronic website](https://github.com/watterott/RPi-FabScan-HAT/blob/master/hardware/RPi-FabScan-HAT_Assembly.pdf).

If you are using another power supply which is capable of delivering 12V and 5V DC you can connect it as well. Details can be found [here](#5V-EXT).

#Connecting the Stepper Motor(s)

There are different kinds of stepper motos. Mostly with 4 or 6 leads. For connecting the stepper motor to the FabScanPi HAT you need to know the corresponding lead pairs of the motor coils. The best way to find out something about the motor is to have a look at the datasheet of the motor manufacturer. In the following desciptions the pairs are called (2B, 1B) and (1A, 2A).





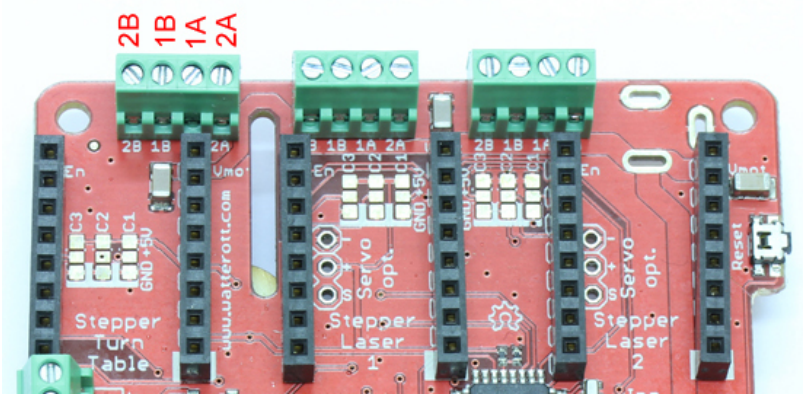
There are several ways to find the pair wires without a datasheet. Some of them are described here:

Method with an ohm-meter

Simply measure pairs of wires for their resistance. If the resistance is a few ohms (< 100 Ω) only, you've found a pair. The other two wires should make up the other pair.

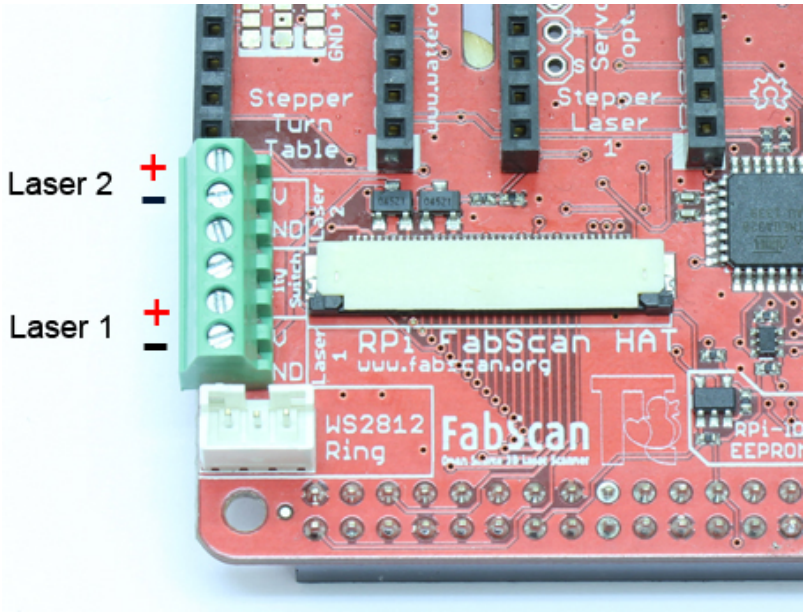
Methods without an ohm-meter

First, try turning the motor with your fingers, and notice how hard it is. Then, stick wires together in pairs. If the motor turns noticeable harder, you've found a pair. Another method is to use an LED, hold any two wires to the ends of a LED and turn the motor (twiddle in both directions), the LED will light if the wires are a pair, swap wires until you light the LED.



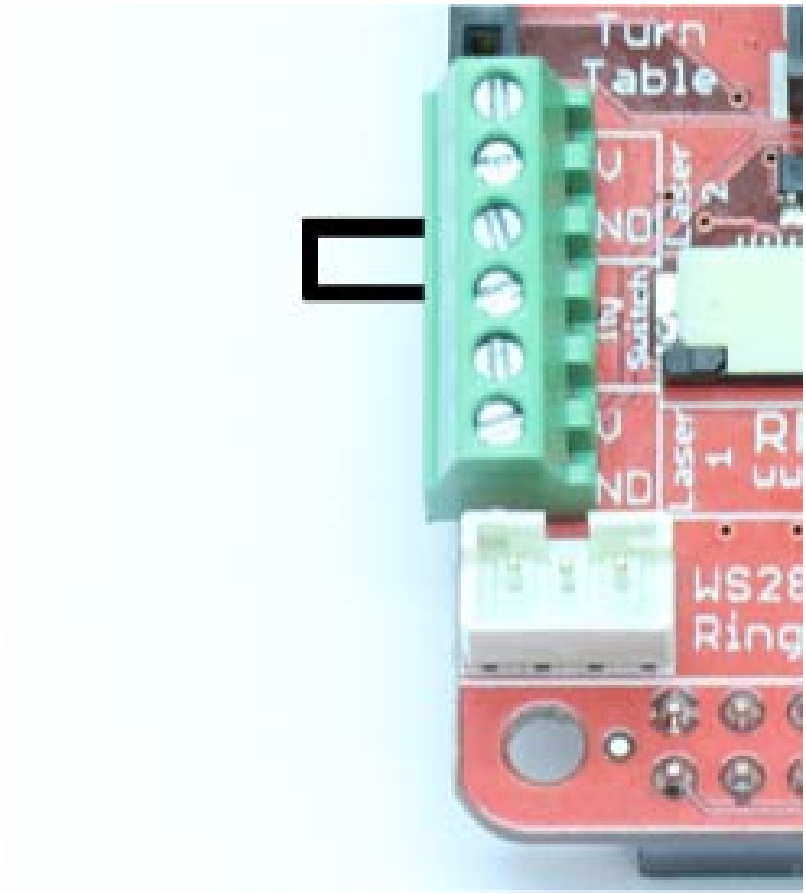
#Connecting the Lasers

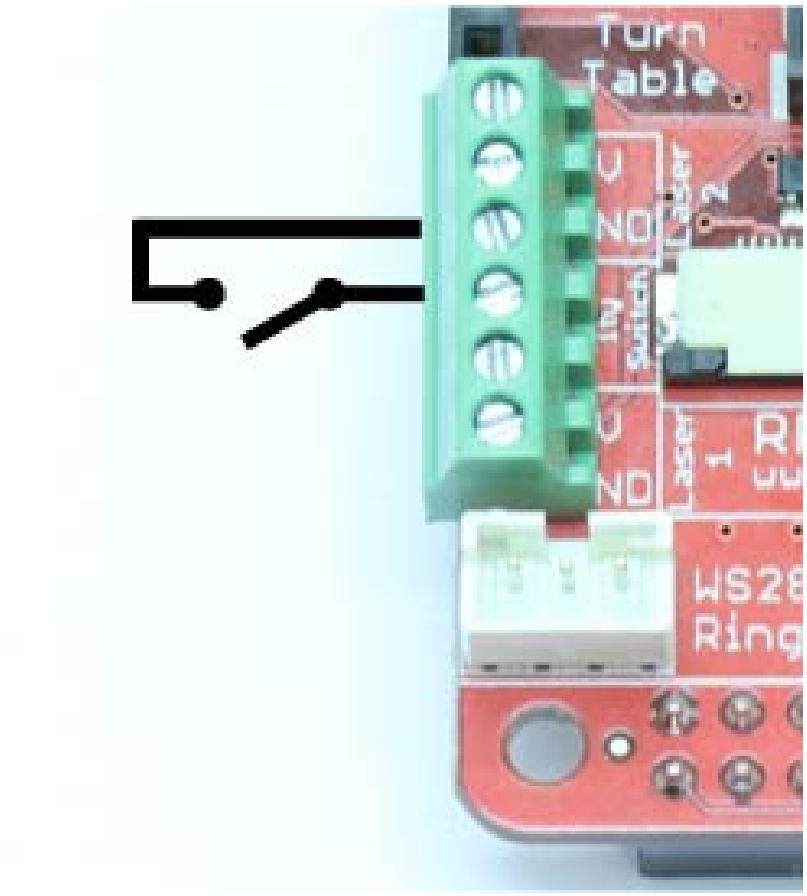
The FabScanPi HAT provides connectors for two lasers. But only one laser is supported until now. Connect your laser to the connectors labeled with . . .



Safety switch

There is the possibility to add a laser safety switch which disables the laser when the lid is opened. The FabScanPi HAT provides a connector for such a switch. If you don't need a switch you still have to bridge this connector with a cable to get the lasers work. (left image: with bridged connector, right image: connecting a switch)





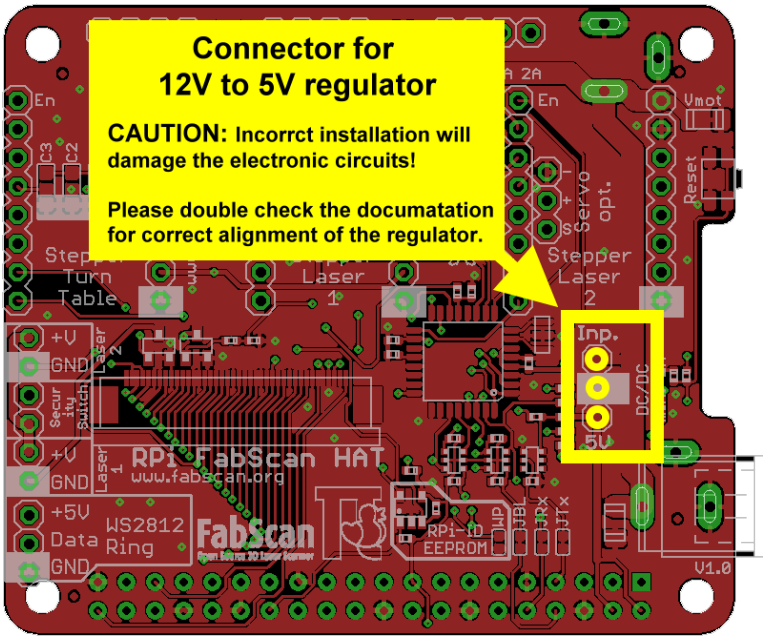
#Connecting the Motor drivers

TODO

#Connecting the Power Supply The FabScanPi will need 12V DC and 5V DC. There are different options to fulfil this requirement:

Option A: (recommended) Connect 12V DC to the HAT (round connector) and install a 12V DC - to - 5V DC regulator on the designated space on the HAT. Make sure the 5V power regulator can provide a current of min. 2.0A

for 1 turntable stepper motor, 1 PiCam and 1 laser.



Silk Label	Description
Inp.	12V from HAT as input for regulator
(white square) 5V	Ground (GND) 5V output from regulator to FabScanPi HAT

For the DC/DC converter option you can find further instructions and pictures on the [Watterott electronic website](https://learn.watterott.com/fabscan/hat_assembly.pdf).

NOTE: For the final stage of constuction a more powerful power supply is needed. At the moment there are no technical specifications available.

Option B:

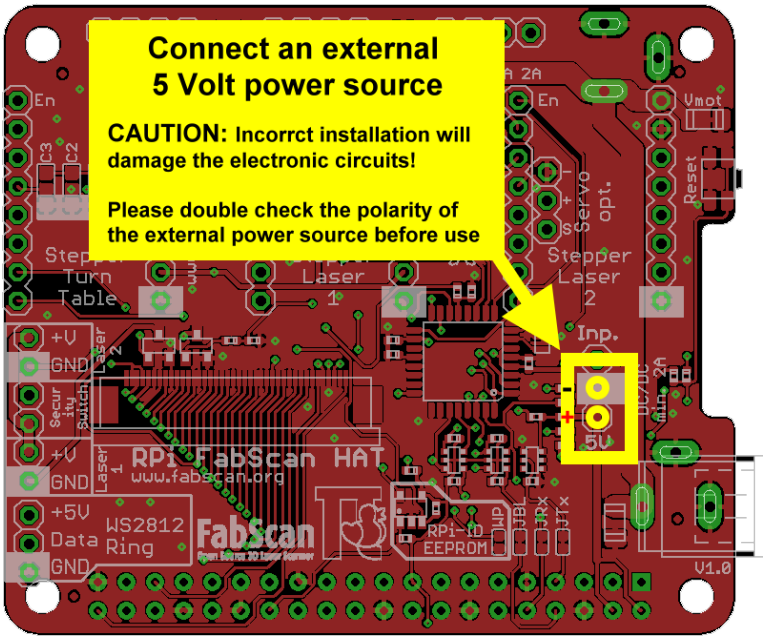
Connect 12V DC to the HAT (round connector) and 5V DC to the raspberry (micro USB connector).

NOTE: Make sure you switch on the both power sources at the same time to avoid software trouble.

Option C:

Connect 12V DC to the HAT (round connector) and connect a 5V DC power source to the 5V pin regulator pin on the HAT. Make sure the 5V power source can provide a current of min. 2.0A for

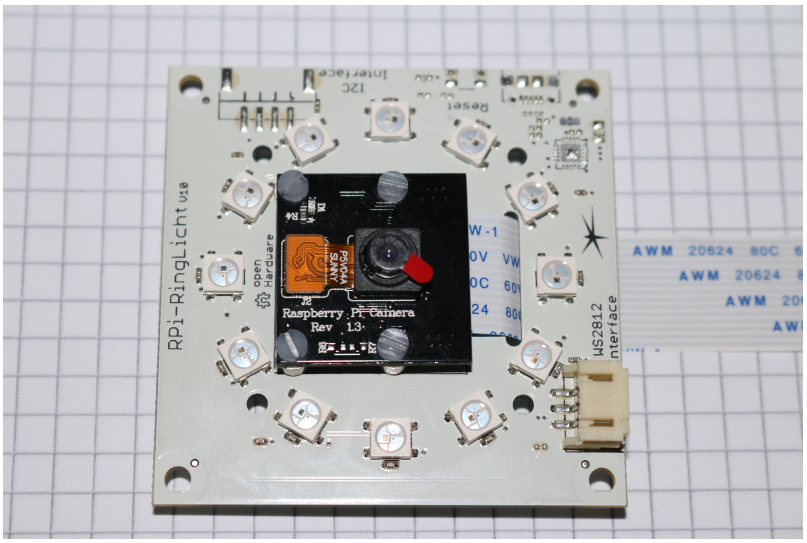
1 turntable stepper motor, 1 PiCam and 1 laser.



Pin label	Description
(white square)	Ground (GND)
5V	Output from external 5V power source

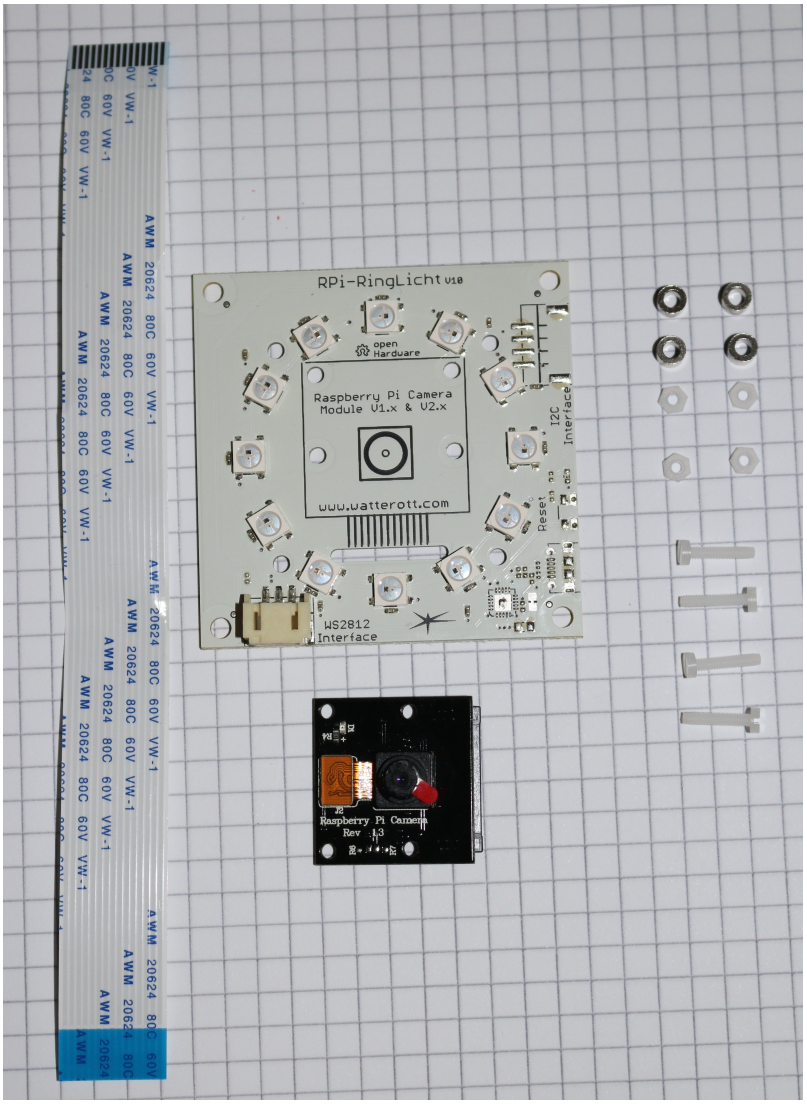
NOTE: For the final stage of constuction a more powerful power supply is needed. At the moment there are no technical specifications available. Make sure you switch on the both power sources at the same time to avoid software trouble.

#Connecting the LED Ring This small manual will help you to assemble the Camera and LED ring combination. You will need a light if you want to perform texture scans (Check mark is set for Color Scan).



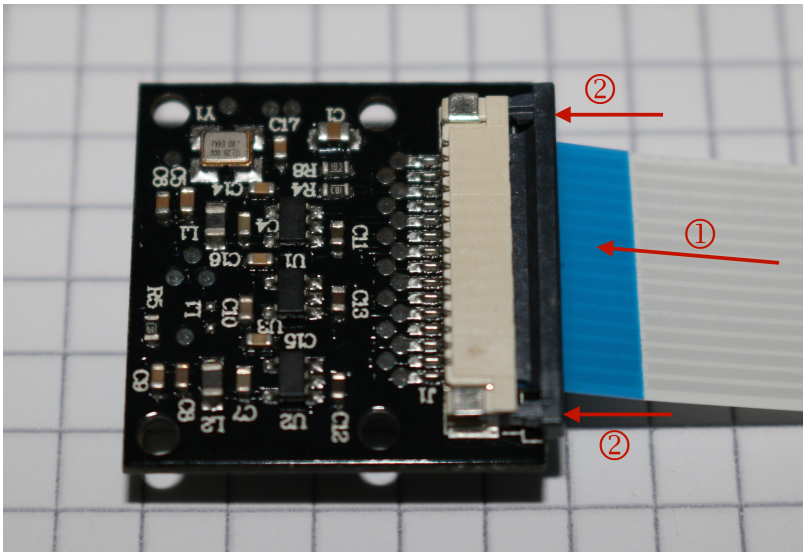
This is how it should look like if you finished the assembly.

The Ring-Light will include the parts you can see in the next picture below (the camera modul is sold separately):

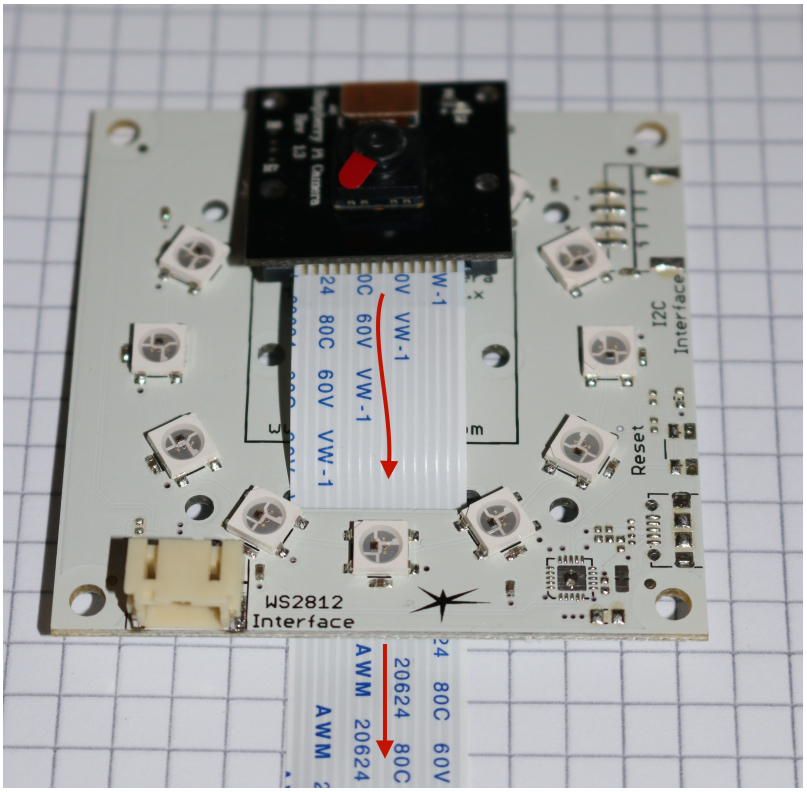


IMPORTANT: You will need the standard camera modul (which has a green pcb). The module used in the pictures is the IR version which has the same dimensions but different optical specifications.

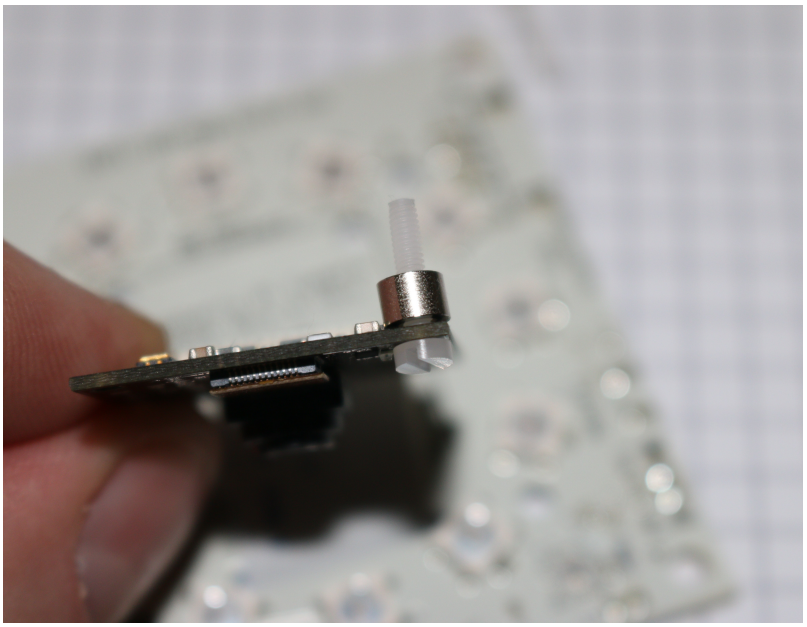
Step 1: Slide one end of the flat ribbon cable into the connector of the camera module (1). Make sure the blue plastic is on the side facing away from the camera. Push the lock bar of the connector gently into the lock position (2).



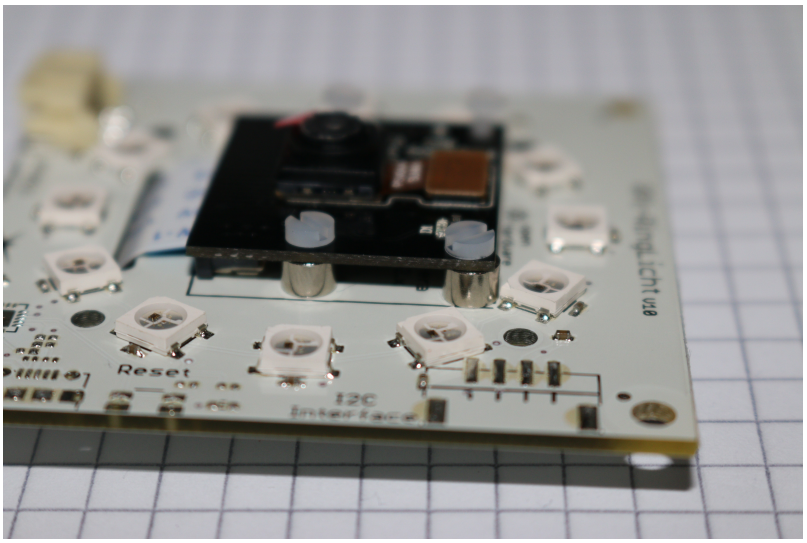
Step 2: Merge the flat ribbon cable through the elongated hole in the LED board.



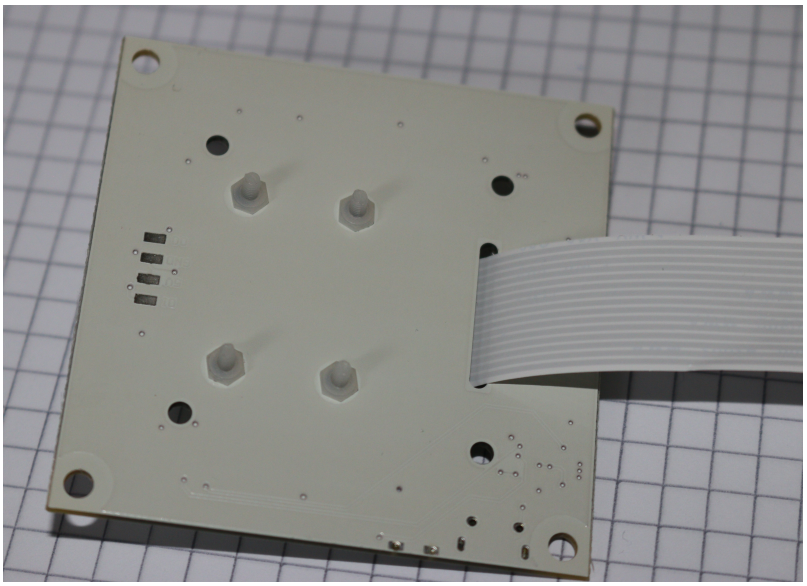
Step 3: Put the four Screws through the holes in the camera board. The heads must be on the side where the lens is situated. Then put a spacer on every screw (as shown in the next picture).



Step 4: Put the Camera modul on the LED board. All four screws must fit the appropriate hole. This can be a bit tricky but it is doable.

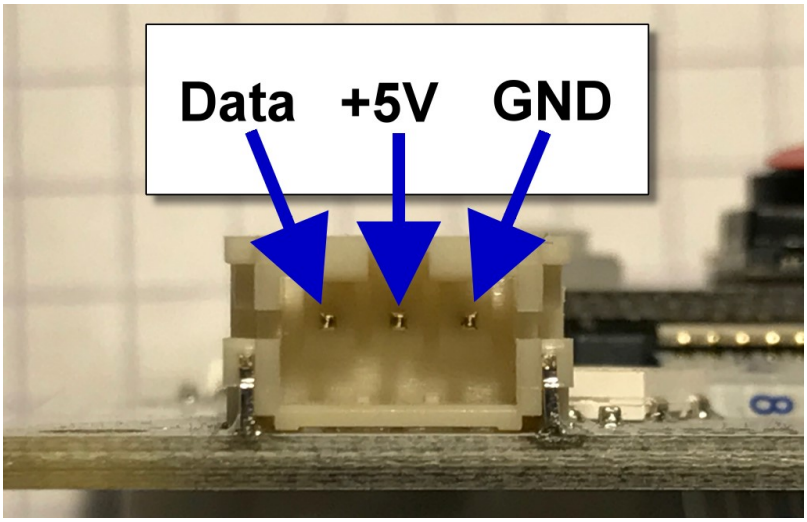


Step 5: Put a nut on each screw an pull them hand-tight.



Now the Camera-LED assembly is ready for installation.

The pins of the JST connector on the LED board have the following configuration:



#Additional Motors for the Lasers

TODO

Installing the Software

There ist no need to flash the Arduino which is located on the FabScan PI HAT. It will be flashed automatically with the current firmware version after the server is started.

Using the Image (recommended)

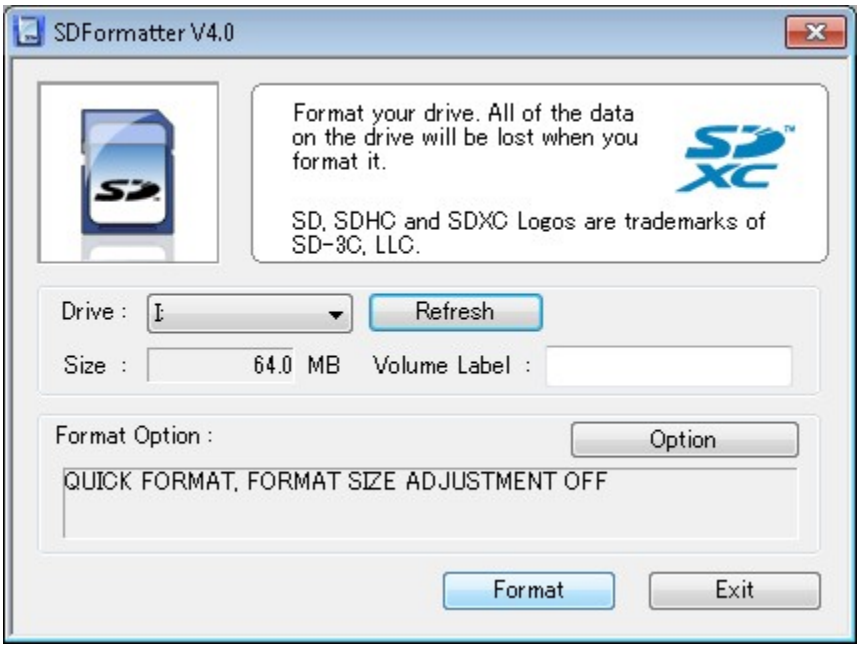
The fastest way to start working with FabScan PI is to use the FabScan PI Raspbian Image.

For the installation you will need the following things:

- A computer with integrated or connected card reader
- A Micro-SD card with at least capacity of 8 GB
- A software to format the SD card (e.g. [SD-Formatter](https://www.sdcard.org/downloads/formatter_4/))
- A software to install the image on the SD card (e.g. [Win32DiskImager](https://sourceforge.net/projects/win32diskimager/))
- The latest [FabScan PI Raspbian image](https://github.com/mariolukas/FabScanPi-Build-Raspbian/releases/latest)

Download the SD-Formatter an the Win32Disk image software and install them on your computer. Tip: During the installation takes place you can already download and unzip the latest FabScan PI image and save some time. In the end you should have a file with .img extension.

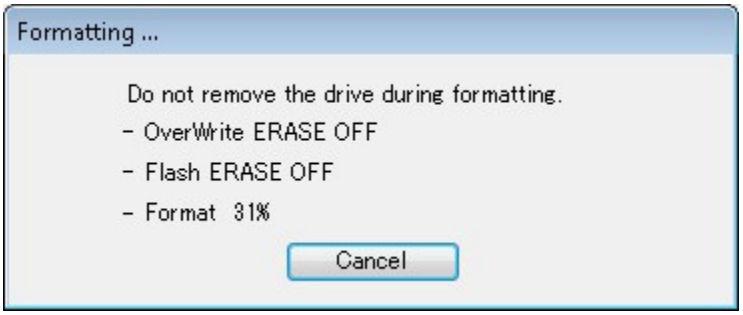
Now insert the Micro-SD card into the card reader which is connected with your computer.



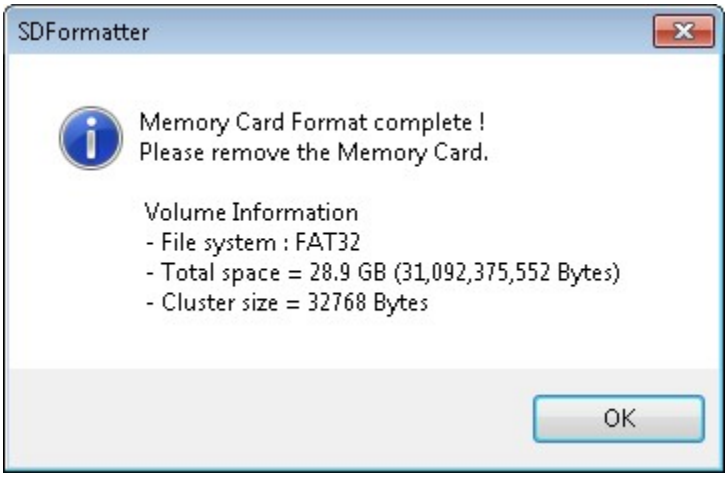
Start the SD-Formatter software and select the correct device letter. Double-check that because otherwise there is a risk of formatting another drive.

Note: The displayed size of the selected card my vary from the physical size. This is because of an old image which is already installed on the card.

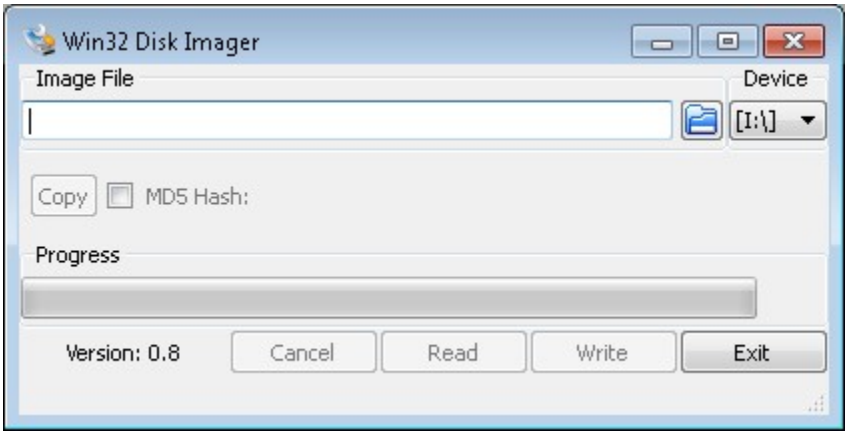
Click on the "Format" button to format the selected SD card.



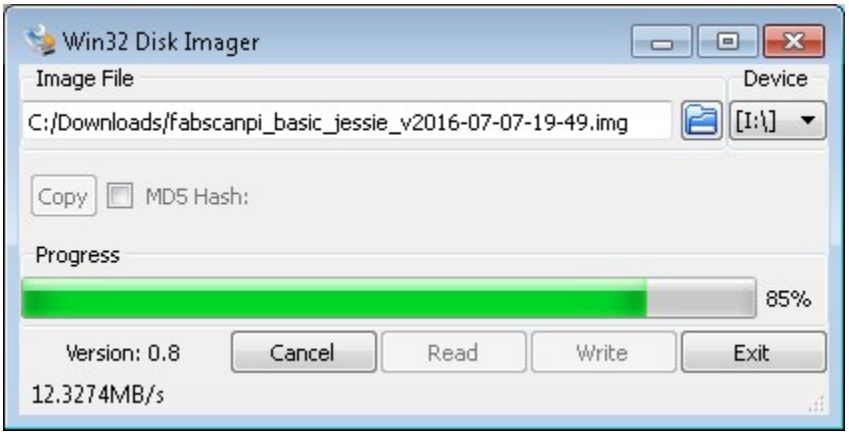
When the formatting process is completed an information window will pop-up. Leave the card in the reader.



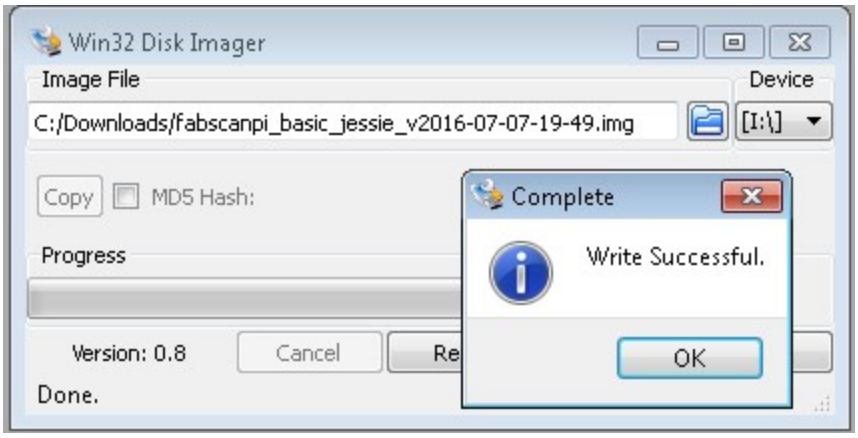
Exit the SD-Formatter and start the Win32DiskImager for transferring the image on the freshly formatted card.



Select same device as before in the SD-Formatter software. Click on the folder icon and select the image file in your file system. Normally it should be in your browser's download folder. Make sure to unzip it first to get the image with .img extension.



Click on the "Write" button and the installation process will begin to start. When it's finished you will be informed by a pop-up. Click on the "exit" button to close Win32DiskImager.



Now your SD-Card is ready to be put into the card slot of your FabScanPi.

After the image is flashed and the Raspberry Pi is up and running follow the instructions in the [Usage section](https://github.com/mariolukas/FabScanPi-Server/blob/master/README.md#useage)

Installing from deb packages

This description assumes that you have a SD card with a fresh Raspbian image on it.

First add the fabscan repository to your source list.

```
echo "deb http://archive.fabscan.org/ jessie main" >> /etc/apt/sources.list
```

Then add the FabScan PI repository key to your key chain.

```
wget http://archive.fabscan.org/fabscan.public.key -O - | sudo apt-key add -
```

Update the package list.

```
apt-get update
```

Finish the installation with the needed packages.

```
apt-get install fabscanpi-server python-opencv-tbb libtbb2  python-pil python-serial␣
↪python-pykka python-picamera avrdude python-semver python-scipy
```

The FabScan PI server can be started with

```
sudo /etc/init.d/fabscanpi-server start
```

Read [Usage](https://github.com/mariolukas/FabScanPi-Server/blob/master/README.md#useage) section for the next steps.

Building a custom image

The image can be build with the FabScanPi Image build script. You will find more

information [here](developing.md#Building FabScanPi Images)

Installing from Source

Dependencies

FabScan PI software depends on some python libraries. You need to install pyserial, pykka, opencv with tbb support and picamera. The easiest way to install all dependencies is to use debians package manager apt. Some of the packages, like opencv with tbb support and libtbb are not provided by the official raspbian mirrors. You need to add the fabscan repository to your apt source list.

Build Debian package Install dependencies

```
sudo apt-get install build-essential dpkg-dev debhelper devscripts fakeroot cdbs␣
↪python-setuptools python-support
```

The package is build by calling

```
make deb
```

Afterwards the package can be installed by

```
dpkg -i fabscabpi-server<package-version>.deb
```

Updating the Software

Updates can be installed with debian's apt-get package manager. Log in via ssh and check/install updates with the following command.

```
sudo apt-get update && sudo apt-get dist-upgrade
```

Configuration

Setting up a WIFI connection

This is the description for enabling wifi on the raspberry pi. Just create a file with the name wpa_supplicant.conf on the /boot folder. You can use nano and do it via a ssh connection to the raspberry pi, or remove the sd-card and use an sd-card reader on your computer. The /boot device will be mounted to your operating system.

Add the following content to the file

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=DE
network={
     ssid="<your_wifi_ssid>"
     psk="<your_password"
     key_mgmt=WPA-PSK
 }
```

Insert the sd-card if you have removed it. Do a reboot of the system (for both options needed).

Scanner Configuration File

How to Edit the Config File

Remote access to change the configuration settings

It is most likely that you don't have a monitor nor mouse and keyboard connected to your FabScanPi all the time. But maybe you need to make some changes to the config file from time to time and you don't want to connect the peripherals everytime.

This can be done via a remote PC which is connected to the same network.

Connecting as Windows User (Putty)

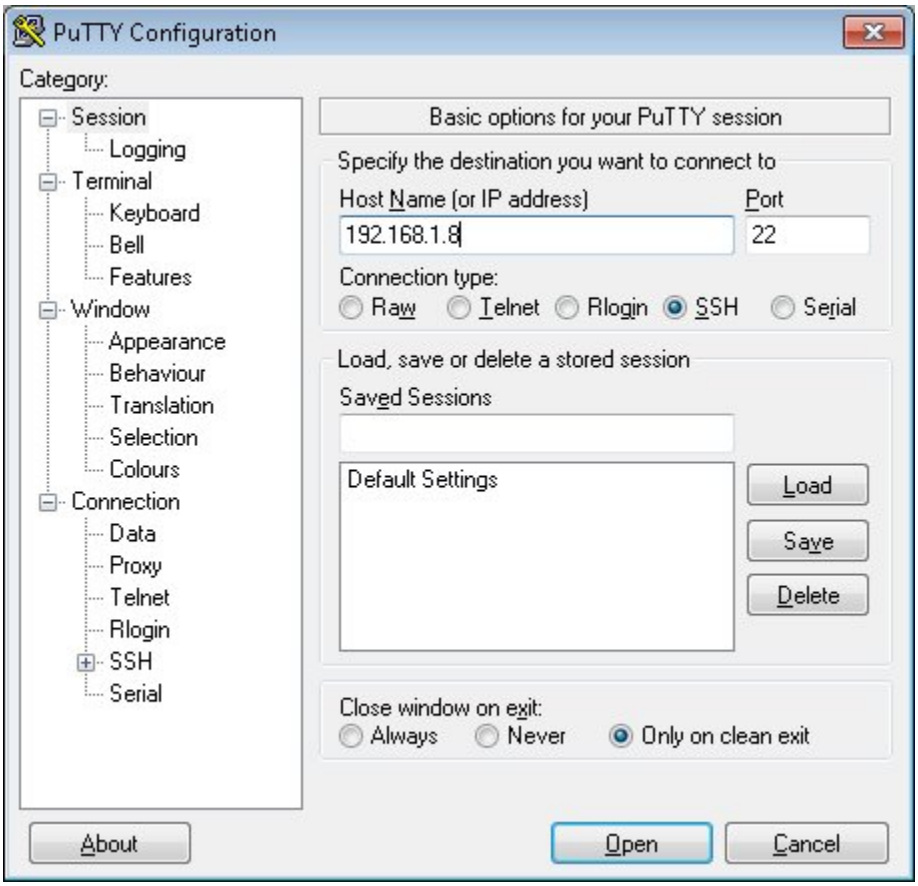
First of all you need to download the tiny program "PuTTY" from http://www.putty.org/.

PuTTY is a SSH client program which establishes the connection to your FabScanPi. There is no no graphic user interface -only a console which allows only the exchange of text. But that's enough to make some changes in the config file or to update your FabScanPi-Software.

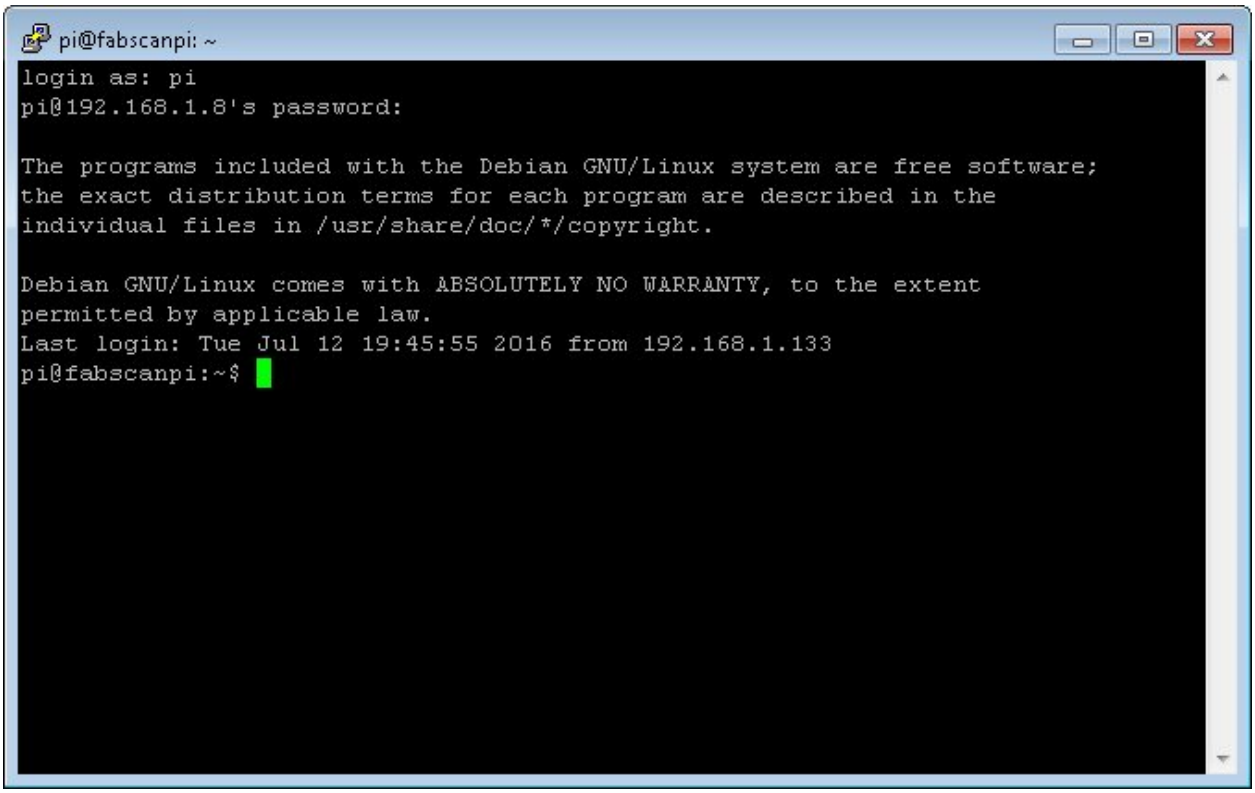
You don't need to make an installation just put the putty.exe in a folder or your desktop. Of course you can start it directly from the download folder as well.

Now you must know the IP-address which has be assigned to your FabScanPi. It is the same address you're using to get access via the webbrowser (e.g. 192.168.1.8). Usually you can check the current IP-address in the user-interface of your web-router or cable modem.

Start **Putty.exe** and a window will pop up.



Type in your IP-address in the appropriate field and click on "OPEN".



Now the console window opens and you must type in "**pi**" as **login-name** and "**raspberry**" as **password** (without the quotes). Now you should be able to see the login prompt (similar to the picture above).

Connecting as Linux or OSX User (Terminal)

Linux users can open a Terminal and type "**ssh pi@fabscanpi.local**" ("**sh pi@ip-address-of-you-pi**"). Next you are asked for a password. Type in "**raspberry**" as **password** (without the quotes). The next steps are described with screenshots for Putty, but the workflow is the same once the connection is established.

Changing the config file.

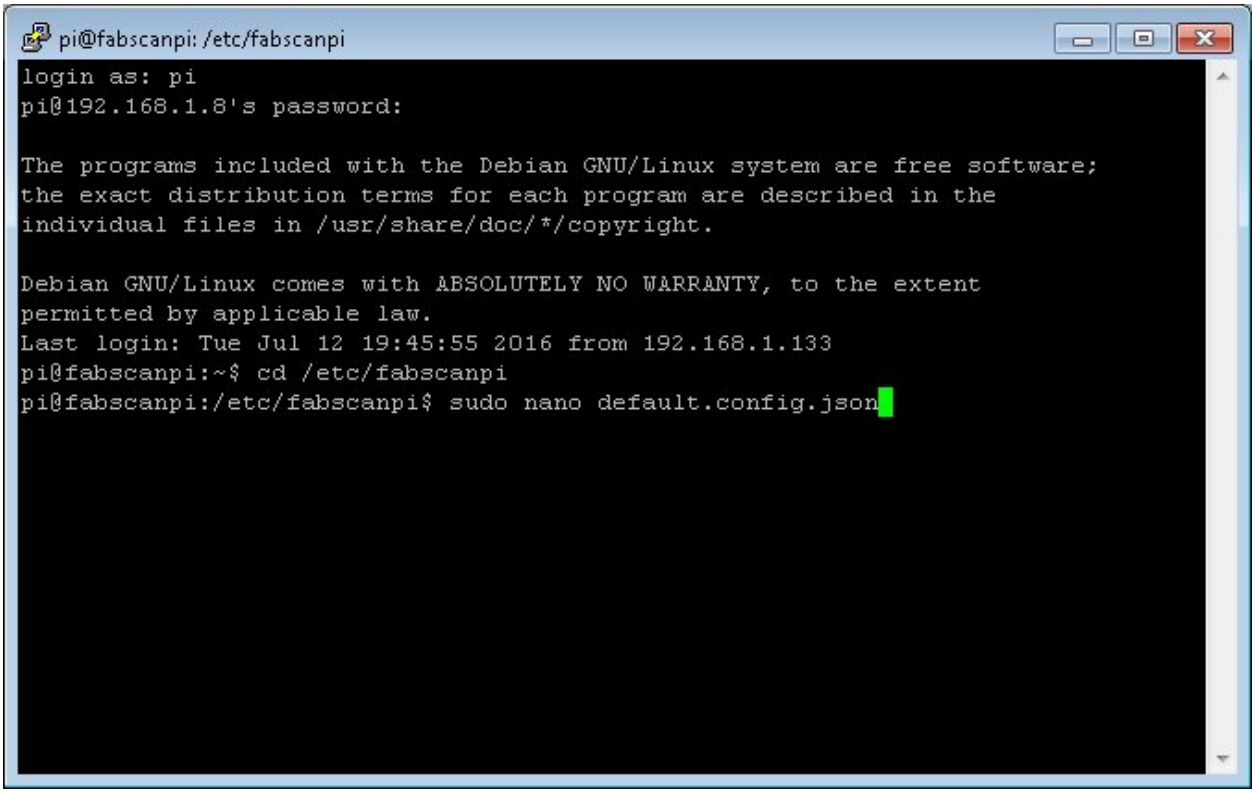
The config file is in a different folder, so you must change into that folder by entering the command:

```
cd /etc/fabscanpi/
```

and press ENTER.

To view and modify the config file (default.config.javascript) you must open it with an editor and using administrator rights to be able to save the changes into the same file. The editor which is already installed is called nano. So type in:

sudo nano default.config.javascript



You have to enter the password "**raspberry**" (without the quotes) again, because you open the editor with administrator rights.

The nano-editor now displays the config-file and maybe you have to enlarge the window to have a better view.

Now you can perform the desired changes by using the keyboard. To navigate you have to use the up-, down-, left- and right-key.

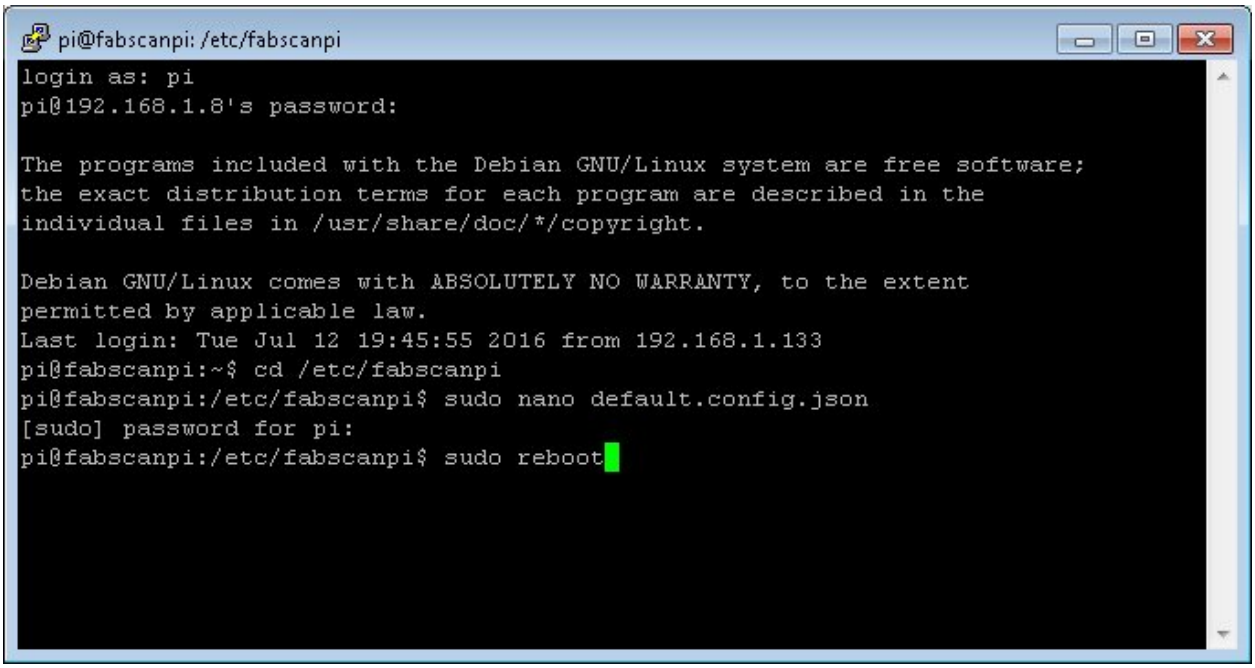
If you finished your modification press you can save the file by pressing and holding **CTRL** and **O** (german keyboard: STRG and O). Press RETURN to confirm the filename.

Now you can exit the editor by pressing and holding **CTRL** and **X** (german keyboard: STRG and X).

For the changes to take effect you must restart your FabScanPi by typing in the command

sudo reboot

and ENTER.



You can now close the PuTTY window.

The FabScanPi is rebooting and after a short time you can refresh your webbrowser and start using the FabScanPi with the new config settings.

General Values

A configuration file can be found in /etc/fabscanpi/default.config.javascript. The content of this file is in javascript format and can be edited with an editor of your choice (e.g. nano). Be careful and don't miss brackets. javascript is really sensitive in it's format.

Note: The following listing snippets are only examples and may deviate from the latest settings.

Folders

In this section you can change the scan output folder and the folder where the ui is located. If you don't know what you are doing, it is a good decision to keep this section untouched.

```
"folders": {
    "www": "/usr/share/fabscanpi/",
    "scans": "/var/scans/"
}
```

Laser

This section describes the laser stepper motor values. The numbers defines how many lasers you are using (recommented values are 1 and 2). The color defines the color of the Laser. Green should work also with a value of "G

(RGB)". The interleaved setting on True means that the scanner will calculate a background difference image while it is detecting the laser. This value on True will increases the scanning time but scanners without a case (e.g. ciclop, cowtech atlas etc.) will work with the interleaved value on True.

```
"laser": {
    "numbers": 1,
    "color": "R (RGB)"
    "interleaved": False
}
```

If you want to use the second laser you should increase the value "numbers" to 2.

Meshlab settings

In this section you can change the path for the converter which transforms the scanned pixel data into another format (e.g. .stl).

```
"meshlab": {
    "path": "/usr/bin/"
}
```

Table settings

In this section you can change the turntable settings. The radius of the turntable is in millimeters (mm). In the default case the motor is set to 1/16 step mode. A motor with 200 steps per turn can then perform 3200 steps. Radius is the radius of the turntable in millimeters (mm).

```
"process_numbers": 4,
"turntable": {
    "steps": 6400,
    "radius": 70
}
```

Note: The default motor can perform a full turn in 400 steps, therefore the default steps value is 6400. If you are using a motor with only 200 steps per turn, change the steps value to 3200. If you don't change this, your scan will look mirrored. See F.A.Q in the forum for more information.

Camera settings

Preview Resolution is the resolution value for the settings window. Resolution is the resolution for the picamera python module. You can have a look to the documentation of picamera. If you set this to other values please be sure what you are doing, not all resolutions are supportedby the picam. Some might lead to slower image capturing.

```
"camera": {
    "resolution": {
        "width": 1640,
        "height": 1232
    },
    "preview_resolution": {
        "width": 240,
        "height": 320
    },
    "rotate": "True",
    "hflip": "True",
    "vflip": "False",
    "type": "PICAM"
}
```

Serial

In this section you can set your port. By default this value is not set, because theFabScanPi Server software autodetcts the correct port. Some Arduino and compatible boards differ in the port name. The port can be set if you are not using an Arduino UNO or compatible Board. In case that your Arduino is not detected and you can see an error in the /var/log/fabscanpi/fabscan.logyou should add the "port" attribute to your config.

The autoflash option is True by default, that means that the firmware is flashed automatically to the Arduino or FabScanPi HAT. If you want to use a custom board e.g. sanguinololu, you can set thisto False and flash the Firmware manually to your board.

```
"serial": {
    "baudrate": 115200,
    "autoflash": "True",
    "port": "/dev/ttyAMA0"
}
```

The default firmware flashing baudrate can be changed by adding "flash_baudrate" to the serial settings.

Texture illumination

In this section you can change the pre-set brightness level of the LED-Ring during texture scan.

```
"texture_illumination": 140
}
```

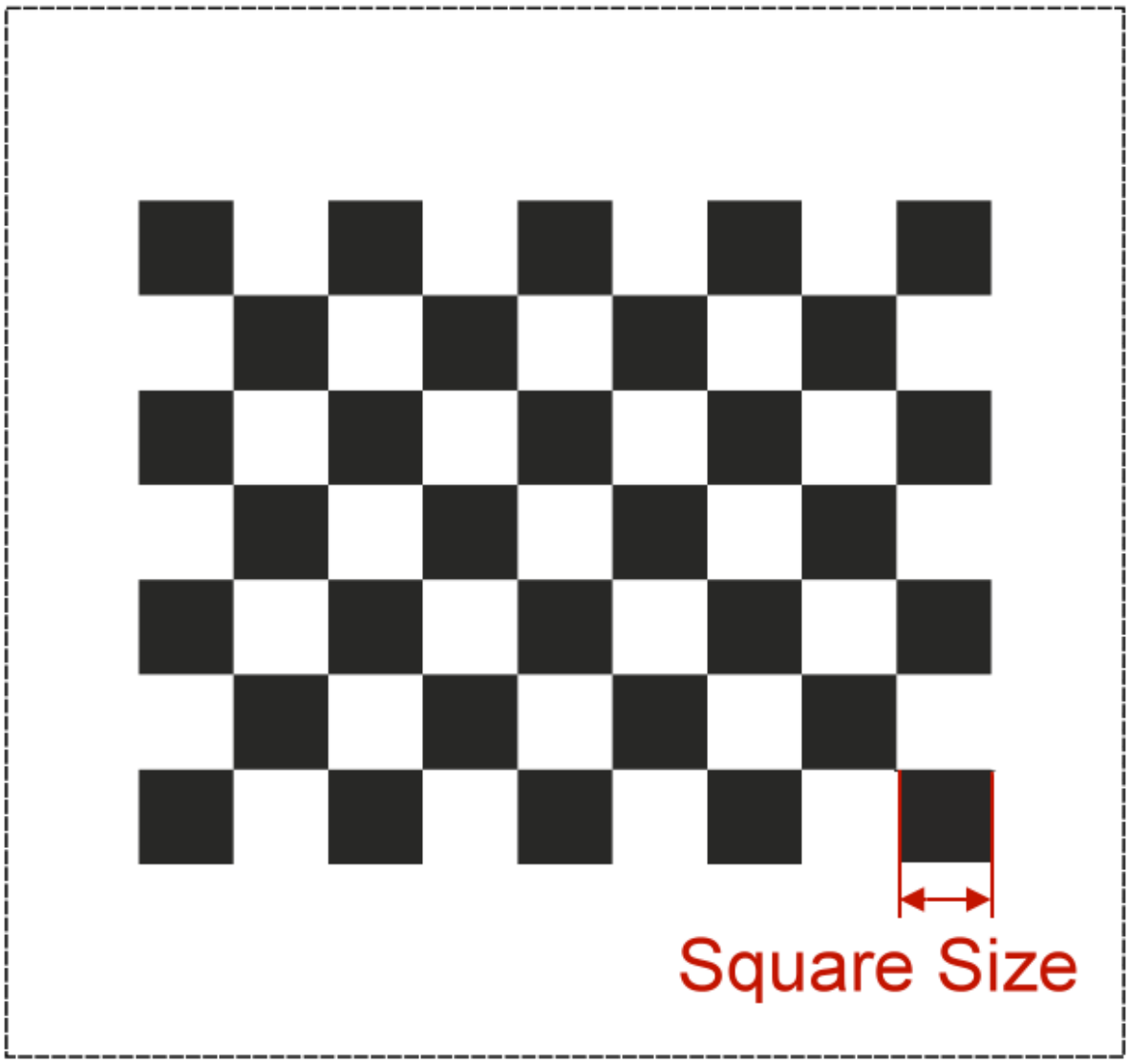
Calibration Values

In this section you can change the parameters of the configuration sheet. If your printout of the calibration sheet has not the exact scale you can adjust the parameters here instead of scaling the print.

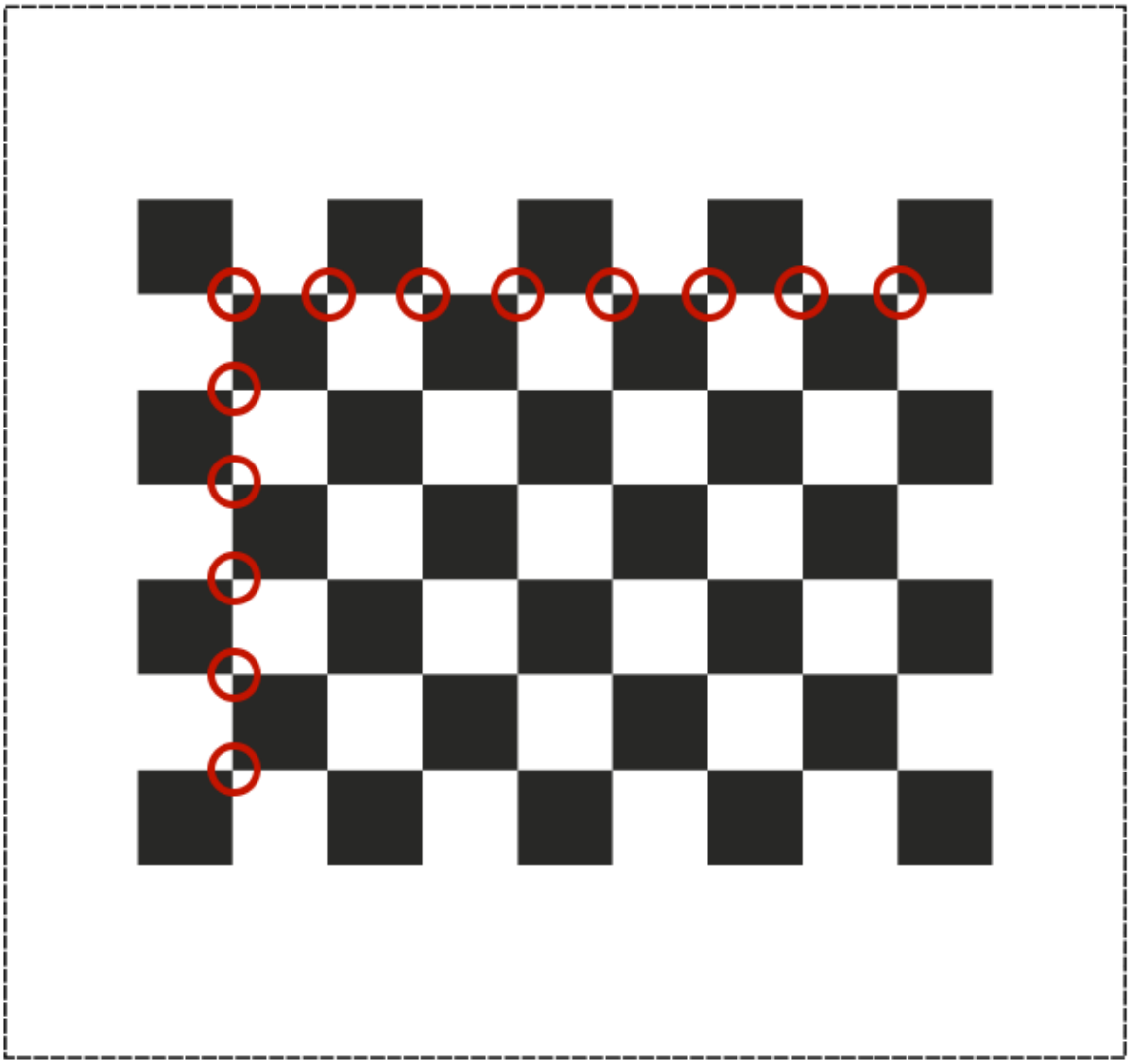
Note: There is a new " 8x6 Calibration Pattern". If you are still using the old 9x6 Pattern you'll need to modify the columns value:

```
"scanner_type": "laserscanner",
"calibration": {
"weight_matrix": [],
"dist_camera_matrix": [],
"pattern": {
    "square_size": 11,
    "rows": 6,
    "columns": 8,
    "origin_distance": 35
}
```

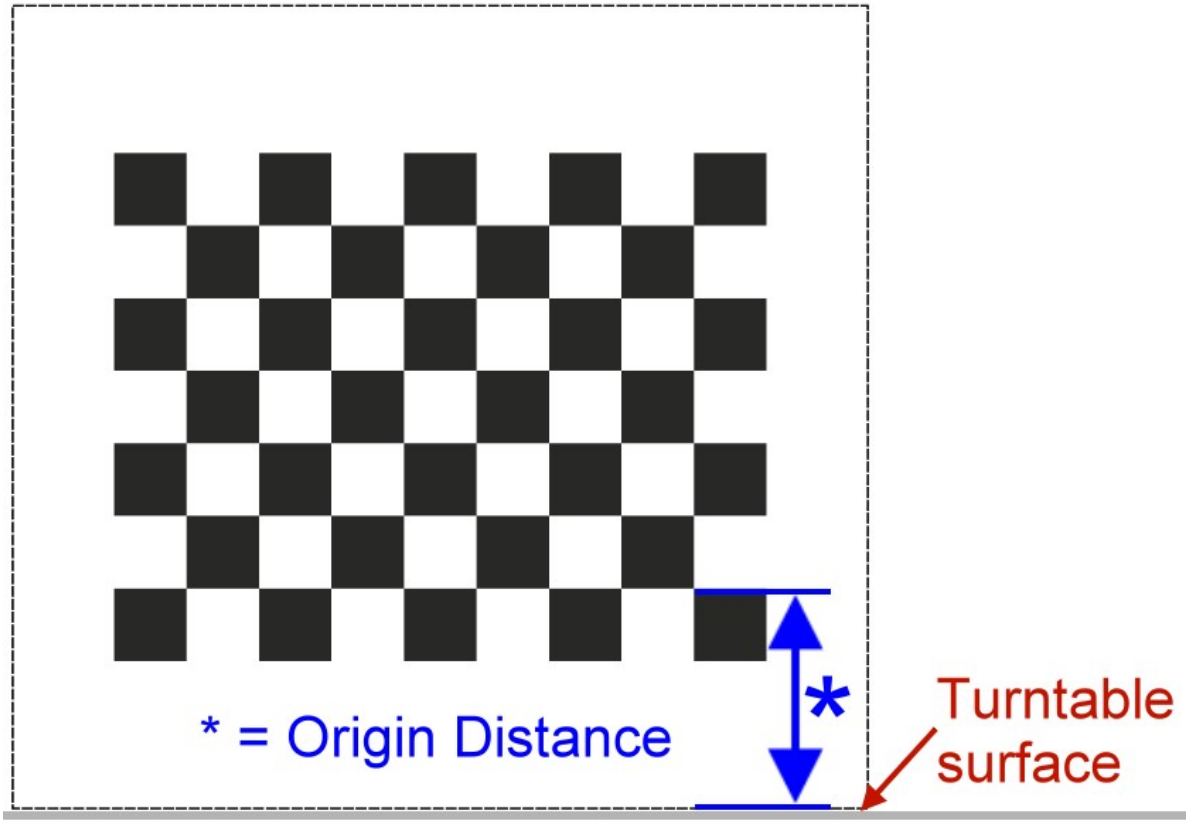
- Square Size is the side length of one black square in millimeters.



- Rows and Columns are the connection points of the black squares. The correct number is 8 for columns and 6 for rows :



- Origin Distance is the distance between turntable surface and the upper edge of the black squares in the row close to the turntable.



Scanner Calibration Values

In this section you can check the calibration parameters. Please make sure you have performed the auto-calibration before starting your first scan. Do not change these values manually. This values are generated by the autocalibration process.

```
    "camera_matrix": [
        [
            1285.5809999999999,
            0.0,
            647.60199999999998
        ],
        [
            0.0,
            1289.9490000000001,
            835.84400000000005
        ],
        [
            0.0,
            0.0,
            1.0
        ]
    ],
    "distortion_vector": [
        0.151,
        -0.20300000000000001,
        -0.0050000000000000001,
        0.0060000000000000001,
```

(continues on next page)

(continued from previous page)

```
            -0.70899999999999996
        ],
        "laser_planes": [
            {
                "deviation": 0.052318819865,
                "distance": 137.366403938,
                "normal": [
                    0.56199451,
                    -0.01896656,
                    0.82692348
                ]
            }
        ],
        "platform_translation": [
            4.21176054e-03,
            4.26178340e+01,
            1.66114592e+02
        ],
        "platform_rotation": [
            [
                0.0,
                9.99977231e-01,
                6.74816764e-03
            ],
            [
                4.51612662e-02,
                6.74128255e-03,
                -9.98956964e-01
            ],
            [
                -0.99903697271524872,
                0.00030800546235732861,
                -0.043875189806843448
            ]
        ]
    }
```

Images

This setting can be used for keeping the raw images.

```
    "keep_raw_images": "True",
```

Online

The following parameters are internet connection settings. the “discoverable” flag can be used to make the fabscan discoverable on find.fabscan.org. This will help you to discover the ip address of your fabscan.

```
    "discoverable": "True",
```

The “online_lookuo_ip” setting is used to check if the fabscan is online or not. This will prevent long timeouts when the user interface is called. The default value is a google dns ip address.

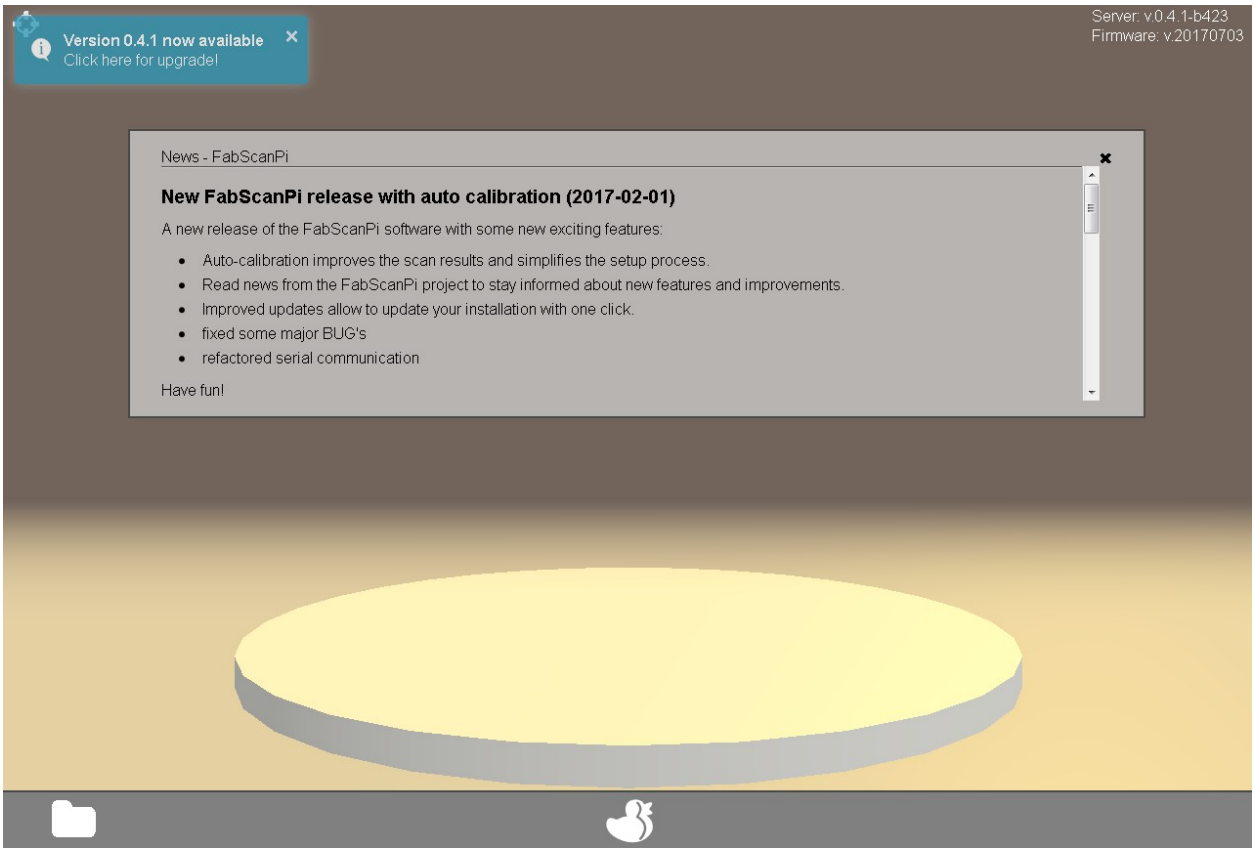
```
    "online_lookup_ip": "8.8.8.8"
```

Using the Software

Getting Started

- By default the FabScanPi server binds to all interfaces on port 8080. Pointing your browser to http://ip-of-your-raspberry-pi:8080 will open the user interface.

  If your FabScanPi has access to the internet an information window with the latest project news will pop-up. You can close it by left-clicking on the black x in the upper right corner of the grey info window. If there is a new software version available you will see a note in the upper left corner:



First start-up

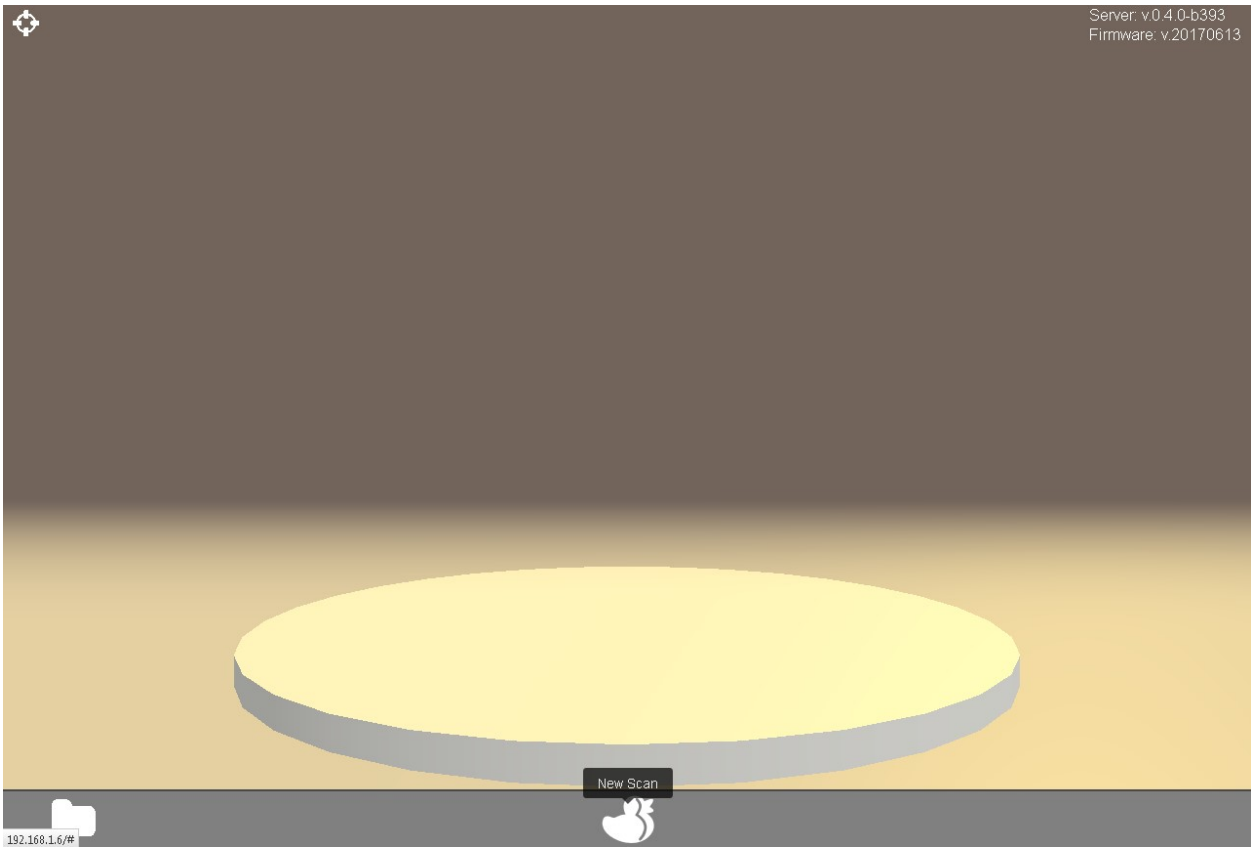
Before you proceed with the calibration and your first scan please check a few things to ensure your FabScanPi is working properly:

- Click on the duck icon (New Scan) to enter the scan menu.
- Now the turntable should turn clockwise in a constant smooth manner.
- The laser must be on and needs to be adjusted vertically and to cross the center of the turntable. You will have to do the adjustment by hand because the servo / stepper support for the laser has not been implemented yet.
- Check if you can see a small video preview in the lower left corner of the scan menu. Maybe there is only a turning hour glass visible. That is a known problem of some browsers but will have not effect the scanner's function.
- Click on the X (Cancel) to get back to the main menu.

Settings

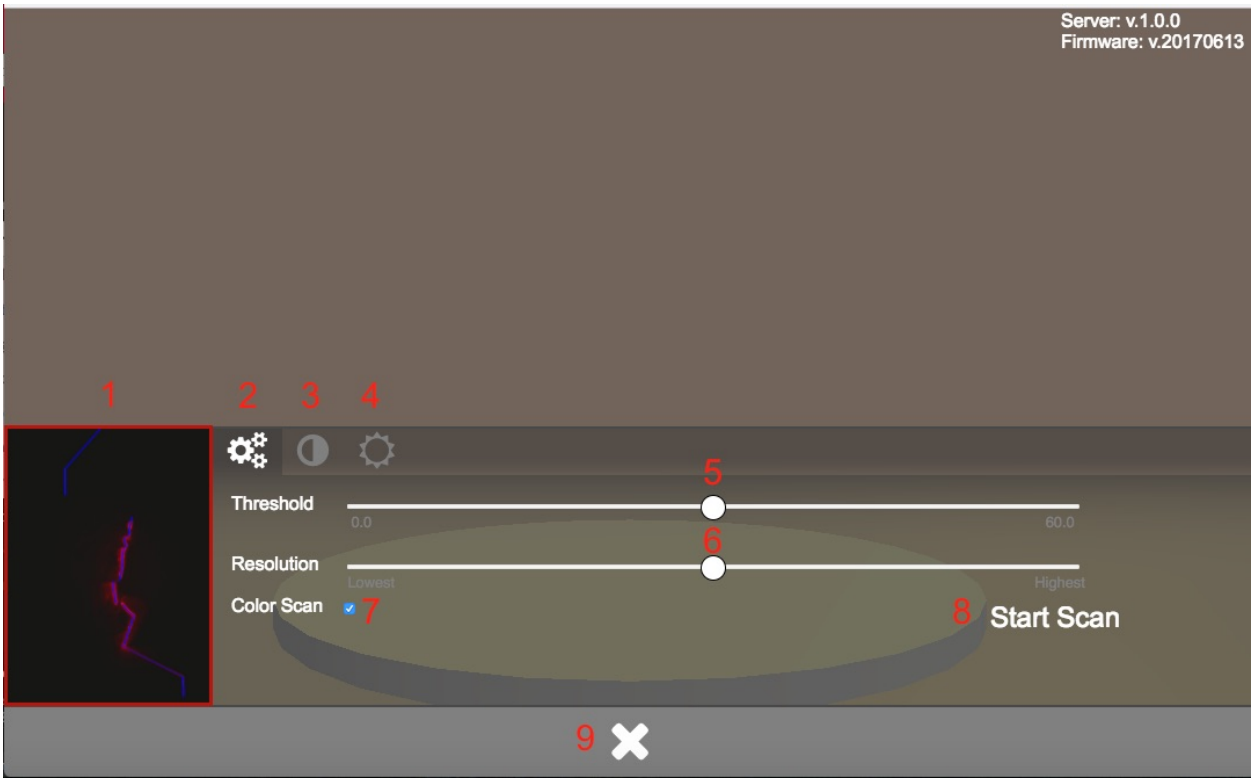
Note: The current settings are only persistent as long as the pi is up and running. The settings are saved with the scan data after a successful scan. They can be loaded to scan another object with the same settings. E.g. an object what consists of the same material, color etc.

- Click on the duckling-symbol to open the scan menu.

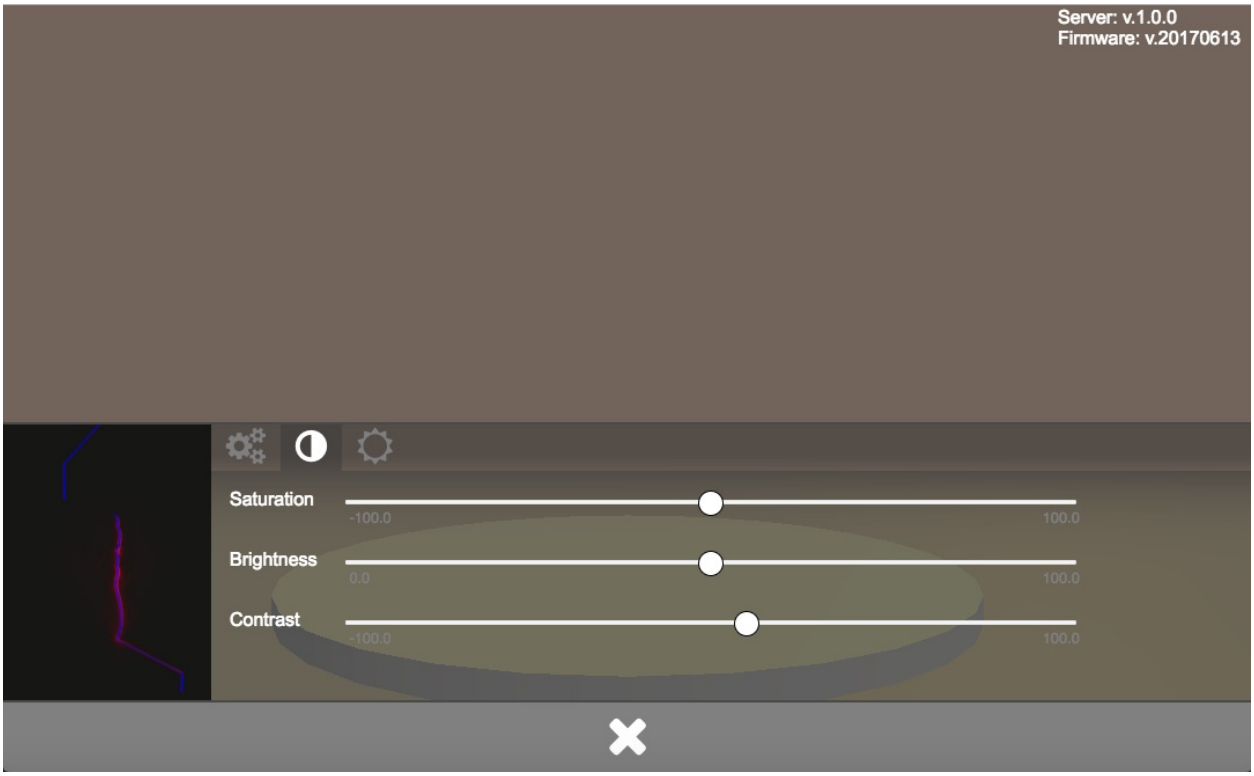


- The threshold-slider (6) can be used to adjust the sensitivity of the captured data. Select the scan quality by using the other slider (7).

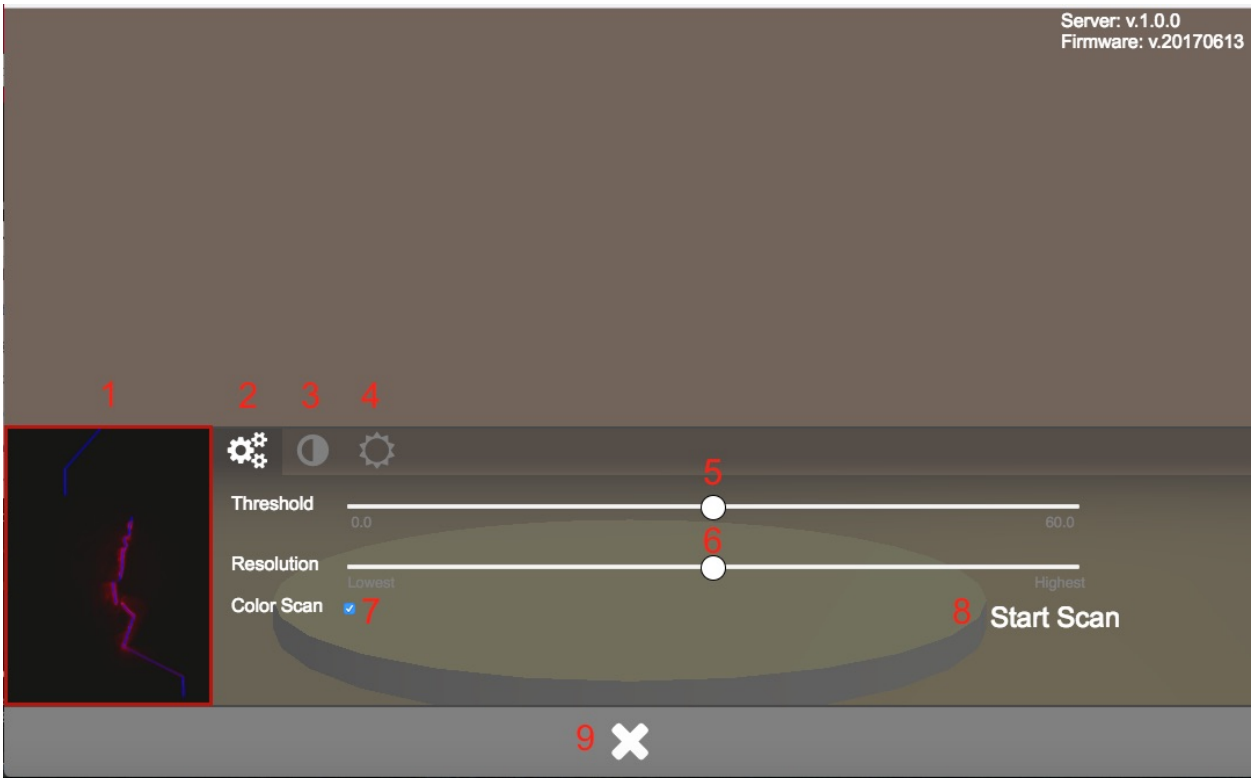
Note: The better the scan the longer is the required capture time. Sometimes it is better to start with a low resolution to control the selected settings result. If the result is nice you can perform a higher resolution scan with the same settings.



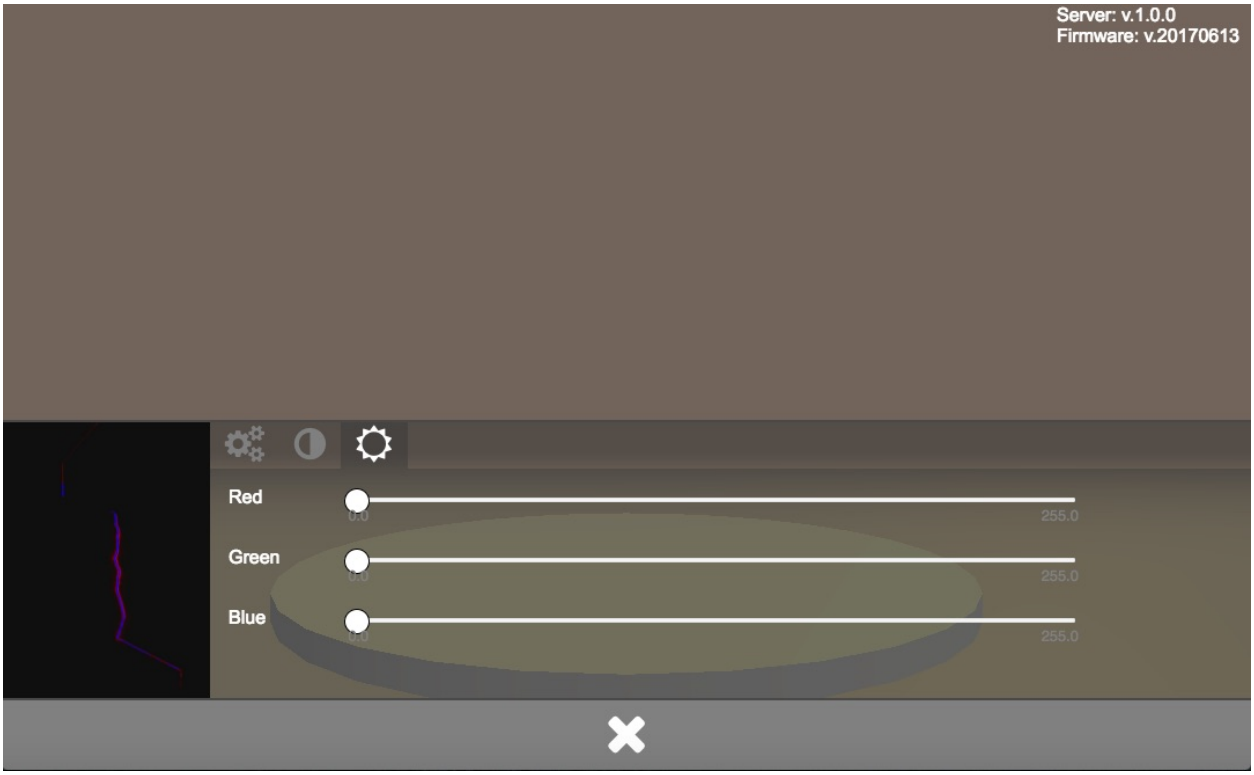
By clicking on the contrast-icon (3) you will get access to the camera settings menu. For adjusting the camera presets three sliders for saturation, brightness and contrast are available.



Click on the light-symbol (4) to get access to the lighting menu.

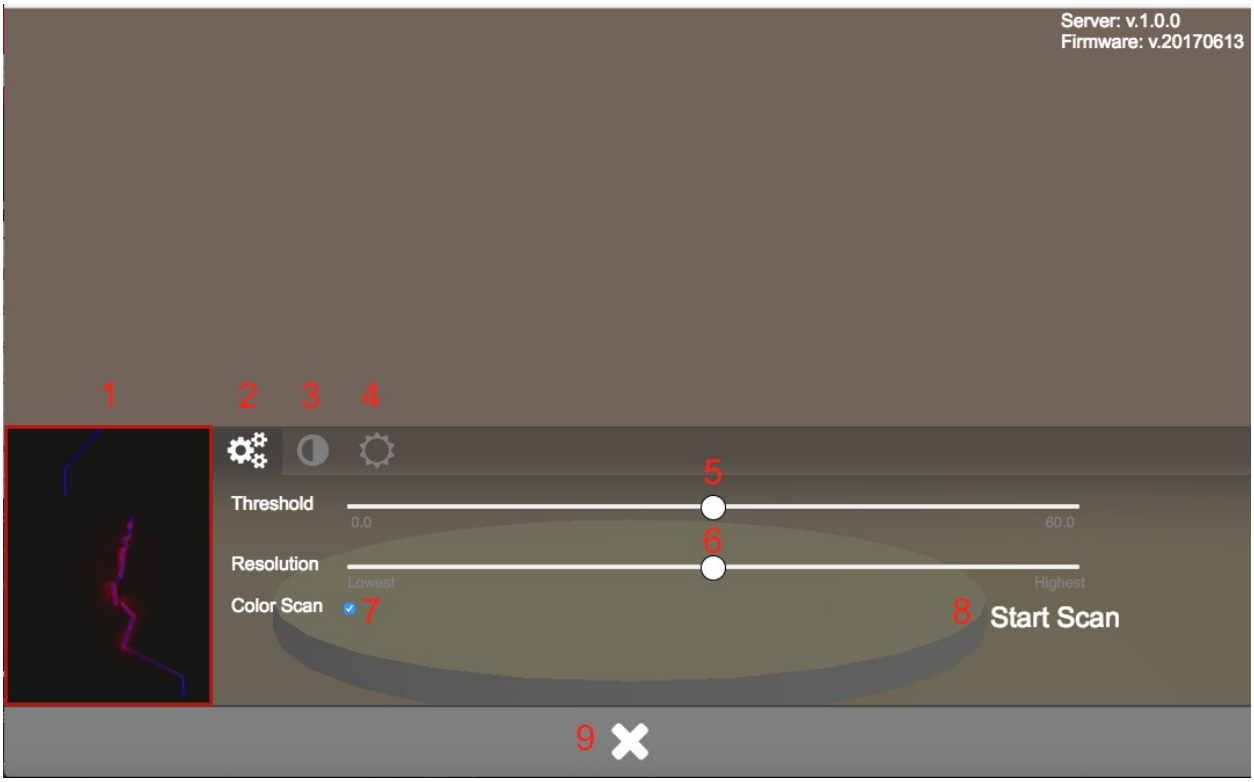


Here you can use the sliders to change the brightness and color of the (optional) light source. When all three sliders are at the very left end the light is off. Watch the preview in the lower left corner of the menu.



Note: The setting in the lighting menu will only cause an effect if an optional WS2812-compatible light source (e.g.

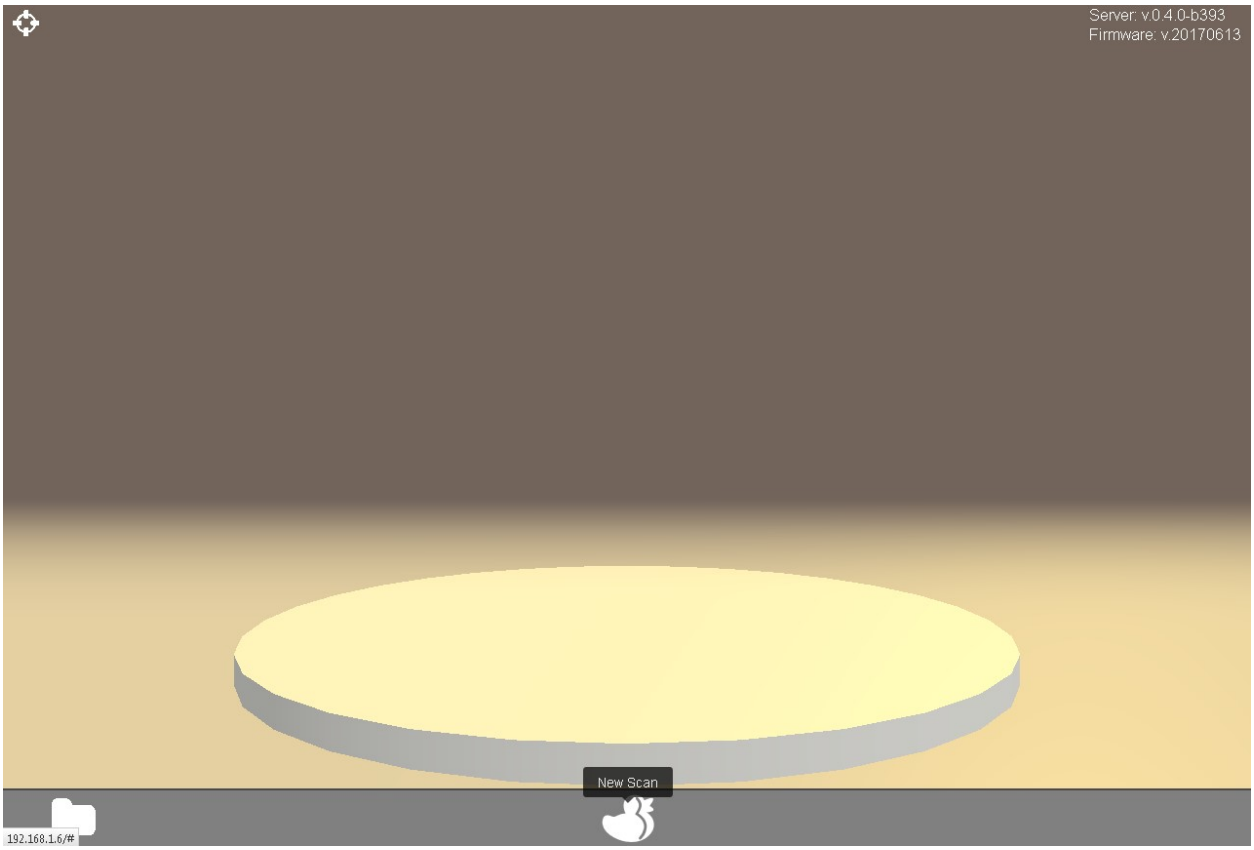
Adafruit NeoPixel LED-Ring or FabScanPi-Camera-Holder) is installed. Click on the arrows-and-circle symbol (5) to get access to the alignment menu.



Perform a San

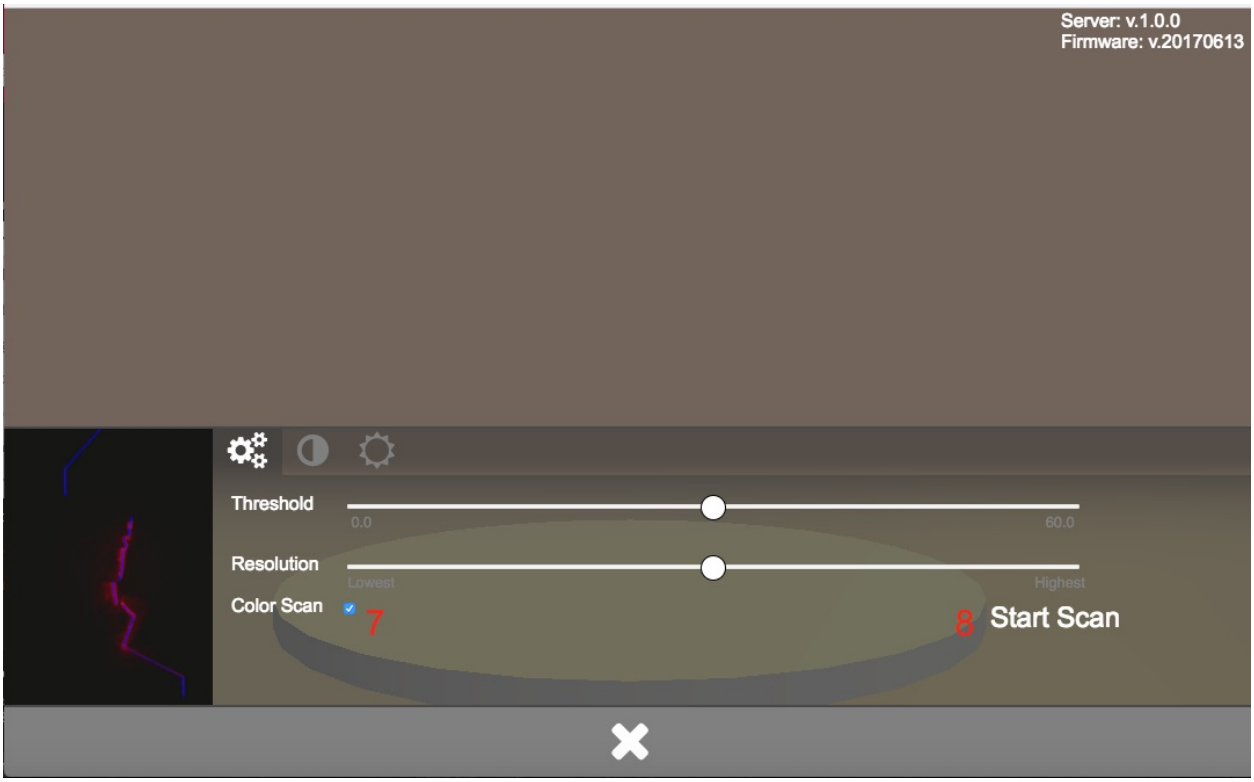
Scan with Color (slower)

- Make sure your FabScanPi is switched on, an object is placed on the turntable and the lid / the optional laser safety switch is closed.
- Open the web-interface as described in chapter [Getting Started](#gettingStarted).
- Click on the duckling-symbol to open the scan menu.

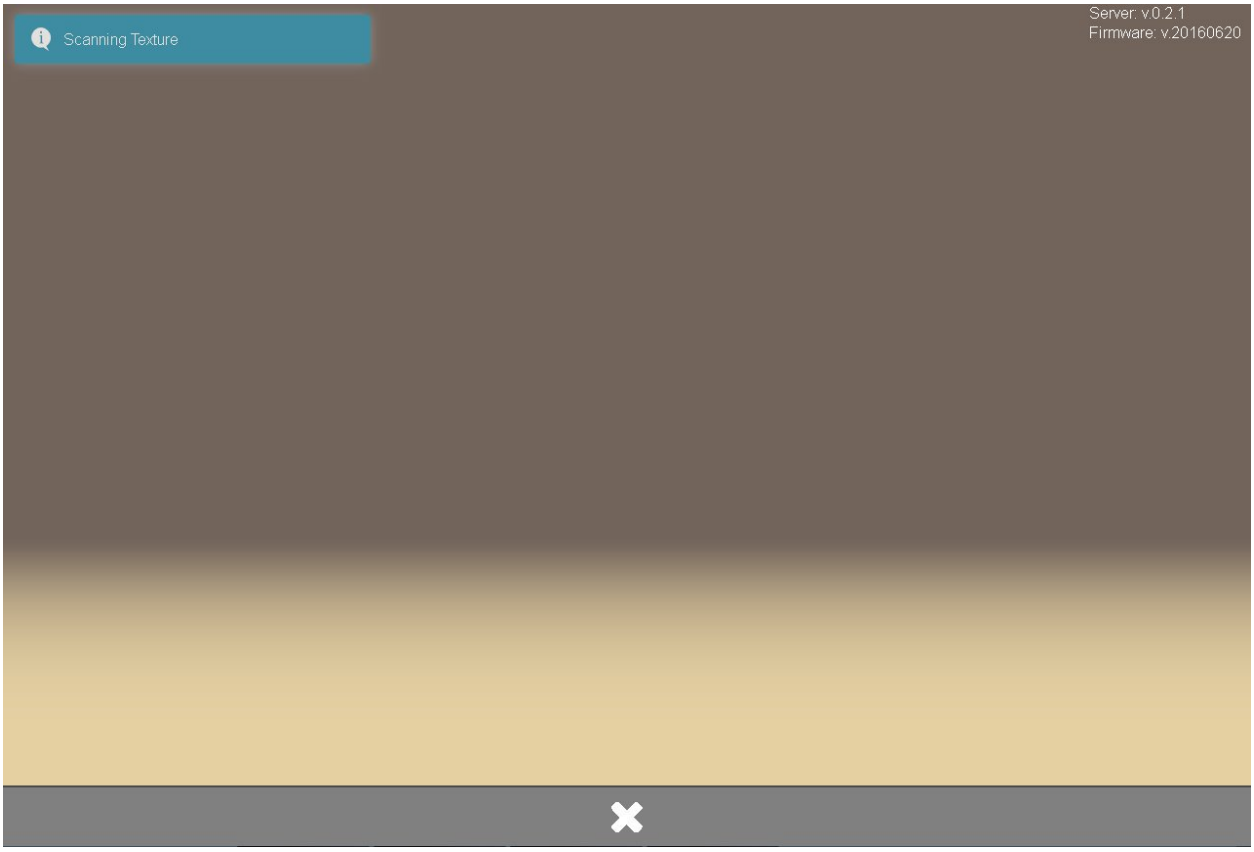


Note: If you do not have installed a light source you should perform a [monochrome scan](#monochromeScan).

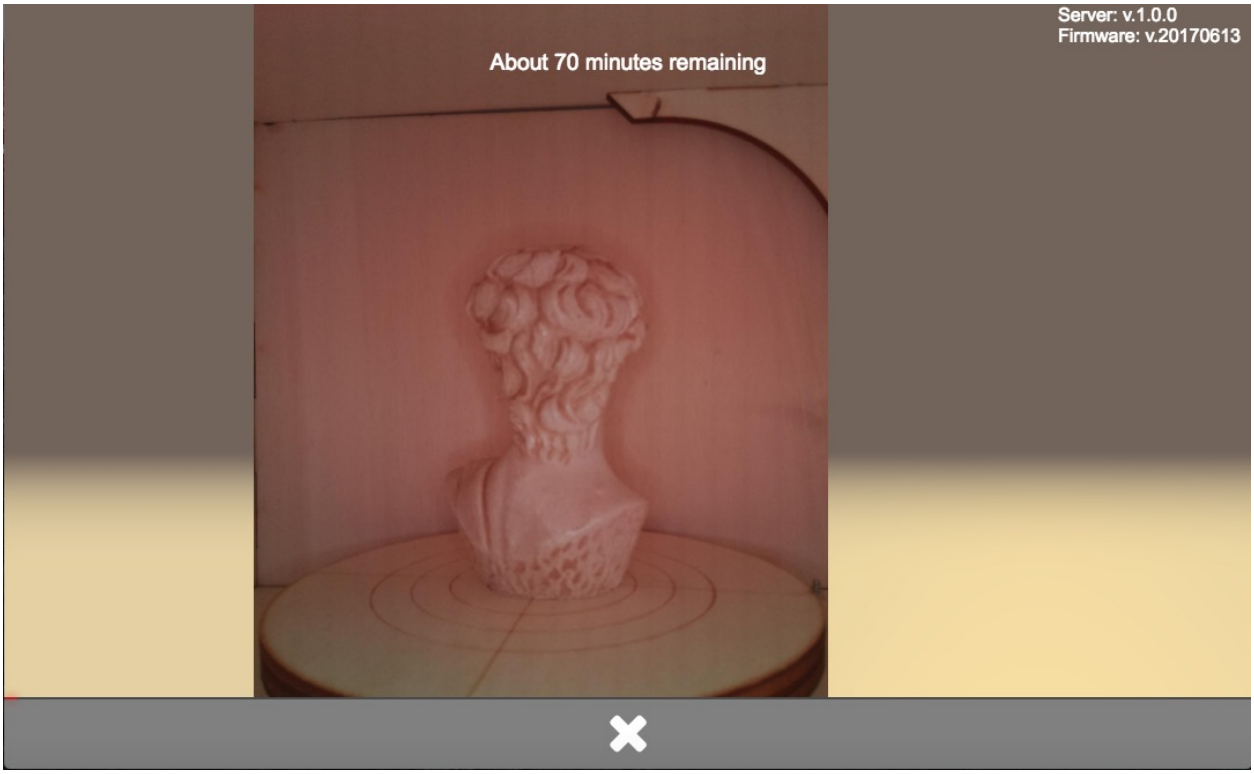
- Adjust the scan preset values to your needs as described in chapter [Presets](#presets).



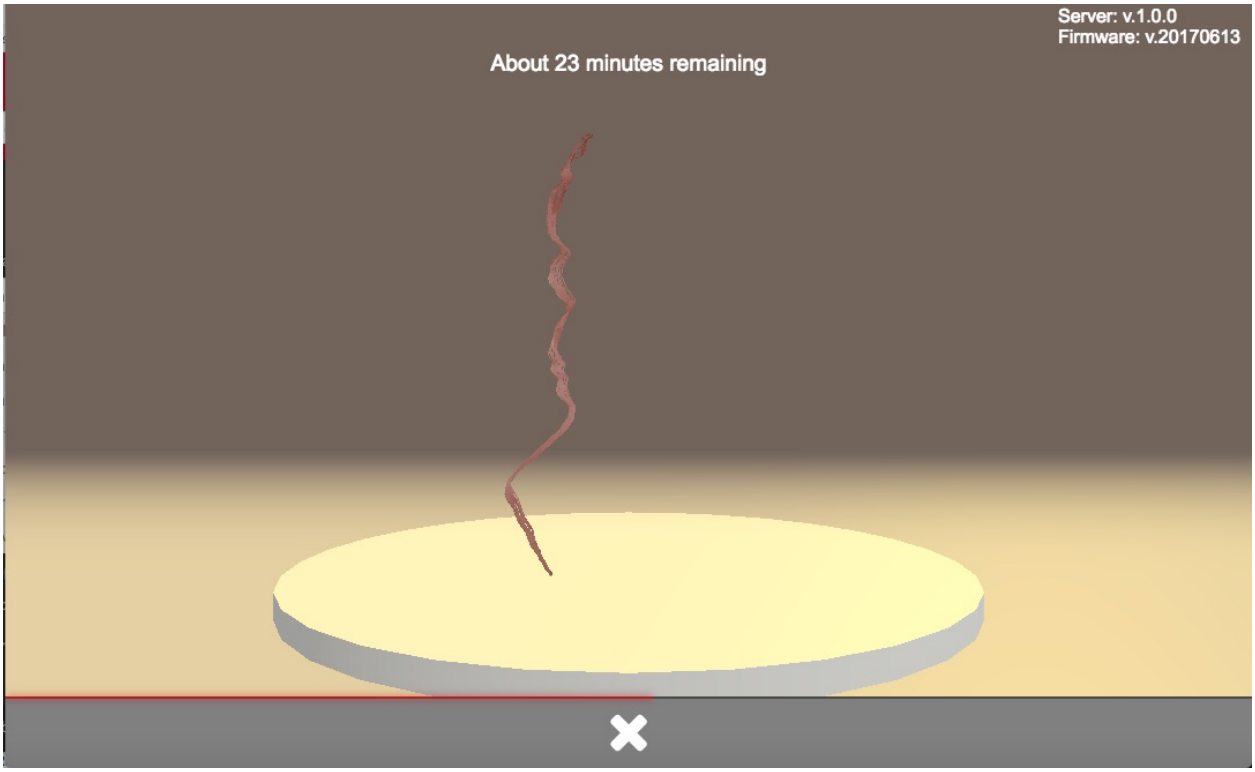
- Click on .Start Scan. to initiate the process A starting message will be displayed. Now the texture will be processed.



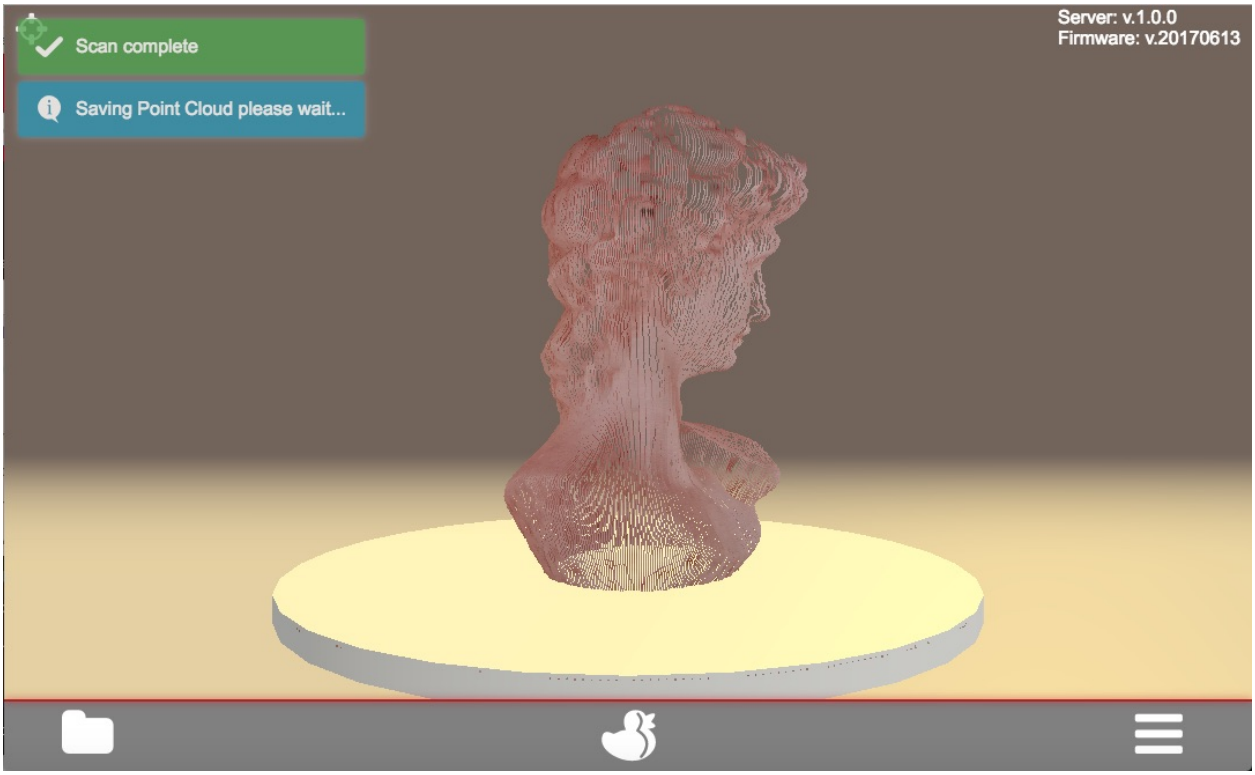
The latest photo will be displayed during the capturing process.



When the texture has been captured (progress bar at 50 percent) the actual scan is initiated. A notification is displayed.



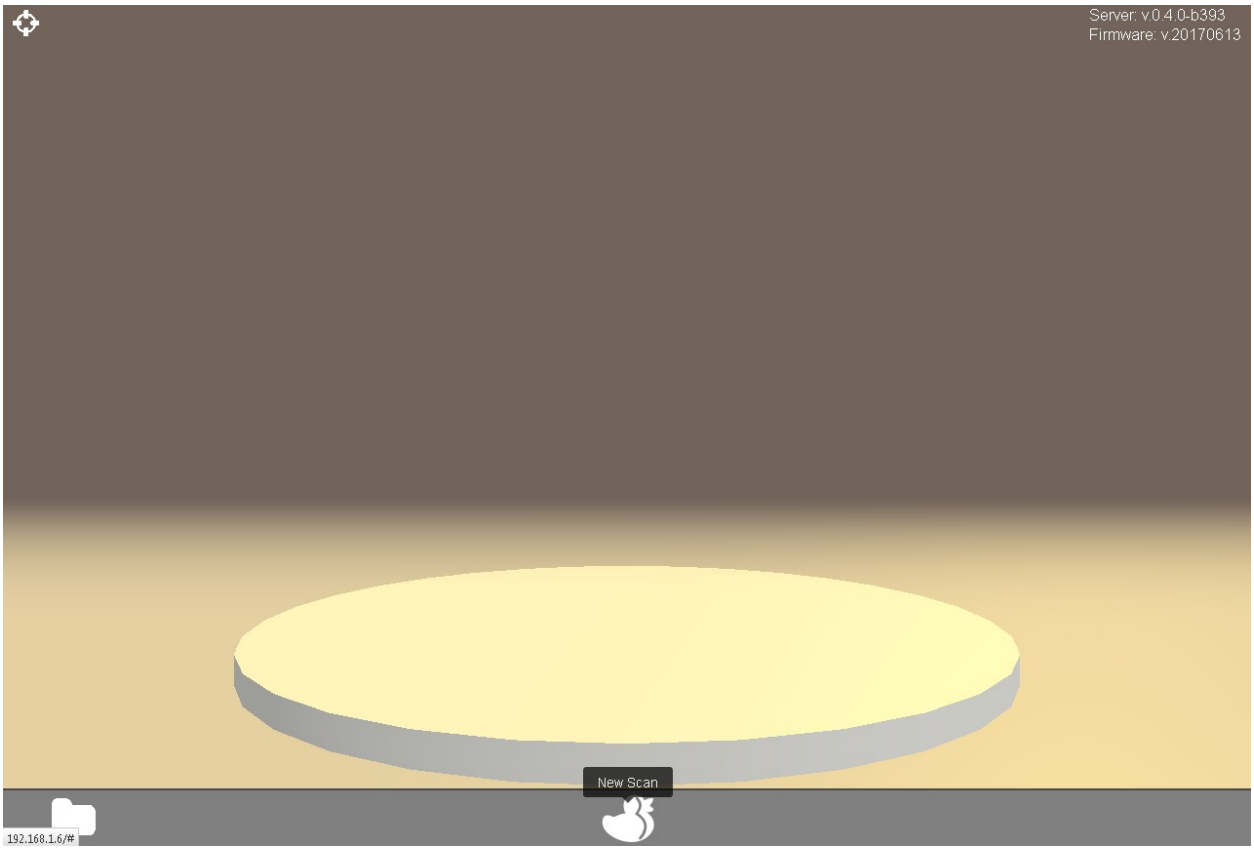
A notification will be displayed when the scan is completed / file is saved.



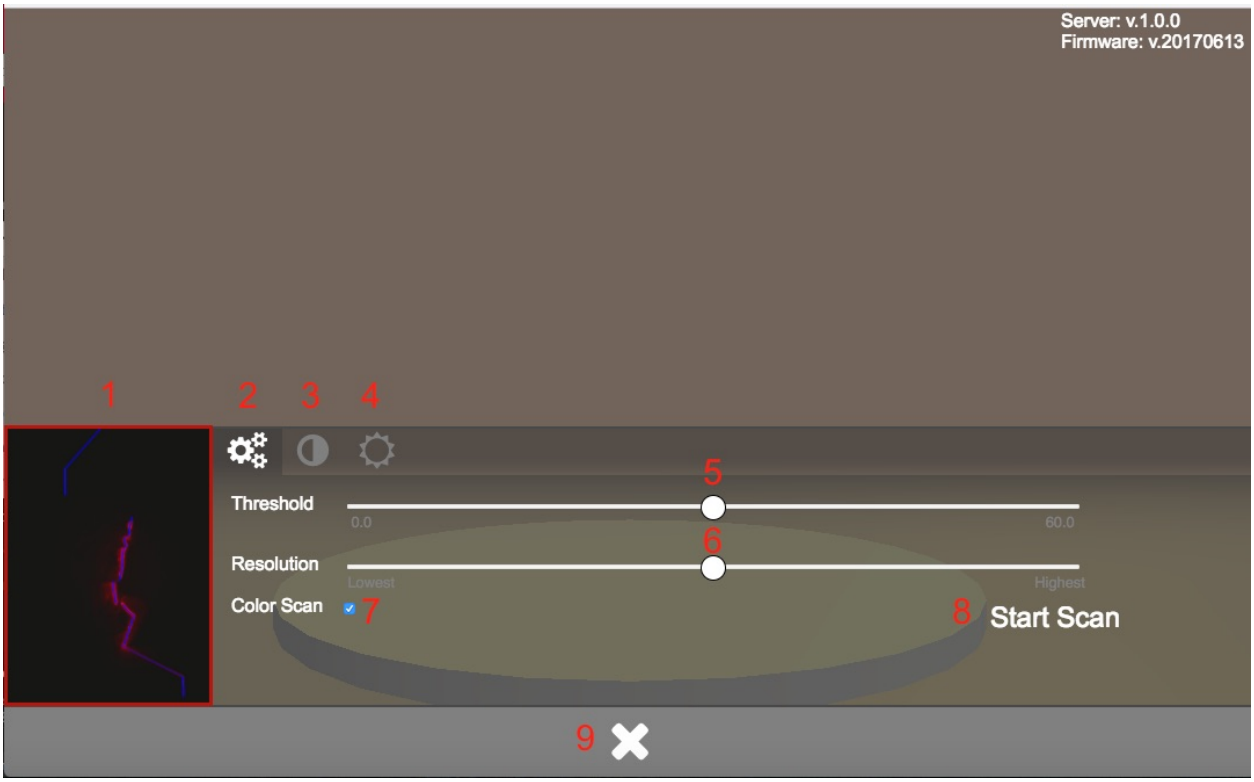
- You can now check, download or delete the scan-data.

Scan without Color (faster)

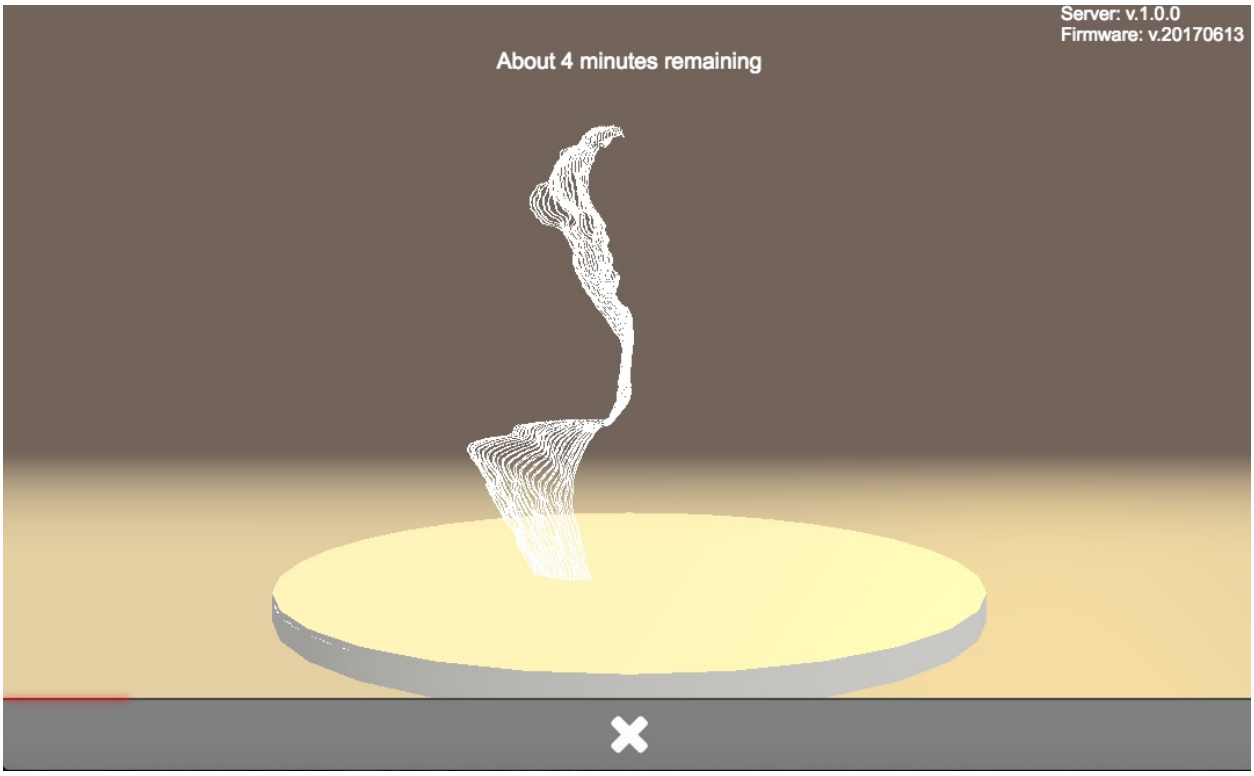
- Make sure your FabScanPi is switched on, an object is placed on the turntable and the lid / the optional laser safety switch is closed.
- Open the web-interface as described in chapter [Getting Started](#gettingStarted).
- Click on the duckling-symbol to open the scan menu.



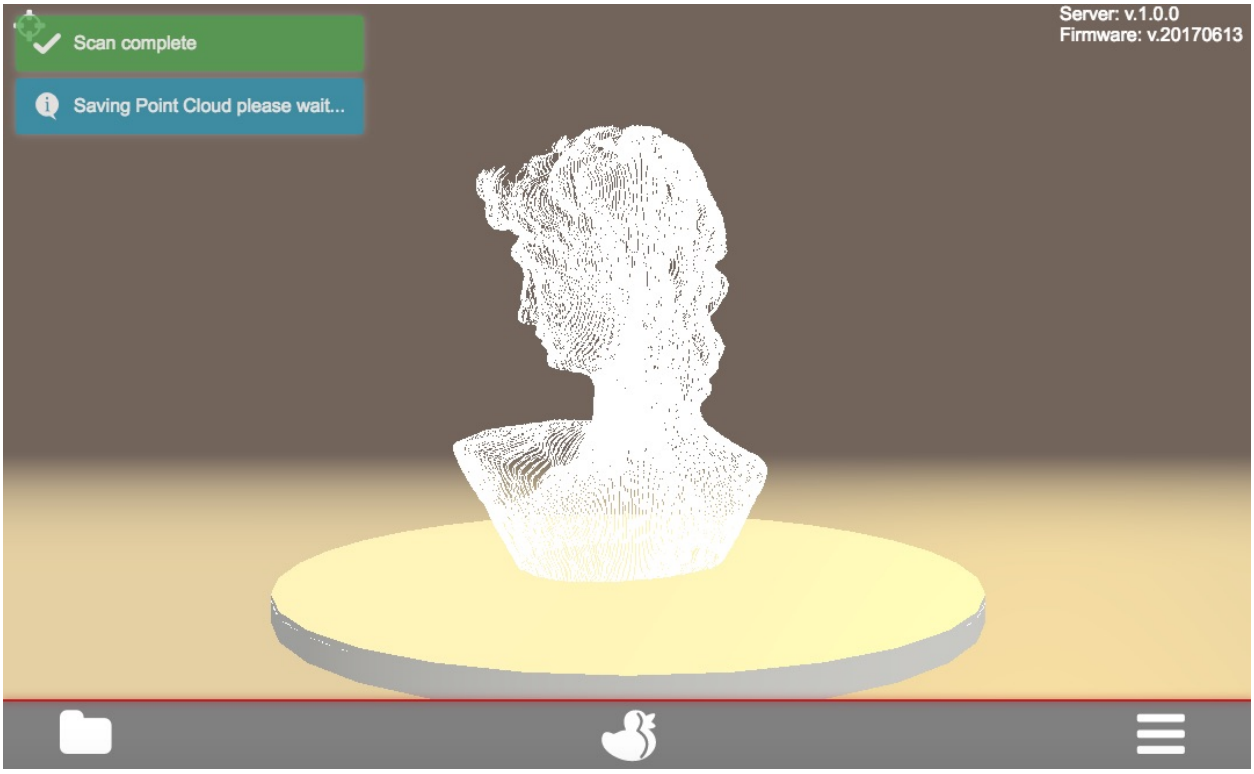
- Adjust the scan preset values to your needs as described in chapter [Presets](#presets).
- Uncheck the color-scan option (8)



- Click on .Start Scan. to initiate the process A starting message will be displayed and the scan process is started.



When the scan is completed / file is saved a notification will be displayed.



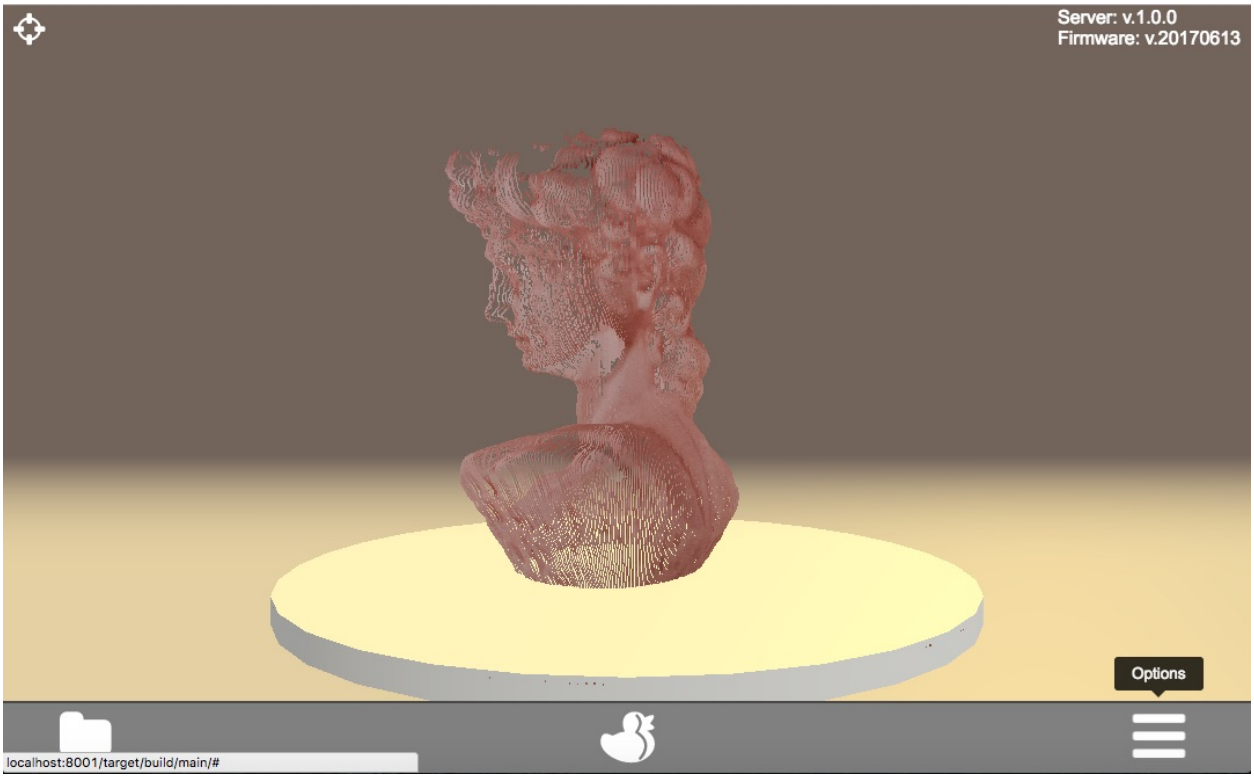
- You can now check, download or delete the scan-data.

Generate Mesh

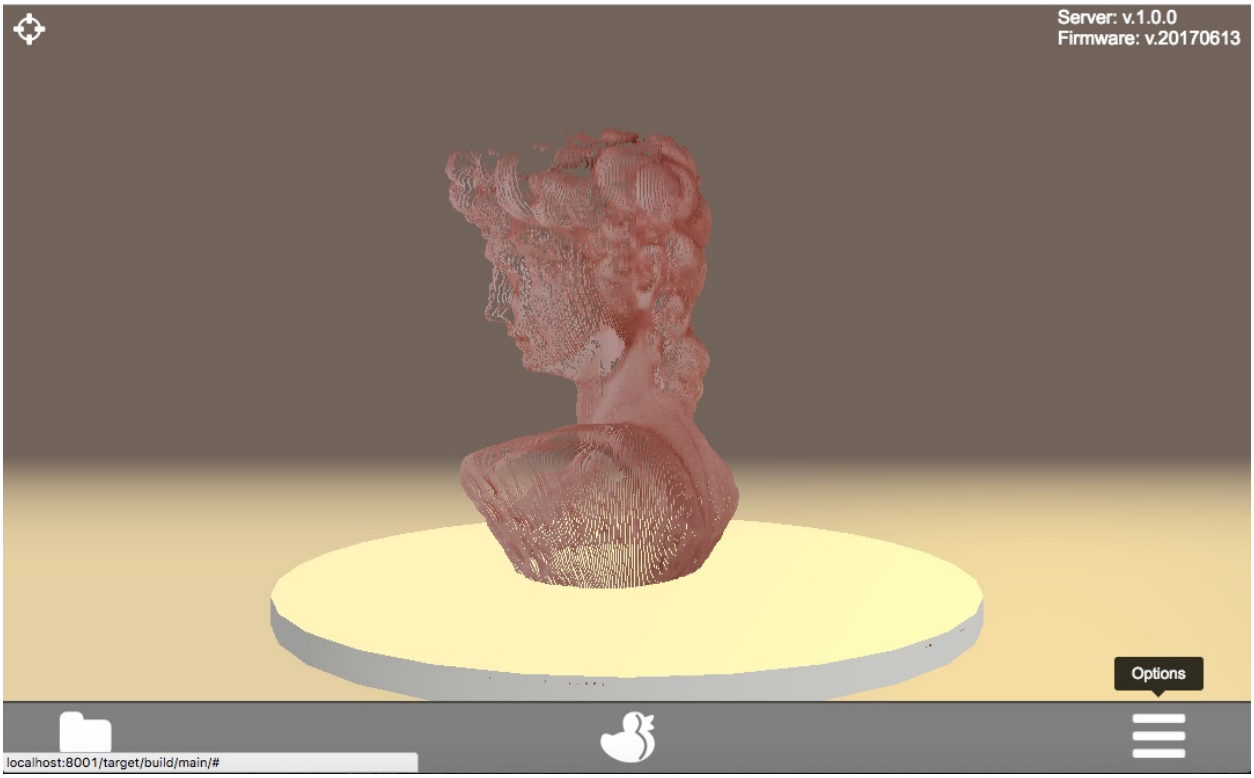
The FabScanPi software includes a feature to convert a scan into a mesh-file. This mesh-file can be used for 3D-printing.

Note: To generate a mesh-file a scan must have been performed. It is also possible to load a scan-file which has been saved previously.

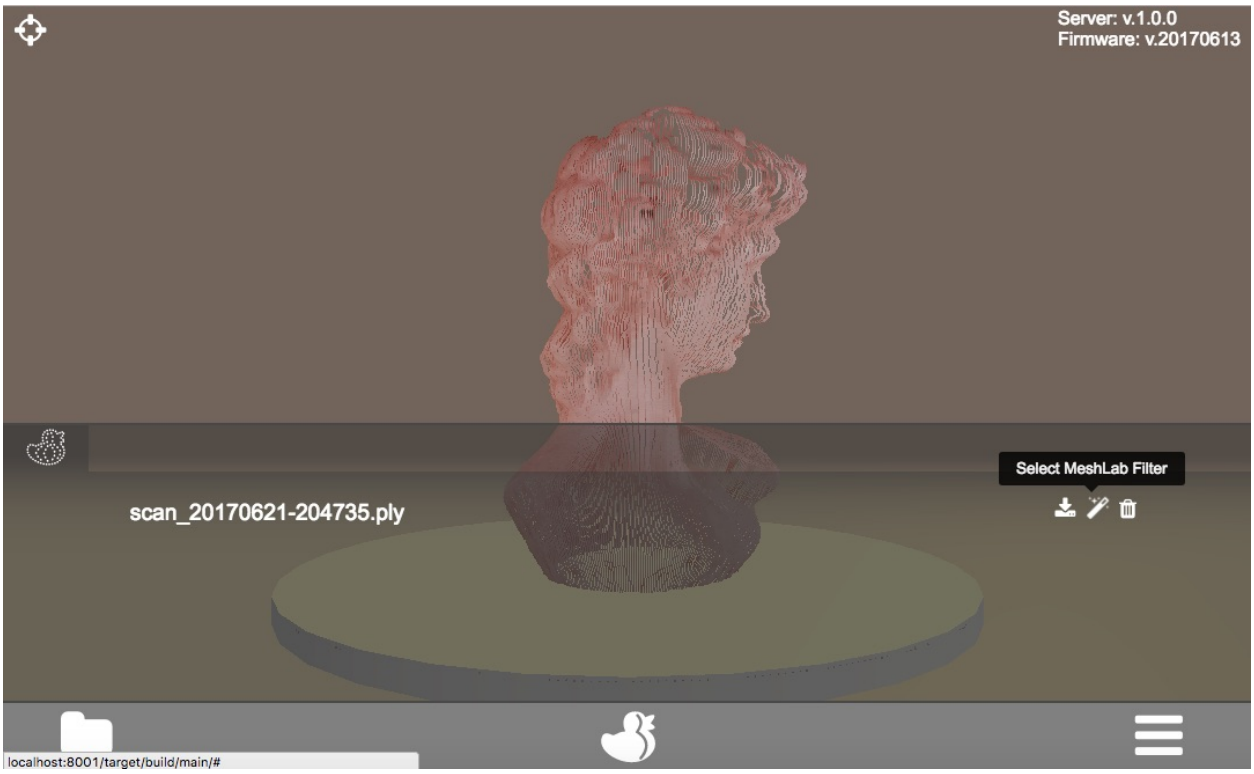
- Click on the options icon to open the options menu.



- The options menu will open and you can see the index card of the loaded file.



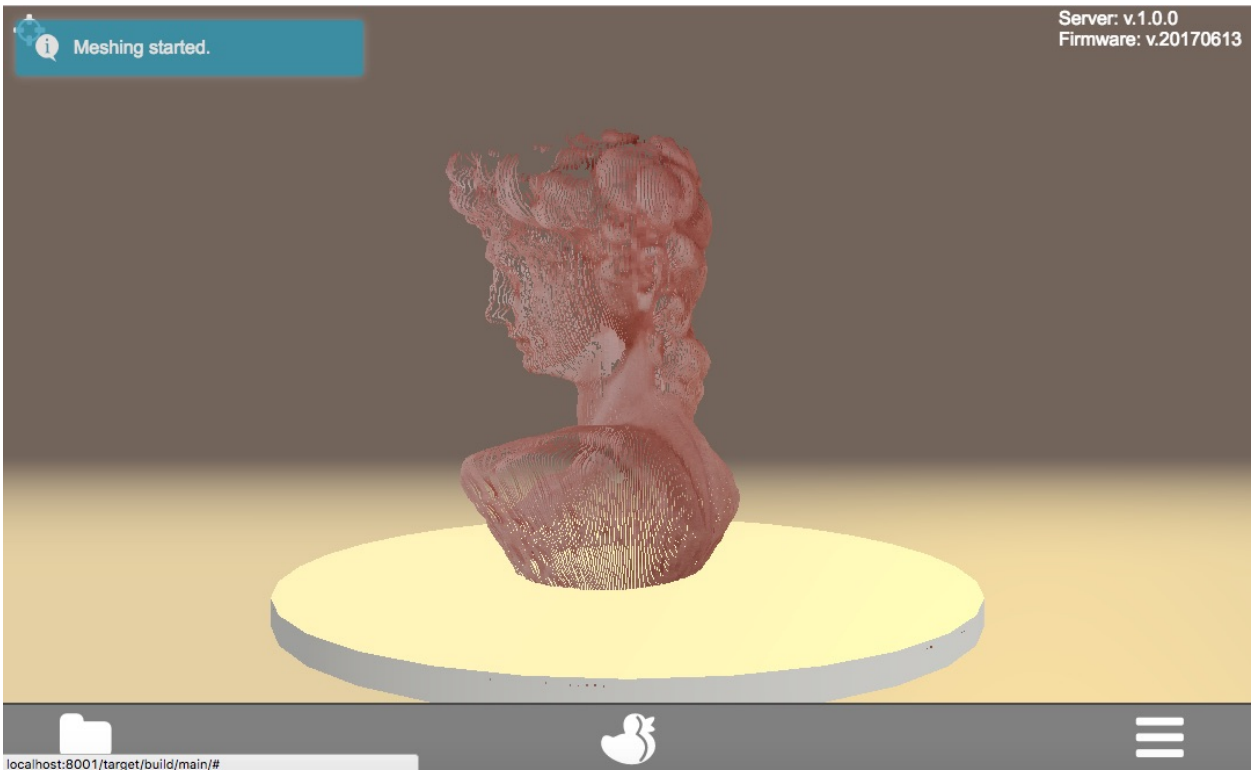
- Click on the magic wand icon to open the menu for the MeshLab filter.



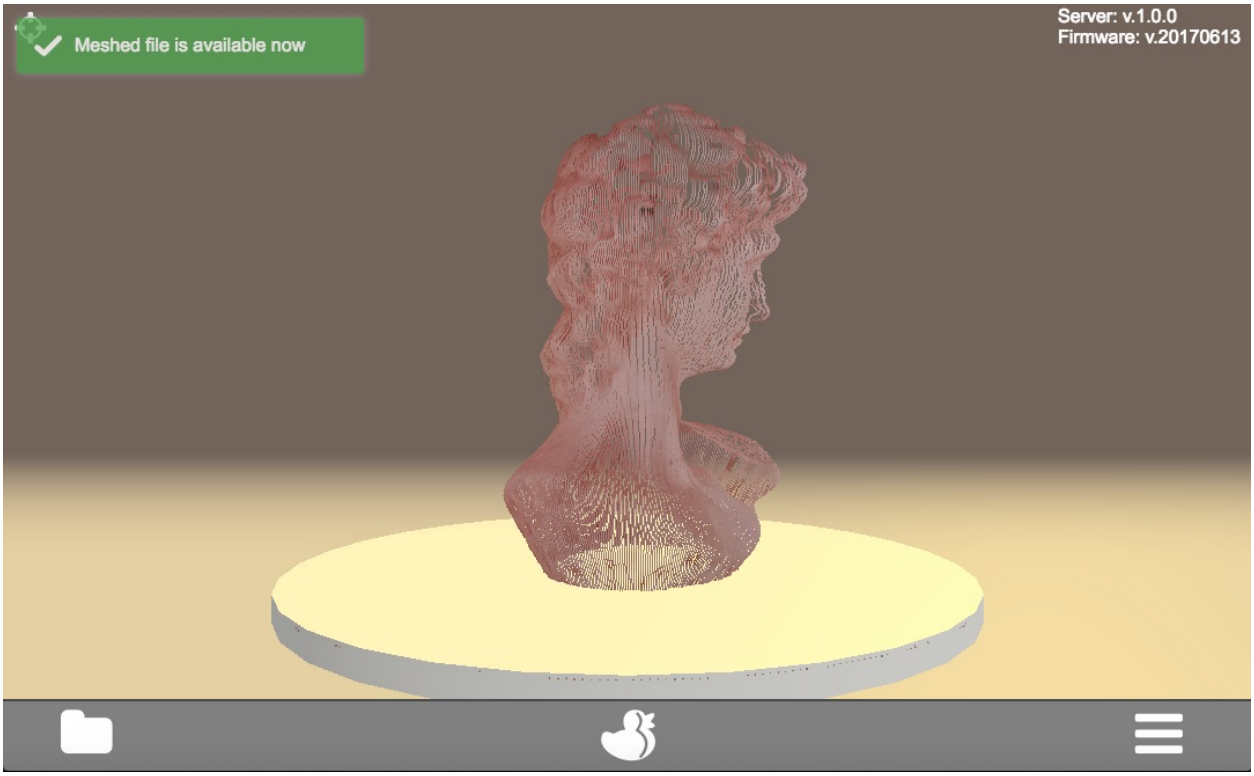
- Now select one of the Meshlab filters and the file format for the future mesh file.
- Click on “Start Meshing” to activate the conversion process.

The conversion starts and the main menu appears. A notification is displayed as well.

Note: Depending on the size and complexity of the scan file as well as the type of selected filter the conversion process may take some time.



When the mesh-file is available a notification is displayed.

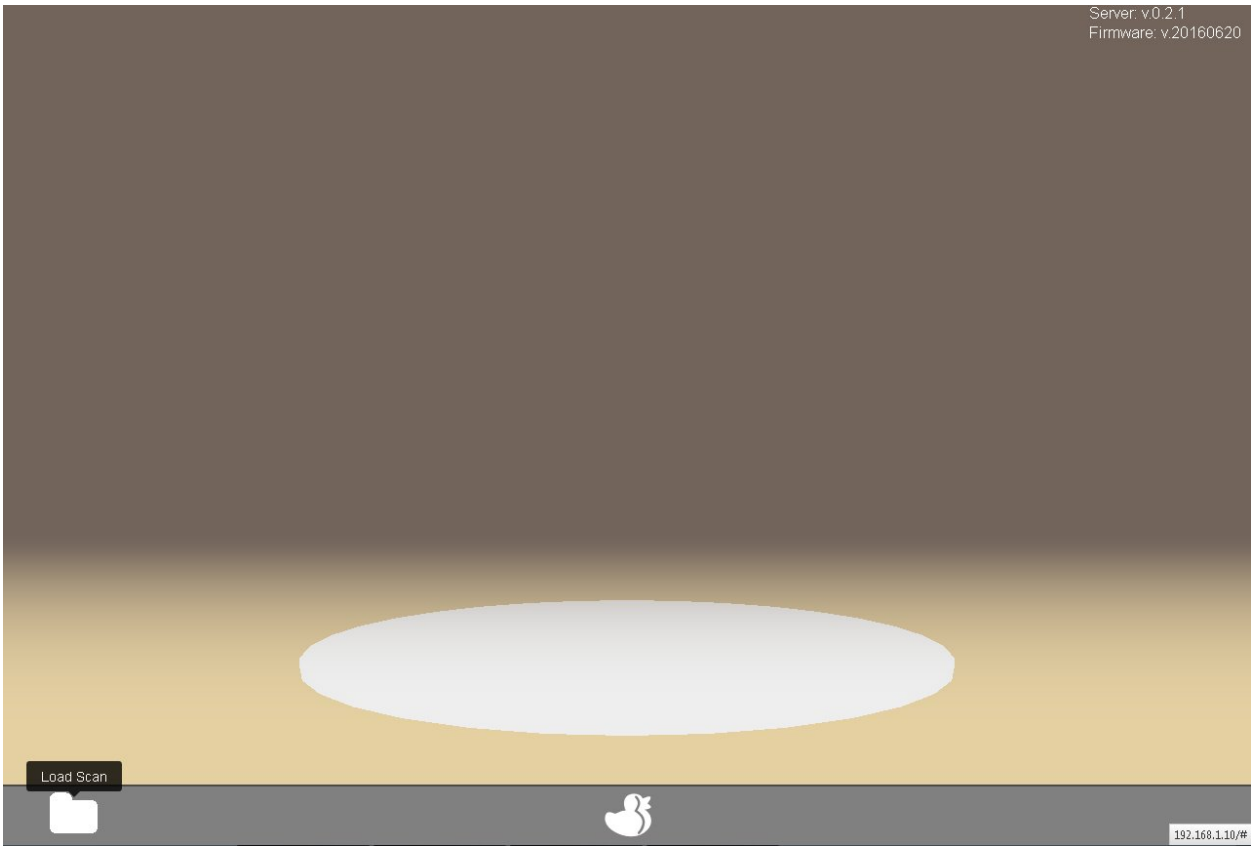


- Again open the options menu. Another index card for the mesh-file has been added.
- Click on the mesh-file index card.
- You can now click on the download-icon to download the mesh-file to your computer or click on the trashbasket icon to delete the mesh-file.

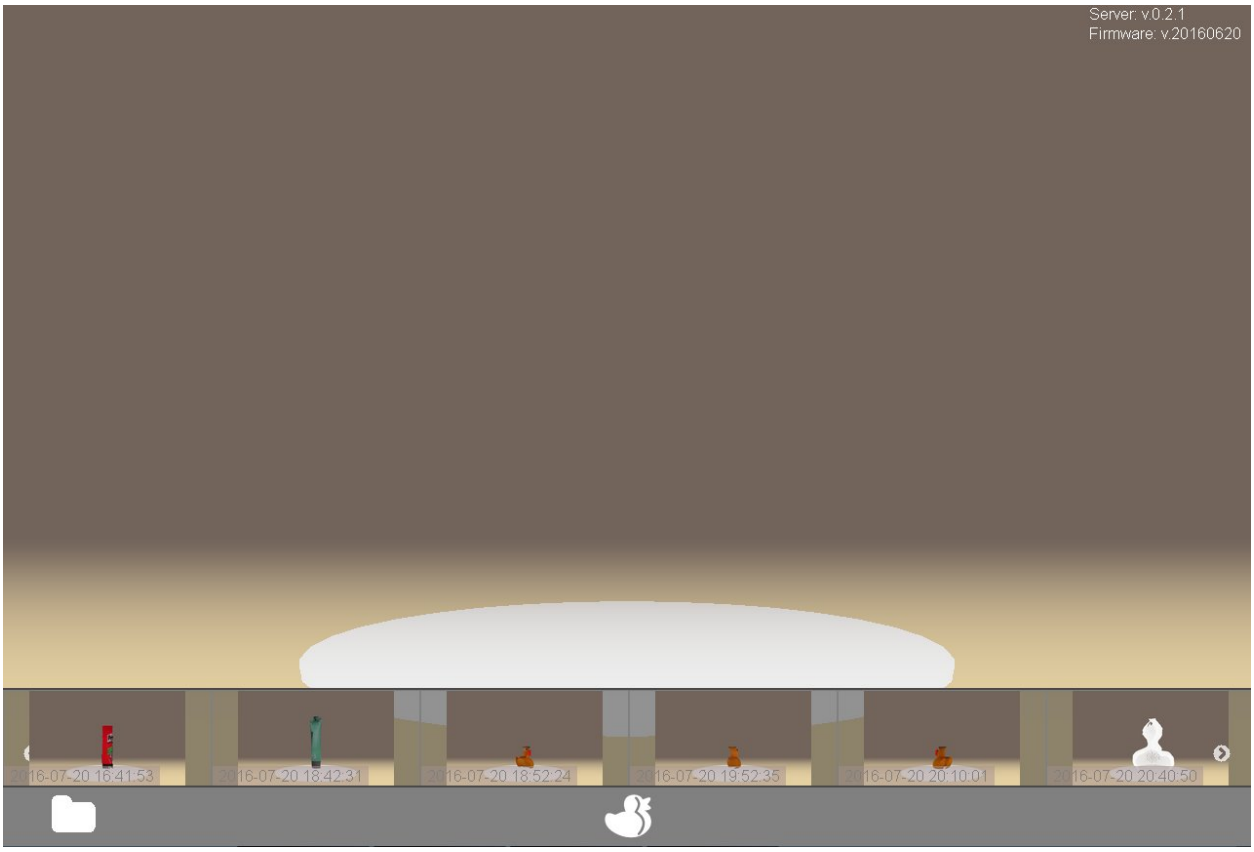
File Operations

- Load files

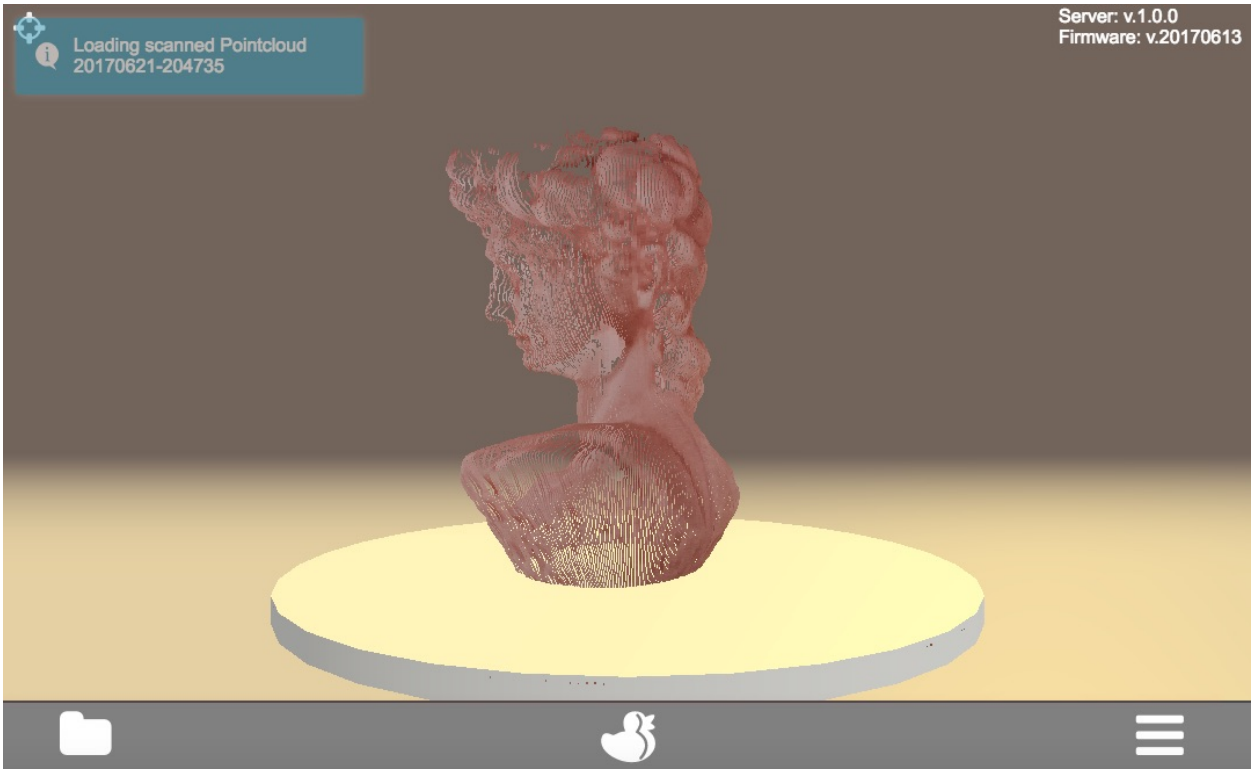
A scan result which has been saved to the FabScanPi memory previously can be reloaded. Go to the main menu and click on the folder-icon at the left side of the menu bar.



- Scroll through the displayed file inventory and click on the icon of the wanted file.



Now the selected file will be loaded which may need some time. After the loading process is finished a notification will be displayed.

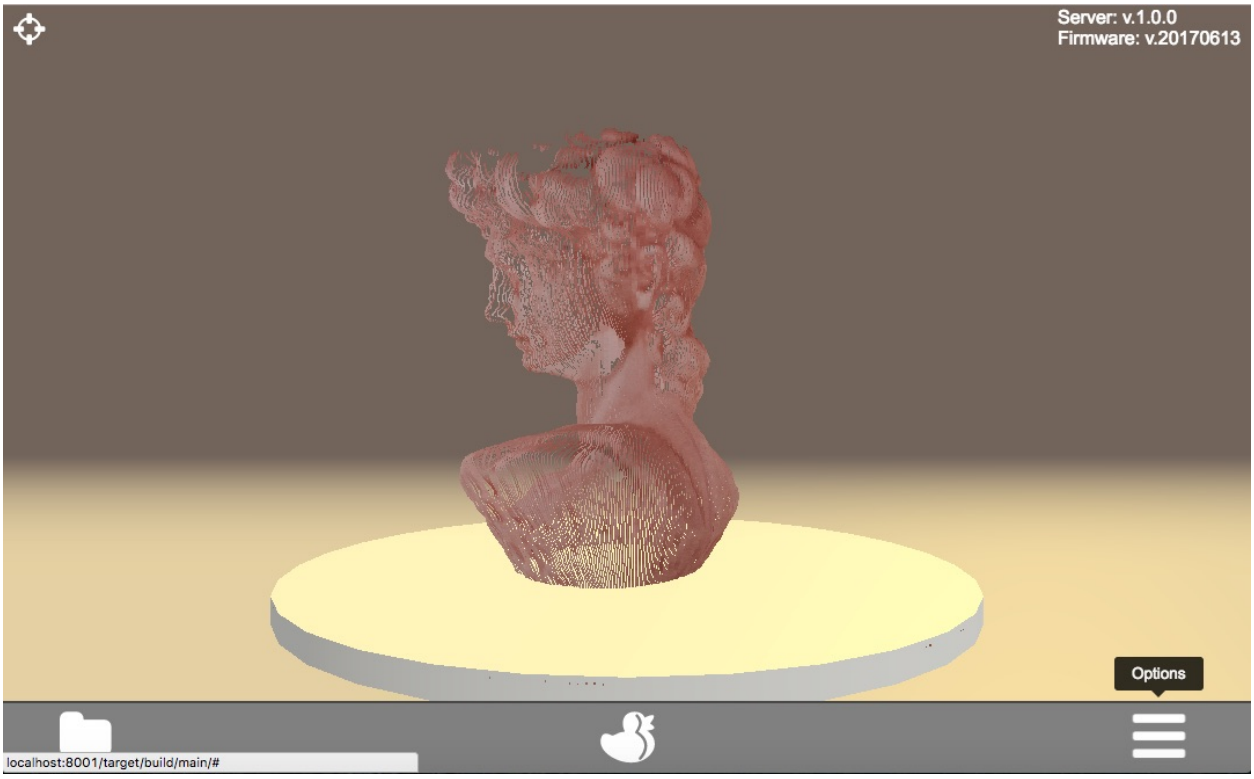


Delete a Scan

- Delete files

Delete a scan-file

A scan result which has been saved to the FabScanPi memory previously can be deleted. To do that it must be loaded and displayed on the virtual turntable in the main menu. -Click on the options-icon on the right side of the menu bar.

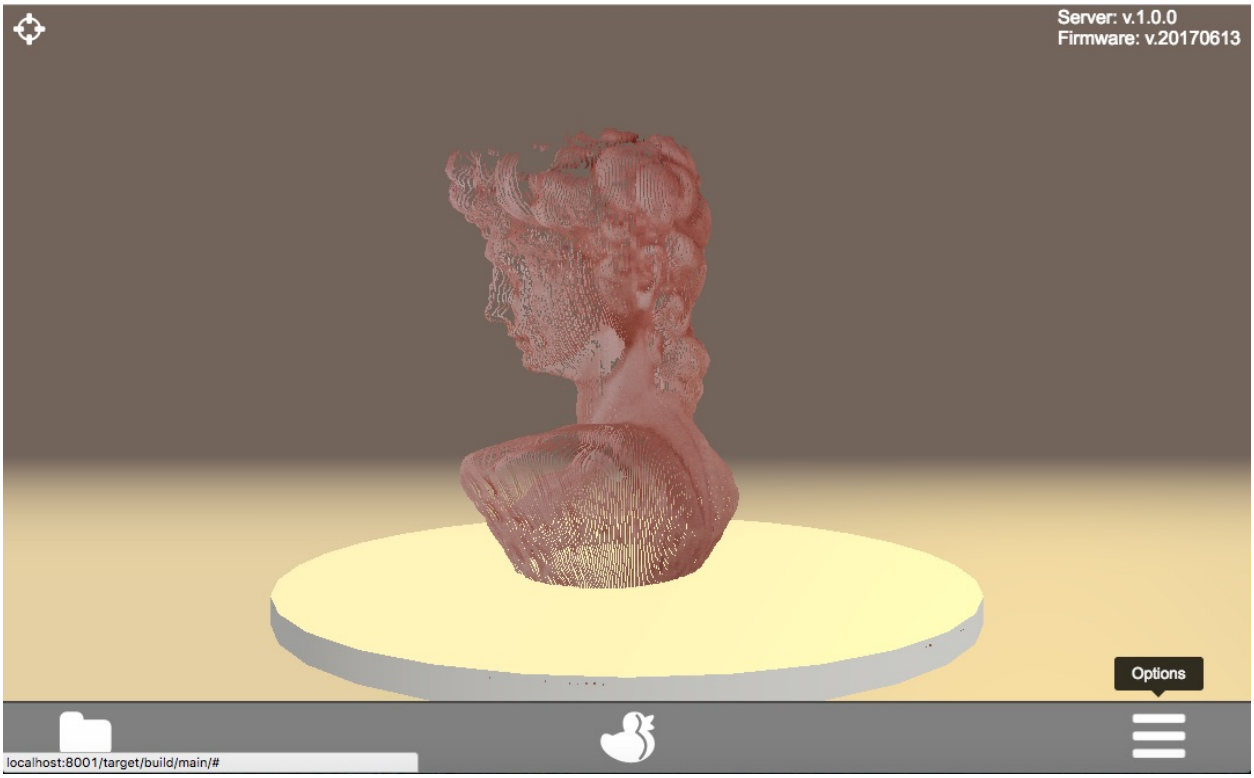


Click on the wastebasket-icon to delete the scan-file.

Note: By deleting a scan file the corresponding mesh file (if available) will be deleted instantly.

Delete a mesh-file

Note: If a mesh file is available a second slide for the mesh file will be displayed.



By selecting the mesh slide and clicking on the wastebasket-icon the mesh-file can be deleted separately.

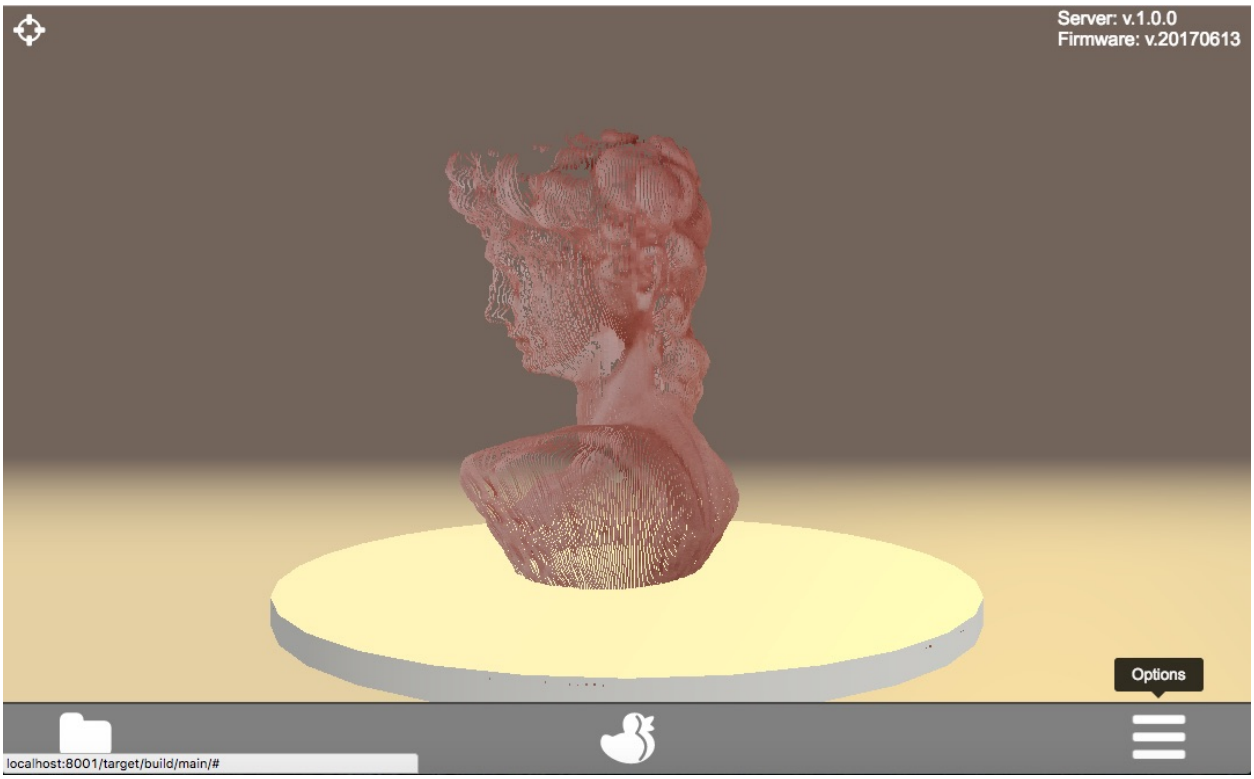
- Download Files It is possible to download generated files (either scan- or mesh-files) from the FabScanPi via the web-based user interface.

Download Results

Download a scan-file

Note: Before you can download a file it must be [loaded](#loadFiles) and displayed on the virtual turntable in the main menu.

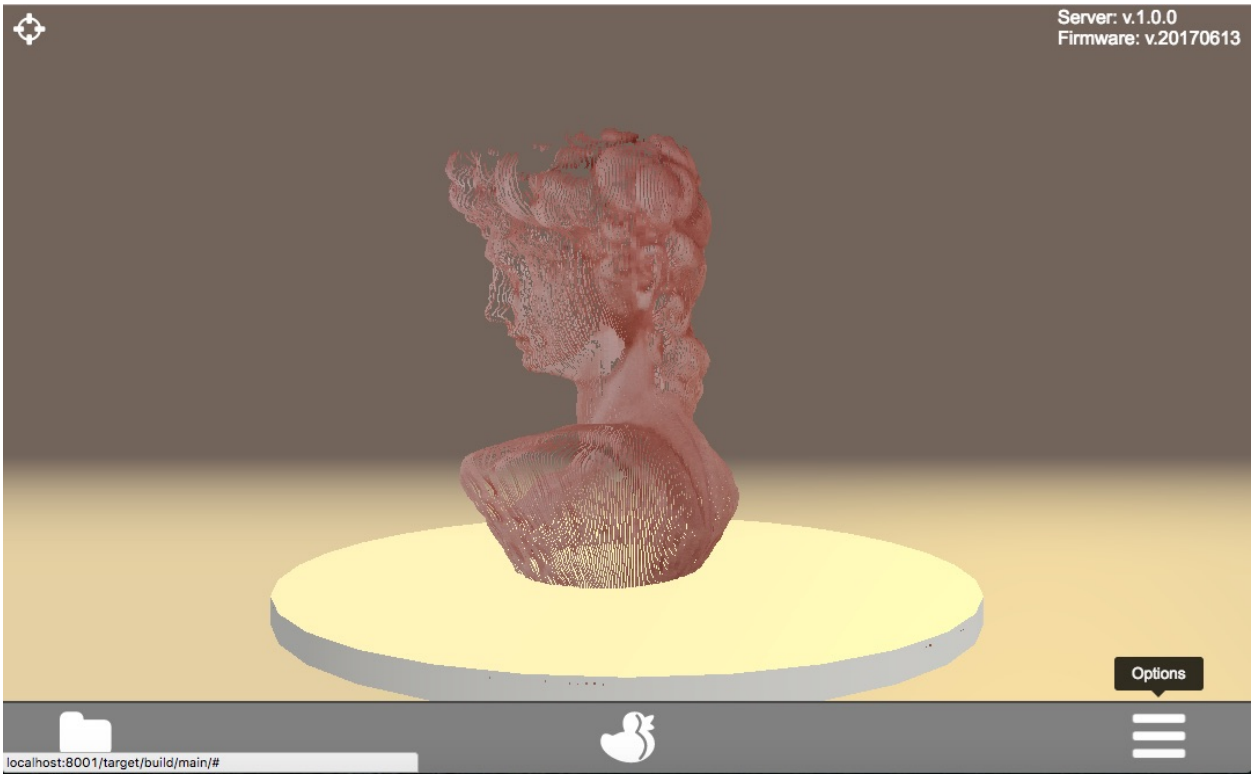
- Go to the main menu.
- Click on the options-icon on the right side of the menu bar.



- Click on the download-icon to download the mesh-file
- A download message (depending on the used web-browser) will be displayed

Download a mesh-file

Note: If a mesh file is available a second slide for the mesh file will be displayed.



- Select the mesh slide
- Click on the download-icon to download the mesh-file
- A download message (depending on the used web-browser) will be displayed

Troubleshooting

Can not connect to Arduino. Is the Arduino connected to the USB port. Check the port in your fabscanpi-server configuration file. Try another port e.g. /dev/ttyUSB0. More information about changing the configuration can be found in "FabScan Pi Configuration" section.

Camera is not connected. Check the cable from the Raspberry Pi to the camera module. Be careful the cable can be very fragil. Try another camera application for checking camera functionality e.g. [raspistill](https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspistill.md).

Prepare for Calibration

Before you can start with your first scans you must perform a calibration. That is necessary because every Scanner housing is a bit different. By calibrating your scanner the software will get the exact parameters of your FabScanPi. You will have to do the calibration only once, the gathered data will be stored. But we recommend to do a calibration every time when the FabScanPi has been shipped, modified or when the scan results look distorted.

Note: A successfull calibration is the key for getting good scan results.

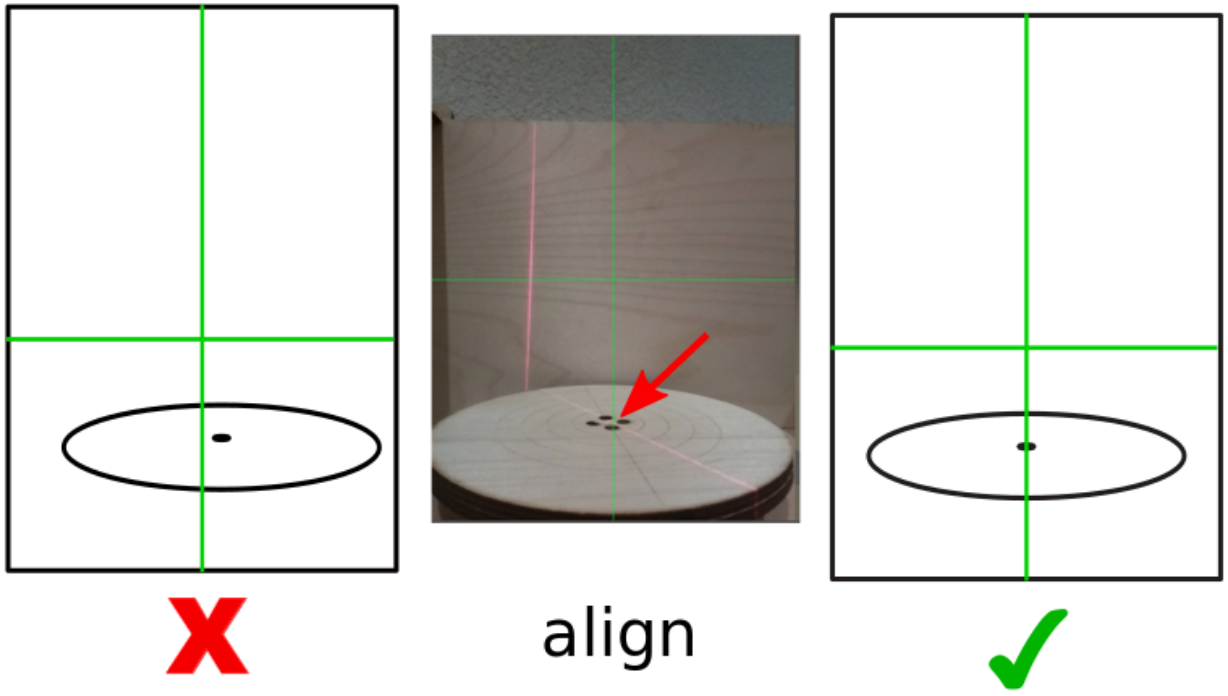
First you need to be sure, that some requirements fit. Especially when you are planing to use two instead of only one laser. The following simple adjustments can improve your calibration quality a lot.

Camera Alignment

Horizontal Camera Position

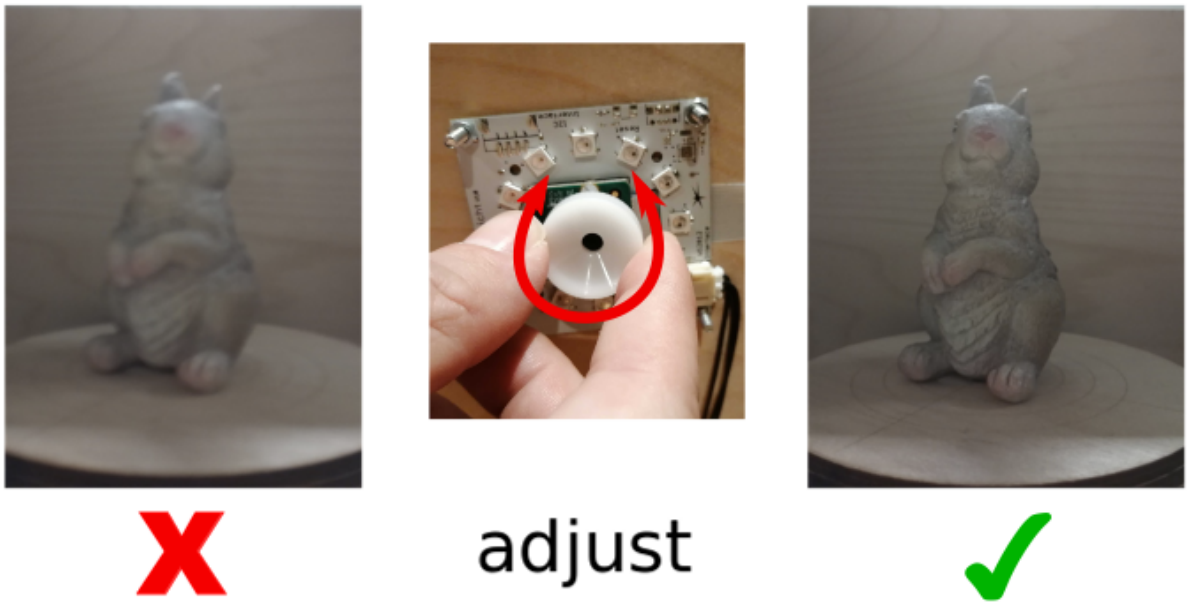
The camera should be alined to the center of the turntable. You can use the green cross overlay in the image as a reference for the center of the image. You can push gently the camera lens into the direction you want.

Note: The camera module is glued to the pcb. It can be different form camera to camera.



Camera Focus

Mosty the Raspberry Pi camera is not in focus by default (espesically V2) There are several ways to fix that. The easiest option is to use the white piece of plasic which is delivered wiht almost every Raspberry Pi cam since a few years. You can stick it on top of the cam and adjust the focus by turning it. It has a hole in the middle so that you can see changes in realtime if you are running a live stream.

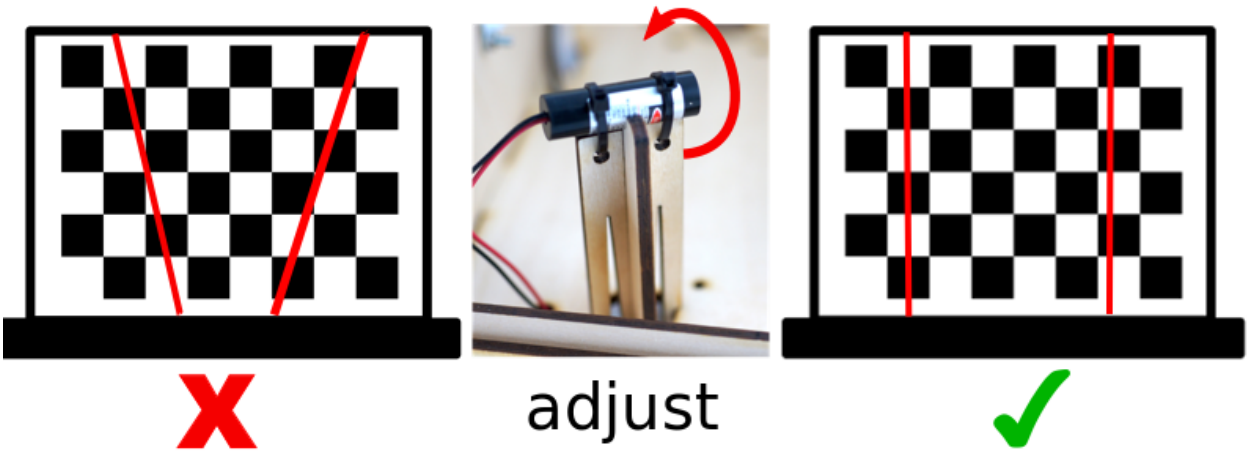


Note: Build your own tool if you don't have the adjustment ring anymore. There are a lot of 3d printable wrenches for the Raspberry Pi camera on thingiverse. https://www.thingiverse.com/thing:1570865

Laser Alignment

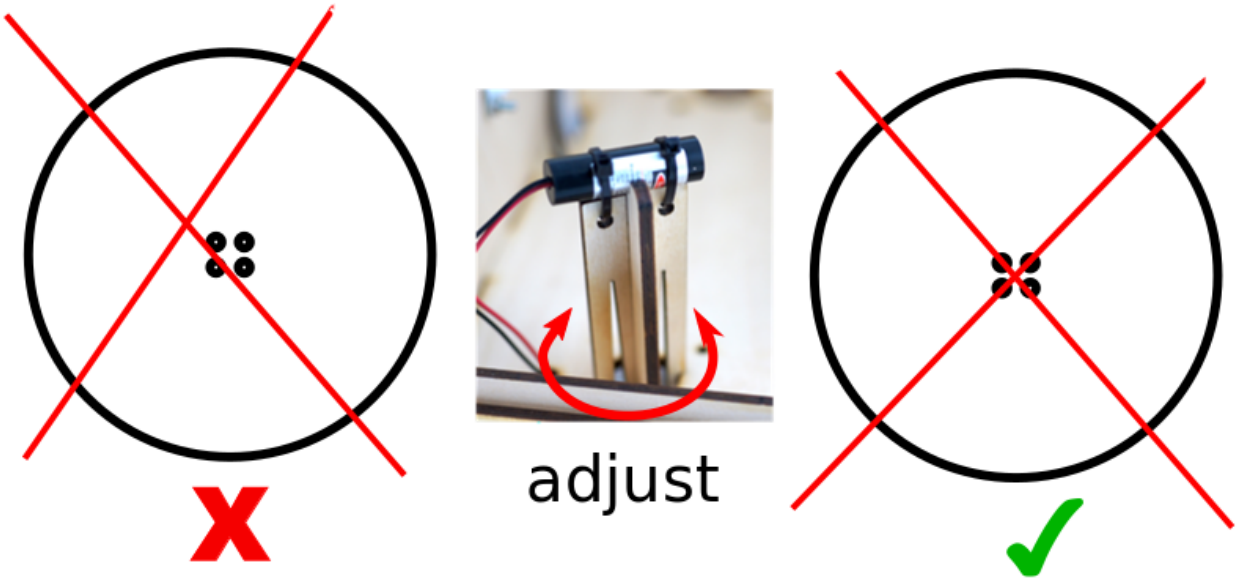
Vertical Laser Alignment

The lasers should hit the calibration sheed in a vertical direction. As close to 90 degree as possible. You can rotate the laser module to adjust the vertical laser alignment.



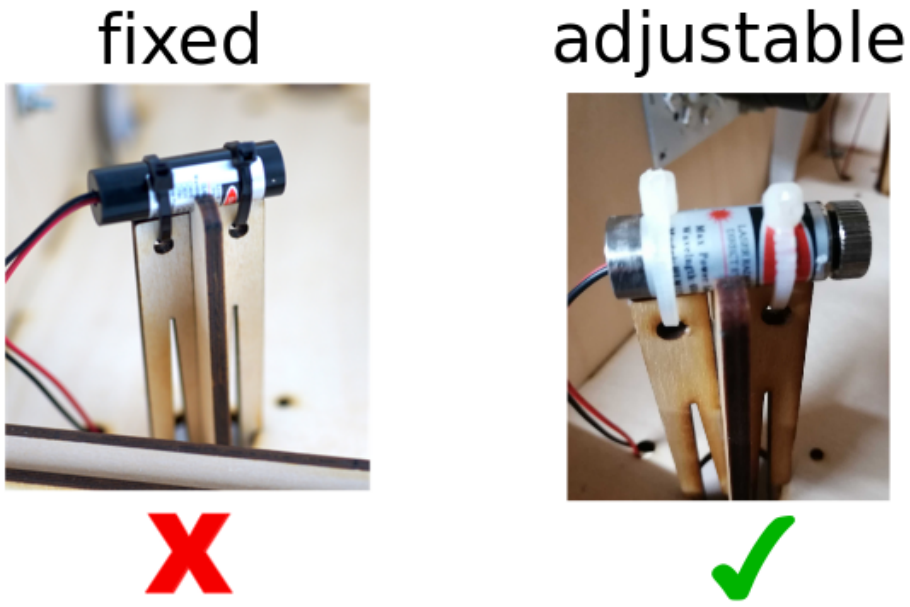
Hozizonatal Laser Alignment

The lasers should cross the center of the turntable. You can rotate the laser mount to adjust the laser poistion on the turntable.

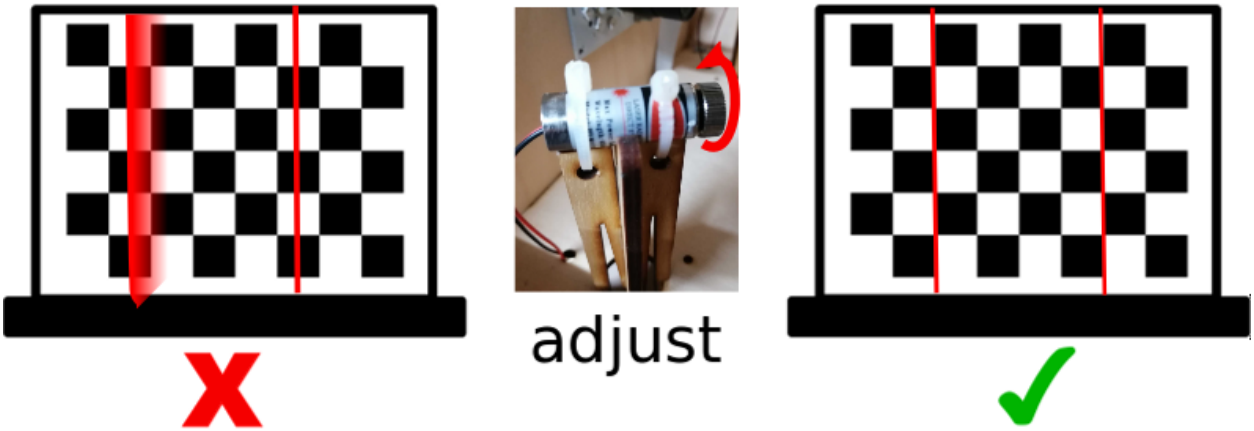


Laser Focus

This step depends on your laser. If you are using a laser with adjustable focus (right image), you should follow this step. When your laser has a fixed focus (left image) continue with *Start Scanner Calibration*.

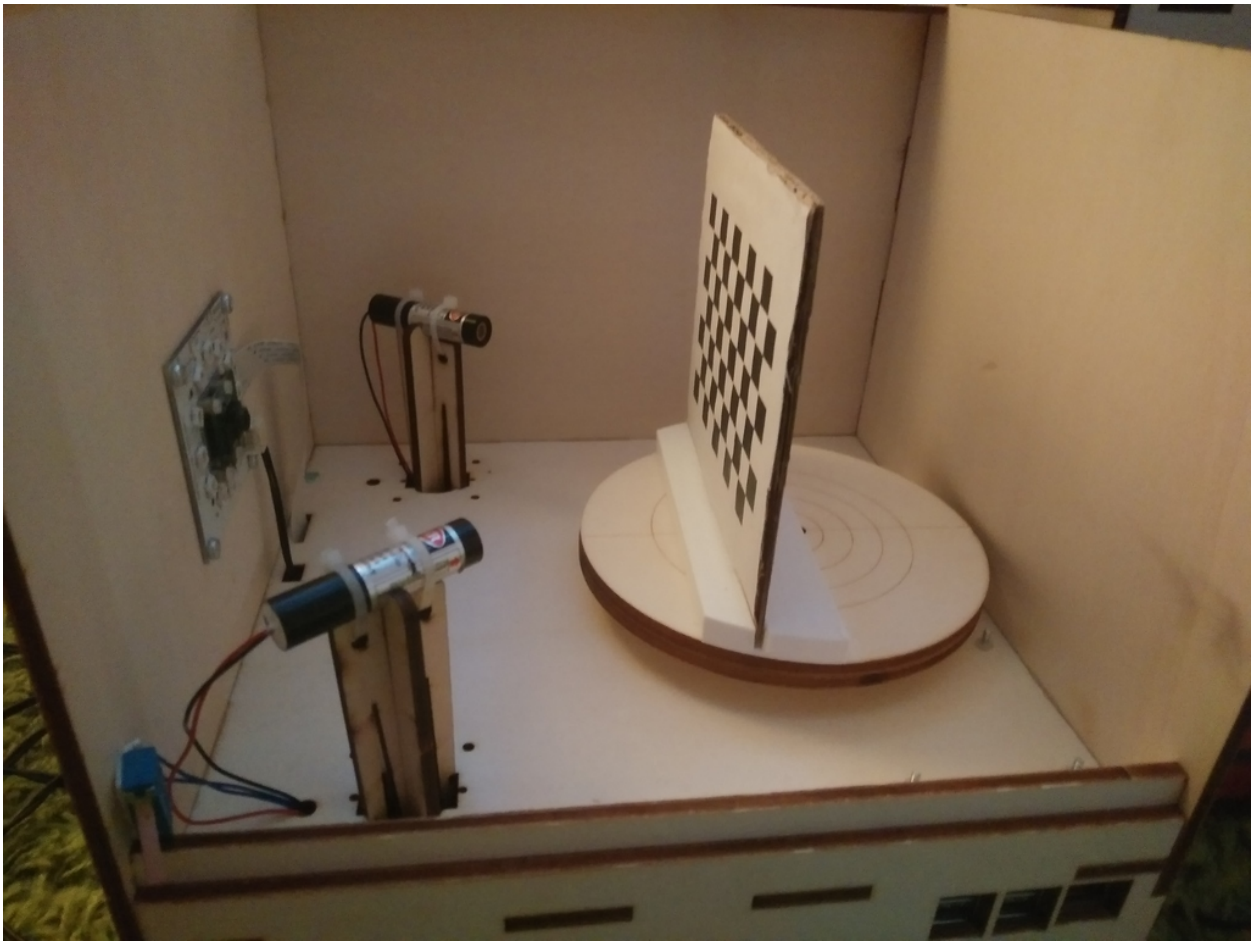


Turn the focus ring on the laser until you get a the sharpes possible line. If you are using two lasers, do it for both.



Start Scanner Calibration

The calibration will be done by scanning an calibration sheet with a specific pattern which must be placed on the turntable. The pattern must face the camera:



Option 1: You can can build your own calibration tool from cardboard. Print the calibration sheet onto paper and glue it to cardboard. Build a stand of another piece of cardboard. Maybe you have to fix the calibration sheet on the turntable by using some adhesive tape.

Option 2: You have an 3D-printer available and can print out our calibration sheet holder. There's also a different calibration sheet for external holder use.

Note: To get a perfectly flat and stable surface glue the calibration sheet on a piece of thick pcb (printed circuit board).

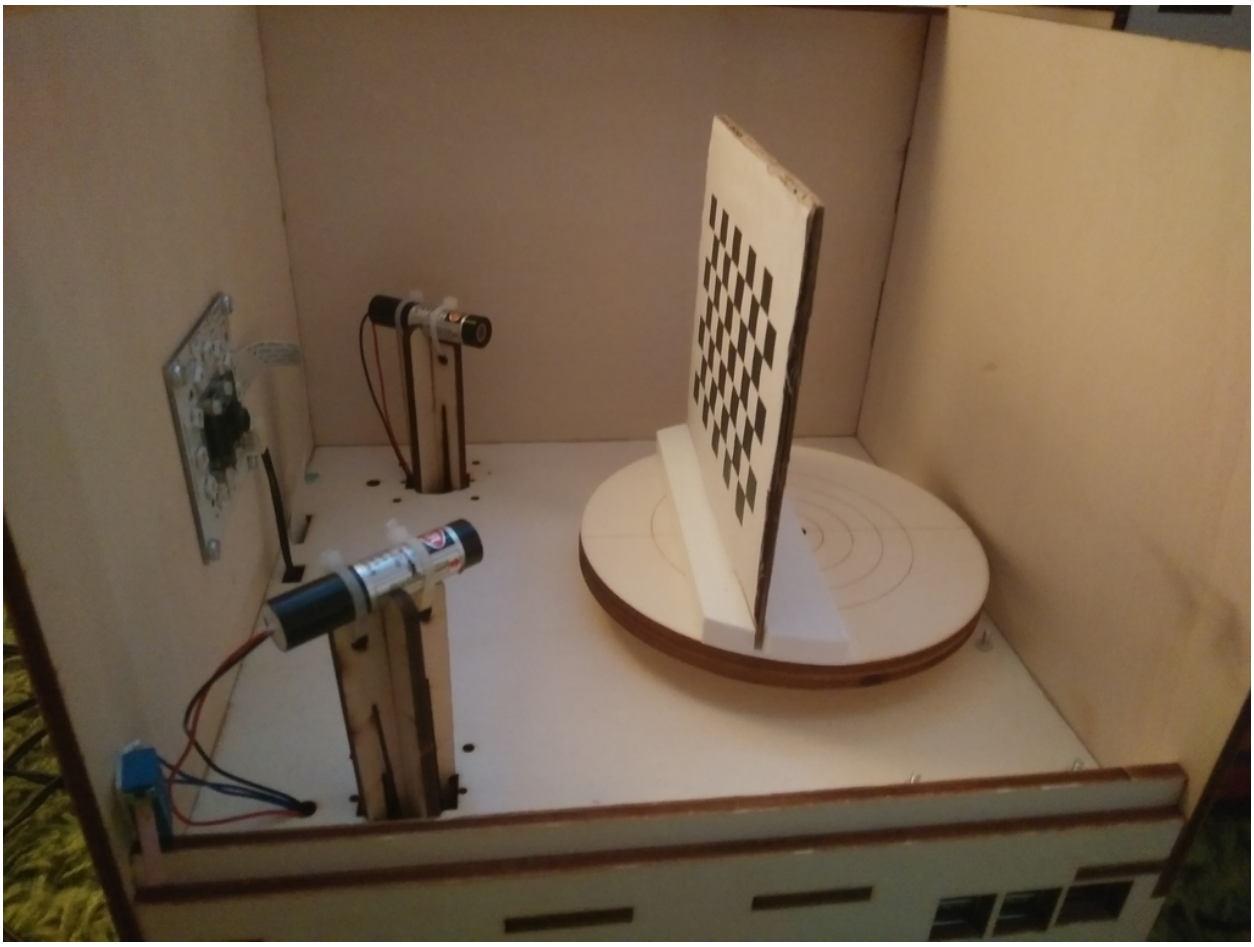
Download calibration mount and sheet!

Version V1 and V2 have the same size. Both will fit. But V2 is more printer friendly.

Note: Please check if your print of the calibration sheet has the correct scale. Each black square should have the same side length as stated on the calibration sheet (e.g.11 mm (1,1 cm). If the size is not correct please correct the parameter "pattern square size" in the configuration file.

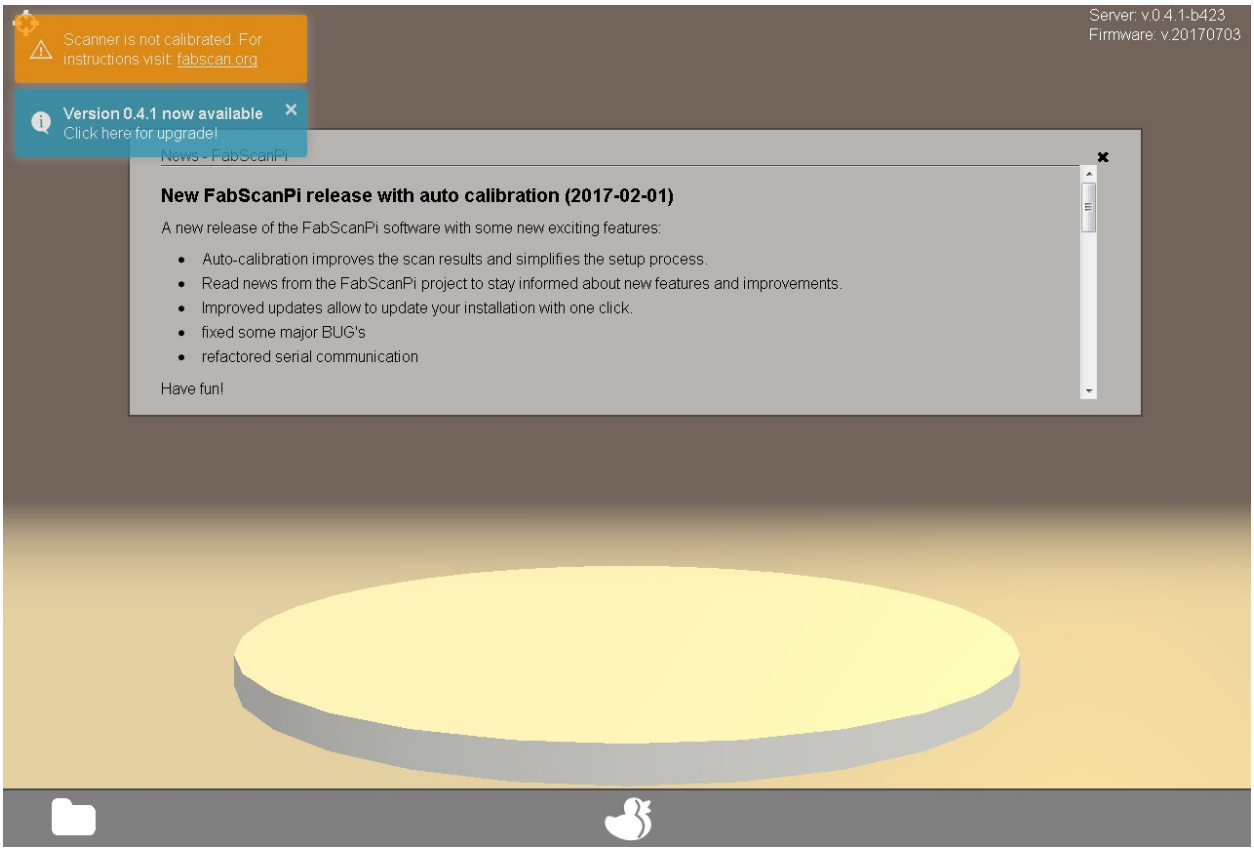
You can find all details in the chapter How to Edit the Config File .

Place the calibration sheet on the turntable facing the camera. See here

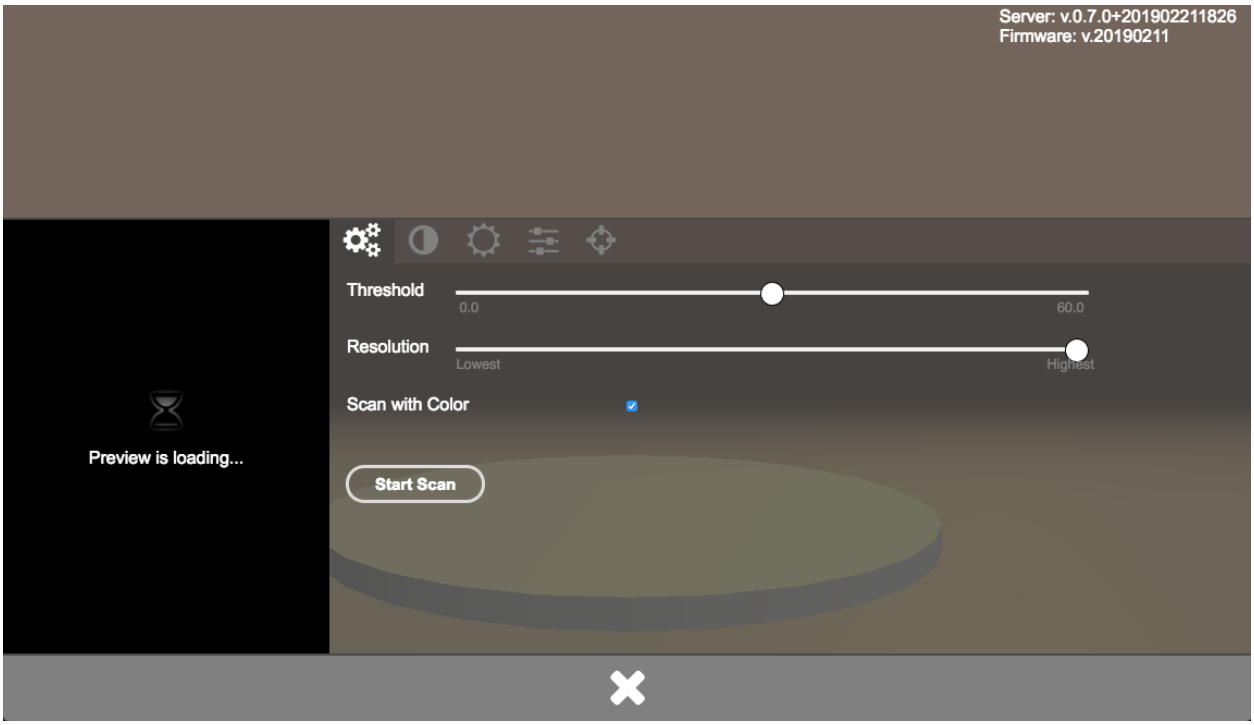


Please close the box to avoid external light sources to impact the calibration process. Start your browser and call http://ip-of-your-raspberry-pi:8080 which will open the user interface.

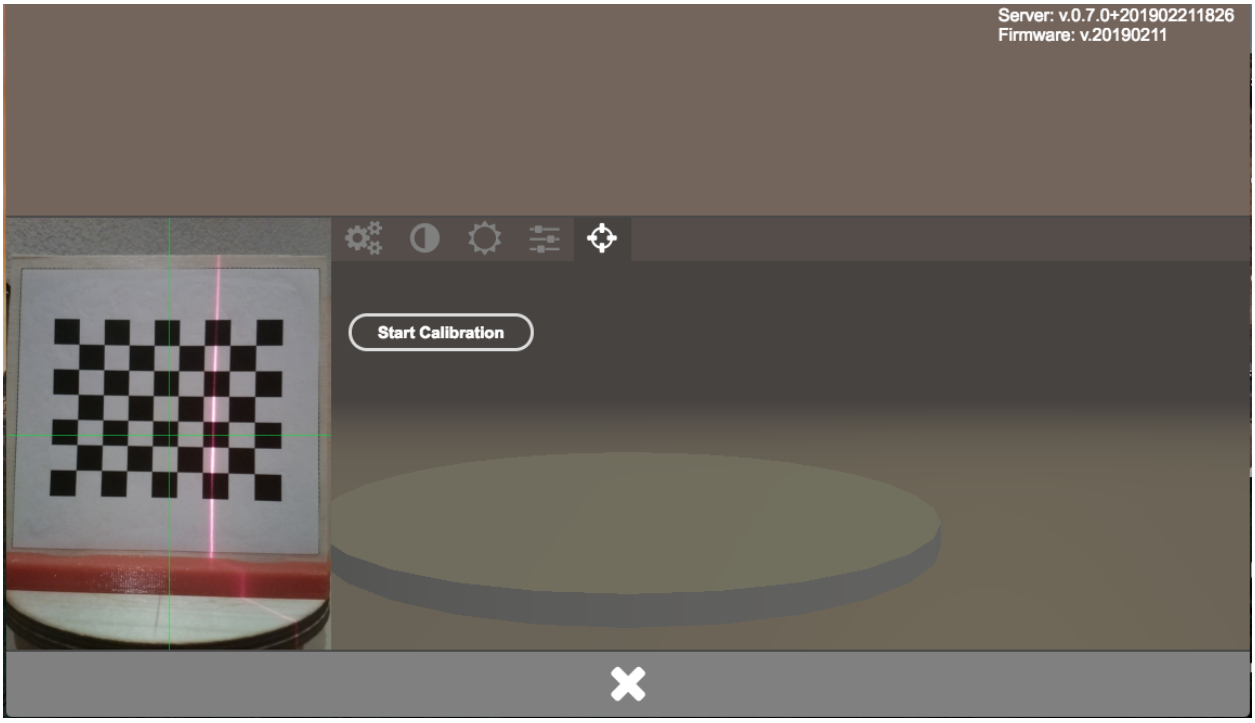
When you start your FabScanPi for the first time you will see an information in the upper left corner that you have to perform a calibration.



Click on the duck icon to open the settings dialog.

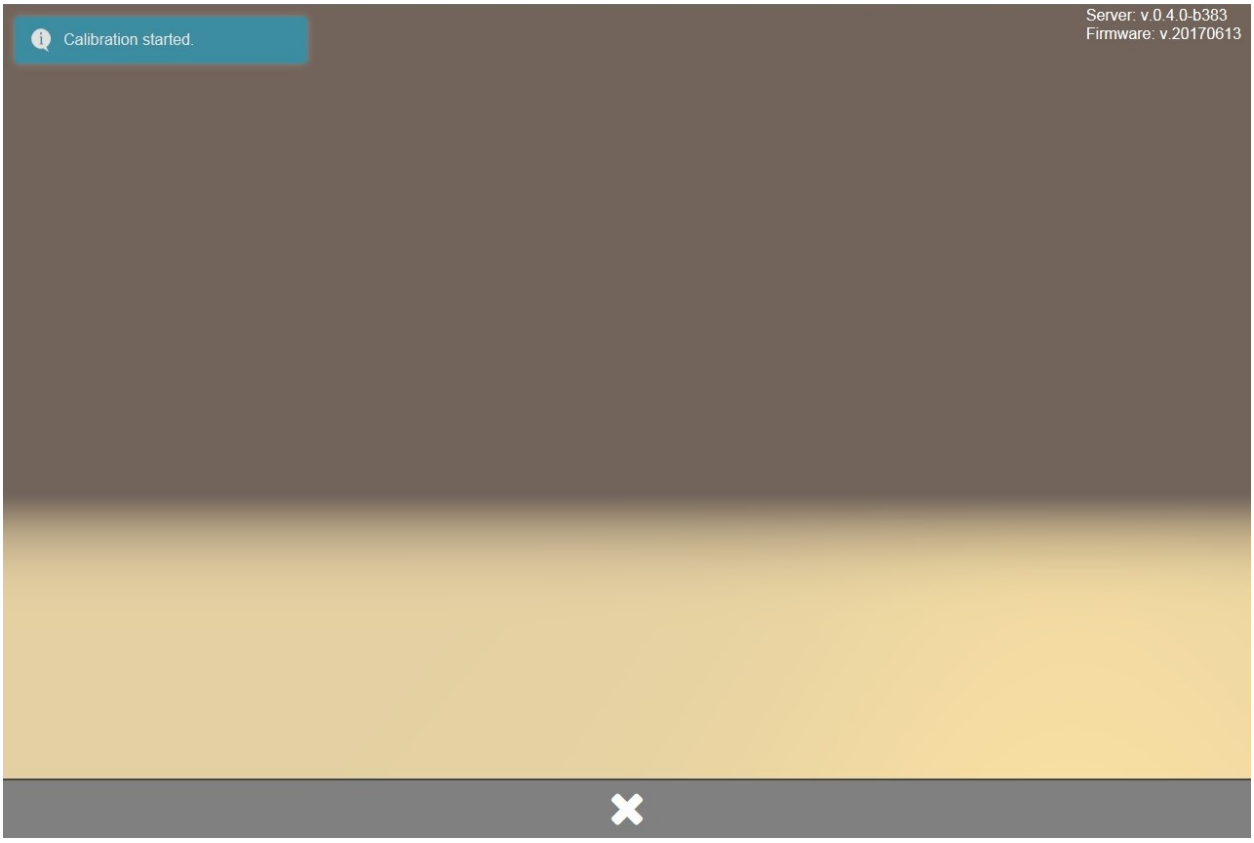


Select the calibration tab. It is the one with the crosshair icon.

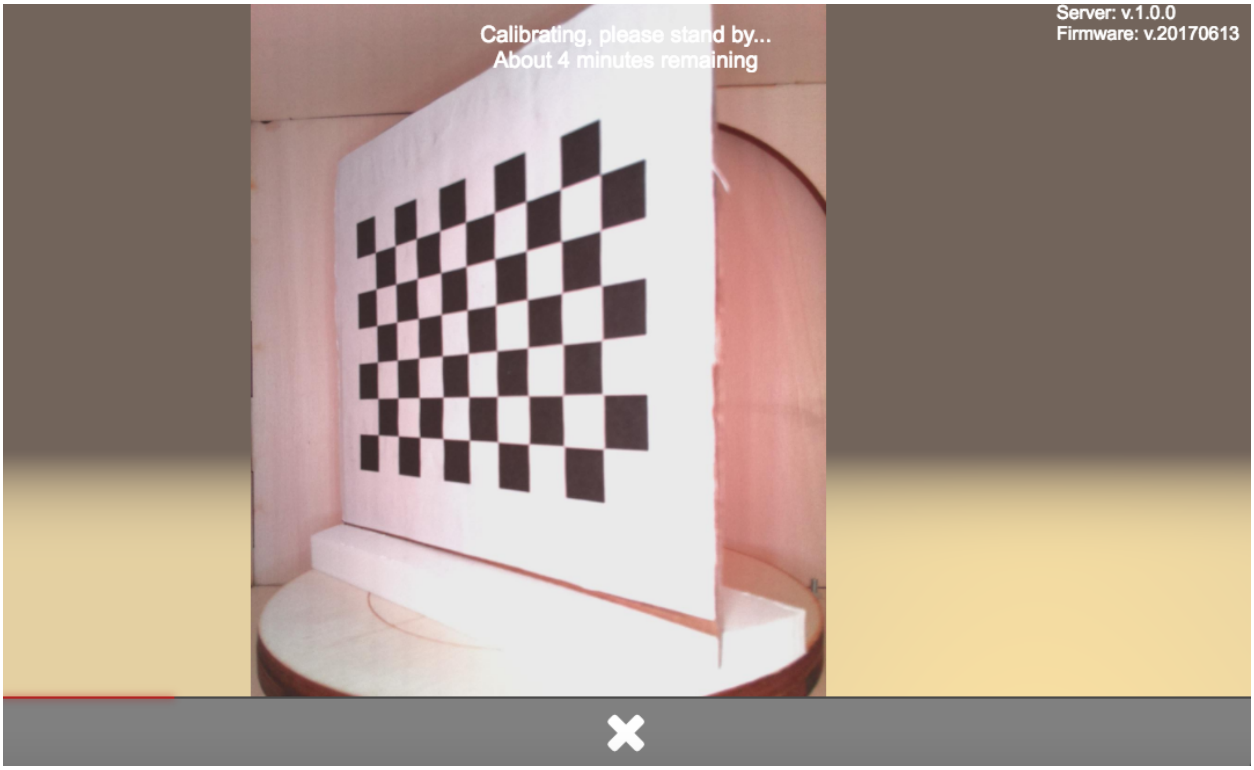


Click on the “Start Calibration” Button. The calibration starts immediately.

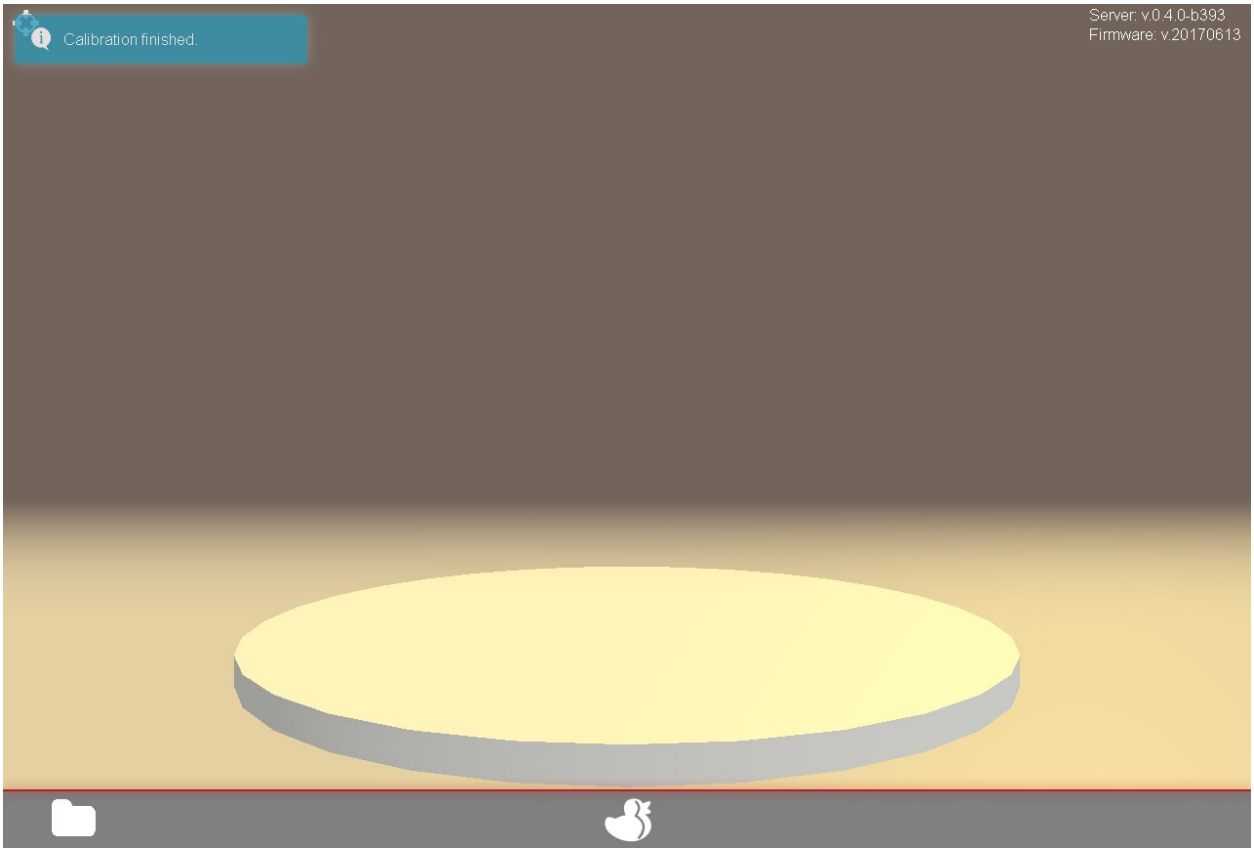
An information “Calibration started” will be displayed in the upper left corner.



The calibration may need several minutes. During the calibration an information about the remaining time is displayed.



When the calibration was finished successfully another notification will appear for a short time in the upper left corner.



Now you can remove the calibration pattern sheet from the turntable and start your first scan.

Note: If your scan results look not like they should, probably distorted or skewed please check the calibration data in the configuration file. You can find details about the different relevant parameters in the [Config File Values](#config-file-values) section of this documentation.

Backend: FabScanPi-Server (Python)

TODO

Frontend: Angular (JavaScript)

TODO

Firmware: Arduino (C++)

TODO

Image: Raspbian Build Script (Bash)

TODO

Code Contribution

Please take a moment to review this document in order to make the contribution process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of the developers managing and developing this open source project. In return, they should reciprocate that respect in addressing your issue or assessing patches and features.

Using the issue tracker

The issue tracker is the preferred channel for [bug reports](#bugs), [features requests](#features) and [submitting pull requests](#pull-requests), but please respect the following restrictions:

- Please **do not** use the issue tracker for personal support requests (use FabScan Forum or Slack).
- Please **do not** derail or troll issues. Keep the discussion on topic and respect the opinions of others.

Bug reports

A bug is a demonstrable problem that is caused by the code in the repository. Good bug reports are extremely helpful - thank you!

Guidelines for bug reports:

1. **Use the GitHub issue search** check if the issue has already been reported.
2. **Check if the issue has been fixed** try to reproduce it using the latest *master* or development branch in the repository.

3. **Isolate the problem** create a reduced test case and a live example.

A good bug report shouldn't leave others needing to chase you up for more information. Please try to be as detailed as possible in your report. What is your environment? What steps will reproduce the issue? What browser(s) and OS experience the problem? What would you expect to be the outcome? All these details will help people to fix any potential bugs.

Example:

> Short and descriptive example bug report title > > A summary of the issue and the browser/OS environment in which it occurs. If > suitable, include the steps required to reproduce the bug. > > 1. This is the first step > 2. This is the second step > 3. Further steps, etc. > > <*url*> - a link to the reduced test case > > Any other information you want to share that is relevant to the issue being > reported. This might include the lines of code that you have identified as > causing the bug, and potential solutions (and your opinions on their > merits).

Feature requests

Feature requests are welcome. But take a moment to find out whether your idea fits with the scope and aims of the project. It's up to *you* to make a strong case to convince the project's developers of the merits of this feature. Please provide as much detail and context as possible.

Pull requests

Good pull requests - patches, improvements, new features - are a fantastic help. They should remain focused in scope and avoid containing unrelated commits.

Please ask first before embarking on any significant pull request (e.g. implementing features, refactoring code, porting to a different language), otherwise you risk spending a lot of time working on something that the project's developers might not want to merge into the project.

Please adhere to the coding conventions used throughout a project (indentation, accurate comments, etc.) and any other requirements (such as test coverage).

Follow this process if you'd like your work considered for inclusion in the project:

1. Fork the project, clone your fork, and configure the remotes:

```
# Clone your fork of the repo into the current directory
git clone https://github.com/<your-username>/<repo-name>
# Navigate to the newly cloned directory
cd <repo-name>
# Assign the original repo to a remote called "upstream"
git remote add upstream https://github.com/<upstream-owner>/<repo-name>
```

2. If you cloned a while ago, get the latest changes from upstream:

```
git checkout <dev-branch>
git pull upstream <dev-branch>
```

3. Create a new topic branch (off the main project development branch) to contain your feature, change, or fix:

```
git checkout -b <topic-branch-name>
```

4. Commit your changes in logical chunks. Please adhere to these git commit message guidelines or your code is unlikely be merged into the main project. Use Git's interactive rebase feature to tidy up your commits before making them public.

5. Locally merge (or rebase) the upstream development branch into your topic branch:

```
git pull [--rebase] upstream <dev-branch>
```

6. Push your topic branch up to your fork:

```
git push origin <topic-branch-name>
```

7. **Open a Pull Request** with a clear title and description.

Important: By submitting a patch, you agree to allow the project owner to license your work under the same license as that used by the project.

Join the Community

FabScan.org

The offical home of the FabScan Project. It provides a blog with news, and a user forum.

Slack

Use the FabScan Slack channel for direct messaging and chat communication. This is the fastest way of getting support.

Signup here: https://fabscan.signup.team{]}(https://fabscan.signup.team

Facebook

Like our Facebook Page. https://www.facebook.com/Fabscan

Twitter

Tweets about FabScan can be found on Twitter. https://www.twitter.com/l_k3