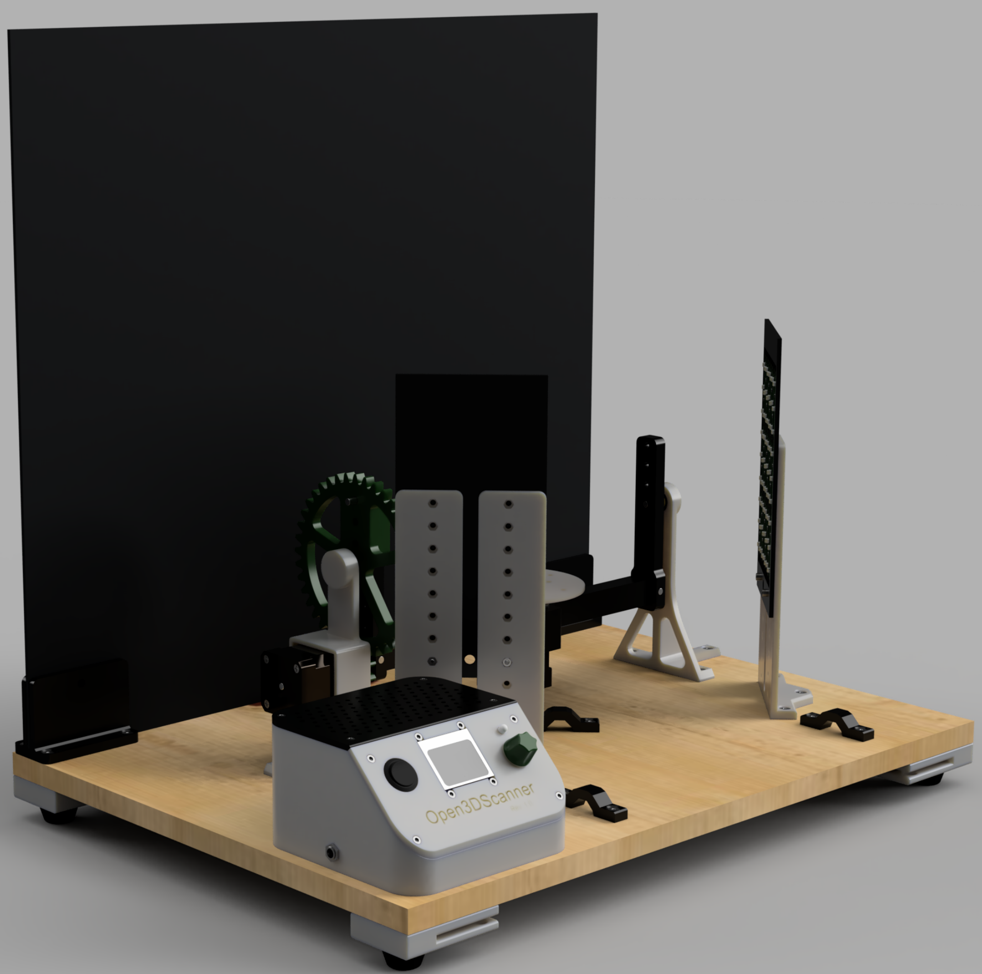
Open3DScanner

Manual — Rev. 1.0

Oct
26
2019



Nazrim

Table of contents

Version	Date	Note
1.0	2019/10/20	Initial release

Table 1 History of this document

License

Nazrim
https://www.myminifactory.com/users/Nazrim
https://www.thingiverse.com/Nazrim
https://www.hackster.io/Nazrim
https://github.com/nazzrim

1

Chapter

Preface

A 3D scanner is a good addition to a 3D printer.

Before a detailed examination of the subject is performed, it is necessary to clarify what is the meaning of the term 3D scanning.

> *3D scanning is the process of analyzing a real-world object or environment to collect data on its shape and possibly its appearance (e.g. colour). The collected data can then be used to construct digital 3D models.*
>
> — 3D scanning — Wikipedia

3D scanning can greatly simplify the process of making a printed copy of an object. For example, when a component is broken and a 3D printer should be used to make a replacement. Or if a physical object for which no 3D files are available should be modified.

While there are great 3D printers available for various budgets, the market for 3D scanners is not that big. At least the devices are not as common as 3D printers.

Similar to 3D printing there are several technological approaches for the realization of 3D scanners, which are connected with different efforts and costs and show differences in the quality of the results.

A particularly interesting approach for the average maker is photogrammetry. In photogrammetry a large number of photos from different viewing angles are taken from an object. Then a special software is used to create a 3D object from the images in a multi-stage process.[1]

Depending on the number and quality (illumination, sharpness, resolution, camera distance to object, …) of the images, an accurate 3D model of the physical model can be created.[2] Theoretically, results with an accuracy of 0.1 mm can be achieved.

1. A great introduction to the photogrammetry pipeline, including references to scientific papers is provided by AliceVision .

2. Further information on the factors which affect the quality of the final model are explained by PhotoModeler Technologies .

Since the cameras in smartphones (especially in high-end devices) are continuously being improved and nowadays deliver considerable quality, most people already have the most important tool for creating their own photogrammetry images.

The other important tool in the photogrammetry pipeline is the software that processes the captured images and calculates the 3D model. There are many different software solutions which have been developed for photogrammetry.[3] Fortunately, there are also several open source projects in this list.

3. An extensive list of software for photogrammetry, which highlights some features for each software, is provided by ALL3DP .

All that is missing is a computer (as powerful as possible) that runs the software. This shouldn't be a problem because most makers already have a home computer.

Thus for many makers all necessary tools are available to create 3D scans with photogrammetry. So the question arises why 3D scanning has such a low popularity compared to 3D printing. One possible answer to this question is that there is simply no need for it. Another possible answer is that the manual creation of the many photos that are needed will discourage most people because it is very time-consuming. Perhaps it is also because a 3D scanner is used much less often than a 3D printer, which is why one does not want to buy a 3D scanner, even if it is similarly inexpensive as a 3D printer.[4]

4. An overview of a large number of available 3D scanners and some of their features can be viewed at ANIWAA . The list contains photogrammetry devices as well as various other technologies, but it is possible to filter the displayed devices.

For this reason the Open3DScanner project was started. The goal of the project is to provide the maker community with a 3D scanner that delivers 3D scans in good quality while keeping costs as low as possible by relying on existing tools such as smartphones and computers. It should be possible to implement the entire project for less than $150.00 , provided the necessary tools are available and the required parts can be procured without extreme (e.g. shipping) costs.

In its standard configuration, the Open3DScanner offers a scanning area that includes a cylinder with a diameter of approximately 26 cm and a height of approximately 16 cm. This restriction can easily be circumvented to a certain extent by configuring the scans accordingly. Detailed information can be found in chapter 7. All parts are designed to fit on the print bed of an original Prusa i3 to allow the use of a variety of 3D printers.

This document serves as a complete reference for the Open3DScanner project. It contains all the necessary information that makers need to build their own Open3DScanner, modify the 3D scanner, or simply get detailed information about the project.

Before a detailed examination of the Open3DScanner, chapter 2 introduces other open source 3D scanners and compares them with the Open3DScanner. Chapter 3 describes the toolchain used to create and design the Open3DScanner and shows dependencies to other projects (e.g. software libraries). Chapter 4 contains a BOM, which contains all parts needed to build the Open3DScanner. In addition, the required tools for building the scanner are listed and hardware required for operation is described. Chapter 5 then describes in detail all the steps necessary to build your own Open3DScanner from the individual components. Chapter 6 contains a user manual which describes the use of the fully assembled Open3DScanner. Finally, chapter 7 presents various tips and tricks that make it easier to create successful 3D scans using photogrammetry.

Thus, depending on the interests of the reader, not all chapters are equally interesting. However, the structure of the document should allow the chapters to be read individually and independently of each other. The necessary cross-references can be found in the necessary places.

There is no schedule for future development of this project. They are based on my needs and the needs of the Open3DScanner users.

The Open3DScanner has been published in various communities to reach a wider audience. The center of development and secure reference point for the latest version of the project is the Open3DScanner repository .

2

Chapter

Related OSS Projects

This chapter looks at other projects from the open source software and open source hardware community which have the goal to develop their own 3D scanners ! . In some places comparisons to the Open3DScanner are given and influences that the projects have on the Open3DScanner are pointed out.

! Important Notice

It should be noted that the list of projects considered in this chapter does not claim to be exhaustive. In particular, there are various variations for the presented projects and the projects itself may already be a variation of an existing project.

In the sections 2.1 and 2.2 other photogrammetry scanners are introduced while in the remaining sections 3D scanners based on other technologies are introduced.

2.1 OpenScan

First of all lets take a look at the project that had the biggest influence on the Open3DScanner, because it served as inspiration for it. The Open3DScanner is only an alternative realization of Thomas Megel's project:

OpenScan

OpenScan is based on automatically rotating the object to be scanned on two axes (X and Z) while automatically shooting the photos.

Figure 2.1 The OpenScan 3D Scanner

The OpenScan project, like the Open3DScanner, relies on existing cameras that are connected to the scanner. One possibility that OpenScan offers, but was omitted from the Open3DScanner, is the usage of various SLR cameras for 3D scanning. The SLR camera is connected to the scanner via an infrared remote shutter which is connected to the scanner via an optocoupler. This option was removed for the Open3DScanner, because I don't own a SLR camera, nor do I plan to buy one. Furthermore, I am convinced that the quality of modern smartphone cameras is sufficient for the production of good quality 3D scans.

A feature of the Open3DScanner that OpenScan does not provide is the possibility to connect LED lights directly to the 3D scanner and let the hardware control them during the scanning process.

While there is no information about the applicable license on the project's homepage, the OpenScan Thingiverse project indicates that the project is published under the CC-BY-NC 3.0 license.

Scans with the OpenScan 3D Scanner are performed fully automatically after the settings for the respective scan have been selected. It is possible to configure how many images are taken per rotation of the z-axis and by which angle the scanner should rotate on the x-axis. In addition, it is possible to determine at how many positions on the x-axis a stop should be made, which in turn results in a complete rotation of the z-axis.

In addition, there is a setting to adjust the time the scanner stops for each photo. This is important to ensure that the camera can refocus if necessary and avoid blurry shots.

> **Note**
>
> Although the Open3DScanner was developed on the basis of the OpenScan project, it is not a simple copy. After OpenScan motivated the development of the Open3DScanner, requirements were defined independently from the original project. All artifacts (3D models, firmware, BOM) were developed especially for the Open3DScanner.

2.2 3D Scanner Turntable

Another open source photogrammetry 3D scanner is the project 3D Scanner Turntable by Dave Clarke.

3D Scanner Turntable

It relies on the use of a smartphone camera and promises that only the filament costs ($30.00) will be incurred for the construction of the 3D scanner. In addition to the smartphone, matching headphones with buttons that allow the camera to be released are required.

In order to achieve the goal of a 3D scanner that is as inexpensive as possible, the project does not use additional electronics that automate scanning. Instead, it is necessary for the user to turn a crank that rotates the object to be scanned and ensures that the smartphone takes 55 photos every full turn.

Figure 2.2 The 3D Scanner Turntable

Unlike the Open3DScanner or the OpenScan, the object is only rotated on one axis during the scan, so it may be necessary to perform several scans and reposition the object each time. It is therefore necessary for the user to interact more strongly with the 3D scanner during use, but this is the only way to keep costs so low compared to other projects.

As with the OpenScan project, the captured images must then be processed with appropriate software in order to obtain a 3D model.

2.3 Ciclop

The Ciclop 3D Scanner is an ambitious project of Jesús Arroyo, published by bq and based on laser triangulation. In addition to the 3D scanner itself, the project also provides its own software (Windows, Linux, and Mac OS X) for performing the 3D scans.

Ciclop

Figure 2.3 The Ciclop 3D Scanner

Even though this project has no influence on the development of the Open3DScanner, it shall be introduced briefly here, as it is a wonderful open source hardware project for the creation of a 3D scanner, which provides detailed source information and documentation.

The result of a scan is a point cloud, which has to be converted into a 3D model with other software (e.g. Blender) before the model can be used further, e.g. for 3D printing.

The whole project is published under the CC-BY-SA 3.0 license as well as the GPL v2.

Unlike the previous projects, the Ciclop is not based on external hardware (like the camera of a smartphone), but is a complete project in itself, requiring only a PC for operation.

The core of the project is formed by a Logitech C270 HD webcam for creating the photos and two class 1 lasers, which are used to sample the object. The object to be scanned is positioned on a plate which is automatically rotated.

3

Chapter

Used Software, Tools, etc.

This chapter provides an overview of the software and tools used to implement the Open3DScanner project. Furthermore used artefacts (libraries, 3D models, ...) as well as the corresponding licenses are presented.

The aim is to create an as complete as possible list of the dependencies of the Open3DScanner in order to have all relevant license information at one central location.

Transitive dependencies, which result from the used artefacts, are excluded from the consideration. If you are interested, please refer to the documentation of the respective artifact, which is linked at the appropriate place of this document, if available.

3.1 Arduino

Since the heart of the Open3DScanner is an ESP32[1] and the development should be done with the Arduino IDE , one of the most important dependencies in the Arduino area is the Arduino core for ESP32 WiFi chip , which allows the use of ESP32 development boards with the Arduino IDE. This simplifies the development of the project considerably.

The Arduino IDE is released under the GPL license, while the included libraries are released under the LGPL license. The LGPL license is also applied to the Arduino core for the ESP32 WiFi chip.

1. Detailed information about the ESP32, including its data sheet, can be found on the Espressif ESP32 product page .

3.1.1 Libraries

In the following the libraries which were used for the implementation of the Open3DScanner and are not included in the Arduino IDE are described.

Library	Version	Author	License	Purpose
nokia-5110-lcd-library	2.0.0	platisd	MIT License	Driving the Nokia 5110 LCD.
StepperDriver	1.1.4	laurb9	not specified	Two-pin stepper motor driver library.
ESP32Encoder	0.1.5	madhephaestus	MIT License	Rotary encoder library using interrupts.
arduino-menusystem	3.0.0	jonblack	MIT License	Ddata structures for menu structures.

Table 3.1 Arduino Libraries used within the Open3DScanner

It can be seen that only a few external libraries are required in total, since a large part of the functionality is already available through the libraries provided by the Arduino IDE and the Arduino core for ESP32 WiFi chip.

In some places the display of the Open3DScanner shows other things as pure text, e.g. to show buttons. For this purpose, corresponding bitmaps were created, which were then translated into corresponding character arrays with the tool LCD Assistant ℹ from Radosław Kwiecień.

The license of the LCD Assistant is not specified by the developer on the project page.

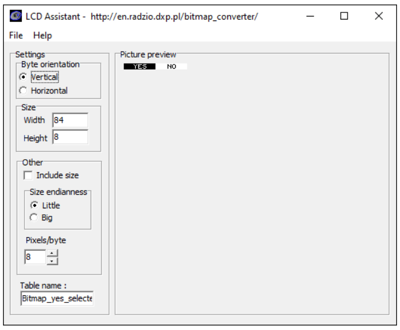


Figure 3.1 LCDAssistant GUI

> **LCDAssistant**
> The LCDAssistant is a useful tool that allows the easy transformation of bitmaps into character arrays, which can directly fed to the various LCD screens. It provides some configuration to match different LCDs.

3.2 Schematics and PCB Design

Although it is practical to test the necessary circuits on a solderless breadboard during development, this is not a long-term solution. This is especially true if you consider that the breadboard creates new sources of error.

In addition to the generally known problems, such as instability (in general, but also with e.g. vibrations) and high space requirements compared to a custom PCB, it must be noted that the individual tracks of the breadboard have high resistances and can introduce unwanted capacitances into the circuit[2].

During development it was not possible to use an LDO[3] regulator on the breadboard and to get a stable output voltage of 5 V. Instead it was necessary to outsource the low-dropout and its components to a prototype PCB to get a stable output voltage. Otherwise, it would not have been possible to test the circuit design, as spontaneous voltage drops led to crashes of the ESP32.

Figure 3.2 shows the circuit of the Open3DScanner on a breadboard. In

2. More information about unwanted side effects that can occur when using breadboards can be found on Hackaday and Breadboard Adventures .

3. Low Dropout

addition the prototype PCB with the LDO regulator is visible. It's easy to see that the circuit is chaotic and therefore difficult to maintain, debug, and develop. For the picture even cables have been removed: The stepper motors are not connected and the lights have been removed.

Due to the size of the components it was necessary to use two breadboards, which makes the whole construction extremely fragile. More than 50 cables were needed to build the circuit on the breadboards.

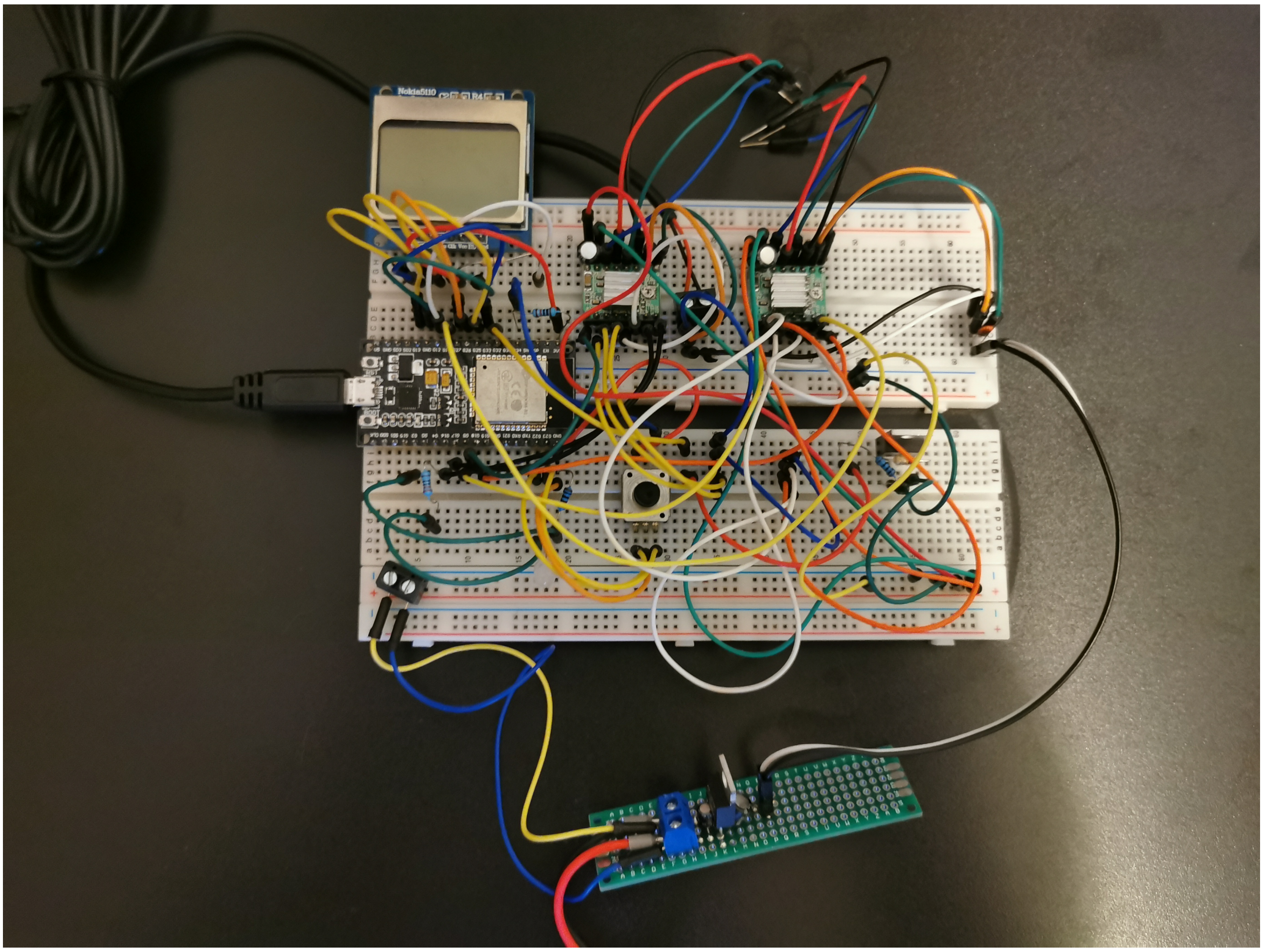


Figure 3.2 The Open3DScanner circuit on a solderless Breadboard

It quickly becomes clear that in the long run it is necessary to design and manufacture a special PCB for the Open3DScanner.

JLCPCP

The PCBs for the first prototype of the Open3DScanner were manufactured by JLCPBP . The manufacturer is located in China and offers very cheap PCBs, which were of very good quality when I ordered them. The only disadvantage, besides the minimum order of 5 boards, is that even the fastest shipping takes about a week. However, this is more than compensated when comparing prices with local suppliers.

For the design of the board the software KiCad was used. KiCad is an open source program package licensed under the GPL.

KiCad contains tools for all steps involved in designing a PCB. The individual tools offer very good integration with each other, which simplifies the design process considerably.

First you create a schematic diagram for the circuit. The individual components are connected with so-called footprints. These footprints define the layout of the individual components on the board (holes, labels, ...). With this information you go over to the actual design of the PCB.

Figure 3.3 shows the schematic diagram for the Open3DScanner.

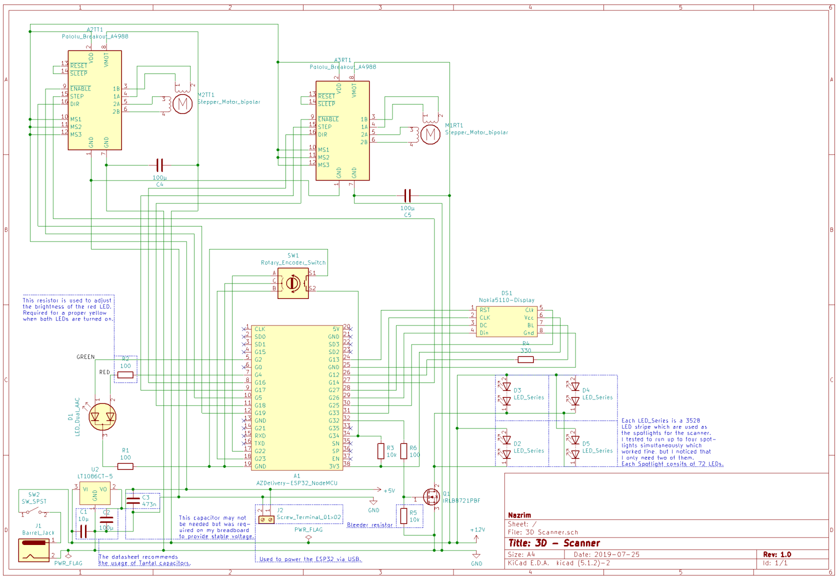


Figure 3.3 Schematic diagram for the Open3DScanner designed in KiCad

All elements are placed on an empty surface, which represents the later board. The user now has to position the individual elements and draw traces for the connections. The user is always shown which pins have to be connected to each other. Figure 3.4 shows the PCB design for the Open3DScanner.

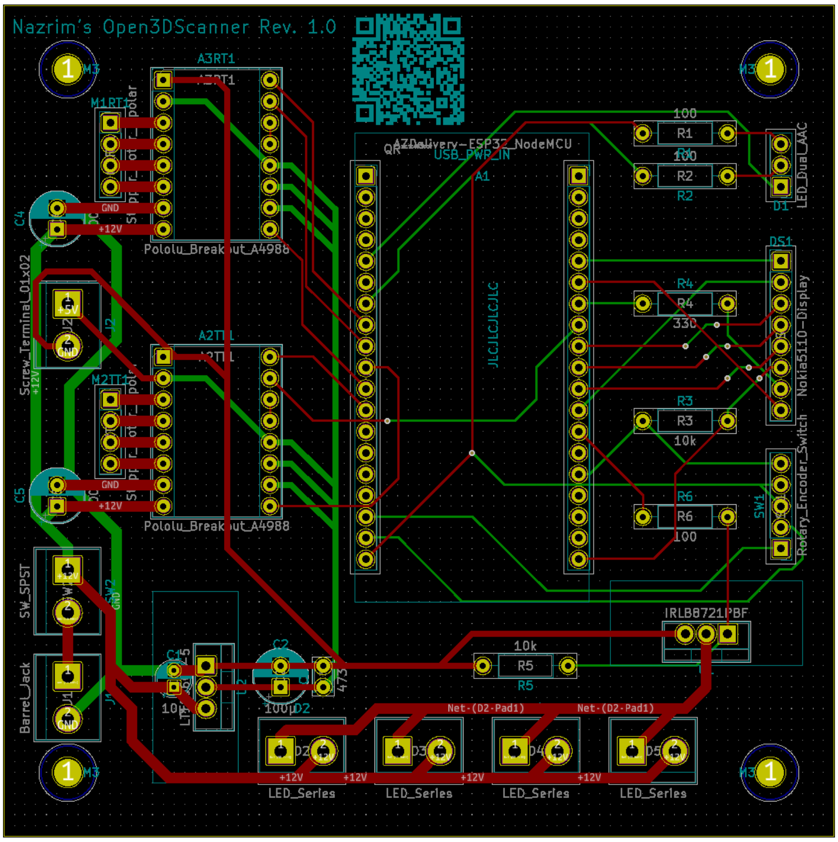


Figure 3.4 PCB design for the Open3DScanner designed in KiCad

At this point all necessary steps for the design of the PCB are completed. The necessary Gerber files can be generated and handed over to an appropriate service provider for production.

> **Gerber Files**
> Manufacturers generally provide instructions on which Gerber files to use and which naming conventions to follow. Often there are even instructions for concrete software.

Alternatively, a 3D model of the finished board (with all components) can be created beforehand. For this it is necessary to link the footprints with 3D models. For the rendering of the 3D model no further effort is necessary.

I consider this step very useful. On the one hand you get a better idea of what the finished board will look like and on the other hand you can check again that no components are in conflict with each other. This is especially important in case of a high packing density of the components. Figure 3.5 shows the rendered 3D model of the Open3DScanner's board including all components.

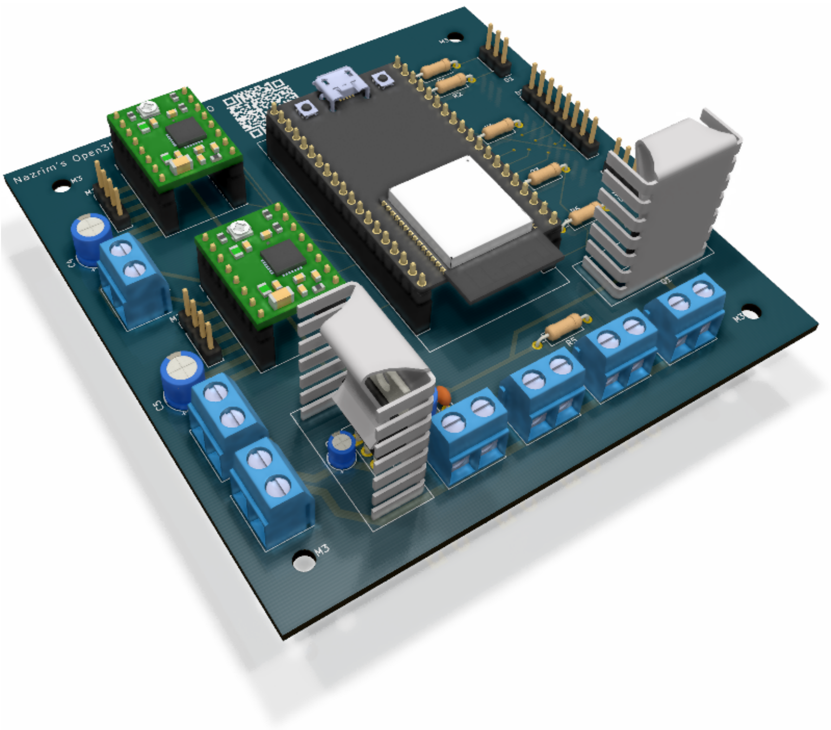


Figure 3.5 3D rendering of the PCB including components for the Open3DScanner designed in KiCad

Due to the large number of existing components, KiCad cannot contain a schematic symbol, footprint, and 3D model for all of them. It is also possible that a different footprint should be selected for a component, for example because it is not connected directly to the board, but via a pin header.

This also applied to several parts of the Open3DScanner. In case of missing schematic symbols and footprints, it is possible to draw the corresponding parts via intuitive integrated editors.

If 3D models for components are missing, it is necessary to provide them (for example as STEP file). If this does not happen, the board can still be rendered, but the corresponding part locations remain empty.

A possible source for corresponding 3D models of components to be used in open source projects is GrabCAD ✖. For licensing reasons I prefer other sources for obtaining 3D models of components. On the one hand many manufacturers already provide corresponding CAD files and on the other hand SnapEDA offers a large database of components containing a schematic symbol, a PCB footprint, and a 3D model. The individual entries are licensed under CC-BY-SA 4.0.

✖ **GrabCAD License**
3D models obtained from GrabCAD may only be used for non-commercial purposes. For commercial use, permission must be obtained from the author of the model.

3.2.1 Used 3D Models

This section lists the 3D models which are not already part of KiCad and which are used to render the Open3DScanner board and their source.

Table 3.2 3D Models used for rendering the Open3DScanner

Component	Source
ESP32 Devkit-C	SnapEDA - ESP32-DEVKITC-32D
Blue two pole screw terminal	SnapEDA - TB002-500-02BE
A4988 stepper driver	Pololu - A4988 product page
TO 220 attachable heatsink	fischer elektronik - FK 245 MI 247 O product page

3.3 3D Design

When it comes to the 3D modeling of the Open3DScanner, there is little special to mention.

All components needed for the project and printed with a 3D printer were designed with Fusion 360 . It should be noted that Fusion 360 contains handy scripts to create certain elements automatically. One of these scripts allows the creation of gears with given parameters. This script was used for the gears in the Open3DScanner.

CAD models of some standard components such as ball bearings are required for the subsequent building instructions and render images of the finished Open3DScanner.

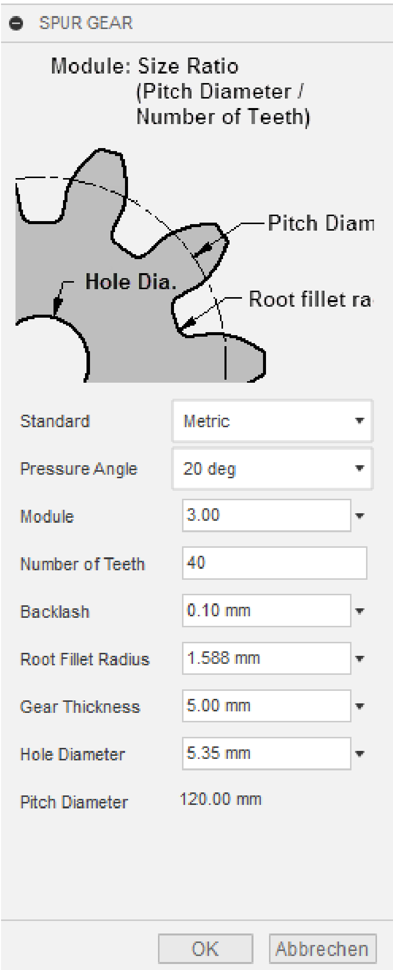


Figure 3.6 Fusion 360 script settings for creating the Open3DScanner's spur gear

Source	Model Type	License
AST Bearings	Bearings	not specified, but free
Cad.Solidworks.Parts	Nema 17	MIT
Octopart	Various	not specified, but free
FreeCAD-library	Dupont Connectors	CC-BY 3.0
Digi-Key	Molex Connectors & Cable Lugs	not specified, but free
SnapEDA	Micro USB-B Connector	CC-BY-SA 4.0

Table 3.3 Sources for CAD models of non-electrical components

There are several vendors with large libraries of CAD models, which may not be freely used and/or distributed. Table 3.3 contains information which offers were used for which models.

3.4 Photogrammetry

The heart of the entire photogrammetry process is the software that creates the 3D model from the images taken.

For this purpose I use the software Meshroom . It is intuitive to use and performs the entire process. The finished (textured) 3D model is created from the fed-in images. Alternative software solutions do not always offer the entire process and generate for example only a point cloud, which has to be converted into a 3D model with another program. This applies for example to VisualSFM .

Meshroom is released under the MPL 2.0 and can be compiled by the user or can be obtained as a ready-to-use installer.

In general, I would recommend the use of Meshroom to anyone, simply because it is very easy to use and produces great results while offering extensive configuration options.

The only limitation is that a CUDA-enabled graphics card (Nvidia) is required for optimal results, as the calculations are done on the GPU. If such a card is not available, a ”preview mode” can be activated, which works without CUDA, but also produces worse results.

3.5 LaTeX

LaTeX was used to create this manual and last but not least we will look at which packages were used to create it.

The document class used is yReport from Harvey Sheppard. The class is part of his project yLatex , which provides document classes and packages to create appealing documents. The entire yLatex project is licensed with LPPL 1.3.

The compiler used is XeLaTeX , which is required by the document class yReport. TeX Live is used as TeX distribution and TeXstudio as editor.

Pictures can be found in various places in this document. If these images are not photos, screenshots, or exports from other software, the images were created with Gimp which is licensed under the GPL.

In addition, various packages are used to realize individual aspects of the document. These are listed below.

Package	Version	Maintainer	License	Purpose
fontawesome5	5.9.0	Marcel Krüger	LPPL 1.3c & OFL	Required by yAuthorBlock and used to show some icons across the document.
yAuthorBlock	unknown	Harvey Sheppard	LPPL 1.3	Used to create the author block on the second page.
tabularx	2.11	The LaTeX Team	LPPL 1.3	Since tabu (used by yReport) is somewhat buggy, this document uses tabularx.
siunitx	2.7s	Joseph Wright	LPPL 1.3c	Used to display numbers and units in a proper way.
indextools	1.5.1	Maïeul Rouquette	LPPL 1.3	Used to build an index of all weblinks for printed documents.
xparse	2019-05-28	The LaTeX Team	LPPL 1.3c	Required for string magic to display urls in the index (Appendix B) in a proper way.
infoBulle	unknown	Harvey Sheppard	LPPL 1.3	Used for displaying various block types (info, warning, ...) in main area.
marginInfoBulle	unknown	Harvey Sheppard	LPPL 1.3	Used for displaying various block types (info, warning, ...) in margin area.
isodate	2.28	Harald Harders	LPPL 1.3c	Used for uniform displaying dates. Required because of strange behaviour of datetime package (included in yReport).
xurl	0.07	Herbert Voß	LPPL 1.3	Allow URL breaks at any alphanumerical character.
hologo	1.13	Heiko Oberdiek	LPPL 1.3	Used to disaplay logos from the LaTeX family.

Table 3.4 Used LaTeX packages within the Open3DScanner's manual

4

Chapter

Required Parts

This chapter shows which things are needed to build the Open3DScanner. It looks at the components of the printer as well as the tools and devices needed to assemble it.

It is recommended to read this chapter carefully if you want to build your own Open3DScanner.

4.1 Bill of Materials

One of the most important points to build your own Open3DScanner is a bill of materials of the required parts.

This is contained in the following sections. The complete list is divided into two partial lists. The first one contains all electrical components which can be found on the PCB or are connected to it. The second list contains all further things that are needed to assemble the Open3DScanner.

4.1.1 Electrical Components

The following table contains all electrical components required to build the Open3DScanner.

In addition to the component and the required quantity, the identifier of the respective component on the circuit board is also specified. In addition to this, information is given for individual components, which must be considered when purchasing.

Component	Quantity	PCB-Identifier	Note
ESP32-DevKitC	1	A1	While I use the AZ-Delivery ESP32-DevKitC , any version that meets the ESP32-DevKitC specification should work.
A4988 Stepper Driver	2	A2TT1 & A3RT1	Any A4988 steper driver should work. I recommend getting matching heat sinks.
Nema 17 Stepper Motor	2	M1RT1 & M2TT1	Make sure to buy the version with D-shaft. I use strong stepper motors, like this 59 N cm Nema 17 stepper motor .
LT1086CT-5	1	U2	Ensure to get the TO-220 type.
IRLB 8721	1	Q1	Ensure to get the TO-220 type.
TO-220 Heatsink	2	Q1 & U2	I highly recommend at least one for the LT1086CT-5 since it gets very hot during operation. The one for the IRLB 8721 is just to be safe. While I am using the FK 245 MI 247 O you can choose whatever fits in the space (about 22.5 mm×10 mm).
1 ml Thermal Paste	1	-	Required for a good connection between TO-220 components and their heatsink. Only a tiny amount is required.
STEC12E08 Rotary Encoder	1	-	This will be connected to the pin header SW1.
Nokia 5110 Display	1	-	This will be connected to the pin header DS1. Buy one with screw holes. Hole distance on x-axis should be 34 mm and 40.5 mm on the y-axis.
5 mm 3-Pin Bi-Color LED	1	-	This will be connected to the pin header D1. I recommend using a red-green one.
5 m×8 mm 300 LED 3528 Strip, 12 V 1.5 A	1	-	This will be connected to the pin headers D2, D3, D4, D5.
Round 20 mm SPST Rocker Switch (R13 112)	1	-	This will be connected to the pin header SW2.
5.5 mm×2.1 mm Female DC Power Jack Panel Mount (L722AS)	1	-	This will be connected to the pin header SW2. Make sure to grab one with threads and a nut for panel mounting.
473 nF Ceramic Capacitor	1	C3	Choose 2.54 mm pin distance and 3.4 mm radius.
10 µF Electrolytic Capacitor	1	C1	Choose 2 mm pin distance and 4 mm radius.
100 µF Electrolytic Capacitor	3	C2, C4, C5	Choose 2.5 mm pin distance and 6.3 mm radius.
100 Ω Resistor	3	R1, R2, R6	Choose whatever you have at hand, like carbon film or metal (oxide) film.
330 Ω Resistor	1	R4	Choose whatever you have at hand, like carbon film or metal (oxide) film.
10 kΩ Resistor	2	R3, R5	Choose whatever you have at hand, like carbon film or metal (oxide) film.

Table 4.1 BOM for all electrical components of the Open3DScanner — Part 1/3

Component	Quantity	PCB-Identifier	Note
2-Pin Screw Terminals 5mm Pitch	7	J1, J2, SW2, D2, D3, D4, D5	Take care to get the ones with 5mm pitch.
2.54 mm 40-Pin Header	1	M1RT1, M2TT2, D1, DS1, SW1	Will be cut into 1×3, 2×4, 1×5, and 1×8.
2.54 mm 40-Pin Socket	2	A1, A2TT1, A3RT1	Will be cut into 2×19 and 4×8. Each cut results in one socket loss.
1×3 Dupont Housing	4	-	Will be used to connect D1 with the bi-color LED on both sides as well as the STEC12E08 rotary encoder on component side.
1×5 Dupont Housing	1	-	Will be used to connect SW1 with the rotary encoder on PCB side.
1×8 Dupont Housing	2	-	Will be used to connect DS1 with the LCD on both sides.
Female Dupont Terminals	33	-	Required for all Dupont housings.
Molex crimp housing — Micro-Fit - 1x2-pin, male (430200201)	4	-	Required for connecting the lights to the Open3DScanner. Part number Molex 430200201
Molex crimp housing — Micro-Fit - 1x2-pin, female	4	-	Required for connecting the lights to the Open3DScanner. Part number Molex 430250200
Molex crimp contact — Micro-Fit, female	8	-	Required for connecting the lights to the Open3DScanner. Part number Molex 430300007
Molex crimp contact — Micro-Fit, male	8	-	Required for connecting the lights to the Open3DScanner. Part number Molex 430310007
Cable Lug	4	-	The cable lugs have to match your power jack and your rocker switch. For me it is 2×2.8 mm (power jack) and 2×4.8 mm (rocker switch).
Power Supply 12 V, 2250 mA with 5.5 mm×2.1 mm male Barrel Jack	1	-	This will power the whole Open3DScanner
15 cm Micro USB-B cable	1	-	Used to connect J2 with the ESP32's usb port. Get an already prepared cable if you can, otherwise you need to cut one yourself.
AWG 18 or 0.75 mm^2 cable	-	-	Used for transmitting power (e.g. from power jack to PCB, within pieces of the LED strip, and towards the LED strip). Since the cables are not exposed to any or only little movement, no special requirements like silicone cables exist. I use simple speaker cable which is sold in rolls of 25 m. I cannot provide an accurate required quantity for the wires since it depends somewhat on individual wiring.

Table 4.2 BOM for all electrical components of the Open3DScanner — Part 2/3

Component	Quantity	PCB-Identifier	Note
AWG 24 or 0.2 mm^2 cable	-	-	Used to connect the various components to the PCB. I cannot provide an accurate required quantity for the wires since it depends somewhat on individual wiring.
PCB	1	-	The Gerber files for the PCB of the Open3DScanner are part of this project and can be used to get PCBs from a manufacturer.

Table 4.3 BOM for all electrical components of the Open3DScanner — Part 3/3

4.1.2 Other Components

The previous section contains the BOM for all electrical parts of the Open3DScanner.

In addition, this section contains a BOM for all remaining parts needed to build the Open3DScanner. The separation should help to better bundle the orders with the respective suppliers.

It should be noted that the quantities specified for the screws indicate maximum quantities. These can be smaller, e.g. if the maximum of four lights are not mounted.

Component	Quantity	Note
400 mm × 550 mm × 16 mm Wooden Board	1	This will be used as the base of the whole Open3DScanner
500 mm × 550 mm × 3 mm PVC Rigid Foam Sheet	1	The rear panel of the Open3DScanner.
800 g Roll ABS Filament	1	Main color. Other materials like PLA may be fine.
800 g Roll ABS Filament	1	Secondary color. Other materials like PLA may be fine.
800 g Roll ABS Filament	1	Accent color. Other materials like PLA may be fine.
Nema 17 Damper	2	For decoupling the motors from the Open3DScanner's structure.
Liquid Glue	1	Required for assembling the lights if the M3 tape which is applied to the LED strip does not hold and to glue some nuts in place.
5 mm × 26 mm Steel Rod	5	Used to connect and secure various parts.
625ZZ Bearing	2	Used to connect the moving parts to the frame with as little friction as possible.

Table 4.4 BOM for all non-electrical components of the Open3DScanner 1/2

Component	Quantity	Note
M3×6 SHCS	6	Used to connect the stepper motors to the dampers as well as the LED to the housing.
M3×8 SHCS	12	Used for various connections.
M3×10 SHCS	12	Used for various connections.
M3×12 SHCS	6	Used for various connections.
M3×20 SHCS	4	Used for various connections.
M3 Nut	40	Used for assembling the lights.
4.0×16 Countersunk Wood Screw	56	Used to connect the individual parts with the base.
5/8" Rubber Seal Ring	1	Used for the stepper driver cable entry in the housing.
30 mm × 15 mm Rubber Feet	4	Used as feet for the Open3DScanner.

Table 4.5 BOM for all non-electrical components of the Open3DScanner 2/2

4.2 Used Tools

Various tools are required to set up the Open3DScanner. These are described in this section.

At first the standard tools are listed, which are needed, but are not described in this chapter. Screwdrivers are needed for the used screws and even if pliers and tweezers are not absolutely necessary, they prove to be useful in some parts of the assembly.

4.2.1 3D Printer

A 3D printer is needed to print all the models from which the Open3DScanner is built. The requirements are relatively low as all parts can be printed from PLA or ABS i .

All parts have been designed to fit in the print volume of an Original Prusa i3 MK3 , which has a print volume of 250 mm × 210 mm × 210 mm. Any other FDM 3D printer can be used, which has at least this print volume. The only thing I am recommending is a heated printing bed, but nowadays it is included in almost all printers.

> **Used Filaments**
> I printed all parts for the Open3DScanner with ABS, so it's the only material I can say for sure that it works. However, it should be possible to use PLA and PETG, as well as other materials, as there are no special forces acting on the parts.

4.2.2 Crimping Tools

In order to achieve a decent and secure wiring of the Open3DScanner, it is necessary to crimp the cables.

Different crimping tools are used for the different connections. A ferrule crimper is used for the connections to the screw terminals, a universal crimping tool is used for the dupont cables, which connect the individual components to the board, and a terminal crimp tool is used for the connections to individual components, which have plugs for cable

> **Universal Crimper**
> While I mostly bought cheap sets for my crimping tools, I bought an Engineer PA-09 for crimping Dupont cables, because I crimp such cables most often. Even if this crimping tool is a bit more expensive than comparable crimping pliers, the quality of the crimping tool pays off. For this reason I recommend the Engineer PA-09 , especially if you frequently crimp Dupont cables.

lugs.

4.2.3 Soldering Iron

A soldering iron is required to assemble the board. Since the Open3DScanner does not use any particularly sensitive or otherwise special parts, almost any soldering iron can be used. It is not necessary to use a digital soldering station.

4.3 Hardware for Photogrammetry

A powerful computer shortens the necessary processing time for the photogrammetry process. As already mentioned in chapter 3.4, a Nvidia GPU is required to use the Meshroom software. Other photogrammetry software may not require a Nvidia GPU.

Otherwise, photogrammetry software benefits greatly from more RAM, which is why 32 GB is a reasonable lower limit, which is also mentioned by various manufacturers of software for photogrammetry.

A further important point that must be considered is that during the calculation large amounts of data are generated in the individual intermediate steps, which have to be persisted. The size of the generated data exceeds the size of the input images by a multiple. One of my example scans with 231 images generated over 20 GB i of data.

> **i Example Scan**
> The mentioned scan consists of four individual scans, whose data were combined. At this point we do not want to go into further detail regarding this scan, corresponding information can be found in the chapters 6 and 7.

For this reason it must be ensured that sufficiently large amounts of storage space are available.

5

Chapter

Build Instructions

This chapter explains step by step the individual steps that are necessary to build an Open3DScanner from the individual components.

The individual work steps are grouped together to summarize similar tasks.

5.1 Printing the Parts

First, the required parts must be printed for assembly. As mentioned before, I recommend using ABS filament as it's the only type of filament I've tested for the Open3DScanner, but I assume that other materials like PLA and PETG will work as well.

So just use PLA if you can't print ABS with your 3D printer, for example because you don't have a closed print area or a heated print bed.

The parts of the Open3DScanner were designed so that they all fit into the print volume of an Original Prusa i3 MK3 (at least individually). This ensures that most 3D printers are able to print the parts for the Open3DScanner.

The tolerance for the connections between the parts, as well as for screw holes etc. is 0.35 mm for all parts at all points.

Test print

It may be helpful to print a part as a sample and check that the tolerances used are appropriate for the printer being used.

With the exception of the Rotor-Stand and Passive-Stand files, all files can be printed without support. For the two mentioned objects it is necessary to print support in the areas where the bearings are inserted.

Apart from the removal of the support, no post-processing of the parts is necessary.

> **Note**
>
> The assignment of the components to colours is arbitrary and corresponds to my personal taste. Of course it is possible to deviate from this "recommendation", it should only be pointed out that this allocation was used for the calculation of the necessary quantities of filament.

The following table lists which part has to be printed how often.

Table 5.1 Quantity and color of 3D printed parts

Part	Quantity	Color
Rotor-Arm	2	Primary
Spotligt-Frame	1 — 4	Primary
Turntable-Arm	1	Primary
Housing-Top	1	Primary
Cable-Holder	5	Primary
Backplate-Holder	2	Primary
Turntable-Medium	1	Secondary
Passive-Stand	1	Secondary
Rotor-Stand	1	Secondary
Spotligt-Stand	1 — 4	Secondary
Housing-Main	1	Secondary
Housing-Plate	1	Secondary
LCD-Holder	4	Secondary
LED-Holder	1	Secondary
Encoder-Holder	1	Secondary
Micro-Fit-Holder-P1	4	Secondary
Micro-Fit-Holder-P2	4	Secondary
Food-Holder-Clamp	4	Secondary
Rotor-Gear	1	Accent
Rotor-Pinion	1	Accent
Housing-Knob	1	Accent

The print settings for the objects are specified below. It is assumed that a 0.4 mm nozzle is used. For other nozzles the values have to be adjusted accordingly.

The amount of print settings listed in the table is kept as small as possible in order to describe only slicer independent settings.

In particular, the use of a specific slicer is not necessary for successful printing. When selecting the remaining settings, it is only necessary to ensure that the printed objects are as stable as possible.

Setting	Value
Layer height	0.2 mm
Perimeters	4
Solid top layers	5
Solid bottom layers	5
Infill	40 %

Table 5.2 Print settings for 3D printed parts with 0.4 mm nozzle

The 3D models for the objects are all available in the orientation in which it is recommended to print them. The parts are aligned in such a way that the stability of the individual parts is improved under later loads (e.g. turntable arm) or that the quality of the printed details is improved in areas where this is important (e.g. spotlight frame).

In case you stick to my color scheme and you build the maximum of four lights you will need a little bit over one 800 g roll of filament of the secondary color. In total you need about 500 g primary color filament, 850 g secondary color filament, and 80 g accent color filament.

5.2 Assembly the PCB

This section describes how to assemble the PCB. This is not a step by step guide, but I pick up some special and important points and describe them.

Such a step by step instruction for soldering the board is not included in this document. The only comparable one is a picture series, which documents all steps, which I have done during the assembly of the board. This picture series can be found in Appendix A.

Instead, the board should be assembled in the way that is easiest for you. All components have an identifier, which can be found in the BOM and is printed on the PCB.

5.2.1 Preparing Pin Headers & Sockets

Since both pin headers and pin sockets were purchased as 40 pin versions, it is necessary to cut the headers and sockets into suitable pieces before use.

With the pin headers, the process is very simple, since there is always a notch between the pins where the headers can be separated with a side cutter. Figure 5.1 shows a three pin header with the clearly visible notches.

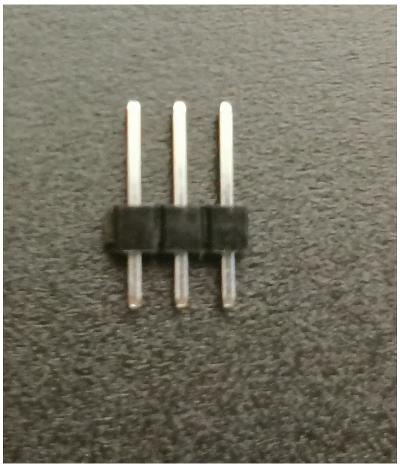


Figure 5.1 The notches between the pins simplify separation

With pin sockets this process is not as easy as with pin headers. Because of missing space there are no notches between the pins.

For this reason it is necessary to first remove a pin with pliers, at which the strip should be separated. If a five pin socket is needed, the sixth pin is removed. Afterwards the socket is cut carefully with a side cutter at the now free place. This allows the sockets to be separated safely and reliably. The rough edges can be smoothed with the side cutter. The process is shown in figure 5.2.

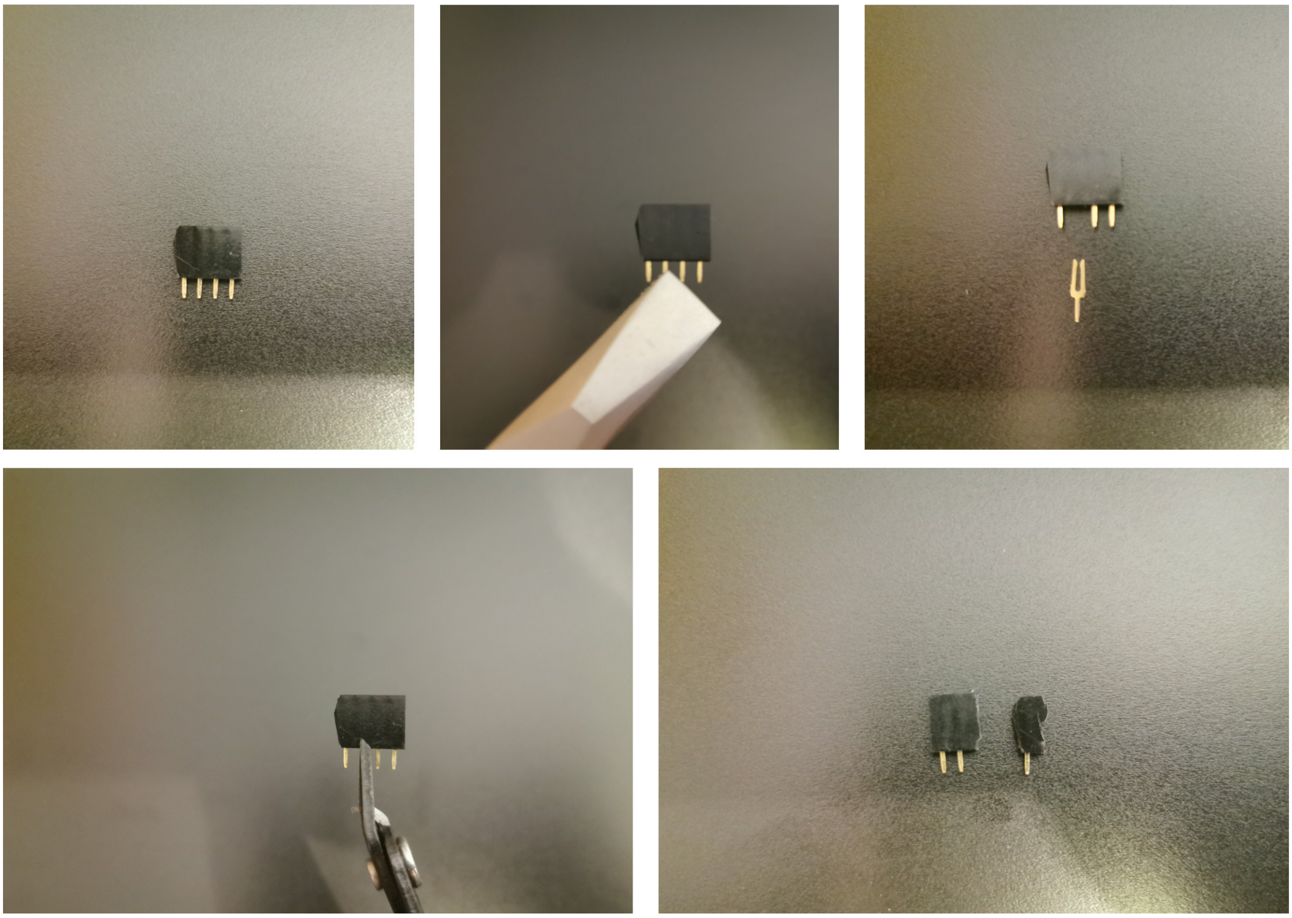


Figure 5.2 How to separate pin sockets

5.2.2 Assemble Pin Sockets

The components which are mounted on pin sockets are the A4988 Stepper Drivers and the ESP32-DevKitC .

Both these components have two pin rows. It is therefore important that the pin sockets are mounted straight and parallel on the PCB so that the components can be inserted later.

Without assistive tools, this is a pretty fiddly task. The simplest solution to this problem is to plug the pin sockets onto the corresponding components and solder them to the PCB. This ensures that the components fit exactly later.

5.2.3 TO-220 & A4988 Cooling

The Open3DScanner does not use active cooling, but it is recommended to cool some parts passively.

On the one hand the A4988 Stepper Drivers should be equipped with a heat sink. If the stepper drivers get too warm, you may experience strange behavior (like missed steps). This is especially true if the stepper drivers are set to above 1 A.

Furthermore, there are two TO-220 components on the board: the IRLB 8721 and the LT1086CT-5 . The LT1086CT-5 generates a lot of heat during operation, which can shorten the lifetime of the component considerably. For this reason I strongly recommend the use of thermal paste and a suitable heat sink.

For the IRLB 8721 I didn't check to what extent it generates heat during operation, but according to the principle "better safe than sorry" I recommend the use of heat conducting paste and heat sink.

Make sure that the heat sinks do not touch any other components or the circuit board. Especially with the LT1086CT-5 it is a bit tight, because a high packing density is needed for an optimal function of the component.

If you are using the TO-220 heat sinks I specified in the BOM, please note the following: Depending on how deep the pins are inserted into the PCB before soldering, it may be necessary to remove the bottom fin from the heat sink. Otherwise the heat sinks do not fit on the PCB, because they cannot put pressure on the component, which is necessary with the attachable heat sinks.

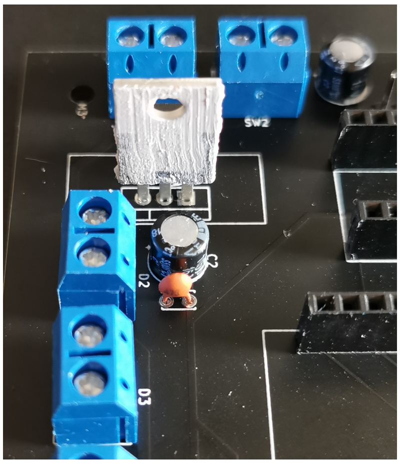


Figure 5.3 Apply a thin layer of thermal paste...

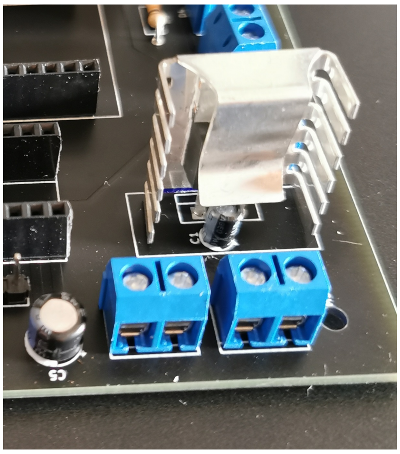


Figure 5.4 ...before installing the heat sink

5.2.4 Prepare the LED Strip

The instructions in this section must be repeated for each light that will be connected to the Open3DScanner. While it is possible to operate the Open3DScanner with up to four lights simultaneously, I only use two.

For the construction of the lights it is first necessary to prepare the LED strip. The LED strip is marked at regular intervals. At these markings it is possible to divide the strip.

For the lights of the Open3DScanner 15 cm long strips are needed. This corresponds to three segments of three LEDs. Figure 5.5 shows such a 15 cm piece of LED strip.

Figure 5.5 One piece of LED strip as used in the Open3DScanner

Each light of the Open3DScanner uses eight of these short strips and thus 72 LEDs.

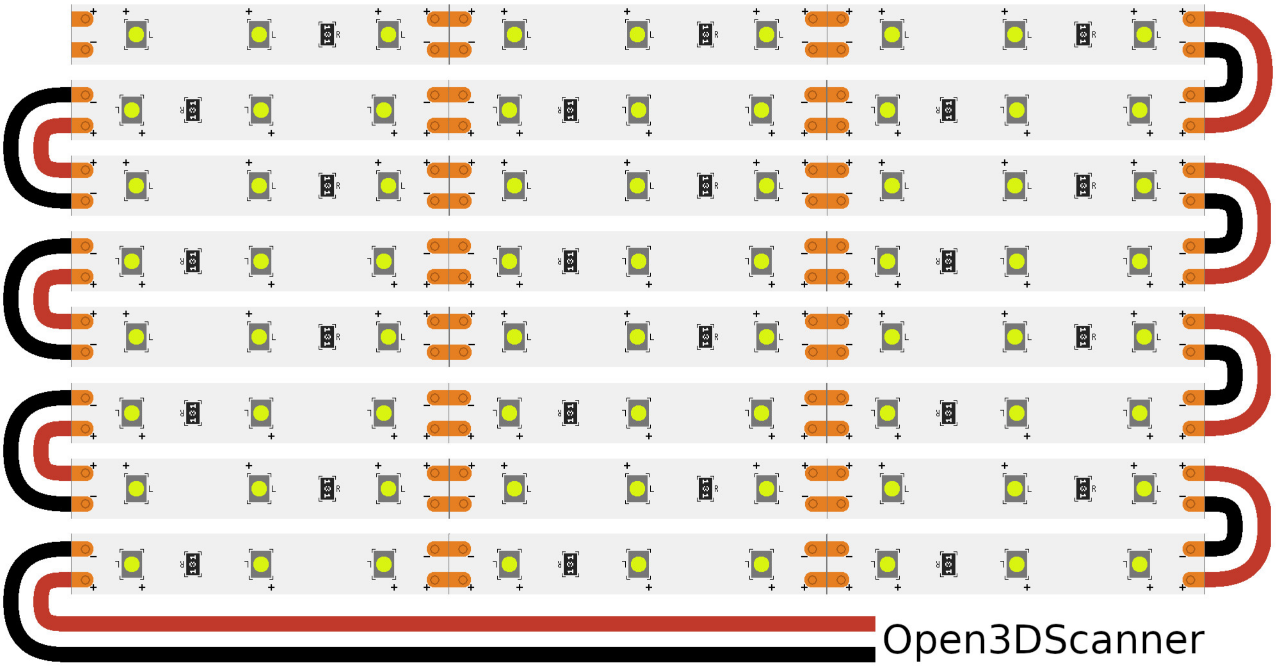


Figure 5.6 Schematic representation of the soldered LED strip for one light

After dividing the LED Strip we need seven about 7 cm long pieces of 0.75 mm^2 cable with two wires each (for + and -). These are used to connect the segments of the LED strip to each other. We also need a approx. 60 cm cable, which is used for the connection to the Open3DScanner. This cable can be shortened as soon as the lights are mounted and the cable is properly laid.

Now the individual pieces of the LED strip are soldered together in series and the long cable is soldered to one end.

It is important that no Molex connector is attached to the long cable at this point, otherwise we will not be able to lay the cable as intended. In general, it makes sense to attach the connector after the cable has been shortened to the correct length.

A schematic representation of how the result of the soldering should look like can be found in figure 5.6. When soldering, be sure to connect the right contacts.

Now the soldered LED strip is glued to the Spotlight-Frame. One strip is glued in the middle of each of the recesses. Make sure that the long cable is at the lower end.

The end of the long cable is led through the hole, which is in the lower area, in the middle of the part. Figure 5.7 shows what the result should look like.

Figure 5.7 Soldered LED strip glued into printed part

The LED strip is glued in with the pre-assembled M3 adhesive tape. If this is not strong enough and the LED strips come loose again, you can simply spread some liquid glue on the printed part and then press the LED strip in again.

The LED strip is now ready for the assembly of the light which is described in section 5.3.

5.2.5 Preparing the Cables

In this section the cables used in the Open3DScanner are described in more detail. This is explicitly not a manual for crimping, there are very good sources for this[1].

1. There are many good instructions for crimping various connections on the Internet. For example, for crimping Dupont cables , Molex Micro Fit , and cable lugs . YouTube is a great source for tutorials, too.

Instead, the following paragraphs describe the individual cables and their connections.

A schematic diagram is given for each cable, which clearly shows how cables are aligned, if necessary.

Power cables are shown in red (+) and black (GND), while all data cables are shown in blue. This colour coding is reused in later illustrations and thus simplifies orientation.

The cable lengths specified do not include the length of the connectors which are used for each cable.

LED Strip

Figure 5.8 Schematic representation of the LED strip cable

Characteristic	Value
Quantity	4
Wire Gauge	0.75 mm^2
Length	15 cm
End A Connector	Ferrule Crimp
End B Connector	Micro-Fit - 1x2-pin, male
End A connects	PCB contacts D2, D3, D4, D5
End B connects	Open3DScanner housing

Table 5.3 LED strip cable characteristics

During assembly, the Molex connectors are clamped into the housing of the Open3DScanner so that they are directly accessible from the outside and the lights can be connected with a suitable Molex plug.

Since ferrule crimps are necessary for the 0.75 mm^2 cables, they may sit very tight in the screw terminals afterwards, which is no problem.

Even if no or less than four lights will be connected to the Open3DScanner, it is necessary to prepare four of these cables so that there will be no gaps in the case later.

The corresponding female connectors need to be added to the LED strips. This cable is not shown here. Just ensure that the polarity matches these cables.

Power Cables

Characteristic	Value
Quantity	2
Wire Gauge	0.75 mm^2
Length	16/20 cm (Power Jack/Rocker Switch)
End A Connector	Ferrule Crimp
End B Connector	Cable Lug
End A connects	PCB contacts J1, SW2
End B connects	Rocker Switch & Power Jack

Table 5.4 Power connection cable characteristics

Figure 5.9 Schematic representation of the power connection cable

Two identical cables are used to connect the power source to the PCB and to connect the rocker switch, which allows the device to be switched on and off.

Make sure that the correct cable lugs are used for both cables. These can be different for different versions of the parts and must be selected appropriately for the components.

USB Cable

Characteristic	Value
Quantity	1
Wire Gauge	Undefined
Length	15 cm
End A Connector	Ferrule Crimp
End B Connector	Micro USB-B
End A connects	PCB contact J2
End B connects	ESP32-DevKitC

Table 5.5 USB cable characteristics

Figure 5.10 Schematic representation of the USB cable

Since an ESP32-DevKitC is used in the Open3DScanner, it is necessary to supply it with power via the Micro-USB-B socket.

The cable diameter for the USB cable is not specified as any USB cable can be used to power the ESP32. The available USB cables on the market have different cable diameters, but it is not necessary to buy a special cable.

Bi-Color LED Cable

Characteristic	Value
Quantity	1
Wire Gauge	0.2 mm^2
Length	11 cm
End A Connector	3-Pin Dupont
End B Connector	3-Pin Dupont
End A connects	PCB contact D1
End B connects	Bi-color LED

Table 5.6 Bi-color LED cable characteristics

Figure 5.11 Schematic representation of the bi-color LED cable

The bi-color LED is plugged directly into the Dupont connector.

When assembling this cable it is important to pay attention to the cable assignment. The two plugs are not connected identically.

Encoder Cable

Figure 5.12 Schematic representation of the encoder cable

Characteristic	Value
Quantity	1
Wire Gauge	0.2 mm^2
Length	15 cm
End A Connector	3-Pin Dupont
End B Connector	2×3-Pin Dupont
End A connects	PCB contact SW1
End B connects	Rotary Encoder

Table 5.7 Encoder cable characteristics

As with the bi-color LED, the cable for the encoder is connected directly to the component. However, there is a special aspect here which is given by the pin layout of the encoder used.

The pins of the encoder are arranged in two rows (2 + 3). The pins of the 2-series are arranged like a 3-series, where the middle pin is missing. For this reason, two 3-pin Dupont housings are used on the component side, where one space remains unoccupied.

Display Cable

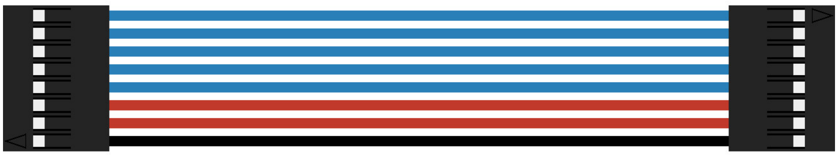


Figure 5.13 Schematic representation of the display cable

Characteristic	Value
Quantity	1
Wire Gauge	0.2 mm^2
Length	15 cm
End A Connector	8-Pin Dupont
End B Connector	8-Pin Dupont
End A connects	PCB contact DS1
End B connects	Display

Table 5.8 Display cable characteristics

Since the used display is mounted on a printed circuit board, the cable is connected to the two pin headers on the display and the PCB.

5.3 Assembly the Open3DScanner

After all the necessary preparatory work has been done, the actual instructions for building the Open3DScanner follow.

Each individual step is illustrated by an figure. In addition, each step is provided with a short instruction and, if useful, further tips, warnings and notes are given.

At each step all used components are listed. If these partially assembled parts are subsequently used in further steps, they will not be listed again.

At the end there are further illustrations, which contain exact dimensions and orientations of parts.

Part	Quantity
Rotor-Stand	1
Passive-Stand	1
625ZZ Bearing	2

Table 5.9 Required parts for step 1

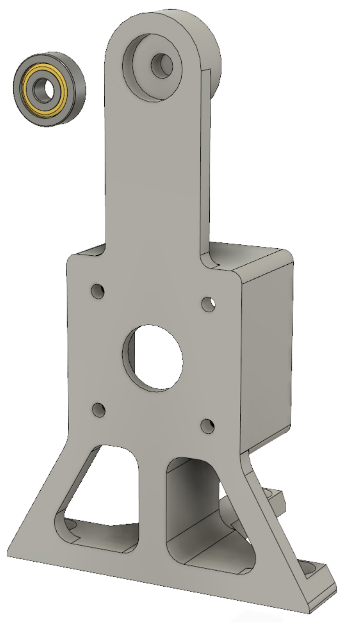
Instructions

Press one bearing into each of the openings provided in the upper area of the printed parts. The bearings should be flush with the surface of the printed parts.

Tip

If the opening for the bearings is a tad too tight, you can use pliers to press the bearings into the opening. In this case a piece of fabric or tissue should be placed between the pliers and the components to avoid craters and deformation.

1.a
Step



1.b
Step

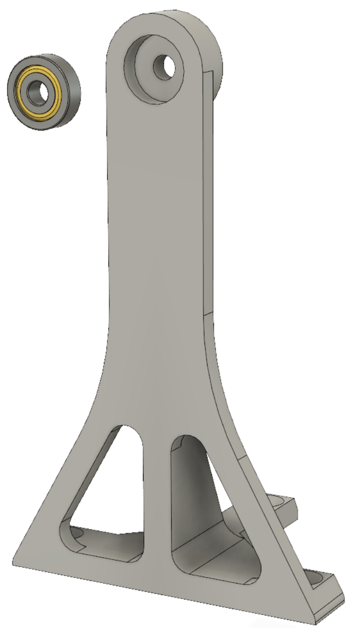


Figure 5.14 Step 1 of the Open3DScanner assembly

2
Step

Part	Quantity
Nema 17 Stepper Motor	2
Nema 17 Damper	2
M3×6	4

Table 5.10 Required parts for step 2

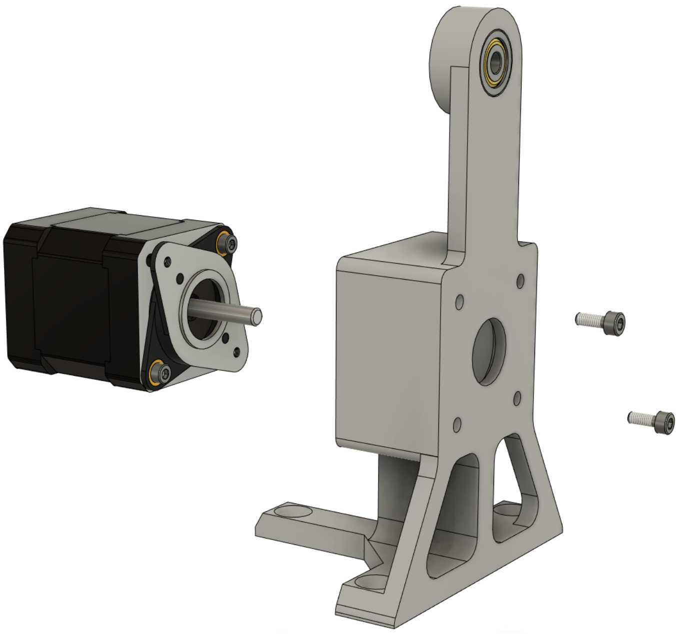
Instructions

Screw the dampers to the stepper motors.

Tip

The orientation of the dampers does not matter, it is only important that the side with the threads points away from the stepper motor.

3
Step



Part	Quantity
M3×8	2

Table 5.11 Required parts for step 3

Instructions

Screw one stepper motor to the Rotor-Stand.

Check

Make sure that the cables are pointing downwards to simplify the wiring later on.

Figure 5.15 Step 2 & 3 of the Open3DScanner assembly

Part	Quantity
Rotor-Gear	1
Rotor-Pinion	1
5 mm × 26 mm Steel Rod	1

Table 5.12 Required parts for step 4

Instructions

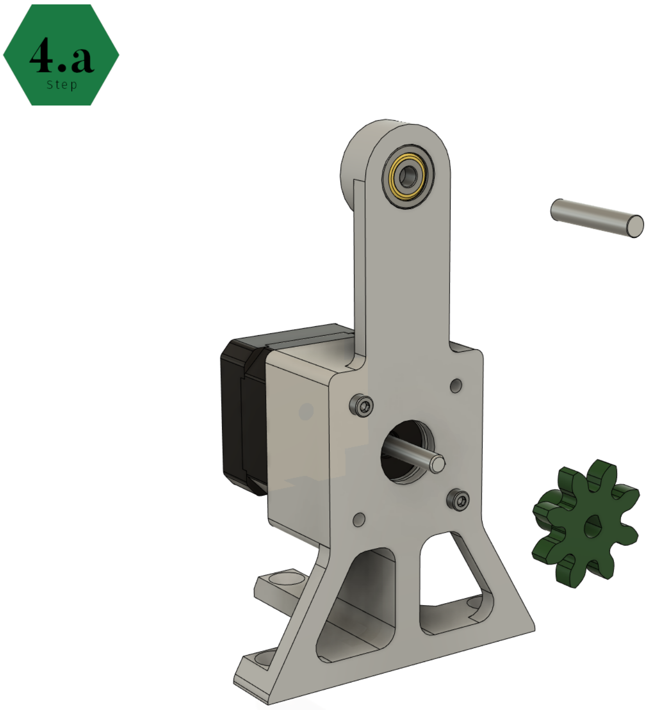
Press the Rotor-Pinion onto the shaft of the stepper motor. Press the steel rod into the bearing so that it is countersunk as far as possible in the printed part without hindering free rotation. Push the Rotor-Gear onto the steel rod. Use the middle hole of the Rotor-Gear for this.

Tip

The Rotor-Gear and the steel rod will not be flush. This is not problematic and improves the stability of the assembly in later steps.

Double check

The design does not ensure alignment between the two gears. It is intended that the rotor gear is centered on the rotor pinion and protrudes about 1mm on both sides. This must be strictly adhered to in order to allow a smooth movement later on. The connections are so tight that the parts should not slip after alignment.



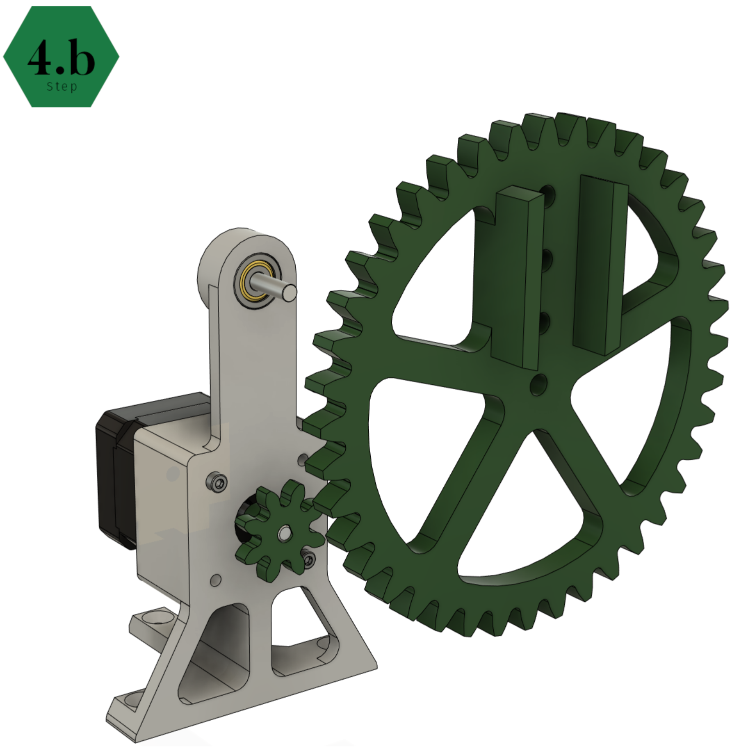
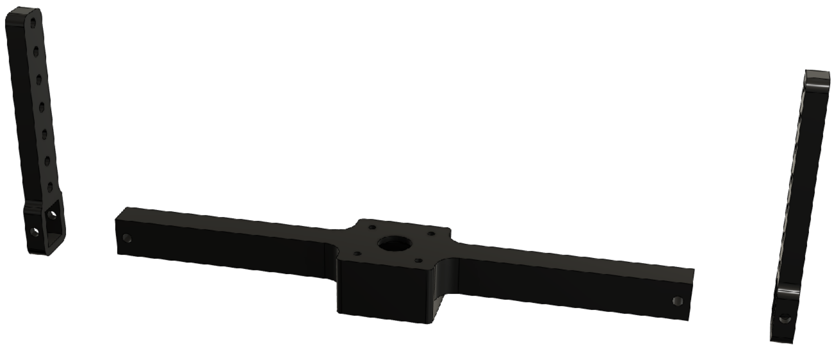


Figure 5.16 Step 4 of the Open3DScanner assembly

5.a
Step

Part	Quantity
Rotor-Arm	2
Turntable-Arm	1
5 mm × 26 mm Steel Rod	2

Table 5.13 Required parts for step 5



Instructions

Press one Rotor-Arm on each end of the Turntable-Arm so that they are flush with the holes. Press a steel rod into each of the holes to firmly join the components together. The steel rods should be flush with the Rotor-Arm on both sides.

Check

Make sure that the Turntable-Arm is correctly turned to prevent subsequent disassembly and correction.

5.b
Step

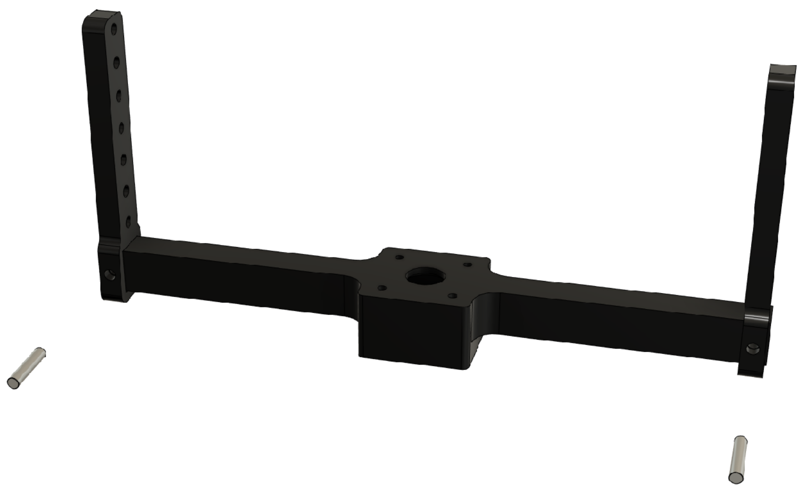
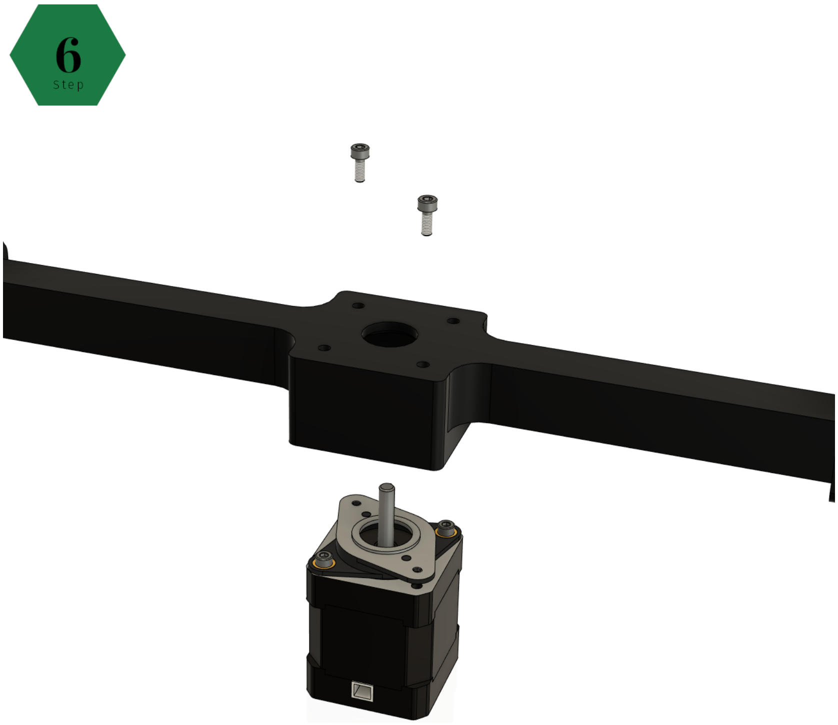


Figure 5.17 Step 5 of the Open3DScanner assembly

Part	Quantity
M3×8	2

Table 5.14 Required parts for step 6

Instructions
Screw the remaining stepper motor to the Turntable-Arm.



Part	Quantity
5 mm × 26 mm Steel Rod	1

Table 5.15 Required parts for step 7

Instructions
Push the steel rod through the upper hole of one Rotor-Arm and then into the outer hole of the Rotor-Gear. The components should be flush with each other and the Rotor-Arm is fixed to the rotor gear by the two jaws.

Information
The Rotor-Arm is additionally fixed by the steel rod, which connects the Rotor-Gear with the Rotor-Stand.

Check
Make sure that the motor cables point towards you when the Rotor-Stand is on the left side. This will allow clean wiring.

Warning
For accurate alignment of the steel bar, consider the image cutout. The rod must protrude a little bit from the rotor arm, otherwise the rotation of the device will be blocked.

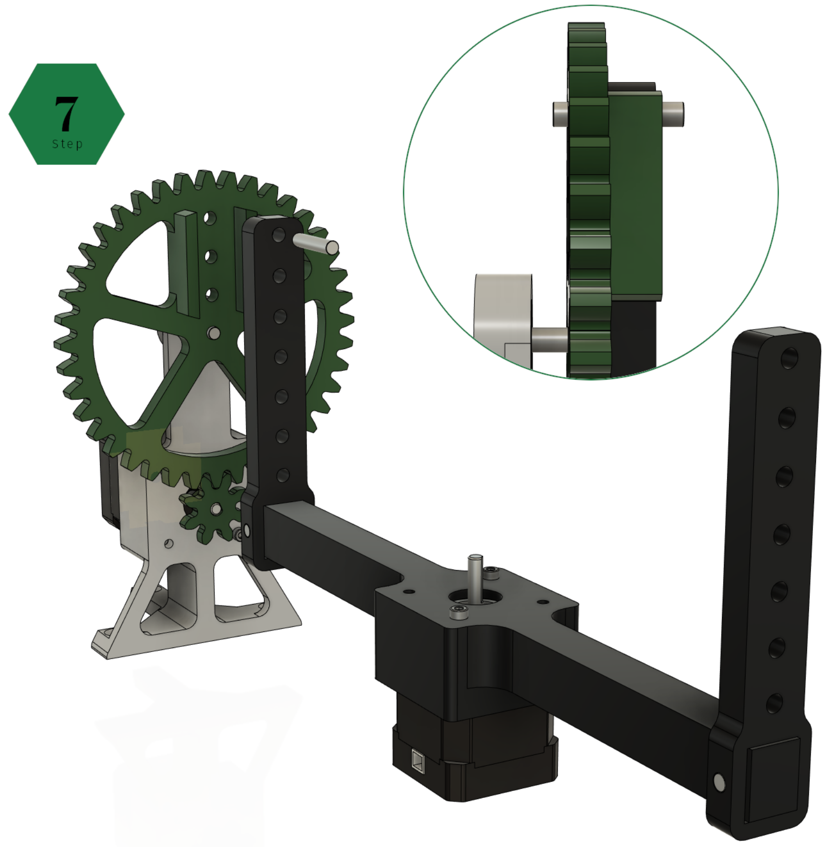


Figure 5.18 Step 6 & 7 of the Open3DScanner assembly

8
Step

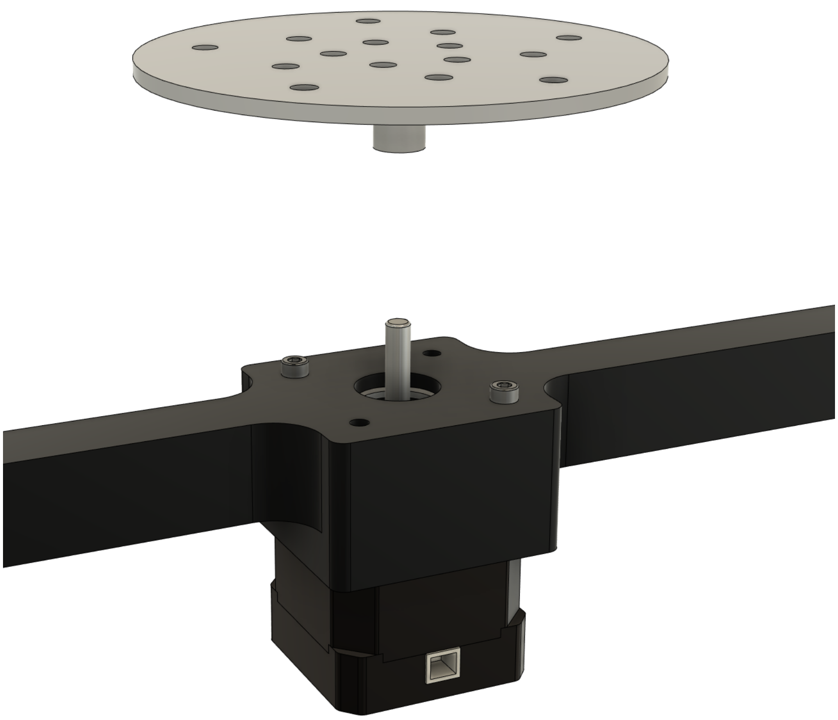
Part	Quantity
5 mm × 26 mm Steel Rod	1

Table 5.16 Required parts for step 8

Instructions

Connect the other Rotor-Arm to the bearing in the Passive-Stand. Use the center hole in the Rotor-Arm for this. The rod will be flush with the Rotor-Arm and protrude a bit from the back of the Passive-Stand.

9
Step



Part	Quantity
Turntable-Medium	1

Table 5.17 Required parts for step 9

Instructions

Push the Turntable-Medium as far as possible onto the shaft of the stepper motor, which is mounted to the Turntable-Arm.

Tip

Stabilize the structure by pressing against the bottom of the Turntable-Arm while pushing the Turntable-Medium onto the motor shaft. That way bending and deformation of the parts can be prevented.

Figure 5.19 Step 8 & 9 of the Open3DScanner assembly

Part	Quantity
Spotlight-Stand	1
Spotlight-Frame	1
M3×8	2
M3 Nut	2

Table 5.18 Required parts for step 10

Instructions

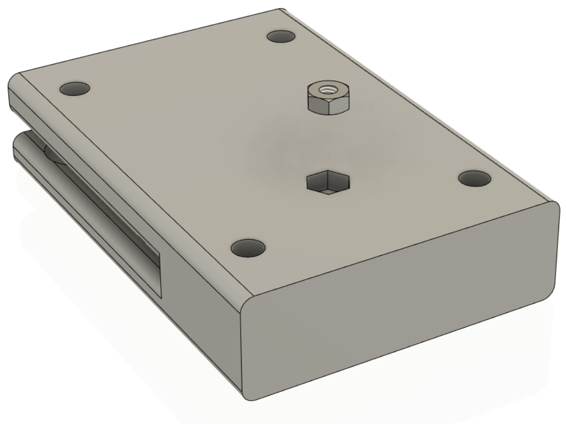
Press the two nuts into the recesses of the Spotlight-Stand. Select two adjacent holes that are at the same height. Next attach the Spotlight-Frame with the Spotlight-Stand. To do this, screw the screws through the holes in the Spotlight-Frame with the nuts on the back of the Spotlight-Stand.

Tip

Repeat this step up to four times for each spotlight you want to use. The height of the Spotlight frame can be changed by inserting the nuts into other recesses. This allows the illumination for the scans to be changed. The hole in the middle of the lower part of the Spotlight-Frame is for cable routing.

Figure 5.20 Step 10 of the Open3DScanner assembly

11.a Step



Part	Quantity
Foot-Holder-Clamp	4
30 mm × 15 mm Rubber Feet	4
M3×20	4
M3 Nut	4

Table 5.19 Required parts for step 11

Instructions

Press the nut into the recess on top of the Foot-Holder-Clamp. Then screw the rubber feet with the screw from the bottom to the nut.

Tip

Repeat this step four times to assemble all feet for the base plate.

11.b Step

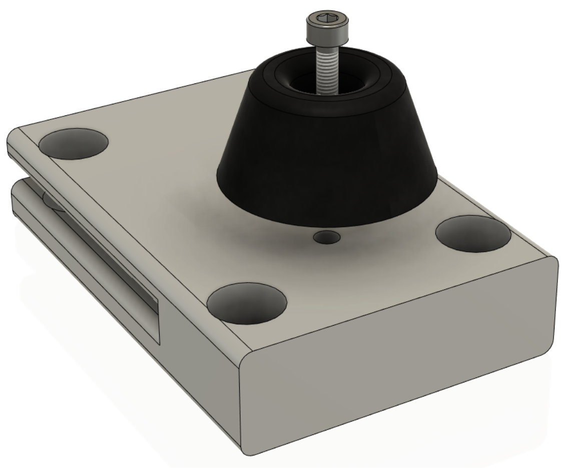


Figure 5.21 Step 11 of the Open3DScanner assembly

Part	Quantity
400 mm × 550 mm × 16 mm Wooden Board	1
Backplate-Holder	2
4.0×16 Countersunk Wood Screwt	24

Table 5.20 Required parts for step 12

Instructions

Screw the parts to the wooden board which is used as base plate. Use four screws for each assembled foot and Backplate-Holder. On the underside of the wooden board, mount one foot flush in each corner. The horizontal opening points into the inner area of the plate. The openings of two feet, which are mounted along the long edge of the base plate, face each other. The backplate holders are mounted from the top of the base plate. They are mounted flush in the corners of one of the long edges of the wooden board. The Backplate-Holder inserts have to be parallel to the long edge of the base plate.

Tip

For scans the backplate is inserted into the Backplate-Holders. When the scanner should be stored the backplate can be slided into the Foot-holder-Clamps to reduce the volume of the scanner.

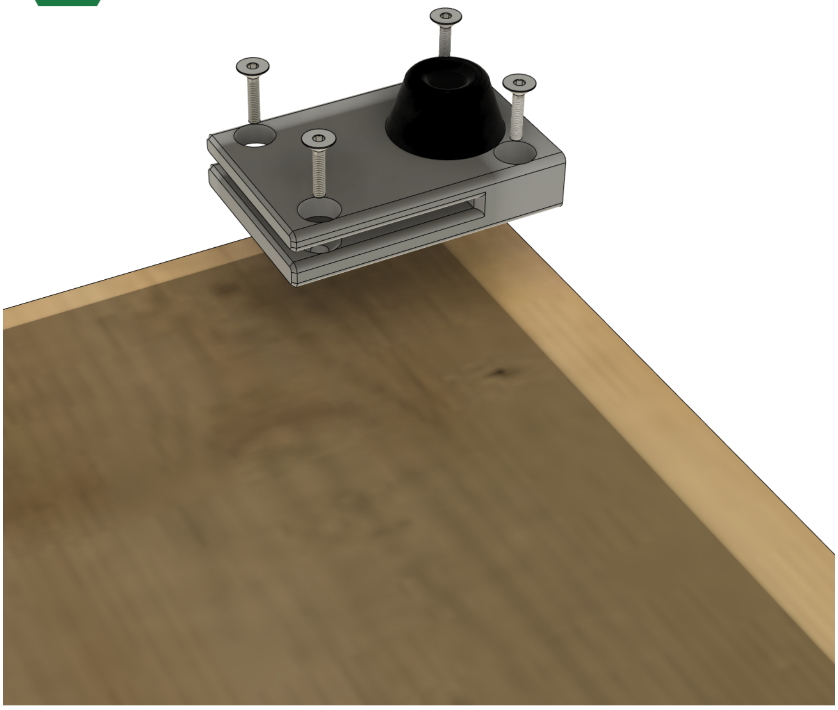
Check

You need to mount four assembled feets and two Backplate-Holder.

Warning

Align the components as well as possible at the corners of the wooden board. Otherwise it can happen that the rear panel does not fit properly or that parts cannot be screwed down because the screws on the top and bottom side collide.

12.a Step



12.b Step

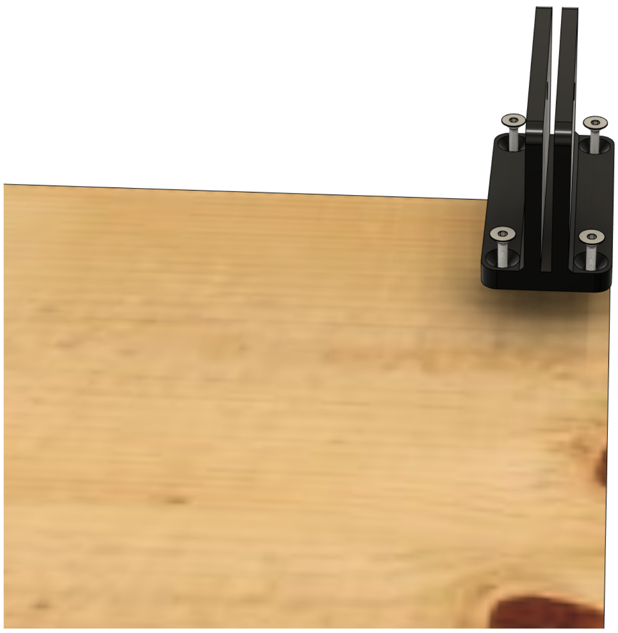
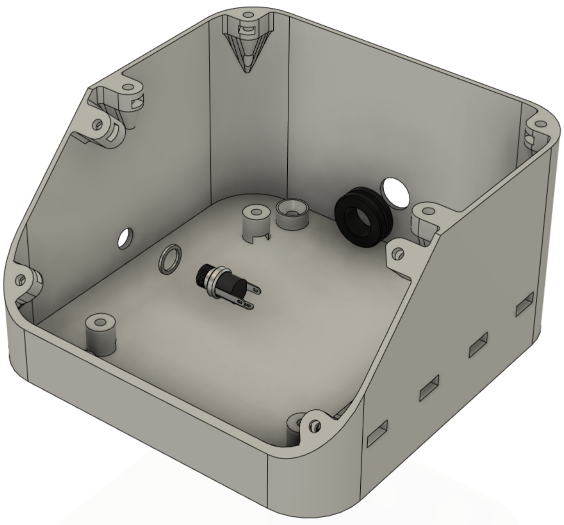


Figure 5.22 Step 12 of the Open3DScanner assembly

13.a
Step



Part	Quantity
Housing-Main	1
5.5 mm × 2.1 mm Female DC Power Jack Panel Mount (L722AS)	1
5/8" Rubber Seal Ring	1

Table 5.21 Required parts for step 13

Instructions

Press the rubber seal into the opening provided in the back of the housing. Mount the power jack in the opening on the left side of the case.

Tip

If your power jack does not have the accessories for mounting, you will need a M8 nut and a 1 mm washer with 8 mm diameter in addition. Pliers simplify the tightening of the nut. In order to simplify the wiring, it makes sense to attach the power cable to the jack first.

Warning

Make sure that the power jack is aligned as shown in the illustration, otherwise the PCB may not fit into the case anymore.

13.b
Step

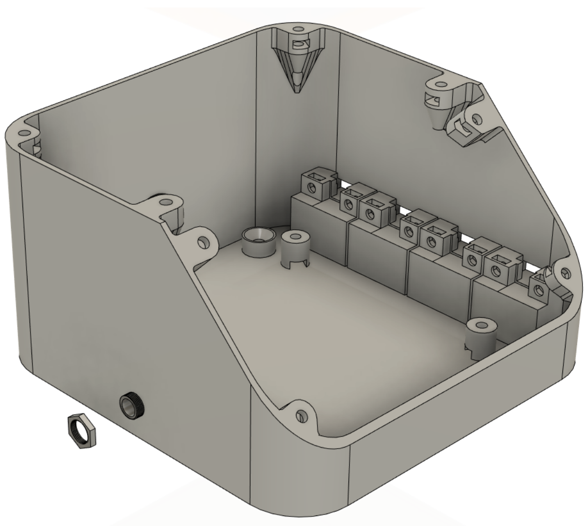


Figure 5.23 Step 13 of the Open3DScanner assembly

Part	Quantity
Molex crimp housing — Micro-Fit - 1x2-pin, male (430200201)	4
M3 Nut	8

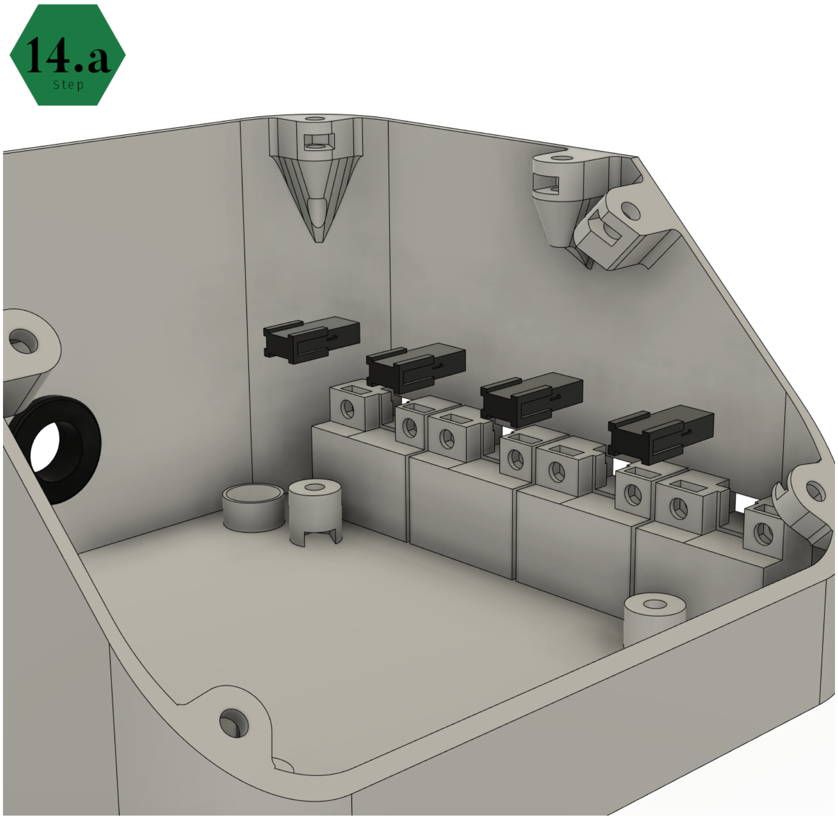
Table 5.22 Required parts for step 14

Instructions

Mount the Micro-Fit connectors. Make sure that the mounting hook points to the rear of the housing. The connectors are first pressed against the housing wall and pushed into the recess. Then they are pushed out a little bit so that they are flush with the outside of the housing. Slide the nuts into the openings. Two nuts are required for each connector.

Tip

Before mounting the Micro-Fit connectors, connect the cables to them.



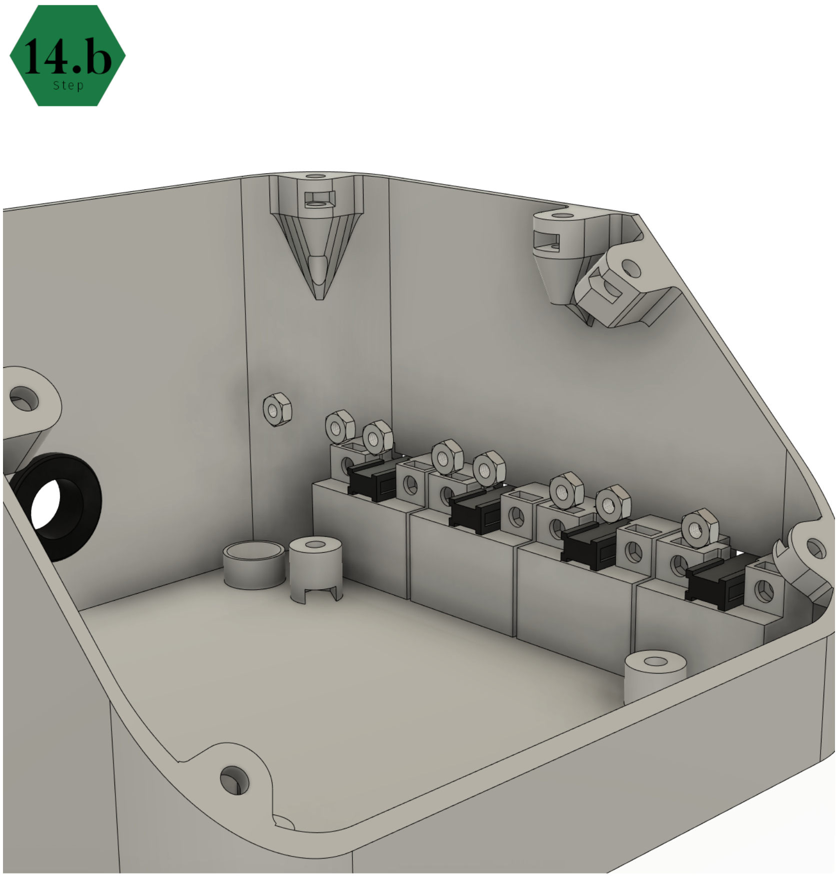
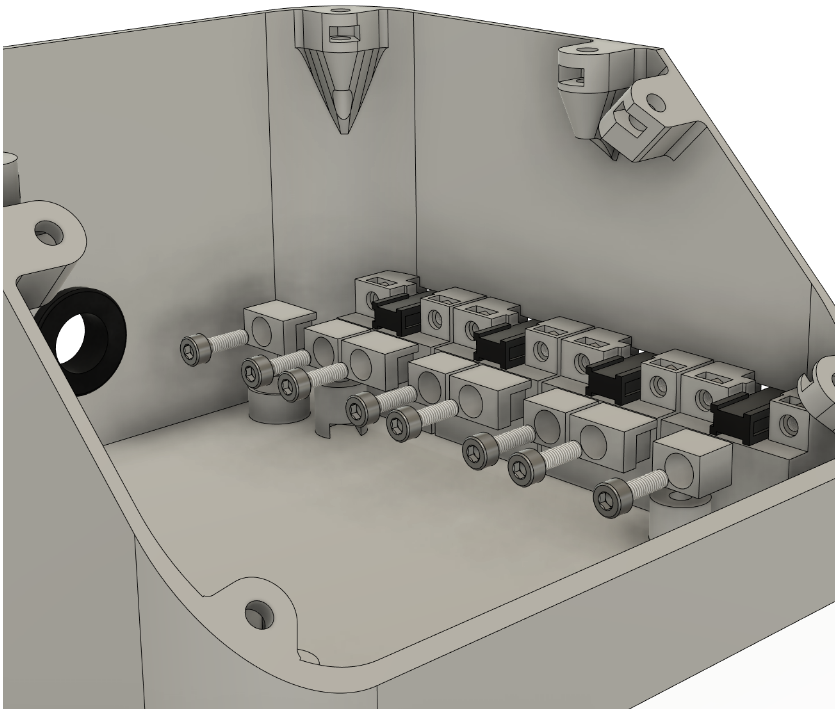


Figure 5.24 Step 14 of the Open3DScanner assembly

15 Step



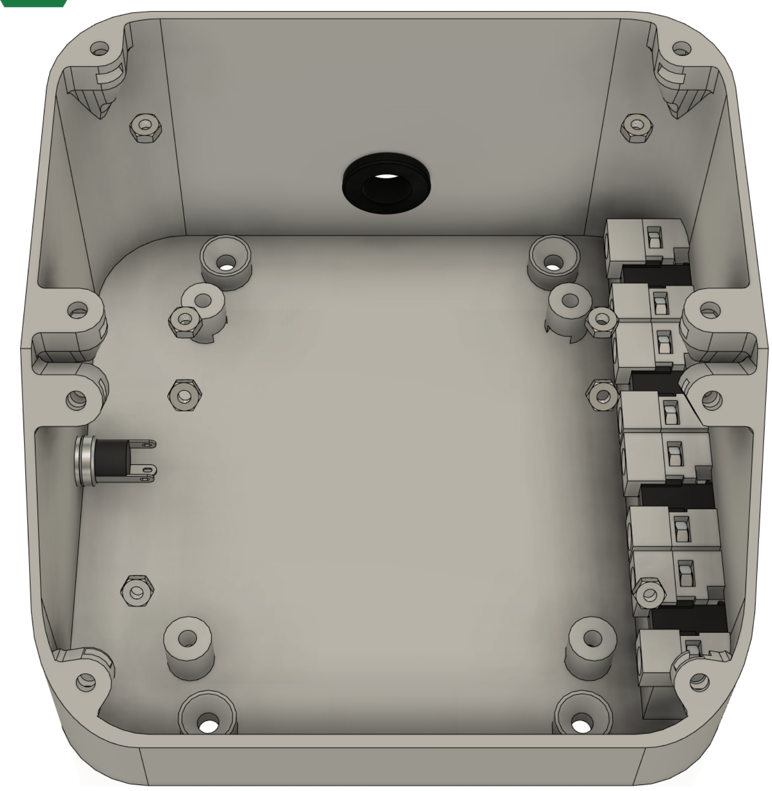
Part	Quantity
Micro-Fit-Holder-P1	4
Micro-Fit-Holder-P2	4
M3×10	8

Table 5.23 Required parts for step 15

Instructions

Use one Micro-Fit-Holder-P1, one Micro-Fit-Holder-P1, and two screws to secure each Micro-Fit connectors.

16 Step



Part	Quantity
M3 Nut	8

Table 5.24 Required parts for step 16

Instructions

Slide a nut into each opening along the top edge of the housing.

Figure 5.25 Step 15 & 16 of the Open3DScanner assembly

Part	Quantity
PCB	1
M3 Nut	4
M3×12	4

Table 5.25 Required parts for step 17

Instructions
Slide one nut each into the PCB stand at the bottom of the housing. Then screw the PCB into the housing.

Tip
The cable from the power jack can be routed below the PCB. This is much easier if it is done before the PCB is screwed down.

Check
Make sure that you align the PCB as shown in the figure, otherwise the wiring may become difficult and the specified cable lengths will no longer fit.

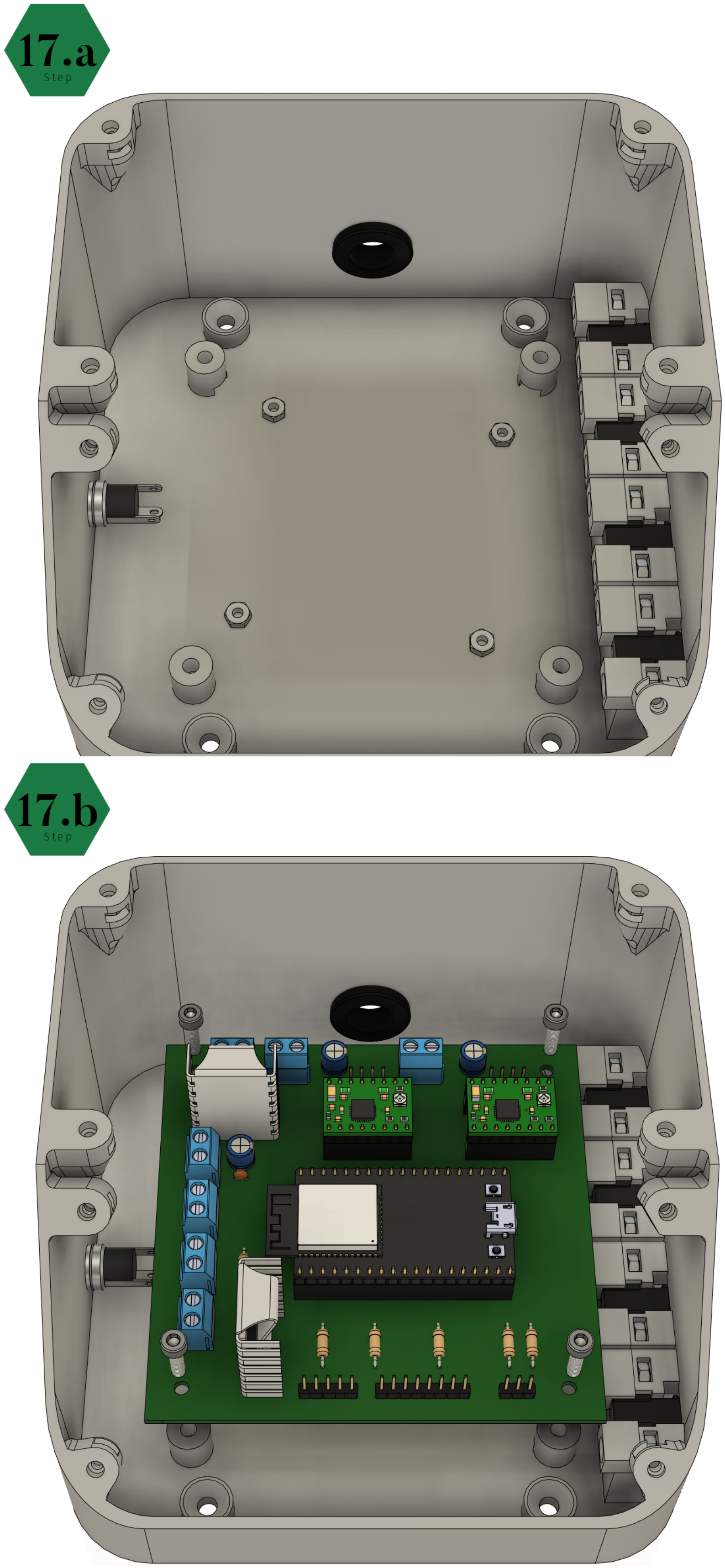
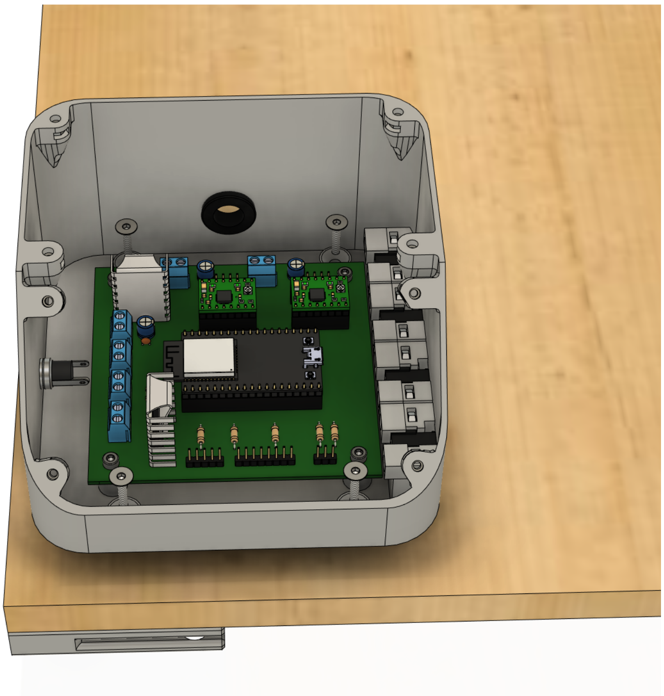


Figure 5.26 Step 17 of the Open3DScanner assembly

18
Step



Part	Quantity
4.0×16 Countersunk Wood Screwt	4

Table 5.26 Required parts for step 18

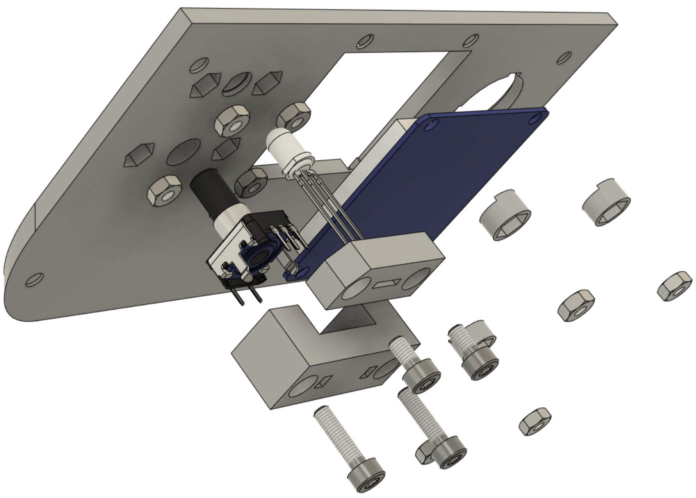
Instructions

Screw the housing onto the wooden base plate.

Tip

For the exact alignment on the base plate, there is a detailed illustration at the end of the assembly instructions.

19
Step



Part	Quantity
STEC12E08 Rotary Encoder	1
5 mm 3-Pin Bi-Color LED	1
Nokia 5110 Display	1
LCD-Holder	4
LED-Holder	1
Encoder-Holder	1
Housing-Front	1
M3×12	2
M3×6	2
M3 Nut	8

Table 5.27 Required parts for step 19

Instructions

Glue the nuts for the LED and the encoder with superglue into the front panel of the housing. Press the remaining nuts into the LCD-holder. Insert the LED, encoder, and LCD into the openings provided in the front panel. Screw the LED and encoder to the front panel. Use LED-Holder and Encoder-Holder for this.

Tip

Depending on the manufacturer of the LCD, it may be necessary to cut a piece out of the back of the front panel, otherwise the solder points will prevent the display from sitting flush in the panel.

Figure 5.27 Step 18 & 19 of the Open3DScanner assembly

Part	Quantity
Housing-Knob	1
Round 20 mm SPST Rocker Switch (R13 112)	1
M3×10	4

Table 5.28 Required parts for step 20

Instructions

Press the rocker switch into the recess of the front panel. Screw the display tight and pay attention to the orientation of the LCD-Holder. Press the Housing-Knob onto the shaft of the encoder.

Tip

Do not press the Housing-Knob too firmly on the shaft, otherwise the glued nuts will come loose.

20 Step

Part	Quantity
Housing-Top	1
M3×10	8

Table 5.29 Required parts for step 21

Instructions

Screw the Housing-Top and the assembled Housing-Front to the housing.

Tip

Before you screw the front panel to the housing, all cables should be connected inside the housing and the stepper motors should also be connected to the PCB. Also ensure that you flashed the firmware to the ESP32. I recommend a basic functionality test before closing the housing.

21 Step

Figure 5.28 Step 20 & 21 of the Open3DScanner assembly

22 Step

Part	Quantity
4.0×16 Countersunk Wood Screwt	6

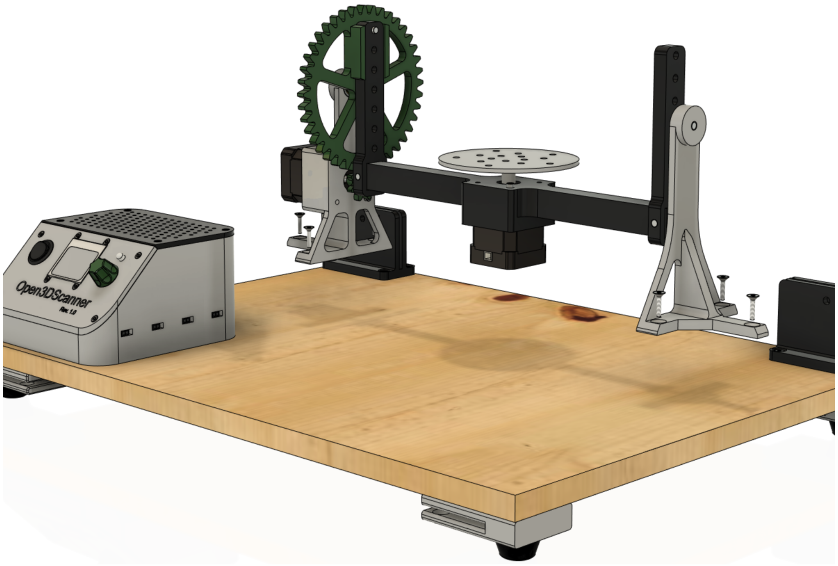
Table 5.30 Required parts for step 22

Instructions

Screw the assembled scanner onto the base plate.

Tip

For the exact alignment on the base plate, there is a detailed illustration at the end of the assembly instructions.



23 Step

Part	Quantity
4.0×16 Countersunk Wood Screwt	6

Table 5.31 Required parts for step 23

Instructions

Screw the assembled lights onto the base plate.

Tip

For the exact alignment on the base plate, there is a detailed illustration at the end of the assembly instructions.

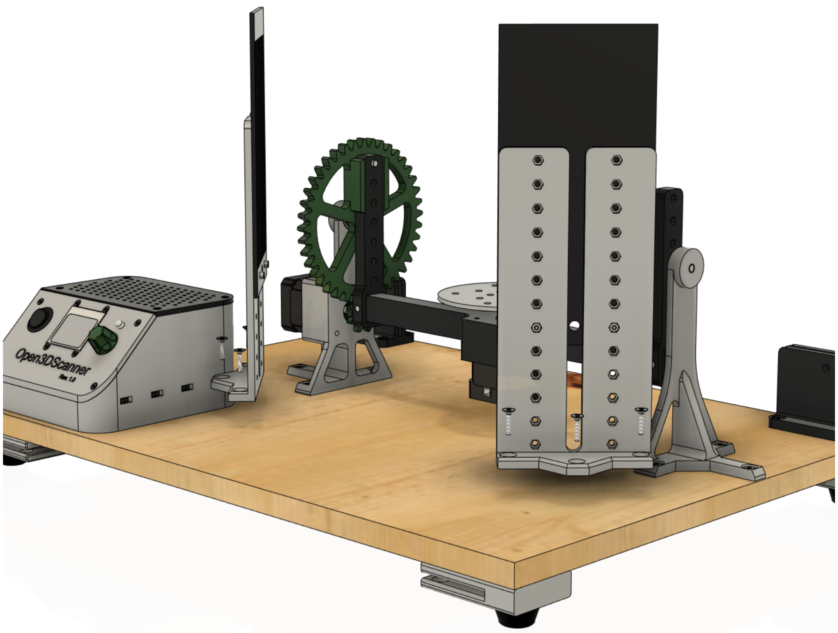


Figure 5.29 Step 22 & 23 of the Open3DScanner assembly

Part	Quantity
Cable-Holder	5
4.0×16 Countersunk Wood Screwt	10

Table 5.32 Required parts for step 24

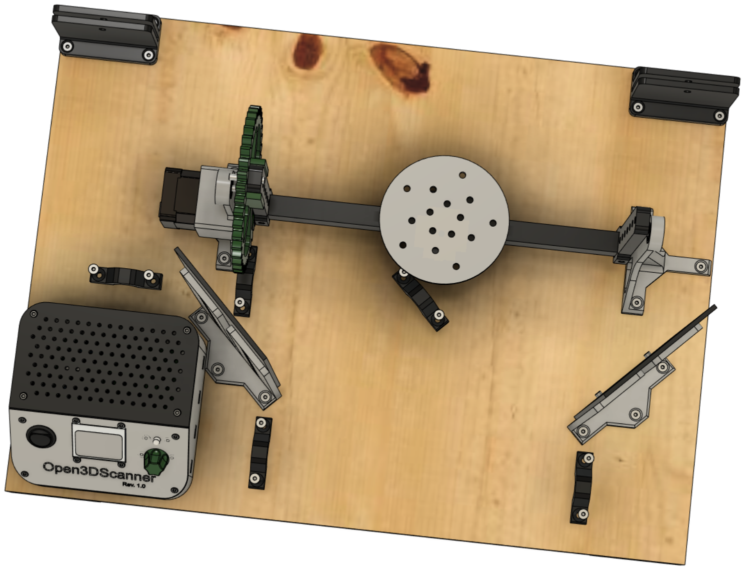
Instructions

Screw the cable holders onto the base plate.

Tip

For the exact alignment on the base plate, there is a detailed illustration at the end of the assembly instructions. All cables have to be routed for this step.

24
Step

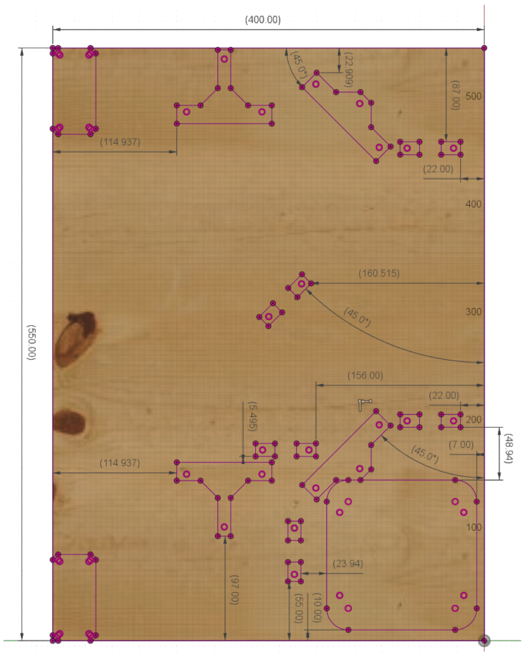


Fin

Figure 5.30 Step 24 & Fin of the Open3DScanner assembly

> **Tip**
> This illustration is an aid for aligning the components on the base plate. All values are given in millimeters. For the individual setup, the values can deviate slightly, especially the distance between the two stands of the 3D scanner. This is due to the fact that a certain tolerance is given by the steel bars. Figure B shows the dimensions of the back plate.

A Alignment



B Backplate

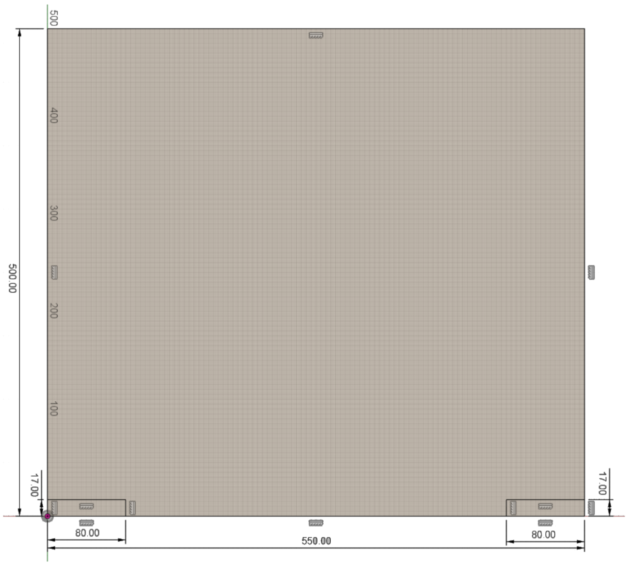


Figure 5.31 Component Alignment on Base Plate & Back Plate Size

6

Chapter

User Guide

This chapter describes how to use the fully assembled Open3D scanner to create 3D scans. First, the menu navigation of the Open3DScanner is explained and then the handling of the hardware.

6.1 Menu Navigation

If the Open3DScanner is switched on, the user sees the main menu on the display. It contains three sub-items which lead to the menus described in the following sections. The navigation within the menu is done by the knob to the right of the display. If it is rotated, the cursor position changes or the currently selected value is changed. To confirm a selection, press the knob.

6.1.1 Scan

The user has two options for starting a scan: Using a preset or creating a custom scan.

The presets are predefined scan settings that allow a quick and easy start of the scan.

If a custom scan is selected, the user must successively specify a series of settings that define the characteristics of the subsequent scan. The selected values are persisted and selected as default values for the next custom scan. This is done under the assumption that similar scans are frequently performed.

First, the user must determine how many images are to be taken each time the turntable on which the object to be scanned is mounted is rotated. Any value between 1 and 3200 (inclusive) can be used. 3200 corresponds to the maximum resolution of the stepper motors at 200 steps per revolution and 1/16 microstepping.

The next step is to select how far the turntable and the object are to be rotated on the x-axis during the scan. Any value between 1 and 359 (incluseive) degrees can be selected ℹ .

> ℹ **Reasonable Values**
> Even if there is a large range of values for the user to choose from, it often makes little sense to rotate the object further than 180 degrees around the x-axis. I usually use values in the range of 45 and 135.

Finally, the number of steps in which the rotation on the x-axis is divided must be selected. Each step corresponds to a complete rotation of the turntable with the initially selected number of steps. The value range always starts at 1 and has a maximum value that corresponds to the maximum resolution of the stepper motor for the previously selected rotation around the x-axis.

After all these parameters have been selected, the user is presented with a summary containing all selected values, the resulting number of images and the duration of the scan. At this point the scan can either be confirmed or aborted. When a preset is selected, the summary is presented directly to the user.

Before the actual scan is started, it must be selected whether the lights of the Open3DScanner are to be switched on or off during the scan. The user is now prompted via the display to position the used smartphone appropriately and (if not already done) to establish a Bluetooth connection with the scanner. The scan cannot be started until the necessary Bluetooth connection has been established. As soon as this has been done, the Turntable can be moved to the desired start position and the scan can be started via the knob.

During scanning, the current progress is shown on the display. It shows the number of recorded images, their total number and the remaining time. If the Bluetooth connection to the smartphone is lost during the scan, the scan is interrupted and the user is prompted to re-establish the connection. If the knob is pressed during the scan, the scan can be aborted or resumed after confirmation.

6.1.2 Settings

The settings menu is divided into categories that group the individual settings.

Scan

In this setting menu, you can set how many milliseconds the scanner stops after a motor movement or after taking a photo. This is necessary to allow the camera to focus images, but also to prevent possible vibrations of the scanner from blurring the images.

Display

In addition to the option of switching the backlight for the display on or off, the contrast of the display can be adjusted here.

Camera

Here you can select which camera type will be used during the scan. This is important because the control of various devices is different. Currently iOs and Android devices are supported.

Steppers

This menu allows you to adjust the behavior of the stepper motors. The selected stepper motor moves 15° back and forth while the settings are adjusted. This makes changes directly visible.

The direction of movement of the motors can be changed if necessary. In addition, it can be determined with how many RPM the motor should move, what the acceleration and deceleration curve should look like, and which maximum values should be used for acceleration and deceleration.

6.1.3 Debug

The debug menu contains options to test some features of the Open3DScanner. Currently it is possible to switch the connected lights on and off and to trigger the camera. The system does not check whether a Bluetooth connection is actually present, so this must be checked in advance on the connected device.

6.2 Hardware Handling

This section contains some information about the built scanner and the interaction with the device.

6.2.1 Object Mounting

For the scan it is necessary to attach the object to be scanned to the Turntable. The holes in the Turntable allow to fix the object with cable ties. But this can lead to artefacts in the scan, so I prefer to use adhesive putty which can be reused many times.

6.2.2 Scanner Adjustment

Depending on the size of the object to be scanned, it may be useful to adjust the height of the lights or the turntable. The parts have been designed to allow this. This makes it possible to move the rotation center of the x-axis near the center of the object to be scanned.

Thus, depending on the object, it can be ensured that it is always in the field of view of the camera, which is important for the automatic creation of images.

6.2.3 Status LED

The Open3DScanner has a bi-color LED, which is used to indicate the state of the scanner. The different state are explained below.

LED	State	Note
The LED lights green	Running	The scanner is turned on and ready for user interaction.
The LED lights yellow	Scanning	The scanner is performing a scan.
The LED flashes green very slowly	Scan finished	A scan has been finished successfully.
The LED flashes yellow slowly	Scan will continue	An interrupted scan will be continued soon.
The LED flashes red fast	No connection	The Bluetooth connection has been lost during scanning.

Table 6.1 Different states of the status LED

7

Chapter

Perform Scans

Even though I am not an expert in the field of photogrammetry, I have gathered some experiences which I would like to share here. The recommendations given in this chapter do not claim to be universally valid and are based on my individual experiences.

7.1 Lighting & Surface

A uniform illumination of the object to be scanned improves the quality of the scans significantly. In addition, glossy and transparent objects are much harder to scan compared to matte objects. For this reason, such objects should be treated (if possible) so that they have a better surface for the scanner. This can be done, for example, by spraying the objects with chalk spray.

7.2 Camera Settings

Most smartphone cameras have several features that automatically improve image quality. These range from autofocus and automatic white balance to AI functions that adjust color schemes based on recognized image content. Even if these functions provide better snapshots in everyday life, they are detrimental to the goals of photogrammetry. These functions make it more difficult to combine the images, which results in poorer scans.

For this reason, the functions should be deactivated as far as possible and photos should be taken with fixed settings. This is especially true for color temperature, auto white balance, and autofocus.

7.3 Augmented Reconstruction

Since the object to be scanned must be attached to the turntable, an area of the object is always not visible on the captured images. This means that a single scan is not sufficient to scan an object completely. This may not be a problem for objects that have a flat base on which they stand and when no texture is required for that area.

For many other objects this is a problem and multiple scans must be performed with different orientations of the object to capture the object from all sides.

At least the software Meshroom has the function augmented reconstruction for this case. It allows to add another data set (an additional scan) to an existing data set (the first scan) and to calculate a new overall result. This can repeated as often as needed and results in a tree like structure of the Meshroom processing graph.

That way it is often possible to create complete scans of objects. It may be necessary to filter some remaining artifacts of e.g. the Turntable of the reconstructed 3D model, but except from that you get a full scan and 3D model of the scanned object.

This feature can also be used if the reconstructed model lacks details in some areas. Just perform an additional (detailed) scan of the respective area and start an augmented reconstruction.

A

Appendix

Picture Series PCB Assembly

This chapter contains a series of pictures that show step by step how the individual components are soldered onto the PCB.

The images are completely uncommented, as they are not intended as instructions, but only as a reference.

More detailed information on assembling the circuit board can be found in section 5.2.

It is quite possible that individual components may look different in your case, this applies especially to resistors and heat sinks. This is not a problem as long as the components are purchased to match the BOM.

It is possible to use the pictures to determine the order in which the individual parts are assembled. However, this is not an instruction or obligation. Assemble the parts in the order that suits you best.

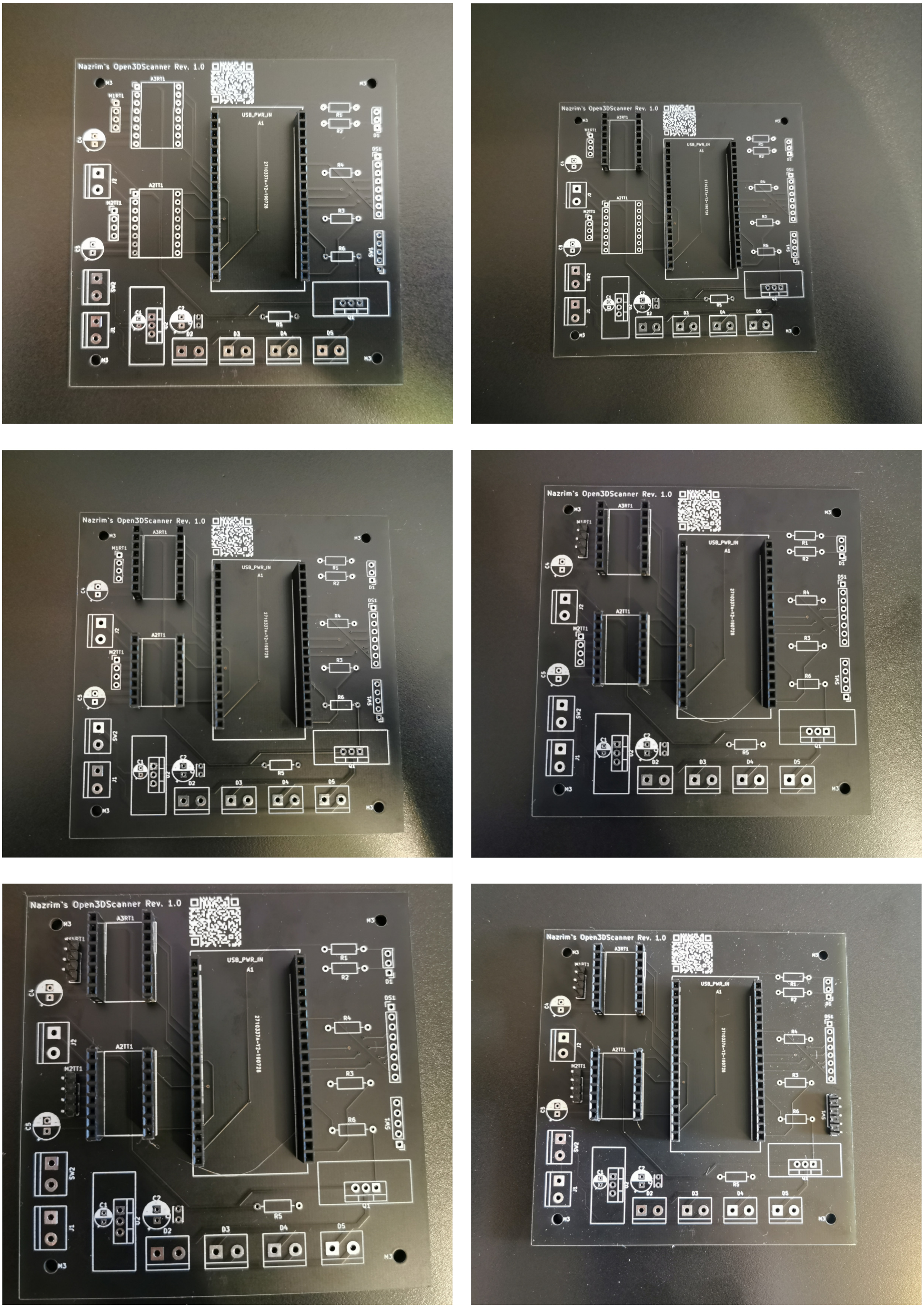


Figure A.1 Steps 1 to 6 of PCB assembly

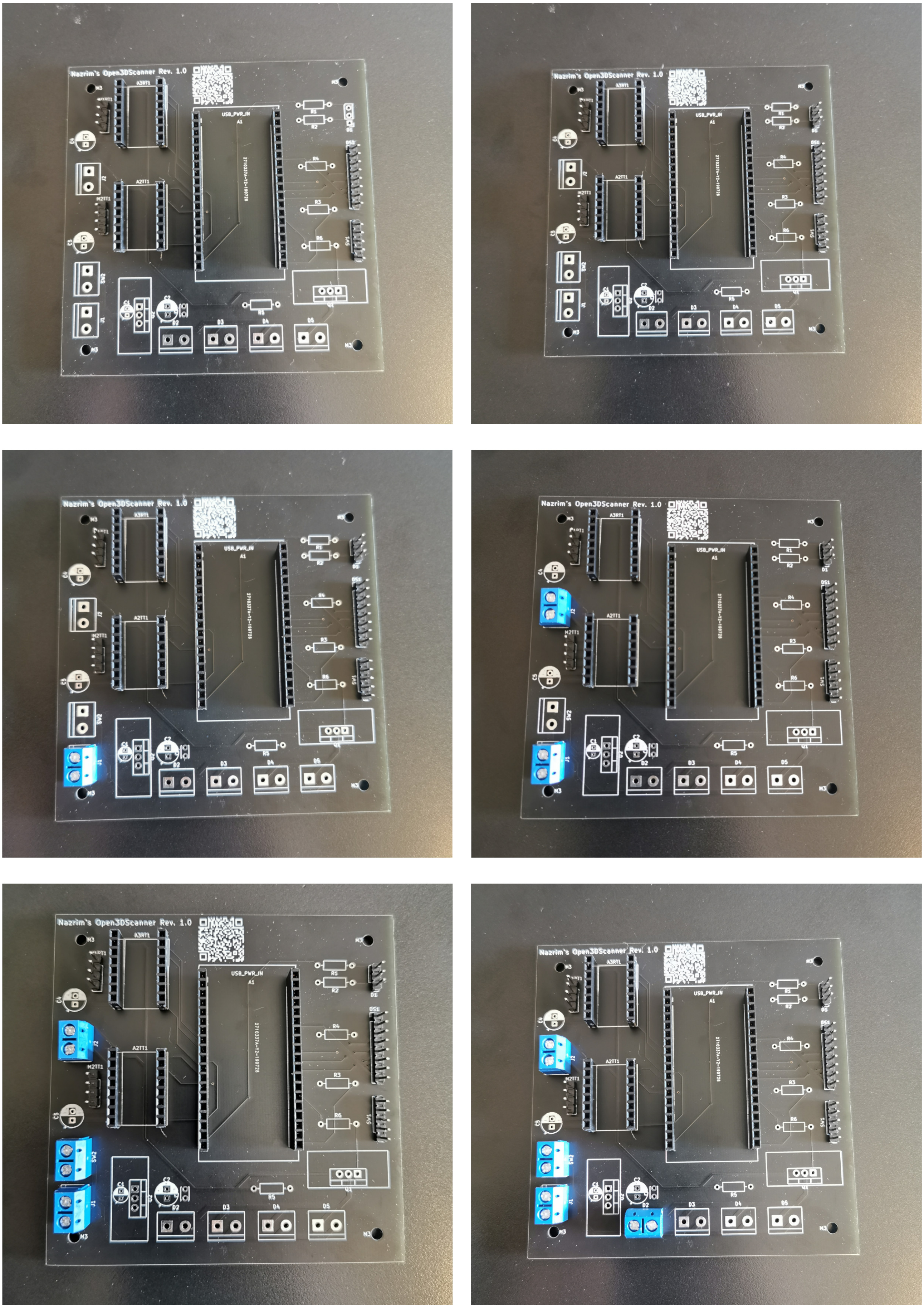


Figure A.2 Steps 7 to 12 of PCB assembly

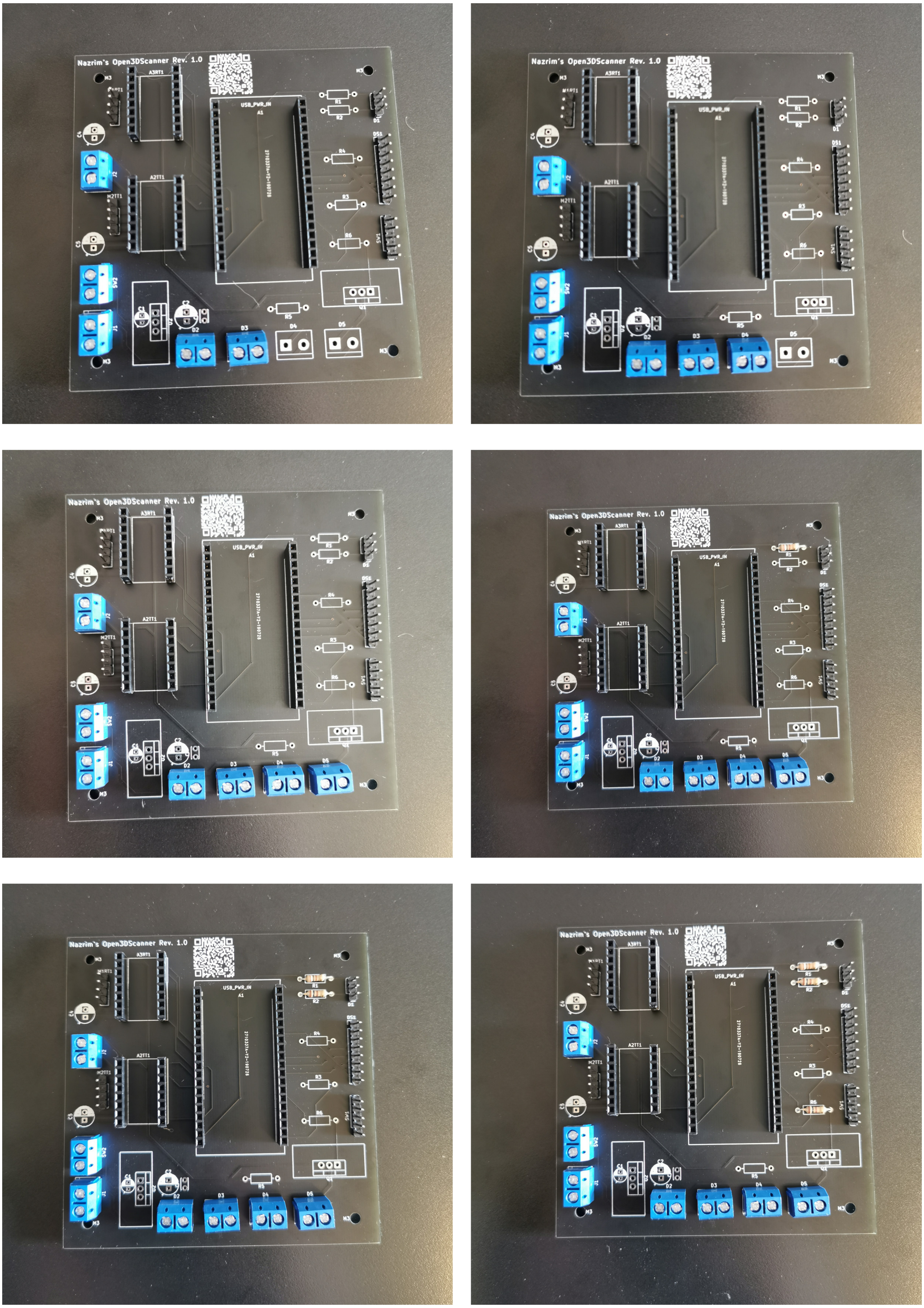


Figure A.3 Steps 13 to 18 of PCB assembly

Figure A.4 Steps 19 to 24 of PCB assembly

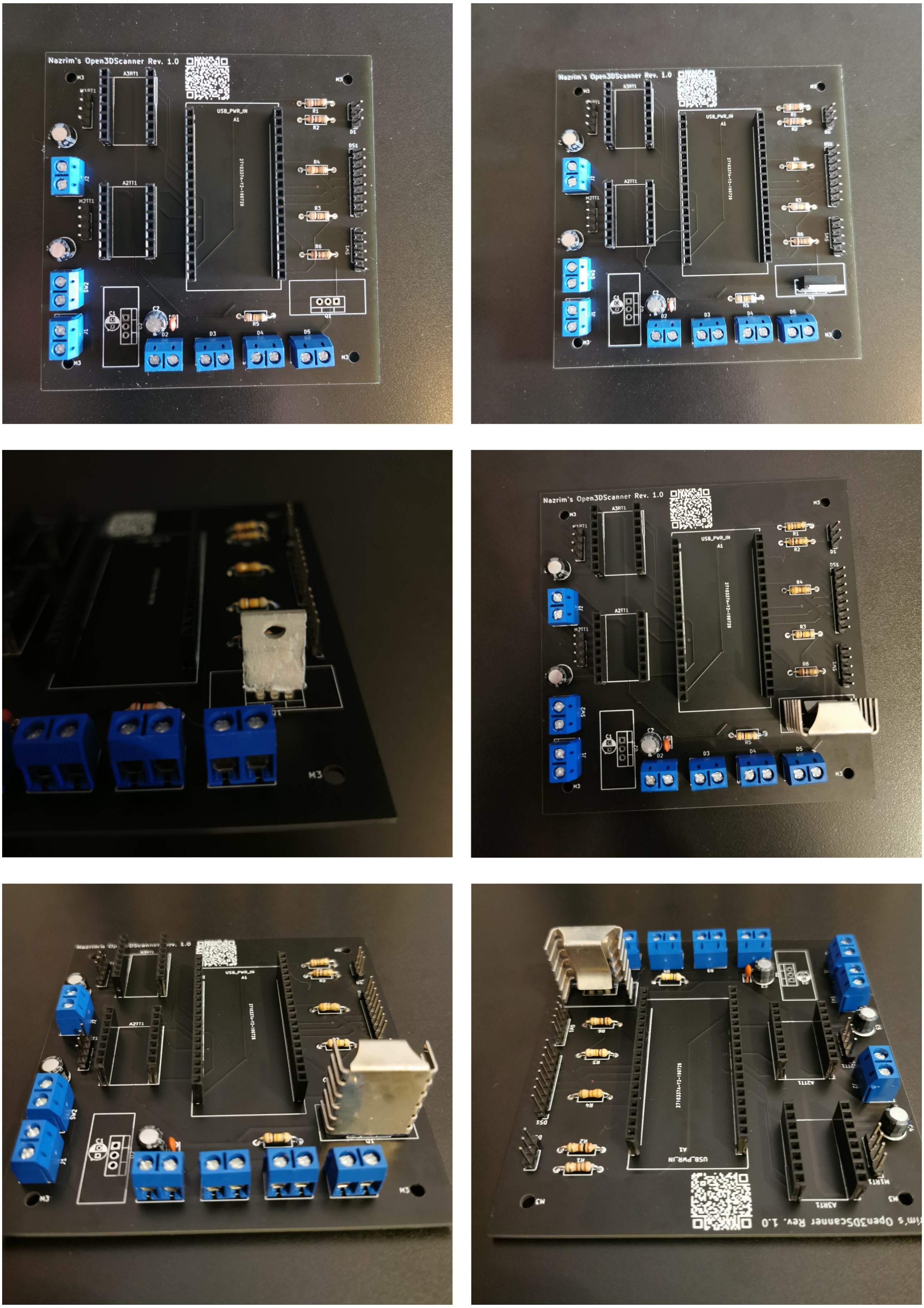


Figure A.5 Steps 25 to 30 of PCB assembly

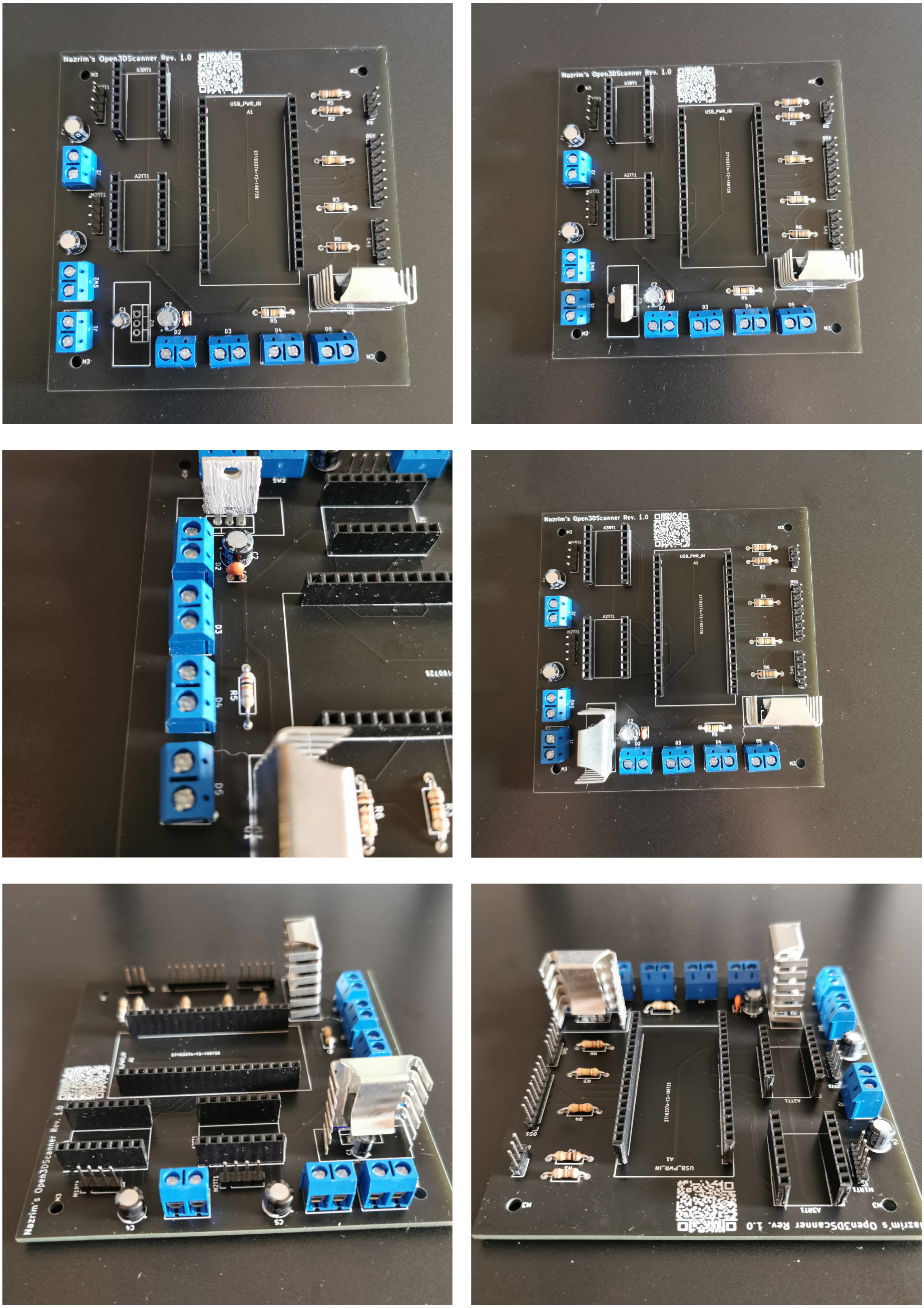


Figure A.6 Steps 31 to 36 of PCB assembly

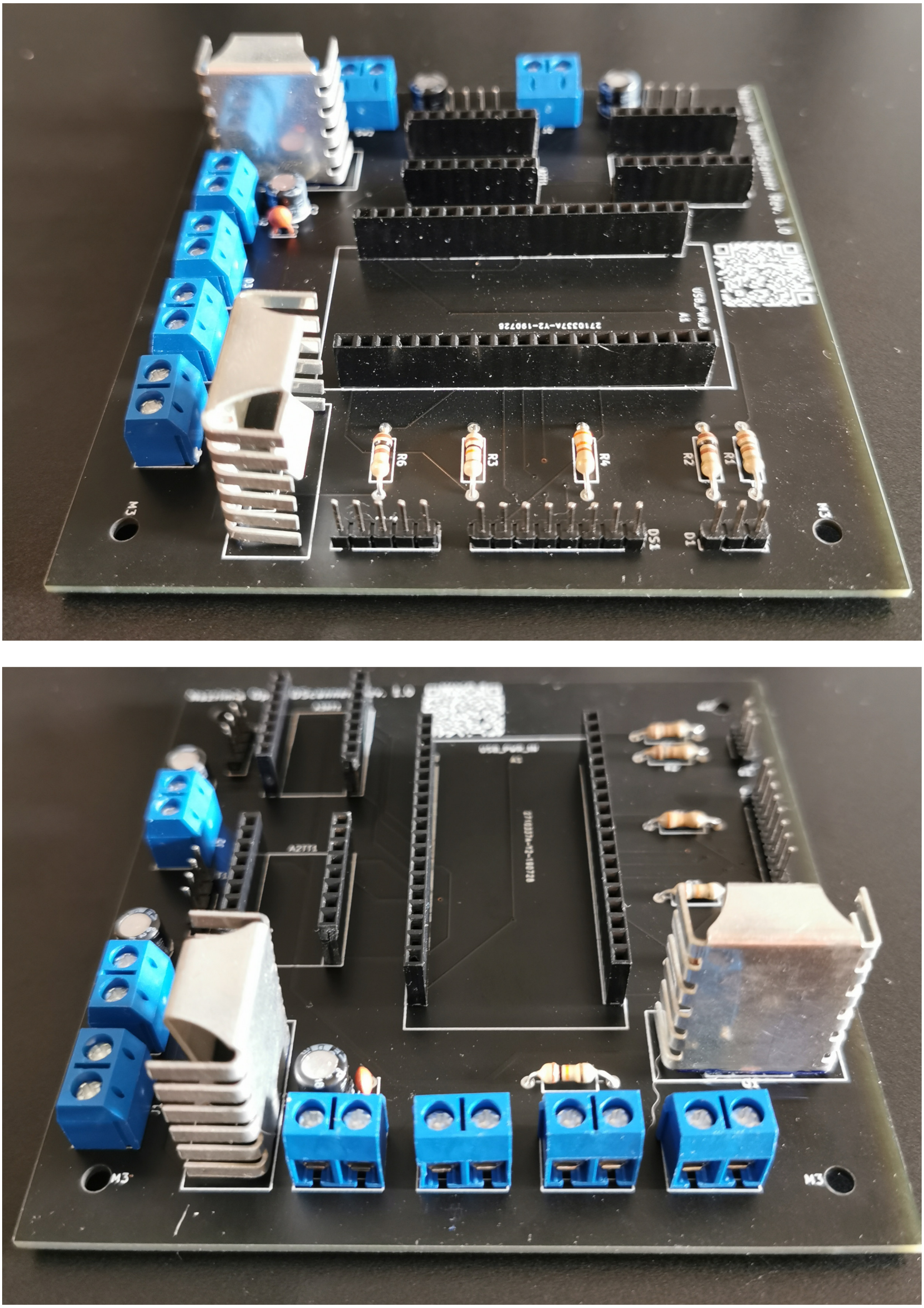


Figure A.7 Steps 37 to 38 of PCB assembly

B

Appendix

References

In case the document is available in printed form, this chapter contains all used links, which are provided with alternative display texts. This also allows these references to be followed.

The references are sorted alphabetically and not according to their appearance in the document.A few of the link texts are interpreted as special characters (e.g. XeLaTeX), so they do not appear at the expected position of the index, but at the beginning.

All links listed in this appendix were checked for validity on 20 October 2019.

2-Pin Screw Terminals 5mm Pitch:
https://www.sparkfun.com/products/8432, 19

3D Scanner Turntable:
https://www.hackster.io/daveclarke/3d-scanner-turntable-for-cell-phones-updated-64fdb8, 7

3D scanning — Wikipedia:
https://en.wikipedia.org/wiki/3D_scanning, 3

TeX Live:
https://www.tug.org/texlive/, 15

TeXstudio:
https://www.texstudio.org/, 15

XeLaTeX:
http://www.tug.org/xetex/, 15

59 N cm Nema 17 stepper motor:
https://www.omc-stepperonline.com/nema-17-stepper-motor/nema-17-bipolar-59ncm-84oz-in-2a-42x48mm-4-wires-w-1m-cable-and-connector-full-d-cut-shaft.html, 18

A4988 Stepper Driver:
https://www.pololu.com/product/1182, 18

A4988 Stepper Drivers:
https://www.pololu.com/product/1182, 26, 27

AliceVision:
https://alicevision.org/#photogrammetry, 3

ALL3DP:
https://all3dp.com/1/best-photogrammetry-software/, 4

ANIWAA:
https://www.aniwaa.com/comparison/3d-scanners/, 4

Arduino core for ESP32 WiFi chip:
https://github.com/espressif/arduino-esp32, 9

Arduino IDE:
https://www.arduino.cc/en/main/software, 9

arduino-menusystem:
https://github.com/jonblack/arduino-menusystem, 10

AST Bearings:
https://www.astbearings.com/free-3d-cad-models-for-bearings.html, 14

AZ-Delivery ESP32-DevKitC:
https://www.az-delivery.de/products/esp32-developmentboard, 18

Blender:
https://www.blender.org/, 8

Breadboard Adventures:
https://breadboardadventures.com/2016/02/14/a-square-and-triangle-wave-vco-part-ii-taming-the-voltage-spikes/, 10

cable lugs:
https://www.klauke.com/en/electrical/sectors-solutions/technical-reports/crimping-cable-lugs-correctly/, 29

Cad.Solidworks.Parts:
https://github.com/Hecatron-Machines/Cad.Solidworks.Parts, 14

Ciclop:
http://diwo.bq.com/en/einfuhrung-ciclop-und-horus/, 8

Digi-Key:
https://www.digikey.de//resources/3d-models, 14

Dupont cables:
https://www.instructables.com/id/Dupont-Crimp-Tool-Tutorial/, 29

Engineer PA-09:
http://www.engineer.jp/en/products/pa09e.html, 21

ESP32-DevKitC:
https://www.espressif.com/en/products/hardware/esp32-devkitc/overview, 18, 26, 30

ESP32Encoder:
https://github.com/madhephaestus/ESP32Encoder/, 10

Espressif ESP32 product page:
https://www.espressif.com/en/products/hardware/esp32/overview, 9

fischer elektronik - FK 245 MI 247 O product page:
https://www.fischerelektronik.de/web_fischer/en_GB/heatsinks/C02/Attachable%20heatsink/PG/